A spatially adaptive iterative method for a class of nonlinear operator eigenproblems

Jarlebring, Elias and Güttel, Stefan

2012

MIMS EPrint: 2012.113

Manchester Institute for Mathematical Sciences
School of Mathematics
The University of Manchester

Reports available from: http://eprints.maths.manchester.ac.uk/
And by contacting: The MIMS Secretary
School of Mathematics
The University of Manchester
Manchester, M13 9PL, UK

ISSN 1749-9097
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 in [17]. 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 [2] with its chebop functionality [11], although our function representation is entirely based on coefficients instead of function values. Our approach also allows for nonlinearities 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.

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 such PDE eigenvalue problems nonlinear in the eigenvalue parameter. Examples of such problems are given, e.g., in [30, 7, 5] and our Section 4.

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

\[
\begin{align*}
M(\lambda)f &= 0, \\
c_1(\lambda, f) &= 0, \\
& \vdots \\
c_k(\lambda, f) &= 0.
\end{align*}
\]

Here, \( M(\lambda) \) denotes a family of operators defined on a common domain \( D = D(M(\lambda)) \subset L^w_2([a, b]) \), independent of \( \lambda \in \mathbb{D}(\mu, r) \). The set \( L^w_2([a, b]) \) denotes functions which are square integrable on the interval \([a, b]\) with a suitable weight \( w \) (which we shall specify in Section 3). The functions \( c_i : \mathbb{C} \times D \to \mathbb{C} \) specify \( k \) conditions that need to be satisfied for an eigenpair \((\lambda, f)\).

We will assume that \( M(\lambda) \) can be represented as

\[
M(\lambda) = g_1(\lambda)L_1 + g_2(\lambda)L_2 + \cdots + g_m(\lambda)L_m,
\]

where the \( L_i : D \to L^w_2([a, b]) \) are closed linear operators and \( g_i : \mathbb{C} \to \mathbb{C} \) are given analytic functions defined in an open neighborhood of the disk \( \mathbb{D}(\mu, r) \). We also assume that the conditions \( c_i \) can be represented in a similar fashion. More precisely, we assume that for all \( i = 1, \ldots, k \) we have

\[
c_i(\lambda, f) = h_{i,1}(\lambda)C_{i,1}f + \cdots + h_{i,n}(\lambda)C_{i,n}f,
\]

where \( h_{i,j} : \mathbb{C} \to \mathbb{C} \) are analytic functions defined in a neighborhood of \( \mathbb{D}(\mu, r) \) and \( C_{i,j} : D \to \mathbb{C} \) are closed linear operators. We further assume that the conditions are given such that

†Dept. of Mathematics, NA group, KTH Royal Institute of Technology, 100 44 Stockholm, Sweden.
‡The University of Manchester, School of Mathematics, Alan Turing Building, M13 9PL, Manchester, United Kingdom. The work of S. G. was supported by Deutsche Forschungsgemeinschaft Fellowship GU 1244/1-1.
the problem (1.1) is well posed in the sense that its solutions $\lambda \in \mathbb{D}(\mu, r)$ are elements of a discrete set.

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

$$L_1 = \frac{\partial}{\partial x}, L_2 = \frac{\partial^2}{\partial x^2}, \ldots, L_m = \frac{\partial^m}{\partial x^m}.$$  

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 operator eigenvalue problem.

The algorithm we propose is closely related to the infinite Arnoldi method presented in [17]. The infinite Arnoldi method can, in principle, resolve 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 that of (1.1). There are a number of approaches available for the nonlinear matrix eigenvalue problem [22, 29, 1, 3, 12]. 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 adapt 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. We select a particular companion-like operator formulation having a structure that is suitable for the Arnoldi method [25] applied to the operator formulation, and our derivation does not require a spatial discretization at that 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 $M(\mu)$ 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 $L_i$-operations (like differentiation and pointwise multiplication) are also given in Section 3. In Section 4 we demonstrate our algorithm with 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 [2] with its chebop functionality [11]. The idea to carry out iterates with functions has been presented in other settings. A variant of GMRES for functions is given in [23] where the functions are represented using Chebfun [2]. See also the discussion of infinite-dimensional numerical linear algebra in [15].
Apart from the notation introduced above, we use the following conventions. Calligraphic style will be used to denote operators, in particular \( I \) will denote the identity operator and \( O \) will denote the zero operator. The set of one-dimensional (or two-dimensional) arrays of functions will be denoted \( L^w_2([a,b])^N \) (or \( L^w_2([a,b])^N \times k \)). The partial derivative with respect to \( \lambda \) will be denoted \( (\cdot)' \), the second partial derivative with respect to \( \lambda \) by \( (\cdot)'' \), etc.

2. Infinite Arnoldi in abstract PDE setting.

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

\[
M_N(\lambda) := M(\mu) + \frac{\lambda - \mu}{1!} M^{(1)}(\mu) + \cdots + \frac{(\lambda - \mu)^N}{N!} M^{(N)}(\mu),
\]

with the operators \( M^{(j)} \) being analogues of the \( j \)th derivative of \( M \) evaluated for \( \mu \),

\[
M^{(j)}(\mu) := g^{(j)}(1)(\mu)L_1 + g^{(j)}(2)(\mu)L_2 + \cdots + g^{(j)}(m)(\mu)L_m.
\]

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

\[
(2.1) \quad c_{i,N}(\lambda,f) := c_i(\mu,f) + \frac{\lambda - \mu}{1!} \left( \frac{\partial}{\partial \lambda} c_i(\lambda,f) \right)_{\lambda=\mu} + \frac{(\lambda - \mu)^2}{2!} \left( \frac{\partial^2}{\partial \lambda^2} c_i(\lambda,f) \right)_{\lambda=\mu} + \cdots + \frac{(\lambda - \mu)^N}{N!} \left( \frac{\partial^N}{\partial \lambda^N} c_i(\lambda,f) \right)_{\lambda=\mu}.
\]

We now consider the truncated operator eigenproblem

\[
(2.2a) \quad M_N(\lambda_N)f_N = 0,
\]

\[
(2.2b) \quad c_{i,N}(\lambda_N,f_N) = 0, \quad i = 1, \ldots, k
\]

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

THEOREM 2.1 (Convergence of operator Taylor-like expansion). Suppose that \( (\lambda_N,f_N) \) is a solution to (2.2) with \( |\lambda_N - \mu| < r \) for any \( N \). Moreover, suppose that the approximations are convergent, i.e., \( (\lambda_N,f_N) \to (\lambda_*,f_*) \). Also suppose \( L_i f_N \) and \( C_{i,j} f_N \) are convergent for any \( i,j \) as \( N \to \infty \). Then there exist positive constants \( \gamma \) and \( \beta < 1 \) independent of \( N \) such that

\[
\|M(\lambda_N)f_N\| \leq \gamma \beta^N
\]

\[
\|c_1(\lambda,f)\| \leq \gamma \beta^N
\]

\[
\vdots
\]

\[
\|c_k(\lambda,f)\| \leq \gamma \beta^N.
\]

Proof. Since the functions \( g_i \) \( (i = 1, \ldots, m) \) are assumed to be analytic in a neighborhood of \( D(\mu,r) \), 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

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

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

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

with \( |\lambda - \mu|/r \leq \hat{\beta} < 1 \). Consequently,

\[ \|\mathcal{M}(\lambda_N) f_N\| = \|\mathcal{M}(\lambda_N) f_N - \mathcal{M}_N(\lambda_N) f_N\| \]
\[ = \|R_{1,N}(\lambda_N) \mathcal{L}_1 f_N + \cdots + R_{m,N}(\lambda_N) \mathcal{L}_m f_N\| \]
\[ \leq \max_{i=1,\ldots,m} \frac{M_{i,r}\hat{\beta}^{N+1}}{1 - \hat{\beta}} \|\mathcal{L}_i f_N\|. \quad (2.3) \]

The conclusion about the bound on \( \|\mathcal{M}(\lambda_N) f_N\| \) 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 constant \( \beta \) and \( \gamma \) are formed by the maximum of the computed bounds which are all of the form (2.3).

\( \Box \)

**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 \((\lambda_N, f_N)\) converges to a pair \((\lambda, f)\). Despite this, note that the operators under consideration are not necessarily bounded, and therefore Theorem 2.1 does not necessarily imply that \( \|\mathcal{M}(\lambda) f\| = 0 \). For example suppose \( \mathcal{M}(\lambda) = \frac{d}{dx} \) and consider a situation where \((\lambda_N, f_N)\) is a solution to the truncated problem and \( f_N(x) = f_*(x) + \frac{\pi}{2N} \sin(Nx) \). Then \( f_N \to f_* \) but \( \mathcal{M}(\lambda_N) f_N \) 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 \( f_\ast \) 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 \( \lambda \). This is essentially an operator version of what is commonly called a *polynomial eigenvalue problem* [26, 20], and such problems are often analyzed and solved with the *companion linearization technique*. There are many types of companion linearizations [21], 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^N_{2}\([a,b]\) \) such that

\[ \mathcal{A}_N \begin{pmatrix} f_1 \\ \vdots \\ f_N \end{pmatrix} := \begin{pmatrix} \mathcal{M}(\mu) & I & & \\ & \ddots & \ddots \\ & & \ddots & I \\ \mathcal{M}(\mu) f_1 \\ \vdots \\ f_N \end{pmatrix} \begin{pmatrix} f_1 \\ \vdots \\ f_N \end{pmatrix} = \begin{pmatrix} \mathcal{M}(\mu) f_1 \\ f_2 \\ \vdots \\ f_N \end{pmatrix}. \quad (2.4) \]
and an operator $B_N$ with action defined by

\[(2.5) \quad B_N \left( \begin{array}{c} f_1 \\ \vdots \\ f_N \end{array} \right) := \left( \begin{array}{cccc} -M^{(1)}(\mu) & -\frac{1}{2}M^{(2)}(\mu) & \cdots & -\frac{1}{N}M^{(N)}(\mu) \\ \mathcal{I} & \frac{1}{2}\mathcal{I} & & \\ & \ddots & \ddots & \\ & & \frac{1}{N-1}\mathcal{I} & 0 \end{array} \right) \left( \begin{array}{c} f_1 \\ \vdots \\ f_N \end{array} \right). \]

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

\[(2.6a) \quad A_N\varphi = (\lambda - \mu)B_N\varphi \]
\[(2.6b) \quad c_i(\mu, \varphi_1) + c'_i(\mu, \varphi_2) + \cdots + c_i^{(N)}(\mu, \varphi_N) = -\frac{\lambda - \mu}{N} c_i^{(N)}(\mu, \varphi_N), \quad i = 1, \ldots, k. \]

This particular companion linearization is useful because, for any $M \geq N$, the leading $N \times N$ blocks in the operators $B_M$ and $A_M$ consist precisely of $A_N$ and $B_N$. This will be implicitly exploited in Section 2.3. The companion operator problem (2.6) is equivalent to the $M \times M$-problem (2.2) in the following sense.

**Theorem 2.3.** Consider $\varphi = (\varphi_1, \ldots, \varphi_N)^T \in L^2([a,b]^N$ with $\varphi_1 = f$. The companion linearization (2.6) and the truncated Taylor expansion (2.2) are equivalent in the sense that the following two statements are equivalent.

a) The pair $(\lambda, \varphi)$ is a solution to (2.6).
b) The pair $(\lambda, f)$ is a solution to (2.2).

**Proof.** Consider a solution $\varphi = (\varphi_1, \ldots, \varphi_N)^T$ to (2.6). Then the last $N - 1$ rows of (2.6a) imply that

\[(2.7) \quad \varphi_2 = (\lambda - \mu)\varphi_1, \quad \varphi_3 = \frac{1}{2}(\lambda - \mu)\varphi_2 = \frac{1}{2}(\lambda - \mu)^2\varphi_1, \quad \varphi_4 = \frac{1}{3}(\lambda - \mu)\varphi_3 = \frac{1}{3}(\lambda - \mu)^3\varphi_1, \quad \vdots \quad \varphi_N = \frac{(-\mu)^{(N-1)}}{(N-1)!}\varphi_1. \]

By inserting (2.7) into the first row in (2.6a) we have

\[(2.8) \quad 0 = M(\mu)\varphi_1 + (\lambda - \mu)M^{(1)}(\mu)\varphi_1 + \frac{(\lambda - \mu)^2}{2!}M^{(2)}(\mu)\varphi_1 + \cdots + \frac{(\lambda - \mu)^N}{N!}M^{(N)}(\mu)\varphi_1. \]

Similarly, (2.6b) implies with (2.7) and the linearity of $c_i(\lambda, f)$ with respect to $f$ that

\[(2.9) \quad 0 = c_i(\mu, \varphi_1) + (\lambda - \mu)c'_i(\mu, \varphi_1) + \frac{(\lambda - \mu)^2}{2!}c''_i(\mu, \varphi_1) + \cdots + \frac{(\lambda - \mu)^N}{N!}c^{(N)}_i(\mu, \varphi_1). \]

The forward implication now follows from the fact that (2.8) is identical to (2.2a) and (2.9) is identical to (2.2b).

In order to show the converse, suppose $f$ is a solution to (2.2) and define $\varphi_1 = f$ and $\varphi_i$ ($i = 2, \ldots, N$) as in (2.7). The relation (2.8) holds because of (2.2), and a similar argument is used for the conditions (2.9). □
2.3. Infinite Arnoldi Algorithm. Now note that (2.6) 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 [25] is one of the most popular approaches. We will now show how to formulate the Arnoldi method* for (2.6), exploit the structure and thereby avoiding the traditional approach of first discretizing the problem. This is similar to the “Taylor version” of the infinite Arnoldi method for nonlinear matrix eigenvalue problems described in [17].

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 [18, 8]. In order to apply the Arnoldi algorithm on the formulation (2.6) we will need

- a procedure which solves the operator problem

\[
A_N \varphi = B_N \psi
\]

\[
(2.10b) \quad c_i(\mu, \varphi_1) + \cdots + c_i^{N-1}(\mu, \varphi_N) = \frac{1}{N} c_i^{(N)}(\mu, \psi_N), \quad i = 1, \ldots, k
\]

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 $A_N$ and $B_N$ is particularly 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

\[
(2.11) \quad \psi = \begin{pmatrix}
\psi_1 \\
0 \\
\vdots \\
0
\end{pmatrix}
\]

where $\psi_1 \in L_2^w([a,b])$. In the first step of the Arnoldi method we need to solve (2.10). By inspection of the structure of $A_N$ and $B_N$, the solution will be of the form

\[
\varphi = \begin{pmatrix}
\varphi_1 \\
\psi_1 \\
0 \\
\vdots \\
0
\end{pmatrix}
\]

Hence, the action corresponding to the non-zero part of the solution of (2.10) is independent of $N$ if we start with a vector consisting of just one leading non-zero block. More generally, the solution of (2.10) can be characterized as follows.

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

\[
(2.12) \quad \psi = \begin{pmatrix}
\psi_1 \\
\vdots \\
\psi_p \\
0 \\
\vdots \\
0
\end{pmatrix}
\]

*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.
where $\psi_1, \ldots, \psi_p \in L^w_2([a,b])$. Consider the operators $A_N$ and $B_N$ defined by (2.4) and (2.5) for any $N > p$. Suppose $\varphi \in L^w_2([a,b])^N$ is a solution to the operator problem (in $L^w_2([a,b])^N$):

\begin{align}
(2.13a) & \quad A_N \varphi = B_N \psi \\
(2.13b) & \quad 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.
\end{align}

Then this solution satisfies

\begin{equation}
\varphi = \begin{pmatrix}
\varphi_1 \\
\frac{1}{p} \psi_1 \\
\vdots \\
\frac{1}{p} \psi_p \\
0 \\
\vdots \\
0
\end{pmatrix}
\end{equation}

where $\varphi_1 \in L^w_2([a,b])$ is the solution to the operator problem (on $L^w_2([a,b])$)

\begin{align}
(2.15a) & \quad M(\mu) \varphi_1 = -M(1)(\mu) \psi_1 - \frac{1}{2} M(2)(\mu) \psi_2 - \cdots - \frac{1}{p} M(p)(\mu) \psi_p \\
(2.15b) & \quad c_i(\mu, \varphi_1) = -c_i(\mu, \psi_1) - \frac{1}{2} c_i(\mu, \psi_2) - \cdots - \frac{1}{p} c_i(\mu, \psi_p), \quad i = 1, \ldots, k.
\end{align}

Proof. The last $N-1$ rows of (2.13a) implies that $\varphi$ has the structure (2.14). Equation (2.15a) follows directly from the insertion of (2.14) and (2.12) into the first row of (2.13a). Note that the terms $M^{(j)}(\mu) \psi_j$ vanish for $j > p$ since $\psi_j = 0$. Similarly, by inserting into (2.13b) the structure of $\varphi$ given in (2.14), and $\psi$ given in (2.12), several terms vanish and (2.15b) is verified. \[\square\]

From the above theorem we make the following key observation.

The non-zero part of the solution to (2.13) for a function $\psi$ with structure (2.12) is independent of $N$ as long as $N > p$.

By only considering functions of the structure (2.12) we can, in a sense, take $N \to \infty$ without changing the non-zero part of the solution. With $N \to \infty$ the truncation error in the Taylor expansion vanishes and (2.2) corresponds to the original problem (1.1) (under the conditions stated in Theorem 2.1 and Remark 2.2).

The key idea is to start the Arnoldi algorithm with an array of functions of the structure (2.11). Due to the fact that the Arnoldi method essentially involves solutions of (2.13) at every iteration, combined with a Gram–Schmidt orthogonalization, all arrays of functions will be of the structure (2.12). 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

(2.16) $\quad V = \begin{pmatrix}
v_{1,1} & v_{1,2} & \cdots & v_{1,k} \\
0 & v_{2,2} & \cdots & \vdots \\
0 & \vdots & \ddots & \vdots \\
0 & \cdots & 0 & v_{k,k}
\end{pmatrix} \in L^w_2([a,b])^{k \times k},$
where \( v_{i,j} \in L^w_w([a, b]) \).

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

\[
(f, g)_w := (f_1, g_1)_w + \cdots + (f_N, g_N)_w,
\]

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, referred to as the infinite Arnoldi method (for nonlinear operator eigenproblems).

**Remark 2.5 (Existence).** Algorithm 1 uniquely defines a sequence function iterates only if there exists a unique solution to (2.15). 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 (2.15) will be guaranteed by the well-posedness of the considered differential equations.

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

**Require:** Starting function \( v_{1,1} \in L^w_w([a, b]) \)

1: \( v_{1,1} = v_{1,1}/\sqrt{\langle v_{1,1}, v_{1,1} \rangle_w} \)

2: for \( k = 1, 2, \ldots, k_{\text{max}} \) do

3: Compute \( \varphi_2, \ldots, \varphi_{k+1} \) from (2.14) where \( \psi_1 = v_{1,k}, \ldots, \psi_k = v_{k,k} \) and \( p = k \).

4: Solve the inhomogeneous differential equation (2.15) for \( \varphi_1 \) with \( \psi_1 = v_{1,k}, \ldots, \psi_k = v_{k,k} \) and \( p = k \).

5: for \( i = 1, \ldots, k \) do

6: \( h_{i,k} = \langle \varphi_i, v_{1,i} \rangle_w + \cdots + \langle \varphi_i, v_{i,i} \rangle_w \)

7: for \( j = 1, \ldots, i \) do

8: \( \varphi_j = \varphi_j - h_{i,k} v_{j,i} \)

9: end for

10: end for

11: \( h_{k+1,k} = \sqrt{\langle \varphi_1, \varphi_1 \rangle_w + \cdots + \langle \varphi_{k+1}, \varphi_{k+1} \rangle_w} \)

12: for \( j = 1, \ldots, k + 1 \) do

13: \( v_{j,k+1} = \varphi_j/h_{k+1,k} \)

14: end for

15: end for

16: Compute the eigenvalues \( \{\mu_i\}_{i=1}^k \) of the Hessenberg matrix with elements \( H_{i,j} = h_{i,j}, \ i, j = 1, \ldots, k_{\text{max}} \).

17: Return eigenvalue approximations \( \{1/\mu_i\}_{i=1}^k \) of (1.1)

3. Multi-level space resolution. The main computational cost in a practical implementation of our nonlinear eigensolver (Algorithm 1) lies in the solution of a differential equation (2.15) 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 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 [2] with its chebop functionality [11], 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 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 space 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 \(L^2_w\) inner product

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

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 \([27, 4, 10]\), where a polynomial \(P_m\) is represented by its function values on a Chebyshev grid with nodes \(y_j = \cos(\pi j/m)\) \((j = 0, 1, \ldots, m)\), we prefer here 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 inner product between coefficient vectors (with the 0th coefficients divided by \(\sqrt{2}\)) corresponds to a weighted \(L^2_w\) inner product between the corresponding polynomials:

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

Note that our infinite Arnoldi method is rich in inner product computations, and this relation allows for an efficient implementation.
3.2. 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 [19], see also [4, Chapter 21], [16, 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 $P_m(x) = \sum_{j=0}^{m} c_j T_j(x)$. For the identities involving Chebyshev polynomials used in the derivation we refer to [13, Section 3].

- **Differentiation:** By the relation for the derivative of a Chebyshev polynomial $T_j(y)$,
  \[
dy{dy} T_j = \begin{cases} 
  jT_0 + 2j(T_2 + T_4 + \cdots + T_{j-1}), & \text{if } j \text{ is odd;} \\
  2j(T_1 + T_3 + \cdots + T_{j-1}), & \text{if } j \text{ is even,}
\end{cases}
\]
  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
  \[
  M_m(Q_n) = \frac{1}{2} \begin{bmatrix}
  d_0 & 0 & 0 & 0 & \cdots \\
  d_1 & 2d_0 & d_1 & d_2 & \cdots \\
  d_2 & d_1 & 2d_0 & d_1 & \cdots \\
  d_3 & d_2 & d_1 & 2d_0 & \cdots \\
  \vdots & \vdots & \vdots & \vdots & \ddots
\end{bmatrix}
  + \begin{bmatrix}
  d_0 & d_1 & d_2 & d_3 & \cdots \\
  d_1 & d_2 & d_3 & \cdots \\
  d_2 & d_3 & \cdots \\
  d_3 & \cdots \\
  \vdots & \vdots & \ddots
\end{bmatrix} \in \mathbb{C}^{(m+n+1) \times (m+n+1)},
\]
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

\[
R_m = \text{diag}(1, -1, 1, -1, \ldots) \in \mathbb{R}^{(m+1) \times (m+1)}
\]

maps the coefficients of \( P_m(x) \) to the coefficients of the “reversed” (right-to-left) polynomial \( P_m(a + b - x) \).

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 of index exceeding the degree of a polynomial are set to 0.) Typically the matrix \( A_m \) is not invertible and \( k \) boundary conditions need to be imposed when solving for \( P_m \) given \( Q_m \) (abbreviated as “\( A_m P_m = Q_m \)” with a slight abuse of notation). 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 (boundary bordering), e.g.,

\[
\begin{bmatrix}
1, -1, 1, -1, \ldots, (-1)^{m+1}
\end{bmatrix}
\quad \text{Dirichlet b.c. on the left}
\]

\[
\begin{bmatrix}
1, 1, \ldots, 1
\end{bmatrix}
\quad \text{Dirichlet b.c. on the right}
\]

\[
\begin{bmatrix}
2 \frac{a}{b-a} [0, 1, 2, 4, \ldots, (1)^m (m-1)^2]
\end{bmatrix}
\quad \text{Neumann b.c. on the left}
\]

\[
\begin{bmatrix}
2 \frac{a}{b-a} [0, 1, 2, 4, \ldots, (m-1)^2]
\end{bmatrix}
\quad \text{Neumann b.c. on the right},
\]

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

\[
Q_m(x) - AP_m(x) = \sum_{j=m+1-k}^{\infty} \epsilon_j T_j(y),
\]

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, \( \epsilon_m = - \sum_{j=m+1-k}^{\infty} \epsilon_j T_j(y) \). 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., [28, Chapter 8]): for some \( \rho > 1 \) and \( C > 0 \) one has

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

If \( f \) has no singularities too close to \([a, b]\), then \( \rho \) is large enough to achieve a fast uniform convergence of \( P_m \) towards \( f \), indicated by a rapid decay of \( P_m \)'s Chebyshev coefficients \( c_0, c_1, \ldots, c_m \). This fact is exploited in the Chebfun system with its chebop functionality for
solving operator equations [11], and we will employ a similar rule of thumb: assume that the weighted \( L^w_2 \) error of a Chebyshev approximant \( P_m \) is of about the same order as its trailing Chebyshev coefficient \( c_m \) (see also [4, 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 \( A_m \) (the spectral matrix with boundary conditions included) and solve the linear system \( A_m P_m = Q_m \) 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

\[
\| v_{i,j} - P_{i,j} \|_w \leq tol \| P_{i,j} \|_w,
\]

where \( tol = 2 \times 10^{-16} \). These polynomial representations are stored in a two-dimensional “cell array” (cf. (2.16))

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

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 \( u : [0, \pi] \times [-\tau, +\infty) \to \mathbb{R} \),

\begin{align*}
(4.1a) \quad u_t(x, t) &= u_{xx}(x, t) - u(x, t - \tau), \\
(4.1b) \quad u(0, t) &= 0, \\
(4.1c) \quad u(\pi, t) &= 0,
\end{align*}

an example which has also been considered in [6, formula (112)]. The linear operator associated with (4.1) is

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

with boundary conditions

\[
\begin{align*}
c_1(\lambda, f) &= f(0), \\
c_2(\lambda, f) &= f(\pi).
\end{align*}
\]
In the implementation of our method we need to provide the derivatives of (4.2), which in this case are explicitly given as

\[ M^{(1)}(\mu) = -I - \tau e^{-\tau \mu} I \]

\[ M^{(k)}(\mu) = (-\tau)^k e^{-\tau \mu} I, \quad k \geq 2. \]

Consequently, in every iteration of our algorithm we need to solve (2.15) which reduces to solving

\[ (4.3) \]

\[ (-\mu I + \frac{\partial^2}{\partial x^2} - e^{-\tau \mu} I) \phi_1 = (1 + \tau e^{-\tau \mu})\psi_1 - \frac{1}{2}(-\tau)^2 e^{-\tau \mu} \psi_2 - \cdots - \frac{1}{p}(-\tau)^p e^{-\tau \mu} \psi_p \]

for \( \varphi_1 \) with boundary condition \( \varphi_1(0) = \varphi_1(\pi) = 0 \).

We have in this first example selected \( M(\lambda) \) 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 \( \gamma \) 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 \( \lambda \) by using the Lambert W function [9]. We find that the eigenvalues of the nonlinear operator eigenvalue problem are given by

\[ (4.4) \]

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

for any \( j \in \mathbb{N}_+ \) and any \( \ell \in \mathbb{Z} \) where \( W_\ell \) is the \( \ell \)th branch of the Lambert W function. Note that different eigenvalues can have the same eigenfunction, as the eigenfunction does not depend on \( \ell \). 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 \( \varphi_1 \) was a random polynomial of degree 5. Figure 4.1 (a) also shows the approximate eigenvalues after 60 iterations of the infinite Arnoldi method, and Figure 4.1 (b) shows the 10 approximate eigenfunctions \( f \) to which this method converged first. (Each two of these eigenfunctions coincide.)

The error norm for each of the 10 approximate eigenpairs 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 order 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 \( ||M(\lambda)f|| \) for each of the 10 approximate eigenpairs \( (\lambda, f) \) 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 show 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.

### 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 \( u : [0, L] \times [-\tau, \infty) \to \mathbb{R} \),

\[
\begin{align*}
\text{(4.5a)} & \quad u_{tt}(x,t) = c^2 u_{xx}(x,t), \\
\text{(4.5b)} & \quad u(0, t) = 0, \\
\text{(4.5c)} & \quad u_x(L, t) = \alpha u_t(L, t - \tau),
\end{align*}
\]

where \( c \) is the wave speed, \( \tau \) is the delay and \( \alpha \) corresponds to a feedback law. See [14], [30] and references therein for PDEs with delays and in particular the wave equation. In our setting, the eigenvalues associated with (4.5) are described by the \( \lambda \)-dependent operator

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

with \( \lambda \)-dependent boundary conditions,

\[
\begin{align*}
\text{c}_1(\lambda, f) & = f(0) \\
\text{c}_2(\lambda, f) & = f'(L) - \alpha \lambda e^{-\tau \lambda} f(L).
\end{align*}
\]

We now provide the implementation details for this example by specifying how to set up the differential equation (2.15). First note that

\[
\begin{align*}
\mathcal{M}^{(1)}(\mu) & = 2 \mu I, \\
\mathcal{M}^{(2)}(\mu) & = 2I.
\end{align*}
\]

In our algorithm we require the derivatives of the boundary condition with respect of \( \lambda \), which are explicitly given for \( k > 0 \) by

\[
\begin{align*}
\text{c}^{(k)}_1(\mu, f) & = 0, \\
\text{c}^{(k)}_2(\mu, f) & = -\alpha f(L)e^{-\tau \mu} (-\tau)^{k-1}(k - \tau \mu).
\end{align*}
\]

Hence, the specialization of (2.15) to this example, is for \( p = k > 1 \),

\[
\begin{align*}
\text{(4.6a)} & \quad \mu^2 \varphi_1(x) - c^2 \varphi''_1(x) = -2\mu \psi_1(x) - \frac{1}{2} 2\psi_2(x) \\
\text{(4.6b)} & \quad \varphi_1(0) = 0 \\
\text{(4.6c)} & \quad \varphi'_1(L) - \alpha \mu e^{-\tau \mu} \varphi_1(L) = \alpha e^{-\tau \mu} \left( \sum_{k=1}^{\infty} \frac{1}{k} \psi_k(L)(-\tau)^{k-1}(k - \tau \mu) \right)
\end{align*}
\]

where the functions \( \psi_1, \ldots, \psi_k \) are given and \( \varphi_1 \in L_1([a, b]) \) needs to be computed. When \( p = k = 1 \), i.e., in the first iteration, the term \( \psi_2 \) should be set to zero in the inhomogeneous term in (4.6a), whereas (4.6b) and (4.6c) remain the same for \( p = k = 1 \). Note that (4.6) is just a second order inhomogeneous second order 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 \( ||\mathcal{M}(\lambda)f|| \) for the first 10 approximate eigenpairs with \( \lambda \) closest to the target \( \mu = -1 \). 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 \( (\lambda, f) \) are resolved nicely while the degree of the Arnoldi vectors grows moderately.
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. We use an interval \([a, b] = [0, \pi]\) and an operator defined by

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

with boundary conditions

\[
\begin{align*}
c_1(\lambda, f) &= f(0) \\
c_2(\lambda, f) &= \lambda f(\pi) - f'(\pi).
\end{align*}
\]

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 \(\mu = 10\) be the target, such that the algorithm is expected to eventually find all eigenvalues in the disk \(\mathbb{D}(10, 15)\).
The derivatives of the operator with respect to $\lambda$ are given by

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

\[
\mathcal{M}^{(k)}(\lambda) = -i(-2)^{-k}(1 \cdot 3 \cdot 5 \cdots (2k - 3))(\lambda - \sigma_1)^{-(2k-1)/2}\mathcal{R}
\]

\[
- i(-2)^{-k}(1 \cdot 3 \cdot 5 \cdots (2k - 3))(\lambda - \sigma_2)^{-(2k-1)/2}\sin(x)\frac{\partial}{\partial x}, \quad k > 1
\]

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 more detailed investigation of the convergence behavior of the infinite Arnoldi method will be subject of future work.

**FIG. 4.3. The artificial example**
5. Concluding remarks. A key contribution of this paper is the formulation of an Arnoldi-type iteration for solving nonlinear operator eigenproblems. Our approach relies on a function representation of the iterates in the infinite Arnoldi algorithm whose resolutions are adapted automatically such that the eigenpairs are computed with an accuracy not too far from machine precision. It would be interesting to see if the spectral method recently proposed in [24] 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., 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 could not be discussed in this paper. For PDEs with complicated geometries, the finite-element method (FEM) is a popular approach for 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 $\lambda$ can be handled in a general fashion, and their effect is simply propagated into the differential equation (2.15), i.e., the equation to be solved at every iteration. Some boundary conditions with nonlinearities in $\lambda$ 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 also wish to point out that in [17] 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.

Acknowledgments. The first author gratefully acknowledges the support of a Dahlquist research fellowship. We have gained much inspiration from the concepts of the software package Chebfun [2], and the authors acknowledge valuable discussions with Alex Townsend, Nick Trefethen, and the rest of the Chebfun team.

REFERENCES