A JACOBI–DAVIDSON TYPE METHOD FOR THE TWO-PARAMETER EIGENVALUE PROBLEM*

MICHIEL E. HOCHSTENBACH†, TOMAŽ KOŠIR‡ AND BOR PLESTENJAK‡

Abstract. We present a new numerical method for computing selected eigenvalues and eigenvectors of the two-parameter eigenvalue problem. The method does not require good initial approximations and is able to tackle large problems that are too expensive for methods that compute all eigenvalues. The new method uses a two-sided approach and is a generalization of the Jacobi–Davidson type method for the right definite two-parameter eigenvalue problems (M. E. Hochstenbach and B. Plestenjak, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 392–410). Here we consider the much wider class of nonsingular problems. In each step we first compute Petrov triples of a small projected two-parameter eigenvalue problem and then expand the left and right search spaces using approximate solutions of appropriate correction equations. Using a selection technique it is possible to compute more than one eigenpair. Some numerical examples are presented.

Key words. Two-parameter eigenvalue problem, subspace method, Jacobi–Davidson method, correction equation, Petrov–Galerkin, two-sided approach.

AMS subject classifications. 65F15, 15A18, 15A69.

1. Introduction. We are interested in computing one or more eigenpairs of the *two-parameter eigenvalue problem*

$$\begin{aligned} A_1x_1 &= \lambda B_1x_1 + \mu C_1x_1, \\ A_2x_2 &= \lambda B_2x_2 + \mu C_2x_2, \end{aligned} \tag{1.1}$$

where A_i, B_i, and C_i are given $n_i \times n_i$ matrices over $\mathbb{C}$, $\lambda, \mu \in \mathbb{C}$ and $x_i \in \mathbb{C}^{n_i}$ for $i = 1, 2$. A pair (λ, μ) is called an *eigenvalue* if it satisfies (1.1) for nonzero vectors x_1, x_2. The tensor product $x_1 \otimes x_2$ is then the corresponding *right eigenvector*. Similarly, $y_1 \otimes y_2$ is the corresponding *left eigenvector* if $0 \neq y_i \in \mathbb{C}^{n_i}$ and $y_i^*(A_i - \lambda B_i - \mu C_i) = 0$ for $i = 1, 2$.

Multiparameter eigenvalue problems of this kind arise in a variety of applications [1], particularly in mathematical physics when the method of separation of variables is used to solve boundary value problems [23]. When the separation constants cannot be decoupled, two-parameter Sturm–Liouville problems of the form

$$-(p_i(x_i)y_i'(x_i))' + q_i(x_i)y_i(x_i) = (\lambda a_{i1}(x_i) + \mu a_{i2}(x_i))y_i(x_i), \tag{1.2}$$

where $x_i \in [a_i, b_i]$, with boundary conditions

$$\begin{aligned} y_i(a_i)\cos\alpha_i - y_i'(a_i)\sin\alpha_i &= 0, \qquad 0 \le \alpha_i \le \pi, \\ y_i(b_i)\cos\beta_i - y_i'(b_i)\sin\beta_i &= 0, \qquad 0 \le \beta_i \le \pi, \end{aligned}$$

can arise, where $\alpha_i \in [0, \pi)$, $\beta_i \in (0, \pi]$ and $p_i', q_i, a_{i1}, a_{i2}$ are real valued and continuous, for $i = 1, 2$. Using discretization the problem (1.2) can be converted into the form (1.1). As an example, let us consider the equation $\Delta u + k^2u = 0$ in $\mathbb{R}^2$ that represents the vibration of a

*Version: November 20, 2003.

†Department of Mathematics, Case Western Reserve University, 10900 Euclid Avenue, Cleveland, Ohio 44106-7058 (hochsten@case.edu). Part of this research was done when the author was employed by Utrecht University.

‡Department of Mathematics, University of Ljubljana, Jadranska 19, SI-1000 Ljubljana, Slovenia (tomaz.kosir@fmf.uni-lj.si, bor.plestenjak@fmf.uni-lj.si). These authors' research was supported in part by the Ministry of Education, Science, and Sport of Slovenia (Research Projects Z1-3136 and PO-0508).

fixed membrane [14]. In a rectangular membrane the separation of variables leads to two Sturm–Liouville equations that can be solved independently. In a circular membrane the two equations (the angular and the radial) form a triangular situation and cannot be solved independently. We can solve them one by one by inserting the parameter from the solution of the angular equation into the radial equation. In an elliptic membrane the separation leads to the Mathieu and the modified Mathieu equation (see e.g. [23])

$$\begin{aligned} y_1''(x_1) + (2\lambda \cosh 2x_1 - \mu) y_1(x_1) &= 0, \\ y_2''(x_2) - (2\lambda \cos 2x_2 - \mu) y_2(x_2) &= 0, \end{aligned} \tag{1.3}$$

that have to be solved simultaneously and thus form a genuine two-parameter eigenvalue problem.

Another problem that can be cast in the form (1.1) is the three-point boundary problem [7]. A typical problem is

$$-(p(x)y'(x))' + q(x)y(x) = (\lambda r(x) + \mu s(x))y(x), \tag{1.4}$$

subject to $y(a) = y(b) = y(c) = 0$, where $a < b < c$. We can treat (1.4) as a two-parameter eigenvalue problem

$$-(p(x_i)y_i'(x_i))' + q(x_i)y_i(x_i) = (\lambda r(x_i) + \mu s(x_i))y_i(x_i)$$

for $i = 1, 2$, where $x_1 \in [a, b]$, $x_2 \in [b, c]$, and the boundary conditions are $y_1(a) = y_1(b) = y_2(b) = y_2(c) = 0$. An example (see [23] for details) is Lamé's equation

$$y''(x) + \frac{1}{2}\left(\frac{1}{x-a} + \frac{1}{x-b} + \frac{1}{x-c}\right) y'(x) + \frac{\lambda + \mu x}{(x-a)(x-b)(x-c)} y(x) = 0,$$

which arises in the solution of Laplace's equation in elliptic coordinates.

Two-parameter problems appear in the algebraic form (1.1) as well. In [16], it is shown that the optimum value of the over-relaxation parameter ω in the Young–Frankel scheme for a separable elliptic partial differential equation in two independent variables can be obtained from the eigenvalues of a certain two-parameter eigenvalue problem. In [15], algorithms for the estimation of material electrical properties from measurements of interdigital dielectrometry sensors are discussed. When the sensors are applied to the material that is composed of two layers, the properties of the individual layers are the eigenvalues of the appropriate two-parameter eigenvalue problem. Yet another example is the dynamic model updating [6]. Suppose that we have a spring-mass model where the mass matrix is known and the stiffness parameter values of two springs have to be updated based on the outside measurements of the natural frequencies. The updated parameters are the eigenvalues of a two-parameter problem. The above examples show the need for numerical solvers of problem (1.1).

Two-parameter problems can be expressed as two coupled generalized eigenvalue problems as follows. On the tensor product space $S := \mathbb{C}^{n_1} \otimes \mathbb{C}^{n_2}$ of the dimension $N := n_1 n_2$ we define

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

(for details on the tensor product and relation to the multiparameter eigenvalue problem, see, for example, [2]). We assume that the two-parameter problem (1.1) is *nonsingular*, that is the corresponding operator determinant Δ_0 is invertible. In this case $\Gamma_1 := \Delta_0^{-1}\Delta_1$ and $\Gamma_2 := \Delta_0^{-1}\Delta_2$ commute and problem (1.1) is equivalent to the associated problem

$$\begin{aligned} \Delta_1 z &= \lambda \Delta_0 z, \\ \Delta_2 z &= \mu \Delta_0 z \end{aligned} \tag{1.5}$$

for decomposable tensors $z \in S$, $z = x \otimes y$ (see [2]). The left and right eigenvectors of (1.1) are Δ_0-orthogonal; i.e., if $x_1 \otimes x_2$ and $y_1 \otimes y_2$ are right and left eigenvector of (1.1), respectively, corresponding to distinct eigenvalues, then

$$(y_1 \otimes y_2)^* \Delta_0 (x_1 \otimes x_2) = \begin{vmatrix} y_1^* B_1 x_1 & y_1^* C_1 x_1 \\ y_2^* B_2 x_2 & y_2^* C_2 x_2 \end{vmatrix} = 0.$$

If (λ, μ) is an eigenvalue of (1.1) then

$$\dim \left(\bigcap_{\substack{i_1+i_2=N \\ i_1,i_2 \geq 0}} \mathrm{Ker} \left[(\Gamma_1 - \lambda I)^{i_1} (\Gamma_2 - \mu I)^{i_2} \right] \right)$$

is the *algebraic multiplicity* of (λ, μ). We say that (λ, μ) is *algebraically simple* when its algebraic multiplicity is one.

The following lemma is a consequence of Lemma 3 in [9].

LEMMA 1.1. *If $\boldsymbol{\lambda}$ is an algebraically simple eigenvalue of the two-parameter eigenvalue problem (1.1) and $x_1 \otimes x_2$ and $y_1 \otimes y_2$ are the corresponding right and left eigenvector, respectively, then the matrix*

$$\begin{bmatrix} y_1^* B_1 x_1 & y_1^* C_1 x_1 \\ y_2^* B_2 x_2 & y_2^* C_2 x_2 \end{bmatrix}$$

is nonsingular.

There exist some numerical methods for two-parameter eigenvalue problems. Most of them require that the problem is real and *right definite*, i.e., that all matrices A_i, B_i, and C_i are real symmetric and that Δ_0 is positive definite, and as a consequence, eigenvalues and eigenvectors are real. Most of the presented two-parameter problems are right definite, for instance (1.3) and the one in [16], but not all, for instance the one in [15] where the eigenvalues are complex. It is the aim of this paper to introduce an algorithm for such non right definite two-parameter eigenvalue problems.

One of the algorithms (also usable for large sparse matrices) for the right definite two-parameter problem is a Jacobi–Davidson type method [11] and ideas from this method are generalized in this paper to handle all nonsingular two-parameter eigenvalue problems.

One possible approach to solve (1.1) is to solve the associated couple of generalized problems (1.5). In the right definite case this can be achieved by numerical methods for simultaneous diagonalization of commutative symmetric matrices [13, 20, 5], while an algorithm for the general nonsingular case using QZ algorithm is presented in this paper (see Algorithm 2.3). Solving the problem via the associated problem is only feasible for problems of low dimension as the size of the matrices of the associated problem is $N \times N$.

Another method that can be used for non right definite two-parameter problems of moderate size is Newton's method [4], which has the deficiency that it requires initial approximations close enough to the solution in order to avoid misconvergence. The continuation method [18] can be used for *weakly elliptic* problems, i.e., such that A_i, B_i and C_i are real symmetric and one of

B_i, C_i is positive definite. We mention that right definite two-parameter problems are also weakly elliptic [17, Lemma 2.1].

In this paper, we introduce a new Jacobi–Davidson type method that can be used to compute selected eigenpairs for nonsingular problems. The method works even without close initial approximations and is suitable for large sparse matrices. Our method computes the eigenvalue (λ, μ) of (1.1), which is closest to a given target $(\lambda_{\mathrm{T}}, \mu_{\mathrm{T}}) \in \mathbb{C}^2$, i.e., the one with minimum $|\lambda - \lambda_{\mathrm{T}}|^2 + |\mu - \mu_{\mathrm{T}}|^2$.

The outline of the paper is as follows. In Section 2, we present a new algorithm for the computation of eigenpairs using the associated problem. This method is only suitable for matrices of moderate size, so we combine it with a subspace method. We generalize the Petrov–Galerkin approach to two-parameter eigenvalue problems in Section 3. In Section 4, we present a two-sided Jacobi–Davidson type method for two-parameter eigenvalue problems. Several possible correction equations are discussed in Section 5. In Section 6, we present a selection technique that allows the computation of more than one eigenpair. The time complexity is given in Section 7, and some numerical examples are presented in Section 8. Conclusions are summarized in Section 9.

2. Algorithm based on the associated problem. We propose the following method to solve the associated problem (1.5). First we compute a QZ decomposition (generalized Schur form) of the matrix pencil (Δ_1, Δ_0). We obtain unitary matrices Q and Z such that $Q^*\Delta_0 Z = R$ and $Q^*\Delta_1 Z = S$ are upper triangular. Since Δ_0 is nonsingular the same is true for R. From

$$\Delta_0^{-1}\Delta_1 = ZR^{-1}SZ^*$$

it follows that the eigenvalues of the first generalized eigenvalue problem in (1.5) are the quotients s_{ii}/r_{ii} of the diagonal elements of matrices S and R.

Next, we sort the generalized Schur form so that multiple eigenvalues of the first generalized eigenvalue problem in (1.5) appear in blocks (see for instance [22]). Let us assume that the generalized Schur form is sorted to meet this requirement and let matrix $R^{-1}S$ be partitioned accordingly as

$$R^{-1}S = \begin{bmatrix} L_{11} & L_{12} & \cdots & L_{1p} \\ 0 & L_{22} & \cdots & L_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & L_{pp} \end{bmatrix}. \tag{2.1}$$

In the above partition, multiple eigenvalues of $\Delta_0^{-1}\Delta_1$ are clustered in upper triangular matrices $L_{11}, \ldots, L_{pp}$ along the diagonal so that $\lambda(L_{ii}) \neq \lambda(L_{jj})$ for $i \neq j$, where $\lambda(L_{kk})$ is the eigenvalue of a block L_{kk}. Let us denote the size of L_{ii} by m_i for $i = 1, \ldots, p$.

LEMMA 2.1. *Let*

$$L = \begin{bmatrix} L_{11} & L_{12} & \cdots & L_{1p} \\ 0 & L_{22} & \cdots & L_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & L_{pp} \end{bmatrix}$$

be a partitioning of a block upper triangular matrix L such that $\Lambda(L_{11}), \ldots, \Lambda(L_{pp})$ are mutually disjoint, where $\Lambda(L_{kk})$ is the set of eigenvalues of L_{kk}. If M commutes with L then M is block upper triangular partitioned conformally with L.

Proof. First we study the case $p = 2$. Let M be partitioned conformally with L as

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}.$$

From $LM - ML = 0$ and the above assumption we obtain the equation $L_{22}M_{21} - M_{21}L_{11} = 0$. Because L_{11} and L_{22} have no eigenvalues in common, this is a nonsingular homogeneous Sylvester equation for M_{21} (see for example [21, p. 223]). Therefore, the unique solution is $M_{21} = 0$.

In case $p > 2$ one can see that M is block upper triangular by applying the above argument on all appropriate 2×2 block partitions of L and M. □

LEMMA 2.2. *$T = Q^*\Delta_2 Z$ partitioned conformally with (2.1) is block upper triangular.*

Proof. As $\Delta_0^{-1}\Delta_1$ and $\Delta_0^{-1}\Delta_2$ commute, so do $R^{-1}S$ and $R^{-1}T$. It follows from Lemma 2.1 that $R^{-1}T$ is block upper triangular partitioned conformally to (2.1). As block upper triangular matrices keep their shape when multiplied by a triangular matrix, it follows from $T = R(R^{-1}T)$ that T is block upper triangular as well. □

Once R, S and T are partitioned conformally with (2.1) as

$$R = \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1p} \\ 0 & R_{22} & \cdots & R_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_{pp} \end{bmatrix}, \quad S = \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1p} \\ 0 & S_{22} & \cdots & S_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & S_{pp} \end{bmatrix}, \quad T = \begin{bmatrix} T_{11} & T_{12} & \cdots & T_{1p} \\ 0 & T_{22} & \cdots & T_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & T_{pp} \end{bmatrix},$$

it is straightforward to compute eigenvalues of (1.1). To each diagonal block L_{ii} of size m_i in $R^{-1}S$ correspond m_i eigenvalues $(\lambda_i, \mu_{i1}), \ldots, (\lambda_i, \mu_{im_i})$, where λ_i is the eigenvalue of L_{ii} and $\mu_{i1}, \ldots, \mu_{im_i}$ are eigenvalues of the generalized eigenvalue problem $T_{ii}w = \mu R_{ii} w$.

Now that we have all eigenvalues (λ_j, μ_j), $j = 1, \ldots, N$, of (1.1) we compute the corresponding eigenvectors $x_{j1} \otimes x_{j2}$. We do this by solving $(A_i - \lambda_j B_i - \mu_j C_i)x_{ji} = 0$, where x_{ji} is normalized, for $i = 1, 2$. In a similar way we can obtain left eigenvectors $y_{j1} \otimes y_{j2}$ when they are required.

The complete procedure is summarized in Algorithm 2.3.

ALGORITHM 2.3. An algorithm for the nonsingular two-parameter eigenvalue problem (1.1).

1. Compute Δ_0, Δ_1 and Δ_2 of the associated problem (1.5).
2. Compute a generalized Schur decomposition $Q^*\Delta_0 Z = R$ and $Q^*\Delta_1 Z = S$, such that Q, Z are unitary, R and S are upper triangular, and the Schur form is sorted so that multiple values of $\lambda_i := s_{ii}/r_{ii}$ are clustered along the diagonal of $R^{-1}S$. As a result of this, R and S are partitioned as

$$R = \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1p} \\ 0 & R_{22} & \cdots & R_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_{pp} \end{bmatrix}, \quad S = \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1p} \\ 0 & S_{22} & \cdots & S_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & S_{pp} \end{bmatrix},$$

   where the size of R_{ii} and S_{ii} is m_i and $m_1 + \cdots + m_p = N$.
3. Compute diagonal blocks $T_{11}, \ldots, T_{pp}$ of $T = Q^*\Delta_2 Z$, partitioned conformally with R and S.
4. Compute the eigenvalues $\mu_{i1}, \ldots, \mu_{im_i}$ of the generalized eigenvalue problem

$$T_{ii}w = \mu R_{ii}w$$

   for $i = 1, \ldots, p$.

5. The eigenvalues of (1.1) are

$$(\lambda_1, \mu_{11}), \ldots, (\lambda_1, \mu_{1m_1}); \ldots; (\lambda_p, \mu_{p1}), \ldots, (\lambda_p, \mu_{pm_p}),$$

reindex them as $(\lambda_1, \mu_1), \ldots, (\lambda_N, \mu_N)$.

6. For each eigenvalue (λ_j, μ_j), $j = 1, \ldots, N$, of (1.1) take for x_{ji} and y_{ji} the smallest right and the smallest left singular vector of $A_i - \lambda_j B_i - \mu_j C_i$, respectively, for $i = 1, 2$ (see Remark 2.5).

REMARK 2.4. *In numerical computation we may cluster not only multiple eigenvalues but also close eigenvalues of $R^{-1}S$. After clustering we take the mean of all eigenvalues in the cluster of size m_i as a multiple eigenvalue of order m_i. This means that we take λ_i as a mean of all eigenvalues of the generalized eigenvalue problem*

$$S_{ii} w = \lambda R_{ii} w$$

for $i = 1, \ldots, p$.

REMARK 2.5. *In practice there will be an error in a detected eigenvalue (λ_j, μ_j). Because of that we take the right singular vector corresponding to the smallest singular value to find the normalized x_{ji} such that $(A_i - \lambda_j B_i - \mu_j C_i) x_{ji} = 0$ for $i = 1, 2$. In a similar way we get the approximation to the left eigenvector.*

Let as assume that A_i, B_i, C_i are dense and that $n_1 = n_2 = n$. The time complexity of Algorithm 2.3 is $\mathcal{O}(n^6)$ for the computation of eigenvalues using QZ decomposition of matrices of size n^2. The maximum additional work for eigenvectors is $\mathcal{O}(n^5)$ as we have to compute $\mathcal{O}(n^2)$ singular value decompositions of matrices of size n. If we are not interested in all eigenvectors (as is often the case for large sparse matrices) then the additional work can be substantially less.

Large time complexity is the reason that Algorithm 2.3 is useful only for matrices of a modest size. For larger problems we embed this method in a subspace method and use Algorithm 2.3 for the small projected problems.

3. Subspace methods and Petrov triples. In this section we study subspace methods for the two-parameter eigenvalue problem. In a subspace method we start with a given search subspace from which approximations to eigenpairs are computed (*extraction*). In the extraction we usually have to solve a smaller eigenvalue problem of the same type as the original one. After each step we expand the subspace by a new direction (*expansion*) and as the search subspace grows, the eigenpair approximations will converge to an eigenpair of the original problem. In this section we discuss the extraction and in the next section we discuss the algorithm and the expansion.

Suppose that we have k-dimensional search spaces $\mathcal{U}_{ik} \subset \mathbb{C}^{n_i}$ and k-dimensional test spaces $\mathcal{V}_{ik} \subset \mathbb{C}^{n_i}$ for $i = 1, 2$. Let the columns of the $n_i \times k$ matrices U_{ik} and V_{ik} form orthogonal bases for $\mathcal{U}_{ik}$ and $\mathcal{V}_{ik}$, respectively, for $i = 1, 2$. The Petrov–Galerkin conditions

$$\begin{aligned}(A_1 - \sigma B_1 - \tau C_1) u_1 &\perp \mathcal{V}_{1k},\\ (A_2 - \sigma B_2 - \tau C_2) u_2 &\perp \mathcal{V}_{2k},\end{aligned}$$

where $u_i \in \mathcal{U}_{ik} \backslash \{0\}$ for $i = 1, 2$, lead to the smaller projected two-parameter problem

$$\begin{aligned}V_{1k}^* A_1 U_{1k} c_1 &= \sigma V_{1k}^* B_1 U_{1k} c_1 + \tau V_{1k}^* C_1 U_{1k} c_1,\\ V_{2k}^* A_2 U_{2k} c_2 &= \sigma V_{2k}^* B_2 U_{2k} c_2 + \tau V_{2k}^* C_2 U_{2k} c_2,\end{aligned} \tag{3.1}$$

where $u_i = U_{ik} c_i \neq 0$ for $i = 1, 2$ and $\sigma, \tau \in \mathbb{C}$.

We say that an eigenvalue (σ,τ) of (3.1) is a *Petrov value* for the two-parameter eigenvalue problem (1.1) with respect to the search spaces $\mathcal{U}_{1k}$ and $\mathcal{U}_{2k}$ and test spaces $\mathcal{V}_{1k}$ and $\mathcal{V}_{2k}$. If (σ,τ) is an eigenvalue of (3.1) and $c_1\otimes c_2$ is the corresponding right eigenvector, then $u_1\otimes u_2$ is a *right Petrov vector*. Similarly, if $d_1\otimes d_2$ is the corresponding left eigenvector of (3.1) then $v_1\otimes v_2$ is a *left Petrov vector*, where $v_i=V_{ik}d_i$ for $i=1,2$. It is easy to check that σ and τ are equal to the *two-sided tensor Rayleigh quotients*

$$\begin{aligned}\sigma=\rho_1(u,v)&=\frac{(v_1\otimes v_2)^*\Delta_1(u_1\otimes u_2)}{(v_1\otimes v_2)^*\Delta_0(u_1\otimes u_2)}=\frac{(v_1^*A_1u_1)(v_2^*C_2u_2)-(v_1^*C_1u_1)(v_2^*A_2u_2)}{(v_1^*B_1u_1)(v_2^*C_2u_2)-(v_1^*C_1u_1)(v_2^*B_2u_2)},\\ \tau=\rho_2(u,v)&=\frac{(v_1\otimes v_2)^*\Delta_2(u_1\otimes u_2)}{(v_1\otimes v_2)^*\Delta_0(u_1\otimes u_2)}=\frac{(v_1^*B_1u_1)(v_2^*A_2u_2)-(v_1^*A_1u_1)(v_2^*B_2u_2)}{(v_1^*B_1u_1)(v_2^*C_2u_2)-(v_1^*C_1u_1)(v_2^*B_2u_2)}.\end{aligned}\tag{3.2}$$

In order to obtain Petrov values, we have to solve small two-parameter eigenvalue problems. For this purpose, we use Algorithm 2.3. Altogether, we obtain k^2 *Petrov triples* $((\sigma_j,\tau_j),u_{j1}\otimes u_{j2},v_{j1}\otimes v_{j2})$ that are approximations to eigentriples $((\lambda_j,\mu_j),x_{j1}\otimes x_{j2},y_{j1}\otimes y_{j2})$ of (1.1) for $j=1,\ldots,k^2$.

4. Jacobi–Davidson type method. The Jacobi–Davidson method [19] is one of the subspace methods that may be used for the numerical solution of one-parameter eigenvalue problems. For an overview of subspace methods see, for example, [3]. In the Jacobi–Davidson method approximate solutions to certain correction equations are used to expand the search space. The search for a new direction is restricted to the subspace that is orthogonal or oblique to the last chosen right (or left) Petrov vector.

A Jacobi–Davidson type method has been successfully applied to the right definite two-parameter eigenvalue problem [11]. The method in [11] is one-sided, which means that the search spaces $\mathcal{V}_i$ in (3.1) are the same as the test spaces $\mathcal{U}_i$. When we tested the one-sided method from [11] on non right definite problems, it turned out that the performance was sometimes not optimal, in particular there were problems with convergence to unwanted eigenvalues or no convergence at all. Therefore we generalize the two-sided Jacobi–Davidson method [10] to two-parameter eigenvalue problems. The idea is to take $\mathcal{U}_i$ as search spaces for the right eigenvectors and $\mathcal{V}_i$ as search spaces for the left eigenvectors. An advantage of a two-sided method is that both the left and the right eigenvector are approximated which implies an accurate approximation of the eigenvalue (see Lemma 5.1). An obvious disadvantage is that such an approach requires twice as much work (in terms of matrix-vector multiplications) for one iteration. Numerical experiments in Section 8 indicate that for non right definite problems the two-sided Jacobi–Davidson type method often gives better results than the one-sided one.

A brief sketch of the two-sided Jacobi–Davidson type method for the two-parameter problem is presented in Algorithm 4.1. In step 2(b) we have to choose a Petrov triple. Some options are given later in this section. In step 2(e), we have to find new search directions in order to expand the search and test subspaces. We discuss several possible correction equations in Section 5.

ALGORITHM 4.1. A two-sided Jacobi–Davidson type method for the nonsingular two-parameter eigenvalue problem.

1. **Start.** Choose initial vectors u_1,u_2,v_1 and v_2 with unit norm.
   (a) Set $U_{i1}=[u_i]$, $V_{i1}=[v_i]$ for $i=1,2$.
   (b) Set $k=1$.
2. **Iterate.** Until convergence or $k>k_{\max}$ do:
   (a) Solve the projected two-parameter eigenvalue problem (3.1) by Algorithm 2.3.
   (b) Select an appropriate Petrov value (σ,τ) and the corresponding right and left Petrov vectors $u_1\otimes u_2$ and $v_1\otimes v_2$, where $u_i=U_{ik}c_i$, $v_i=V_{ik}d_i$ for $i=1,2$, respectively.

(c) Compute the right and left residuals

$$r_i^R = (A_i - \sigma B_i - \tau C_i) u_i, \tag{4.1}$$
$$r_i^L = (A_i - \sigma B_i - \tau C_i)^* v_i \tag{4.2}$$

for $i = 1, 2$.

(d) Stop if $\rho_k \le \varepsilon$, where

$$\rho_k = \left(\|r_1^R\|^2 + \|r_2^R\|^2 + \|r_1^L\|^2 + \|r_2^L\|^2\right)^{1/2}. \tag{4.3}$$

(e) Solve approximately one of the proposed correction equations (see Section 5) and obtain new directions s_i and t_i for $i = 1, 2$.

(f) Expand the search subspaces. Set

$$U_{i,k+1} = \text{RGS } (U_{ik}, s_i),$$
$$V_{i,k+1} = \text{RGS } (V_{ik}, t_i),$$

where RGS denotes the repeated Gram–Schmidt orthonormalization, for $i = 1, 2$.

(g) Set $k = k + 1$.

(h) Restart. If the dimension of the image of U_{ik} and V_{ik} exceeds $l_{\max}$, then replace U_{ik}, V_{ik} with new orthonormal bases of dimension $l_{\min}$.

To apply this algorithm, we need to specify a target $(\lambda_{\text{T}}, \mu_{\text{T}})$, a tolerance ε, a maximum number of steps $k_{\max}$, a maximum dimension of the search subspaces $l_{\max}$, and a number $l_{\min} < l_{\max}$ that specifies the dimension of the search subspaces after a restart (see Remark 4.2).

We also have to specify a criteria for Step 2(b). Suppose that we are looking for the eigenvalue closest to the target $(\lambda_{\text{T}}, \mu_{\text{T}})$. We suggest to combine two approaches. In the first part we select the Petrov value (σ, τ) closest to the target until the residual ρ_k drops below $\varepsilon_{\text{change}}$. In the second part we take the Petrov triple with the smallest residual (4.3). Both stages can be seen as an accelerated inexact Rayleigh quotient iteration.

REMARK 4.2. *As Algorithm 2.3 is able to solve only low-dimensional two-parameter problems (3.1) in a reasonable time, we expand the search spaces up to the preselected dimension $l_{\max}$ and then restart the algorithm. For a restart, we take the $l_{\min}$ eigenvector approximations with the smallest residuals (4.3) as a basis for the initial search space.*

REMARK 4.3. *In Step 2(d) we could also stop the algorithm if either the norm of the right residuals r_1^R and r_2^R or the norm of the left residuals r_1^L and r_2^L is small enough. In either case we can expect that (σ, τ) is a good approximation to an eigenvalue and we can compute the corresponding right or left eigenvectors by solving one (orthogonal) correction equation, see also [10].*

In the following section we discuss the expansion in Step 2(e) and derive several correction equations.

5. Correction equations. Let (σ, τ) be a Petrov value that approximates the eigenvalue (λ, μ) of (1.1) and let $u_1 \otimes u_2$ and $v_1 \otimes v_2$ be its corresponding left and right Petrov vector, respectively. Let us assume that u_1, u_2, v_1 and v_2 are normalized.

We are searching for orthogonal improvements of the left and right Petrov vectors of the form

$$(A_i - \lambda B_i - \mu C_i)\ (u_i + s_i) = 0, \tag{5.1}$$
$$(A_i - \lambda B_i - \mu C_i)^*(v_i + t_i) = 0, \tag{5.2}$$

where $s_i \perp a_i$ and $t_i \perp b_i$ for $i = 1, 2$. We will discuss the choices for a_i and b_i later, at this time we require just that $a_i \not\perp u_i$ and $b_i \not\perp v_i$.

Using (4.1) and (4.2), we can rewrite (5.1) and (5.2) as

$$\begin{aligned}(A_i - \sigma B_i - \tau C_i)\, s_i &= -r_i^R + (\lambda - \sigma)B_i u_i + (\mu - \tau)C_i u_i \\ &\quad + (\lambda - \sigma)B_i s_i + (\mu - \tau)C_i s_i,\end{aligned} \tag{5.3}$$

$$\begin{aligned}(A_i - \sigma B_i - \tau C_i)^* t_i &= -r_i^L + (\lambda - \sigma)^* B_i^* v_i + (\mu - \tau)^* C_i^* v_i \\ &\quad + (\lambda - \sigma)^* B_i^* t_i + (\mu - \tau)^* C_i^* t_i.\end{aligned} \tag{5.4}$$

LEMMA 5.1. *If $u_i = x_i - s_i$ and $v_i = y_i - t_i$, for $i = 1, 2$, are close enough approximations to a left and a right eigenvector of (1.1) for the same algebraically simple eigenvalue (λ, μ) then the two-sided Rayleigh quotient $(\sigma, \tau) = (\rho_1(u, v), \rho_2(u, v))$ is an $\mathcal{O}(\|s_1\|\|t_1\| + \|s_2\|\|t_2\|)$ approximation to (λ, μ), i.e.,*

$$\left\| \begin{bmatrix} \lambda - \sigma \\ \mu - \tau \end{bmatrix} \right\| = \mathcal{O}(\|s_1\|\|t_1\| + \|s_2\|\|t_2\|). \tag{5.5}$$

Proof. We write the residual (4.1) as

$$r_i^R = -(A_i - \lambda B_i - \mu C_i)s_i + (\lambda - \sigma)B_i u_i + (\mu - \tau)C_i u_i. \tag{5.6}$$

When we multiply equation (5.6) by v_i^* and take into account that $v_i^* r_i^R = 0$ and

$$v_i^*(A_i - \lambda B_i - \mu C_i) = -t_i^*(A_i - \lambda B_i - \mu C_i)$$

for $i = 1, 2$, then we obtain

$$\begin{bmatrix} v_1^* B_1 u_1 & v_1^* C_1 u_1 \\ v_2^* B_2 u_2 & v_2^* C_2 u_2 \end{bmatrix} \begin{bmatrix} \lambda - \sigma \\ \mu - \tau \end{bmatrix} = - \begin{bmatrix} t_1^*(A_1 - \lambda B_1 - \mu C_1)s_1 \\ t_2^*(A_2 - \lambda B_2 - \mu C_2)s_2 \end{bmatrix}. \tag{5.7}$$

If $\|s_i\|$ and $\|t_i\|$ are small enough then (5.7) is a nonsingular system because of Lemma 1.1 and continuity. We can deduce from (5.7) that

$$\left\| \begin{bmatrix} \lambda - \sigma \\ \mu - \tau \end{bmatrix} \right\| = \left\| \begin{bmatrix} v_1^* B_1 u_1 & v_1^* C_1 u_1 \\ v_2^* B_2 u_2 & v_2^* C_2 u_2 \end{bmatrix}^{-1} \begin{bmatrix} t_i^*(A_1 - \lambda B_1 - \mu C_1)s_i \\ t_2^*(A_2 - \lambda B_2 - \mu C_2)s_i \end{bmatrix} \right\|$$

and so obtain (5.5). $\square$

It follows from Lemma 5.1 that asymptotically (i.e., when we have good approximate right and left eigenvectors), we can consider s_i and t_i as first order corrections, $(\lambda-\sigma)B_i u_i + (\mu-\tau)C_i u_i$ and $(\lambda-\sigma)^* B_i^* v_i + (\mu-\tau)^* C_i^* v_i$ as second order corrections, and finally, $(\lambda-\sigma)B_i s_i + (\mu-\tau)C_i s_i$ and $(\lambda - \sigma)^* B_i^* t_i + (\mu - \tau)^* C_i^* t_i$ can be interpreted as third order corrections.

5.1. First order based correction equations. If we ignore the second and higher order terms in (5.3) then we obtain the equation

$$(A_i - \sigma B_i - \tau C_i)s_i = -r_i^R. \tag{5.8}$$

Because r_i^R is orthogonal to v_i, we can multiply (5.8) with an oblique projection $I - \frac{c_i v_i^*}{v_i^* c_i}$, where $c_i \not\perp v_i$, that does not change r_i^R. Secondly, since s_i is orthogonal to a_i, we can write $\left(I - \frac{u_i a_i^*}{a_i^* u_i}\right) s_i$ instead of s_i. Thus we obtain the correction equation for the vector u_i

$$\left(I - \frac{c_i v_i^*}{v_i^* c_i}\right) (A_i - \sigma B_i - \tau C_i) \left(I - \frac{u_i a_i^*}{a_i^* u_i}\right) s_i = -r_i^R \tag{5.9}$$

for $i=1,2$. In a similar way we obtain from (5.4) the correction equation for the vector v_i

$$\left(I-\frac{d_iu_i^*}{u_i^*d_i}\right)(A_i-\sigma B_i-\tau C_i)^*\left(I-\frac{v_ib_i^*}{b_i^*v_i}\right)t_i=-r_i^L \tag{5.10}$$

for $i=1,2$, where $d_i \not\perp u_i$.

We solve these correction equations only approximately, for instance using some Krylov subspace method. Since the operator in (5.9) maps $a_i^\perp$ onto $v_i^\perp$, it is suitable to take $a_i=v_i$ in order to apply Krylov solver without a preconditioner (see for example the discussion in [10, Section 4.2]). If $a_i\neq v_i$, then we need a preconditioner that maps the image space $v_i^\perp$ bijectively onto $a_i^\perp$. Similarly, we need a preconditioner for (5.10) when $b_i\neq u_i$.

Different choices of vectors a_i,b_i,c_i,d_i lead to different correction equations. We discuss some options.

1. For the first correction equation we take $a_i=d_i=v_i$, $b_i=c_i=u_i$. We obtain a pair of correction equations

$$\begin{aligned}\left(I-\frac{u_iv_i^*}{v_i^*u_i}\right)(A_i-\sigma B_i-\tau C_i)\left(I-\frac{u_iv_i^*}{v_i^*u_i}\right)s_i&=-r_i^R,\\ \left(I-\frac{v_iu_i^*}{u_i^*v_i}\right)(A_i-\sigma B_i-\tau C_i)^*\left(I-\frac{v_iu_i^*}{u_i^*v_i}\right)t_i&=-r_i^L\end{aligned} \tag{5.11}$$

   for $s_i\perp v_i$, $t_i\perp u_i$ for $i=1,2$. The operator in the first equation is the conjugate transpose of the operator in the second equation and we can solve these equations simultaneously by bi-conjugate gradients (BiCG). It is also possible to solve equations in (5.11) separately by the generalized minimum residual method (GMRES).
2. For this correction equation we take $a_i=c_i=u_i$, $b_i=d_i=v_i$.
   It is a natural approach for (5.9) and (5.10) to take $a_i=u_i$ and $b_i=v_i$ as in this case we are looking for updates orthogonal to the current approximation. As it turns out later in Section 5.2, when we use preconditioning, an interesting choice for c_i and d_i is to take $c_i=u_i$ and $d_i=v_i$, which leads to a pair of correction equations

$$\begin{aligned}\left(I-\frac{u_iv_i^*}{v_i^*u_i}\right)(A_i-\sigma B_i-\tau C_i)(I-u_iu_i^*)s_i&=-r_i^R,\\ \left(I-\frac{v_iu_i^*}{u_i^*v_i}\right)(A_i-\sigma B_i-\tau C_i)^*(I-v_iv_i^*)t_i&=-r_i^L\end{aligned} \tag{5.12}$$

   for $s_i\perp u_i$, $t_i\perp v_i$ for $i=1,2$. In order to solve (5.12) approximately by a Krylov solver we need a preconditioner because $a_i\neq v_i$, see Section 5.2.
3. In this case we take $a_i=u_i$, $b_i=v_i$, $c_i=g_i$, $d_i=h_i$, where

$$\begin{aligned}g_i&=(\lambda_{\mathrm T}-\sigma)B_iu_i+(\mu_{\mathrm T}-\tau)C_iu_i,\\ h_i&=(\lambda_{\mathrm T}-\sigma)^*B_i^*v_i+(\mu_{\mathrm T}-\tau)^*C_i^*v_i.\end{aligned}$$

   The idea behind the choice of c_i and d_i is that when the target $(\lambda_{\mathrm T},\mu_{\mathrm T})$ is close to the eigenvalue then the projections with g_i and h_i almost annihilate the second order terms in equations (5.3) and (5.4) and thus reduce the neglected quantity.
   We derive the correction equations

$$\begin{aligned}\left(I-\frac{g_iv_i^*}{v_i^*g_i}\right)(A_i-\sigma B_i-\tau C_i)(I-u_iu_i^*)s_i&=-r_i^R,\\ \left(I-\frac{h_iu_i^*}{u_i^*h_i}\right)(A_i-\sigma B_i-\tau C_i)^*(I-v_iv_i^*)t_i&=-r_i^L\end{aligned} \tag{5.13}$$

for $s_i \perp u_i$, $t_i \perp v_i$ for $i = 1, 2$. Again, if we want to solve (5.13) approximately by a Krylov solver then we need a preconditioner as $a_i \neq v_i$, see the next section.

5.2. Preconditioned first order based correction equations. We mentioned that we need a preconditioner for a Krylov solver for the correction equation (5.9) when the domain subspace $a_i^\perp$ and the range subspace $v_i^\perp$ do not agree. But we can also use a preconditioner when the domain and the range agree to speed up the convergence.

Suppose that a left preconditioner M_i is available for $A_i - \sigma B_i - \tau C_i$ such that $M_i^{-1}(A_i - \sigma B_i - \tau C_i) \approx I$. A calculation shows that if we assume that $a_i^* M_i^{-1} c_i \neq 0$ then the inverse of the map

$$\left(I - \frac{c_i v_i^*}{v_i^* c_i}\right) M_i \left(I - \frac{u_i a_i^*}{a_i^* u_i}\right)$$

from $a_i^\perp$ to $v_i^\perp$ is the map

$$\left(I - \frac{M_i^{-1} c_i a_i^*}{a_i^* M_i^{-1} c_i}\right) M_i^{-1} \left(I - \frac{c_i v_i^*}{v_i^* c_i}\right)$$

from $v_i^\perp$ to $a_i^\perp$. Therefore, using left preconditioning changes (5.9) into

$$\begin{aligned}\left(I - \frac{M_i^{-1} c_i a_i^*}{a_i^* M_i^{-1} c_i}\right) M_i^{-1} \left(I - \frac{c_i v_i^*}{v_i^* c_i}\right)(A_i - \sigma B_i - \tau C_i) \left(I - \frac{u_i a_i^*}{a_i^* u_i}\right) s_i \\ = -\left(I - \frac{M_i^{-1} c_i a_i^*}{a_i^* M_i^{-1} c_i}\right) M_i^{-1} r_i^R\end{aligned}$$

for $i = 1, 2$. Correction equation (5.10) for the left eigenvector can be dealt with similarly. A preconditioner for $A_i - \sigma B_i - \tau C_i$ automatically suggests a preconditioner for $(A_i - \sigma B_i - \tau C_i)^*$.

We can combine different preconditioners with different correction equations. Here are some possibilities.

1. Our suggestion for the preconditioner is

$$M_i = A_i - \lambda_{\rm T} B_i - \mu_{\rm T} C_i, \tag{5.14}$$

   where $(\lambda_{\rm T}, \mu_{\rm T})$ is the target. Instead of exact inversion we can also take an inexact inverse, for example one obtained using an incomplete LU decomposition.
2. The simplest option is to take the identity as a preconditioner in order to be able to use a Krylov solver for the correction equation. For example, if we take correction equation (5.12) and the identity as a preconditioner, then we have to multiply (5.9) and (5.10) by orthogonal projectors $I - u_i u_i^*$ and $I - v_i v_i^*$, respectively. From $(I - u_i u_i^*)\left(I - \frac{u_i v_i^*}{v_i^* u_i}\right) = I - u_i u_i^*$ and $(I - v_i v_i^*)\left(I - \frac{v_i u_i^*}{u_i^* v_i}\right) = I - v_i v_i^*$ we get

$$\begin{aligned}(I - u_i u_i^*)(A_i - \sigma B_i - \tau C_i)(I - u_i u_i^*) s_i &= -(I - u_i u_i^*) r_i^R, \\ (I - v_i v_i^*)(A_i - \sigma B_i - \tau C_i)^*(I - v_i v_i^*) t_i &= -(I - v_i v_i^*) r_i^L\end{aligned} \tag{5.15}$$

   for $i = 1, 2$. One can recognize (5.15) as the correction equations of standard Jacobi–Davidson applied to $A_i - \sigma B_i - \tau C_i$ and $(A_i - \sigma B_i - \tau C_i)^*$.

5.3. Second order based correction equation. For this case we generalize the correction equation with oblique projections for the right definite two-parameter eigenvalue problem [11]. If we define

$$K = \begin{bmatrix} A_1 - \sigma B_1 - \tau C_1 & 0 \\ 0 & A_2 - \sigma B_2 - \tau C_2 \end{bmatrix},$$

$$r^R = \begin{bmatrix} r_1^R \\ r_2^R \end{bmatrix}, \quad r^L = \begin{bmatrix} r_1^L \\ r_2^L \end{bmatrix},$$

then we can reformulate (5.3) and (5.4) (neglecting third order correction terms) as

$$K \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = -r^R + (\lambda - \sigma) \begin{bmatrix} B_1 u_1 \\ B_2 u_2 \end{bmatrix} + (\mu - \tau) \begin{bmatrix} C_1 u_1 \\ C_2 u_2 \end{bmatrix} \tag{5.16}$$

and

$$K^* \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} = -r^L + (\lambda - \sigma)^* \begin{bmatrix} B_1^* v_1 \\ B_2^* v_2 \end{bmatrix} + (\mu - \tau)^* \begin{bmatrix} C_1^* v_1 \\ C_2^* v_2 \end{bmatrix}. \tag{5.17}$$

Let V_{R} be a $(n_1 + n_2) \times 2$ matrix with orthonormal columns such that

$$\mathrm{span}(V_{\mathrm{R}}) = \mathrm{span}\left(\begin{bmatrix} B_1 u_1 \\ B_2 u_2 \end{bmatrix}, \begin{bmatrix} C_1 u_1 \\ C_2 u_2 \end{bmatrix} \right)$$

and let

$$W_{\mathrm{R}} = \begin{bmatrix} v_1 & 0 \\ 0 & v_2 \end{bmatrix}.$$

With the oblique projection

$$P_{\mathrm{R}} = I - V_{\mathrm{R}} (W_{\mathrm{R}}^* V_{\mathrm{R}})^{-1} W_{\mathrm{R}}^*$$

onto $\mathrm{span}(W_{\mathrm{R}})^{\perp}$ along $\mathrm{span}(V_{\mathrm{R}})$, it follows that

$$P_{\mathrm{R}} r^R = r^R \quad \text{and} \quad P_{\mathrm{R}} \begin{bmatrix} B_1 u_1 \\ B_2 u_2 \end{bmatrix} = P_{\mathrm{R}} \begin{bmatrix} C_1 u_1 \\ C_2 u_2 \end{bmatrix} = 0.$$

Therefore, from multiplying (5.16) by P_{R} we obtain

$$P_{\mathrm{R}} K \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = -r^R.$$

Suppose that we are looking for corrections such that $s_i \perp v_i$ and $t_i \perp u_i$. Then

$$P_{\mathrm{R}} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}$$

and the result is the correction equation

$$P_{\mathrm{R}} K P_{\mathrm{R}} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = -r^R. \tag{5.18}$$

REMARK 5.2. *If $u_1 \otimes u_2$ and $v_1 \otimes v_2$ are close approximations to eigenvectors $x_1 \otimes x_2$ and $y_1 \otimes y_2$, corresponding to a single eigenvalue of (1.1), then it follows from Lemma 1.1 that $W_R^* V_R$ is nonsingular. If the above is not true, then it is possible that V_R does not exist or that $W_R^* V_R$ is singular. In either of these two cases we can use one of the correction equations from Section 5.1 to expand the search and test spaces.*

In a similar manner we obtain a correction equation for t_1 and t_2. If V_{L}, W_{L}, and P_{L} are defined similarly for (5.17), then we have

$$P_{\mathrm{L}} K^* P_{\mathrm{L}} \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} = -r^L. \tag{5.19}$$

We separately solve (5.18) and (5.19) approximately using a few steps of GMRES.

Better results can be expected if we use preconditioners. Suppose that M is a left preconditioner for K, for instance a block preconditioner with the preconditioners M_i in (5.14) as blocks. One can show that if $W_{\mathrm{R}}^* M^{-1} V_{\mathrm{R}}$ is nonsingular then the inverse of a map $P_{\mathrm{R}} M P_{\mathrm{R}}$ from $\mathrm{span}(W_{\mathrm{R}})^\perp$ to $\mathrm{span}(W_{\mathrm{R}})^\perp$ is

$$\left(I - M^{-1} V_{\mathrm{R}} (W_{\mathrm{R}}^* M^{-1} V_{\mathrm{R}})^{-1} W_{\mathrm{R}}^*\right) M^{-1} P_{\mathrm{R}}.$$

Thus we obtain a preconditioned correction equation

$$\begin{aligned} &\left(I - M^{-1} V_{\mathrm{R}} (W_{\mathrm{R}}^* M^{-1} V_{\mathrm{R}})^{-1} W_{\mathrm{R}}^*\right) M^{-1} P_{\mathrm{R}} K P_{\mathrm{R}} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \\ &\qquad = \left(I - M^{-1} V_{\mathrm{R}} (W_{\mathrm{R}}^* M^{-1} V_{\mathrm{R}})^{-1} W_{\mathrm{R}}^*\right) M^{-1} r^R. \end{aligned} \tag{5.20}$$

In a similar manner we get a preconditioned equation for t_1 and t_2.

5.4. One-sided approach. Instead of the two-sided we could also apply the one-sided approach where the search subspace is the same as the test subspace. One-sided versions can be easily derived from the above two-sided correction equations. All one has to do is to use $V_i = U_i$ for $i = 1, 2$, and to solve only the correction equations for s_1 and s_2.

The advantage of the one-sided approach is that it requires roughly half as much work for one outer iteration. On the other side, numerical results in Section 8 show that two-sided approach gives more accurate results. Also, if we use the one-sided approach then we can not apply Lemma 1.1 as we did in Remark 5.2.

6. Computing more eigenpairs. Suppose that we are interested in $p > 1$ eigenpairs of (1.1). In one-parameter eigenvalue problems various deflation techniques can be applied in order to compute more than one eigenpair. The difficulties that are met when we try to translate standard deflation ideas from one-parameter problems to two-parameter problems are discussed in [11].

For a general two-parameter eigenvalue problem we can apply a similar technique as in [11] for the right definite problem using the Δ_0-orthogonality of left and right eigenvectors. Suppose that we have already found p eigenvalues (λ_i, μ_i) with the corresponding left and right eigenvectors $x_{1i} \otimes x_{2i}$ and $y_{1i} \otimes y_{2i}$ for $i = 1, \ldots, p$. Now we adjust Algorithm 4.1 so that in Step 2b we consider only those Petrov triples for which $u_1 \otimes u_2$ and $v_1 \otimes v_2$ satisfy

$$\min\Big(|(v_1 \otimes v_2)^* \Delta_0 (x_{1i} \otimes x_{2i})|, |(y_{1i} \otimes y_{2i})^* \Delta_0 (u_1 \otimes u_2)| \Big) < \eta \text{ for } i = 1, \ldots, p \tag{6.1}$$

for an $\eta > 0$. A suggestion for η (used in Example 8.4 in Section 8) is

$$\eta = \frac{1}{2} \min_{i=1,\ldots,p} \Big((y_{1i} \otimes y_{2i})^* \Delta_0 (x_{1i} \otimes x_{2i}) \Big).$$

If no triple satisfies this condition then we take the one with the smallest left side of (6.1).

Let us mention that an efficient way to compute (6.1) is to apply the relation (cf. (3.2))

$$(x_1 \otimes x_2)^* \Delta_0 (y_1 \otimes y_2) = (x_1^* B_1 y_1)(x_2^* C_2 y_2) - (x_1^* C_1 y_1)(x_2^* B_2 y_2).$$

If we want to compute more eigenpairs using the one-sided approach then we have to compute the left eigenvectors separately for each converged eigenvalue. If we use the two-sided approach then left and right eigenvectors are already computed.

7. Time complexity. The analysis of time complexity of Algorithm 4.1 is similar to the analysis for the Jacobi–Davidson algorithm for right definite two-parameter eigenvalue in [11, Section 6]. Because of that the details are omitted and the main results are stated.

If we assume that $n = n_1 = n_2$ and that m steps of GMRES are used for the approximate solutions of the correction equations, then the time complexity of one outer step of Algorithm 4.1 for dense matrices is $\mathcal{O}(mn^2)$. Also important is the storage requirement. If an algorithm works with matrices A_i, B_i, and C_i as Algorithm 4.1 does then it requires $\mathcal{O}(n^2)$ memory. On the other hand, Algorithm 2.3 that works with the associated system (1.5) needs $\mathcal{O}(n^4)$ memory, which may fast exceed the available memory, even for modest values of n.

If the matrices A_i, B_i, and C_i are sparse, then the time complexity of the outer step of Algorithm 4.1 is of order $\mathcal{O}(m\mathrm{MV})$, where MV stands for a matrix-vector multiplication by an $n \times n$ matrix.

8. Numerical examples. The following numerical results were obtained with Matlab 6.5.

In the first examples we use a two-parameter eigenvalue problem with known eigenpairs, which enables us to check the obtained results. The construction is similar to the one in [11] and therefore the details are omitted.

We take matrices

$$A_i = V_i F_i U_i, \quad B_i = V_i G_i U_i, \quad C_i = V_i H_i U_i \tag{8.1}$$

of dimension $n \times n$, where F_i, G_i, and H_i are complex diagonal matrices and U_i, V_i are random complex matrices for $i = 1, 2$. We select diagonal elements of matrices F_1, F_2, G_2, and H_1 as complex numbers $\alpha + i\beta$ where α and β are uniformly distributed random numbers from the interval $(-0.5, 0.5)$. The same applies to the elements of U_i and V_i.

All the eigenvalues can be computed from the diagonal elements of F_i, G_i, and H_i for $i = 1, 2$. In the following numerical examples we use matrices (8.1) of size $n = 100$.

Example 8.1. We compare different correction equations without preconditioning. For the initial vectors we perturb the exact eigenvectors with a random perturbation of order 10^{-3}. In each Step 2b of Algorithm 4.1 we take the Petrov triple with the smallest residual (4.3).

Table 8.1 contains the number of steps required for the residual (4.3) to become smaller than 10^{-8}. The other parameters are $l_{\max} = 10$, $l_{\min} = 2$ and $k_{\max} = 200$. We compared three two-sided correction equations without preconditioning:

- NP1 - first order correction equation (5.11), where $s_i \perp v_i$ and $t_i \perp u_i$.
- NP2 - first order correction equation (5.15), where $s_i \perp u_i$ and $t_i \perp v_i$. Although it is preconditioned, we treat this equation as an unpreconditioned one because the preconditioner is the identity.

TABLE 8.1

Comparison of three correction equations NP1, NP2, and NP3 without preconditioning for the initial vectors $\|u_i - x_{1i}\| = \mathcal{O}(10^{-3})$ *and* $\|v_i - y_{1i}\| = \mathcal{O}(10^{-3})$*. GMRES: the number of steps used in GMRES for the approximate solution of the correction equation; Iterations: the number of outer iterations for convergence.*

NP1		NP2		NP3	
GMRES	Iterations	GMRES	Iterations	GMRES	Iterations
90	> 200	90	> 200	180	50
95	46	95	36	190	25
99	3	99	3	199	5

- NP3 - second order correction equation (5.18) and (5.19).

The results in the table indicate that the convergence is slow or we have no convergence at all if the correction equations are not solved accurately, and this happens as the number of GMRES steps gets closer to the size of the matrices. Let us remark that the number of GMRES steps for the second order correction equation is larger because the size of the matrices is twice the size of the matrices in the first order correction equations.

EXAMPLE 8.2. For the second example we take the same initial vectors and parameters as in Example 8.1, but, this time we use preconditioned correction equations. For a preconditioner we take (5.14). We compared the following three two-sided preconditioned correction equations:

- P1 - preconditioned first order correction equation NP1 from Example 8.1, where $s_i \perp v_i$ and $t_i \perp u_i$.
- P2 - preconditioned first order correction equation (5.13), where $s_i \perp u_i$, $t_i \perp v_i$, and where the second order terms are small close to the eigenvalue.
- P3 - (5.20) preconditioned second order correction equation NP3 from Example 8.1.

TABLE 8.2

Comparison of three correction equations P1, P2, and P3 with preconditioning for initial vectors $\|u_i - x_{1i}\| = \mathcal{O}(10^{-3})e$ *and* $\|v_i - y_{1i}\| = \mathcal{O}(10^{-3})$*. GMRES: the number of steps used in GMRES for the approximate solution of the correction equation; iterations: the number of outer iterations for convergence.*

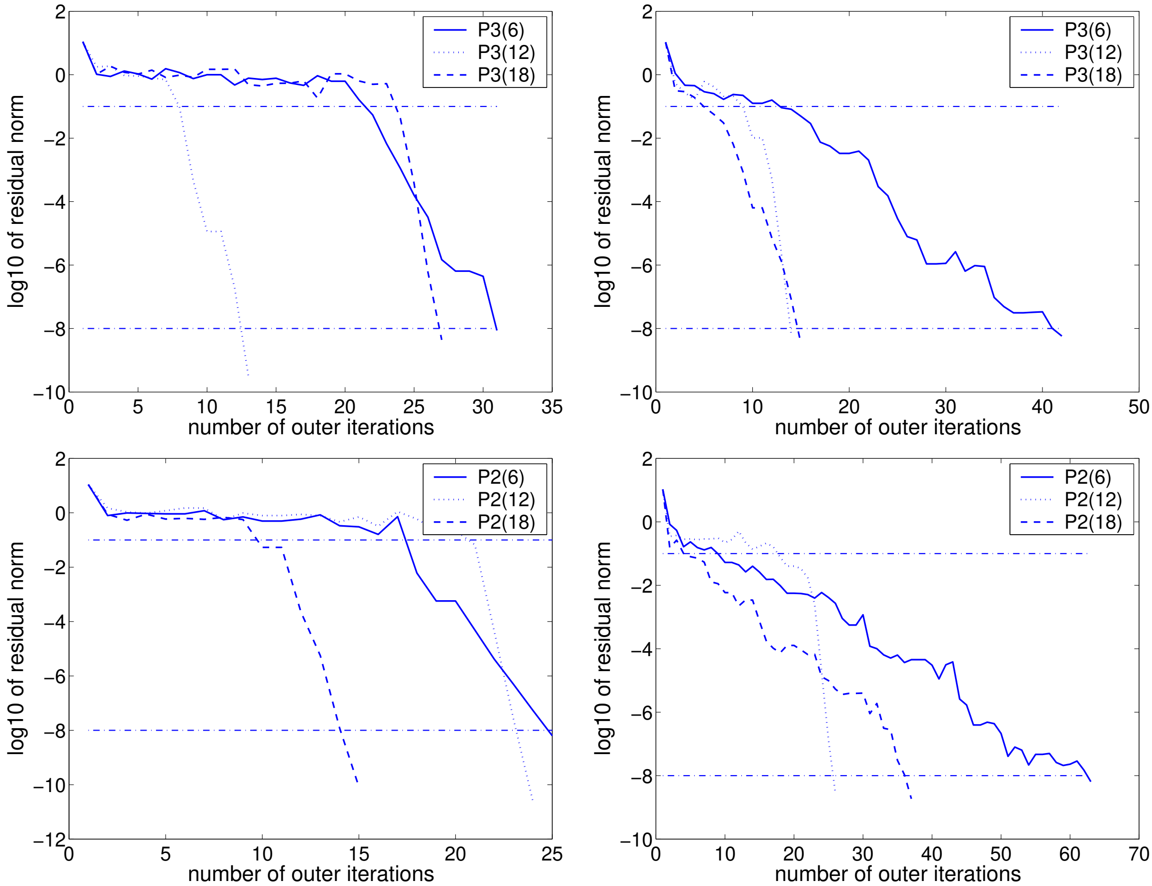
P1		P2		P3	
GMRES	Iterations	GMRES	Iterations	GMRES	Iterations
1	63	1	63	1	99
2	70	2	59	2	36
4	28	4	28	4	24
8	6	8	6	8	6
15	4	15	4	15	3

The results in Table 8.2 indicate that correction equations with preconditioners work better than the ones that are not preconditioned and we have a fast convergence for a modest number of GMRES steps.

EXAMPLE 8.3. In this example we take initial vectors $u_1 = u_2 = v_1 = v_2 = [1 \ \cdots \ 1]^T$. Our goal is the eigenvalue closest to the origin. In Step 2b of Algorithm 4.1 we pick the Petrov triple with the Petrov value closest to the target $(0,0)$ until the residual ρ_k is less than $\varepsilon_{\text{change}} = 0.1$. After that we take the Petrov triple with the smallest residual (4.3).

Figure 8.1 shows the convergence plot for two and one sided correction equations P2 and P3 using various number of GMRES steps to solve the correction equation. One can see that once the residual becomes smaller than $\varepsilon_{\text{change}}$ (top horizontal dotted line in the figures) and we are

FIG. 8.1. *Convergence plot for the eigenvalue closest to* $(0,0)$ *for* $u_i = v_i = [1 \ \cdots \ 1]^T$. *The plots show the* $\log_{10}$ *of the residual norm (4.3) versus the outer iteration number for the Jacobi–Davidson type method using correction equation with 6 (solid line), 12 (dotted line), and 18 (dashed line) GMRES steps to solve the correction equation. The correction equations are: two-sided P3 (top left), one-sided P3 (top right), two-sided P2 (bottom left) and one-sided P2 (bottom right).*



close to the eigentriple, the number of GMRES steps determines the speed of the convergence.

There is no guarantee that the process will converge to the eigenvalue closest to the target. Table 8.3 shows the indices of the obtained eigenvalues if the eigenvalues are ordered by their distance from the target. This example shows that although the one-sided methods may converge faster than the two-sided methods (especially measured in number of matrix-vector multiplications), they often converge to an undesired eigenvalue.

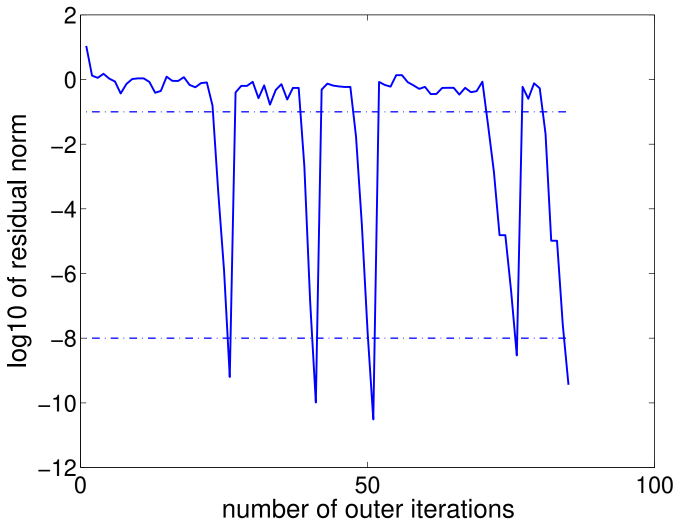
TABLE 8.3
Indices of the obtained eigenvalues from Figure 8.1.

GMRES	two-sided P3	one-sided P3	two-sided P2	one-sided P2
6	1	21	2	21
12	1	1	1	5
18	3	24	17	70

EXAMPLE 8.4. In this example we test the selection technique from Section 6 that enables us to compute more than one eigenvalue. Figure 8.2 shows a convergence plot for the first five computed eigenvalues. For each eigenvalue we select the closest Petrov value to the origin until

the residual becomes smaller than $\varepsilon_{\text{change}}$ and in the remaining steps we select Petrov triple with the minimum residual. We consider only Petrov triples that satisfy the condition (6.1). Indeed, we get the five eigenvalues closest to the target $(0,0)$. The eigenvalues are not always computed in a row by their indices but the statistics in the following example show that the probability of a successful convergence is high if we carefully tune the parameters of the method.

FIG. 8.2. *Convergence plot for the first five computed eigenvalues using the selection technique from Section 6. Used is correction equation P3 with 15 GMRES steps and* $\varepsilon_{\text{change}} = 0.1$.



EXAMPLE 8.5. We test the preconditioned correction equation P3 on the same set of 10 random initial vectors. For each initial vectors the goal was to compute the 10 eigenvalues closest to the target using the same approach as in the previous example. We set the maximum number of outer steps to 400 and use different number of GMRES steps and different $\varepsilon_{\text{change}}$.

The numbers in Table 8.4 show that the probability of computing the correct eigenvalues is high when the parameters are carefully chosen. A small value of $\varepsilon_{\text{change}}$ does not necessarily improve the probability. If $\varepsilon_{\text{change}}$ is too small then in the first phase, when we select the closest Petrov value to the origin, the method requires too many iterations until the residual is smaller than $\varepsilon_{\text{change}}$. On the other hand, if $\varepsilon_{\text{change}}$ is too large then the method is likely to converge fast, but to an unwanted eigenvalue. More GMRES steps reduce the number of outer iterations and enlarge the probability, but we must keep in mind that the total amount of work is dependent to the number of matrix-vector multiplications, and thus roughly equal to the product of the number of GMRES steps and outer iterations.

The results show that we can compute more eigenvalues close to the target if we use the two-sided method. The performance of the one-sided method is less optimal, for large values of $\varepsilon_{\text{change}}$ it may converge faster than the two-sided method but often converges to undesired eigenvalues, while for smaller values of $\varepsilon_{\text{change}}$ we often have no convergence at all.

EXAMPLE 8.6. In the last example we study the three-point problem

$$y'' + (\lambda + \mu \cos x)y = 0 \tag{8.2}$$

with boundary conditions

$$y(0) = y(2.5) = y(5) = 0.$$

TABLE 8.4

Statistics of the Jacobi–Davidson type method using the same set of 10 random initial vectors for computing 10 closest eigenvalues to the origin using correction equation P3 and different number of GMRES steps and $\varepsilon_{\text{change}}$. *The parameters are* $l_{\max} = 10$ *and* $l_{\min} = 2$*, maximum number of outer iterations is* 400. *GMRES: the number of steps used in GMRES for the approximate solution of the correction equation; In 10: the average number of the computed eigenvalues among the 10 closest eigenvalues to the origin; Conv: the average number of the computed eigenvalues; Avg.: the average index of the computed eigenvalues if the eigenvalues are ordered by their distance from the target; Iter.: the average number of outer iterations for convergence.*

two-sided correction equation P3												
	$\varepsilon_{\text{change}} = 10^{-1}$				$\varepsilon_{\text{change}} = 10^{-1.5}$				$\varepsilon_{\text{change}} = 10^{-2}$			
GMRES	In 10	Conv.	Avg.	Iter.	In 10	Conv.	Avg.	Iter.	In 10	Conv.	Avg.	Iter.
5	3.4	4.2	6.4	400.0	3.3	3.9	4.7	400.0	2.7	3.0	4.3	400.0
10	4.7	7.4	19.4	324.5	5.9	8.0	10.2	387.8	5.3	6.2	5.6	400.0
20	6.8	9.4	15.2	255.3	6.6	9.2	26.1	301.8	6.9	9.4	14.5	300.3
40	7.2	9.5	29.1	284.0	7.3	9.5	16.2	292.3	7.0	9.0	14.2	354.9
one-sided correction equation P3												
	$\varepsilon_{\text{change}} = 10^{-1}$				$\varepsilon_{\text{change}} = 10^{-1.5}$				$\varepsilon_{\text{change}} = 10^{-2}$			
GMRES	In 10	Conv.	Avg.	Iter.	In 10	Conv.	Avg.	Iter.	In 10	Conv.	Avg.	Iter.
5	2.0	5.2	21.3	400.0	1.3	1.3	6.9	400.0	1.2	0.5	1.6	400.0
10	2.9	7.1	21.7	357.3	2.6	3.0	4.1	400.0	1.9	1.9	1.6	400.0
20	3.5	9.9	72.7	189.5	3.0	3.7	24.0	400.0	1.9	2.1	4.2	400.0
40	3.0	9.9	75.1	143.8	3.5	4.0	5.1	380.5	2.9	3.2	25.6	400.0

Instead of (8.2) we can study the two-parameter problem

$$y_i'' + (\lambda + \mu \cos x_i) y_i = 0, \quad i = 1, 2, \tag{8.3}$$

where $x_1 \in [0, 2.5]$, $x_2 \in [2.5, 5]$ and the boundary conditions are $y_1(0) = y_1(2.5) = 0$ and $y_2(2.5) = y_2(5) = 0$. One can see from the determinant

$$\begin{vmatrix} 1 & \cos(x_1) \\ 1 & \cos(x_2) \end{vmatrix} = \cos(x_2) - \cos(x_1)$$

that (8.3) is not right definite.

We can compute eigenvalues of (8.3) using finite differences. If we take $h = 1/(n-1)$, $x_{1i} = ih$, $x_{2i} = x_{1i} + 2.5$ for $i = 1, \ldots, n$, then the $n \times n$ matrices that form the two-parameter problem are

$$A_1 = A_2 = \frac{1}{h^2} \operatorname{tridiag}(1, -2, 1),$$

$$B_1 = B_2 = I, \tag{8.4}$$

$$C_1 = \operatorname{diag}(\cos(x_{11}), \ldots, \cos(x_{1n})), \quad C_2 = \operatorname{diag}(\cos(x_{21}), \ldots, \cos(x_{2n})).$$

Eigenfunctions for the six closest eigenvalues to $(0, 0)$ are shown on Figure 8.3.

Using finite differences and $n = 100$ we test preconditioned correction equation P3 using the same set of 50 random initial vectors and various number of GMRES steps. The goal is to compute the 10 closest eigenvalues to the target $(0, 0)$. Results in Table 8.5 show that it is possible to compute a selection of the closest eigenvalues to the target using the Jacobi–Davidson type method. It appears that the optimal solution in this case is to take a modest number of GMRES steps.

FIG. 8.3. *Eigenfunctions of the three-point boundary problem (8.2) for the six closest eigenvalues to (0,0):* $(\lambda_1,\mu_1) = (-1.5790, 0)$, $(\lambda_2,\mu_2) = (-6.3145, 0)$, $(\lambda_3,\mu_3) = (-2.1197, 6.5418)$, $(\lambda_4,\mu_4) = (-5.1698, -5.4264)$, $(\lambda_5,\mu_5) = (-8.9898, 8.4441)$ *and* $(\lambda_6,\mu_6) = (-14.2019, 0)$.

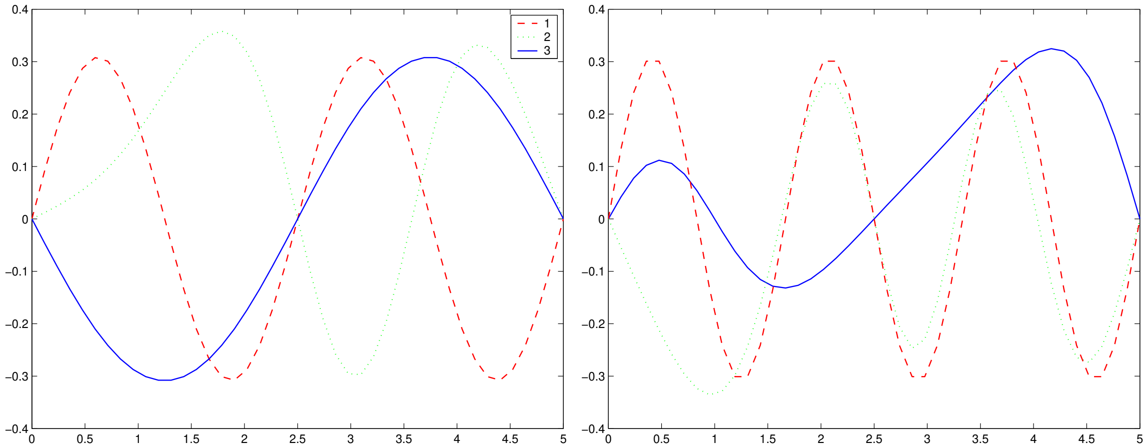


TABLE 8.5

Statistics of the Jacobi–Davidson type method using the same set of 10 random initial vectors for computing 10 closest eigenvalues to the origin using correction equation P3 and different number of GMRES steps for problem (8.4) and $n = 100$. *The parameters are* $l_{\max} = 10$, $l_{\min} = 2$, *and* $\varepsilon_{\text{change}} = 10^{-1.5}$. *GMRES: the number of steps used in GMRES for the approximate solution of the correction equation; In 10: the average number of the computed eigenvalues among the 10 closest eigenvalues to the origin; Iterations: the average number of outer iterations for convergence.*

Corr. equation	GMRES	In 10	Iterations
two-sided P3	5	10.0	155.6
two-sided P3	10	9.8	64.3
two-sided P3	20	10.0	58.8
two-sided P3	40	9.9	86.4
one-sided P3	5	9.5	123.4
one-sided P3	10	9.2	61.7
one-sided P3	20	9.5	62.8
one-sided P3	40	9.7	83.3

In this example the difference in the performance of the one-sided and the two-sided approach is smaller as in Example 8.5. This happens because the matrices are real symmetric and therefore the left and right eigenvectors of real eigenvalues agree. The discretized problem (8.4) has complex eigenvalues as well but the ones that we are interested in are all real.

9. Conclusions. We have presented a new Jacobi–Davidson type method for the nonsingular two-parameter eigenvalue problem. This problem is a very challenging one, where we have to use many available techniques to be successful: a two-sided subspace approach, preconditioning, selection techniques instead of deflating, and the use of a target.

Numerical examples show that the two-sided subspace approach is often more expensive, but also more reliable. An additional advantage of the two-sided approach is that during the process we have approximate left and right eigenvectors, and hence in principle (see [12] for details) an approximation to the condition number of the eigenvalue to which we are converging.

The new method can compute selected eigenpairs without good initial approximations and it can tackle very large two-parameter problems, especially if the matrices A_i, B_i, and C_i are sparse. In such situations, preconditioning is of great importance.

Let us also mention that Algorithm 2.3 and Algorithm 4.1 both offer a simple generalization to multiparameter problems with more than two parameters.

Acknowledgments. The authors are grateful to the referees for the careful reading of the paper and several helpful comments.

REFERENCES

[1] F. V. Atkinson, *Multiparameter spectral theory*, Bull. Amer. Math. Soc., 74 (1968), pp. 1–27.
[2] F. V. Atkinson, *Multiparameter eigenvalue problems*, Academic Press, New York, 1972.
[3] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, H. van der Vorst, *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, SIAM, Philadelphia, 2000.
[4] Z. Bohte, *Numerical solution of some two-parameter eigenvalue problems*, in Anton Kuhelj Memorial Volume, Slovenian Academie of Science and Art, Ljubljana, Slovenia, 1982, pp. 17–28.
[5] A. Bunse-Gerstner, R. Byers, V. Mehrmann, *Numerical methods for simultaneous diagonalization*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 927–949.
[6] N. Cottin, *Dynamic model updating — a multiparameter eigenvalue problem*, Mech. Syst. Signal Pr., 15 (2001), pp. 649–665.
[7] L. Fox, L. Hayes, D. F. Mayers, *The double eigenvalue problems*, in *Topics in numerical analysis, Proceedings of the Royal Irish academy conference on numerical analysis*, 1972, ed. J. J. H. Miller, Academic Press, New York, pp. 93–112.
[8] G. H. Golub, C. F. van Loan, *Matrix computations*, 3rd ed., Johns Hopkins University Press, Baltimore, 1996.
[9] T. Košir, *Finite dimensional multiparameter spectral theory: the nonderogatory case*, Linear Algebra Appl., 212/213 (1994), pp. 45–70.
[10] M. E. Hochstenbach, G. L. G. Sleijpen, *Two-sided and alternating Jacobi–Davidson*, Linear Algebra Appl., 358 (2003), pp. 145–172.
[11] M. E. Hochstenbach, B. Plestenjak, *A Jacobi–Davidson type method for a right definite two-parameter problem*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 392–410.
[12] M. E. Hochstenbach, B. Plestenjak, *Backward error, condition numbers, and pseudospectrum for the multiparameter eigenvalue problem*, Linear Algebra Appl., 375 (2003), pp. 63–81.
[13] X. Ji, *Numerical solution of joint eigenpairs of a family of commutative matrices*, Appl. Math. Lett., 4 (1991), pp. 57–60.
[14] J. R. Kuttler, V. G. Sigillito, *Eigenvalues of the Laplacian in two dimensions*, SIAM Review, 26 (1984), pp. 163–193.
[15] B. C. Lesieutre, A. V. Mamishev, Y. Du, E. Keskiner, M. Zahn, G. C. Verghese, *Forward and inverse parameter estimation algorithms of interdigital dielectrometry sensors*, IEEE T. Dielect. El. In., 8 (2001), pp. 577–588.
[16] M. R. Osborne, *Iterative procedures for solving finite-difference approximations to separable partial differential equation*, Comput. J., 6 (1963), pp. 93–99.
[17] B. Plestenjak, *A continuation method for a right definite two-parameter eigenvalue problem*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1163–1184.
[18] B. Plestenjak, *A continuation method for a weakly elliptic two-parameter eigenvalue problem*, IMA J. Numer. Anal., 21 (2001), pp. 199–216.
[19] G. L. G. Sleijpen, H. A. van der Vorst, *A Jacobi–Davidson iteration method for linear eigenvalue problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 401–425.
[20] T. Slivnik, G. Tomšič, *A numerical method for the solution of two-parameter eigenvalue problems*, J. Comput. Appl. Math., 15 (1986), pp. 109–115.
[21] G. W. Stewart, J. G. Sun, *Matrix perturbation theory*, Academic Press, Boston, 1990.
[22] P. Van Dooren, *A generalized eigenvalue approach for solving Ricatti equations*, SIAM J. Sci. Comput., 2 (1981), pp. 121–135.
[23] H. Volkmer, *Multiparameter problems and expansion theorems*, Lecture notes in Math. 1356, Springer-Verlag, New York, 1988.