Journal of Computational and Applied Mathematics 288 (2015) 251–263

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Jacobi–Davidson methods for polynomial two-parameter eigenvalue problems

Michiel E. Hochstenbach[a], Andrej Muhič[b,*], Bor Plestenjak[b]

[a] Department of Mathematics and Computer Science, TU Eindhoven, PO Box 513, 5600 MB Eindhoven, The Netherlands

[b] Department of Mathematics, University of Ljubljana, Jadranska 19, SI-1000 Ljubljana, Slovenia

ARTICLE INFO

Article history:
Received 22 August 2014

Keywords:
Polynomial two-parameter eigenvalue problem (PMEP)
Jacobi–Davidson
Singular generalized eigenvalue problem
Bivariate polynomial equations
Determinantal representation
Delay differential equations (DDEs)

ABSTRACT

We propose Jacobi–Davidson type methods for polynomial two-parameter eigenvalue problems (PMEP). Such problems can be linearized as singular two-parameter eigenvalue problems, whose matrices are of dimension $k(k+1)n/2$, where k is the degree of the polynomial and n is the size of the matrix coefficients in the PMEP. When k^2n is relatively small, the problem can be solved numerically by computing the common regular part of the related pair of singular pencils. For large k^2n, computing all solutions is not feasible and iterative methods are required.

When k is large, we propose to linearize the problem first and then apply Jacobi–Davidson to the obtained singular two-parameter eigenvalue problem. The resulting method may for instance be used for computing zeros of a system of scalar bivariate polynomials close to a given target. On the other hand, when k is small, we can apply a Jacobi–Davidson type approach directly to the original matrices. The original matrices are projected onto a low-dimensional subspace, and the projected polynomial two-parameter eigenvalue problems are solved by a linearization.

1. Introduction

We consider the polynomial two-parameter eigenvalue problem (PMEP)

$$\begin{aligned} P_1(\lambda,\mu)\,\mathbf{x}_1 &= \sum_{i=0}^{k}\sum_{j=0}^{k-i}\lambda^i\mu^jA_{ij}\,\mathbf{x}_1=\mathbf{0},\\ P_2(\lambda,\mu)\,\mathbf{x}_2 &= \sum_{i=0}^{k}\sum_{j=0}^{k-i}\lambda^i\mu^jB_{ij}\,\mathbf{x}_2=\mathbf{0}, \end{aligned} \tag{1}$$

where P_1 and P_2 are bivariate polynomials with matrix coefficients of sizes $n_1\times n_1$ and $n_2\times n_2$, respectively. In this paper we will assume for convenience, but without loss of generality, that $n_1=n_2=n$.

We say that (λ,μ) is an eigenvalue of (1) and the tensor product $\mathbf{x}_1\otimes\mathbf{x}_2$ is the corresponding eigenvector, where $\mathbf{x}_1\neq\mathbf{0}$ and $\mathbf{x}_2\neq\mathbf{0}$. PMEPs arise, for instance, in the study of the critical delays for delay differential equations (DDEs) [1,2]. Another

[*] Corresponding author.
E-mail addresses: andrej.muhic@fmf.uni-lj.si (A. Muhič), bor.plestenjak@fmf.uni-lj.si (B. Plestenjak).
URL: http://www.win.tue.nl/~hochsten/ (M.E. Hochstenbach).

http://dx.doi.org/10.1016/j.cam.2015.04.019
0377-0427/© 2015 Elsevier B.V. All rights reserved.

application is a system of bivariate polynomials if $n = 1$. We will discuss both of these applications in Section 2. In the special case that $k = 2$ we speak of the quadratic multiparameter eigenvalue problem (QMEP); this problem has already been discussed in [1,3,4].

It is shown in [4] that, analogously to one-parameter matrix polynomials, a PMEP (1) can be linearized as a two-parameter eigenvalue problem (MEP) of the form

$$\begin{aligned} W_1(\lambda,\mu)\,\mathbf{z}_1 &= (A_1 + \lambda B_1 + \mu C_1)\ \mathbf{z}_1 = \mathbf{0},\\ W_2(\lambda,\mu)\,\mathbf{z}_2 &= (A_2 + \lambda B_2 + \mu C_2)\ \mathbf{z}_2 = \mathbf{0}, \end{aligned} \tag{2}$$

where the matrices A_i, B_i, and C_i are of size $N \times N$, with

$$N = \frac{1}{2}\,k\,(k+1)\,n.$$

More details on possible linearizations for the quadratic case ($k = 2$) are given in [3].

The standard approach to solve a MEP of the form (2) is to consider the related coupled pair of generalized eigenvalue problems

$$\begin{aligned} \Delta_1\,\mathbf{z} &= \lambda\,\Delta_0\,\mathbf{z},\\ \Delta_2\,\mathbf{z} &= \mu\,\Delta_0\,\mathbf{z} \end{aligned} \tag{3}$$

on the tensor product space $\mathbb{C}^N \otimes \mathbb{C}^N$, where $\mathbf{z} = \mathbf{z}_1 \otimes \mathbf{z}_2$, and the matrices

$$\begin{aligned} \Delta_0 &= B_1 \otimes C_2 - C_1 \otimes B_2,\\ \Delta_1 &= C_1 \otimes A_2 - A_1 \otimes C_2,\\ \Delta_2 &= A_1 \otimes B_2 - B_1 \otimes A_2 \end{aligned} \tag{4}$$

are the so-called operator determinants. Usually we assume that the MEP (2) is nonsingular, i.e., that the corresponding operator determinant Δ_0 is nonsingular. In that case (see, e.g., [5]), the matrices $\Delta_0^{-1}\Delta_1$ and $\Delta_0^{-1}\Delta_2$ commute and the eigenvalues of (2) agree with the eigenvalues of (3). Because of this relation, a nonsingular MEP can be numerically solved using a modified QZ method for the coupled generalized eigenvalue problems; see [6] for an algorithm. If N^2 is large, the computation of all eigenvalues is not feasible. Instead, one may exploit iterative methods to compute a small number of eigenvalues close to a given target. A particular method of choice is Jacobi–Davidson. For details on Jacobi–Davidson type methods for the nonsingular MEP we refer to [6–8].

Unfortunately, the MEP obtained by a linearization of the PMEP (1) is singular and the approach described in the previous paragraph cannot be applied. In this case, as shown in [9], if all eigenvalues of (1) are algebraically simple, which is the generic case, they agree with the finite regular eigenvalues of the pair of singular matrix pencils (3). A numerical algorithm from [4], based on the staircase algorithm by Van Dooren [10], may then be used to compute the common regular part of (3) and extract the finite regular eigenvalues. The algorithm returns matrices Q and P with orthonormal columns such that the finite regular eigenvalues are the eigenvalues of the following pair of generalized eigenvalue problems

$$\begin{aligned} \widetilde{\Delta}_1\,\mathbf{z} &= \lambda\,\widetilde{\Delta}_0\,\mathbf{z},\\ \widetilde{\Delta}_2\,\mathbf{z} &= \mu\,\widetilde{\Delta}_0\,\mathbf{z}, \end{aligned} \tag{5}$$

where $\widetilde{\Delta}_i = P^*\Delta_i Q$ for $i = 0, 1, 2$ and $\widetilde{\Delta}_0$ is nonsingular. As in the nonsingular case, this approach is feasible only when N^2 is relatively small. In this paper we consider the case when N^2 is large, and develop Jacobi–Davidson type methods that may be applied to compute some eigenvalues of (1) close to a given target.

The outline of the paper is as follows. We first review some applications in Section 2. In Section 3 we briefly outline how the Jacobi–Davidson type methods from [6,7] can be extended to the regular singular MEP. We may apply this extension to the linearization of PMEP for large k and small n, and special case where the task is to find some roots of a bivariate scalar polynomial system ($n = 1$). In Section 4 we develop a new Jacobi–Davidson type method directly for the PMEP (1) when k is small and n is large. This method may be viewed as a generalization of the Jacobi–Davidson method for polynomial eigenvalue problems [11]. We present three different techniques for the subspace extraction and extend the results presented in [11]. In the last section we give some numerical results with the applications in Section 2.

2. Motivating problems

We can linearize the PMEP (1) as a MEP (2). If we assume that $\det(P_1(\lambda,\mu))$ and $\det(P_2(\lambda,\mu))$ do not share a common factor, then we obtain a regular singular problem, which we can solve by the Jacobi–Davidson type method described in Algorithm 1.

Problem 1. The PMEP appears in the study of critical delays of DDEs [1,2]. For example, the neutral commensurate DDE with m delays

$$\sum_{k=0}^{m} B_k\,\dot{x}(t-\tau k)=\sum_{k=0}^{m} A_k\,x(t-\tau k)$$

leads to the following PMEP of degree $m+1$

$$\begin{aligned} A_0\mathbf{x} &= \lambda\sum_{k=0}^{m}\mu^k B_k\mathbf{x}-\sum_{k=1}^{m}\mu^k A_k\mathbf{x},\\ -A_m^*\mathbf{y} &= \lambda\sum_{k=0}^{m}\mu^k B_{m-k}^*\mathbf{y}+\sum_{k=1}^{m}\mu^k A_{m-k}^*\mathbf{y}. \end{aligned}$$

In [2], a Jacobi–Davidson type method is proposed for the special case $m = 1$ and $B_1 = 0$. The methods proposed in this paper can be used for the general case where $m > 1$, but also for other problems such as those reviewed in this section.

The PMEP from Problem 1 has many zero coefficients and this can be exploited in a linearization (see Example 13). To reveal the structure of the linearization for a general PMEP, we consider the cubic case. Details on a linearization for a general degree k can be found in [9, Appendix].

Problem 2. A cubic two-parameter eigenvalue problem has the form

$$\begin{aligned} (A_{00}+\cdots+\lambda^3A_{30}+\lambda^2\mu A_{21}+\lambda\mu^2A_{12}+\mu^3A_{03})\,\mathbf{x}_1 &= \mathbf{0},\\ (B_{00}+\cdots+\lambda^3B_{30}+\lambda^2\mu B_{21}+\lambda\mu^2B_{12}+\mu^3B_{03})\,\mathbf{x}_2 &= \mathbf{0}. \end{aligned} \tag{6}$$

In the generic case the problem has $9n^2$ eigenvalues. We linearize the first equation of (6) as

$$\begin{bmatrix} A_{00} & A_{10} & A_{01} & A_{20}+\lambda A_{30} & A_{11}+\lambda A_{21} & A_{02}+\lambda A_{12}+\mu A_{03}\\ \lambda I & -I & 0 & 0 & 0 & 0\\ \mu I & 0 & -I & 0 & 0 & 0\\ 0 & \lambda I & 0 & -I & 0 & 0\\ 0 & 0 & \lambda I & 0 & -I & 0\\ 0 & 0 & \mu I & 0 & 0 & -I \end{bmatrix}\begin{bmatrix}\mathbf{x}_1\\ \lambda\mathbf{x}_1\\ \mu\mathbf{x}_1\\ \lambda^2\mathbf{x}_1\\ \lambda\mu\mathbf{x}_1\\ \mu^2\mathbf{x}_1\end{bmatrix}=\mathbf{0}.$$

In a similar way we write the second equation and thus linearize (6) as a MEP $W_1(\lambda,\mu)\,\mathbf{z}_1=\mathbf{0}$ and $W_2(\lambda,\mu)\,\mathbf{z}_2=\mathbf{0}$ with matrices of dimension $6n$. The corresponding Δ_0 is singular.

Problem 3. A special case of the PMEP, where all coefficients are scalars, is a bivariate polynomial system

$$\begin{aligned} p_1(x,y) &= \sum_{i=0}^{k}\sum_{j=0}^{k-i} a_{ij}\,x^iy^j=0,\\ p_2(x,y) &= \sum_{i=0}^{k}\sum_{j=0}^{k-i} b_{ij}\,x^iy^j=0. \end{aligned} \tag{7}$$

As described in [4, Theorem 22], we may linearize this system as a regular singular MEP

$$\begin{aligned} W_1(x,y)\,\mathbf{z}_1 &= (A_1+xB_1+yC_1)\ \mathbf{z}_1=\mathbf{0},\\ W_2(x,y)\,\mathbf{z}_2 &= (A_2+xB_2+yC_2)\ \mathbf{z}_2=\mathbf{0} \end{aligned}$$

with matrices of dimension $\frac{1}{2}k(k+1)$. This allows us to use standard numerical linear algebra tools to compute all or some of the solutions in a similar way as companion matrix is used in Matlab's command `roots` [12] to compute the zeros of a scalar univariate polynomial. Similar ideas can be found in [13,14]. An advantage of our approach is that it does not require symbolic computation; a disadvantage are very large matrices. However, these matrices are also very sparse and each matrix–vector (MV) multiplication costs $O(k^2)$ operations, which is the same order as the evaluation of the polynomials p_1 and p_2 at a given pair (x,y). See Example 14 for the computation of some roots close to a given target (x_0,y_0).

3. Jacobi–Davidson applied to the linearization

We consider a singular MEP (2), where all linear combinations of the corresponding Δ-matrices (4) are singular. We assume that the problem is *regular singular*, which means that neither of the polynomials $p_1(\lambda,\mu)=\det(W_1(\lambda,\mu))$ and

$p_2(\lambda, \mu) = \det(W_2(\lambda, \mu))$ is identically zero and they do not share a common factor [9]. We obtain a problem with such properties when we linearize the PMEP (1) as a MEP [9].

Algorithm 1: A Jacobi–Davidson method for a regular singular MEP.

```
    Input: a regular singular MEP $(A_1 + \lambda B_1 + \mu C_1)\,\mathbf{z}_1 = \mathbf{0}$, $(A_2 + \lambda B_2 + \mu C_2)\,\mathbf{z}_2 = \mathbf{0}$
        starting vectors $\mathbf{u}_1, \mathbf{u}_2$, a target $(\sigma, \tau)$, and a tolerance $\varepsilon$.
    Output: an approximate eigenpair $((\theta, \eta), \mathbf{u}_1 \otimes \mathbf{u}_2)$.
1:  Set $\mathbf{s}_1 = \mathbf{u}_1,\ \mathbf{s}_2 = \mathbf{u}_2,\ U_{10} = [\,]$, and $U_{20} = [\,]$.
    for $k = 1, 2, \ldots$
2:    Expand the search spaces. Set $U_{ik} = \mathrm{rgs}(U_{i,k-1}, \mathbf{s}_i)$ for $i = 1, 2$
3:    Update the appropriate test spaces $V_{ik}$ for $i = 1, 2$
4:    Select an appropriate Petrov value $(\theta, \eta)$ and the corresponding Petrov vector $\mathbf{u}_1 \otimes \mathbf{u}_2$
      from the projected (potentially singular) MEP
          $(V_{1k}^* A_1 U_{1k} + \theta\, V_{1k}^* B_1 U_{1k} + \eta\, V_{1k}^* C_1 U_{1k})\,\mathbf{c}_1 = \mathbf{0}$,
          $(V_{2k}^* A_2 U_{2k} + \theta\, V_{2k}^* B_2 U_{2k} + \eta\, V_{2k}^* C_2 U_{2k})\,\mathbf{c}_2 = \mathbf{0}$,
      where $\mathbf{u}_i = U_{ik}\mathbf{c}_i$ for $i = 1, 2$.
5:    Compute the residual $\mathbf{r}_i = (A_i + \theta B_i + \eta C_i)\,\mathbf{u}_i$ for $i = 1, 2$
6:    Stop if $\|\mathbf{r}_1\|, \|\mathbf{r}_2\| \le \varepsilon$
7:    Solve one of the proposed correction equations approximately, e.g.,
          $(I - \mathbf{u}_1\mathbf{u}_1^*)(A_1 + \theta B_1 + \eta C_1)\,\mathbf{s}_1 = -\mathbf{r}_1$,    $\mathbf{s}_1 \perp \mathbf{u}_1$,
          $(I - \mathbf{u}_2\mathbf{u}_2^*)(A_2 + \theta B_2 + \eta C_2)\,\mathbf{s}_2 = -\mathbf{r}_2$,    $\mathbf{s}_2 \perp \mathbf{u}_2$.
```

A pseudocode of the Jacobi–Davidson method for a regular singular MEP is presented in Algorithm 1. The Jacobi–Davidson method belongs to a class of subspace methods. Its main ingredients are the extraction and the expansion phase. In the extraction phase we compute approximations of eigenpairs by solving smaller eigenvalue problems of type (2). After each step we expand the search subspace by a new direction, and as the search subspace grows, we expect a convergence of eigenpair approximations to an eigenpair of the problem (2). Some features such as alternative correction equations, restarts, and preconditioning, are not included. For more details, see [6, Algorithm 4.1] for a nonsingular MEP, as well as the harmonic Rayleigh–Ritz extraction in [7].

In Step 2, rgs denotes repeated Gram–Schmidt orthogonalization. In Step 3 the test space is updated. If we use a two-sided approach as in [6], then we need initial vectors for the test space and we expand it by approximate solutions of additional correction equations. The idea of the two-sided Jacobi–Davidson method is that we get approximations for the right and the left eigenvectors. In the one-sided version the test spaces are equal to the search spaces, i.e., $V_{ik} = U_{ik}$ for $i = 1, 2$. Then we just approximate right eigenvectors and we have no guarantee about the approximation of an eigenvalue. To improve this, we may use the harmonic approach described in [7]. Here, if we are looking for eigenvalues close to the target (λ_T, μ_T), we take $V_{ik} = (A_i + \lambda_T B_i + \mu_T C_i)U_{ik}$ for $i = 1, 2$.

In Step 4 we have k-dimensional search spaces $\mathcal{U}_{ik} = \mathrm{span}(U_{ik})$ and test spaces $\mathcal{V}_{ik} = \mathrm{span}(V_{ik})$ for $i = 1, 2$. We get eigenvalue approximations from the Petrov–Galerkin conditions $(A_i + \theta\, B_i + \eta\, C_i)\,\mathbf{u}_i \perp \mathcal{V}_{ik}$, where $\mathbf{u}_i \in \mathcal{U}_{ik}$, for $i = 1, 2$. If we write $\mathbf{u}_i = U_{ik}\mathbf{c}_i$ then these conditions lead to the smaller projected MEP in Step 4. The main difference to algorithms for nonsingular MEP is that the projected MEP might be singular, therefore we numerically solve it using the method for singular MEP from [9].

4. Jacobi–Davidson method applied directly to the polynomial system

If k is small enough, then another option is to apply a Jacobi–Davidson type method directly to the PMEP (1) instead to the corresponding linearization in the form of a MEP. The same options are possible (see, e.g., [11]) in the one-parameter polynomial eigenvalue problem, where we can either linearize the problem first and apply the Jacobi–Davidson method to the linearized problem or we can apply the Jacobi–Davidson method directly to the initial matrices.

The method that we will describe next may be viewed as a generalization of the method for the polynomial eigenvalue problem presented in [11]. If we use search subspaces of size m, then the linearization of the projected PMEP leads to Δ-matrices of size $\frac{1}{4}m^2k^2(k+1)^2$. The presence of the term k^4 limits the largest usable m and this is the reason why this approach is limited only to PMEP of a small order k.

4.1. Subspace extraction

Suppose we have m-dimensional search spaces $\mathcal{U}_i$ for the vectors $\mathbf{x}_i$ and let the columns of U_i form orthonormal bases for $\mathcal{U}_i$, where $i = 1, 2$. We look for an approximation $((\theta, \eta), \mathbf{u}_1 \otimes \mathbf{u}_2)$ to an eigenpair $((\lambda, \mu), \mathbf{x}_1 \otimes \mathbf{x}_2)$, such that $\mathbf{u}_i \in \mathcal{U}_i$. Hence, we can write $\mathbf{u}_i = U_i\mathbf{c}_i$ for certain vector $\mathbf{c}_i \in \mathbb{C}^m$ of unit length. We first focus on the extraction of the approximate eigenvector, and next discuss the approximate eigenvalue.

It is natural to call the Galerkin conditions

$$\begin{aligned} &P_1(\theta, \eta)\,\mathbf{u}_1 \perp \mathcal{U}_1, \\ &P_2(\theta, \eta)\,\mathbf{u}_2 \perp \mathcal{U}_2 \end{aligned}$$

the standard Rayleigh–Ritz extraction. These conditions lead to a low-dimensional projected problem of the same type:

$$\begin{aligned} U_1^* P_1(\theta,\eta)\, U_1\mathbf{c}_1 &= \mathbf{0},\\ U_2^* P_2(\theta,\eta)\, U_2\mathbf{c}_2 &= \mathbf{0}. \end{aligned}$$

If exact eigenvectors are present in the search space, they satisfy the Galerkin conditions that define the standard extraction. However, as is also common for eigenvalue problems of other types (see, e.g., [15]), the standard extraction may not be suitable for the selection of interior eigenvalues. For this purpose, we now introduce the refined and harmonic extraction for the polynomial multiparameter eigenvalue problem.

Suppose that we are looking for eigenvalues close to the target (λ_T,μ_T). The *refined extraction* aims at minimizing the residual

$$\widehat{\mathbf{c}}_i = \operatorname*{argmin}_{\|\mathbf{c}_i\|=1} \|P_i(\lambda_T,\mu_T)U_i\,\mathbf{c}_i\|$$

for $i = 1, 2$. The vectors $\widehat{\mathbf{u}}_1 = U_1\widehat{\mathbf{c}}_1$ and $\widehat{\mathbf{u}}_2 = U_2\widehat{\mathbf{c}}_2$ are called refined Ritz vectors. The refined extraction has the advantage of having minimal residuals, but will generally not select exact eigenvectors that are present in the search space.

Proposition 4. *For the residual of the refined Ritz vector we have*

$$\|P_i(\lambda_T,\mu_T)\widehat{\mathbf{u}}_i\| \le \frac{\xi_{1i}\,|\lambda_T-\lambda| + \xi_{2i}\,|\mu_T-\mu| + \|P_i(\lambda_T,\mu_T)\|\,\sin(\mathcal{U}_i,\mathbf{x}_i)}{\sqrt{1-\sin^2(\mathcal{U}_i,\mathbf{x}_i)}},$$

where, for $i = 1, 2$,

$$\begin{aligned} \xi_{1i} &= \max_{t\in[0,1]} \left\| \frac{\partial P_i}{\partial\lambda}(t(\lambda-\lambda_T)+\lambda_T,\mu_T)\right\|,\\ \xi_{2i} &= \max_{t\in[0,1]} \left\| \frac{\partial P_i}{\partial\mu}(\lambda_T, t(\mu-\mu_T)+\mu_T)\right\|. \end{aligned}$$

Proof. Decompose $\mathbf{x}_i = \gamma_i\widehat{\mathbf{x}}_i + \sigma_i\mathbf{s}_i$, where $\widehat{\mathbf{x}}_i := U_iU_i^*\mathbf{x}_i/\|U_i^*\mathbf{x}_i\|$ is the orthogonal projection of $\mathbf{x}_i$ onto $\mathcal{U}_i$, $\|\widehat{\mathbf{x}}_i\| = \|\mathbf{s}_i\| = 1$, $\gamma_i = \cos(\mathcal{U}_i,\mathbf{x}_i)$, and $\sigma_i = \sin(\mathcal{U}_i,\mathbf{x}_i)$. First, we notice that

$$\begin{aligned} P_i(\lambda_T,\mu_T) &= P_i(\lambda_T,\mu_T) - P_i(\lambda,\mu_T) + P_i(\lambda,\mu_T) - P_i(\lambda,\mu)\\ &= \int_0^1 \frac{d}{d\alpha}P_i(\alpha(\lambda_T-\lambda)+\lambda,\mu_T)\,d\alpha + \int_0^1 \frac{d}{d\alpha}P_i(\lambda,\alpha(\mu_T-\mu)+\mu)\,d\alpha\\ &= (\lambda_T-\lambda)\left(\int_0^1 \frac{\partial P_i}{\partial\lambda}(\alpha(\lambda_T-\lambda)+\lambda,\mu_T)\,d\alpha\right) + (\mu_T-\mu)\left(\int_0^1 \frac{\partial P_i}{\partial\mu}(\lambda,\alpha(\mu_T-\mu)+\mu)\,d\alpha\right), \end{aligned}$$

so

$$\|P_i(\lambda_T,\mu_T)\,\mathbf{x}_i\| \le \xi_{1i}\,|\lambda_T-\lambda| + \xi_{2i}\,|\mu_T-\mu|.$$

Since $P_i(\lambda_T,\mu_T)\,\widehat{\mathbf{x}}_i = (P_i(\lambda_T,\mu_T)\,\mathbf{x}_i - \sigma_iP_i(\lambda_T,\mu_T)\,\mathbf{s}_i)/\gamma_i$, we have by the definition of a refined Ritz vector

$$\begin{aligned} \|P_i(\lambda_T,\mu_T)\widehat{\mathbf{u}}\| &\le \|P_i(\lambda_T,\mu_T)\,\widehat{\mathbf{x}}_i\|\\ &\le (\xi_{1i}\,|\lambda_T-\lambda| + \xi_{2i}\,|\mu_T-\mu| + \sigma_i\,\|P_i(\lambda_T,\mu_T)\|)\,/\gamma_i. \quad \Box \end{aligned}$$

This result suggests that for the convergence ($\|P_i(\lambda_T,\mu_T)\widehat{\mathbf{u}}_i\| \to 0$) it is not sufficient that $\sin(\mathcal{U}_i,\mathbf{x}_i) \to 0$; in addition, the target (λ_T,μ_T) has to converge to an exact eigenvalue. This means that we have to vary the target during the process, which increases computational requirements.

A *harmonic extraction* attempts to combine the advantages of recognizing the exact eigenvectors if they are present in the search space (as in the standard extraction), and having small residuals (as in the refined approach). We look for Galerkin conditions of the form

$$\begin{aligned} P_1(\theta,\eta)\,\mathbf{u}_1 &\perp \widetilde{\mathcal{U}}_1,\\ P_2(\theta,\eta)\,\mathbf{u}_2 &\perp \widetilde{\mathcal{U}}_2 \end{aligned}$$

for certain test spaces $\widetilde{\mathcal{U}}_1$ and $\widetilde{\mathcal{U}}_2$, striving for small residuals. Since

$$P_i(\theta,\eta)\,\mathbf{u}_i = P_i(\lambda_T,\mu_T)\,\mathbf{u}_i + (\theta-\lambda_T)\frac{\partial P_i}{\partial\lambda}(\lambda_T,\mu_T)\,\mathbf{u}_i + (\eta-\mu_T)\frac{\partial P_i}{\partial\mu}(\lambda_T,\mu_T)\,\mathbf{u}_i + \mathcal{O}\Big(\big(|\theta-\lambda_T| + |\eta-\mu_T|\big)^2\Big),$$

we are interested in the Galerkin conditions

$$\begin{aligned} P_1(\theta,\eta)\,\mathbf{u}_1 &\perp P_1(\lambda_T,\mu_T)\,\mathcal{U}_1,\\ P_2(\theta,\eta)\,\mathbf{u}_2 &\perp P_2(\lambda_T,\mu_T)\,\mathcal{U}_2. \end{aligned}$$

The motivation for this is the following.

$$\begin{aligned} 0 &= \mathbf{u}_i^* P_i(\lambda_T, \mu_T)^* P_i(\theta, \eta)\, \mathbf{u}_i \\ &\approx \mathbf{u}_i^* P_i(\lambda_T, \mu_T)^* \Big(P_i(\lambda_T, \mu_T)\, \mathbf{u}_i + (\theta - \lambda_T) \frac{\partial P_i}{\partial \lambda}(\lambda_T, \mu_T)\, \mathbf{u}_i + (\eta - \mu_T) \frac{\partial P_i}{\partial \mu}(\lambda_T, \mu_T)\, \mathbf{u}_i \Big) \\ &= \|P_i(\lambda_T, \mu_T)\, \mathbf{u}_i\|^2 + \mathbf{u}_i^* P_i(\lambda_T, \mu_T)^* \Big((\theta - \lambda_T) \frac{\partial P_i}{\partial \lambda}(\lambda_T, \mu_T)\, \mathbf{u}_i + (\eta - \mu_T) \frac{\partial P_i}{\partial \mu}(\lambda_T, \mu_T)\, \mathbf{u}_i \Big). \end{aligned}$$

Invoking Cauchy–Schwarz, we get

$$\|P_i(\lambda_T, \mu_T)\, \mathbf{u}_i\| \lesssim |\theta - \lambda_T| \left\| \frac{\partial P_i}{\partial \lambda}(\lambda_T, \mu_T)\, \mathbf{u}_i \right\| + |\eta - \mu_T| \left\| \frac{\partial P_i}{\partial \mu}(\lambda_T, \mu_T)\, \mathbf{u}_i \right\|$$

for $i = 1, 2$. This means that for the harmonic extraction, in contrast to the refined extraction, we are guaranteed to have small residuals if there is a harmonic Ritz value (θ, η) close to (λ_T, μ_T).

The refined and harmonic extractions can be efficiently implemented by introducing the QR decompositions

$$P_1(\lambda_T, \mu_T)\, U_1 = Q_1 R_1, \qquad P_2(\lambda_T, \mu_T)\, U_2 = Q_2 R_2,$$

which can be incrementally computed during the process. Using these, in the harmonic approach we have to solve the projected problem

$$Q_1^* P_1(\theta, \eta)\, U_1 \mathbf{c}_1 = \mathbf{0}, \qquad Q_2^* P_2(\theta, \eta)\, U_2 \mathbf{c}_2 = \mathbf{0}.$$

We note that this harmonic extraction is a generalization of the approaches in [11] (for the one-parameter polynomial eigenvalue problem) and [7] (for the linear two-parameter eigenvalue problem).

In our numerical experiments we obtained the best results by the standard and harmonic extraction. The refined extraction did not improve the convergence and we did not include it in results in Section 5.

4.2. Subspace expansion

Suppose $((\theta, \eta), \mathbf{u}_1 \otimes \mathbf{u}_2)$ is an approximation to an eigenpair $((\lambda, \mu), \mathbf{x}_1 \otimes \mathbf{x}_2)$. We would like to improve our vectors by orthogonal updates $\mathbf{s}_i \perp \mathbf{u}_i$, such that

$$P_i(\lambda, \mu)(\mathbf{u}_i + \mathbf{s}_i) = \mathbf{0}.$$

Rewriting this gives

$$P_i(\theta, \eta)\, \mathbf{s}_i = -P_i(\theta, \eta)\, \mathbf{u}_i + (P_i(\theta, \eta) - P_i(\lambda, \mu))\, \mathbf{u}_i + (P_i(\theta, \eta) - P_i(\lambda, \mu))\, \mathbf{s}_i \tag{8}$$

for $i = 1, 2$. We need the following lemma.

Lemma 5. *For given nonzero vectors $\mathbf{w}_1 \in \mathbb{C}^n$ and $\mathbf{w}_2 \in \mathbb{C}^n$, let*

$$F(\theta, \eta, \mathbf{u}_1, \mathbf{u}_2) = \begin{bmatrix} \mathbf{w}_1^* P_1(\theta, \eta)\, \mathbf{u}_1 \\ \mathbf{w}_2^* P_2(\theta, \eta)\, \mathbf{u}_2 \end{bmatrix}.$$

If

$$\frac{\partial F}{\partial(\theta, \eta)} := \begin{bmatrix} \dfrac{\partial F_1}{\partial \theta} & \dfrac{\partial F_1}{\partial \eta} \\ \dfrac{\partial F_2}{\partial \theta} & \dfrac{\partial F_2}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \mathbf{w}_1^* \dfrac{\partial P_1}{\partial \theta} \mathbf{u}_1 & \mathbf{w}_1^* \dfrac{\partial P_1}{\partial \eta} \mathbf{u}_1 \\ \mathbf{w}_2^* \dfrac{\partial P_2}{\partial \theta} \mathbf{u}_2 & \mathbf{w}_2^* \dfrac{\partial P_2}{\partial \eta} \mathbf{u}_2 \end{bmatrix}$$

is nonsingular, then

$$\begin{aligned} |\theta(\mathbf{u}_1 + \mathbf{s}_1, \mathbf{u}_2 + \mathbf{s}_2) - \theta(\mathbf{u}_1, \mathbf{u}_2)| &= \mathcal{O}(\|\mathbf{s}_1\| + \|\mathbf{s}_2\|), \\ |\eta(\mathbf{u}_1 + \mathbf{s}_1, \mathbf{u}_2 + \mathbf{s}_2) - \eta(\mathbf{u}_1, \mathbf{u}_2)| &= \mathcal{O}(\|\mathbf{s}_1\| + \|\mathbf{s}_2\|). \end{aligned}$$

Proof. Under the hypothesis, the Implicit Function Theorem yields

$$\frac{\partial(\theta, \eta)}{\partial(\mathbf{u}_1, \mathbf{u}_2)} = -\left(\frac{\partial F}{\partial(\theta, \eta)} \right)^{-1} \frac{\partial F}{\partial(\mathbf{u}_1, \mathbf{u}_2)},$$

from which the result follows easily. □

We will revisit the assumption of Lemma 5 in Proposition 6. Regarding (8), we have

$$P_i(\theta, \eta) - P_i(\lambda, \mu) = (\theta - \lambda) \frac{\partial P_i}{\partial \lambda}(\theta, \eta) + (\eta - \mu) \frac{\partial P_i}{\partial \mu}(\theta, \eta) + \mathcal{O}\Big(\big(|\lambda - \theta| + |\mu - \eta| \big)^2 \Big)$$

for $i = 1, 2$. Using the lemma, we derive

$$\begin{aligned} |\lambda-\theta| &= |\theta(\mathbf{u}_1+\mathbf{s}_1,\mathbf{u}_2+\mathbf{s}_2)-\theta(\mathbf{u}_1,\mathbf{u}_2)| = \mathcal{O}(\|\mathbf{s}_1\|+\|\mathbf{s}_2\|),\\ |\mu-\eta| &= |\eta(\mathbf{u}_1+\mathbf{s}_1,\mathbf{u}_2+\mathbf{s}_2)-\eta(\mathbf{u}_1,\mathbf{u}_2)| = \mathcal{O}(\|\mathbf{s}_1\|+\|\mathbf{s}_2\|),\end{aligned}$$

where we overloaded the symbols θ and η for both scalar values as well as functions. Using this we see that the rightmost terms in (8) are of order $\mathcal{O}((\|\mathbf{s}_1\|+\|\mathbf{s}_2\|)^2)$. Therefore, we discard these terms. We define the residual by

$$r = \begin{bmatrix} P_1(\theta,\eta)\,\mathbf{u}_1 \\ P_2(\theta,\eta)\,\mathbf{u}_2 \end{bmatrix}. \tag{9}$$

So, up to the second-order terms in $\lambda-\theta$ and $\mu-\eta$ we have

$$\begin{bmatrix} P_1(\theta,\eta) & \\ & P_2(\theta,\eta)\end{bmatrix}\begin{bmatrix}\mathbf{s}_1\\ \mathbf{s}_2\end{bmatrix} = -\mathbf{r} + (\lambda-\theta)\begin{bmatrix} \dfrac{\partial P_1}{\partial\theta}(\theta,\eta)\,\mathbf{u}_1 \\ \dfrac{\partial P_2}{\partial\theta}(\theta,\eta)\,\mathbf{u}_2\end{bmatrix} + (\mu-\eta)\begin{bmatrix} \dfrac{\partial P_1}{\partial\eta}(\theta,\eta)\,\mathbf{u}_1 \\ \dfrac{\partial P_2}{\partial\eta}(\theta,\eta)\,\mathbf{u}_2\end{bmatrix}. \tag{10}$$

Now we define

$$Z = \begin{bmatrix} \dfrac{\partial P_1}{\partial\theta}(\theta,\eta)\,\mathbf{u}_1 & \dfrac{\partial P_1}{\partial\eta}(\theta,\eta)\,\mathbf{u}_1 \\ \dfrac{\partial P_2}{\partial\theta}(\theta,\eta)\,\mathbf{u}_2 & \dfrac{\partial P_2}{\partial\eta}(\theta,\eta)\,\mathbf{u}_2\end{bmatrix}, \qquad U = \begin{bmatrix}\mathbf{u}_1 & \\ & \mathbf{u}_2\end{bmatrix},$$

and the projector

$$Q = I - Z(U^*Z)^{-1}U^*. \tag{11}$$

This projector annihilates the second and the third term on the right-hand side of (10), and additionally fixes the residual. By projecting (10) we get the correction equation

$$Q\begin{bmatrix} P_1(\theta,\eta) & \\ & P_2(\theta,\eta)\end{bmatrix}\begin{bmatrix}\mathbf{s}_1\\ \mathbf{s}_2\end{bmatrix} = -\mathbf{r}, \quad \mathbf{s}_1\perp\mathbf{u}_1,\ \mathbf{s}_2\perp\mathbf{u}_2. \tag{12}$$

This equation may be solved exactly or inexactly. We will study the resulting convergence in the next section.

4.3. Convergence

To generalize theoretical results on the convergence from the linear MEP to the polynomial MEP we consider a nonlinear two-parameter eigenvalue problem

$$\begin{aligned} T_1(\lambda,\mu)\,\mathbf{z}_1 &= \mathbf{0},\\ T_2(\lambda,\mu)\,\mathbf{z}_2 &= \mathbf{0},\end{aligned} \tag{13}$$

where $T_i(\cdot,\cdot): \mathbb{C}\times\mathbb{C}\to\mathbb{C}^{n_i\times n_i}$ is differentiable for $i = 1, 2$. If (13) is satisfied for nonzero vectors $\mathbf{z}_1$ and $\mathbf{z}_2$, then (λ,μ) is an eigenvalue and $\mathbf{z}_1\otimes\mathbf{z}_2$ is the corresponding right eigenvector. The corresponding left eigenvector is $\mathbf{w}_1\otimes\mathbf{w}_2$ such that $\mathbf{w}_i\neq\mathbf{0}$ and $\mathbf{w}_i^*T_i(\lambda,\mu)=\mathbf{0}$ for $i = 1, 2$. We will need the following result; cf. also Lemma 5.

Proposition 6 ([9, Proposition 3.2]). *Let (λ,μ) be an algebraically simple eigenvalue of the nonlinear two-parameter eigenvalue problem (13) and let $\mathbf{z}_1\otimes\mathbf{z}_2$ and $\mathbf{w}_1\otimes\mathbf{w}_2$ be the corresponding right and left eigenvector. Then the matrix*

$$M_0 := \begin{bmatrix} \mathbf{w}_1^*\dfrac{\partial T_1}{\partial\lambda}(\lambda,\mu)\,\mathbf{z}_1 & \mathbf{w}_1^*\dfrac{\partial T_1}{\partial\mu}(\lambda,\mu)\,\mathbf{z}_1 \\ \mathbf{w}_2^*\dfrac{\partial T_2}{\partial\lambda}(\lambda,\mu)\,\mathbf{z}_2 & \mathbf{w}_2^*\dfrac{\partial T_2}{\partial\mu}(\lambda,\mu)\,\mathbf{z}_2\end{bmatrix} \tag{14}$$

is nonsingular.

Using this proposition we can prove the following results about the asymptotic convergence.

Proposition 7. *Suppose that the Jacobi–Davidson method for the PMEP, where we solve the correction equation (12) exactly, converges to an algebraically simple eigenvalue (λ,μ). Then the method converges asymptotically quadratically.*

Proof. Let Q be as in (11). The true updates $\mathbf{s}_1\perp\mathbf{u}_1$ and $\mathbf{s}_2\perp\mathbf{u}_2$ satisfy (cf. (8))

$$Q\begin{bmatrix} P_1(\theta,\eta) & \\ & P_2(\theta,\eta)\end{bmatrix}\begin{bmatrix}\mathbf{s}_1\\ \mathbf{s}_2\end{bmatrix} = -\mathbf{r} + Q\begin{bmatrix}\Delta P_1 & \\ & \Delta P_2\end{bmatrix}\begin{bmatrix}\mathbf{u}_1\\ \mathbf{u}_2\end{bmatrix} + Q\begin{bmatrix}\Delta P_1 & \\ & \Delta P_2\end{bmatrix}\begin{bmatrix}\mathbf{s}_1\\ \mathbf{s}_2\end{bmatrix}, \tag{15}$$

where $\Delta P_i = P_i(\theta,\eta) - P_i(\lambda,\mu)$ for $i = 1, 2$. For the computed updates $\widetilde{\mathbf{s}}_1 \perp \mathbf{u}_1$ and $\widetilde{\mathbf{s}}_2 \perp \mathbf{u}_2$ we have

$$Q\begin{bmatrix} P_1(\theta,\eta) & \\ & P_2(\theta,\eta)\end{bmatrix}\begin{bmatrix}\widetilde{\mathbf{s}}_1\\ \widetilde{\mathbf{s}}_2\end{bmatrix} = -\mathbf{r}.$$

We obtain

$$Q\begin{bmatrix} P_1(\theta,\eta) & \\ & P_2(\theta,\eta)\end{bmatrix}\begin{bmatrix}\mathbf{s}_1-\widetilde{\mathbf{s}}_1\\ \mathbf{s}_2-\widetilde{\mathbf{s}}_2\end{bmatrix} = Q\begin{bmatrix}\Delta P_1 & \\ & \Delta P_2\end{bmatrix}\begin{bmatrix}\mathbf{u}_1\\ \mathbf{u}_2\end{bmatrix} + Q\begin{bmatrix}\Delta P_1 & \\ & \Delta P_2\end{bmatrix}\begin{bmatrix}\mathbf{s}_1\\ \mathbf{s}_2\end{bmatrix}. \tag{16}$$

Because of Proposition 6 and Lemma 5, the last term is $\mathcal{O}((\|\mathbf{s}_1\| + \|\mathbf{s}_2\|)^2)$. Similarly, we get

$$\left\| Q\begin{bmatrix}\Delta P_1 & \\ & \Delta P_2\end{bmatrix}\begin{bmatrix}\mathbf{u}_1\\ \mathbf{u}_2\end{bmatrix}\right\| = \mathcal{O}\left(\left(|\theta-\lambda| + |\eta-\mu|\right)^2\right) = \mathcal{O}\big((\|\mathbf{s}_1\| + \|\mathbf{s}_2\|)^2\big).$$

Considering the asymptotic situation, we define

$$U = \begin{bmatrix}\mathbf{x}_1 & \\ & \mathbf{x}_2\end{bmatrix} \quad\text{and}\quad Z = \begin{bmatrix}\dfrac{\partial P_1}{\partial\lambda}(\lambda,\mu)\,\mathbf{x}_1 & \dfrac{\partial P_1}{\partial\mu}(\lambda,\mu)\,\mathbf{x}_1\\ \dfrac{\partial P_2}{\partial\lambda}(\lambda,\mu)\,\mathbf{x}_2 & \dfrac{\partial P_2}{\partial\mu}(\lambda,\mu)\,\mathbf{x}_2\end{bmatrix}.$$

To prove that $\|\mathbf{s}_1 - \widetilde{\mathbf{s}}_1\| + \|\mathbf{s}_2 - \widetilde{\mathbf{s}}_2\| = \mathcal{O}\big((\|\mathbf{s}_1\| + \|\mathbf{s}_2\|)^2\big)$, we need to show that the operator

$$Q\begin{bmatrix} P_1(\lambda,\mu) & \\ & P_2(\lambda,\mu)\end{bmatrix}Q \tag{17}$$

is an invertible operator from $\operatorname{span}(U)^{\perp}$ to $\operatorname{span}(U)^{\perp}$. If this operator maps $\mathbf{w} = \begin{bmatrix}\mathbf{w}_1^T & \mathbf{w}_2^T\end{bmatrix}^T$ $(U^*\mathbf{w} = \mathbf{0})$ to $\mathbf{0}$, then it follows that

$$\begin{bmatrix} P_1(\lambda,\mu)\,\mathbf{w}_1\\ P_2(\lambda,\mu)\,\mathbf{w}_2\end{bmatrix} = Z\begin{bmatrix}\alpha_1\\ \alpha_2\end{bmatrix} \quad\text{for}\quad \alpha_1,\ \alpha_2 \in \mathbb{C}.$$

By left multiplying by matrix

$$V = \begin{bmatrix}\mathbf{y}_1 & \\ & \mathbf{y}_2\end{bmatrix}$$

we obtain

$$V^*Z\begin{bmatrix}\alpha_1\\ \alpha_2\end{bmatrix} = \begin{bmatrix}\mathbf{y}_1^*\dfrac{\partial P_1}{\partial\theta}(\lambda,\mu)\,\mathbf{x}_1 & \mathbf{y}_1^*\dfrac{\partial P_1}{\partial\eta}(\lambda,\mu)\,\mathbf{x}_1\\ \mathbf{y}_2^*\dfrac{\partial P_2}{\partial\theta}(\lambda,\mu)\,\mathbf{x}_2 & \mathbf{y}_2^*\dfrac{\partial P_2}{\partial\eta}(\lambda,\mu)\,\mathbf{x}_2\end{bmatrix}\begin{bmatrix}\alpha_1\\ \alpha_2\end{bmatrix} = \mathbf{0}.$$

Because of Proposition 6, it follows that $\alpha_1 = \alpha_2 = 0$. So, $P_1(\lambda,\mu)\,\mathbf{w}_1 = P_2(\lambda,\mu)\,\mathbf{w}_2 = \mathbf{0}$ and since $\mathbf{w}_1 \perp \mathbf{x}_1,\ \mathbf{w}_2 \perp \mathbf{x}_2$, and (λ,μ) is an algebraically simple eigenvalue, we conclude that $\mathbf{w}_1 = \mathbf{w}_2 = \mathbf{0}$. □

The following proposition proves that the inexact Jacobi–Davidson method typically converges asymptotically linearly. This is often experienced in practice. By "inexact" we mean that the correction equation is solved until the residual norm is reduced by a fixed factor.

Proposition 8. *We assume the same hypotheses as in Proposition 7, only now we solve the correction equation (12) inexactly, such that $\widetilde{\mathbf{s}}_1 \perp \mathbf{u}_1$ and $\widetilde{\mathbf{s}}_2 \perp \mathbf{u}_2$ satisfy*

$$\left\| Q\begin{bmatrix} P_1(\theta,\eta) & \\ & P_2(\theta,\eta)\end{bmatrix}\begin{bmatrix}\widetilde{\mathbf{s}}_1\\ \widetilde{\mathbf{s}}_2\end{bmatrix} + \mathbf{r}\right\| \le \sigma\,\|\mathbf{r}\|$$

for $\sigma < 1$. If $\sigma < \kappa^{-1}$, where κ is the condition number of the operator in (17), seen as an operator from $S = \{\mathbf{z} = \begin{bmatrix}\mathbf{z}_1\\ \mathbf{z}_2\end{bmatrix} : \mathbf{z}_1 \perp \mathbf{x}_1,\ \mathbf{z}_2 \perp \mathbf{x}_2\}$ to S, then the asymptotic convergence of the Jacobi–Davidson method is linear.

Proof. In this case, we have

$$Q\begin{bmatrix} P_1(\theta,\eta) & \\ & P_2(\theta,\eta)\end{bmatrix}\begin{bmatrix}\mathbf{s}_1-\widetilde{\mathbf{s}}_1\\ \mathbf{s}_2-\widetilde{\mathbf{s}}_2\end{bmatrix} = \sigma_1\,\|\mathbf{r}\|\,\mathbf{f} + \mathcal{O}((\|\mathbf{s}_1\| + \|\mathbf{s}_2\|)^2)$$

for a certain vector $\mathbf{f} \in S$, $\|\mathbf{f}\| = 1$; the second-order terms are as in (16). Here σ_1 is the precision used to solve the correction equation and, by assumption, $0 \le \sigma_1 \le \sigma$. From Proposition 7 we know that the operator $Q\ \mathrm{diag}\,(P_1(\theta, \eta), P_2(\theta, \eta))\ Q$ is asymptotically invertible. Moreover, from (15) it is clear that, neglecting higher-order terms,

$$\|\mathbf{r}\| \le \|Q\ \mathrm{diag}\,(P_1(\lambda, \mu), P_2(\lambda, \mu))\ Q\,\| \left\| \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix} \right\|.$$

Therefore,

$$\left\| \begin{bmatrix} \mathbf{s}_1 - \widetilde{\mathbf{s}}_1 \\ \mathbf{s}_2 - \widetilde{\mathbf{s}}_2 \end{bmatrix} \right\| \le \eta\,\kappa \left\| \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix} \right\| + \text{higher-order terms},$$

where κ denotes the condition number of (17). □

Rather than a fixed residual reduction, we take a fixed number of inner iterations to solve the correction equation (12) in the numerical experiments in the following section.

4.4. Preconditioning

If we have approximations $M_1 \approx P_1(\sigma, \tau)$ and $M_2 \approx P_2(\sigma, \tau)$, then we can use them as a preconditioner to solve the correction equations more efficiently. With the notation $M = \mathrm{diag}(M_1, M_2)$, a step of preconditioner amounts to solving

$$QM \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} = \mathbf{b}, \quad \mathbf{y}_1 \perp \mathbf{u}_1,\ \mathbf{y}_2 \perp \mathbf{u}_2,$$

for a given right-hand side $\mathbf{b}$. This means that

$$M \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} = \mathbf{b} + Z \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

for certain α and β which are determined by the orthogonality conditions for $\mathbf{y}_1$ and $\mathbf{y}_2$. From

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} = M^{-1}\mathbf{b} + M^{-1}Z \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad \text{and} \quad U^* \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} = \mathbf{0},$$

one may verify that the solution to this equation is given by

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} = \left(I - M^{-1}Z \left(U^*M^{-1}Z\right)^{-1} U^*M^{-1}\right)\mathbf{b}.$$

Since $M^{-1}Z$ may be precomputed at the beginning of the inner iteration, we need $j+2$ actions with our preconditioner for j inner iterations.

4.5. Pseudocode

In Algorithm 2 we give a pseudocode of the method. In Step 4 we can choose between the different extraction techniques described in Section 4.1, depending on the eigenvalues of interest. Every outer iteration costs one matrix–vector product (MV) with each of the coefficient matrices (see Step 4). In addition, j steps of the inner iteration cost $4j$ MVs and, if applicable, $j+2$ applications of the preconditioner. Not included in this pseudocode, but included in our implementation for the experiments in Section 5, are selection, preconditioning, and restarts. Also, we remark that in Step 9 of the algorithm, we may replace the shift (θ, η) by a given target (λ_T, μ_T), if applicable. This may be sensible in the beginning of the process if the Rayleigh quotient is not yet very accurate. Some practical options are further discussed in the next section.

Algorithm 2: A Jacobi–Davidson type method for the PMEP.

```
    Input: PMEP $P_1(\lambda, \mu)\,\mathbf{x}_1 = \sum_{i=0}^{k}\sum_{j=0}^{k-i} \lambda^i\mu^j A_{ij}\mathbf{x}_1 = \mathbf{0}$, $P_2(\lambda, \mu)\,\mathbf{x}_2 = \sum_{i=0}^{k}\sum_{j=0}^{k-i} B_{ij}\mathbf{x}_2 = \mathbf{0}$,
        starting vectors $\mathbf{u}_1, \mathbf{u}_2$, a target $(\sigma, \tau)$, and a tolerance $\varepsilon$.
    Output: approximate eigenpair $((\theta, \eta), \mathbf{u} \otimes \mathbf{v})$.
1:  Set $\mathbf{s}_1 = \mathbf{u}_1,\ \mathbf{s}_2 = \mathbf{u}_2,\ U_{10} = [\,]$, and $U_{20} = [\,]$.
    for $k = 1, 2, \ldots$
2:    Expand the search subspace. Set $U_{ik} = \mathrm{rgs}(U_{i,k-1}, \mathbf{s}_i)$ for $i = 1, 2$
3:    Update the appropriate test spaces $V_{ik}$ for $i = 1, 2$
4:    Compute $k$th rows and columns of $V_{1k}^*A_{ij}U_{1k}$ and $V_{2k}^*B_{ij}U_{2k}$ for all $A_{ij}$ and $B_{ij}$
5:    Extract a (standard, harmonic, or refined) Ritz pair $((\theta, \eta), \mathbf{c}_1 \otimes \mathbf{c}_2)$
6:    Set $\mathbf{u}_i = U_{ik}\mathbf{c}_i$ for $i = 1, 2$
7:    Compute the residual $\mathbf{r}$ from (0)
8:    Stop if $\|\mathbf{r}\| \le \varepsilon$
9:    Take $Q$ from (11) and solve (approximately) $\mathbf{s}_1$ and $\mathbf{s}_2$ from:
          $Q \begin{bmatrix} P_1(\theta, \eta) & \\ & P_2(\theta, \eta) \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix} = -\mathbf{r}, \quad \mathbf{s}_1 \perp \mathbf{u}_1, \quad \mathbf{s}_2 \perp \mathbf{u}_2.$
```

4.6. Computing several eigenvalues

To compute more than one eigenvalue with Algorithm 2, we use the following approach. For a nonlinear two-parameter eigenvalue problem (13) we introduce a generalized divided difference

$$T[(\lambda_1,\mu_1),(\lambda_2,\mu_2)] = \begin{vmatrix} \lim\limits_{\lambda\to\lambda_2} \dfrac{T_1(\lambda,\mu_1)-T_1(\lambda_1,\mu_1)}{\lambda-\lambda_1} & \lim\limits_{\mu\to\mu_2} \dfrac{T_1(\lambda_2,\mu)-T_1(\lambda_2,\mu_1)}{\mu-\mu_1} \\ \lim\limits_{\lambda\to\lambda_2} \dfrac{T_2(\lambda,\mu_1)-T_2(\lambda_1,\mu_1)}{\lambda-\lambda_1} & \lim\limits_{\mu\to\mu_2} \dfrac{T_2(\lambda_2,\mu)-T_2(\lambda_2,\mu_1)}{\mu-\mu_1} \end{vmatrix}_{\otimes},$$

where $\begin{vmatrix} A & B \\ C & D \end{vmatrix}_{\otimes}$ stands for $A\otimes D - B\otimes C$.

Suppose that all eigenvalues are algebraically simple. Let (λ_1,μ_1) be an eigenvalue of (13) with the corresponding eigenvector $\mathbf{x}_1\otimes\mathbf{x}_2$ and let (λ_2,μ_2) be such an eigenvalue of (13) that $\lambda_2\neq\lambda_1$ and $\mu_2\neq\mu_1$, with the corresponding left eigenvector $\mathbf{y}_1\otimes\mathbf{y}_2$. Then

$$(\mathbf{y}_1\otimes\mathbf{y}_2)^*T[(\lambda_1,\mu_1),(\lambda_2,\mu_2)](\mathbf{x}_1\otimes\mathbf{x}_2) = \frac{1}{(\lambda_2-\lambda_1)(\mu_2-\mu_1)} \begin{vmatrix} \mathbf{y}_1^*T_1(\lambda_2,\mu_1)\,\mathbf{x}_1 & -\mathbf{y}_1^*T_1(\lambda_2,\mu_1)\,\mathbf{x}_1 \\ \mathbf{y}_2^*T_2(\lambda_2,\mu_1)\,\mathbf{x}_2 & -\mathbf{y}_2^*T_2(\lambda_2,\mu_1)\,\mathbf{x}_2 \end{vmatrix} = 0.$$

It is easy to see that the same holds if $\lambda_1=\lambda_2,\ \mu_1\neq\mu_2$ or $\lambda_1\neq\lambda_2,\ \mu_1=\mu_2$.

On the other hand, if $\mathbf{x}_1\otimes\mathbf{x}_2$ and $\mathbf{y}_1\otimes\mathbf{y}_2$ are the right and the left eigenvector of the same eigenvalue (λ_1,μ_1), then it follows by Proposition 6 that

$$(\mathbf{y}_1\otimes\mathbf{y}_2)T[(\lambda_1,\mu_1),(\lambda_1,\mu_1)](\mathbf{x}_1\otimes\mathbf{x}_2) = \begin{vmatrix} \mathbf{y}_1^*\dfrac{\partial T_1}{\partial\lambda}(\lambda_1,\mu_1)\,\mathbf{x}_1 & \mathbf{y}_1^*\dfrac{\partial T_1}{\partial\mu}(\lambda_1,\mu_1)\,\mathbf{x}_1 \\ \mathbf{y}_2^*\dfrac{\partial T_2}{\partial\lambda}(\lambda_1,\mu_1)\,\mathbf{x}_2 & \mathbf{y}_2^*\dfrac{\partial T_2}{\partial\mu}(\lambda_1,\mu_1)\,\mathbf{x}_2 \end{vmatrix} \neq 0.$$

We use the above $T[\cdot,\cdot]$-orthogonality of the left and right eigenvectors to compute more eigenvalues. Suppose that we already have p eigenvalues (λ_i,μ_i) with the corresponding left and right eigenvectors $\mathbf{x}_{1i}\otimes\mathbf{x}_{2i}$ and $\mathbf{y}_{1i}\otimes\mathbf{y}_{2i}$ for $i=1,\ldots,p$. In Step 5 of Algorithm 2 we then consider only Ritz values (σ_1,τ_1) and corresponding vectors $\mathbf{u}_1\otimes\mathbf{u}_2$ that satisfy

$$\max\Big(|(\mathbf{y}_{1i}\otimes\mathbf{y}_{2i})^*T[(\sigma_1,\tau_1),(\lambda_i,\mu_i)]_T(\mathbf{u}_1\otimes\mathbf{u}_2)|\Big) < \zeta \quad \text{for } i=1,\ldots,p, \tag{18}$$

where we take

$$\zeta = \frac{1}{2}\min_{i=1,\ldots,p}\left|(\mathbf{y}_{1i}\otimes\mathbf{y}_{2i})^*T[(\lambda_i,\mu_i),(\lambda_i,\mu_i)](\mathbf{x}_{1i}\otimes\mathbf{x}_{2i})\right|.$$

Remark 9. In the case of the QMEP (19) the generalized divided difference simplifies to

$$T[(\lambda_1,\mu_1),(\lambda_2,\mu_2)] = \begin{vmatrix} A_{10}+(\lambda_1+\lambda_2)A_{20}+\mu_1A_{11} & A_{01}+(\mu_1+\mu_2)A_{02}+\lambda_2A_{11} \\ B_{10}+(\lambda_1+\lambda_2)B_{20}+\mu_1B_{11} & B_{01}+(\mu_1+\mu_2)B_{02}+\lambda_2B_{11} \end{vmatrix}_{\otimes}.$$

5. Numerical experiments

The numerical examples were obtained on a 64-bit Windows version of Matlab R2012b running on Intel 8700 processor and 8 GB of RAM. We compare the following methods:

(a) HJDP, TJDP, and OJDP are the harmonic, two-sided, and one-sided Jacobi–Davidson method, respectively, from Algorithm 2 (that is, applied to the PMEP directly);
(b) HJDL, TJDL, and OJDL are the harmonic, two-sided, and one-sided Jacobi–Davidson method, respectively, applied to the linearization.

In all numerical experiments we use the oblique correction equation, exact preconditioners $M_i = P_i(\sigma,\tau)$ for $i=1,2$, where (σ,τ) is the target, and the starting vectors $[1,\ldots,1]^T$.

Example 10. We consider the QMEP

$$\begin{aligned} Q_1(\lambda,\mu)\,\mathbf{x}_1 &:= (A_{00}+\lambda A_{10}+\mu A_{01}+\lambda^2A_{20}+\lambda\mu A_{11}+\mu^2A_{02})\,\mathbf{x}_1 = \mathbf{0}, \\ Q_2(\lambda,\mu)\,\mathbf{x}_2 &:= (B_{00}+\lambda B_{10}+\mu B_{01}+\lambda^2B_{20}+\lambda\mu B_{11}+\mu^2B_{02})\,\mathbf{x}_2 = \mathbf{0}, \end{aligned} \tag{19}$$

where A_{ij} and B_{ij} are random complex $n\times n$ matrices generated by the Matlab commands

Table 1
Number of outer steps and computational times for variants of the Jacobi–Davidson method applied to a random QMEP with matrices of size 100 × 100.

GMRES	JD on linearization				Algorithm 2			
	TJDL		HJDL		TJDP		HJDP	
	Steps	Time	Steps	Time	Steps	Time	Steps	Time
1	37	6.1	31	4.1	No conv.		200	35.6
2	32	4.8	21	2.4	No conv.		32	5.4
4	19	2.4	23	2.8	52	9.4	19*	3.1
8	15	2.5	15*	2.3	13	2.0	40	6.4
16	19	3.5	31*	4.9	22	3.7	20*	3.4
32	17	5.0	56*	12.5	34	7.4	40*	7.5

[*] The method did not converge to the eigenvalue $(\lambda, \mu) \approx (-0.043255 + 0.046032i, 0.040836 + 0.019801i)$, which we believe is the closest eigenvalue to $(0, 0)$.

```
rand('state', 0); k = 2;
for r = 0:k
    for c=0:(k-r)
        A{r+1, c+1} = rand(n)+i*rand(n);
        B{r+1, c+1} = rand(n)+i*rand(n);
    end
end
```

We explore how the number of GMRES steps influences the convergence in the example with matrices of size $n = 100$, which means that the problem has 40 000 eigenvalues. The goal is the eigenvalue closest to the target $(0, 0)$. In the extraction phase we first pick eigenvalues closest to the target $(0, 0)$ until the norm of the residual is less than $\varepsilon_{\text{change}} = 10^{-2.5}$, then we choose eigenvalues with the minimal residual. We iterate until the norm of the residual is not below $\varepsilon = 10^{-8}$. The remaining parameters are $l_{\max} = 15$ and $l_{\min} = 5$ for TJDL and HJDL, and $l_{\max} = 5$ and $l_{\min} = 2$ for TJDP and HJDP, where $l_{\max}$ is the maximum subspace size before restart and $l_{\min}$ is the subspace size after restart. With these settings the largest Δ-matrices that appear in the projected problem are of size 225 × 225 for all methods. In this example the methods OJDL and OJDP do not perform so well, so their results are not included.

The results are presented in Table 1. For each method we give the number of outer steps and the computational time in seconds. We can see that the optimal solution is to use a moderate number of GMRES steps. If we choose too many GMRES steps, then the convergence becomes erratic. There can be no improvement for many steps and then sudden convergence in couple of steps, often to an unwanted eigenvalue. On the other hand, too few GMRES steps usually results in slow convergence or no convergence at all, but, if the method converges, then we get the desired eigenvalue. Although there is no guarantee that the methods converge to the eigenvalue closest to the target, the two-sided methods seem to be more reliable in this respect.

Example 11. We use the criteria from Section 4.6 for HJDP and TJDP to compute many eigenvalues of the QMEP from Example 10. We apply also HJDL and TJDL with a similar orthogonality selection criteria described in [6]. We use 8 GMRES steps to solve the correction equation approximately. The goal is to compute as many eigenvalues closest to the target $(0, 0)$ in 500 outer iterations. All other settings are the same as in Example 10.

With TJDL we get 3 eigenvalues in 104 s, with HJDL 12 eigenvalues in 90 s, with TJDP 3 eigenvalues in 93 s, and with HJDP 9 eigenvalues in 105 s. OJDL and OJDP find one eigenvalue each in 500 outer iterations. Based on these results and other experiments we suggest to use harmonic versions to compute a small number of eigenvalues close to a given target.

Example 12. To show that the method can be applied to polynomial two-parameter eigenvalue problems of higher order, we consider the cubic two-parameter eigenvalue problem generated in the same way as the quadratic one using $k = 3$ and $n = 200$. The goal is to compute as many eigenvalues close to the target $(0.3, 0.5)$ in 500 outer steps.

Using HJDP and settings $l_{\min} = 2$, $l_{\max} = 4$, and 8 steps of GMRES we get 3 eigenvalues in 585 s. A larger value of $l_{\max}$ is unattractive as the Δ-matrices are of dimension $\left(\frac{1}{2} l_{\max} k(k+1)\right)^2$ and this implies an unreasonably large computational time to solve a projected regular singular problem. With $l_{\max} = 4$, 85% of time is used to solve the projected problem, if we set $l_{\max} = 5$, where the Δ-matrices are of size 900 × 900, this goes to 95%. The large size of the Δ-matrices of the projected problem is an obstacle for practical use of methods based on Algorithm 2 on polynomials of degree higher than three.

If we apply a Jacobi–Davidson type method on the linearization, we do not have such problems with the high degree. With HJDL and settings $l_{\min} = 5$ and $l_{\max} = 15$ we get 11 eigenvalues in 406 s. In this case 16% of the computational time is spent for solving the projected problem and 66% for matrix–vector multiplications as the matrices and vectors of the linearization are of size 1200 (compared to 200 in HJDP). We remark that the problem has 360 000 eigenvalues and we have no other method to verify if the computed eigenvalues are indeed the closest ones to the target. Based on the obtained results we believe that the two closest eigenvalues to the target are

$$(\lambda_1, \mu_1) \approx (0.31173 + 0.044865i, 0.46771 + 0.014214i),$$
$$(\lambda_2, \mu_2) \approx (0.29049 + 0.035873i, 0.44667 - 0.035292i).$$

Table 2
Critical delay $\tau = \tau_1 = \frac{1}{2}\tau_2$ for Eq. (20) computed by solving the PMEP associated to the discretized equation.

n	τ	Steps	Time
25	0.2013409	7	2.0
50	0.2013679	10	2.6
100	0.2013749	11	3.4
200	0.2013767	21	9.1
400	0.2013772	61	52.5

Example 13. We consider the delayed partial differential equation with two delays from [16]

$$u_t = u_{xx} + a_0(x)u + a_1(x)u(x, t - \tau_1) + a_2(x)u(x, t - \tau_2), \qquad u(0, t) = u(\pi, t) = 0, \tag{20}$$

where $a_0(x) = 2 + 0.3\sin(x)$, $a_1(x) = -2 + 0.2x(1 - e^{x-\pi})$, and $a_2(x) = -2 - 0.3x(\pi - x)$. Discretization of the equation with a finite difference scheme leads to the characteristic equation

$$(-i\omega I + A_0 + A_1 e^{-i\tau_1\omega} + A_2 e^{-i\tau_2\omega})\mathbf{v} = \mathbf{0}. \tag{21}$$

We are looking for the critical delay under the assumption that the delays are commensurate as $\tau_2 = 2\tau_1$. We write $\lambda = i\omega$ and $\mu = e^{-\tau_1\lambda}$. From (21) and its conjugate, where we know that in the critical point we have the $\overline{\lambda} = -\lambda$ and $\overline{\mu} = 1/\mu$, we get a PMEP

$$\begin{aligned} &(A_0 - \lambda I + \mu A_1 + \mu^2 A_2)\,\mathbf{x} = \mathbf{0},\\ &(A_1 + \mu A_0 + \mu^2 A_2 + \lambda\mu^2 I)\,\mathbf{y} = \mathbf{0}, \end{aligned} \tag{22}$$

which is composed of a quadratic and a cubic polynomial. On this PMEP we apply TJDP, the two-sided version of Algorithm 2. As we want to obtain the critical delay, we adjust the selection criteria so that in each step we pick the Ritz value (θ, η) such that θ is close to the positive part of the imaginary axis, η is close to the unit circle, and the corresponding $\tau = -\log(\eta)/\theta$ is close to be real positive and small as possible. In the selection for our experiments we used the function

$$g(\theta, \eta, \tau) = |\tau| \cdot (1 + |\text{ang}(\tau)|) + |\text{ang}(i\lambda)| + |1 - |\eta||,$$

where $\text{ang}(z) \in (-\pi, \pi)$ stands for the polar angle of a complex number z.

In each step of the Jacobi–Davidson method we have to solve a projected problem of the form (22). We linearize it as

$$\left(\begin{bmatrix} A_0 & A_1 \\ 0 & -I \end{bmatrix} + \lambda \begin{bmatrix} -I & 0 \\ 0 & 0 \end{bmatrix} + \mu \begin{bmatrix} 0 & A_2 \\ I & 0 \end{bmatrix}\right) \begin{bmatrix} \mathbf{x} \\ \mu\mathbf{x} \end{bmatrix} = \mathbf{0}$$

$$\left(\begin{bmatrix} A_2 & A_1 & 0 \\ 0 & -I & 0 \\ 0 & 0 & -I \end{bmatrix} + \lambda \begin{bmatrix} 0 & 0 & I \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \mu \begin{bmatrix} 0 & A_0 & 0 \\ I & 0 & 0 \\ 0 & I & 0 \end{bmatrix}\right) \begin{bmatrix} \mathbf{y} \\ \mu\mathbf{y} \\ \mu^2\mathbf{y} \end{bmatrix} = \mathbf{0},$$

which is more efficient than the generic linearizations for the quadratic and the cubic polynomial. An alternative for the solution of the projected problem is to linearize (22) as a quartic matrix polynomial with a linearization proposed in [1].

With the above approach we are able to compute the critical delay for much larger matrices as reported in [16]. The obtained values together with the number of outer steps and computational times are reported in Table 2, where n stands for the size of the matrices A_0, A_1, and A_2. We use $l_{\min} = 3$, $l_{\max} = 7$, and 25 GMRES steps for all n.

Example 14. We would like to compute few roots of a bivariate scalar polynomial system (7) where both scalar polynomials $p_1(\lambda, \mu)$ and $p_2(\lambda, \mu)$ are of degree $k = 50$ with random complex coefficients generated in Matlab with the code from Example 10 using $k = 50$ and $n = 1$.

The bivariate system is linearized as a regular singular two-parameter eigenvalue problem, where the matrices of size $\frac{1}{2}k(k+1)$ are represented as sparse. We get better results by using small search spaces and small number of GMRES steps. Using HJDL with the target $(0, 0)$, $l_{\min} = 3$, $l_{\max} = 6$, $\varepsilon_{\text{change}} = 10^{-1.5}$, $\varepsilon = 10^{-5}$, 6 GMRES steps, and 2000 outer steps, we get 10 zeros in 53 s. For comparison we computed all zeros using Matlab interface PHClab [17] for PHCpack [18], which finds 2462 zeros in 653 s. If we order all zeros by their distance to the target starting with the closest one, then the closest computed eigenvalues with HJDL have indices 1, 3, 6, 7, and 11. We have to admit that the approach is quite sensitive to the changes in the parameters, for instance, if we slightly increase or decrease the number of GMRES steps or the size of the search space, we get completely different set of zeros which might even not include the closest zero to the target.

For computing all solutions, this approach is not so efficient as some special algorithms. For instance, the algorithm [19] from chebfun2 [20] is able to compute all real solutions contained in $[-1, 1] \times [-1, 1]$ very efficiently, but it does not compute complex solutions. The best methods at the moment that compute all solutions use continuation method, such as PHCpack. Their drawback is that they cannot compute only the solutions that are close to a given target. Although the presented approach can be used to compute only solutions close to a target, to make it really useful, we have to make it more robust and find alternative linearizations of bivariate polynomials with smaller matrices.

Acknowledgments

This research was performed in part while the third author was visiting the CASA group at the TU Eindhoven. The author wishes to thank the NWO for the visitor's grant and the CASA group for the hospitality.

References

[1] E. Jarlebring, M.E. Hochstenbach, Polynomial two-parameter eigenvalue problems and matrix pencil methods for stability of delay-differential equations, Linear Algebra Appl. 431 (2009) 369–380.
[2] K. Meerbergen, C. Schröder, H. Voss, A Jacobi–Davidson method for two-real-parameter nonlinear eigenvalue problems arising from delay-differential equations, Numer. Linear Algebra Appl. 20 (2013) 852–868.
[3] M.E. Hochstenbach, A. Muhič, B. Plestenjak, On linearizations of the quadratic two-parameter eigenvalue problems, Linear Algebra Appl. 436 (2012) 2725–2743.
[4] A. Muhič, B. Plestenjak, On the quadratic two-parameter eigenvalue problem and its linearization, Linear Algebra Appl. 432 (2010) 2529–2542.
[5] F.V. Atkinson, Multiparameter Eigenvalue Problems, Academic Press, New York, 1972.
[6] M.E. Hochstenbach, T. Košir, B. Plestenjak, A Jacobi–Davidson type method for the nonsingular two-parameter eigenvalue problem, SIAM J. Matrix Anal. Appl. 26 (2005) 477–497.
[7] M.E. Hochstenbach, B. Plestenjak, Harmonic Rayleigh–Ritz for the multiparameter eigenvalue problem, Electron. Trans. Numer. Anal. 29 (2008) 81–96.
[8] M.E. Hochstenbach, B. Plestenjak, A Jacobi–Davidson type method for a right definite two-parameter eigenvalue problem, SIAM J. Matrix Anal. Appl. 24 (2002) 392–410.
[9] A. Muhič, B. Plestenjak, On the singular two-parameter eigenvalue problem, Electron. J. Linear Algebra 18 (2009) 420–437.
[10] P. van Dooren, The computation of Kronecker's canonical form of a singular pencil, Linear Algebra Appl. 27 (1979) 103–141.
[11] M.E. Hochstenbach, G.L.G. Sleijpen, Harmonic and refined Rayleigh–Ritz for the polynomial eigenvalue problem, Numer. Linear Algebra Appl. 15 (2008) 35–54.
[12] MATLAB, The MathWorks, Inc.., Natick, Massachusetts, United States.
[13] P. Dreesen, K. Batselier, B. De Moor, Back to the roots: Polynomial system solving, linear algebra, systems theory, in: Proceedings of the 16th IFAC Symposium on System Identification, Brussels, Belgium, 2012, pp. 1203–1208.
[14] H.J. Stetter, Numerical Polynomial Algebra, SIAM, Philadelphia, 2004.
[15] G.W. Stewart, Matrix Algorithms Volume 2: Eigensystems, SIAM, 2001.
[16] E. Jarlebring, The spectrum of delay-differential equations: numerical methods, stability and perturbation (Ph.D. thesis), TU Braunschweig, 2008.
[17] Y. Guan, J. Verschelde, PHClab: A MATLAB/Octave interface to PHCpack, in: M. Stillman, J. Verschelde, N. Takayama (Eds.), Software for Algebraic Geometry, in: The IMA Volumes in Mathematics and its Applications, vol. 148, Springer, New York, 2008, pp. 15–32.
[18] J. Verschelde, Algorithm 795: PHCpack: a general-purpose solver for polynomial systems by homotopy continuation, ACM Trans. Math. Software 25 (1999) 251–276.
[19] Y. Nakatsukasa, V. Noferini, A. Townsend, Computing the common zeros of two bivariate functions via Bézout resultants, Numer. Math. 129 (2015) 181–209.
[20] A. Townsend, L.N. Trefethen, An extension of Chebfun to two dimensions, SIAM J. Sci. Comput. 35 (2013) 495–518.