An Arnoldi like method for the delay eigenvalue problem

Elias Jarlebring
Karl Meerbergen
Wim Michiels

Report TW 558, January 2010



Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

An Arnoldi like method for the delay eigenvalue problem

Elias Jarlebring
Karl Meerbergen
Wim Michiels

Report TW 558, January 2010

Department of Computer Science, K.U.Leuven

Abstract

The method called *Arnoldi* is currently a very popular method to solve large-scale eigenvalue problems. The general purpose of this paper is to generalize Arnoldi to the characteristic equation of a *delay-differential equation* (DDE), here called a *delay eigenvalue problem*.

The DDE can equivalently be expressed with a linear infinite dimensional operator which eigenvalues are the solutions to the delay eigenvalue problem. A common approach to solve the delay eigenvalue problem is to discretize the operator and compute the eigenvalues of a standard or generalized eigenvalue problem.

We derive a new method by applying Arnoldi to the generalized eigenvalue problem associated with a spectral discretization of the operator. It turns out that the structure of the problem can be heavily exploited and we derive an efficient matrix vector product. More importantly, the structure is such that if the Arnoldi scheme is started in an appropriate way, it is mathematically equivalent to an Arnoldi scheme with an infinite matrix, corresponding to the limit where we have an infinite number of discretization points. The resulting method is a scheme where we expand a subspace, not only in the traditional way done in Arnoldi, but the subspace vectors are also expanded with one block of rows in each iteration. In this way, the number of discretization points increases with the number of Arnoldi iterations such that the number of discretization points does not have to be fixed before the iteration starts.

It turns out that there is a complete equivalence between the presented method and Arnoldi applied in the setting of the infinite dimensional operator. More precisely, we show that if the functions space on which the operator acts is equipped with an appropriate scalar product, the vectors in the Arnoldi iteration can be interpreted as the coefficients in a Chebyshev expansion of a function, and the Hessenberg matrix produced by the presented method and the standard Arnoldi scheme applied to the linear infinite dimensional operator are equal.

AN ARNOLDI LIKE METHOD FOR THE DELAY EIGENVALUE PROBLEM

ELIAS JARLEBRING , KARL MEERBERGEN , AND WIM MICHIELS*

Abstract. The method called *Arnoldi* is currently a very popular method to solve large-scale eigenvalue problems. The general purpose of this paper is to generalize Arnoldi to the characteristic equation of a *delay-differential equation* (DDE), here called a *delay eigenvalue problem*.

The DDE can equivalently be expressed with a linear infinite dimensional operator which eigenvalues are the solutions to the delay eigenvalue problem. A common approach to solve the delay eigenvalue problem is to discretize the operator and compute the eigenvalues of a standard or generalized eigenvalue problem.

We derive a new method by applying Arnoldi to the generalized eigenvalue problem associated with a spectral discretization of the operator. It turns out that the structure of the problem can be heavily exploited and we derive an efficient matrix vector product. More importantly, the structure is such that if the Arnoldi scheme is started in an appropriate way, it is mathematically equivalent to an Arnoldi scheme with an infinite matrix, corresponding to the limit where we have an infinite number of discretization points. The resulting method is a scheme where we expand a subspace, not only in the traditional way done in Arnoldi, but the subspace vectors are also expanded with one block of rows in each iteration. In this way, the number of discretization points increases with the number of Arnoldi iterations such that the number of discretization points does not have to be fixed before the iteration starts.

It turns out that there is a complete equivalence between the presented method and Arnoldi applied in the setting of the infinite dimensional operator. More precisely, we show that if the functions space on which the operator acts is equipped with an appropriate scalar product, the vectors in the Arnoldi iteration can be interpreted as the coefficients in a Chebyshev expansion of a function, and the Hessenberg matrix produced by the presented method and the standard Arnoldi scheme applied to the linear infinite dimensional operator are equal.

1. Introduction. Consider the linear time-invariant differential equation with several discrete delays

$$\dot{x}(t) = A_0 x(t) + \sum_{i=1}^{m} A_i x(t-\tau_i), \tag{1.1}$$

where $A_0, \ldots, A_m \in \mathbb{R}^{n\times n}$ and $\tau_1, \ldots, \tau_m \in \mathbb{R}$. Without loss of generality we will order the delays and let $0 =: \tau_0 < \cdots < \tau_m$. The differential equation (1.1) is known as a delay-differential equation (DDE) and is often characterized and analyzed using the solutions of the associated characteristic equation

$$\det \Delta(\lambda) = 0, \tag{1.2}$$

where

$$\Delta(\lambda) = \lambda I - A_0 - \sum_{i=1}^{m} A_i e^{-\tau_i \lambda}.$$

We will call the problem of finding $\lambda \in \mathbb{C}$ such that $\det(\Delta(\lambda)) = 0$ the *delay eigenvalue problem* (DEP) and the solutions $\lambda \in \mathbb{C}$ are called the characteristic roots or the eigenvalues of the DDE (1.1). The eigenvalues of (1.1) play the same important role for DDEs as eigenvalues play for matrices and standard differential equations without delay. The eigenvalues can be used to study stability, design and optimization in

*K.U. Leuven, Department of Computer Science, 3001 Heverlee, Belgium, {Elias.Jarlebring,Karl.Meerbergen,Wim.Michiels}@cs.kuleuven.be

the frequency domain, etc. See [MN07] for recent results on eigenvalues of delay-differential equations and their applications.

The Arnoldi method is a very popular algorithm for large standard and generalized eigenvalue problems. An essential component in the Arnoldi scheme is the repeated expansion of a subspace with a new basis vector. In this paper we present an Arnoldi like method for the delay eigenvalue problem where the subspace is extended not only in the traditional way done in Arnoldi. The basis vectors are also extended with one block row in each iteration.

The method turns out to be very efficient and robust (which we illustrate with the examples in Section 5). The reliability of the method can be understood from several properties and close relations with the Arnoldi scheme, which we will use to derive and understand the method.

An important property of (1.2) is that the solution can be equivalently written as the eigenvalues of an infinite dimensional operator, denoted $\mathcal{A}$. A common approach to compute λ is to discretize this operator $\mathcal{A}$ and compute the eigenvalues of a matrix [BMV05, BMV09]. We will start (in Section 2) from this approach and derive a sparse representation of the problem. Based on the pencil structure of this eigenvalue problem we derive the Arnoldi type algorithm (in Section 3). Performing a fixed number of iterations with the algorithm is mathematically equivalent to an Arnoldi scheme applied to the discretized problem, where the number of grid points coincide with the number of iterations.

However, the scheme makes explicit use of the sparsity structure of the discretized problem, such that it can be implemented efficiently and more importantly, it can be implemented dynamically in the sense that the number of discretization points does not have to be chosen before the iteration starts.

The constructed algorithm is equivalent to Arnoldi in a different setting. The operator $\mathcal{A}$ is a linear map of a function segment to a function segment. With an appropriately defined inner product, it is easy to conceptually construct Arnoldi in an infinite dimensional setting. We prove (in Section 4) that the presented method is mathematically equivalent to Arnoldi applied to an infinite operator with a special inner product.

Although there are numerous methods for the DEP, the presented method has several appealing properties not present in other methods. There are several methods based on discretization such as [BM00, Bre06, BMV05, BMV09]. In particular, the software package DDE-BIFTOOL [ELR02, ELS01] uses a discretization approach combined with an iterative method. It is a two-step method based on

1) estimating many roots with a discretization; and then
2) correcting the computed solutions with an iterative method.

In our approach there is no explicit need for a second step as sufficient accuracy can be achieved directly by the iteration. Moreover, the grid of the discretization has to be fixed a priori in classical discretization approaches. See [VLR08] for a heuristic choice for the discretization used in DDE-BIFTOOL. Moreover, in comparison to the discretization approaches, our method only involves linear algebra operations with matrices of dimension $n \times n$, making it suitable for large and sparse problems.

There are other approaches for the DEP, e.g., the method QPMR [VZ09] which is based on the coefficients of characteristic equation and hence likely not very suitable for very large and sparse problems which is the context of the presented approach. See [Jar08, Chapter 2] for more methods.

The DEP belongs to a class of problems called nonlinear eigenvalue problems.

There are several general purpose method for nonlinear eigenvalue problems; see [Ruh73, MV04]. There is for instance the Newton-type methods [Sch08, Neu85] and nonlinear version of Jacobi-Davidson [BV04]. There is also an Arnoldi like method in [Vos04]. We wish to stress that despite the similar name, the algorithm presented in this paper and the method in [Vos04] (which has been applied to the delay eigenvalue problem in [Jar08, Chapter 2]) has very little in common. In comparison to these methods, the presented method is expected to be more reliable since it inherits most properties of Arnoldi, e.g., robustness and simultaneous convergence to several eigenvalues. The block Newton method has recently been generalized to nonlinear eigenvalue problems [Kre09]. It has been applied to the delay eigenvalue problem. The differences between block Newton and Arnoldi for standard eigenvalue problems seem to hold here as well. Arnoldi is often more efficient for very large systems since only the right-hand side and not the matrix of the linear system changes in each iteration. A decomposition (e.g. LU-decomposition) can be computed before the iteration starts and the linear system can be solved very efficiently. Moreover, in Arnoldi, it is not necessary to fix the number of wanted solutions before the iteration starts.

The polynomial eigenvalue problem (PEP) is an important nonlinear eigenvalue problem. There is recent theory and numerical algorithms for polynomial eigenvalue problems; see, e.g., [MMMM06, MV04, Lan02]. In particular, it was recently shown in [ACL09] how to linearize a PEP if it is expressed in a Chebyshev polynomial basis. We will see in Section 4 that our approach has a relation with a Chebyshev expansion. The derivation of our approach is based on a discretization of an infinite dimensional operator and there is no explicit need to use the theory of linearizations.

Finally, we note that the Arnoldi scheme presented here is inspired by the Arnoldi like methods for polynomial eigenvalue problems, in particular the quadratic eigenvalue problem in [BS05, Mee08].

2. Discretization and companion-type formulation. Some theory available for DDEs is derived in a setting where the DDE is stated as an infinite-dimensional linear system. A common infinite dimensional representation will be briefly summarized in Section 2.1. The first step in the derivation of the main numerical method of this paper (given in Section 3) is based on a discretization of this infinite dimensional representation. The discretization is different from other traditional discretizations available in the literature because the discretized matrix can be expressed with a companion like matrix and a block triangular matrix. The discretization and the associated manipulations are given in Section 2.2 and Section 2.3.

2.1. Infinite dimensional first order form. Let $X := \mathcal{C}([-\tau_m,\ 0],\ \mathbb{C}^n)$ be the Banach space of continuous functions mapping the interval $[-\tau_m,\ 0]$ onto $\mathbb{C}^m$ and equipped with the supremum norm. Consider the linear operator $\mathcal{A}$ on X be defined by

$$\begin{aligned} &\mathcal{D}(\mathcal{A}) = \left\{\phi \in X :\ \tfrac{d\phi}{d\theta} \in X,\ \phi'(0) = A_0\phi(0) + \textstyle\sum_{k=1}^{m} A_k\phi(-\tau_k)\right\}, \\ &\mathcal{A}\ \phi = \tfrac{d\phi}{d\theta}. \end{aligned} \tag{2.1}$$

The equation (1.1) can now be reformulated as an abstract ordinary differential equation over X

$$\dot{z}(t) = \mathcal{A}z(t). \tag{2.2}$$

See [HV93] for a detailed description of $\mathcal{A}$. The corresponding solutions of (1.1) and (2.2) are related by $z(t) \equiv x(t+\theta),\ \theta \in [-\tau_m,\ 0]$.

The spectral properties of the operator $\mathcal{A}$ are described in detail in [MN07, Chapter 1]. The operator only features a point spectrum. Therefore its spectrum is fully determined by the eigenvalue problem

$$\mathcal{A}\ z = \lambda z, \quad z \in X,\ z \neq 0. \tag{2.3}$$

The connections with the characteristic roots are as follows. The characteristic roots are the eigenvalues of the operator $\mathcal{A}$. Moreover, if $\lambda \in \sigma(\mathcal{A})$, then the corresponding eigenfunction takes the form

$$ve^{\lambda\theta},\ \theta \in [-\tau_m,\ 0], \tag{2.4}$$

where $v \in \mathbb{C}^n \setminus \{0\}$ satisfies

$$\Delta(\lambda)v = 0. \tag{2.5}$$

Conversely, if the pair (v, λ) satisfies (2.5) and $v \neq 0$, then (2.4) is an eigenfunction of $\mathcal{A}$ corresponding to the eigenvalue λ.

We have summarized the following important relation. The characteristic roots appear as solutions of both the linear infinite-dimensional eigenvalue problem (2.3) and the finite-dimensional nonlinear eigenvalue problem (2.5). This connection plays an important role in many methods for computing characteristic roots and assessing their sensitivity [MN07].

2.2. Spectral discretization. A common way to numerically analyze the spectrum of an infinite-dimensional operator is to discretize it. Here, we will take an approach based on approximating $\mathcal{A}$ by a matrix using a *spectral discretization method* (see, e.g. [Tre00, BMV05]).

Given a positive integer N, we consider a mesh Ω_N of $N+1$ distinct points in the interval $[-\tau_m,\ 0]$:

$$\Omega_N = \{\theta_{N,i},\ i = 1, \ldots, N+1\}, \tag{2.6}$$

where

$$-\tau_m \leq \theta_{N,1} < \ldots < \theta_{N,N} < \theta_{N,N+1} = 0.$$

Now consider the discretized problem where we have replaced X with the space X_N of discrete functions defined over the mesh Ω_N, i.e., any function $\phi \in X$ is discretized into a block vector $x = [x_1^T \cdots\ x_{N+1}^T]^T \in X_N$ with components

$$x_i = \phi(\theta_{N,i}) \in \mathbb{C}^n, \quad i = 1, \ldots, N+1.$$

Let $\mathcal{P}_N x,\ x \in X_N$, be the unique $\mathbb{C}^n$ valued interpolating polynomial of degree smaller or equal than N, satisfying

$$\left(\mathcal{P}_N x\right)(\theta_{N,i}) = x_i, \quad i = 1, \ldots, N+1.$$

In this way we can approximate the operator $\mathcal{A}$ over X with the matrix $\mathcal{A}_N :\ X_N \to X_N$, defined as

$$\begin{array}{lll} \left(\mathcal{A}_N\ x\right)_i = & \left(\mathcal{P}_N x\right)'(\theta_{N,i}), & i = 1, \ldots, N, \\ \left(\mathcal{A}_N\ x\right)_{N+1} = & A_0 \mathcal{P}_N x(0) + \sum_{i=1}^m A_i \mathcal{P}_N x(-\tau_i). & \end{array} \tag{2.7}$$

Using the Lagrange representation

$$\mathcal{P}_N x = \sum_{k=1}^{N+1} l_{N,k}\ x_k,$$

where the Lagrange polynomials $l_{N,k}$ are real valued polynomials of degree N satisfying

$$l_{N,k}(\theta_{N,i}) = \begin{cases} 1 & i = k, \\ 0 & i \neq k, \end{cases}$$

we get an explicit form for the matrix $\mathcal{A}_N$,

$$\mathcal{A}_N = \begin{bmatrix} d_{1,1} & \dots & d_{1,N+1} \\ \vdots & & \vdots \\ d_{N,1} & \dots & d_{N,N+1} \\ a_1 & \dots & a_{N+1} \end{bmatrix} \in \mathbb{R}^{(N+1)n \times (N+1)n}, \tag{2.8}$$

where

$$\begin{array}{rcll} d_{i,k} & = & l'_{N,k}(\theta_{N,i}) I_n, & i \in \{1,\dots,N\},\ k \in \{1,\dots,N+1\}, \\ a_k & = & A_0 l_{N,k}(0) + \sum_{i=1}^{m} A_i l_{N,k}(-\tau_i), & k \in \{1,\dots,N+1\}. \end{array}$$

The numerical methods for computing characteristic roots, described in [BMV05, BMV06, BMV09], are based on solving the discretized linear eigenvalue problem

$$\mathcal{A}_N\ x = \lambda x,\ \lambda \in \mathbb{C},\ x \in \mathbb{C}^{(N+1)n},\ x \neq 0, \tag{2.9}$$

by constructing the matrix $\mathcal{A}_N$ and computing all its eigenvalues with the QR method.

With an appropriately chosen grid Ω_N in the discretization, the convergence of the individual eigenvalues of $\mathcal{A}_N$ to corresponding eigenvalues of $\mathcal{A}$ is fast. In [BMV05] it is proven that spectral accuracy (approximation error $O(N^{-N})$) is obtained with a grid consisting of (scaled and shifted) Chebyshev extremal points.

2.3. A companion-type reformulation. The matrix $\mathcal{A}_N$ in equation (2.8) does not have a simple apparent matrix structure. We will now see that $\mathcal{A}_N$ is equivalent (similar) to a pair of matrices where one is a block triangular matrix and the other is a companion like matrix, if we choose the grid points appropriately. The matrix structure will be exploited in Section 3.

We start with the observation that the discretized eigenvalue problem (2.9) can be directly obtained by requiring that there exists a polynomial of degree N,

$$(\mathcal{P}_N x)(t) = \sum_{k=0}^{N} l_{N,k}(t)\ x_k,$$

which satisfies the conditions

$$(\mathcal{P}_N x)'(\theta_{N,i}) = \lambda \mathcal{P}_N x(\theta_{N,i}), \quad i \in \{1,\dots,N\}, \tag{2.10}$$

$$A_0 \mathcal{P}_N x(0) + \sum_{i=1}^{m} A_i \mathcal{P}_N x(-\tau_i) = \lambda \mathcal{P}_N x(0). \tag{2.11}$$

Hence, an eigenvalue problem equivalent to (2.9) can be obtained by expressing $\mathcal{P}_N x$ in another basis and imposing the same conditions.

Consider the representation of $\mathcal{P}_N x$ in a basis of Chebyshev polynomials

$$(\mathcal{P}_N x)(t) = \sum_{i=0}^{N} c_i T_i \left(2\frac{t}{\tau_m} + 1 \right), \tag{2.12}$$

where T_i is the Chebyshev polynomial of the first kind and order i, and $c_i \in \mathbb{C}^{N\times 1}$ for $i = 0, \ldots, N$. By requiring that this polynomial satisfies the conditions (2.10)-(2.11) and by taking into account the property

$$T_i'(t) = iU_{i-1}(t),\ i \geq 1, \tag{2.13}$$

where U_i is the Chebyshev polynomial of the second kind and order i, we obtain the following equivalent eigenvalue problem for (2.9):

$$\left(\lambda \left[\begin{array}{c|c} 1\ T_1(1)\ \cdots\ T_{N-1}(1) & T_N(1) \\ \hline \Gamma_1 & \Gamma_2 \end{array} \right] \otimes I_n - \left[\begin{array}{c|c} R_0 & R_1\ \cdots\ R_N \\ \hline 0 & U \otimes I_n \end{array} \right] \right) \left[\begin{array}{c} c_0 \\ \vdots \\ c_N \end{array} \right] = 0, \tag{2.14}$$

where the submatrices are defined as

$$\Gamma_1 = \left[\begin{array}{cccc} 1 & T_1(\alpha_1) & \cdots & T_{N-1}(\alpha_1) \\ \vdots & & & \vdots \\ 1 & T_1(\alpha_N) & \cdots & T_{N-1}(\alpha_N) \end{array} \right],\ \ \Gamma_2 = \left[\begin{array}{c} T_N(\alpha_1) \\ \vdots \\ T_N(\alpha_N) \end{array} \right], \tag{2.15}$$

$$U = \frac{2}{\tau_m} \left[\begin{array}{cccc} U_0(\alpha_1) & 2U_1(\alpha_1) & \cdots & NU_{N-1}(\alpha_1) \\ \vdots & & & \vdots \\ U_0(\alpha_N) & 2U_1(\alpha_N) & \cdots & NU_{N-1}(\alpha_N) \end{array} \right], \tag{2.16}$$

$$R_i = A_0 T_i(1) + \sum_{k=1}^{m} A_k T_i \left(-2\frac{\tau_k}{\tau_m} + 1 \right),\ i = 0, \ldots, m,$$

and

$$\alpha_i = 2\frac{\theta_i}{\tau_m} + 1,\quad i = 1, \ldots, N+1.$$

are the normalized grid-points, that is, scaled and shifted to the interval $[-1,\ 1]$.

In order to simplify (2.14), we use the property

$$T_1(t) = \frac{1}{2}U_1(t),\quad T_i(t) = \frac{1}{2}U_i(t) - \frac{1}{2}U_{i-2}(t),\ i \geq 2, \tag{2.17}$$

which implies

$$\Gamma_1 = UL, \tag{2.18}$$

where L is a band matrix, defined as

$$L_N = \frac{\tau_m}{4} \left[\begin{array}{ccccc} 2 & 0 & -1 & & \\ & \frac{1}{2} & 0 & -\frac{1}{2} & \\ & & \frac{1}{3} & 0 & \ddots \\ & & & \frac{1}{4} & \ddots \\ & & & & \ddots \end{array} \begin{array}{c} \\ \\ \\ \\ -\frac{1}{N-2} \\ 0 \\ \frac{1}{N} \end{array} \right]. \tag{2.19}$$

If the grid points are such that U is nonsingular, we can use the relation (2.18) to transform (2.14) to the sparse eigenvalue problem

$$\left(\lambda \left[\begin{array}{c|c} 1\ T_1(1)\ \cdots\ T_{N-1}(1) & T_N(1) \\ \hline L & U^{-1}\Gamma_2 \end{array}\right] \otimes I_n - \left[\begin{array}{c|c} R_0 & R_1\ \cdots\ R_N \\ \hline 0 & I_{Nn} \end{array}\right]\right) \begin{bmatrix} c_0 \\ \vdots \\ c_N \end{bmatrix} = 0. \quad (2.20)$$

An important property of the eigenvalue problem (2.20) is that all information about the grid Ω_N is *concentrated* in the column $U^{-1}\Gamma_2$. We now show that with an appropriately chosen Chebyshev type grid, the structure of L is continued in this column. For this, choose the nonzero grid points as scaled and shifted zeros of U_N, the latter given by

$$\alpha_i = -\cos\frac{\pi i}{N+1},\ i = 1, \ldots, N. \quad (2.21)$$

First, this choice implies that the matrix (2.16) is invertible. Second, from (2.17) it can be seen that the numbers (2.21) satisfy

$$T_N(\alpha_i) = -\frac{1}{2}U_{N-2}(\alpha_i),\ i = 1, \ldots, N, \quad N \geq 2,$$

which implies on its turn that

$$U^{-1}\Gamma_2 = \begin{bmatrix} 0 & \cdots & 0 & -\frac{\tau_m}{4(N-1)} & 0 \end{bmatrix}^T.$$

By combining the results above, we arrive at the following theorem.

THEOREM 2.1. *If the grid points in the spectral dicretization of the operator* (2.1) *are chosen as*

$$\theta_i = \frac{\tau_m}{2}(\alpha_i - 1), \quad \alpha_i = -\cos\frac{\pi i}{N+1},\ i = 1, \ldots, N+1, \quad (2.22)$$

then the discretized eigenvalue problem (2.9) *is equivalent with*

$$(\lambda\Pi_N - \Sigma_N)\, c = 0,\ \lambda \in \mathbb{C},\ c \in \mathbb{C}^{(N+1)n},\ c \neq 0, \quad (2.23)$$

where

$$\Pi_N = \frac{\tau_m}{4}\begin{bmatrix} \frac{4}{\tau_m} & \frac{4}{\tau_m} & \frac{4}{\tau_m} & \cdots & & \cdots & \frac{4}{\tau_m} \\ 2 & 0 & -1 & & & & \\ & \frac{1}{2} & 0 & -\frac{1}{2} & & & \\ & & \frac{1}{3} & 0 & \ddots & & \\ & & & \frac{1}{4} & \ddots & -\frac{1}{N-2} & \\ & & & & \ddots & 0 & -\frac{1}{N-1} \\ & & & & & \frac{1}{N} & 0 \end{bmatrix} \otimes I_n \quad (2.24)$$

and

$$\Sigma_N = \begin{bmatrix} R_0 & R_1 & \cdots & R_N \\ & I_n & & \\ & & \ddots & \\ & & & I_n \end{bmatrix}, \quad (2.25)$$

with

$$R_i = A_0 + \sum_{k=1}^{m} A_k T_i\left(-2\frac{\tau_k}{\tau_m} + 1\right),\ i = 0, \ldots, N.$$

REMARK 2.2 (The choice of discretization points). *A grid consisting of the points* α_i *as in* (2.22) *is very similar to a grid consisting of Chebyshev extremal points as in [BMV05], where the latter is defined as*

$$-\cos\frac{\pi(i-1)}{N},\ i = 1, \ldots, N+1. \tag{2.26}$$

Note that the convergence theory of spectral discretizations typically shown using reasoning with a potential function defined from a limit of the grid distribution. See, e.g., [Tre00, Theorem 5]. Since the discretization here (2.22) *and the grid in [BMV05], i.e.,* (2.26) *have the same asymptotic distribution, we expect the convergence properties to be the same. For instance, the eigenvalues of* $\mathcal{A}_N$ *exhibit spectral convergence to eigenvalues of* $\mathcal{A}$*. Moreover, the part of the spectrum of* $\mathcal{A}_N$ *which has not yet converged to corresponding eigenvalues of* $\mathcal{A}$*, is typically located to the* left *of the converged eigenvalues, which is an important property when assessing stability.*

We have chosen a new grid as this grid allows us to construct matrices with a particularly useful structure. The structure which will be used in the next section is that the matrices (2.24) *and* (2.25) *are such that* Π_{N_1} *and* Σ_{N_1} *are* submatrices *of* Π_{N_2} *and* Σ_{N_2} *whenever* $N_2 \geq N_1$*.*

3. Delay Arnoldi. The discretization in the previous section can be directly used to find approximations of the eigenvalues of (1.1) by computing the eigenvalues of the sparse generalized eigenvalue problem $(\lambda\Pi_N - \Sigma_N)x = 0$ with a general purpose eigenvalue solver. The Arnoldi method (first introduced in [Arn51]) is one popular general purpose eigenvalue solver and we will adapt a version Arnoldi to this generalized eigenvalue problem. This somewhat traditional approach has the drawback that the matrices Σ_N and Π_N are large if N is large. Ideally, we would like N to be large since large N is necessary to have accurate eigenvalue approximations.

The method we present in this section is in a sense a perfect solution to this issue. It turns out that if we start Arnoldi in an appropriate way the computed approximations are the same as Arnoldi applied to the eigenvalue problem associated with the limit $N \to \infty$.

We will use the natural limit interpretation of Σ_N and Π_N. Let $\mathrm{vec}(\mathbb{C}^{n\times\infty})$ denote the set of all ordered infinite sequences of vectors of length n exponentially convergent to zero. The natural interpretation of the limits of the operators Σ_∞ and Π_∞ is with this notation $\Sigma_\infty : \mathrm{vec}(\mathbb{C}^{n\times\infty}) \to \mathrm{vec}(\mathbb{C}^{n\times\infty})$ and $\Pi_\infty : \mathrm{vec}(\mathbb{C}^{n\times\infty}) \to \mathrm{vec}(\mathbb{C}^{n\times\infty})$. We will call an element of $\mathrm{vec}(\mathbb{C}^{n\times\infty})$ an infinite vector and Σ_∞ and Π_∞ infinite matrices. In many results and applications of Arnoldi, the scheme is implicitly equipped with the Eucledian scalar product. We will use natural extension of the Eucledian scalar product to infinite vectors, $\langle x, y\rangle = x^*y = \sum_{i=0}^{\infty} x_i^* y_i$.

Arnoldi is a construction of an orthogonal basis of the set of linear combinations of a power sequence associated with matrix $A \in \mathbb{R}^{n\times n}$ and vector $b \in \mathbb{R}^n$,

$$\mathcal{K}_k(A, b) := \mathrm{span}\{b, Ab, \ldots, A^{k-1}b\}.$$

This subspace is called a *Krylov subspace*. The Arnoldi method approximates eigenvalues of A by the eigenvalues of $H_k = V_k^H A V_k$ (which are called the Ritz values) where

the columns of $V_k \in \mathbb{C}^{n\times k}$ form an orthogonal basis of $\mathcal{K}_k(A,b)$. The eigenvalues of H_k converge to the extreme well-separated eigenvalues first [Saa92, Chapter VI]. In this paper we are interested in eigenvalues close to the origin. For our applications, those are usually not very well-separated, so convergence is expected to be slow. Incidentally, convergence can be drastically improved by applying Arnoldi to A^{-1}, as the eigenvalues of A near zero become typically well-separated extreme eigenvalues of A^{-1}, and so, fast convergence is guaranteed. For this reason we will apply Arnoldi to form $\mathcal{K}_k(A^{-1},b)$.

As in the definition of the Krylov subspace, the matrix vector product is an important component in the Arnoldi method. The underlying property which we will use next is the matrix vector product associated with the infinite matrix $\Sigma_\infty^{-1}\Pi_\infty$. It turns out to be structured in such a way that it has a closed form for a special type of vector.

THEOREM 3.1 (Matrix vector product). *Suppose $\sum_{i=0}^m A_i$ is non-singular. Let $\Sigma_\infty, \Pi_\infty$ be as in Theorem 2.1 and $Y \in \mathbb{C}^{n\times k}$. Then*

$$\Sigma_\infty^{-1}\Pi_\infty \mathrm{vec}(Y,0,\ldots) = \mathrm{vec}(\hat{x},Z,0,\ldots)$$

and

$$Z = YL_k^T, \tag{3.1}$$

where $L_k \in \mathbb{R}^{k\times k}$ is given by (2.19)*, i.e.,*

$$L_k = \frac{\tau_m}{4}\begin{bmatrix} 2 & 0 & -1 & & & \\ & \frac{1}{2} & 0 & -\frac{1}{2} & & \\ & & \frac{1}{3} & 0 & \ddots & \\ & & & \frac{1}{4} & \ddots & -\frac{1}{k-2} \\ & & & & \ddots & 0 \\ & & & & & \frac{1}{k} \end{bmatrix}.$$

and

$$\hat{x} = \left(\sum_{j=0}^m A_j\right)^{-1}\left(\sum_{i=0}^{k-1} y_i - A_0\sum_{i=0}^{k-1} z_i - \sum_{j=1}^m A_j\left(\sum_{i=0}^{k-1} T_{i+1}(1-2\frac{\tau_j}{\tau_m})z_i\right)\right). \tag{3.2}$$

Proof. The proof is based on forming the limits for $\Sigma_\infty\mathrm{vec}(\hat{x},Z,0,\ldots)$ and $\Pi_\infty\mathrm{vec}(Y,0,\ldots)$ and noting that the result yields (3.1) and (3.2). Note that from (2.25) we have that

$$\Sigma_\infty\mathrm{vec}(\hat{x},Z,0,\ldots) \;=\; \begin{bmatrix} R_0 & R_1 & \cdots \\ & I_n & \\ & & \ddots \end{bmatrix}\mathrm{vec}(\hat{x},Z,0,\ldots) \;=\; \mathrm{vec}(y,Z,0,\ldots),$$

where $y = R_0\hat{x} + R_1 z_0 + \cdots + R_k z_{k-1}$. Let $v_k^T = (1,\ldots,1)^T \in \mathbb{R}^k$. We now have from (2.24) and rules of vectorization and Kronecker products that

$$\Pi_\infty\mathrm{vec}(Y,0,\ldots) = \left(\begin{pmatrix} v_\infty^T \\ L_\infty \end{pmatrix}\otimes I\right)\mathrm{vec}(Y,0,\ldots) = \\ \mathrm{vec}\left((Y,0,\ldots)v_\infty, (Y,0,\ldots)L_\infty^T,\right) = \mathrm{vec}\left(Yv_k, YL_k^T, 0,\ldots\right).$$

The proof is completed by solving $y = Yv_k$. ∎

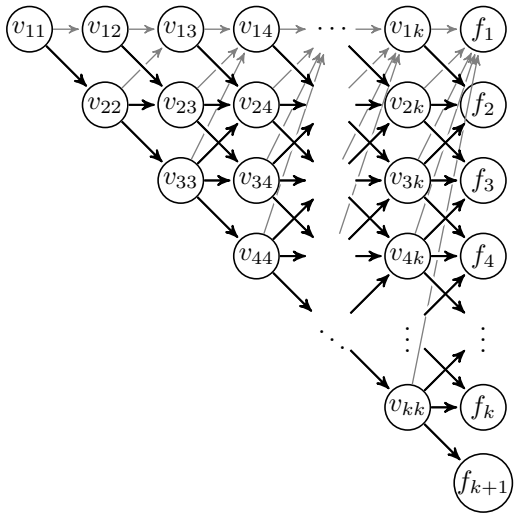


FIGURE 3.1. *Dependency tree of the basis matrix V_k. The column which is added in Step 11 in Algorithm 1 has been denoted $f = v_{k+1}$.*

3.1. Algorithm. Now consider the power sequence for the operator $\Sigma_\infty^{-1}\Pi_\infty$ started with $\mathrm{vec}(x, 0, \cdots)$, $x \in \mathbb{R}^n$. From Theorem 3.1 we see that the non-zero part of the infinite vector grows by one vector (of length n) in each iteration such that at the jth step, the resulting infinite vector is $\mathrm{vec}(Y, 0, \ldots)$ where $Y \in \mathbb{R}^{n\times(j+1)}$.

Arnoldi builds the Krylov sequence vector by vector, where in addition, the vectors are orthogonalized. In step k, the orthogonalization is a linear combination of the $k+1$st vector and the previously computed k vectors. Hence, the orthogonalization at the kth iteration does not change the general structure of the $k+1$st vector.

This allows us to construct an Arnoldi like scheme where we dynamically increase the size of basis vectors. Let V_k be the matrix consisting of the basis vectors and $v_{ij} \in \mathbb{C}^n$ the vector corresponding to block element i, j. The dependency tree of the basis vectors is given in Figure 3.1, where the gray arrows represent the computation of the first component $\hat{x}$.

The algorithm which is given in Algorithm 1 is (from the reasoning above) mathematically equivalent to Arnoldi applied to the matrix $\Sigma_N^{-1}\Pi_N$ as well as $\Sigma_\infty^{-1}\Pi_\infty$. We use notation common for Arnoldi iterations; we let $\underline{H}_k \in \mathbb{C}^{(k+1)\times k}$ denote the dynamically constructed rectangular Hessenberg and $H_k \in \mathbb{C}^{k\times k}$ the corresponding $k \times k$ upper part.

3.2. Implementation details. The efficiency and robustness of Algorithm 1 can only be guaranteed if the important implementational issues are addressed. We will use some techniques used in standard Arnoldi implementations and some techniques which are adapted for this problem.

As is usually done in eigenvalue computations using Arnoldi's method, $\sum_{k=0}^m A_k$ is factorized by a sparse direct solver and then each Arnoldi step requires a backward solve with the factors for computing $(\sum_{k=0}^m A_k)^{-1}y$. Examples of such direct solvers are [ADLK01, SG04, DEG+99, Dav04]. In Step 8, the vector x should be orthogonalized against V. We use iterative reorthogonalization as in ARPACK [LSY98].

The Ritz vectors associated with Step 13 in Algorithm 1 are $u = V_k z$ where z is an eigenvector of the $k \times k$ Hessenberg matrix H_k. Note that the eigenvalues μ of

Algorithm 1 Delay Arnoldi

Require: $x_0 \in \mathbb{C}^n$ and time-delay system (1.1)

1: Let $v_1 = x_0/\|x_0\|_2$, $k = 1$, $\underline{H}_0$ =empty matrix
2: Factorize $\sum_{i=0}^{m} A_i$
3: **for** $k = 1, 2, \ldots$ until converged **do**
4: Let $\mathrm{vec}(Y) = v_k$
5: Compute Z according to (3.1) with sparse L_k
6: Compute $\hat{x}$ according to (3.2) using the factorization of the inverse computed in Step 2
7: Expand V_k with one block row (zeros)
8: Let $w_k := \mathrm{vec}(\hat{x}, Z)$, compute $h_k = V_k^* w_k$ and then $\hat{w}_k = w_k - V_k h_k$
9: Compute $\beta_k = \|\hat{w}_k\|_2$ and let $v_{k+1} = \hat{w}_k/\beta_k$
10: Let $\underline{H}_k = \begin{bmatrix} \underline{H}_{k-1} & h_k \\ 0 & \beta_k \end{bmatrix} \in \mathbb{C}^{(k+1)\times k}$
11: Expand V_k into $V_{k+1} = [V_k, v_{k+1}]$
12: **end for**
13: Compute the eigenvalues μ from the Hessenberg matrix H_k
14: Return approximations $1/\mu$

H_k are approximations to eigenvalues of $\Sigma_N^{-1}\Pi_N$, so $\lambda = \mu^{-1}$ are the corresponding approximate eigenvalues of (1.1). Note that $u \in \mathbb{C}^{nk}$ and that an eigenpair of a time-delay system can be represented by the eigenvalue $\lambda \in \mathbb{C}$ and a (short) eigenvector $v \in \mathbb{C}^n$. A user is typically only interested in approximations of v and not approximations of u. For this reason we will now discuss adequate ways to extract $v \in \mathbb{C}^n$ from $u \in \mathbb{C}^{nk}$. This will also be used to derive stopping criteria.

Note that the eigenvectors of $\Sigma_N^{-1}\Pi_N$ are of the form $w = c \otimes v$. Given a Ritz vector $u \in \mathbb{C}^{nk}$ we will construct the vector $v \in \mathbb{C}^n$ from the first n components of u. This can be motivated by the following observation in Figure 3.1. Note that the vector $v_{k-p,k}$ does not depend on the nodes in the right upper triangle with sides of length $p-1$ in the graph. In fact, $v_{k-p,k}$ is a linear combination of $v_{1,i}$, $i = 1, \ldots, p+1$. Hence, the quality of the vector $v_{k-p,k}$ can not be expected to be much better than the first p iterations. With inductive reasoning, we conclude that the first block of n rows of V contains the information with the highest quality.

Another natural way to extract v is by computing the vector corresponding to the dominant singular value. In general our numerical experiments are not conclusive and show little different between the two approaches about which extraction is better. We propose to use the former approach.

REMARK 3.2 (Residuals). *The termination criteria in standard Arnoldi is typically an expression involving the residual. In the setting of Algorithm 1 there are two natural ways to define residuals. There is the residual*

$$r := \Sigma_N^{-1}\Pi_N u - \lambda^{-1} u \in \mathbb{C}^{nN}$$

and the (short) residual $\hat{r} := \Delta(\lambda)v \in \mathbb{C}^n$. *The norm of the residual* r *is cheaply available as a by-product of the Arnoldi iteration as for standard Arnoldi: let* $H_k z = \lambda^{-1} z$ *with* $\|z\|_2 = 1$, *then* $\|r\|_2 = h_{k+1,k}|e_k^T z|$. *It is however more natural to have termination criteria involving* $\|\hat{r}\|$ *since from the residual norm it is easy to derive a backward error. Unfortunately, even though* v *can easily be extracted from* u *(as is mentioned above) the computation* $\Delta(\lambda)v$ *is too expensive to evaluate in each iteration*

for each eigenvector candidate. In the examples section we will use a fixed number of iterations, but in a general purpose implementation we propose to use a heuristic combination, where the cheap residual norm $\|r\|$ is used until it is sufficiently small and in a post-processing step, the residual norms $\|\hat{r}\|$ can be used to check the result. The residual $\|r\|_2$ will be further interpreted in Remark 4.6.

REMARK 3.3 (Reducing memory requirements). *The consumption of memory resources is the bottleneck in many applications of standard Arnoldi. Note that the situation is expected to be even worse for Algorithm 1 due to the dynamic growth of the size of the basis vectors. The memory requirement in Algorithm 1 is $O(k^2 n)$, whereas for standard Arnoldi it is only $O(kn)$.*

We note however that a structure exploitation technique used in [BS05] and [Mee08] can be adapted to Algorithm 1. This can be seen from the fact that all elements of the vector f in Figure 3.1 are linear combinations of the vectors in the first block row of V. The propagation is also present in the matrix vector product (Theorem 3.1). The algorithm can hence be optimized to use only the first block row or last column of V, i.e., $O(kn)$.

Although the technique in [Mee08] reduces the memory cost by a factor of roughly 50% for the solution of the quadratic eigenvalue problem, the computational cost slightly increases and there may be a risk of loss of numerical stability in some cases. Due to the fact that the numerical stability properties of the implicit representation are unclear and that memory is not an issue in the examples here, this optimization is not necessary for the purposes of this paper.

4. Equivalence with standard Arnoldi with linear infinite dimensional operator. The original problem to find λ is already a standard eigenvalue problem in the sense that λ is an eigenvalue of the infinite dimensional operator $\mathcal{A}$,

$$\mathcal{A}z = \lambda z$$

where $z \in \mathcal{C}([-\tau_m,\ 0],\ \mathbb{C}^n)$. Since it is a linear operator, it is natural to consider Arnoldi applied to $\mathcal{A}^{-1}$ in an abstract setting, such that Arnoldi constructs a Krylov subspace of functions, i.e.,

$$\mathcal{K}_k(\mathcal{A}^{-1}, \varphi) := \operatorname{span}\{\varphi, \mathcal{A}^{-1}\varphi, \ldots, \mathcal{A}^{-(k-1)}\varphi\}, \tag{4.1}$$

and projecting on it. In this section we will see that Algorithm 1 has a complete interpretation in this setting if a scalar product is appropriately defined. The vector in Algorithm 1 turns out to play the same role as the coefficients in the Chebyshev expansion. The Krylov subspace (4.1) is constructed for the inverse of $\mathcal{A}^{-1}$ which is explicitly given as follows.

PROPOSITION 4.1 (The inverse of $\mathcal{A}$). *The inverse of $\mathcal{A} : X \to X$ exists iff $A_0 + \sum_{i=1}^m A_i$ is regular. Moreover, it is explicitly given as*

$$\begin{array}{lcl} \mathcal{D}(\mathcal{A}^{-1}) & = & X \\ \left(\mathcal{A}^{-1}\ \phi\right)(\theta) & = & \int_0^\theta \phi(s)ds + C(\phi),\ \theta \in [-\tau_m,\ 0], \qquad \phi \in \mathcal{D}(\mathcal{A}^{-1}), \end{array} \tag{4.2}$$

where the constant $C(\phi)$ satisfies

$$C(\phi) = \left(A_0 + \sum_{i=1}^m A_i\right)^{-1} \left[\phi(0) - \sum_{i=1}^m A_i \int_0^{-\tau_i} \phi(s)ds\right]. \tag{4.3}$$

Proof. First assume $A_0 + \sum_{i=1}^{m} A_i$ is regular and note that if $\phi \in X$ then ϕ is continuous and bounded on the closed interval $[-\tau_m, 0]$. Hence, the integrals in (4.2) and (4.3) exist and (4.2) defines an operator, which we first denote by $\mathcal{T}$. It can be easily verified that

$$\begin{aligned} \mathcal{T}\mathcal{A}\phi &= \phi, \quad \phi \in \mathcal{D}(\mathcal{A}), \\ \mathcal{A}\mathcal{T}\phi &= \phi, \quad \phi \in \mathcal{D}(\mathcal{T}). \end{aligned}$$

Hence, $\mathcal{T} = \mathcal{A}^{-1}$. It remains to show that the inverse is not uniquely defined if $A_0 + \sum_{i=1}^{m} A_i$ is singular. Let v be a null vector of $A_0 + \sum_{i=1}^{m} A_i$. Consider a constant function $\varphi(t) = kv$ with k an arbitrary complex number. We can now apply $\mathcal{A}$ to φ with the definition (2.1) since $\varphi \in \mathcal{D}(\mathcal{A})$. Moreover, $\mathcal{A}\varphi = 0$. Since k is arbitrary, several functions are mapped to zero and the inverse of $\mathcal{A}$ is not uniquely defined. □

4.1. Action and Krylov subspace equivalence. The key to the functional setting duality of this section is that we consider a scaled and shifted Chebyshev expansion of entire functions. Consider the expansion of two entire functions ψ and ϕ in series of of scaled Chebyshev polynomials,

$$\begin{aligned} \phi(t) &= \sum_{i=0}^{\infty} c_i T_i \left(2\tfrac{t}{\tau_m} + 1\right) \\ \psi(t) &= \sum_{i=0}^{\infty} d_i T_i \left(2\tfrac{t}{\tau_m} + 1\right), \quad t \in [-\tau_m,\ 0]. \end{aligned} \tag{4.4}$$

We will now see that the operation $\psi = \mathcal{A}^{-1}\phi$ can be expressed as a mapping of the coefficients, $c_0, c_1, \ldots$ and $d_0, d_1, \ldots$. This mapping turns out to reduce to the matrix vector product in Theorem 3.1. Suppose $\psi = \mathcal{A}^{-1}\phi$, then

$$\psi \in \mathcal{D}(\mathcal{A}), \tag{4.5}$$

$$\mathcal{A}\psi = \psi' = \phi. \tag{4.6}$$

From the fact that the derivative of a Chebyshev polynomial of the first kind can be expressed as a Chebyshev polynomial of the second kind, we note that

$$\psi'(t) = \sum_{i=1}^{\infty} \tfrac{2d_i i}{\tau_m} U_{i-1} \left(2\tfrac{t}{\tau_m} + 1\right).$$

Moreover, the relation between Chebyshev polynomials of the first kind and Chebyshev polynomials of the second kind, i.e., property (2.13), yields

$$\begin{aligned} \phi(t) &= c_0 U_0 \left(2\tfrac{t}{\tau_m} + 1\right) + \tfrac{1}{2} U_1 \left(2\tfrac{t}{\tau_m} + 1\right) + \sum_{i=2}^{\infty} \tfrac{c_i}{2} \left(U_i \left(2\tfrac{t}{\tau_m} + 1\right) - U_{i-2} \left(2\tfrac{t}{\tau_m} + 1\right)\right) \\ &= c_0 U_0 \left(2\tfrac{t}{\tau_m} + 1\right) + \tfrac{1}{2} c_1 U_1 \left(2\tfrac{t}{\tau_m} + 1\right) \\ &\quad + \sum_{i=3}^{\infty} \tfrac{c_{i-1}}{2} U_{i-1} \left(2\tfrac{t}{\tau_m} + 1\right) - \sum_{i=1}^{\infty} \tfrac{c_{i+1}}{2} U_{i-1} \left(2\tfrac{t}{\tau_m} + 1\right) \end{aligned}$$

By matching coefficients in (4.6) we obtain the following recurrence relation for the coefficients,

$$d_i = \begin{cases} \frac{\tau_m}{4}(2c_0 - c_2) & i = 1, \\ \frac{\tau_m}{4} \frac{c_{i-1} - c_{i+1}}{i} & i \geq 2. \end{cases} \tag{4.7}$$

From (4.5) and (4.6) we get

$$\phi(0) = A_0\psi(0) + \sum_{k=1}^{m} A_k \psi(-\tau_k).$$

Hence,

$$\sum_{i=0}^{\infty} c_i T_i(1) = \sum_{i=0}^{\infty}\sum_{k=0}^{m} A_k T_i\left(-2\frac{\tau_k}{\tau_m}+1\right) d_i = \sum_{i=0}^{\infty} R_i d_i. \tag{4.8}$$

By combining the results above and the fact that the Chebyshev coefficients of entire functions decay exponentially [Tre00, Theorem 1] (as they are the Fourier coefficients of an entire function) we have proven the following relation.

THEOREM 4.2 (Action equivalence). *Consider two entire functions ϕ and ψ and the associated Chebyshev expansion*

$$\begin{array}{rcl} \phi(t) & = & \sum_{i=0}^{\infty} c_i T_i\left(2\frac{t}{\tau_m}+1\right) \\ \psi(t) & = & \sum_{i=0}^{\infty} d_i T_i\left(2\frac{t}{\tau_m}+1\right), \quad t\in[-\tau_m,\ 0]. \end{array} \tag{4.9}$$

Denote $c = (c_0^T, c_1^T, \ldots)^T, d = (d_0^T, d_1^T, \ldots)^T \in \mathrm{vec}(\mathbb{C}^{n\times\infty})$. Suppose $\sum_{i=0}^{m} A_i$ is non-singular. Then the following two statements are equivalent

i) $\psi = \mathcal{A}^{-1}\phi$

ii) $d = \Sigma_\infty^{-1}\Pi_\infty c$ where c, d fulfill (4.7)-(4.8).

Moreover, if $c = (c_0^T, \ldots, c_{k-1}^T, 0, \ldots, 0)^T = \mathrm{vec}(Y, 0, \ldots, 0)$, $Y \in \mathbb{R}^{n\times k}$ then $d = \mathrm{vec}(\hat{x}, Z, 0, \ldots, 0)$ in where $\hat{x}$ and Z are the formulas in Theorem 3.1.

REMARK 4.3 (Krylov subspace equivalence). *Note the equivalence between $\mathcal{A}^{-1}$ and $\Sigma_\infty^{-1}\Pi_\infty$ in Theorem 4.2 propagates to an equivalence between the Krylov subspace. Let $\phi_0(t) = x_0$ be a constant function. Theorem 4.2 implies that*

$$\phi \in \mathcal{K}_k(\mathcal{A}^{-1}, \phi_0),$$

if and only if

$$\mathrm{vec}(c_0, c_1, \ldots, c_{k-1}) \in \mathcal{K}_k(\Sigma_k^{-1}\Pi_k, \mathrm{vec}(x_0, 0, \ldots, 0)),$$

where $c_0, \ldots, c_{k-1}$ are the Chebyshev coefficients of ϕ, i.e., (4.9).

4.2. Orthogonalization equivalence. We saw in Section 4.1 that the matrix vector operation associated with $\Sigma_\infty^{-1}\Pi_\infty$ is equivalent to the operation $\mathcal{A}^{-1}$, in the sense that $\mathcal{A}$ applied to a function corresponds to a map between Chebyshev coefficients of two functions. The associated Krylov subspaces are also equivalent.

Arnoldi is a way to project on a Krylov subspace. In order to define the projection and computate the elements of H_k, we need to define a scalar product. In Algorithm 1 we use the natural way to define an scalar product, the Euclidiedian scalar product on the Chebyshev coefficients. In order to define an projection equivalent to Algorithm 1 in a consistent way, we define a scalar product in the function setting as

$$\langle \phi, \psi \rangle := c^* d = \sum_{i=0}^{\infty} c_i^* d_i, \tag{4.10}$$

where ϕ, ψ, c and d are as in Theorem 4.2. We combine this with Theorem 4.2 to conclude that the Hessenberg matrix generated in Algorithm 1 and the Hessenberg matrix generated by standard Arnoldi applied to $\mathcal{A}^{-1}$ with the scalar product (4.10) are equal.

THEOREM 4.4 (Hessenberg equivalence). *Let ϕ, ψ, c and d be as in Theorem 4.2. With the definition of the scalar product* (4.10) *the Hessenberg matrix computed (with*

exact arithmetic) in Algorithm 1 is identical to the Hessenberg matrix of Arnoldi applied to $\mathcal{A}^{-1}$ *with the scalar product* (4.10) *and the constant starting vector* $\varphi(t) = x_0$.

The definition (4.10) involves the coefficients of a Chebyshev expansion. We will now see that this definition can be reformulated to an explicit expression with weighted integrals involving the functions ϕ and ψ. First note that Chebyshev polynomials are orthogonal (but not orthonormal) in the following sense,

$$\frac{2}{\tau_m}\int_{-\tau_m}^{0} \frac{T_i(\frac{2}{\tau_m}t+1)T_j(\frac{2}{\tau_m}t+1)}{\sqrt{1-(\frac{2}{\tau_m}t+1)^2}}\,dt = \int_{-1}^{1}\frac{T_i(x)T_j(x)}{\sqrt{1-x^2}}\,dx = \begin{cases} 0, & i \neq j, \\ \pi, & i = j = 0, \\ \pi/2 & i = j \neq 0. \end{cases}$$

In order to simplify the notation, we introduce the functional

$$I(f) := \frac{2}{\tau_m}\int_{-\tau_m}^{0} \frac{f(t)}{\sqrt{1-(\frac{2}{\tau_m}t+1)^2}}\,dt.$$

We will now show that

$$\langle \phi, \psi \rangle = \frac{2}{\pi} I(\phi^*\psi) - \frac{1}{\pi^2} I(\phi^*) I(\psi), \tag{4.11}$$

where $(\phi^*\psi)(t) = \phi(t)^*\psi(t)$, by inserting the expansion of ϕ and ψ, i.e., (4.4), into (4.11). Note that from the orthogonality of Chebyshev polynomials we have that

$$I(\phi^*\psi) = \sum_{i,j=0}^{\infty} c_i^* d_j \frac{2}{\tau_m}\int_{-\tau_m}^{0} \frac{T_i(\frac{2}{\tau_m}t+1)T_j(\frac{2}{\tau_m}t+1)}{\sqrt{1-(\frac{2}{\tau_m}t+1)^2}}\,dt = \frac{\pi}{2}(\sum_{i=0}^{\infty} c_i^* d_j + c_0^* d_0),$$

and from the fact that $T_0(x) = 1$,

$$I(\phi^*) = \sum_{i=0}^{\infty} c_i^* \frac{2}{\tau_m}\int_{-\tau_m}^{0} \frac{T_i(\frac{2}{\tau_m}t+1)T_0(\frac{2}{\tau_m}t+1)}{\sqrt{1-(\frac{2}{\tau_m}t+1)^2}}\,dt = \pi c_0^*.$$

Analogously $I(\psi) = \pi d_0$. We have shown that $\langle \phi, \psi \rangle = \sum_{i=0}^{\infty} c_i^* d_i$.

REMARK 4.5 (Computation with functions). *From the reasoning above we see that Algorithm 1 can be interpreted as Arnoldi applied to* $\mathcal{A}^{-1}$ *with the scalar product* (4.11) *for functions on* $\mathcal{C}([-\tau_m,\ 0],\ \mathbb{C}^n)$ *with a constant starting function, where the computation is carried out by mapping Chebyshev coefficients. We note that the representation of functions with Chebyshev coefficients and associated manipulations are also done in the software package* `chebfun` *[BT04].*

REMARK 4.6 (Residual equivalence). *Note that there is a complete duality between* $\Sigma_\infty^{-1}\Pi_\infty$ *and* $\mathcal{A}^{-1}$. *A direct consequence is that the residual norm, which can be used as a stopping criteria as described in Remark 3.2, has an interpretation as the norm of function residual with respect to the norm induced by the scalar product* (4.10). *That is,* $\|(\Sigma_\infty^{-1}\Pi_\infty - \mu)\tilde{u}\|_2 = \|(\Sigma_N^{-1}\Pi_N - \mu)u\|_2 = h_{k+1,k}|e_k^T z| = \|\mathcal{A}^{-1}\phi - \mu\phi\|_c := \sqrt{\langle \mathcal{A}^{-1}\phi - \mu\phi, \mathcal{A}^{-1}\phi - \mu\phi\rangle}$.

4.3. Block Arnoldi and full spectral discretization. In the setting of functions, the scalar product (4.10) and (4.11) seem artificial in the sense that one can

not easily identify a property why this definition is better than any other definition of a scalar product.

In fact, in earlier works [JMM10] we derived a scheme similar to Algorithm 1 by using a Taylor expansion instead of a Chebyshev discretization. It is not difficult to show that the Taylor approach also can be interpreted in a function setting, where the scalar product is defined such that the monomials are orthogonal, i.e., $\langle \phi, \psi \rangle_T := (1/2\pi) \int_0^{2\pi} \phi(e^{i\theta})^* \psi(e^{i\theta})\, d\theta$.

In this paper, the attractive convergence properties of Algorithm 1 are motivated by the connection with the discretization scheme. Discretization schemes similar to what we present in Section 2 have in the literature and are known to be efficient for the delay eigenvalue problem.

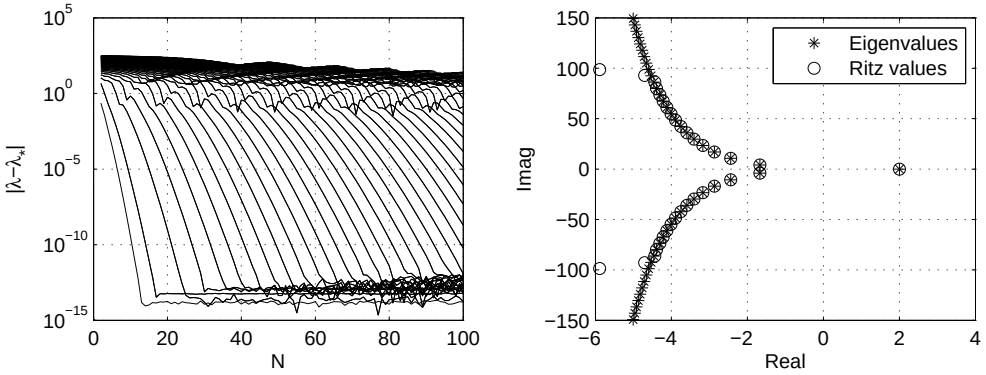
Algorithm 1 is not only Arnoldi applied to the discretized problem. The block version of Algorithm 1 produces the same approximation as the full discretization. This can be seen as follows.

Block Arnoldi is a variant of Arnoldi, where each vector is replaced by a number of vectors, which are kept orthogonal in a block sense. Block Arnoldi is described in [BDD+00]. It is straightforward to construct a block version of Algorithm 1. In the following result we see that this construction is equivalent to the full spectral approach, if we choose the block size n, i.e., equal to the dimension of the system.

THEOREM 4.7. *Let* $V_{[N]} = [V_1, \ldots, V_N]$, $V_i^T = [W_{i,1}, \ldots, W_{i,i}, 0, \cdots]$ *where* $W_{i,j} \in \mathbb{R}^{n\times n}$. *Suppose* $V_{[N]}$ *is orthogonal, i.e.,* $V_{[N]}^* V_{[N]} = I \in \mathbb{R}^{nN\times nN}$, *then*

$$H = V_{[N]}^* \Sigma_\infty^{-1} \Pi_\infty V_{[N]} \sim \Sigma_N^{-1} \Pi_N \sim \mathcal{A}_N^{-1}$$

In words, when performing N steps of block Arnoldi, the resulting Ritz values are the same approximations as those from [BMV05] (but with the grid points (2.21)) where N discretization points are used.



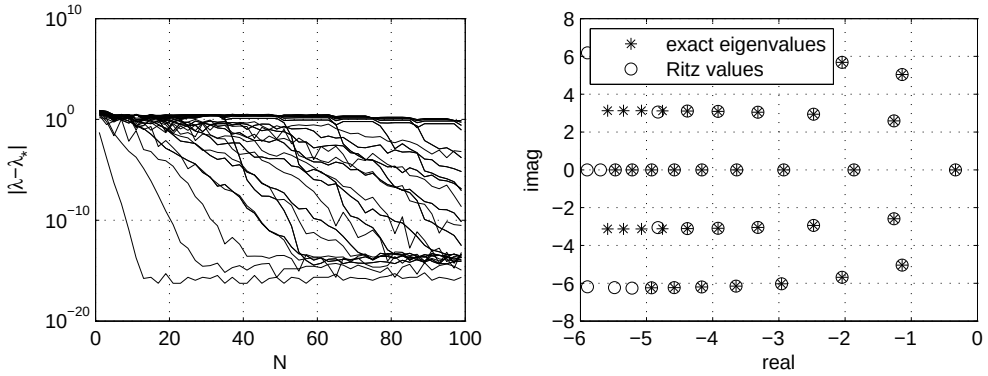
(a) Absolute error

(b) Eigenvalues and Ritz values

FIGURE 4.1. *Convergence and eigenvalue approximations for Example 4.8.*

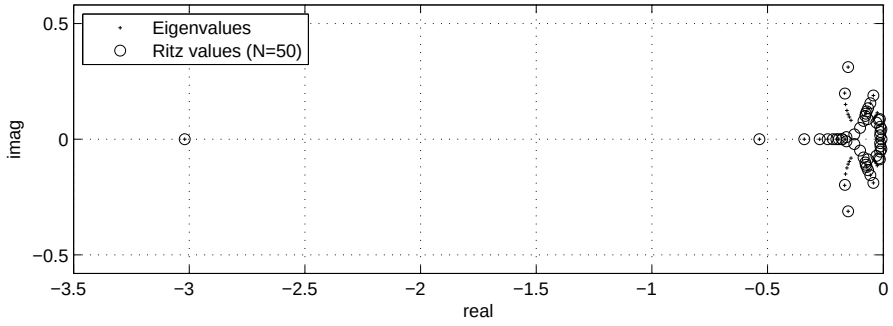
EXAMPLE 4.8. *Consider the scalar DDE*

$$\dot{x}(t) = (2 - e^{-2})x(t) + x(t-1),$$



(a) Absolute error

(b) Eigenvalues and Ritz values



(c) The inverted spectrum

FIGURE 4.2. *Convergence and eigenvalue approximations for the example in Section 5.*

which is also studied in [Bre06]. We use this example as a simplified illustration of the block Arnoldi equivalence in Theorem 4.7. In this case Algorithm 1 is equivalent to standard Arnoldi and at the same time equivalent to the full spectral discretization approach given in Section 2.

The convergence of the Ritz values are shown in Figure 4.1a. We observe geometric convergence, i.e., asymptotically linear error curves in a lin-log-plot. This observation is consistent with the convergence theory for standard Arnoldi as well as the convergence theory for the spectral discretization method. The spectral method is expected to converge exponentially [Tre00]. (Note that geometric convergence is a type of exponential convergence.) The generic situation for Arnoldi is that the angle between the Krylov subspace and the eigenvector converge geometrically [Saa92, Section VI.7].

5. Numerical Example. The numerical methods for sparse standard eigenvalue problems have turned out to be very useful because many applications are discretizations of partial differential equations which are by construction large and

sparse. In order to illustrate that the presented method is particularly well suited for large and sparse problems we consider a PDE with a delayed term. See [Wu96] for further phenomena modeled as PDEs with delay. Consider

$$\frac{\partial v(x,t)}{\partial t} = \frac{\partial^2 v(x,t)}{\partial x^2} + a_0(x)v(x,t) + a_1(x)v(\pi - x, t-1).$$

with $a_0(x) = -2\sin(x)$, $a_1(x) = 2\sin(x)$ and $v_x(0,t) = v_x(\pi,t) = 0$.

We discretize the PDE and get a DDE by approximating the second derivative in time with central difference. This is a variant of a PDE considered in [BMV09], where we have modified the delayed term. In the original formulation, the matrices are tridiagonal, which is not representative for our purposes.

The convergence is visualized in Figure 4.2. For a system of size $n = 5000$, we do 100 Arnoldi iterations, in a total cpu-time $69s$. For this particular example, the linear system $(A_0 + A_1)^{-1}x$ can be solved very efficiently. In our implementation we use the factorization Matlab factorization `[L,U,P,Q]=lu(A0+A1)`. In fact, the cpu-time consumption in the orthogonalization part was $43s$ whereas the cpu-time consumption for the linear system part only $25s$. Note that in Figure 4.2 we see that after 30 iterations we have 4 eigenvalues to reasonable accuracy.

We see in Figure 4.2c that the iteration converges first to the extreme eigenvalues of the inverted spectrum (which are well isolated). This is the behavior we would expect from standard Arnoldi.

6. Conclusions and outlook. The approach of this paper is in a sense very natural. It is known from the literature that spectral discretization methods tend to be efficient for the DEP. The main computational part of a discretization approach is to solve a large eigenvalue problem. Arnoldi is typically very efficient for large standard and generalized eigenvalue problem. Our construction is natural in the sense that we combine an efficient discretization method (a spectral discretization) with an efficient eigenvalue solver (Arnoldi) and exploit the non-zero pattern in the iteration vectors and the connection with an infinite dimensional operator. The approach is also natural in terms of error. The error of Arnoldi and the spectral discretization are both exponential and hence roughly balanced.

Although the approach is very natural, several issues related to Arnoldi appear difficult to extend in a natural way. We will now list some techniques and theory for standard Arnoldi which appear to extend easily and some which appear to be more involved.

Algorithm 1 can conceptually be fitted with explicit or implicit restart (as in e.g. [Sor92, LS96]) after k iterations by restarting the iteration with a vector of length kn. However, the reduction of processing time and memory would not be as dramatic as the standard case since the starting vector would be of length kn. There are different approaches to convergence theory of the Arnoldi method. Some of the convergence theory in [Saa92] is expressed in terms of angles between subspaces. The scalar product in Section 4 induces an angle definition, and it is to expect that at least some theory in [Saa92] is applicable with the appropriate angle definition. There is also theory based on potential theory [Kui06].

In this paper we assumed we are looking for eigenvalues close to the origin. Note that this assumption is not a restriction since the matrices $A_0, \ldots, A_m$ can be shifted and scaled such that an arbitrary point is shifted to the origin. Changing the shift throughout the iteration in the sense of rational Krylov [Ruh98] seems somewhat involved.

Acknowledgment. This article present results of the Belgian Programme on Interuniversity Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture, the Optimization in Engineering Centre OPTEC of the K.U.Leuven, and the project STRT1-09/33 of the K.U.Leuven Research Foundation.

REFERENCES

[ACL09] A. Amiraslani, R. Corless, and P. Lancaster. Linearization of matrix polynomials expressed in polynomial bases. *IMA J. Numer. Anal.*, 29(1):141–157, 2009.

[ADLK01] P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, and J. Koster. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM J. Matrix Anal. Appl.*, 23(1):15–41, 2001. `http://graal.ens-lyon.fr/MUMPS/`.

[Arn51] W. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Q. appl. Math.*, 9:17–29, 1951.

[BDD+00] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. A. van der Vorst, editors. *Templates for the solution of algebraic eigenvalue problems. A practical guide*. SIAM, Society for Industrial and Applied Mathematics, 2000.

[BM00] A. Bellen and S. Maset. Numerical solution of constant coefficient linear delay differential equations as abstract Cauchy problems. *Numer. Math.*, 84(3):351–374, 2000.

[BMV05] D. Breda, S. Maset, and R. Vermiglio. Pseudospectral differencing methods for characteristic roots of delay differential equations. *SIAM J. Sci. Comput.*, 27(2):482–495, 2005.

[BMV06] D. Breda, S. Maset, and R. Vermiglio. Pseudospectral approximation of eigenvalues of derivative operators with non-local boundary conditions. *Applied Numerical Mathematics*, 56:318–331, 2006.

[BMV09] D. Breda, S. Maset, and R. Vermiglio. Numerical approximation of characteristic values of partial retarded functional differential equations. *Numer. Math.*, 2009. doi:10.1007/s00211-009-0233-7.

[Bre06] D. Breda. Solution operator approximations for characteristic roots of delay differential equations. *Appl. Numer. Math.*, 56:305–317, 2006.

[BS05] Z. Bai and Y. Su. SOAR: A second-order Arnoldi method for the solution of the quadratic eigenvalue problem. *SIAM J. Matrix Anal. Appl.*, 26(3):640–659, 2005.

[BT04] Z. Battles and L. N. Trefethen. An extension of MATLAB to continuous functions and operators. *SIAM J. Sci. Comput.*, 25(5):1743–1770, 2004.

[BV04] T. Betcke and H. Voss. A Jacobi-Davidson type projection method for nonlinear eigenvalue problems. *Future Generation Computer Systems*, 20(3):363–372, 2004.

[Dav04] T. A. Davis. Algorithm 832: UMFPACK V4.3 – an unsymmetric-pattern multifrontal method. *ACM Trans. Math. Softw.*, 30(2):196–199, 2004.

[DEG+99] J. Demmel, S. Eisenstat, J. R. Gilbert, X.-G. Li, and J. W. Liu. A supernodal approach to sparse partial pivoting. *SIAM J. Matrix Anal. Appl.*, 20(3):720–755, 1999.

[ELR02] K. Engelborghs, T. Luzyanina, and D. Roose. Numerical bifurcation analysis of delay differential equations using DDE-BIFTOOL. *ACM Trans. Math. Softw.*, 28(1):1–24, 2002.

[ELS01] K. Engelborghs, T. Luzyanina, and G. Samaey. DDE-BIFTOOL v. 2.00: a Matlab package for bifurcation analysis of delay differential equations. Technical report, K.U.Leuven, Leuven, Belgium, 2001.

[HV93] J. Hale and S. M. Verduyn Lunel. *Introduction to functional differential equations*. Springer-Verlag, 1993.

[Jar08] E. Jarlebring. *The spectrum of delay-differential equations: numerical methods, stability and perturbation*. PhD thesis, TU Braunschweig, 2008.

[JMM10] E. Jarlebring, K. Meerbergen, and W. Michiels. An Arnoldi method with structured starting vectors for the delay eigenvalue problem. In *Proceedings of the 9th IFAC workshop on time-delay systems, Prague*, 2010. submitted.

[Kre09] D. Kressner. A block Newton method for nonlinear eigenvalue problems. *Numer. Math.*, 114(2):355–372, 2009.

[Kui06] A. B. Kuijlaars. Convergence analysis of Krylov subspace iterations with methods from potential theory. *SIAM Rev.*, 48(1):3–40, 2006.

[Lan02] P. Lancaster. *Lambda-matrices and vibrating systems*. Mineola, NY: Dover Publications, 2002.

[LS96] R. Lehoucq and D. Sorensen. Deflation techniques for an implicitly restarted arnoldi iteration. *SIAM J. Matrix Anal. Appl.*, 17(4):789–821, 1996.

[LSY98] R. Lehoucq, D. Sorensen, and C. Yang. *ARPACK user's guide. Solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*. SIAM publications, 1998.

[Mee08] K. Meerbergen. The quadratic Arnoldi method for the solution of the quadratic eigenvalue problem. *SIAM J. Matrix Anal. Appl.*, 30(4):1463–1482, 2008.

[MMMM06] S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann. Vector spaces of linearizations for matrix polynomials. *SIAM J. Matrix Anal. Appl.*, 28:971–1004, 2006.

[MN07] W. Michiels and S.-I. Niculescu. *Stability and Stabilization of Time-Delay Systems: An Eigenvalue-Based Approach*. Advances in Design and Control 12. SIAM Publications, Philadelphia, 2007.

[MV04] V. Mehrmann and H. Voss. Nonlinear eigenvalue problems: A challenge for modern eigenvalue methods. *GAMM Mitteilungen*, 27:121–152, 2004.

[Neu85] A. Neumaier. Residual inverse iteration for the nonlinear eigenvalue problem. *SIAM J. Numer. Anal.*, 22:914–923, 1985.

[Ruh73] A. Ruhe. Algorithms for the nonlinear eigenvalue problem. *SIAM J. Numer. Anal.*, 10:674–689, 1973.

[Ruh98] A. Ruhe. Rational Krylov: A practical algorithm for large sparse nonsymmetric matrix pencils. *SIAM J. Sci. Comput.*, 19(5):1535–1551, 1998.

[Saa92] Y. Saad. *Numerical methods for large eigenvalue problems*. Manchester University Press, 1992.

[Sch08] K. Schreiber. *Nonlinear Eigenvalue Problems: Newton-type Methods and Nonlinear Rayleigh Functionals*. PhD thesis, TU Berlin, 2008.

[SG04] O. Schenk and K. Gärtner. Solving unsymmetric sparse systems of linear equations with PARDISO. *Future Generation Computer Systems*, 20(3):475–487, 2004.

[Sor92] D. Sorensen. Implicit application of polynomial filters in a k-step Arnoldi method. *SIAM J. Matrix Anal. Appl.*, 13(1):357–385, 1992.

[Tre00] L. N. Trefethen. *Spectral Methods in MATLAB*. SIAM Publications, Philadelphia, 2000.

[VLR08] K. Verheyden, T. Luzyanina, and D. Roose. Efficient computation of characteristic roots of delay differential equations using LMS methods. *J. Comput. Appl. Math.*, 214(1):209–226, 2008. doi: 10.1016/j.cam.2007.02.02.

[Vos04] H. Voss. An Arnoldi method for nonlinear eigenvalue problems. *BIT numerical mathematics*, 44:387 – 401, 2004.

[VZ09] T. Vyhlídal and P. Zítek. Mapping based algorithm for large-scale computation of quasi-polynomial zeros. *IEEE Trans. Autom. Control*, 54(1):171–177, 2009.

[Wu96] J. Wu. *Theory and applications of partial functional differential equations*. Applied Mathematical Sciences. 119. New York, NY: Springer., 1996.