# A Jacobi-Davidson method for two real parameter nonlinear eigenvalue problems arising from delay differential equations

Karl Meerbergen* Christian Schröder† Heinrich Voss‡

October 22, 2010

**Abstract**

The critical delays of a delay-differential equation can be computed by solving a nonlinear two-parameter eigenvalue problem. The solution of this two-parameter problem can be translated to solving a quadratic eigenvalue problem of squared dimension. We present a structure preserving QR-type method for solving such quadratic eigenvalue problem that only computes real valued critical delays, i.e. complex critical delays, which have no physical meaning, are discarded. For large scale problems, we propose new correction equations for a Newton type or Jacobi-Davidson style method, that also forces real valued critical delays. We present three different equations: one real valued equation using a direct linear system solver, one complex valued equation using a direct linear system solver, and one Jacobi-Davidson style correction equation which is suitable for an iterative linear system solver. We also present an alternative to the recurrence relation in the case the Jacobi-Davidson method cannot be continued with a real unconverged critical delay. We show numerical examples for large scale problems arising from PDEs

**Keywords:** Two-parameter eigenvalue problem, Jacobi-Davidson, nonlinear eigenvalue problem, delay differential equation, critical delay

## 1 Introduction

This paper considers the time-invariant linear delay-differential equation

$$M\dot{x}(t) + Ax(t) + Bx(t-\tau) = 0 \tag{1.1}$$

where, $M, A, B \in \mathbb{C}^{n\times n}$ are given system matrices and $\tau \geq 0$ is the delay. This system is (asymptotically) stable if, for every bounded initial condition, it holds that $x(t) \to 0$ as $t \to \infty$, and a necessary and sufficient condition is that the spectrum of the eigenvalue problem

$$\lambda Mu + Au + e^{-\tau\lambda}Bu = 0 \tag{1.2}$$

is contained in the open left half–plane (cf. [8], e.g.).

Due to the continuous dependence of eigenvalues on the parameter $\tau$, the stability behavior of (1.2) can only change as the delay $\tau$ varies if an eigenvalue of (1.2) crosses the imaginary

*Department of Computer Science, K.U.Leuven, Karl.Meerbergen@cs.kuleuven.be

†Institut für Mathematik, TU Berlin, schroed@math.tu-berlin.de; during the creation of this work he was substitute professor at TU Hamburg–Harburg

‡Institute of Numerical Simulation, Hamburg University of Technology, voss@tuhh.de

axis. Hence, our goal is to find real parameters $\tau \geq 0$ called critical (or switching or crossing) delays such that $\lambda = i\omega$, $\omega \in \mathbb{R}$ is an eigenvalue of (1.2), i.e. such that the parameter dependent delay eigenvalue problem (DEVP)

$$T(\omega, \tau)u := i\omega Mu + Au + e^{-i\omega\tau}Bu = 0 \tag{1.3}$$

has a real eigenvalue $\omega$.

Eq. (1.3) is a specific case of the more general problem appearing in the analysis of the stability of steady state solutions of a dynamical system: given a map $T : \mathbb{R} \times \mathbb{R} \to \mathbb{C}^{n\times n}$ the *two real parameter eigenvalue problem* is to find real values $\omega, \tau \in \mathbb{R}$ and a complex nonzero vector $u \in \mathbb{C}^n \setminus \{0\}$ such that

$$T(\omega, \tau)u = 0. \tag{1.4}$$

This problem is discussed in detail in [7] in the context of the determination of Hopf bifurcatons of linear systems of differential equations. Other examples arise from the stability analysis of hydrodynamic systems [4] and chemical reactors [9], where the parameter $\tau$ is the Reynolds number or the temperature or another physical parameter. An obvious approach is continuation or homotopy, i.e. to follow an eigenvalue curve $\lambda(\tau)$ or a sequence of invariant subspaces solving a sequence of eigenvalue problems (1.2) for slightly changing $\tau$. This is not only an inefficient process, but may also lead to wrong conclusions as was pointed out in [3].

Another approach to determine a critical delay consists of the transformation $\mu = e^{-i\omega\tau}$ in (1.3) and elimination of the eigenvalue $\omega$ using Kronecker products on the system matrices. This reduction from two to one parameter leads to the following quadratic eigenvalue problem in $\mu$ of order $n^2$,

$$\mu^2(B \otimes \bar{M})z + \mu(A \otimes \bar{M} + M \otimes \bar{A})z + (M \otimes \bar{B})z = 0 \tag{1.5}$$

called Kronecker eigenvalue problem (KEVP). Since we are only interested in real eigenvalues $\omega$, we are looking for $\mu$'s on the unit circle.

Kronecker products have been used in the literature for transforming a delay eigenvalue problem to a matrix pencil or a polynomial or a rational or a two parameter eigenvalue problem in order to determine critical delays [2, 5, 11, 12, 15, 17] for different classes of delay eigenvalue problems. Fassbender et al. [5] pointed out that the KEVP corresponding to (1.3) (even for a wider class of neutral time-delay systems) has a generalized palindromic structure which allows for structure preserving eigensolvers. In Chapter 2 we derive a real matrix pencil of dimension $2n^2$ such that the eigenvalues of (1.5) on the unit circle and the corresponding eigenvectors can be reconstructed from the real eigenvalues and corresponding eigenvectors of this pencil. Since the real eigenvalues of a real pencil can be determined in a stable way by the real QZ algorithm, the eigenvalues of (1.5) on the unit circle, and therefore the critical delays can be identified in a reliable way.

Unfortunately the complexity of this approach is $\mathcal{O}(n^6)$ such that it only applies to relatively small dimensions. To deal with medium sized to large scale problems, we combine the structure preserving eigensolver with an iterative projection method of Jacobi-Davidson type. Let $V \in \mathbb{C}^{n\times k}$, $k \ll n$ be an orthonormal basis of an ansatz space, and let $(\omega, \tau, y)$, $\omega, \tau \in \mathbb{R}$ be a solution of the projected $k$ dimensional eigenvalue problem

$$V^H T(\omega, \tau)Vy = 0 \tag{1.6}$$

which has the same structure as the original problem but can be solved by the structure preserving method of Section 2. If the resulting approximate solution $(\omega, \tau, Vy)$ of (1.3) does not satisfy our accuracy requirements then $V$ is expanded following an analogous goal as in the Jacobi–Davidson method for nonlinear eigenvalue problems in [1, 20]: (an approximation to) the Newton direction of (1.3) has to be contained in the new search space span$[V, v]$.

Since we allow real $\omega$ and $\tau$ in the Newton step, the approach in [1, 20] does not apply directly. In Section 3 we construct a modified correction equation, and we discuss its iterative solution by a preconditioned Krylov solver. It is noteworthy that in the whole procedure only vectors of dimension $n$ are involved.

The paper is organized as follows. In Section 2 we present the Kronecker eigenvalue problem related to the delay eigenvalue problem and we develop a structure preserving method. Section 3 develops a Jacobi-Davidson method for solving the large scale delay eigenvalue problems, and we discuss a practical algorithm for determining critical delays, including finding starting values for $\mu$ and $\omega$. Section 4 illustrates the method for delay equations arising from discretized PDEs. They demonstrate that the method is very robust with respect to inexact solves of the correction equation, coarse preconditioners, and the starting vector. We formulate the main conclusions in Section 5.

# 2 A structure preserving method for the Kronecker eigenvalue problem

In this section, we derive a KEVP for the DEVP by eleminating $\omega$. We then develop a structure preserving method for its solution.

## 2.1 The Kronecker eigenvalue problem

Recall the DEVP

$$(i\omega M + A + e^{-i\omega\tau}B)u = 0 \tag{2.1}$$

where $M, A, B \in \mathbb{C}^{n\times n}$. We are interested in solutions $(\omega, \tau, u) \in \mathbb{R} \times \mathbb{R} \times \mathbb{C}^n$.

Introducing the parameter $\mu = e^{-i\omega\tau}$ translates the problem to the two-parameter problem

$$i\omega Mu + Au + \mu Bu = 0. \tag{2.2}$$

Note that $\mu$ lies on the unit circle, thus $\bar{\mu} = \mu^{-1}$. Hence the complex conjugate equation of (2.2) reads

$$-i\omega \bar{M}\bar{u} + \bar{A}\bar{u} + \mu^{-1}\bar{B}\bar{u} = 0. \tag{2.3}$$

Using (2.2) and (2.3) we arrive at

$$\begin{aligned} 0 &= -(i\omega Mu) \otimes \bar{M}\bar{u} - Mu \otimes (-i\omega \bar{M}\bar{u}) \\ &= (Au + \mu Bu) \otimes \bar{M}\bar{u} + Mu \otimes (\bar{A}\bar{u} + \mu^{-1}\bar{B}\bar{u}) \\ &= ((A + \mu B) \otimes \bar{M} + M \otimes (\bar{A} + \mu^{-1}\bar{B}))(u \otimes \bar{u}). \end{aligned} \tag{2.4}$$

The last equation is a rational eigenvalue problem. Expanding and multiplication by $\mu$ yields the quadratic eigenvalue problem

$$\mu^2(B \otimes \bar{M})z + \mu(A \otimes \bar{M} + M \otimes \bar{A})z + (M \otimes \bar{B})z = 0. \tag{2.5}$$

We have thus shown that if $(\omega, \tau, u)$ is a solution of (2.1) and $\mu = e^{-i\omega\tau}$ is on the unit circle, then $(\mu, u \otimes \bar{u})$ is an eigenpair of (2.5). This is a standard result, see e.g. [12] or [5]. The converse is also true:

**Theorem 1** *Let $M, A, B \in \mathbb{C}^{n,n}$ with $M$ nonsingular. Then any eigenvector $z \in \mathbb{C}^{n^2}$ of (2.5) corresponding to a simple eigenvalue $\mu \in \mathbb{C}$ can be written as $z = \alpha u_1 \otimes u_2$ for some vectors $u_1, u_2 \in \mathbb{C}^n$ and some $\alpha \in \mathbb{C}$.*

*Moreover, if $|\mu| = 1$ then $u_1 = \bar{u}_2 = u$ and there is an $\omega \in \mathbb{R}$ such that (2.2) holds.*

**Proof.** The proof follows a similar reasoning as the proof of Theorem 2.2 in [16].

Consider the Schur decompositions of $M^{-1}(A + \mu B)$ and $M^{-1}(\bar{A} + \mu^{-1}\bar{B})$, i.e.

$$\begin{aligned} A + \mu B &= MXSX^H \\ \bar{A} + \mu^{-1}\bar{B} &= \bar{M}YRY^H \end{aligned}$$

with $X^H X = I$, $Y^H Y = I$ and $S$ and $R$ upper triangular. Then (2.4) can be written as

$$(MX \otimes \bar{M}Y)(S \otimes I + I \otimes R)(X^H \otimes Y^H)z = 0$$

and, since $M$ is nonsingular, there is a $t = X^H \otimes Y^H z$ so that

$$(S \otimes I + I \otimes R)t = 0$$

with $t \neq 0$. This is only possible when $R$ and $-S$ have common eigenvalues, i.e. there are $\omega \in \mathbb{C}$ and nonzero $t_1$ and $t_2$ so that $St_1 + i\omega t_1 = 0$ and $Rt_2 - i\omega t_2 = 0$. Then $t_1 \otimes t_2$ is a vector in the nullspace of $S \otimes I + I \otimes R$. Because $\mu$ is simple, this nullspace is of dimension one, so $t = \alpha t_1 \otimes t_2$. Consequently, $z = \alpha X t_1 \otimes Y t_2 =: \alpha u_1 \otimes u_2$.

When $|\mu| = 1$ or alternatively, $\bar{\mu} = \mu^{-1}$, then $R$ and $Y$ can be chosen as $R = \bar{S}$, $Y = \bar{X}$. As a result, $St_1 + i\omega t_1 = 0$ and $S\bar{t}_2 + i\bar{\omega}\bar{t}_2 = 0$. If $\mu$ is a simple eigenvalue, i.e. $S \otimes I + I \otimes R$ has a nullspace of dimension one, we must have that $t_2 = \alpha \bar{t}_1$ and thus $\omega = \bar{\omega}$. It follows that $\omega$ must be real and we can choose $u_1 = \bar{u}_2 = u = Xt_1$. □

We have thus transformed the problem of finding solutions of (2.2) to the problem of finding eigenvalues of modulus one of (2.5). This can be done by the structure preserving method presented in Section 2.2.

Once an eigenpair $(\mu, z)$ of (2.5) is known, $u$ can be recovered from $z = \alpha u \otimes \bar{u}$. One possibility is to compute the dominant singular vector of the $n \times n$ matrix $Z$ with $z = \mathrm{vec}(Z)$. Alternatively, one could chose $u$ as a column of $Z$ scaled such that $u$ has norm one. Subsequently $\omega$ can be obtained by projecting (2.2), i.e.

$$\omega = \frac{-\mathrm{Im}\big((Mu)^H(Au + \mu Bu)\big)}{\|Mu\|_2^2}. \tag{2.6}$$

Finally, $\tau$ may be computed from $\mu$ and $\omega$ as $\tau = -\mathrm{Im}(\ln(\mu))/\omega$.

## 2.2 A structure preserving eigenvalue solver

We need to compute all eigenpairs $(\mu, z)$ of (2.5) with $|\mu| = 1$. For small $n$, it is feasible to compute all eigenvalues of (2.5) and then sort out those not on the unit circle. An obstacle occurs however: with rounding errors it is difficult to decide whether a computed eigenvalue is on or close to the unit circle. We will exploit the structure in (2.5) to circumvent this difficulty. The following is essentially the method from [5], but specialized to quadratic problems with symmetric permutation $P$. This simplifies the method considerably, because the factorization (2.9) can be written down explicitly without computations.

Recall that the Kronecker product does, in general, not commute, i.e. $A \otimes B \neq B \otimes A$. However there is a symmetric permutation matrix $P$ such that $A \otimes B = P(B \otimes A)P$. The permutation vector $p$, corresponding to $P$, is obtained by `p=reshape(reshape(1:n^2,n,n)',1,n^2)` in MATLAB. Note that using this permutation, (2.5) is of the form

$$(\mu^2 A_2 + \mu A_1 + A_0)z = 0, \quad \text{where } A_2 = P\bar{A}_0 P,\ A_1 = P\bar{A}_1 P, \text{ where } P = P^{-1} = P^T \in \mathbb{R}^{n^2 \times n^2}. \tag{2.7}$$

Such problems are called PCP-palindromic eigenvalue problems. Their solution is discussed in [5].

The method can be divided into three steps. First, the quadratic eigenvalue problem is reformulated as

$$\left(\mu \begin{bmatrix} 0 & P \\ P & 0 \end{bmatrix} \begin{bmatrix} \bar{A}_1 - \bar{A}_2 & \bar{A}_0 \\ \bar{A}_0 & \bar{A}_0 \end{bmatrix} \begin{bmatrix} 0 & P \\ P & 0 \end{bmatrix} + \begin{bmatrix} A_1 - A_2 & A_0 \\ A_0 & A_0 \end{bmatrix}\right) \begin{bmatrix} \mu z \\ z \end{bmatrix} = 0, \tag{2.8}$$

which is a linear PCP-palindromic problem, because with $P$ also $\left[\begin{smallmatrix} & P \\ P & \end{smallmatrix}\right]$ is a real symmetric permutation.

Second, using the factorization

$$\begin{bmatrix} 0 & P \\ P & 0 \end{bmatrix} = UU^T, \quad \text{where} \quad U = \frac{1}{\sqrt{2}} \begin{bmatrix} I & iP \\ P & -iI \end{bmatrix} \tag{2.9}$$

we define

$$C := U^H \begin{bmatrix} A_1 - A_2 & A_0 \\ A_0 & A_0 \end{bmatrix} \bar{U}. \tag{2.10}$$

Since $U$ is unitary, i.e. $U^H U = I = U^T \bar{U}$, the pencil in (2.8) is equivalent to

$$\begin{aligned} &U^H \left(\mu \begin{bmatrix} 0 & P \\ P & 0 \end{bmatrix} \begin{bmatrix} \bar{A}_1 - \bar{A}_2 & \bar{A}_0 \\ \bar{A}_0 & \bar{A}_0 \end{bmatrix} \begin{bmatrix} 0 & P \\ P & 0 \end{bmatrix} + \begin{bmatrix} A_1 - A_2 & A_0 \\ A_0 & A_0 \end{bmatrix}\right) \bar{U} \\ =&\mu U^T \begin{bmatrix} \bar{A}_1 - \bar{A}_2 & \bar{A}_0 \\ \bar{A}_0 & \bar{A}_0 \end{bmatrix} U + U^H \begin{bmatrix} A_1 - A_2 & A_0 \\ A_0 & A_0 \end{bmatrix} \bar{U} \\ =&\mu \bar{C} + C. \end{aligned}$$

Third, consider an eigenpair $(\theta, x)$ of the real generalized eigenvalue problem

$$\mathrm{Re}(C)x = \theta \mathrm{Im}(C)x \tag{2.11}$$

or, equivalently,

$$Cx + \frac{i+\theta}{i-\theta}\bar{C}x = 0. \tag{2.12}$$

Note that $\mu := (i+\theta)/(i-\theta)$ is on the unit circle if and only if $\theta$ is real or $\theta = \infty$ (the latter resulting in $\mu = 1$). Note that real simple eigenvalues of real pencils can be stably computed by e.g. the real QZ algorithm.

Eigenvectors of (2.8) and (2.11) are related by

$$\begin{bmatrix} \mu z \\ z \end{bmatrix} = \bar{U}x = \frac{1}{\sqrt{2}} \begin{bmatrix} I & -iP \\ P & iI \end{bmatrix} x. \tag{2.13}$$

Hence, (a multiple of) an eigenvector of the original quadratic problem (2.7) respectively (2.5) is given by the lower (as well as the upper) half of $\bar{U}x$:

$$z = \frac{1}{\sqrt{2}} \begin{bmatrix} I & -iP \end{bmatrix} x \text{ or } z = \frac{1}{\sqrt{2}} \begin{bmatrix} P & iI \end{bmatrix}. \tag{2.14}$$

Putting all equations together, our algorithm to solve small scale DEVPs reads as follows.

**Algorithm 1**

**Input:** $M, A, B \in \mathbb{C}^{n \times n}$

**Output:** solutions $(\omega_j, \tau_j, u_j)_{j=1,\dots}$ of $(i\omega M + A + e^{-i\omega\tau} B)u = 0$

1: $A_0 = M \otimes \bar{B}$, $A_1 = A \otimes \bar{M} + M \otimes \bar{A}$
2: Construct the permutation $P$
3: $C = \frac{1}{2} \begin{bmatrix} I & iP \\ P & -iI \end{bmatrix}^H \begin{bmatrix} A_1 - P\overline{A_0}P & A_0 \\ A_0 & A_0 \end{bmatrix} \overline{\begin{bmatrix} I & iP \\ P & -iI \end{bmatrix}}$
4: Compute all eigenpairs $(\theta_j, x_j)$ of $\mathrm{Re}(C)x = \theta \mathrm{Im}(C)x$ where $\theta_j$ is real
5: **for** j=1,... **do**
6: $\quad \mu_j = \frac{i+\theta_j}{i-\theta_j}$
7: $\quad z_j = [I \;\; -iP]x_j$
8: $\quad$ Compute $u_j$ as dominant singular vector of $\mathrm{mat}(z_j)$
9: $\quad \omega_j = \frac{-\mathrm{Im}((Mu_j)^H (Au_j + \mu_j Bu_j))}{\|Mu_j\|_2^2}$
10: $\quad \tau_j = \frac{-\mathrm{Im}(\ln(\mu_j))}{\omega_j}$
11: **end for**

The dominant computational step in Algorithm 1 is step 4 with a complexity of $\mathcal{O}(n^6)$.

# 3 A Jacobi-Davidson method for the two real parameter eigenvalue problem

Due to its high numerical complexity the structure preserving algorithm in Section 2.2 is only applicable to problems of relatively small dimension. For medium sized to large scale problems we combine it with a subspace method of Jacobi–Davidson type.

The Jacobi–Davidson method is an iterative projection method which was introduced by Sleijpen and van der Vorst [21] for linear eigenvalue problems and generalized to one parameter nonlinear ones [1, 20]. Here the search space is expanded in a way that the Newton iterate at the current approximate eigenpair is contained in the expanded subspace, and it was shown to be very robust with respect to inexact Newton iterates [21, 23] which makes it particularly qualified for large–scale problems. In this section we develop a Jacobi–Davidson type method for the nonlinear two-parameter eigenvalue problem (1.3) with the particularity that the parameters are required to be real.

Algorithm 2 contains a template of the Jacobi–Davidson method.

**Algorithm 2**

**Input:** $T : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{C}^{n \times n}$, its partial derivatives, and tolerance $\eta > 0$

**Output:** solutions $(\omega_j, \tau_j, u_j)_{j=1,\dots} \in \mathbb{R} \times \mathbb{R} \times \mathbb{C}^n$ of $T(\omega, \tau)u = 0$

construct a suitable basis $V \in \mathbb{C}^{n \times k_0}$, $V^H V = I$ of the initial search space
**for** $k = k_0, k_0 + 1, \dots$ **do**

3: solve the projected eigenvalue problem

$$V^H T(\omega,\tau) V z = 0. \tag{3.1}$$

4: extract eigentriples $(\hat{\omega}_i, \hat{\tau}_i, \hat{z}_i)$ with real $\hat{\omega}_i$ and $\hat{\tau}_i$.
5: compute the associated Ritz vectors $\hat{u}_i = V\hat{z}_i$
6: check for convergence: $\|T(\hat{\omega}_i, \hat{\tau}_i)\hat{u}_i\|_2 \leq \eta$.
7: (append $\overline{\hat{u}_i}$ to $V$ if $(\hat{\omega}_i, \hat{\tau}_i, \hat{u}_i)$ has converged and $\overline{\hat{u}_i} \notin \operatorname{span}(V)$)
8: stop if enough eigentriples have converged
9: select an approximate eigentriple to continue the JD method
10: reduce search space if necessary
11: determine expansion direction $c$
12: expand search space $V = \operatorname{orth}([V, c])$
13: **end for**

Some comments on the respective steps are in order.

1: If approximations of eigenvectors are known a priori, they can be used as initial search space. Otherwise one or a few random vectors work fine.

3:–4: The method to solve the projected problem depends on the individual two parameter eigenvalue problem at hand. In case of the delay eigenvalue problem (1.3) we have to solve

$$i\omega\widehat{M}z + \widehat{A}z + e^{-i\tau\omega}\widehat{B}z = 0 \tag{3.2}$$

with $\widehat{M} = V^H M V$, $\widehat{A} = V^H A V$, and $\widehat{B} = V^H B V$. Problem (3.2) is solved by the structure preserving Algorithm 1.

6: The stopping criterion is based on the residual $r = T(\hat{\omega}, \hat{\tau})\hat{u}$ as a measure of the backward error of the eigentriple. An eigentriple is considered accurate when $\|r\| \leq \eta$ for a prescribed tolerance $\eta$.

7: This step applies to DEVPs with real matrices $M, A, B$ only. If $(\omega, \tau, u)$ is an eigentriple then the same holds true for $(-\omega, \tau, \bar{u})$. Hence, for every converged eigentriple we append $\bar{u}$ to $\operatorname{span}(V)$ to obtain $(-\omega, \tau, \bar{u})$ immediately. Otherwise several JD iterations have to be performed to obtain $(-\omega, \tau, \bar{u})$ also.

9: The projected problem (3.1) often has more than one real eigenpair $(\omega, \tau)$. We choose the eigentriple which has not yet converged with the minimal residual. If (3.2) does not produce any eigentriples with which we can continue the method, we use an alternative selection as described in Subsection 3.4.

10: As the dimension of the search space grows, the cost of solving the projected problem (3.1) can become prohibitively large. In the case of the DEVP the cost for the QZ algorithm in the structure preserving eigensolver of the projected eigenvalue problem swells like $\mathcal{O}(k^6)$. Hence, the search space has to be reduced whenever $k$ exceeds a certain amount. We continue the algorithm with an orthonormal basis of the space spanned by the already converged eigenvectors and additionally all or a selection of vectors corresponding to the not yet converged eigenvectors of the current projected problem (3.1) with the smallest residual norm.

11: The orthogonal correction for an eigenvector approximation by Newton's method is given by a linear system, and we only have to make sure that the corrections of $\omega$ and $\tau$ are real. However, for truly large problems this linear system can not be solved by a direct solver. For this case we have to provide a correction equation which can be solved efficiently by an iterative solver. This will be discussed in Subsections 3.1, 3.2 and 3.3.

12: Any orthogonalization procedure can be employed. We use classical Gram Schmidt with reorthogonalization.

In the following subsections we shall discuss the expansion of the search space and how to find an initial basis.

### 3.1 Correction equation for medium sized problems

Let $(\hat{\omega}, \hat{\tau}, \hat{u})$ be an approximate eigentriple of the DEVP (1.3). We are looking for a correction $(\hat{\omega}+\delta, \hat{\tau}+\varepsilon, \hat{u}+c)$ where $\delta, \varepsilon$ are real and $c$ is complex and is orthogonal to $\hat{u}$. This orthogonality constraint makes the direction of $c$ unique.

Applying a single step of Newtons method to the root finding problem

$$f(c, \delta, \varepsilon) = \begin{bmatrix} T(\hat{\omega} + \delta, \hat{\tau} + \varepsilon)(\hat{u} + c) \\ \hat{u}^H c \end{bmatrix} = 0 \tag{3.3}$$

leads to the *Newton correction equation*

$$\begin{bmatrix} T(\hat{\omega}, \hat{\tau}) & T_\omega(\hat{\omega}, \hat{\tau})\hat{u} & T_\tau(\hat{\omega}, \hat{\tau})\hat{u} \\ \hat{u}^H & 0 & 0 \end{bmatrix} \begin{bmatrix} c \\ \delta \\ \varepsilon \end{bmatrix} = \begin{bmatrix} -T(\hat{\omega}, \hat{\tau})\hat{u} \\ 0 \end{bmatrix}. \tag{3.4}$$

This is a linear system of $(n+1)$ complex equations in $n$ complex and 2 real unknowns.

To determine a solution of (3.4) with real components $\delta$ and $\varepsilon$, and to take advantage of existing software for solving linear systems we separate real and imaginary parts in (3.4) leading to the equivalent real linear system of dimension $2n+2$

$$\begin{bmatrix} \mathrm{Re}(\hat{T}) & -\mathrm{Im}(\hat{T}) & \mathrm{Re}(\hat{T}_\omega\hat{u}) & \mathrm{Re}(\hat{T}_\tau\hat{u}) \\ \mathrm{Im}(\hat{T}) & \mathrm{Re}(\hat{T}) & \mathrm{Im}(\hat{T}_\omega\hat{u}) & \mathrm{Im}(\hat{T}_\tau\hat{u}) \\ \mathrm{Re}(\hat{u}^H) & -\mathrm{Im}(\hat{u}^H) & 0 & 0 \\ \mathrm{Im}(\hat{u}^H) & \mathrm{Re}(\hat{u}^H) & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathrm{Re}(c) \\ \mathrm{Im}(c) \\ \delta \\ \varepsilon \end{bmatrix} = \begin{bmatrix} \mathrm{Re}(-\hat{T}\hat{u}) \\ \mathrm{Im}(-\hat{T}\hat{u}) \\ 0 \\ 0 \end{bmatrix}, \tag{3.5}$$

which we call the *real correction equation.* Here we used the notation $\hat{T} := T(\hat{\omega}, \hat{\tau})$, $\hat{T}_\omega := T_\omega(\hat{\omega}, \hat{\tau})$, and $\hat{T}_\tau := T_\tau(\hat{\omega}, \hat{\tau})$.

A different approach that guarantees the realness of $\delta$ and $\varepsilon$ is to complement system (3.4) by its conjugate complex counterpart, which we call the *double correction equation.* The name double comes from the fact that we join the correction equations for both $u$ and $\bar{u}$.

$$\begin{bmatrix} \hat{T} & 0 & \hat{T}_\omega\hat{u} & \hat{T}_\tau\hat{u} \\ 0 & \overline{\hat{T}} & \overline{\hat{T}_\omega\hat{u}} & \overline{\hat{T}_\tau\hat{u}} \\ \hat{u}^H & 0 & 0 & 0 \\ 0 & \overline{\hat{u}}^H & 0 & 0 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} = \begin{bmatrix} -\hat{T}\hat{u} \\ -\overline{\hat{T}\hat{u}} \\ 0 \\ 0 \end{bmatrix}. \tag{3.6}$$

In [18], it was argued that complex linear systems that respect the sparse structure are preferred to real valued linear system such as (3.5) that lose this structure. Indeed, (3.6) can be solved by a single factorization of $\hat{T}$, provided $\hat{T}$ is nonsingular.

To show that the components $d_3$ and $d_4$ of a solution of (3.6) are real and $d_1$ and $d_2$ are conjugate complex of each other we first show the following results.

**Lemma 1** *If the complex valued linear system*

$$\begin{bmatrix} \alpha_1 & \alpha_2 \\ \bar{\alpha}_1 & \bar{\alpha}_2 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} = \begin{bmatrix} \beta \\ \bar{\beta} \end{bmatrix} \tag{3.7}$$

*has a unique solution, then* $\xi_1$ *and* $\xi_2$ *are real.*

**Proof.** By linear combination of the rows of the complex system, we find the real system

$$\begin{bmatrix} \text{Re}(\alpha_1) & \text{Re}(\alpha_2) \\ \text{Im}(\alpha_1) & \text{Im}(\alpha_2) \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} = \begin{bmatrix} \text{Re}(\beta) \\ \text{Im}(\beta) \end{bmatrix}$$

which proves the lemma. □

**Theorem 2** *Let the linear system* (3.6) *be nonsingular. Then its solution is* $(c^T, \bar{c}^T, \delta, \epsilon)^T$ *where* $c$, $\delta$ *and* $\varepsilon$ *satisfy* (3.4) *and* $\delta$ *and* $\varepsilon$ *are real.*

**Proof.** We first prove that $d_3$ and $d_4$ are real and $d_2 = \bar{d}_1$. Since the linear system is nonsingular, Gaussian elimination (if needed with pivoting) on the first and third block rows and similarly the second and fourth block rows produces

$$\begin{bmatrix} I & 0 & h_\omega & h_\tau \\ 0 & I & \bar{h}_\omega & \bar{h}_\tau \\ 0 & 0 & \alpha_\omega & \alpha_\tau \\ 0 & 0 & \bar{\alpha}_\omega & \bar{\alpha}_\tau \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} = \begin{bmatrix} g \\ \bar{g} \\ \gamma \\ \bar{\gamma} \end{bmatrix}. \tag{3.8}$$

Then $d_3$ and $d_4$ are solved from the subsystem consisting of the third and fourth rows of (3.8). Following Lemma 1, $d_3$ and $d_4$ are real. Since $d_3$ and $d_4$ are real, it is easy to see from (3.8) that $d_2 = \bar{d}_1$.

It is also quite clear that the solution of (3.6) satisfies (3.4): the first and third rows on the one hand and the complex conjugate of the second and fourth rows of (3.6) on the other hand correspond to (3.4). □

Both systems of equations, (3.5) and (3.6) are called correction equations of the Jacobi–Davidson method. Expanding the search space by the solution $c$ of one of them we can expect quadratic convergence [21]. However, solving them directly becomes too expensive for truly large problems, and so they are only appropriate for medium sized problem.

## 3.2 Correction equation for large-scale problems

In this subsection we eliminate $d_3$ and $d_4$ from system (3.6) to obtain a form of the correction equations which is suitable for solving it by a preconditioned Krylov solver. A similar technique was used in [10] to derive a second-order correction equation for two-parameter linear eigenvalue problems.

With

$$\mathbb{T} = \begin{bmatrix} \hat{T} & 0 \\ 0 & \bar{\hat{T}} \end{bmatrix}, \quad K = \begin{bmatrix} \hat{T}_\omega \hat{u} & \hat{T}_\tau \hat{u} \\ \overline{\hat{T}_\omega \hat{u}} & \overline{\hat{T}_\tau \hat{u}} \end{bmatrix}, \quad U = \begin{bmatrix} \hat{u} & 0 \\ 0 & \bar{\hat{u}} \end{bmatrix}$$

and $r = \hat{T}\hat{u}$ equation (3.6) reads

$$\begin{bmatrix} \mathbb{T} & K \\ U^H & 0 \end{bmatrix} \begin{bmatrix} c \\ \bar{c} \\ \delta \\ \varepsilon \end{bmatrix} = \begin{bmatrix} -r \\ -\bar{r} \\ 0 \\ 0 \end{bmatrix}. \tag{3.9}$$

Assume that $U^H K$ has rank two (i.e. full rank). Then we deduce from (3.9) that

$$U^H \mathbb{T} \begin{bmatrix} c \\ \bar{c} \end{bmatrix} + U^H K \begin{bmatrix} \delta \\ \varepsilon \end{bmatrix} = -U^H \begin{bmatrix} r \\ \bar{r} \end{bmatrix}.$$

Solving this equation for $\delta$ and $\varepsilon$ and pluging them back into (3.9) we obtain

$$(\mathbb{T} - K(U^H K)^{-1} U^H \mathbb{T}) \begin{bmatrix} c \\ \bar{c} \end{bmatrix} = -(I - K(U^H K)^{-1} U^H) \begin{bmatrix} r \\ \bar{r} \end{bmatrix}. \tag{3.10}$$

Imposing that $\hat{u}^H c = 0$ we finally arrive at

$$(I - K(U^H K)^{-1} U^H) \mathbb{T} (I - UU^H) \begin{bmatrix} c \\ \bar{c} \end{bmatrix} = -(I - K(U^H K)^{-1} U^H) \begin{bmatrix} r \\ \bar{r} \end{bmatrix}. \tag{3.11}$$

which looks like the common correction equation of the Jacobi–Davidson method for nonlinear eigenvalue problems: the equation $\mathbb{T} \begin{bmatrix} c \\ \bar{c} \end{bmatrix} = - \begin{bmatrix} r \\ \bar{r} \end{bmatrix}$ is complemented by the oblique projector $I - K(U^H K)^{-1} U^H$ which provokes that the Newton direction is contained in $\operatorname{span}\{\hat{u}, c\}$. We call (3.11) the *JD correction equation.*

The same procedure could be used to obtain a correction equation from (3.5) which does not depend on $\delta$ and $\varepsilon$. However, it will be demonstrated in the next subsection that preconditioned Krylov solvers can be applied easily to (3.11). In particular, given a preconditioner $P$ of $\hat{T}$ one can easily implement the preconditioner $\operatorname{diag}\{P, \bar{P}\}$ for system (3.11) which does not make sense for the real system corresponding to (3.11) obtained from (3.5).

### 3.3 Preconditioning the correction equation

When an iterative solver is used for (3.11), a preconditioner is highly recommended. Let $P$ be a preconditioner for $\hat{T} = T(\hat{\omega}, \hat{\tau})$. Define

$$\mathbb{P} = \begin{bmatrix} P & 0 \\ 0 & \bar{P} \end{bmatrix}.$$

Then the projected matrix

$$(I - K(U^H K)^{-1} U^H) \mathbb{P} (I - UU^H) \tag{3.12}$$

is an approximation to $(I - K(U^H K)^{-1} U^H) \mathbb{T} (I - UU^H)$ which we use as preconditioner for (3.11). The inverse of (3.12) in the range of $(I - UU^H)$ is

$$(I - UU^H) \mathbb{P}^{-1} (I - K(U^H \mathbb{P}^{-1} K)^{-1} U^H \mathbb{P}^{-1}).$$

In every step of a left preconditioned Krylov solver for (3.11) we have to determine

$$\begin{bmatrix} t \\ \bar{t} \end{bmatrix} = [(I - UU^H)\mathbb{P}^{-1}(I - K(U^H\mathbb{P}^{-1}K)^{-1}U^H\mathbb{P}^{-1})](I - K(U^HK)^{-1}U^H)\mathbb{T}(I - UU^H) \begin{bmatrix} v \\ \bar{v} \end{bmatrix}$$

for some given $v$ with $\hat{u}^H v = 0$.

Simplifications produce the following equations.

$$\begin{aligned} \begin{bmatrix} t \\ \bar{t} \end{bmatrix} &= (I - UU^H)\mathbb{P}^{-1}(I - K(U^H\mathbb{P}^{-1}K)^{-1}U^H\mathbb{P}^{-1})\mathbb{T}(I - UU^H) \begin{bmatrix} v \\ \bar{v} \end{bmatrix} \\ \begin{bmatrix} t \\ \bar{t} \end{bmatrix} &= (I - UU^H)(I - \mathbb{P}^{-1}K(U^H\mathbb{P}^{-1}K)^{-1}U^H)\mathbb{P}^{-1}\mathbb{T}(I - UU^H) \begin{bmatrix} v \\ \bar{v} \end{bmatrix} \\ \begin{bmatrix} t \\ \bar{t} \end{bmatrix} &= (I - \mathbb{P}^{-1}K(U^H\mathbb{P}^{-1}K)^{-1}U^H)\mathbb{P}^{-1}\mathbb{T}(I - UU^H) \begin{bmatrix} v \\ \bar{v} \end{bmatrix} \qquad (3.13) \end{aligned}$$

The vector $t$ can be computed very efficiently taking advantage of the fact $\hat{u}^H t = 0$. (Of course, $\bar{t}$ is not computed explicitly.) Using the structure of $U$, $\mathbb{T}$ and $\mathbb{P}$, (3.13) becomes

$$\begin{bmatrix} t \\ \bar{t} \end{bmatrix} = (I - \mathbb{P}^{-1}K(U^H\mathbb{P}^{-1}K)^{-1}U^H) \begin{bmatrix} P^{-1}\hat{T}(I - \hat{u}\hat{u}^H)v \\ \bar{P}^{-1}\bar{\hat{T}}(I - \overline{\hat{u}\hat{u}^H})\bar{v} \end{bmatrix}$$

or since $(I - \hat{u}\hat{u}^H)v = v$

$$\begin{aligned} t &= P^{-1}Tv + P^{-1}[\hat{T}_\omega \hat{u}, \hat{T}_\tau \hat{u}]a \qquad (3.14) \\ \bar{t} &= \bar{P}^{-1}\bar{\hat{T}}v + \bar{P}^{-1}[\bar{\hat{T}}_\omega \bar{u}, \bar{\hat{T}}_\tau \bar{u}]a \qquad (3.15) \end{aligned}$$

where $a$ is chosen so that $\hat{U}^H t = 0$.

Multiplying by $\hat{u}^H$ and $\overline{\hat{u}}$, respectively we obtain a linear system like (3.7), and it follows from Lemma 1 that $a$ must be real.

The preconditioned right-hand side for the iterative solver is

$$\begin{aligned} \begin{bmatrix} b \\ \bar{b} \end{bmatrix} &= -[(I - UU^H)\mathbb{P}^{-1}(I - K(U^H\mathbb{P}^{-1}K)^{-1}U^H\mathbb{P}^{-1})](I - K(U^HK)^{-1}U^H) \begin{bmatrix} r \\ \bar{r} \end{bmatrix} \\ &= -(I - UU^H)\mathbb{P}^{-1}(I - K(U^H\mathbb{P}^{-1}K)^{-1}U^H\mathbb{P}^{-1}) \begin{bmatrix} r \\ \bar{r} \end{bmatrix} \\ &= -(I - \mathbb{P}^{-1}K(U^H\mathbb{P}^{-1}K)^{-1}U^H)\mathbb{P}^{-1} \begin{bmatrix} r \\ \bar{r} \end{bmatrix} \end{aligned}$$

which can again be written as

$$b = P^{-1}r - P^{-1}[\hat{T}_\omega \hat{u}, \hat{T}_\tau \hat{u}]a_r \qquad (3.16)$$

with $a_r \in \mathbb{C}^2$. With the same reasoning as before, using $\hat{u}^H b = 0$ and Lemma 1, it can be seen that $a_r$ must be real. (Of course, $\bar{b}$ is not computed explicitly.)

Equations (3.14) and (3.16) demonstrate how the preconditioned Krylov solver for (3.11) can be implemented efficiently. The matrix $D := P^{-1}[\hat{T}_\omega \hat{u}, \hat{T}_\tau \hat{u}]$ is independent of $v$ and $r$ and can be pre-calculated. Then the vector $t$ is obtained from

$$t = P^{-1}\hat{T}v + Da \quad \text{with} \begin{bmatrix} \mathrm{Re}(\hat{u}^H D) \\ \mathrm{Im}(\hat{u}^H D) \end{bmatrix} a = - \begin{bmatrix} \mathrm{Re}(\hat{u}^H P^{-1}\hat{T}v) \\ \mathrm{Im}(\hat{u}^H P^{-1}\hat{T}v) \end{bmatrix},$$

and $b$ is obtained similarly. Hence, each iteration requires only one matrix-vector product $\hat{T}v$ and one solve with the preconditioner, and the only additional cost taking into account the projectors in the preconditioning are two matrix-vector products and three solves with the preconditioner $P$ to determine $D$ and to initialize the Krylov solver. This favorable property that the inclusion of the projectors into the preconditioner comes nearly for free was already pointed out by Sleijpen and van der Vorst in [21].

### 3.4 Alternative selection

In Algorithm 2 in step 9 we have to select an approximate eigenvector $\hat{u}$ from the search space and appropriate real $\hat{\omega}, \hat{\tau}$. Usually we take an unconverged Ritz triple of (3.1). When such a Ritz triple does not exist, we have a problem. This is frequently the case in the very early stages of the algorithm, but may also occur in the middle of the JD iteration. In these cases an alternative selection strategy has to be employed, which depends on the two parameter problem to solve.

In our approach for the DEVP, we try and find a valid Ritz value near a fixed point on the unit circle, $\sigma$, say. We therefore fix $\hat{\omega} = 0$ and $\hat{\mu} = \sigma$. The aim is thus to seek the valid eigentriple with $\omega$ nearest zero and $\mu$ nearest $\sigma$. The Ritz vector $\hat{u}$ is selected as $\hat{u} = Vz$ so that $\|z\|_2 = 1$ and the norm $\|(A + \sigma B)\hat{u}\|_2$ of the residual is minimum. Such $\hat{u}$ is called a refined Ritz vector [14]. We call this the *alternative selection*. The Jacobi-Davidson method is continued with this selection.

The strategy is implemented by replacing step 9 of Algorithm 2 by

1: select an approximate eigenpair to continue the JD method.
2: **if** no such eigentriple exists **then**
3: Select $\hat{\omega} = 0$ and $\hat{\mu} = \sigma$
4: Compute $\hat{u} = V\hat{z}$, with $\|\hat{z}\|_2 = 1$ so that $\|(A + \sigma B)\hat{u}\|_2$ is minimal.
5: **else**
6: select the unconverged Ritz triple with smallest residual norm to continue the JD method
7: **end if**

One interpretation for this approach is the following one. When we solve the Newton correction equation (3.4) with $\hat{\omega} = 0$ and (formally) $\hat{\tau}$ so that $e^{-i\hat{\tau}\hat{\omega}} = \sigma$, we have that $T_\omega = iM$ and $T_\tau = -\sigma\omega B$, which is zero in this case. As a result, we expand the subspace with the vector

$$(A + \sigma B)^{-1} M\hat{u}$$

This can be interpreted as an inverse iteration step for the linear generalized eigenvalue problem

$$(A + \sigma B)\hat{u} + i\omega M\hat{u} = 0$$

suggesting we are looking for the $\hat{\omega}$'s nearest zero. When an iterative linear system solver is used, the JD correction equation (3.11) is equivalent to an inexact inverse iteration step, see, e.g., [21].

We want to warn the reader that for $\hat{\omega} = 0$, the correction equation is singular, because in this case $\hat{T}_\tau = 0$. We noticed similar problems for small absolute values of $\hat{\omega}$. In this case, (3.6) can be reduced to

$$\begin{bmatrix} \hat{T} & \hat{T}_\omega \hat{u} \\ \hat{u}^H & 0 \end{bmatrix} \begin{bmatrix} c \\ \delta \end{bmatrix} = \begin{bmatrix} -r \\ 0 \end{bmatrix}. \tag{3.17}$$

Note that the value of $\delta$ is not necessarily real.

# 4 Numerical examples

## 4.1 Example 1

The first example is taken from [13]:

$$-\frac{\partial}{\partial t}x(\xi,t)+\frac{\partial^2}{\partial \xi^2}x(\xi,t)+a(\xi)x(\xi,t)+b(\xi)x(\pi-\xi,t-\tau)=0 \quad \text{for } \xi\in(0,\pi),\ t>0 \tag{4.1}$$

with boundary conditions: $\frac{\partial}{\partial\xi}x(0,t) = 0 = \frac{\partial}{\partial\xi}x(\pi,t)$. Discretized in space with central differences (i.e. $x(t)=[x_1(t),\ldots,x_n(t)]^T$ with $x_i(t)=x((i-1)h,t)$ and $h=\pi/(n-1)$) yields

$$I\dot{x}(t)+\left(\frac{1}{h^2}\begin{bmatrix}1 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 1\end{bmatrix}-\begin{bmatrix}a(0) & & & & \\ & a(h) & & & \\ & & \ddots & & \\ & & & a(\pi-h) & \\ & & & & a(\pi)\end{bmatrix}\right)x(t)$$

$$-\begin{bmatrix} & & & & b(0) \\ & & & b(h) & \\ & & \iddots & & \\ & b(\pi-h) & & & \\ b(\pi) & & & & \end{bmatrix}x(t-\tau)=0$$

We used $n=500$ and $a(\xi)=-2\sin(\xi)$, $b(\xi)=2\sin(\xi)+1$. (In [13] $b(\xi)=2\sin(\xi)$ is used, but then the problem has a double solution.)

We first used Algorithm 2, where the correction equation (3.6) was solved with a direct sparse solver, using MATLAB's backslash. We accepted an eigentriple as converged if its residual norm was less than $10^{-10}$. Table 1 shows the Ritz values $\omega$ and $\tau$ for the different iterations with their residual norms. In the last column we indicated by 'Ritz sel.' that the associated Ritz pair was selected for solving the correction equation, by 'alt. sel.' that no eigenvalues of the Kronecker eigenvalue problem were found on the unit circle and the search space was expanded by the method described in Subsection 3.4, and we marked a row by 'conv.' if a Ritz triple $(\omega,\tau,u)$ had converged. Note that we used $\sigma=1$ in the alternative selection for all examples. In this case we complemented the search space by the complex conjugate $\bar{u}$ of the Ritz vector $u$ such that $(-\omega,\tau,\bar{u})$ was found immediately.

Starting with a random vector $x=\texttt{randn}(500,1)+i\cdot\texttt{randn}(500,1)$ it is not surprising that the projected problem has no real eigenpair $(\omega,\tau)$. It took two alternative expansion steps to obtain a projected problem with a real eigenpair. Three expansions of the search space with Newton steps produced a residual $3.63\ 10^{-13}$ such that the first eigentriple has converged. After complementing the search space by $\bar{u}$ the next eigentriple was found, but there was no further real eigenpair $(\omega,\tau)$ corresponding to a non-converged eigentriple. After one alternative expansion of the search space two more real eigenpairs of the projected problem were found. Two further Newton expansions made one eigenpair converge, and complementing the search space again by $\bar{u}$ yielded the missing eigentriple. The development of the residuals indicate the quadratic convergence.

| Iteration | $\hat{\omega}$ | $\hat{\tau}$ | residual norm | remark |
|---|---|---|---|---|
| 1 | 0 | | $6.38 \cdot 10^{-1}$ | alt. sel. |
| 2 | 0 | | $2.56 \cdot 10^{-4}$ | alt. sel. |
| 3 | 0.604560 | −0.548913 | $1.78 \cdot 10^{-4}$ | Ritz sel. |
| | −0.604560 | −0.548913 | $1.81 \cdot 10^{-4}$ | |
| 4 | 1.761073 | −0.520579 | $6.62 \cdot 10^{-5}$ | Ritz sel. |
| | −1.761073 | −0.520579 | $1.17 \cdot 10^{-4}$ | |
| 5 | 1.785539 | −0.533075 | $1.33 \cdot 10^{-7}$ | Ritz sel. |
| | −1.785539 | −0.533075 | $6.87 \cdot 10^{-6}$ | |
| 6 | 1.785556 | −0.533055 | $3.63 \cdot 10^{-13}$ | conv. |
| | −1.785556 | −0.533055 | $1.85 \cdot 10^{-5}$ | compl. by $\bar{u}$ |
| 7 | −1.785556 | −0.533055 | $4.73 \cdot 10^{-13}$ | |
| | 1.785556 | −0.533055 | $1.87 \cdot 10^{-13}$ | conv. |
| | 0 | | $2.56 \cdot 10^{-5}$ | alt. sel. |
| 8 | 1.785556 | −0.533055 | $1.68 \cdot 10^{-13}$ | |
| | −1.785556 | −0.533055 | $3.91 \cdot 10^{-13}$ | |
| | 0.101889 | 30.291077 | $9.94 \cdot 10^{-6}$ | Ritz sel. |
| | −0.101889 | 30.291077 | $1.09 \cdot 10^{-5}$ | |
| 9 | 1.785556 | −0.533055 | $1.84 \cdot 10^{-13}$ | |
| | −1.785556 | −0.533055 | $3.68 \cdot 10^{-13}$ | |
| | 0.119263 | 25.799283 | $1.04 \cdot 10^{-9}$ | Ritz sel. |
| | −0.119263 | 25.799283 | $2.76 \cdot 10^{-7}$ | |
| 10 | 1.785556 | −0.533055 | $1.38 \cdot 10^{-13}$ | |
| | −1.785556 | −0.533055 | $3.62 \cdot 10^{-13}$ | |
| | 0.119263 | 25.799285 | $2.81 \cdot 10^{-16}$ | conv. |
| | −0.119263 | 25.799285 | $2.81 \cdot 10^{-7}$ | compl. by $\bar{u}$ |
| 11 | 1.785556 | −0.533055 | $1.00 \cdot 10^{-13}$ | |
| | −1.785556 | −0.533055 | $2.61 \cdot 10^{-13}$ | |
| | 0.119263 | 25.799285 | $3.34 \cdot 10^{-16}$ | |
| | −0.119263 | 25.799285 | $5.13 \cdot 10^{-16}$ | conv. |

Table 1: Results for Example 1 using a direct linear system solver

We then used Algorithm 2, where the correction equation (3.11) was solved with GMRES [19]. It turned out that the method is extremely robust with respect to inexact solves of the correction equation. As preconditioner we used an incomplete Cholesky factorization of the real matrix $A$ with drop tolerance $10^{-2}$, but incomplete LU factorizations of $T(\omega, \tau)$ for various choices of real $\omega$ and $\tau$ worked similarly well. The GMRES method was terminated after at most 5 steps or if the initial residual was reduced by at least $10^{-1}$.

Columns two, three and four of Table 2 show the Ritz values for the different iterations with their residual norms. Column 5 contains the number of iteration steps of GMRES which were required to reduce the residual of the correction equation by $10^{-1}$, and Column 6 contains remarks similar to Table 1.

### 4.2 Example 2

Consider the equation

$$u_t - \nabla((1 + x^2 + y^2)\nabla u) - \alpha(1 + xy)u(t - \tau) = 0 \tag{4.2}$$

with spatial variables $x$ and $y$ on $\Omega = (0, 1) \times (0, 1)$ with Dirichlet boundary condition $u = 0$ on $\partial\Omega$. We used COMSOL [1] for a discretization with finite elements using piecewise quadratic ansatz functions on a triangular grid which led to an eigenvalue problem of the form (1.2) of dimension $n = 104257$.

We solved the problem for $\alpha = 23$ which is close to a delay-independent stable system (actually for $\alpha = 22.6$ the system is stable for all non-negative delay values $\tau$), and therefore the domain of attraction of Newton's method should be quite small. We preconditioned with `P=luinc(A,1e-3)`, and we terminated the GMRES method for solving the correction equation after at most 10 iteration steps or if the residual was reduced at least by $10^{-1}$ and we started Algorithm 2 by a complex random vector. Figure 1 shows the convergence history.

Five alternative expansion steps (marked by triangles) are required until a pair of eigentriples is detected for the projected problem. Thereafter the method needs 6 Jacobi–Davidson steps until the residual for one of the solutions $(\omega, \tau, u)$ does not exceed $10^{-10}$ (the residuals are marked by bullets whereas the residuals of the second solution of the projected problems are marked by stars). Appending $\bar{u}$ to the search space the residual of the second solution is also reduced to less than $10^{-10}$. The computations were performed on a Pentium (R) D computer with 3.2 GHz and 3.5 GB RAM under MATLAB R2008b. 40.5 seconds were needed for computing both Ritz triples and additionally 13.4 seconds to determine the preconditioner.

For $\alpha = 89$ the system has four pairs of eigentriples. We solved it using the same preconditioner and the same parameters for the GMRES solver of the correction equation. We restarted the method if the dimension of the search space exceeded $k = 12$ (recall the complexity $\mathcal{O}(k^6)$ of the structure preserving solver of the projected problems), and we retained the Ritz vectors found so far in the search space after restart [22, 6]. 157.6 seconds were needed to compute all 8 eigentriples. The convergence history is shown in Figure 2. Five restarts were necessary after steps 12, 16, 21, 25 and 28 but this loss of information does not seem to slow down the convergence.

Two alternative expansion steps are needed to find the first eigentripel (marked with black triangles). After 11 Jacobi–Davidson steps the first eigentriple $(\omega, \tau, u)$ has converged (black bullets). The residual of the approximation to $(-\omega, \tau, \bar{u})$ at that time is nearly untouched

[1] `http://www.comsol.com`

| Iteration | $\hat{\omega}$ | $\hat{\tau}$ | residual norm | # it.step. | remark |
|---|---|---|---|---|---|
| 1 | 0 | | $6.78 \cdot 10^{-1}$ | | alt. sel. |
| 2 | 0 | | $3.91 \cdot 10^{-4}$ | | alt. sel. |
| 3 | 0.898411 | −0.503098 | $2.42 \cdot 10^{-4}$ | 2 | Ritz sel. |
| | −0.898411 | −0.503098 | $2.48 \cdot 10^{-4}$ | | |
| 4 | 1.784253 | −0.532967 | $1.19 \cdot 10^{-5}$ | 3 | Ritz sel. |
| | −1.784253 | −0.532967 | $4.87 \cdot 10^{-5}$ | | |
| 5 | 1.785717 | −0.533051 | $9.96 \cdot 10^{-8}$ | 3 | Ritz sel. |
| | −1.785717 | −0.533051 | $4.30 \cdot 10^{-5}$ | | |
| 6 | 1.785559 | −0.533054 | $4.08 \cdot 10^{-9}$ | 3 | Ritz sel. |
| | −1.785559 | −0.533054 | $8.90 \cdot 10^{-5}$ | | |
| 7 | 1.785556 | −0.533055 | $3.59 \cdot 10^{-11}$ | | conv. |
| | −1.785556 | −0.533055 | $6.54 \cdot 10^{-5}$ | | compl. by $\bar{u}$ |
| 8 | 1.785556 | −0.533055 | $2.36 \cdot 10^{-11}$ | | |
| | −1.785556 | −0.533055 | $3.90 \cdot 10^{-11}$ | | conv. |
| | 0.093000 | 33.238803 | $1.42 \cdot 10^{-5}$ | 2 | Ritz sel. |
| | −0.093000 | 33.238803 | $1.54 \cdot 10^{-5}$ | | |
| 9 | 1.785556 | −0.533055 | $7.49 \cdot 10^{-12}$ | | |
| | −1.785556 | −0.533055 | $3.56 \cdot 10^{-11}$ | | |
| | 0.119251 | 25.802012 | $2.95 \cdot 10^{-8}$ | 2 | Ritz sel. |
| | −0.119251 | 25.802012 | $6.56 \cdot 10^{-7}$ | | |
| 10 | 1.785556 | −0.533055 | $3.97 \cdot 10^{-12}$ | | |
| | −1.785556 | −0.533055 | $3.56 \cdot 10^{-11}$ | | |
| | 0.119263 | 25.799287 | $4.55 \cdot 10^{-10}$ | 1 | Ritz sel. |
| | −0.119263 | 25.799287 | $7.09 \cdot 10^{-7}$ | | |
| 11 | 1.785556 | −0.533055 | $3.22 \cdot 10^{-12}$ | | |
| | −1.785556 | −0.533055 | $3.51 \cdot 10^{-11}$ | | |
| | 0.119263 | 25.799284 | $1.33 \cdot 10^{-11}$ | | conv. |
| | 0.119263 | 25.799282 | $5.32 \cdot 10^{-7}$ | | compl. by $\bar{u}$ |
| 12 | 1.785556 | −0.533055 | $9.61 \cdot 10^{-12}$ | | |
| | −1.785556 | −0.533055 | $2.55 \cdot 10^{-11}$ | | |
| | 0.119263 | 25.799285 | $3.45 \cdot 10^{-12}$ | | |
| | −0.119263 | 25.799285 | $1.45 \cdot 10^{-11}$ | | conv. |

Table 2: Results for Example 1 using GMRES for solving the correction equation

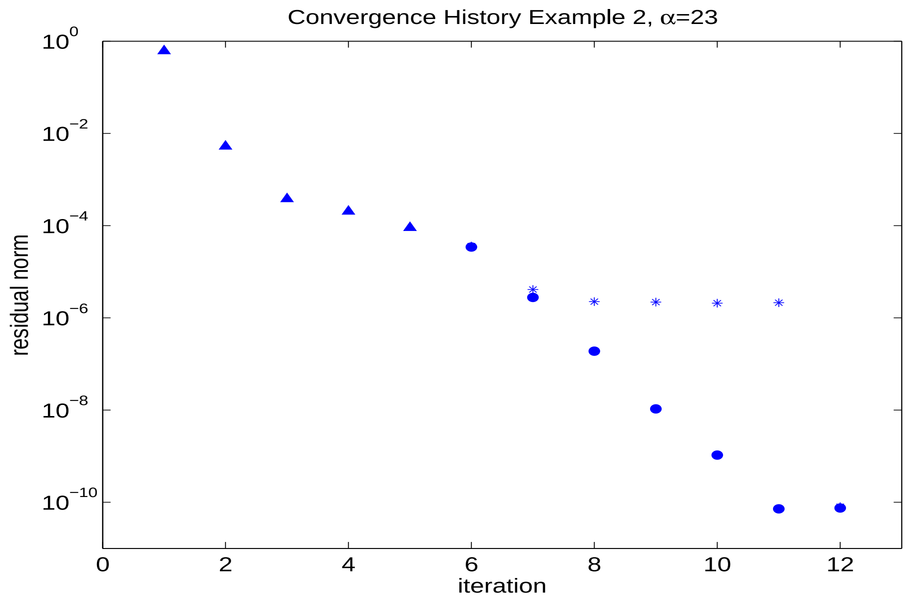


Figure 1: Convergence history for Example 2 with parameter $\alpha = 23$

(marked by a black square), but appending $\bar{u}$ to the search space convergence is obtained in the next step. A second, third and fourth pair of eigentriples appears in the projected problems in steps 6, 7, and 12, respectively. The convergence of these Ritz triples is displayed in the same way as for the first one in blue, red and green, where we marked the residuals with bullets if the Jacobi-Davidson method is aiming at them and with stars otherwise.

### 4.3 Example 3

Consider

$$u_t - \nabla((1 + x^2 + y^2 + z^2)\nabla u) + [1, 0, -1]\nabla u + u - \alpha(1 + x^2)u(t - \tau) = 0 \qquad (4.3)$$

with spatial variables $x$, $y$ and $z$ on $\Omega = (0,1)\times(0,1)\times(0,1)$ with Dirichlet boundary condition $u = 0$ on $\partial\Omega$. A discretization with piecewise quadratic ansatz functions on a tetrahedral grid using COMSOL yielded an eigenvalue problem (1.2) of dimension $n = 80623$. For $\alpha = 100$ the problem has four pairs of eigentriples.

Again we used `P=luinc(A,1e-3)` as preconditioner (which due to the higher population of the matrix $A$ required a CPU time of 94.4 seconds) and the same restart strategy and the same parameters for the GMRES solver as in Example 2. The method determined all 8 eigentriples needing 188.4 seconds. The convergence history is shown in Figure 3. It is organized in the same way as in Example 2.

## 5 Conclusions

We presented a new Jacobi-Davidson type method for the solution of the two-real-parameter eigenvalue problem governing critical delays of a linear system of delay differential equations.

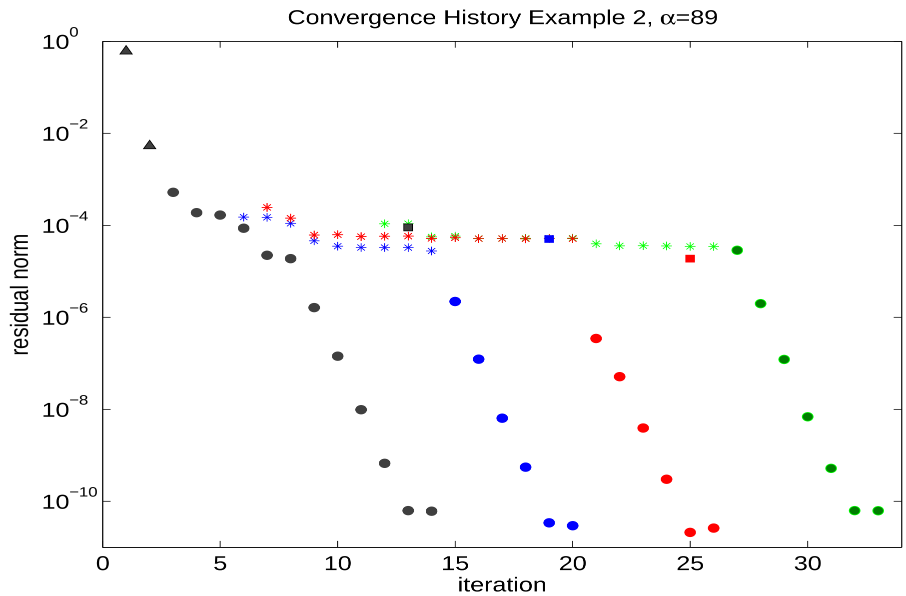


Figure 2: Convergence history for Example 2 with parameter $\alpha = 89$

The paper contributed in two important ways. Firstly, a structure preserving method was presented for solving the small scale problem, arising from the projection of the large scale problem on the search space. This method only computes real Ritz pairs $(\omega, \tau)$. Secondly, a correction equation was presented that forces $(\omega, \tau)$ to be real, as desired. We experienced that one cannot always rely on the availability of Ritz triples to continue the JD iteration. We came up with an alternative selection strategy that in all cases managed to bridge the steps until Ritz triples could be used. For all test problems, the method converges well, and (if the correction equations are solved accurately enough) shows quadratic convergence behaviour which is typical to this type of methods. We also showed that the method is reliable with respect to inexact solves of the correction equation, coarse preconditioners and the choice of starting vector.

## Acknowledgement

This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State Science Policy Office and the Center of Excellence OPTEC of the K.U.Leuven. The scientific responsibility rests with its author(s).

The work by Christian Schröder is supported by MATHEON, the DFG research Center in Berlin.

## References

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

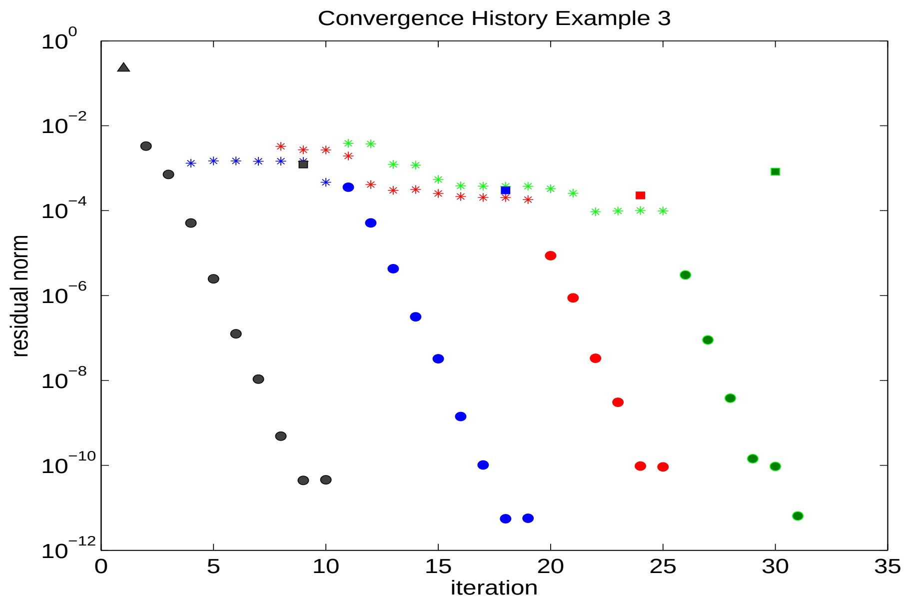


Figure 3: Convergence history for Example 3 with parameter $\alpha = 100$

[2] J. Chen, G. Gu, and C.N. Nett. A new method for computing delay margins for stability of linear delay systems. *Systems & Control Letters*, 26:107 – 117, 1995.

[3] J.J Dongarra, B. Straughan, and D.W. Walker. Chebyshev tau-QZ algorithm methods for calculating spectra of hydrodynamic stability problems. *Applied Numerical Mathematics*, 22:399–434, 1996.

[4] P. G. Drazin. *Introduction to hydrodynamic stability.* Cambridge University Press, Cambridge, UK, 2002.

[5] H. Fassbender, D.S. Mackey, N. Mackey, and C. Schroeder. Structured polynomial eigenproblems related to time-delay systems. *ETNA*, 31:306–330, 2008.

[6] D. Fokkema, G.L.G. Sleijpen, and H.A. van der Vorst. Jacobi-Davidson style QR and QZ algorithms for the reduction of matrix pencils. *SIAM Journal on Scientific Computing*, 20(1):pp.94–125, 1999.

[7] W. Govaerts. *Numerical methods for bifurcations of dynamical equilibria.* SIAM, Philadelphia, PA, USA, 2000.

[8] K. Gu, V.L. Kharitonov, and J. Chen. *Stability of Time–Delay Systems.* Birkhäuser, Boston, 2003.

[9] R.F. Heinemann and A.B. Poore. Multiplicity, stability, and oscillatory dynamics of a tubular reactor. *Chemical Engineering Science*, 36:1411–1419, 1981.

[10] M. E. Hochstenbach, T. Kosir, and B. Plestenjak. A Jacobi–Davidson type method for the two-parameter eigenvalue problem. *SIAM Journal on Matrix Analysis and Applications*, 26(2):477–497, 2005.

[11] E. Jarlebring. Critical delays and polynomial eigenvalue problems. *Journal on Computational and Applied Mathematics*, 224:296 – 306, 2009.

[12] E. Jarlebring and M.E. Hochstenbach. Polynomial two-parameter eigenvalue problems and matrix pencil methods for stability of delay-differential equations. *Linear Algebra and its Applications*, 431(3–4):369–380, 2009.

[13] E. Jarlebring, K. Meerbergen, and W. Michiels. An Arnoldi like method for the delay eigenvalue problem. *SIAM Journal on Scientific Computing*, 2010. Accepted for publication.

[14] Z. Jia. Polynomial characterizations of the approximate eigenvectors by the refined Arnoldi method and the implicitly restarted refined Arnoldi algorithm. *Linear Algebra and its Applications*, 287:191–214, 1998.

[15] J. Louisell. A matrix method for determining the imaginary axis eigenvalues of delay systems. *IEEE Trans. Automatic Control*, 46:2008 – 2012, 2001.

[16] K. Meerbergen and A. Spence. Shift-and-invert iteration for purely imaginary eigenvalues with application to the detection of Hopf bifurcations in large scale problems. *SIAM Journal on Matrix Analysis and Applications*, 31(4):1463–1482, 2010.

[17] S.-I. Niculecu. Stability and hyperbolicity of linear systems with delayed state: a matrix-pencil approach. *IMA J. Math. Contr. Inform.*, 15:331 – 347, 1998.

[18] B.N. Parlett and Y. Saad. Complex shift and invert strategies for real matrices. *Linear Algebra and its Applications*, 88/89:575–595, 1987.

[19] Y. Saad and M.H Schultz. GMRES : a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7:856–869, 1986.

[20] G.L. Sleijpen, G.L. Booten, D.R. Fokkema, and H.A. van der Vorst. Jacobi-Davidson type methods for generalized eigenproblems and polynomial eigenproblems. *BIT*, 36:595 – 633, 1996.

[21] G.L.G. Sleijpen and H.A. van der Vorst. A Jacobi-Davidson iteration method for linear eigenvalue problems. *SIAM Journal on Matrix Analysis and Applications*, 17:401–425, 1996.

[22] S. Stathopoulos, Y. Saad, and K. Wu. Dynamic thick restarting of the Davidson, and the implicitly restarted Arnoldi methods. *SIAM Journal on Scientific Computing*, 19(1):227–245, 1998.

[23] H Voss. A new justification of the Jacobi–Davidson method for large eigenproblems. *Linear Algebra and its Applications*, 424:448–455, 2007.