A Jacobi-Davidson type method for the product eigenvalue problem

A Jacobi–Davidson type method for the product eigenvalue problem

Michiel E. Hochstenbach*

Abstract. We propose a Jacobi–Davidson type method to compute selected eigenpairs of the product eigenvalue problem $A_m \cdots A_1 x = \lambda x$, where the matrices may be large and sparse. To avoid difficulties caused by a high condition number of the product matrix, we split up the action of the product matrix and work with several search spaces. We generalize the Jacobi–Davidson correction equation and the harmonic and refined extraction for the product eigenvalue problem. Numerical experiments indicate that the method can be used to compute eigenvalues of product matrices with extremely high condition numbers.

Key words: Product eigenvalue problem, product SVD (PSVD), subspace method, Jacobi–Davidson, correction equation, cyclic matrix, cyclic eigenvalue problem, harmonic extraction, refined extraction.

1 Introduction

We are interested in a partial solution to the product eigenvalue problem, that is, we would like to compute some eigenpairs (λ, x) of a product of matrices

$$\mathbf{A}x = \lambda x, \qquad \mathbf{A} = A_m \cdots A_1, \tag{1.1}$$

where $A_1, \ldots, A_{m-1}, A_m$ are complex, possibly nonsquare, $n_2 \times n_1, \ldots, n_m \times n_{m-1}, n_1 \times n_m$ matrices, respectively.

Well-known special cases are the products A^*A and AA^*, for a possibly nonsquare matrix A, which may be of interest for a partial singular value decomposition (partial SVD). Other applications that may lead to a product eigenvalue problem include the eigenvalue problem for totally nonnegative or pseudosymmetric matrices, queueing models, periodic systems, and Floquet multipliers. Moreover, the SVD of a product of matrices $A_m \cdots A_1$, called the product SVD (PSVD), which arises for instance in Lyapunov exponents for differential or difference equations [15, 22], is closely related to the eigenproblem of a product of the form $A_1^* \cdots A_m^* A_m \cdots A_1$. See [14, 25] and the references therein for a more extensive list of applications.

Three decades ago, Van Loan considered a product/quotient eigenvalue problem of the form $A_4 A_3 (A_1 A_2)^{-1}$ in his Ph.D. thesis (see [24]). The eigenvalue problem, generalized eigenvalue problem, singular value problem and generalized singular value problem are, with some regularity requirements, mathematically all of this form; in

*Version September 29, 2006. Department of Mathematics and Computing Science, Eindhoven University of Technology, PO Box 513, 5600 MB, The Netherlands. m.e.hochstenbach@tue.nl, www.win.tue.nl/∼hochsten. The research of this author was supported in part by NSF grant DMS-0405387.

general however, it may not be a good idea to numerically treat these problems in the mentioned form.

Several methods have been proposed to accurately compute the SVD of products of (small) matrices A_j [4, 5, 7, 15, 22]; Watkins [25] studied GR algorithms to determine the eigenvalue decomposition of products of small matrices. Recently, Kressner [14] proposed an Arnoldi type method, which is suitable for a product eigenvalue problem where the factor matrices are large. One of the crucial techniques involved here is a Krylov–Schur type restart. Although the methods in [13, 14] are for square matrices, they could be adapted for nonsquare products.

The present paper focuses on a Jacobi–Davidson type method to compute some of the eigenpairs of the product of (possibly large sparse) matrices A_j. The method may be seen as a generalization of the Jacobi–Davidson method for the singular value problem [9, 10], which in turn is inspired by the original method by Sleijpen and Van der Vorst [19] for the (standard) eigenvalue problem.

In our method we do not form the product matrix $\mathbf{A}$ explicitly. This is standard practice to avoid loss of efficiency and accuracy; moreover, in some applications the matrices A_j may not be explicitly available, but instead given as functions. In addition, what is special about many applications leading to this problem is that the (sometimes many, say 100–1000) factors each may have condition number of (say) $\mathcal{O}(10^4)$, so that the condition number of $\mathbf{A}$ may well exceed $\mathcal{O}(10^{100})$ or even $\mathcal{O}(10^{1000})$. A huge $\mathbf{A}$-norm means that a matrix-vector multiplication with $\mathbf{A}$ may cause a large error (see, e.g., Higham [6, Ch. 3]). In fact, we may even experience overflow.

These difficulties may hinder any eigenvalue method (such as Arnoldi or Jacobi–Davidson) which acts on the product matrix $\mathbf{A}$, even if we carefully leave $\mathbf{A}$ in a factored form. Moreover, an unacceptably high condition number may affect the speed of convergence as well as the final accuracy. For example, it is well possible that the residual will not get small, even if the eigenpair would already have been correctly identified. An Arnoldi type method may have great difficulties finding small eigenvalues due to the tiny relative gap (although it may profit from a large relative gap for large eigenvalues). Moreover, in the Jacobi–Davidson method solving the correction equation may be very hard due to a high $\mathbf{A}$-condition number.

By splitting up the action of $\mathbf{A}$ in several parts and working with several corresponding search spaces, we hope, at the cost of more memory usage and more computational effort, to gain several advantages:

- by intermediate orthogonalizations the rounding errors are generally reduced (somewhat similar to the superiority of modified Gram-Schmidt over its standard version);
- for an Arnoldi type method the relative gap for the smallest eigenvalues is more favorable;
- for the Jacobi–Davidson type method, as proposed in this paper, the correction equation is of higher dimension but much more well conditioned.

The rest of this paper is organized as follows. In the next section we will recall the close connection between the product eigenproblem and the cyclic eigenvalue problem. Sections 3 and 4 focus on the two main stages for the Jacobi–Davidson type subspace method: the subspace extraction and the subspace expansion. Section 5 will be devoted to various sides of the algorithm such as deflation, preconditioning, restarts, and convergence. After some numerical experiments in Section 6, we will summarize our conclusions in Section 7.

2 The cyclic eigenvalue problem

To start with, let us split up the action of $\mathbf{A}$ in (1.1) into m parts: for an eigenvector $x = x_1$ we have

$$\begin{aligned} A_1 x_1 &= \nu_2 x_2, \\ &\cdots \\ A_{m-1} x_{m-1} &= \nu_m x_m, \\ A_m x_m &= \nu_1 x_1, \end{aligned} \tag{2.1}$$

for some associated vectors $x_2, \ldots, x_m$ and values $\nu_1, \ldots, \nu_m$, such that $\lambda = \nu_1 \cdots \nu_m$. Let $\|\cdot\|$ denote the two-norm. There are two natural scaling options for the vectors x_j:

(a) $\|x_1\| = \cdots = \|x_m\| = 1$; the ν_j satisfy $\nu_1 \cdots \nu_m = \lambda$ but are different in general; or

(b) $\|x_1\| = 1$ and $\nu := \nu_1 = \nu_2 = \cdots = \nu_m = \lambda^{1/m}$, for any mth root of λ, while $x_2, \ldots, x_m$ generally do not have unit norm.

Although we will come across these two possibilities throughout the paper, we will mainly use the latter.

The associated vectors $x_2, \ldots, x_m$ are uniquely defined (up to multiplication by a nonzero scalar) if $\lambda \neq 0$, or equivalently, if all $\nu_j \neq 0$. As we will also see below, the case of a zero eigenvalue is an exception in many aspects; several of the theoretical results will exclude this case. However, it may occur that we are interested in eigenvalues of $\mathbf{A}$ *near* zero, for instance if we want to compute the smallest singular values of $\mathbf{A}$, related to the smallest eigenvalues of $\mathbf{A}^*\mathbf{A}$. We will see further on that computing eigenvalues near zero may in fact be one of the strengths of the proposed method.

Inspired by (2.1), we will employ m search spaces $\mathcal{V}_1, \ldots, \mathcal{V}_m$ of increasing dimension k for the vectors x_1 through x_m, respectively. The corresponding *search matrices*

$$V_1 = [v_1^{(1)} \cdots v_k^{(1)}], \quad \ldots, \quad V_m = [v_1^{(m)} \cdots v_k^{(m)}]$$

contain columns that form orthonormal bases for each of the spaces: for $j = 1, \ldots, m$ we have $\mathcal{V}_j = \operatorname{span}(v_1^{(j)}, \ldots, v_k^{(j)})$. For $n := n_1 + \cdots + n_m$, define the $n \times n$ *cyclic matrix*

$$C = \begin{bmatrix} & & & A_m \\ A_1 & & & \\ & \ddots & & \\ & & A_{m-1} & \end{bmatrix}. \tag{2.2}$$

Then we can rewrite (2.1) as $C\,[x_1, \ldots, x_m]^T = [\nu_1 x_1, \ldots, \nu_m x_m]^T$. Using scaling option (b) this becomes

$$C\,\mathbf{x} = \nu\,\mathbf{x}, \tag{2.3}$$

where we denote $\mathbf{x} := [x_1, \ldots, x_m]^T$; hence we are interested in eigenpairs of C. A basic property of the eigendata of C is the following.

Proposition 2.1 *(cf. [25, Th. 1]) The nonzero λ is an eigenvalue of $\mathbf{A}$ if and only if $\omega^j \lambda$ $(j = 0, \ldots, m-1)$ are all eigenvalues of C, where $\omega = e^{2\pi i/m}$ is an mth root of unity. And if $\mathbf{x} = [x_1, \ldots, x_m]^T$ is an eigenvector corresponding to eigenvalue λ, then*

$$[\omega^{m-1} x_1,\, \omega^{m-2} x_2,\, \cdots,\, \omega x_{m-1},\, x_m]^T$$

is an eigenvector corresponding to $\omega\lambda$.

This result is not necessarily true for zero eigenvalues. The simplest example of this is "SVD case" with $m = 2$: if $\mathbf{A} = A^*A$, where $A = [1\ 0]^T$, then 0 is an eigenvalue of the cyclic matrix C without being an eigenvalue of $\mathbf{A}$. These "ghost" zero eigenvalues are a difficulty of working with the cyclic matrix; cf. also [10].

Motivated by the cyclic matrix, in this paper we will use the index j modulo m, that is, we identify j with $m + j$.

3 Subspace extraction

In the subspace extraction phase, we attempt to get good approximate eigendata from given search spaces. Suppose we have k-dimensional search spaces $\mathcal{V}_1, \ldots, \mathcal{V}_m$ for the vectors $x_1, \ldots, x_m$, respectively, at our disposal. Inspired by (2.1), we would like to determine approximate vectors $\mathcal{V}_j \ni v_j \approx x_j$, $j = 1, \ldots, m$, and approximate values $\mu_1, \ldots, \mu_m$ such that $\theta := \mu_1 \cdots \mu_m \approx \lambda$. In practice we may often be interested in eigenvalues near a specified target τ, which may amongst others be a complex number, "$+\infty$" (for eigenvalues with the largest real part), or "the line on infinity" (for eigenvalues with the largest magnitude).

3.1 Standard extraction

One way to extract approximate eigenpairs is to impose m Galerkin conditions on the m different components:

$$\begin{aligned} A_1 v_1 - \mu_2 v_2 &\perp \widetilde{\mathcal{V}}_2, \\ &\cdots \\ A_{m-1} v_{m-1} - \mu_m v_m &\perp \widetilde{\mathcal{V}}_m, \\ A_m v_m - \mu_1 v_1 &\perp \widetilde{\mathcal{V}}_1, \end{aligned} \tag{3.1}$$

where the $\widetilde{\mathcal{V}}_j$ are the test spaces. The standard extraction consist of taking the search spaces as test spaces: $\widetilde{\mathcal{V}}_j = \mathcal{V}_j$ for $j = 1, \ldots, m$. Since we require $v_j \in \mathcal{V}_j$, we can write $v_j = V_j d_j$ for $d_j \in \mathbb{C}^k$, $j = 1, \ldots, m$. Rearranging the equations gives the projected cyclic eigenvalue problem

$$\begin{bmatrix} & & & V_1^* A_m V_m \\ V_2^* A_1 V_1 & & & \\ & \ddots & & \\ & & V_m^* A_{m-1} V_{m-1} & \end{bmatrix} \begin{bmatrix} d_1 \\ \vdots \\ \vdots \\ d_m \end{bmatrix} = \begin{bmatrix} \mu_1 d_1 \\ \vdots \\ \vdots \\ \mu_m d_m \end{bmatrix}. \tag{3.2}$$

To get a standard cyclic eigenvalue problem, we may take a scaling choice similar to (b) from the previous section: we require that $\|d_1\| = 1$ and $\mu_1 = \cdots = \mu_m =: \mu$; the norm of the other vectors d_j (and hence the v_j) is generally not equal to one. If we write

$$\operatorname{diag}(V_j) = \begin{bmatrix} V_1 & & \\ & \ddots & \\ & & V_m \end{bmatrix},$$

a block diagonal matrix, then (3.2) is the projected cyclic eigenproblem

$$\operatorname{diag}(V_j)^* C \operatorname{diag}(V_j)\, \mathbf{d} = \mu\, \mathbf{d}, \tag{3.3}$$

with $\mathbf{d} = [d_1, \ldots, d_m]^T$. In this case the m different Galerkin conditions (3.1) in fact reduce to one Galerkin condition on the cyclic matrix

$$(C - \mu I)\,\mathrm{diag}(V_j)\,\mathbf{d} \perp \mathrm{span}(\mathrm{diag}(V_j)).$$

From Proposition 2.1, we know that the nonzero eigenvalues of (3.3) actually come in series of m of the form $\mu, \omega\mu, \ldots, \omega^{m-1}\mu$, where $\omega = e^{2\pi i/m}$ and $\theta = \mu^m$ is an eigenvalue of the product matrix

$$(V_1^* A_m V_m)(V_m^* A_{m-1} V_{m-1}) \cdots (V_2^* A_1 V_1) =: H_m H_{m-1} \cdots H_1. \tag{3.4}$$

Therefore, since we are interested in the eigenvalues of $\mathbf{A}$ projected as in (3.4), we may identify all the eigenvalues $\mu, \omega\mu, \ldots, \omega^{m-1}\mu$ of (3.3) and consider only the eigenvalues μ with phase angle φ in the complex plane for which $-\pi/m < \varphi \leq \pi/m$. In fact, solving one of the projected problems (3.3) or (3.4) may also be challenging. Although the dimension of the problem has been reduced, the projected factors may still have high condition numbers. Moreover, by the finite precision arithmetic the m-pairing of the eigenvalues of the projected cyclic matrix may be lost by "ordinary" methods such as the (standard) QR method applied to (3.3). Instead, we can use a periodic QR method as introduced by Bojanczyk, Golub, and Van Dooren [2] and Hench and Laub [8] (see also Kressner [14]) for (3.4) to perform this extraction in a numerically reliable and structure-respecting way.

Summarizing, an eigenvalue θ of (3.4) corresponds to a class of m eigenvalues μ of (3.3) (with $\theta = \mu^m$ for each of these μ) and a class of m tuples $(\mu_1, \ldots, \mu_m)$ of (3.2) (with $\theta = \mu_1 \cdots \mu_m$ for each of these tuples). A value θ, which we call a Ritz value of $\mathbf{A}$ with respect to the search spaces $\mathcal{V}_1, \ldots, \mathcal{V}_m$, is an approximate eigenvalue of $\mathbf{A}$.

A corresponding eigenvector of (3.4), which is a scalar multiple of the d_1 component of (3.2) and (3.3), is called a primitive Ritz vector; it determines a Ritz vector $v_1 = V_1 d_1$ which is an approximate eigenvector for $\mathbf{A}$. The vectors $d_2, \ldots, d_m$ in (3.2) and (3.3), which are multiples of $H_1 d_1, \ldots, H_{m-1} d_{m-1}$, respectively, determine the vectors $v_2, \ldots, v_m$ associated with an approximate eigenvector v_1 by $v_2 = V_2 d_2, \ldots, v_m = V_m d_m$. Note that in (3.2) and (3.3), the vectors $d_2, \ldots, d_m$ (and hence $v_2, \ldots, v_m$) are scaled in the two different ways explained in the previous section. (We remark that given d_1, there is an alternative to determine the vectors $d_2, \ldots, d_m$ which may be suitable in particular for small eigenvalues, see the next section.)

The values $\mu_1, \ldots, \mu_m$ are given in terms of the associated vectors by (cf. (3.1))

$$\mu_1 = \frac{v_1^* A_m v_m}{v_1^* v_1}, \quad \mu_2 = \frac{v_2^* A_1 v_1}{v_2^* v_2}, \quad \ldots, \quad \mu_m = \frac{v_m^* A_{m-1} v_{m-1}}{v_m^* v_m}; \tag{3.5}$$

these values are determined either during the extraction or as a post-processing step without any extra matrix-vector products involving the A_j. These values have the orthogonality properties $A_1 v_1 - \mu_2 v_2 \perp v_2$, $\ldots$, $A_{m-1} v_{m-1} - \mu_m v_m \perp v_m$, $A_m v_m - \mu_1 v_1 \perp v_1$. This implies that this choice for the μ_j also minimizes the norm of the residual

$$r = \begin{bmatrix} A_1 v_1 - \mu_2 v_2 \\ \vdots \\ A_{m-1} v_{m-1} - \mu_m v_m \\ A_m v_m - \mu_1 v_1 \end{bmatrix} \tag{3.6}$$

over all possible values.

The standard extraction has the following justification. Given the $n_j \times k$ search matrices V_j, define the $n_{j+1} \times k$ residual matrices $R_j(K_j)$, where $K_j \in \mathbb{C}^{k\times k}$, by

$$\begin{aligned} R_1(K_1) &= A_1V_1 - V_2K_1, \\ &\cdots \\ R_{m-1}(K_{m-1}) &= A_{m-1}V_{m-1} - V_mK_{m-1}, \\ R_m(K_m) &= A_mV_m - V_1K_m. \end{aligned}$$

One may check that if all of these residual matrices are 0, then the span of the columns of V_1 forms an invariant subspace of $\mathbf{A}$. This implies that every eigenpair of the projected matrix is an eigenpair of the original matrix $\mathbf{A}$. Therefore, for general search spaces $\mathcal{V}_j$, an idea is to take the K_j such that the norms $\|R_j(K_j)\|$ are minimized. The next proposition states that this is done by the $H_j = V_{j+1}^* A_j V_j$; therefore, in this sense the H_j are the best projections of the A_j.

Proposition 3.1 *For given $n_j \times k$ matrices V_j with orthonormal columns, let $H_j = V_{j+1}^* A_j V_j$ (with the "mod m" convention of the index j). Then for all $k \times k$ matrices K_j we have*

$$\|R_j(H_j)\| \le \|R_j(K_j)\|.$$

Moreover, the H_j are unique with respect to the Frobenius norm: $\|R_j(H_j)\|_F \le \|R_j(K)\|_F$ with equality only when $K_j = H_j$.

Proof: The proof is identical to that of [9, Thm. 4.1]; cf. also [16, Thm. 11.4.2] and Proposition 3.3. □

As a generalization of [10, Thm. 2.5], which is on its turn a generalization of Saad's theorem [17, p. 136], we have the following result. Let and $P_{\mathcal{V}_j}$ and P_{v_j} denote the orthogonal projection onto $\mathcal{V}_j$ and span(v_j), respectively. Note that since $v_j \in \mathcal{V}_j$ the projections satisfy

$$P_{\mathcal{V}_j} P_{v_j} = P_{v_j} P_{\mathcal{V}_j} = P_{v_j};$$

moreover, the standard extraction implies $P_{\mathcal{V}_j} A_{j-1} v_{j-1} = \mu_j v_j$. The next theorem expresses the quality of the Ritz vectors in terms of the quality of the search spaces. We take the scaling choice (b) from Section 2 so that $\mu_1 = \cdots = \mu_m =: \mu$.

Theorem 3.2 *Let $(\theta, v_1, \ldots, v_m)$ be a Ritz tuple and $\mu = \theta^{1/m}$ any m-th root of θ, and let $(\lambda, x_1, \ldots, x_m)$ an eigentuple with $\nu = \lambda^{1/m}$ any m-th root of λ. Then*

$$\max_j \sin(v_j, x_j) \le \sqrt{1 + m\,\xi^2 \frac{\gamma^2}{\delta^2}}\ \max_j \sin(\mathcal{V}_j, x_j),$$

where

$$\begin{aligned} \gamma &= \max_j \|P_{\mathcal{V}_{j+1}} A_j (I_{n_j} - P_{\mathcal{V}_j})\|, \\ \xi &= \max_j \|x_j\| \,/\, \min_j \|x_j\|, \\ \delta &= \sigma_{\min}(diag(P_{\mathcal{V}_j} - P_{v_j})(C - \nu I_n) diag(P_{\mathcal{V}_j} - P_{v_j})) \le \min_{\widetilde{\mu}} |\widetilde{\mu} - \nu|, \end{aligned}$$

where $\sigma_{\min}$ denotes the minimal singular value, $\widetilde{\mu}$ ranges over values such that $\widetilde{\mu}^m$ is a Ritz value other than $\mu^m = \theta$.

Proof: Again with $\mathbf{x} = [x_1, \ldots, x_m]^T$, we start with the splitting

$$\mathbf{x} = \mathrm{diag}(P_{v_j})\,\mathbf{x} + \mathrm{diag}(P_{\mathcal{V}_j} - P_{v_j})\,\mathbf{x} + \mathrm{diag}(I_{n_j} - P_{\mathcal{V}_j})\,\mathbf{x},$$

multiply both sides on the left by $\mathrm{diag}(P_{\mathcal{V}_j})(C - \nu I_n)$, and use $(C - \nu I_n)\,\mathbf{x} = 0$ to get

$$\begin{aligned} 0 = \mathrm{vct}((\mu v_{j-1}^* x_{j-1} - \nu v_j^* x_j)\,v_j) \quad &+ \quad \mathrm{diag}(P_{\mathcal{V}_j})(C - \nu I_n)\mathrm{diag}(P_{\mathcal{V}_j} - P_{v_j})\,\mathbf{x} \\ &+ \quad \mathrm{diag}(P_{\mathcal{V}_j})(C - \nu I_n)\mathrm{diag}(I_n - P_{\mathcal{V}_j})\,\mathbf{x}, \end{aligned}$$

where $\mathrm{vct}(a_j)$ stands for $[a_1^T \cdots a_n^T]^T$. Splitting $\mathrm{diag}(P_{\mathcal{V}_j})$ in the second term on the right-hand side into $\mathrm{diag}(P_{\mathcal{V}_j} - P_{v_j})$ and $\mathrm{diag}(P_{v_j})$ and using Pythagoras' Theorem we get

$$\begin{aligned} \|\mathrm{diag}(P_{\mathcal{V}_j})(C - \nu I_n)\mathrm{diag}(I_{n_j} - P_{\mathcal{V}_j})\,\mathbf{x}\| \geq \\ \|\mathrm{diag}(P_{\mathcal{V}_j} - P_{v_j})(C - \nu I_n)\mathrm{diag}(P_{\mathcal{V}_j} - P_{v_j})\,\mathbf{x}\| \end{aligned} \tag{3.7}$$

The left-hand side is the norm of

$$\begin{bmatrix} & & & P_{\mathcal{V}_1} A_m (I_{n_m} - P_{\mathcal{V}_m}) \\ P_{\mathcal{V}_2} A_1 (I_{n_1} - P_{\mathcal{V}_1}) & & & \\ & \ddots & & \\ & & P_{\mathcal{V}_m} A_{m-1} (I_{n_{m-1}} - P_{\mathcal{V}_{m-1}}) & \end{bmatrix} \begin{bmatrix} (I_{n_1} - P_{\mathcal{V}_1})\,x_1 \\ \vdots \\ (I_{n_m} - P_{\mathcal{V}_m})\,x_m \end{bmatrix}$$

hence it is bounded from above by $\gamma\sqrt{m}\,\max_j \|(I_{n_j} - P_{\mathcal{V}_j})\,x_j\|$. The right-hand side of (3.7) is bounded from below by

$$\delta \max_j \|(P_{\mathcal{V}_j} - P_{v_j})\,x_j\|.$$

Since $\|(I_{n_j} - P_{\mathcal{V}_j})\,x_j\| = \sin(\mathcal{V}_j, x_j)\,\|x_j\|$ and $\|(I_{n_j} - P_{v_j})\,x_j\| = \sin(v_j, x_j)\,\|x_j\|$, the result now follows from

$$\frac{\|(I_{n_j} - P_{v_j})\,x_j\|^2}{\|x_j\|^2} = \frac{\|(I_{n_j} - P_{\mathcal{V}_j})\,x_j\|^2}{\|x_j\|^2} + \frac{\|(P_{\mathcal{V}_j} - P_{v_j})\,x_j\|^2}{\|x_j\|^2}.$$

□

This result means that if the search spaces contain an eigenvector and its associated vectors, then they are also detected as a Ritz vector with associated vectors—unless the δ in the previous theorem is zero. This may happen if θ is a multiple Ritz value ("other than" in the statement of the theorem does not necessarily mean "different from"); then the extraction "does not know which Ritz vector to take".

If we have a target τ, we would be inclined to select the approximate eigenpair for which θ is closest to τ. However, as is usual for the standard eigenvalue problem (see for instance [23, p. 282]) and also for the singular value problem [9, 10], the standard extraction may be more suitable for exterior eigenvalues (for instance the ones with maximal magnitude or maximal real part) than for interior ones (closest to a complex number τ in the interior of the spectrum). The main point here is that the Galerkin orthogonality conditions (3.1) do not imply that the norm of the residual (3.6) is small. This motivates the following two alternative extraction processes.

3.2 Refined extraction

The refined extraction for the standard eigenvalue problem was advocated in the Arnoldi context by Jia [11]; the process that will be proposed in this subsection is a generalization

of the refined extraction for the singular value problem as proposed in [10]. The idea behind the refined subspace extraction is the following. Suppose that $\mu_1 \cdots \mu_m$ is a Rayleigh quotient as in Section 3.1 (the μ_j may be equal or not depending on the scaling choice (a) or (b), see Section 2) or equal to a target τ (which may be a prudent choice in case the Rayleigh quotient is still an inaccurate approximation to the eigenvalue, as is quite common in the beginning of the process).

If $\mu_1 \cdots \mu_m$ is equal to an eigenvalue of $\mathbf{A}$, then $C - \mathrm{diag}(\mu_j I_{n_j})$ is singular (cf. (2.3)). Therefore, if $\mu_1 \cdots \mu_m$ is a Ritz value or a target, and we have search spaces $\mathcal{V}_1, \ldots, \mathcal{V}_m$, we may be interested in the vector $[d_1, \ldots, d_m]^T \in \mathbb{C}^{mk}$ of unit length such that

$$\|(C - \mathrm{diag}(\mu_j I))\mathrm{diag}(V_j)\,\mathbf{d}\| = \left\| \begin{bmatrix} -\mu_1 V_1 & & & A_m V_m \\ A_1 V_1 & -\mu_2 V_2 & & \\ & \ddots & \ddots & \\ & & A_{m-1}V_{m-1} & -\mu_m V_k \end{bmatrix} \begin{bmatrix} d_1 \\ \vdots \\ \vdots \\ d_m \end{bmatrix} \right\|$$

is minimal: the smallest right singular vector of $(C - \mathrm{diag}(\mu_j I))\mathrm{diag}(V_j)$. As before, the d_j determine approximations to the associated vectors x_j: $x_j \approx v_j := V_j d_j$. If desired, we can normalize the d_j and consequently also the v_j.

When one of the vectors d_j is zero, the refined extraction does not lead to an approximate vector x_j. This happens in particular in the special and important case that the μ_js are zero, if the eigenvalues of $\mathbf{A}$ closest to the origin are sought (that is, $\tau = 0$). In general, the solution will have only one nonzero d_j component: the component corresponding to the $A_j V_j$ whose minimal singular value is the smallest. For the sake of presentation, we will assume that $d_1 \neq 0$. We now can form appropriate approximate $d_2, \ldots, d_m$ in the following two ways.

- Inspired by

$$\begin{aligned} \lambda v_1 &\approx A_m \cdots A_2 A_1 V_1 d_1 \\ &\approx A_m V_m V_m^* \cdots V_3 V_3^* A_2 V_2 V_2^* A_1 V_1 d_1 = A_m V_m H_{m-1} \cdots H_2 H_1 d_1 \end{aligned}$$

  we can take $d_2 = H_1 d_1, \ldots, d_m = H_{m-1} d_{m-1}$, as in the standard extraction.

- Suppose that the factors $A_1, \ldots, A_m$ are square and invertible, then in view of

$$\begin{aligned} \lambda^{-1} v_1 &\approx A_1^{-1} \cdots A_m^{-1} V_1 d_1 \\ &\approx A_1^{-1} V_2 V_2^* \cdots V_m V_m^* A_m^{-1} V_1 d_1 \approx A_1^{-1} V_2 H_2^{-1} \cdots H_m^{-1} d_1 \end{aligned}$$

  we can define $d_m = H_m^{-1} d_1, \ldots, d_2 = H_2^{-1} d_3$. Note that we can take these approximations for the d_j also in the case that we do not have regularity assumptions on the factors A_j; the only requirement is that the factors H_j be nonsingular. (In the case that this is not met, we can expand the spaces, for instance by the coordinates of the residual (3.6), and perform a new extraction.)

Since in this context we look for the small eigenvalues, multiplying by the A_j and H_j will generally damp out the eigenvector component of interest, while an action with A_j^{-1} and H_j^{-1} will magnify it; therefore, we choose for the second alternative (cf. also [10]). Finally, from the refined Ritz vectors we may take the Rayleigh quotients as derived approximate values, see Section 3.1 and (3.5).

However, as we will also see from the numerical experiments, in the context of the product eigenvalue problem the refined extraction often does not work as well as as

for the standard eigenproblem, which can be explained heuristically as follows. Since the eigenvalue λ of $\mathbf{A}$ corresponds with $\lambda^{1/m}$ of the cyclic matrix, the minimal singular value of the shifted matrix $C - \tau^{1/m} I$ may differ significantly from zero if the target τ is not a very accurate approximation to the wanted eigenvalue (the derivative of the function $\alpha \mapsto \alpha^{1/m}$ in $\alpha = 0$ is infinite for $m > 1$).

3.3 Harmonic extraction

For interior eigenvalues of the standard eigenvalue problem an alternative for the refined extraction is formed by the harmonic Rayleigh–Ritz extraction. This extraction process determines a set of different approximate vectors, the so-called harmonic Ritz vectors. For a matrix C, a search space $\mathcal{V}$, and a target σ, the harmonic Ritz pairs $(\widetilde{\theta}, \widetilde{v})$ are determined by the Galerkin condition (see, for instance, [23, p. 292])

$$(C - \widetilde{\theta} I)\, \widetilde{v} \perp (C - \sigma I)\, \mathcal{V}, \tag{3.8}$$

If the columns of V form an orthonormal basis for $\mathcal{V}$, and if we write $\widetilde{v} = V\widetilde{d}$, this leads to the projected generalized eigenproblem

$$V^*(C - \sigma I)^*(C - \sigma I)V\widetilde{d} = (\widetilde{\theta} - \sigma)\, V^*(C - \sigma I)^* V\widetilde{d}.$$

Since harmonic Ritz pairs satisfy $\|(C - \sigma I)\, \widetilde{v}\| \leq |\widetilde{\theta} - \sigma|$ [23], we are interested in those pairs of which the harmonic Ritz value $\widetilde{\theta}$ is closest to σ, thus ensuring a small residual. With the QR-decomposition $(C - \sigma I)V = QU$, this projected eigenproblem can be written elegantly as

$$U\widetilde{d} = (\widetilde{\theta} - \sigma)\, Q^* V\widetilde{d}. \tag{3.9}$$

We now apply (3.8) to the product/cyclic eigenvalue problem by taking the cyclic matrix (2.2) for C and the "decoupled search matrix" $\mathrm{diag}(V_j)$ for V, substituting the QR-decomposition

$$(C - \mu I)\mathrm{diag}(V_j) = \begin{bmatrix} -\mu V_1 & & & A_m V_m \\ A_1 V_1 & -\mu V_2 & & \\ & \ddots & \ddots & \\ & & A_{m-1}V_{m-1} & -\mu V_m \end{bmatrix} = QU$$

into (3.9). The components $\widetilde{d}_j$ of $\widetilde{d} = [d_1, \ldots, d_m]^T$ can be used to determine approximate vectors $\widetilde{v}_j = V_j \widetilde{d}_j \approx x_j$. For the shift σ we take μ, where μ^m is equal to the target τ or a Rayleigh quotient θ.

If we are interested in the eigenvalues near zero, we can also decouple the m equations as follows. Suppose for the moment that all A_j are square and invertible. Galerkin conditions, as for instance the one in Section 3.1, are generally favorable for exterior eigenvalues. The (interior) eigenvalues near 0 are exterior eigenvalues of $\mathbf{A}^{-1} = A_1^{-1} \cdots A_m^{-1}$. Therefore the idea is to impose Galerkin conditions on modified equations involving the A_j^{-1}.

First, we note that we can write the standard Galerkin conditions ((3.1) with $\widetilde{\mathcal{V}}_j = \mathcal{V}_j$) as

$$\begin{aligned} A_1^{-1} v_2 - \mu_2^{-1} v_1 &\perp A_1^* \mathcal{V}_2, \\ &\cdots \\ A_{m-1}^{-1} v_m - \mu_m^{-1} v_{m-1} &\perp A_{m-1}^* \mathcal{V}_m, \\ A_m^{-1} v_1 - \mu_1^{-1} v_m &\perp A_m^* \mathcal{V}_1. \end{aligned}$$

Instead, we now take different test spaces:

$$\begin{aligned} A_1^{-1}v_2 - \mu_2^{-1}v_1 &\perp A_1^* A_1 \mathcal{V}_1, \\ &\cdots \\ A_{m-1}^{-1}v_m - \mu_m^{-1}v_{m-1} &\perp A_{m-1}^* A_{m-1} \mathcal{V}_{m-1}, \\ A_m^{-1}v_1 - \mu_1^{-1}v_m &\perp A_m^* A_m \mathcal{V}_m. \end{aligned}$$

This is equivalent to requiring

$$\begin{aligned} A_1^{-1}v_2 - \mu_2^{-1}v_1 &\perp_{A_1^* A_1} \mathcal{V}_1, \\ &\cdots \\ A_{m-1}^{-1}v_m - \mu_m^{-1}v_{m-1} &\perp_{A_{m-1}^* A_{m-1}} \mathcal{V}_{m-1}, \\ A_m^{-1}v_1 - \mu_1^{-1}v_m &\perp_{A_m^* A_m} \mathcal{V}_m, \end{aligned}$$

where $x \perp_B y$ means $y^* B x = 0$. Therefore, we have Galerkin conditions on the A_j^{-1} in a different inner product to avoid working with inverses and thus making the computation attractive. The m conditions are equivalent to

$$\begin{aligned} V_1^* A_1^* A_1 V_1 d_1 &= \mu_2 V_1^* A_1^* V_2 d_2, \\ &\cdots \\ V_{m-1}^* A_{m-1}^* A_{m-1} V_{m-1} d_{m-1} &= \mu_m V_{m-1}^* A_{m-1}^* V_m d_m, \\ V_m^* A_m^* A_m V_m d_m &= \mu_1 V_m^* A_m^* V_1 d_1. \end{aligned}$$

When all $V_j^* A V_{j+1}$ are invertible, the $\mu_1 \cdots \mu_m$ are eigenvalues of the product/quotient eigenvalue problem

$$\widetilde{H}_m \cdots \widetilde{H}_1, \qquad \widetilde{H}_j := (V_j^* A_j^* V_{j+1})^{-1} V_j^* A_j^* A_j V_j.$$

After an appropriate scaling of the d_j we can again assume that $\mu_1 = \cdots = \mu_m =: \mu$. The numerical solution can be done efficiently and stably by incrementally computing the QR-decompositions $A_j V_j = Q_{j+1} U_j$ (i.e., in every step the Q_js and U_js are enlarged by one extra column). Then the generalized cyclic eigenvalue problem reads

$$\begin{bmatrix} Q_1^* V_1 & & & \\ & Q_2^* V_2 & & \\ & & \ddots & \\ & & & Q_m^* V_m \end{bmatrix} \begin{bmatrix} d_1 \\ \vdots \\ \vdots \\ d_m \end{bmatrix} = \mu^{-1} \begin{bmatrix} & & & U_m \\ U_1 & & & \\ & \ddots & & \\ & & U_{m-1} & \end{bmatrix} \begin{bmatrix} d_1 \\ \vdots \\ \vdots \\ d_m \end{bmatrix},$$

where one matrix is cyclic with the other block diagonal; our interest is in the vectors corresponding to the smallest $|\mu|$. The corresponding product/quotient eigenvalue problem is

$$(Q_1^* V_1)^{-1} U_m \cdots (Q_3^* V_3)^{-1} U_2 (Q_2^* V_2)^{-1} U_1.$$

The periodic QZ algorithm [2,8,12] is an appropriate way to deal with this problem. As for the refined Ritz vectors, we may take the Rayleigh quotients of the harmonic Ritz vectors as approximate values.

The harmonic extraction for zero target has a justification analogous to Proposition 3.1. Given the $n_j \times k$ search matrices V_j, define the $n_j \times k$ residual matrices

$\widetilde{R}_j(K_j)$, where $K_j \in \mathbb{C}^{k\times k}$, by

$$\begin{aligned}
\widetilde{R}_1(K_1) &= A_1^{-1}V_2 - V_1K_1^{-1},\\
&\cdots\\
\widetilde{R}_{m-1}(K_{m-1}) &= A_{m-1}^{-1}V_m - V_{m-1}K_{m-1}^{-1},\\
\widetilde{R}_m(K_m) &= A_m^{-1}V_1 - V_mK_m^{-1}.
\end{aligned}$$

Denote by $\|\cdot\|_{A_j^*A_j}$ the norm derived from the inner product defined by $(x,y)_{A_j^*A_j} = y^*A_j^*A_jx$. Moreover, we define the $A_j^*A_j$-Frobenius norm of a matrix Z by

$$\|Z\|^2_{A_j^*A_j,\,F} = \mathrm{trace}(Z^*A_j^*A_jZ). \tag{3.10}$$

The next result states that $\|\widetilde{R}_j(K_j)\|_{A_j^*A_j}$ are minimized by the $\widetilde{H}_j$; therefore, in this sense the $\widetilde{H}_j^{-1}$ are the best projections of the A_j^{-1}, which is appropriate for eigenvalues near the origin.

Proposition 3.3 *For given $n_j\times k$ matrices V_j such that the V_j have $A_j^*A_j$-orthonormal columns (that is, $V_j^*A_j^*A_jV_j = I$), let*

$$\widetilde{H}_j = (V_j^*A_j^*V_{j+1})^{-1}V_j^*A_j^*A_jV_j \quad (= (V_j^*A_j^*V_{j+1})^{-1}).$$

Then for all $k\times k$ matrices K_j we have

$$\|\widetilde{R}_j(\widetilde{H}_j)\|_{A_j^*A_j} \le \|\widetilde{R}_j(K_j)\|_{A_j^*A_j}.$$

Moreover, the $\widetilde{H}_j$ are unique with respect to the $A_j^*A_j$-Frobenius norm (3.10).

Proof:

$$\begin{aligned}
\widetilde{R}_j(K_j)^*A_j^*A_j\widetilde{R}_j(K_j) &= V_{j+1}^*V_{j+1} + K_j^{-*}K_j^{-1} - V_{j+1}^*A_jV_jK_j^{-1} - K_j^{-*}V_j^*A_j^*V_{j+1}\\
&= V_{j+1}^*V_{j+1} - \widetilde{H}_j^{-*}\widetilde{H}_j^{-1} + (K_j^{-1}-\widetilde{H}_j^{-1})^*(K_j^{-1}-\widetilde{H}_j^{-1})\\
&= \widetilde{R}_j(\widetilde{H}_j)^*A_j^*A_j\widetilde{R}_j(\widetilde{H}_j) + (K_j^{-1}-\widetilde{H}_j^{-1})^*(K_j^{-1}-\widetilde{H}_j^{-1}).
\end{aligned}$$

Since $(K_j^{-1}-\widetilde{H}_j^{-1})^*(K_j^{-1}-\widetilde{H}_j^{-1})$ is positive semidefinite, it follows that (see, e.g., [23, p. 42])

$$\begin{aligned}
\|\widetilde{R}_j(K_j)\|^2_{A_j^*A_j} &= \lambda_{\max}(\widetilde{R}_j(K_j)^*A_j^*A_j\widetilde{R}_j(K_j))\\
&\ge \lambda_{\max}(\widetilde{R}_j(\widetilde{H}_j)^*A_j^*A_j\widetilde{R}_j(\widetilde{H}_j))\\
&= \|\widetilde{R}_j(\widetilde{H}_j)\|^2_{A_j^*A_j}.
\end{aligned}$$

For uniqueness, we realize, using (3.10), that for $K_j \neq \widetilde{H}_j$

$$\begin{aligned}
\|\widetilde{R}_j(K_j)\|^2_{A_j^*A_j,\,F} &= \mathrm{trace}(\widetilde{R}_j(K_j)^*A_j^*A_j\widetilde{R}_j(K_j))\\
&= \mathrm{trace}(\widetilde{R}_j(\widetilde{H}_j)^*A_j^*A_j\widetilde{R}_j(\widetilde{H}_j)) + \|K_j^{-1}-\widetilde{H}_j^{-1}\|^2_{A_j^*A_j,\,F}\\
&> \|\widetilde{R}_j(\widetilde{H}_j)\|^2_{A_j^*A_j,\,F}.
\end{aligned}$$

□

Note that the condition in the proposition that the A_jV_j should be orthonormal bases expresses the fact that we work with respect to $A_j^*A_j$-inner products.

4 Subspace expansion

In this section we will suppose that we currently have an approximate tuple

$$(\theta, v_1, \ldots, v_m), \qquad \theta = \mu_1 \cdots \mu_m,$$

where $v_1 \in \mathcal{V}_1^{(k)}$, $\ldots$, $v_m \in \mathcal{V}_m^{(k)}$, with $v_j \approx x_j$ for $j = 1, \ldots, m$ (with one of the two scaling options from Section 2), for instance derived by any of the methods in the previous section. We would like to expand each of the m search spaces by one extra direction. Our goal is to find orthogonal updates $t_j \perp v_j$, for $j = 1, \ldots, m$ such that

$$\begin{aligned} A_m(v_m + t_m) &= \nu_1(v_1 + t_1), \\ A_1(v_1 + t_1) &= \nu_2(v_2 + t_2), \\ &\cdots \\ A_{m-1}(v_{m-1} + t_{m-1}) &= \nu_m(v_m + t_m), \end{aligned} \tag{4.1}$$

with $\lambda = \nu_1 \cdots \nu_m$. Since the ν_j are unknown during the process, we will work with the known quantities $\mu_1, \ldots, \mu_m$. We can rewrite (4.1) as

$$(C - \mathrm{diag}(\mu_j I_{n_j})) \begin{bmatrix} t_1 \\ \vdots \\ t_m \end{bmatrix} = -r + \begin{bmatrix} (\nu_1 - \mu_1)\, v_1 \\ \vdots \\ (\nu_m - \mu_m)\, v_m \end{bmatrix} + \begin{bmatrix} (\nu_1 - \mu_1)\, t_1 \\ \vdots \\ (\nu_m - \mu_m)\, t_m \end{bmatrix}. \tag{4.2}$$

We will neglect the last term on the right-hand side, which is $\mathcal{O}(\|[t_1, \ldots, t_m]^T\|^2)$ as shown by the following lemma.

Lemma 4.1 *Let $\nu_1 \cdots \nu_m$ be an eigenvalue, $v_1, \ldots, v_m$ approximate vectors and $\mu_1 \cdots \mu_m$ an approximate eigenvalue satisfying* (3.5)*, and $t_1, \ldots, t_m$ updates as defined in* (4.1)*. For $j = 1, \ldots, m$ we have $|\mu_j - \nu_j| = \mathcal{O}(\|[t_1, \ldots, t_m]^T\|)$.*

Proof: From (4.1) and (3.5) it follows that

$$\nu_j = v_j^* A_{j-1}(v_{j-1} + t_{j-1}) \,/\, v_j^* v_j = \mu_j + \mathcal{O}(\|[t_1, \ldots, t_m]^T\|).$$

□

This forms the foundation of asymptotically quadratic convergence, see Section 5.1. Moreover, we would like to project (4.2) such that the second term on the right-hand side vanishes while the residual is fixed. This is done by the orthogonal projection

$$P = \mathrm{diag}\left(I_{n_j} - \frac{v_j v_j^*}{v_j^* v_j} \right); \tag{4.3}$$

with $\mathbf{t} := [t_1, \ldots, t_m]^T$, the correction equation becomes

$$\mathrm{diag}\left(I_{n_j} - \frac{v_j v_j^*}{v_j^* v_j} \right) (C - \mathrm{diag}(\mu_j I_{n_j})) \mathrm{diag}\left(I_{n_j} - \frac{v_j v_j^*}{v_j^* v_j} \right) \mathbf{t} = -r, \tag{4.4}$$

where it is our goal to solve for the $t_j \perp v_j$. This equation forms a generalization of the JDSVD correction equation for the singular value problem [10]. We may solve this equation inexactly, for instance by a few steps of the (preconditioned) GMRES method, see also Section 5.5.

5 Various issues

5.1 An inexact Newton process

We now generalize a result by Sleijpen and Van der Vorst [20] by showing that this Jacobi–Davidson type method can be seen as an inexact Newton procedure. Define the function

$$F(v_1,\ldots,v_m) = \begin{bmatrix} A_m v_m - \mu_1 v_1 \\ A_1 v_1 - \mu_2 v_2 \\ \vdots \\ A_{m-1} v_{m-1} - \mu_m v_m \end{bmatrix},$$

with

$$\mu_1 = \frac{w_1^* A_m v_m}{w_1^* v_1}, \quad \mu_2 = \frac{w_2^* A_1 v_1}{w_2^* v_2}, \quad \ldots, \quad \mu_m = \frac{w_m^* A_{m-1} v_{m-1}}{w_m^* v_{m-1}},$$

for certain test vectors $w_1,\ldots,w_m$. Moreover, in the definition of F the eigenvectors and associated vectors can still be scaled. Choosing the vectors v_1, …, v_m such that $a_1^* v_1 = \cdots = a_m^* v_m = 1$, for certain "scaling vectors" $a_1,\ldots,a_m$, implies that the Newton updates t_j for v_j should satisfy $t_j \perp a_j$ for $j = 1,\ldots,m$. By a few straightforward calculations, it can be shown that if $w_j^* v_j \neq 0$ and $a_j^* v_j \neq 0$ for $j = 1,\ldots,m$, the equation for the Newton updates $[t_1,\ldots,t_m]^T$ is given by

$$\operatorname{diag}\left(I - \frac{v_j w_j^*}{w_j^* v_j}\right) \begin{bmatrix} -\mu_1 I & & & A_m \\ A_1 & -\mu_2 I & & \\ & \ddots & \ddots & \\ & & A_{m-1} & -\mu_m I \end{bmatrix} \operatorname{diag}\left(I - \frac{v_j a_j^*}{a_j^* v_j}\right) \begin{bmatrix} t_1 \\ \vdots \\ t_m \end{bmatrix} = -r.$$

We get the correction equation (4.4) if we choose the "acceleration" $a_j = w_j = v_j$ in every step.

Since the method can be seen as a Newton process, we know that the asymptotic convergence (that is, the convergence close to an eigenpair) is quadratic if the Jacobian is asymptotically nonsingular. This is confirmed by the following lemma.

Lemma 5.1 *Let $\mu_1 \cdots \mu_m = \lambda \neq 0$ be a simple eigenvalue of $\mathbf{A}$ with associated vectors $x_1 (= x), \ldots, x_m$, then*

$$J := \mathit{diag}\left(I - \frac{x_i x_i^*}{x_i^* x_i}\right) \begin{bmatrix} -\mu_1 I & & & A_m \\ A_1 & -\mu_2 I & & \\ & \ddots & \ddots & \\ & & A_{m-1} & -\mu_m I \end{bmatrix} \mathit{diag}\left(I - \frac{x_i x_i^*}{x_i^* x_i}\right)$$

is invertible from $\operatorname{span}(x_1)^\perp \times \cdots \times \operatorname{span}(x_m)^\perp$ *to itself.*

Proof: Suppose $J\,[t_1,\ldots,t_m]^T = 0$, where $t_j \perp x_j$. Then there are α_j such that

$$\begin{aligned} A_m t_m &= \mu_1 t_1 + \alpha_1 x_1, \\ A_1 t_1 &= \mu_2 t_2 + \alpha_2 x_2, \\ &\cdots \\ A_{m-1} t_{m-1} &= \mu_m t_m + \alpha_m x_m. \end{aligned}$$

From these equations it follows that $\mathbf{A}t_1 = \lambda t_1 + \beta x_1$ for a $\beta \in \mathbb{C}$. Therefore

$$\left(I - \frac{x_1 x_1^*}{x_1^* x_1}\right) \mathbf{A} \left(I - \frac{x_1 x_1^*}{x_1^* x_1}\right) t_1 = \lambda t_1.$$

Since $\lambda \neq 0$ is assumed to be simple, we must have $t_1 = 0$. Then, from $0 = \mu_2 t_2 + \alpha_2 x_2$, $\mu_2 \neq 0$, and $t_2 \perp x_2$ it follows that $t_2 = 0$. With similar arguments it now follows that all the t_j $(j = 1, \ldots, m)$ are zero. So J is injective and hence a bijection. $\square$

In practice we often solve the correction equation (4.4) inexactly; the resulting process is sometimes called an inexact accelerated Newton process. This method typically displays asymptotically linear convergence.

5.2 Startup

Since Jacobi–Davidson type methods can be seen as inexact accelerated Newton processes, they often have favorable asymptotic convergence qualities. However, the convergence behavior far away from the solution may be less attractive. In practice it is sometimes favorable to start with sensible initial search spaces, to avoid spending numerical effort (many matrix-vector products) on solving the correction equation for a poor approximation.

For instance, we can perform a startup of the method by an Arnoldi approach for the product eigenvalue problem, cf. [14]. This method starts with a normalized vector $V_1 = v_1^{(1)}$ and incrementally computes $m-1$ QR-decompositions and one Hessenberg relation:

$$\begin{aligned} A_1 V_1^{(k)} &= V_2^{(k)} R_1^{(k)}, \\ &\cdots \\ A_{m-1} V_{m-1}^{(k)} &= V_m^{(k)} R_{m-1}^{(k)}, \\ A_m V_m^{(k)} &= V_1^{(k+1)} H_m^{(k)}. \end{aligned}$$

Here the $k \times k$ matrices $R_j^{(k)}$ are right upper triangular while the $(k+1) \times k$ matrix $H_1^{(k)}$ is upper Hessenberg. This implies that

$$\mathbf{A} V_1^{(k)} = V_1^{(k+1)} H_m^{(k)} R_{m-1}^{(k)} \cdots R_1^{(k)}.$$

One can easily see that

$$\begin{aligned} V_1^{(k)} &= \mathcal{K}_k(A_m \cdots A_1,\ v_1), \\ V_2^{(k)} &= \mathcal{K}_k(A_1 A_m \cdots A_2,\ A_1 v_1), \\ &\cdots \\ V_m^{(k)} &= \mathcal{K}_k(A_{m-1} \cdots A_1 A_m,\ A_{m-1} \cdots A_1 v_1), \end{aligned}$$

where $\mathcal{K}_k(B, v)$ denotes the Krylov space $\operatorname{span}\{v, Bv, \ldots, B^{k-1}v\}$. After a certain number of steps of Arnoldi expansion, we may perform a standard, harmonic, or refined subspace extraction as discussed in Section 3, and proceed with the Jacobi–Davidson type method.

5.3 Deflation

Suppose we have computed an eigenpair λ with eigenvector $x = x_1$, and associated vectors $x_2, \ldots, x_m$. Then we may continue with the deflated product eigenvalue problem

$$\widetilde{\mathbf{A}}x = \lambda x, \qquad \widetilde{\mathbf{A}} = \widetilde{A}_m \cdots \widetilde{A}_1,$$

where

$$\begin{aligned}
\widetilde{A}_1 &= \left(I - \frac{x_2 x_2^*}{x_2^* x_2}\right) A_1 \left(I - \frac{x_1 x_1^*}{x_1^* x_1}\right), \\
&\cdots \\
\widetilde{A}_{m-1} &= \left(I - \frac{x_m x_m^*}{x_m^* x_m}\right) A_{m-1} \left(I - \frac{x_{m-1} x_{m-1}^*}{x_{m-1}^* x_{m-1}}\right), \\
\widetilde{A}_m &= \left(I - \frac{x_1 x_1^*}{x_1^* x_1}\right) A_m \left(I - \frac{x_m x_m^*}{x_m^* x_m}\right).
\end{aligned}$$

This problem has the same eigenvalues as the original problem involving $\mathbf{A}$, except for the fact that the eigenvalue λ has been replaced by the eigenvalue 0. Indeed, if $(\widetilde{\lambda}, \widetilde{y}_1, \ldots, \widetilde{y}_m)$ is another eigentuple of $\widetilde{\mathbf{A}}$, then we first define $y_j := \left(I - \frac{x_j x_j^*}{x_j^* x_j}\right) \widetilde{y}_j$. One can check that $[x_1 \;\; y_1]$, $\ldots$, $[x_m \;\; y_m]$ are orthogonal bases with the projected matrices $[x_2 \;\; y_2]^* A_1 [x_1 \;\; y_1]$, $\ldots$, $[x_1 \;\; y_1]^* A_m [x_m \;\; y_m]$ upper triangular. We see that this deflation gives rise to a partial *periodic Schur decomposition* of $\mathbf{A}$, cf. [14] and the references therein. Eigentuples for $\mathbf{A}$ are easily determined from the product of the (small) upper triangular matrices.

5.4 Restarts

In practice we often need restarts to make the algorithm computationally attractive. If the search spaces $\mathcal{V}_1, \ldots, \mathcal{V}_m$ have dimension `maxdim`, we restart with orthonormal bases for the best `mindim` approximate vectors. What "the best" vectors are depends on the extraction technique employed. Another option that is built in our code is to use a thick restart technique similar to the one for the standard eigenvalue problem described in [21].

5.5 Preconditioning

To precondition the correction equation (4.4), we have to solve, for a $b \in \mathbb{C}^n$, $\mathbf{t} = [t_1, \ldots, t_m]^T$, and $t_j \perp v_j$,

$$\operatorname{diag}\left(I - \frac{v_j v_j^*}{v_j^* v_j}\right) \cdot M \cdot \operatorname{diag}\left(I - \frac{v_j v_j^*}{v_j^* v_j}\right) \mathbf{t} = b,$$

where M is a relatively cheaply invertible approximation to the shifted cyclic matrix $C - \operatorname{diag}(\mu_j I)$. One may check that the solution is given by

$$\mathbf{t} = \left(I - M^{-1}\operatorname{diag}(v_j) \left(\operatorname{diag}(v_j)^* M^{-1} \operatorname{diag}(v_j)\right)^{-1} \operatorname{diag}(v_j)^*\right) M^{-1} b;$$

therefore, the main computational effort amounts to applying the preconditioner m times on the v_j per outer iteration, and in addition once per inner iteration.

In general it may not be a trivial task to come up with a sensible M, although for instance an inexact LU-decomposition of the shifted cyclic matrix could be tried;

cf. also [1]. In the special case that we are interested in eigenvalues with minimal magnitude (that is, target $\tau = \mu_1 = \cdots = \mu_m = 0$), and all the matrices $A_1, \ldots, A_m$ are square and invertible and have preconditioners $M_j \approx A_j$ for $j = 1, \ldots, m$, then

$$M^{-1} = \begin{bmatrix} & M_1^{-1} & & \\ & & \ddots & \\ & & & M_{m-1}^{-1} \\ M_m^{-1} & & & \end{bmatrix} \tag{5.1}$$

is a natural preconditioner for C. In this case the action of the preconditioner decouples, making it computationally more attractive.

5.6 The product SVD

If we are interested in the (partial) product SVD of $\mathbf{A} = A_m \cdots A_1$, we can apply the proposed method to $\mathbf{A}^*\mathbf{A} = A_1^* \cdots A_m^* A_m \cdots A_1$. We compute one or more eigentuples of the form $(\lambda, x_1, \ldots, x_{2m})$; then $(\sqrt{\lambda}, x_1, x_{m+1})$ are singular triples. Only in the case that the condition number of $\mathbf{A}$ is modest, an alternative for this approach is to use the JDSVD method [9,10] on $\mathbf{A}$, which employs a two-sided projection technique on $\mathbf{A}$ and works with the augmented matrix

$$\begin{bmatrix} 0 & A_m \cdots A_1 \\ A_1^* \cdots A_m^* & 0 \end{bmatrix}$$

in the subspace expansion phase.

5.7 Pseudocode

Pseudocode for the Jacobi–Davidson type method for the product eigenvalue problem, which we will abbreviate by JDPROD for use in the next section, is given in Algorithm 1.

In line 3, RGS stands for repeated Gram–Schmidt or any other numerically stable method to add a vector to an already orthonormal set of vectors. For simplicity we omit a possible startup procedure (Section 5.2), deflation (Section 5.3), and restarts (Section 5.4) . Note that if we use the Arnoldi type startup procedure, we only need one initial vector v_1 instead of m vectors $v_1, \ldots, v_m$. The computation of the residual in line 8 involves no extra matrix-vector products. In line 10, the shift μ on the left-hand side relates to the Rayleigh quotients (3.5) in the sense that $\mu^m = \mu_1 \ldots \mu_m$.

6 Numerical experiments

For the numerical experiments, we take the following parameters, unless mentioned otherwise. The maximum dimension of the search spaces is 20, after which we restart with dimension 10. We solve the correction equations with 10 steps of unpreconditioned GMRES, and solve the projected problems with the standard extraction. We take the target as the shift in the left-hand side of the correction equation, unless the residual norm is less than 0.01, then we take the Rayleigh quotient. The tolerance for the outer iteration is 10^{-6} with maximally 200 steps. The m starting vectors $v_1, \ldots, v_m$ are random.

Algorithm 1 A Jacobi–Davidson type method for the product eigenproblem

Input: A device to compute $A_j x$ for $x \in \mathbb{C}^{n_j}$, starting vectors v_j $(j = 1, \ldots, m)$, a target τ, a tolerance ε

Output: An approximate eigenpair (θ, u) of $\mathbf{A} = A_m \cdots A_1$ with a prescribed tolerance ε on the corresponding cyclic matrix C, and $\theta \approx \tau$

1: $t_j = v_j,\ V_j^{(0)} = [\,] \qquad (j = 1, \ldots, m)$
2: **for** $k = 1, 2, \ldots$ **do**
3: $\quad \mathrm{RGS}(V_j^{(k-1)}, t_j) \to V_j^{(k)} \qquad (j = 1, \ldots, m)$
4: $\quad$ Compute kth columns of $W_j^{(k)} = A_j V_j^{(k)} \qquad (j = 1, \ldots, m)$
5: $\quad$ Compute kth rows and columns of $H_j^{(k)} = (V_{j+1}^{(k)})^* A_j V_j^{(k)} = (V_{j+1}^{(k)})^* W_j^{(k)}$
6: $\quad$ Extract a Ritz tuple $(\mu_1, \ldots, \mu_m, d_1, \ldots, d_m)$ with $\theta = \mu_1 \cdots \mu_m \approx \tau$ (standard, refined, harmonic extraction), normalize the d_j if desired
7: $\quad v_j = V_j^{(k)} d_j \qquad (j = 1, \ldots, m)$
8: $\quad r = (C - \mathrm{diag}(\mu_j I_{n_j}))[v_1, \ldots, v_m]^T$
9: $\quad$ Stop if $\|r\| \le \varepsilon$
10: $\quad$ Solve (approximately) $t_j \perp v_j$ from:
$$\left(I_{n_j} - \tfrac{v_j v_j^*}{v_j^* v_j}\right)(C - \mu I)\left(I_{n_j} - \tfrac{v_j v_j^*}{v_j^* v_j}\right)[t_1, \ldots, t_m]^T = -r$$
11: **end for**

Experiment 6.1 For the first experiment we take a typical challenge for a product eigenvalue method:

$$D = \mathrm{diag}(\sqrt[4]{1}, \sqrt[4]{2}, \ldots, \sqrt[4]{1000}), \quad A_1 = A_3 = 10^{10} D, \quad A_2 = A_4 = 10^{-10} D.$$

JDPROD finds the largest eigenvalue ($\lambda = 1000$) quickly, but due to the enormous (difference in) condition numbers, the residual does not get smaller than $\mathcal{O}(10^{-4})$.

Experiment 6.2 Next, we take an example similar to one from [14], but of size 1000×1000:

$$A_1 = A_2 = A_3 = \mathrm{diag}(1, 10^{-1}, 10^{-2}, \ldots, 10^{-999}).$$

Within 9 outer iterations, JDPROD with a tolerance of 10^{-8} finds the 7 largest eigenvalues with at least 14 correct digits.

Experiment 6.3 The following experiment is also similar to one from [14]. To compute the Hankel singular values of the discretized model of a clamped beam we may compute the positive square roots of the eigenvalues of a matrix product of the form $\mathbf{A} = SR^*RS^*$. The Cholesky factors R and S (size 348×348) can be obtained from the model reduction benchmark collection [3]. JDPROD finds the 10 largest eigenvalues of $\mathbf{A}$:

$\lambda_1 \approx 5.7\,e\,6, \quad \lambda_2 \approx 4.7\,e\,6, \quad \lambda_3 \approx 7.4\,e\,4, \quad \lambda_4 \approx 7.1\,e\,4, \quad \lambda_5 \approx 2.0\,e\,3,$
$\lambda_6 \approx 1.9\,e\,3, \quad \lambda_7 \approx 1.1\,e\,2, \quad \lambda_8 \approx 1.0\,e\,2, \quad \lambda_9 \approx 12.94, \quad \lambda_{10} \approx 9.7.$

in 24 outer iterations. If we just look for the eigenvalue closest to $\tau = 13$ (that is, λ_9), the standard extraction needs 23 iterations, and the refined extraction 80 iterations. Also other (unreported) experiments indicate that for the product eigenvalue problem the performance of the refined extraction is generally weaker than that of the standard and harmonic extractions; for an explanation see the remarks at the end of Section 3.2.

Experiment 6.4 Another challenge is formed by a huge condition number of $\mathbf{A}$; therefore we take $A_1 = \cdots = A_{30} = \mathrm{diag}(1{:}100)$, such that $\kappa(\mathbf{A}) = 10^{60}$. Note that the corresponding cyclic eigenproblem has dimension 3000. JDPROD has no difficulties finding the 5 largest eigenvalues (respectively after 20, 30, 39, 48, and 56 outer iterations).

Experiment 6.5 For the following experiment, we take nonsquare matrices. Let A_1, A_2, and A_3 be matrices with uniformly random elements from $[-\frac{1}{2}, \frac{1}{2}]$ of size 1001×1000, 1002×1001, and 1000×1002, respectively. JDPROD finds the largest eigenvalue (in absolute value sense) $\lambda \approx 588 + 493i$ in 71 outer iterations. With target $\tau = 600 + 500i$ (taking the closest Ritz value to the target with the standard extraction in every iteration), it finds the same value in 56 iterations.

Experiment 6.6 Finally, we take three random 1000×1000 matrices and aim at the eigenvalue closest to the origin (target $\tau = 0$). JDPROD with the standard extraction without preconditioning fails to converge in 200 outer iterations. If we take an inexact LU decomposition (drop tolerance $\delta = 10^{-3}$) of the different factors as in (5.1), we find $\lambda \approx -0.0013 - 0.014i$ in 11 iterations. For $\delta = 10^{-4}$ convergence takes one fewer iteration, while for drop tolerance $\delta = 10^{-2}$ there is no convergence. However, when we use the harmonic extraction instead of the standard extraction with a preconditioner with $\delta = 10^{-2}$, we get convergence in 37 steps; this is an example where the harmonic extraction is indeed better suited for interior eigenvalues.

7 Conclusions

The product eigenvalue problem is a numerical challenge. The condition number of $\mathbf{A}$ is often enormous, effectively forcing the use of "intermediate" search spaces and accordingly more memory usage. The standard extraction is sometimes not favorable for interior eigenvalues, while for nonzero target the harmonic and refined extraction need a decomposition of the projected cyclic matrix, of which the size may still be considerable if the number of factors m is large. Another aspect of the cyclic form in comparison with the product form is that the large eigenvalues of the cyclic form are relatively more clustered, while the small eigenvalues are relatively more separated.

We proposed a Jacobi–Davidson type method for the product eigenvalue problem. The method is designed for product matrices with a large condition number. There are three main advantages of the presented techniques. First, in particular the harmonic extraction method tends to give better approximations for interior eigenvalues; the method decouples for a zero target, making its computation more attractive. Second, the correction equation can be seen as an inexact Rayleigh quotient or inexact inverse iteration, which in practice is often needed to compute interior eigenvalues. Third, we can use preconditioning in the correction equation. A zero target is also practical here, since in that case the preconditioner may decouple. Working with a sensible preconditioner for the shifted cyclic matrix with nonzero shift may become computationally less attractive for a nonzero target. On the other hand, preconditioning $\mathbf{A}$ directly seems even less attractive. We note that if one looks for the largest eigenvalues and if these are relatively well-separated, an Arnoldi type method for the product eigenvalue problem [14] may be the method of choice.

The method proposed here reduces to the Jacobi–Davidson method for the standard eigenvalue problem [19] in the case $\mathbf{A} = A$ (the product of just one matrix), and to the Jacobi–Davidson type method for the singular value problem (JDSVD, [9, 10]) in the case $\mathbf{A} = A^*A$ (the product of A and its conjugate transpose). MATLAB code is available from the author upon request.

Acknowledgments I would like to thank Daniel Kressner for interesting discussions and the referee for helpful comments.

References

[1] P. Amodio, J. R. Cash, G. Roussos, R. W. Wright, G. Fairweather, I. Gladwell, G. L. Kraut, and M. Paprzycki, *Almost block diagonal linear systems: sequential and parallel solution techniques, and applications*, Numer. Linear Algebra Appl., 7 (2000), pp. 275–317.

[2] A. Bojanczyk, G. H. Golub, and P. Van Dooren, *The periodic Schur decomposition; algorithm and applications.*, in Proc. SPIE Conference, vol. 1770, 1992, pp. 31–42.

[3] Y. Chahlaoui and P. Van Dooren, *Benchmark examples for model reduction of linear time-invariant dynamical systems*, in Dimension Reduction of Large-Scale Systems, P. Benner, V. Mehrmann, and D. C. Sorensen, eds., vol. 45 of Lecture Notes in Computational Science and Engineering, Springer, Heidelberg, 2005, pp. 379–392.

[4] Z. Drmač, *Accurate computation of the product-induced singular value decomposition with applications*, SIAM J. Numer. Anal., 35 (1998), pp. 1969–1994.

[5] G. Golub, K. Sølna, and P. Van Dooren, *Computing the SVD of a general matrix product/quotient*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 1–19.

[6] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2002.

[7] M. T. Heath, A. J. Laub, C. C. Paige, and R. C. Ward, *Computing the singular value decomposition of a product of two matrices*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 1147–1159.

[8] J. J. Hench and A. J. Laub, *Numerical solution of the discrete-time periodic Riccati equation*, IEEE Trans. Automat. Control, 39 (1994), pp. 1197–1210.

[9] M. E. Hochstenbach, *A Jacobi–Davidson type SVD method*, SIAM J. Sci. Comput., 23 (2001), pp. 606–628.

[10] ——, *Harmonic and refined extraction methods for the singular value problem, with applications in least squares problems*, BIT, 44 (2004), pp. 721–754.

[11] Z. Jia, *Refined iterative algorithms based on Arnoldi's process for large unsymmetric eigenproblems*, Linear Algebra Appl., 259 (1997), pp. 1–23.

[12] D. Kressner, *An efficient and reliable implementation of the periodic QZ algorithm.*, in IFAC Workshop on Periodic Control Systems, 2001.

[13] ——, *Numerical Methods and Software for General and Structured Eigenvalue Problems*, PhD thesis, Technische Universität Berlin, 2004.

[14] ——, *A periodic Krylov-Schur algorithm for large matrix products*, Numer. Math., 103 (2006), pp. 461–483.

[15] S. Oliveira and D. E. Stewart, *Exponential splittings of products of matrices and accurately computing singular values of long products*, Linear Algebra Appl., 309 (2000), pp. 175–190.

[16] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1998. Corrected reprint of the 1980 original.

[17] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press, Manchester, UK, 1992.

[18] H. D. Simon and H. Zha, *Low-rank matrix approximation using the Lanczos bidiagonalization process with applications*, SIAM J. Sci. Comput., 21 (2000), pp. 2257–2274.

[19] G. L. G. Sleijpen and 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] ——, *The Jacobi–Davidson method for eigenvalue problems and its relation with accelerated inexact Newton schemes*, in Iterative Methods in Linear Algebra, II., S. D. Margenov and P. S. Vassilevski, eds., vol. 3 of IMACS Series in Computational and Applied Mathematics, New Brunswick, NJ, U.S.A., 1996, IMACS, pp. 377–389.

[21] A. Stathopoulos and Y. Saad, *Restarting techniques for the (Jacobi–) Davidson symmetric eigenvalue methods*, Electron. Trans. Numer. Anal., 7 (1998), pp. 163–181.

[22] D. E. Stewart, *A new algorithm for the SVD of a long product of matrices and the stability of products*, Electron. Trans. Numer. Anal., 5 (1997), pp. 29–47.

[23] G. W. Stewart, *Matrix algorithms. Vol. II*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2001.

[24] C. F. Van Loan, *A general matrix eigenvalue algorithm*, SIAM J. Numer. Anal., 12 (1975), pp. 819–834.

[25] D. Watkins, *Product eigenvalue problems*, SIAM Rev., 47 (2005), pp. 3–40.