The infinite Arnoldi method and an application to time-delay systems with distributed delays

Elias Jarlebring, Wim Michiels, Karl Meerbergen

Abstract The Arnoldi method, which is a well-established numerical method for standard and generalized eigenvalue problems, can conceptually be applied to standard but infinite-dimensional eigenvalue problems associated with an operator. In this work, we show how such a construction can be used to compute the eigenvalues of a time-delay system with distributed delays, here given by $\dot{x}(t) = A_0x(t) + A_1x(t-\tau) + \int_{-\tau}^{0} F(s)x(t+s)\,ds$, where $A_0, A_1, F(s) \in \mathbb{C}^{n\times n}$. The adaption is based on formulating a more general problem as an eigenvalue problem associated with an operator and showing that the action of this operator has a finite-dimensional representation when applied to polynomials. This allows us to implement the infinite-dimensional algorithm using only (finite-dimensional) operations with matrices and vectors of size n. We show, in particular, that for the case of distributed delays, the action can be computed from the Fourier cosine transform of a function associated with F, which in many cases can be formed explicitly or computed efficiently.

1 Introduction

Consider a linear time-invariant time-delay system with a distributed delay term,

$$\dot{x}(t) = A_0x(t) + A_1x(t-\tau) + \int_{-\tau}^{0} F(s)x(t+s)\,ds. \tag{1}$$

Time-delay systems without distributed terms ($F(s) = 0$) have been extensively studied in the literature. See, e.g., the survey papers [7, 19]. There are

Elias Jarlebring, Wim Michiels, Karl Meerbergen
KU Leuven, Celestijnenlaan 200 A, 3001 Heverlee, Belgium e-mail: {elias.jarlebring, wim.michiels,karl.meerbergen}@cs.kuleuven.be

however numerous applications where the delay is not localized to one value, e.g., when the derivative of the current state depends on an expectation value of a continuum of previous states. In such situations, it is natural to consider the generalized form (1). See [13] and [4] and references therein for literature on time-delay systems with distributed delays.

The eigenvalues of (1) are given by the solutions to the nonlinear eigenvalue problem,

$$M(\lambda)v = 0, \tag{2}$$

where $\lambda \in \mathbb{C}$ and $v \in \mathbb{C}\backslash\{0\}$ and

$$M(\lambda) := -\lambda I + A_0 + A_1 e^{-\lambda\tau} + \int_{-\tau}^{0} F(s)e^{\lambda s}\, ds. \tag{3}$$

Our approach is based on first (in Section 2) reformulating the problem (2) as a linear infinite-dimensional eigenvalue problem. The associated operator is an integration operator and we show that for this operator we can carry out the method, which for standard eigenvalue problems is known as the *Arnoldi method*.

We show that the action of the operator (and the corresponding Arnoldi algorithm) can be carried out with standard linear algebra operations, including operations with the matrices

$$R_j := \int_{-\tau}^{0} F(s)\hat{T}_j(s), j = 0, \ldots, N,$$

where $\hat{T}_j$ are scaled and shifted Chebyshev polynomials. The coefficients R_j can be computed before the iteration starts and can often be computed accurately and cheaply.

The attractive properties of the method which are a consequence of the equivalence with the Arnoldi method are illustrated with examples in Section 5.

Although there are numerous methods for the eigenvalues of (1) with $F(s) = 0$, e.g., [6, 22], only a few approachs have been generalized to the distributed case, e.g., [4, 13]. There exist also many numerical methods for the nonlininear eigenvalue problem (2), presented in different generality settings. See reviews in [15,20] and problem collection [2]. There exists methods based on subspace projection [1,14,21], and iterative locally convergent methods [12,17].

2 Operator formulation

Suppose $\lambda = 0$ is not a solution and consider the transformation of the nonlinear eigenvalue problem (2) by introducing

$$B(\lambda) := \frac{1}{\lambda} M(0)^{-1}(M(0) - M(\lambda)). \tag{4}$$

This implies that

$$\lambda B(\lambda)x = x. \tag{5}$$

We will assume that B is an entire function, where in the case of (4) we define B such that it is analytic also in $\lambda = 0$, by also assuming that $M(0)$ is non-singular. With this definition of B we can introduce an operator $\mathcal{B}$, which reciprocal eigenvalues turn out to be the solutions to the nonlinear eigenvalue problem (5) or equivalently (2).

Definition 1 (The operator $\mathcal{B}$). Let $\mathcal{B}$ denote the operator defined by the domain $\mathcal{D}(\mathcal{B}) := \{\varphi \in C_\infty(\mathbb{C}, \mathbb{C}^n) : \sum_{i=0}^{\infty} B_i \varphi^{(i)}(0) \text{ is finite}\}$ and the action

$$(\mathcal{B}\varphi)(\theta) = C(\varphi) + \int_0^\theta \varphi(\theta)\, d\theta, \tag{6}$$

where

$$C(\varphi) := \sum_{i=0}^{\infty} B_i \varphi^{(i)}(0) = \left(B(\frac{d}{d\theta})\varphi \right)_{\theta=0}. \tag{7}$$

Theorem 1 (Operator equivalence). *Let $x \in \mathbb{C}^n \backslash \{0\}$, $\lambda \in \mathbb{C}$ and denote $\varphi(\theta) := xe^{\lambda\theta}$. Then the following statements are equivalent.*

i) The pair (λ, x) is a solution to the nonlinear eigenvalue problem (5).
ii) The pair (λ, φ) is a solution to the infinite dimensional eigenvalue problem

$$\lambda\mathcal{B}\varphi = \varphi. \tag{8}$$

Moreover, all eigenfunctions of $\mathcal{B}$ depend exponentially on θ, i.e., if $\lambda\mathcal{B}\psi = \psi$ then $\psi(\theta) = xe^{\lambda\theta}$.

Proof. See [11, Theorem 1].

3 The infinite Arnoldi method

In the previous section we saw that the solutions of the nonlinear eigenvalue problem (2) could be characterized as solutions to a linear, but infinite-dimensional eigenvalue problem corresponding to the operator $\mathcal{B}$. We will now carry out the Arnoldi method for the operator $\mathcal{B}$, which by the equivalence above (Theorem 1) will yield solutions of (2).

The Arnoldi method applied to $\mathcal{B}$, can be interpreted as first constructing an orthogonal basis of the space

$$K_k(\mathcal{B}, \varphi) := \text{span}(\varphi, \mathcal{B}\varphi, \ldots, \mathcal{B}^{k-1}\varphi),$$

which is known as a *Krylov subspace*. The orthogonal basis is used to project the operator $\mathcal{B}$ to this space, from which the eigenvalue approximations can be computed.

If φ is a polynomial, then $\mathcal{B}\varphi$ is also a polynomial. In the following theorem we see that the action of $\mathcal{B}$ on a polynomial can be carried out by operations with matrices and vectors. In this work we will focus on representing the polynomials as linear combinations of scaled and shifted Chebyshev polynomials for the interval $[-\tau, 0]$,

$$\hat{T}_j(\theta) := \cos\left(j \arccos\left(\frac{2}{\tau}\theta + 1\right)\right).$$

Theorem 2 (Chebyshev coefficient mapping). *Let φ denote an arbitrary vector of polynomials of degree N and $(x_0, \ldots, x_{N-1}) =: X$ the corresponding coefficients in a Chebyshev basis, for Chebyshev polynomials scaled to the interval $I = [a, b]$, i.e.,*

$$\varphi(\theta) =: \sum_{i=0}^{N-1} \hat{T}_i(\theta) x_i.$$

Moreover, let $y_0, \ldots, y_N$ denote the coefficients of $\mathcal{B}\varphi$, i.e.,

$$(\mathcal{B}\varphi)(\theta) =: \sum_{i=0}^{N} \hat{T}_i(\theta) y_i.$$

Then,

$$(y_1, \ldots, y_N) = X L_{N,N}, \tag{9}$$

where

$$L_{N,N}^T = \frac{b-a}{4} \begin{pmatrix} 2 & 0 & -1 & & & \\ \frac{1}{2} & 0 & -\frac{1}{2} & & & \\ & \frac{1}{3} & 0 & \ddots & & \\ & & \frac{1}{4} & \ddots & -\frac{1}{N-2} & \\ & & & \ddots & 0 & \\ & & & & \frac{1}{N} & \end{pmatrix} \tag{10}$$

and

$$y_0 = \left(\sum_{i=0}^{N-1} B(\frac{d}{d\theta}) \hat{T}_i(\theta) x_i\right)_{\theta=0} - \sum_{i=1}^{N} \hat{T}_i(0) y_i. \tag{11}$$

Proof. See [11, Theorem 2].

Note that all components of the action can be carried out in finite arithmetic, if a finite-dimensional expression for y_0 in (11) can be found. This can indeed be done in many cases. An explicit formula for the case of distributed

delays will be derived in Section 4. See [11, Remark 3] for other approaches to compute y_0.

We now use this action to construct an Arnoldi algorithm for $\mathcal{B}$. In this construction, we have (as for the standard Arnoldi method) that the projected eigenvalue problem can be obtained as a by-product of the orthogonalization. We arrive at the algorithm summarized in Algorithm 1, which we will refer to as the *infinite Arnoldi method*.

We use the following notation. The upper block of the rectangular Hessenberg matrix $\underline{H}_k \in \mathbb{C}^{(k+1)\times k}$ is denoted $H_k \in \mathbb{C}^{k\times k}$ and the (i,j) element of $\underline{H}_k$ is denoted $h_{i,j}$.

Algorithm 1: The infinite Arnoldi method

1. Let $V_1 = x_0/\|x_0\|_2$, $k = 1$, $\underline{H}_0$ =empty matrix
2. For $k = 1, 2, \ldots$ until converged
3. Let $\mathrm{vec}(X) = v_k$
4. Compute $y_1, \ldots, y_{k+1}$ according to (9) with sparse L_k
5. Compute y_0 according to (11)
6. Expand V_k with one block row (zeros)
7. Let $w_k := \mathrm{vec}(y_0, \ldots, y_{k+1})$, compute $h_k = V_k^* w_k$ and then $\hat{w}_k = w_k - V_k h_k$
8. Compute $\beta_k = \|\hat{w}_k\|_2$ and let $v_{k+1} = \hat{w}_k/\beta_k$
9. Let $\underline{H}_k = \begin{bmatrix} \underline{H}_{k-1} & h_k \\ 0 & \beta_k \end{bmatrix} \in \mathbb{C}^{(k+1)\times k}$
10. Expand V_k into $V_{k+1} = [V_k, v_{k+1}]$
11. Compute the eigenvalues $\{\mu_i\}_{i=1}^k$ of the Hessenberg matrix H_k
12. Return approximations $\{1/\mu_i\}_{i=1}^k$

Note that the scalar product, used for the orthogonalization of functions, corresponds to the scalar product of the coefficient vectors in the Chebyshev basis. In [11, Section 6] this choice is motivated, based on a connection with a spectral discretization.

4 Adaption for time-delay systems with distributed delays

The infinite Arnoldi method is a general method and in order to apply it, for each given problem, we have to establish

- the interval on to which the Chebyshev polynomials are scaled, and
- a formula for y_0.

We now wish to adapt the infinite Arnoldi method to (1) and need to consider the interval and y_0. In general, for (linear) functional differential

equations the natural interval is $I = [-\tau, 0]$. This follows from the connection of the algorithm as a discretized problem [11, Section 6.2]. Formulas for y_0 are derived next.

4.1 Computing y_0 for distributed delays

In order to implement Algorithm 1 on a computer, we need to compute y_0, defined by (11), for given values of $x_0, \ldots, x_k$ and possibly $y_1, \ldots, y_{k+1}$.

For notational convenience, we will use the function q defined by

$$q(\lambda, s) := \frac{1 - e^{\lambda s}}{\lambda}. \tag{12}$$

The transformed nonlinear eigenvalue problem (5) can now be expressed as

$$B(\lambda) = \frac{1}{\lambda} M(0)^{-1}(M(0) - M(\lambda)) = \\ (A_0 + A_1 + R_0)^{-1} \left(I + A_1 q(\lambda, -\tau) + R(\lambda)\right), \tag{13}$$

where

$$R(\lambda) := \int_{-\tau}^{0} F(s) q(\lambda, s)\, ds \text{ and } R_0 = \int_{-\tau}^{0} F(s)\, ds.$$

Note that $A_0 + A_1 + R_0$ is invertible if and only if $\lambda = 0$ is not a solution to the nonlinear eigenvalue problem (2). This condition can easily be checked by hand before starting the algorithm.

The last term in (11) is already given in a form which is numerically tractable and we now need to study the first term,

$$\left(\sum_{i=0}^{N-1} B(\frac{d}{d\theta}) \hat{T}_i(\theta) x_i \right)_{\theta=0}. \tag{14}$$

The derivation is based on the fact that the function q corresponds to integration in the sense that

$$\left(q(\frac{d}{d\theta}, s)\varphi \right)_{\theta=0} = \int_{s}^{0} \varphi(\theta)\, d\theta. \tag{15}$$

This follows from inserting the Taylor expansion of φ on the right-hand side and the Taylor expansion of q on the left-hand side. The Taylor expansion of q follows from insertion of the Taylor expansion of $e^{\lambda s}$ in (12). The derivation is also available in [11, Equation (11)].

Now note that in Theorem 2, the function $\psi(\theta) = \sum_{i=1}^{N} \hat{T}_i y_i$ is a primitive function of $\varphi = \sum_{i=0}^{N-1} \hat{T}_i x_i$. By combining this observation with (15), the

term involving R in (14) can be reduced to

$$\sum_{i=0}^{N-1}(R(\frac{d}{d\theta})\hat{T}_i x_i)_{\theta=0} = \int_{-\tau}^{0} R(s)\sum_{i=1}^{N}(\hat{T}_i(0)-\hat{T}_i(s))y_i\,ds = \sum_{i=1}^{N}\left(R_0\hat{T}_i(0)-\int_{-\tau}^{0}F(s)\hat{T}_i(s)\,ds\right)y_i. \quad (16)$$

We will now for notational convenience define,

$$R_i := \int_{-\tau}^{0} F(s)\hat{T}_i(s)\,ds \text{ for } i = 0.\ldots \quad (17)$$

Later, (in Section 4.2) we will see how R_i can be computed in practice. From (16) it follows that the formula for y_0 is equivalently,

$$y_0 = (A_0+A_1+R_0)^{-1}\left(\sum_{i=0}^{N-1}x_i + A_1\sum_{i=1}^{N}(1-\hat{T}_i(-\tau))y_i + \sum_{i=1}^{N}(R_0-R_i)y_i\right) - \sum_{i=1}^{N}y_i, \quad (18)$$

where we used that $\hat{T}_i(0) = 1$. By factorizing the last term into the expression involving $(A_0+A_1+R_0)^{-1}$, we have established a compact formula for y_0. We summarize it as follows.

Theorem 3. *Consider the nonlinear eigenvalue problem* (2) *corresponding to the time-delay system with distributed delays* (1)*. Let B be defined by* (4)*. Let x_i and y_i be as in Theorem 2. Then, $y_0 \in \mathbb{C}^n$, defined in* (11)*, is given by*

$$y_0 = (A_0+A_1+R_0)^{-1}\left(\sum_{i=0}^{N-1}x_i - A_0\sum_{i=1}^{N}y_i - A_1\sum_{i=1}^{N}\hat{T}_i(-\tau)y_i - \sum_{i=1}^{N}R_i y_i\right), \quad (19)$$

where R_i is defined by (17)*.*

4.2 Connection with the Fourier cosine transform

The formula (19) involves R_i, which is given as the integral (17). In many problems, R has such a structure that it can be written as a sum of a low number of constant matrices, times scalar nonlinearities. In those situations the computation reduces to numerically computing integrals of scalar func-

tions. This can usually be done cheaply and accurately, in comparison to the computation time of Algorithm 1, if n is sufficiently large.

In some situations the integral can be computed analytically. For instance, if $F(s) = C$ is constant, then

$$R_k = C\frac{\tau}{2}\frac{(-1)^k - 1}{k}$$

More generally, if $F(s) = Cf(s)$, where

$$f(s) = \begin{cases} 1 & s \in [a, b] \\ 0 & \text{otherwise,} \end{cases}$$

then from the properties of Chebyshev polynomials we have,

$$R_0 = C(b - a) \tag{20a}$$

$$R_1 = C\left(\frac{1}{\tau}(b^2 - a^2) + b - a\right) \tag{20b}$$

$$R_k = \frac{\tau}{4}C\left(\frac{\hat{T}_{k+1}(b) - \hat{T}_{k+1}(a)}{k+1} - \frac{\hat{T}_{k-1}(b) - \hat{T}_{k-1}(a)}{k-1}\right), k > 1. \tag{20c}$$

We now provide an alternative way to compute R_i, by deriving a relation with the Fourier cosine transform. Let $\cos(\theta) = \frac{2}{\tau}s + 1$. Then

$$R_i = \int_{-\tau}^{0} F(s)\cos\left(i\arccos\left(\frac{2}{\tau}s + 1\right)\right) ds = \frac{\tau}{2}\int_0^{\pi} \left[\sin(\theta)F(\frac{\tau}{2}(\cos(\theta) - 1))\right]\cos(i\theta)\, d\theta. \tag{21}$$

Note that (17) is essentially the Fourier cosine transform of $\sin(\theta)F(\frac{\tau}{2}(\cos(\theta) - 1))$. For the computation of the right-hand side of (21), numerical integration methods can be used. For small values of i, this integration is fairly cheap with standard integration software provided F is not oscillating heavily in the integration interval. For large values of i, the integrand is definitely highly oscillating and dedicated integration techniques are highly recommended in order to keep the computational cost low. For further literature on oscillatary integrals see [8, 9, 18] and the references therein.

5 Examples

5.1 Example 1: A rectangular function

We consider the example in [3, 5]

$$\dot{x}(t) = \begin{pmatrix} -3 & 1 \\ -24.646 & -35.430 \end{pmatrix} x(t) + \begin{pmatrix} 1 & 0 \\ 2.35553 & 2.00365 \end{pmatrix} x(t-1) + \begin{pmatrix} 2 & 2.5 \\ 0 & -0.5 \end{pmatrix} \int_{-0.3}^{-0.1} x(t+s)\, ds - \int_{-1}^{-0.5} x(t+s)\, ds. \quad (22)$$

In order to carry out the algorithm, we compute the coefficients R_i analytically. The coefficients are given by

$$R_0 = 0.2 \begin{pmatrix} 2 & 2.5 \\ 0 & -0.5 \end{pmatrix} - 0.5I$$

and

$$R_i = \beta_{0,i} \begin{pmatrix} 2 & 2.5 \\ 0 & -0.5 \end{pmatrix} - \beta_{1,i} I, \quad i = 1, \ldots, 100$$

where we use (20) to compute $\beta_{0,i}, \beta_{1,i}$, $i = 0, \ldots, 100$,

$$(\beta_{0,1}, \beta_{0,2}, \ldots, \beta_{0,100}) \approx (0.1200, -0.0507, -0.1680, -0.1434, \ldots),$$
$$(\beta_{1,1}, \beta_{1,2}, \ldots, \beta_{1,100}) \approx (-0.2500, -0.1667, 0.2500, -0.0333 \ldots).$$

In Figure 1, we observe convergence similar to the standard Arnoldi method, i.e., simultaneous convergence to several eigenvalues. The eigenvalues closest to the origin are found first. After $k = 100$ iterations, 42 eigenvalues are found to an accuracy of 10^{-10}.

5.2 Example 2: A Gaussian distribution

Now consider the following system where the distributed term is a Gaussian distribution,

$$\dot{x}(t) = \frac{1}{10} \begin{pmatrix} 25 & 28 & -5 \\ 18 & 3 & 3 \\ -23 & -14 & 35 \end{pmatrix} x(t) + \frac{1}{10} \begin{pmatrix} 17 & 7 & -3 \\ -24 & -21 & -2 \\ 20 & 7 & 4 \end{pmatrix} x(t-\tau) + \int_{-\tau}^{0} \begin{pmatrix} 14 & -13 & 4 \\ 14 & 7 & 10 \\ 6 & 16 & 17 \end{pmatrix} \frac{e^{(s+\frac{1}{2})^2} - e^{\frac{1}{4}}}{10} x(t+s)\, ds. \quad (23)$$

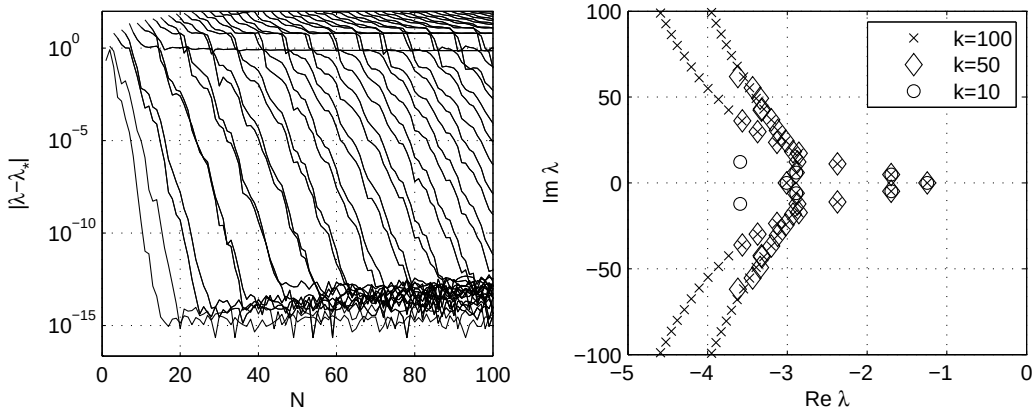

Fig. 1 Convergence and computed solution of the example given in Section 5.1

with $\tau = 1$. Unlike Example 1, there is no simple analytic expression for R_i. The coefficients are

$$R_i = \frac{1}{10}\begin{pmatrix} 14 & -13 & 4 \\ 14 & 7 & 10 \\ 6 & 16 & 17 \end{pmatrix} \beta_i,$$

where we compute β_i, $i = 0, \ldots, 100$ by numerical integration,

$$(\beta_0, \beta_1, \ldots, \beta_{100}) \approx (0, 0.1142, 0, -0.0138, 0, -0.0024, \ldots)$$

Similar to the previous example, we observe (in Figure 2) the convergence characteristic typical for the Arnoldi method. After 100 iterations, we have found 44 eigenvalues to an accuracy 10^{-10}. This success justifies that numerical integration can be used to compute the coefficients R_i.

6 Conclusions

The infinite Arnoldi method is a general method for nonlinear eigenvalue problems which needs to be adapted to the problem at hand. In this work, we considered (in particular) systems with distributed delays (1). The adaption consisted of deriving a computational formula for y_0 in Theorem 3. When comparing to other problems in [10,11], it is clear that the distributed terms appear as additional terms in the formula for y_0. From this observation it is straightforward to generalize the formula by combining results for, e.g., neutral systems with distributed delays.

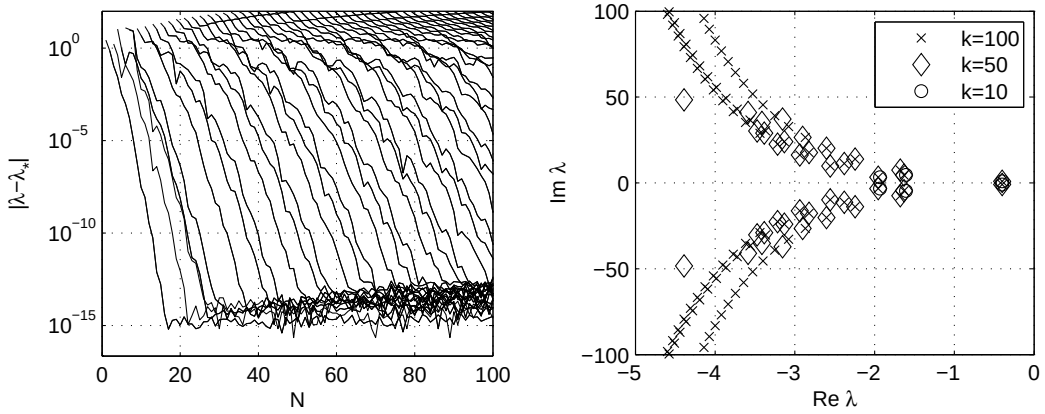

Fig. 2 Convergence and computed solution of the example given in Section 5.2

We also wish to point out that we have here, for clarity, focused on the classical variant of the Arnoldi method. When considering large-scale problems, it is important to consider implementation and other technical aspects of the Arnoldi method. Issues such as restarting and deflation are currently being investigated by the authors. See also the discussion in the conclusions in [10, 11]. The presented algorithm can also be used for model order reduction, similar to the time-delays without distributed delays [16].

Acknowledgements This work represents 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, the projects STRT1-09/33, OT/10/038 of the K.U.Leuven Research Foundation and G0712.11 of the Research Council Flanders (FWO). We thank Prof. Daan Huybrechs and Andreas Asheim of the department of computer science, KU Leuven, for clarifying aspects related to oscillatory integrals.

References

1. 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)
2. T. Betcke, N. J. Higham, V. Mehrmann, C. Schröder, and F. Tisseur.: NLEVP: A collection of nonlinear eigenvalue problems. Technical report, Manchester Institute for Mathematical Sciences (2008)
3. D. Breda.: Solution operator approximations for characteristic roots of delay differential equations. Appl. Numer. Math., **56**:305–317 (2006)
4. 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)

5. D. Breda, S. Maset, and R. Vermiglio.: Pseudospectral approximation of eigenvalues of derivative operators with non-local boundary conditions. Appl. Numer. Math., **56**, 318–331 (2006)
6. 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)
7. K. Gu and S.-I. Niculescu.: Survey on recent results in the stability and control of time-delay systems. J. Dyn. Syst.-T. ASME, **125**, 158–165 (2003)
8. D. Huybrechs.: On the evaluation of highly oscillatory integrals by analytic continuation. SIAM Journal on Numerical Analysis, **44(3)**, 1026–1048 (2006)
9. A. Iserles and P. Nørsett.: Efficient quadrature of highly oscillatory integrals using derivatives. Proceedings of the Royal Society A, **461**, 1383–1399 (2005)
10. E. Jarlebring, K. Meerbergen, and W. Michiels.: A Krylov method for the delay eigenvalue problem. SIAM J. Sci. Comput., **32(6)**, 3278–3300 (2010)
11. E. Jarlebring, W. Michiels, and K. Meerbergen.: A linear eigenvalue algorithm for the nonlinear eigenvalue problem. Technical report, Dept. Comp. Sci., KU Leuven (2010) submitted.
12. D. Kressner.: A block Newton method for nonlinear eigenvalue problems. Numer. Math., **114(2)**, 355–372 (2009)
13. T. Luzyanina, D. Roose, and K. Engelborghs.: Numerical stability analysis of steady state solutions of integral equations with distributed delays. Appl. Numer. Math., **50(1)**, 75–92 (2004)
14. K. Meerbergen.: The quadratic Arnoldi method for the solution of the quadratic eigenvalue problem. SIAM J. Matrix Anal. Appl., **30(4)**, 1463–1482 (2008)
15. V. Mehrmann and H. Voss.: Nonlinear eigenvalue problems: A challenge for modern eigenvalue methods. GAMM Mitteilungen, **27**, 121–152 (2004)
16. W. Michiels, E. Jarlebring, and K. Meerbergen.: Krylov based model order reduction of time-delay systems. Technical report, Dept. Comp. Sci., KU Leuven (2010), submitted.
17. A. Neumaier.: Residual inverse iteration for the nonlinear eigenvalue problem. SIAM J. Numer. Anal., **22**, 914–923 (1985)
18. S. Olver.: Moment-free numerical integration of highly oscillatory functions. IMA Journal of Numerical Analysis, **26**, 213–227 (2006)
19. J.-P. Richard.: Time-delay systems: an overview of some recent advances and open problems. Automatica, **39(10)**, 1667–1694 (2003)
20. A. Ruhe.: Algorithms for the nonlinear eigenvalue problem. SIAM J. Numer. Anal., **10**, 674–689, (1973)
21. H. Voss.: An Arnoldi method for nonlinear eigenvalue problems. BIT, **44**, 387 – 401 (2004)
22. 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)