Numer Algor (2014) 66:681–703
DOI 10.1007/s11075-013-9754-3

ORIGINAL PAPER

An indefinite variant of LOBPCG for definite matrix pencils

Daniel Kressner · Marija Miloloža Pandur · Meiyue Shao

Received: 1 May 2013 / Accepted: 1 August 2013 / Published online: 14 August 2013

Abstract In this paper, we propose a novel preconditioned solver for generalized Hermitian eigenvalue problems. More specifically, we address the case of a definite matrix pencil $A - \lambda B$, that is, A, B are Hermitian and there is a shift λ_0 such that $A - \lambda_0 B$ is definite. Our new method can be seen as a variant of the popular LOBPCG method operating in an indefinite inner product. It also turns out to be a generalization of the recently proposed LOBP4DCG method by Bai and Li for solving product eigenvalue problems. Several numerical experiments demonstrate the effectiveness of our method for addressing certain product and quadratic eigenvalue problems.

Keywords Eigenvalue · Definite matrix pencil · Minimization principle · LOBPCG

1 Introduction

The Locally Optimal Block Preconditioned Conjugate Gradient method (LOBPCG) proposed by Knyazev [16] aims at computing the smallest eigenvalues of a matrix pencil

$$A - \lambda B, \tag{1}$$

D. Kressner · M. Miloloža Pandur · M. Shao (✉)
ANCHP, MATHICSE, EPF Lausanne, 1015 Lausanne, Switzerland
e-mail: meiyue.shao@epfl.ch

D. Kressner
e-mail: daniel.kressner@epfl.ch

M. Miloloža Pandur
Department of Mathematics, J. J. Strossmayer University of Osijek,
31000 Osijek, Croatia
e-mail: mmililoz@mathos.hr

where $A, B \in \mathbb{C}^{n\times n}$ are both Hermitian and, additionally, B is positive definite. LOBPCG is particularly well suited for the situation when A, B are the stiffness and mass matrices from the finite element discretization of an elliptic PDE-eigenvalue problem. In this case, the Lanczos method [3] applied to A (in the inner product induced by B) can be expected to converge very slowly towards the smallest eigenvalues. In principle, this issue can be resolved by applying the Lanczos method to A^{-1}. However, depending on the application, the exact inversion of A may be considered too costly. In contrast to the Lanczos method, LOBPCG allows for the effective use of a preconditioner T instead of A^{-1}. For example, a multigrid preconditioner T results in convergence rates that do not depend on the mesh width of the finite element discretization [17].

In this paper, we consider variants of LOBPCG suitable for *indefinite* B. More specifically, we consider the case where A, B are Hermitian and there is a shift λ_0 such that $A - \lambda_0 B$ is (positive) definite. A matrix pencil with this property is called *(positive) definite*. Eigenvalue problems with positive definite matrix pencils arise, for example, in computational quantum chemistry [4] and mechanics [37].

The theoretical properties of positive definite matrix pencils have been studied intensively, see [37] and the references therein. Also, a number of numerical algorithms have been proposed. Often, these algorithms can be seen as "indefinite extensions" of standard symmetric eigenvalue solvers, that is, they operate in the indefinite inner product induced by B. Not all of these extensions require (or make use of) the definiteness of the pencil. Examples of such extensions include a Rayleigh quotient method [22], indefinite Lanczos methods [3, 22], indefinite Jacobi algorithms [13, 36], as well as many structure-preserving methods for Hamiltonian matrices, see [7] for an overview. In this work, we will propose an indefinite extension of LOBPCG. We are not aware of any other preconditioned eigenvalue solver tailored to definite pencils.

This work has been inspired by the work of Bai and Li [4–6], who proposed an LOBPCG-like algorithm for computing the smallest eigenvalues of the matrix product KM, where K, M are Hermitian positive semidefinite matrices and one of them is definite. This problem is equivalent to computing the eigenvalues closest to zero of the matrix pencil

$$\begin{bmatrix} K & 0 \\ 0 & M \end{bmatrix} - \lambda \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}. \tag{2}$$

Clearly, this pencil is positive definite (with shift $\lambda_0 = 0$), when both K and M are positive definite.

The rest of this paper is organized as follows. In Section 2, we recall existing results on eigenvalue interlacing and trace minimization properties for definite matrix pencils. Section 3 contains our newly proposed algorithm, along with detailed implementation remarks. Moreover, a connection to the preconditioned inverse iteration will provide some insight into the expected convergence behavior and give rise to a variant of the algorithm that incorporates two preconditioners. When applied to the product eigenvalue problem (2), it will be shown in Section 4 that this algorithm essentially coincides with the method by Bai and Li, after some reorganization.

In Section 5, we illustrate the numerical behavior of our new algorithm for several examples.

2 Preliminaries

In the following, we briefly recall some known facts for definite matrix pencils, see, e.g., the textbooks [11, 37] for more detailed treatments.

The most fundamental property of a definite pencil is that it can be diagonalized by a congruence transformation.

Theorem 2.1 ([21, 24]) *Let $A - \lambda B$ be an $n \times n$ positive definite matrix pencil such that B has inertia*[1] $\mathsf{In}(B) = (n_+, n_-, n_0)$. *Then there is an invertible matrix W such that*

$$W^H A W - \lambda W^H B W = \begin{bmatrix} \Lambda_+ & & \\ & -\Lambda_- & \\ & & I_{n_0} \end{bmatrix} - \lambda \begin{bmatrix} I_{n_+} & & \\ & -I_{n_-} & \\ & & 0_{n_0} \end{bmatrix}, \tag{3}$$

where $\Lambda_+ = \mathrm{diag}(\lambda_1^+, \ldots, \lambda_{n_+}^+)$, $\Lambda_- = \mathrm{diag}(\lambda_1^-, \ldots, \lambda_{n_-}^-)$ *with*

$$\lambda_{n_-}^- \le \cdots \le \lambda_1^- < \lambda_1^+ \le \cdots \le \lambda_{n_+}^+. \tag{4}$$

Clearly, the decomposition (3) implies that $A - \lambda_0 B$ is positive definite if and only if $\lambda_1^- < \lambda_0 < \lambda_1^+$ holds. In the following, any such λ_0 will be called *definitizing shift* and interval $(\lambda_1^-, \lambda_1^+)$ will be called *definiteness interval.*

Definition 2.2 Let $B \in \mathbb{C}^{n\times n}$ be Hermitian. A vector $x \in \mathbb{C}^n$ is called *B-positive*, *B-negative*, *B-neutral* if $x^H Bx > 0$, $x^H Bx < 0$, $x^H Bx = 0$, respectively.

Since the columns of W are eigenvectors of $A - \lambda B$, it follows from (3) that an eigenvector x belonging to any of the eigenvalues $\lambda_1^+, \ldots, \lambda_{n_+}^+$ $(\lambda_1^-, \ldots, \lambda_{n_-}^-)$ is B-positive (B-negative). In particular, any eigenvector x belonging to a finite eigenvalue of $A - \lambda B$ cannot be B-neutral. This allows to always normalize x such that $|x^H Bx| = 1$. Moreover, eigenvectors x_1, x_2 belonging to different eigenvalues are B-orthogonal: $x_1^H Bx_2 = 0$.

2.1 Eigenvalue interlacing and trace minimization

The classical Cauchy interlacing theorem for Hermitian matrices extends to definite pencils.

[1] Note that the inertia $\mathsf{In}(B)$ of a Hermitian matrix B is defined as the triple containing the number of positive, negative, and zero eigenvalues of B.

Theorem 2.3 *Let $A - \lambda B$ be an $n \times n$ positive definite matrix pencil and let $U \in \mathbb{C}^{n \times p}$ have full column rank. Then the compressed matrix pencil $U^H AU - \lambda U^H BU$ is also positive definite; hence its eigenvalues are real and can be ordered as follows:*

$$\theta_{p_-}^- \leq \cdots \leq \theta_1^- < \theta_1^+ \leq \cdots \leq \theta_{p_+}^+, \tag{5}$$

with $\mathrm{In}(U^H BU) = (p_+, p_-, p_0)$. Moreover, we have the eigenvalue interlacing properties

$$\lambda_i^+ \leq \theta_i^+ \leq \lambda_{i+n-p}^+ \quad \textit{for} \quad 1 \leq i \leq p_+, \tag{6}$$

$$\lambda_j^- \geq \theta_j^- \geq \lambda_{j+n-p}^- \quad \textit{for} \quad 1 \leq j \leq p_-, \tag{7}$$

where we formally set $\lambda_i^+ = +\infty$ for $i > n_+$ and $\lambda_j^- = -\infty$ for $j > n_-$.

Proof The statement of this theorem is a variation of a result in [20, Thm 2.1], from which it can be easily deduced. It also follows directly from the minimax principle [22, Thm 3.1] for definite matrix pencils. □

As explained in [20, 24], the result of Theorem 2.3 can be used to prove the following Ky-Fan-type theorem, also known as *trace minimization principle*.

Theorem 2.4 *Let $A - \lambda B$ be an $n \times n$ positive definite matrix pencil with the eigenvalues ordered as in (4). Moreover, let*

$$J_k = \begin{bmatrix} I_{k_+} & \\ & -I_{k_-} \end{bmatrix}$$

for some integers k_+, k_- satisfying $(k_+, k_-, 0) \leq \mathrm{In}(B)$, where the inequality is understood elementwise. Then

$$\min_{\substack{X \in \mathbb{C}^{n \times k} \\ X^H BX = J_k}} \operatorname{trace}(X^H AX) = \sum_{i=1}^{k_+} \lambda_i^+ - \sum_{j=1}^{k_-} \lambda_j^-. \tag{8}$$

Moreover, any minimizer $X_{\min}$ of (8) has the property that its first k_+ columns consist of eigenvectors belonging to $\lambda_1^+, \ldots, \lambda_{k_+}^+$ and its last k_- columns consist of eigenvectors belonging to $\lambda_1^-, \ldots, \lambda_{k_-}^-$.

It follows from the first part of Theorem 2.3 that the conditions of Theorem 2.4 are also satisfied for

$$U^H AU - \lambda U^H BU,$$

provided that $U \in \mathbb{C}^{n\times p}$ has full column rank. If we moreover assume that

$$(k_+, k_-, 0) \leq \mathsf{In}(U^H BU) \tag{9}$$

then Theorem 2.4 implies

$$\min_{\substack{Y \in \mathbb{C}^{p\times k} \\ (UY)^H B(UY) = J_k}} \operatorname{trace}(UY)^H A(UY) = \sum_{i=1}^{k_+} \theta_i^+ - \sum_{j=1}^{k_-} \theta_j^-, \tag{10}$$

with the eigenvalues θ_i^+, θ_j^- of $U^H AU - \lambda U^H BU$ ordered as in (5). It now follows from the second part of Theorem 2.3 that

$$\min_{\substack{X \in \mathbb{C}^{n\times k} \\ X^H BX = J_k}} \operatorname{trace}(X^H AX) \leq \min_{\substack{Y \in \mathbb{C}^{p\times k} \\ (UY)^H B(UY) = J_k}} \operatorname{trace}(UY)^H A(UY), \tag{11}$$

with equality if and only if U is spanned by eigenvectors belonging $\lambda_1^+, \ldots, \lambda_{k_+}^+$ and $\lambda_1^-, \ldots, \lambda_{k_-}^-$.

Finally, we will provide a basic result which follows directly from Sylvester's law of inertia and will be useful to guarantee a certain property of our algorithm.

Lemma 2.5 *Let $B \in \mathbb{C}^{n\times n}$ be Hermitian, and consider a partitioned matrix $U = [X, Y] \in \mathbb{C}^{n\times p}$. Moreover, let $\mathsf{In}(X^H BX) =: (k_+, k_-, k_0)$ and $\mathsf{In}(U^H BU) =: (p_+, p_-, p_0)$. Then*

$$k_+ \leq p_+, \qquad k_- \leq p_-.$$

3 Algorithms

In the following, we will develop LOBPCG-like methods for approximating the inner eigenvalues $\lambda_1^+, \ldots, \lambda_{\ell_+}^+$ and $\lambda_1^-, \ldots, \lambda_{\ell_-}^-$ of a positive definite pencil $A - \lambda B$. In principle, the integers ℓ_+, ℓ_- can be freely chosen, but our algorithms aim at the case when both ℓ_+ and ℓ_- are small.

In the spirit of the standard LOBPCG method we produce a sequence of matrices

$$X^{(0)}, X^{(1)}, X^{(2)}, \ldots \in \mathbb{C}^{n\times k}, \qquad k = k_+ + k_- \geq \ell_+ + \ell_-,$$

as follows. In the ith iteration of the algorithm, we consider the subspace

$$\mathcal{U} = \operatorname{span}\big[X^{(i)}, W^{(i)}, X^{(i-1)}\big],$$

with the *preconditioned residual*

$$W^{(i)} = T \cdot \Big(AX^{(i)} - BX^{(i)}\Theta^{(i)}\Big)$$

for some Hermitian positive definite matrix $T \in \mathbb{C}^{n\times n}$ and

$$\Theta^{(i)} := \big((X^{(i)})^H BX^{(i)}\big)^{-1}(X^{(i)})^H AX^{(i)}.$$

Then we choose $X^{(i+1)} \in \mathbb{C}^{n\times k}$ according to the trace minimization principle (8), but under the additional constraint $\operatorname{span}(X^{(i+1)}) \subset \mathcal{U}$:

$$\begin{aligned} X^{(i+1)} &:= \operatorname*{arg\,min}_{\substack{\operatorname{span}(X)\subset\mathcal{U}\\ X^H BX = J_k}} \operatorname{trace}(X^H AX) \\ &= \operatorname*{arg\,min}_{\substack{X = UY,\ Y\in\mathbb{C}^{3k\times k}\\ X^H BX = J_k}} \operatorname{trace}(X^H AX), \end{aligned} \tag{12}$$

where $U \in \mathbb{C}^{n\times 3k}$ is any basis of $\mathcal{U}$. Clearly, (13) has a finite value if and only if $(k_+, k_-, 0) \le \operatorname{In}(U^H BU)$. Based on (10), we can then compute $X^{(i+1)} = UY^{(i+1)}$ by letting $Y^{(i+1)} \in \mathbb{C}^{3k\times k}$ contain the eigenvectors belonging to the eigenvalues $\theta_1^+, \dots, \theta_{k_+}^+, \theta_1^-, \dots, \theta_{k_-}^-$ of $U^H AU - \lambda U^H BU$. We assume these eigenvectors to be normalized such that

$$\big(Y^{(i+1)}\big)^H\big(U^H BU\big)Y^{(i+1)} = J_k. \tag{13}$$

3.1 An indefinite LOBPCG method with one preconditioner

The discussion above leads to Algorithm 1, our first indefinite variant of LOBPCG.

Several remarks on the use and implementation of Algorithm 1 are in order:

Input The user is required to supply a valid initial guess $X^{(0)}$ in the sense that

$$\operatorname{In}\big((X^{(0)})^H BX^{(0)}\big) = (k_+, k_-, k_0) \ge (\ell_+, \ell_-, 0)$$

holds. Note that the inequality $k_\pm \le p_\pm$ on $\operatorname{In}\big((U^{(i)})^H BU^{(i)}\big) = (p_+, p_-, p_0)$ then holds for all iterations. This fact follows by induction from Lemma 2.5. In many applications, choosing such a valid initial guess is quite straightforward as B often has a particular structure, for example when B is diagonal.

When there are clusters in the spectrum, it can sometimes be helpful to choose k_+ and/or k_- *strictly* larger than the number of desired eigenvalues, see also the discussion in [5].

Lines 1 and 12 For the standard LOBPCG method it has been observed that choosing an orthonormal basis leads to improved numerical stability [14]. We have made a similar observation when choosing a B-orthonormal basis in Algorithm 1. We therefore apply a Gram-Schmidt procedure (in the B-inner product) to the columns

Algorithm 1 Indefinite LOBPCG method with one preconditioner

Input: $A, B \in \mathbb{C}^{n\times n}$: coefficients of positive definite pencil $A - \lambda B$;
$T \in \mathbb{C}^{n\times n}$: Hermitian positive definite preconditioner;
ℓ_+, ℓ_-: #desired B-positive/B-negative eigenvalues;
$X^{(0)} \in \mathbb{C}^{n\times k}$: initial guess s.t. $(\ell_+, \ell_-, 0) \le \mathsf{In}\big[(X^{(0)})^H B X^{(0)}\big]$.
Output: ℓ_+ smallest B-positive eigenpairs and ℓ_- largest B-negative eigenpairs.

```
$B$-orthonormalize $X^{(0)}$.
$(\Theta^{(0)}, V^{(0)}) \leftarrow \mathrm{RR}(X^{(0)}, A, B)$.
$X^{(0)} \leftarrow X^{(0)} V^{(0)}$, $P^{(0)} \leftarrow []$.
for $i = 0, 1, \ldots$ do
  $R^{(i)} = AX^{(i)} - BX^{(i)}\Theta^{(i)}$.
  (optional) Deflate converged eigenvalues.
  if all desired eigenvalues are converged then
    Exit loop.
  end if
  $W^{(i)} \leftarrow T \cdot R^{(i)}$.
  $U^{(i)} \leftarrow [X^{(i)}, W^{(i)}, P^{(i)}]$.
  $B$-orthonormalize $U^{(i)}$.
  $(\Theta^{(i+1)}, V^{(i+1)}) \leftarrow \mathrm{RR}(U^{(i)}, A, B)$.
  $P^{(i+1)} \leftarrow U_2^{(i)} V_2^{(i+1)}$,   $X^{(i+1)} \leftarrow U_1^{(i)} V_1^{(i+1)} + P^{(i+1)}$.
end for
```

of the current basis $U^{(i)} \equiv U = [u_1, u_2, \ldots, u_p]$. The first step of this procedure consists of normalizing u_1: $u_1 \leftarrow u_1 \big/ |u_1^H B u_1|^{1/2}$. Let us now suppose that the first $\ell - 1$ columns $U_{\ell-1} = [u_1, u_2, \ldots, u_{\ell-1}]$ are already B-orthonormal. Then the ℓth step takes the form

$$u_\ell \leftarrow u_\ell - U_{\ell-1} J U_{\ell-1}^H B u_\ell,$$
$$u_\ell \leftarrow u_\ell \big/ |u_\ell^H B u_\ell|^{1/2},$$

where, by construction, $J := U_{\ell-1}^H B U_{\ell-1} = \mathrm{diag}\{\pm 1\}$ is a diagonal matrix with ± 1 on the diagonal. As discussed in [3, Sec. 8], a careful implementation of this scheme should use reorthogonalization or a modified Gram-Schmidt procedure to avoid instabilities.

There is an additional complication due to the fact that the Gram-Schmidt procedure may encounter B-neutral vectors u_ℓ even when $U^H B U$ is nonsingular. This can be avoided by preprocessing U such that $|u_\ell^H B u_\ell|$ is monotonically decreasing. One way to achieve this is to perform a factorization of the form

$$\Pi^H U^H B U \Pi = L D L^H,$$

where Π is a permutation matrix and D is a diagonal matrix with diagonal entries of decreasing magnitude.

We refer, e.g., to [12] for a description of the algorithm. Zero diagonal entries correspond to B-neutral vectors, which can (and should) be purged from the basis. In exact arithmetic, the columns of $U \leftarrow U\Pi L^{-H}$ are already B-orthogonal and we could proceed by simply scaling the columns of U. However, we have observed that it is numerically safer to nevertheless perform a Gram-Schmidt procedure on the preprocessed U.

Lines 2 and 13 The function $\mathrm{RR}(U, A, B)$ performs standard Ritz pair extraction by computing the eigenvalues and eigenvectors of the compressed pencil

$$\tilde{A} - \lambda\tilde{B} := U^H(A - \lambda B)U.$$

Suppose that these eigenvalues are ordered such that

$$\theta^-_{p_-} \leq \cdots \leq \theta^-_1 < \theta^+_1 \leq \cdots \leq \theta^+_{p_+},$$

Note that the correct signature of the eigenvalues can be determined by, e.g., testing whether the corresponding eigenvector is $\tilde{B}$-positive or $\tilde{B}$-negative.

Then $\mathrm{RR}(U, A, B)$ returns

$$\Theta = \operatorname{diag}\left(\theta^-_{k_-}, \ldots, \theta^-_1, \theta^+_1, \ldots, \theta^+_{k_+}\right)$$
$$V = \left[v^-_{k_-}, \ldots, v^-_1, v^+_1, \ldots, v^+_{k_+}\right],$$

where $v^\pm_i$ is the eigenvector of $\tilde{A} - \lambda\tilde{B}$ belonging to $\theta^\pm_i$. The Ritz pairs are then given $(\theta^\pm_i, Uv^\pm_i)$.

Line 6 Once some Ritz pair has converged within the desired accuracy, it is sensible to deflate it to avoid unnecessary further computations. In our algorithm, we use a conservative "deflate from the middle" strategy: A B-positive Ritz value θ^+_j is deflatable if and only if all smaller Ritz values θ^+_i, with $1 \leq i \leq j-1$, are deflatable and

$$\|Ax^+_j - \theta^+_j Bx^+_j\|_2 \leq \mathsf{tol} \cdot |\theta^+_j| \, \|B\|_2 \|x^+_j\|_2, \tag{14}$$

where $x^+_j = Uv^+_j$ is the corresponding Ritz vector and tol is a tolerance specified by the user. This strategy helps reduce the chance that we deflate a Ritz value approximating an undesired eigenvalue λ^+_ℓ with $\ell > k_+$, even if this seems to be a rare event in practice. An analogous criterion is applied to B-negative Ritz values. The deflation of Ritz pairs proceeds in the same way as in the standard LOBPCG method [23]: Deflated Ritz vectors do not participate in the computation of $R^{(i)}$ or $P^{(i)}$. However, they still need to participate in the B-orthonormalization process to avoid repeated convergence to the same eigenvalue.

Lines 11 and 14 Because $X^{(i-1)}$ and $X^{(i)}$ tend to contain the same information as i increases, the natural basis $\left[X^{(i)}, W^{(i)}, X^{(i-1)}\right]$ for $\mathcal{U}$ is severely ill-conditioned. To avoid this effect, we choose a different basis $\left[X^{(i)}, W^{(i)}, P^{(i)}\right]$, pretty much in the same way as suggested for the standard LOBPCG method [14, 16, 23]. For this purpose, the $3k \times k$ matrix $V^{(i+1)}$ returned by RR is partitioned as

$$V^{(i+1)} = \begin{bmatrix} V^{(i+1)}_1 \\ V^{(i+1)}_2 \end{bmatrix}, \quad V^{(i+1)}_1 \in \mathbb{C}^{k\times k}, \quad V^{(i+1)}_2 \in \mathbb{C}^{2k\times k},$$

and $U^{(i)} = [U_1^{(i)}, U_2^{(i)}]$ is partitioned accordingly. Then the update

$$P^{(i+1)} \leftarrow U_2^{(i)} V_2^{(i+1)}, \qquad X^{(i+1)} \leftarrow U_1^{(i)} V_1^{(i+1)} + P^{(i+1)}.$$

yields a basis that — in exact arithmetic — spans the same space as the natural basis. Note that this makes use of the fact that the first k columns of $U^{(i)}$ (which are already B-orthogonal) are not changed in a careful implementation of the B-orthogonalization process.

3.2 Convergence analysis of Algorithm 1

As for the standard LOBPCG method, a complete convergence analysis of Algorithm 1 is currently out of reach. However, a connection to simpler gradient methods, such as PINVIT [27, 28] (preconditioned inverse iteration) and the preconditioned subspace iteration [18, 29], provides some intuition when we can expect good convergence. For simplicity, we discuss this connection only for the setting when Algorithm 1 is used with $k = 1$ and only λ_1^+ is desired.

Set $\tilde{A} := A - \lambda_0 B$ for a definitizing shift λ_0. Then $B - \mu\tilde{A}$ is a Hermitian pencil with positive definite $\tilde{A}$ and its eigenvalues are related to the eigenvalues of $A - \lambda B$ via $\mu = 1/(\lambda - \lambda_0)$. Suppose that the preconditioner $T \approx \tilde{A}^{-1}$ satisfies

$$\|I - T\tilde{A}\|_{\tilde{A}} \leq \gamma < 1, \tag{15}$$

where $\|\cdot\|_{\tilde{A}}$ denotes the matrix norm induced by $\tilde{A}$. Starting from an initial vector $x^{(0)} \neq 0$, we consider the following gradient iteration [9, 19]:

$$\begin{aligned} \mu^{(i)} &= \frac{(x^{(i)})^H B x^{(i)}}{(x^{(i)})^H \tilde{A} x^{(i)}}, \\ x^{(i+1)} &= x^{(i)} + \frac{1}{\mu^{(i)} - \mu_{\min}} T \cdot \big(Bx^{(i)} - \mu^{(i)}\tilde{A}x^{(i)}\big), \end{aligned} \tag{16}$$

where $\mu_{\min} = 1/(\lambda_1^- - \lambda_0)$.

Provided that $\mu_2 < \mu^{(i)} \leq \mu_1$ and (15) holds, the analysis in [18, 19] provides the sharp convergence estimate

$$\frac{\mu_1 - \mu^{(i+1)}}{\mu^{(i+1)} - \mu_2} \leq \eta^2 \frac{\mu_1 - \mu^{(i)}}{\mu^{(i)} - \mu_2} \quad \text{with} \quad \eta = 1 - (1-\gamma)\frac{\mu_1 - \mu_2}{\mu_1 - \mu_{\min}},$$

where $\mu_j := 1/(\lambda_j^+ - \lambda_0)$ for $j = 1, 2$. Asymptotically, as $\mu^{(i)} \to \mu_1$, we have $(\mu^{(i+1)} - \mu_2)/(\mu^{(i)} - \mu_2) \to 1$. In terms of the eigenvalues of $A - \lambda B$, we therefore obtain for $\lambda^{(i)} := 1/\mu^{(i)} + \lambda_0$ that

$$\frac{\lambda^{(i+1)} - \lambda_1^+}{\lambda^{(i)} - \lambda_1^+} \lesssim \eta^2 \quad \text{with} \quad \eta = 1 - (1-\gamma)\frac{\lambda_2^+ - \lambda_1^+}{\lambda_1^+ - \lambda_1^-} \cdot \frac{\lambda_0 - \lambda_1^-}{\lambda_2^+ - \lambda_0}. \tag{17}$$

When applying Algorithm 1 with $k = 1$, the Ritz vector defining the next iterate is chosen from a larger subspace $\mathcal{U} = \operatorname{span}\{x^{(i)}, w^{(i)}, x^{(i-1)}\}$. Using

$$\theta^{(i)} = \frac{(x^{(i)})^H A x^{(i)}}{(x^{(i)})^H B x^{(i)}} = \frac{1}{\mu^{(i)}} + \lambda_0,$$

the preconditioned residual satisfies

$$w^{(i)} = T \cdot \left(Ax^{(i)} - \theta^{(i)} B x^{(i)}\right) = -\frac{1}{\mu^{(i)}} T \cdot \left(Bx^{(i)} - \mu^{(i)} \tilde{A} x^{(i)}\right).$$

In particular, this shows that the next iterate $x^{(i+1)}$ produced by the gradient method (16) is contained in $\mathcal{U}$. Hence, the eigenvalue interlacing property of Theorem 2.3 implies that the Ritz value θ_1^+ computed in Algorithm 1 is at least as close to λ_1^+ as the eigenvalue approximation $\lambda^{(i+1)} = 1/\mu^{(i+1)} + \lambda_0$ computed by (16). Hence, Algorithm 1 converges at least linearly with the asymptotic convergence rate η^2. This convergence rate crucially depends on a reasonable gap between λ_1^- and λ_0 relative to the gap between λ_0 and λ_2^+, as well as on the quality of the preconditioner measured by (15). This shows that one needs some knowledge on the definiteness interval to construct effective preconditioners, despite the fact that Algorithm 1 itself does not require this information.

For $k > 1$, an analysis of Algorithm 1 can be performed via a similar connection to the preconditioned subspace iteration, using the convergence analysis in [18, 29].

It is important to note that the convergence of Algorithm 1 will usually be much faster than predicted by the analysis above. There is a variety of more refined analyses, yielding improved convergence bounds for the standard LOBPCG method; see [30, 31] and the references therein. It is not unlikely that these improved bounds extend to Algorithm 1, see also [26].

3.3 An indefinite LOBPCG with two preconditioners

The convergence of Algorithm 1 heavily relies on a good choice of the preconditioner T, especially in the presence of eigenvalue clusters. The analysis of Section 3.2 suggests that choosing $T_+ \approx (A - \lambda_0^+ B)^{-1}$ with a definitizing shift λ_0^+ close to λ_1^+ yields reasonable convergence for the smallest B-positive eigenvalues. Similarly, a preconditioner $T_- \approx (A - \lambda_0^- B)^{-1}$ with a definitizing shift λ_0^- close to λ_1^- is well suited for the largest B-negative eigenvalues. However, it will be difficult to find a shift that works equally well for both sets of eigenvalues. A pragmatic remedy is to run Algorithm 1 two times with the two different preconditioners T_+ and T_-; one run focusses on the B-positive eigenvalues and the other run focusses on the B-negative eigenvalues.

However, there is a more elegant solution. We can simply include both preconditioned residuals into the subspace:

$$\operatorname{span}\left[X^{(i)}, T_+ R^{(i)}, T_- R^{(i)}, X^{(i-1)}\right],$$

with $R^{(i)} = AX^{(i)} - BX^{(i)}\Theta^{(i)}$. This increases the dimension of the subspace. To avoid this, we split the residual in two parts: R_+ and R_- associated with B-positive

Ritz values and B-negative Ritz values, respectively. Taking the focus of the preconditioner into account, it makes sense to apply T_+ only to R_+ and T_- only to R_-. This idea yields the subspace

$$\operatorname{span}\big[X^{(i)}, T_+R_+^{(i)}, T_-R_-^{(i)}, X^{(i-1)}\big].$$

The indefinite variant of the LOBPCG method resulting from this choice is given in Algorithm 2. The discussion of Section 3.1 concerning implementation details of Algorithm 1 extends in a direct manner to Algorithm 2.

3.4 Error estimates for computed eigenvalues

At the termination of our algorithms, the obtained Ritz pairs satisfy the bound (14) on the norm of their residuals. By a perturbation analysis, one can then derive upper bounds on the accuracy of the computed eigenvalues and eigenvectors. Such perturbation analyses for definite pencils can be found, e.g., in [34, 35].

To illustrate such an eigenvalue perturbation bound, consider a Ritz pair $(\hat{\lambda}, \hat{x})$ with the corresponding residual vector $r = A\hat{x} - \hat{\lambda}B\hat{x}$. Then one can construct backward errors ΔA, ΔB such that $(\hat{\lambda}, \hat{x})$ is an exact eigenpair of $(A+\Delta A)-\hat{\lambda}(B+\Delta B)$.

Algorithm 2 Indefinite LOBPCG method with two preconditioners

```
Input:  $A, B \in \mathbb{C}^{n\times n}$: coefficients of positive definite pencil $A - \lambda B$;
  $T_+, T_- \in \mathbb{C}^{n\times n}$: Hermitian positive definite preconditioners;
  $\ell_+, \ell_-$: #desired $B$-positive/$B$-negative eigenvalues;
  $X^{(0)} \in \mathbb{C}^{n\times k}$: initial guess s.t. $(\ell_+, \ell_-, 0) \le \mathrm{In}\big[(X^{(0)})^H B X^{(0)}\big]$.
Output:  $\ell_+$ smallest $B$-positive eigenpairs and $\ell_-$ largest $B$-negative eigenpairs.

$B$-orthonormalize $X^{(0)}$.
$(\Theta^{(0)}, V^{(0)}) \leftarrow \mathrm{RR}(X^{(0)}, A, B)$.
$X^{(0)} \leftarrow X^{(0)}V^{(0)}$, $P^{(0)} \leftarrow []$.
for $i = 0, 1, \dots$ do
  $R^{(i)} = AX^{(i)} - BX^{(i)}\Theta^{(i)}$.
  (optional) Deflate converged eigenvalues.
  if all desired eigenvalues are converged then
    Exit loop.
  end if
  $W_+^{(i)} \leftarrow T_+ \cdot R_+^{(i)}$, $W_-^{(i)} \leftarrow T_- \cdot R_-^{(i)}$.
  $U^{(i)} \leftarrow [X^{(i)}, W_+^{(i)}, W_-^{(i)}, P^{(i)}]$.
  $B$-orthonormalize $U^{(i)}$.
  $(\Theta^{(i+1)}, V^{(i+1)}) \leftarrow \mathrm{RR}(U^{(i)}, A, B)$.
  $P^{(i+1)} \leftarrow U_2^{(i)}V_2^{(i+1)}$,    $X^{(i+1)} \leftarrow U_1^{(i)}V_1^{(i+1)} + P^{(i+1)}$.
end for
```

For sufficiently small $\|r\|_2$, this perturbed pair remains positive definite and we have

$$|\hat{\lambda} - \lambda| \leq 2\left|\frac{\lambda - \lambda_0}{\lambda_1^+ - \lambda_1^-}\right| \cdot \frac{\|r\|_2}{\|\hat{x}\|_2}, \tag{18}$$

where $\lambda_0 = (\lambda_1^+ + \lambda_1^-)/2$. This bound can be estimated once sufficiently accurate approximations to $\lambda_1^\pm$ have been obtained.

The bound (18) is derived by applying Weyl's inequality to an equivalent Hermitian eigenvalue problem. Similarly, a Temple-Kato type quadratic residual bound is derived from [32, Thm. 11.7.1]:

$$|\hat{\lambda} - \lambda| \leq \frac{4}{\mathrm{gap}(\hat{\lambda})(\lambda_1^+ - \lambda_1^-)} \cdot \left|\frac{\lambda - \lambda_0}{\hat{\lambda} - \lambda_0}\right| \cdot \frac{\|r\|_2^2}{\|\hat{x}\|_2^2}, \tag{19}$$

where

$$\mathrm{gap}(\hat{\lambda}) = \inf\left\{\left|(\hat{\lambda} - \lambda_0)^{-1} - (\lambda - \lambda_0)^{-1}\right| : \ \lambda \text{ is an eigenvalue of } A - \lambda B\right\}.$$

Using techniques in [25, 33], the residual bounds can be extended to the case when a subspace is computed.

4 Application to the product eigenvalue problem

In this section, we consider the application of our indefinite LOBPCG method to compute the ℓ smallest eigenvalues of matrix product KM for Hermitian positive definite $K, M \in \mathbb{C}^{m\times m}$. As discussed in the introduction, this is equivalent to computing the inner eigenvalues of the matrix pencil

$$A - \lambda B = \begin{bmatrix} K & 0 \\ 0 & M \end{bmatrix} - \lambda \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}. \tag{20}$$

More specifically, λ^2 is an eigenvalue of KM if and only if $\pm\lambda$ are eigenvalues of $A - \lambda B$. Moreover, the corresponding eigenvectors of $A - \lambda B$ are given by

$$\begin{bmatrix} x \\ y \end{bmatrix}, \qquad \begin{bmatrix} x \\ -y \end{bmatrix}, \tag{21}$$

where x and y are eigenvectors of MK and KM, respectively, belonging to λ^2.

Since the pencil $A - \lambda B$ in (20) is positive definite, Algorithm 2 can be applied straightaway to it. However, it turns out that we can reorganize this algorithm to reflect the structure (21) of the eigenvectors and reduce its computational cost. In the following, we illustrate how this is achieved.

Trace minimization Due to the symmetry of the eigenvalues, the trace minimization principle (8) applied to the pencil (20) reads

$$\min_{\substack{Z \in \mathbb{C}^{2m \times k} \\ Z^H BZ = J_k}} \operatorname{trace}(Z^H AZ) = \sum_{i=1}^{k_+} \lambda_i^+ - \sum_{j=1}^{k_+} \lambda_j^- = 2 \sum_{i=1}^{k_+} \lambda_i^+, \tag{22}$$

where $k = 2k_+$ and $\lambda_1^+, \ldots, \lambda_{k_+}^+$ are the smallest positive eigenvalues of $A - \lambda B$. Because of the symmetry (21) of the eigenvectors, the minimizer Z_* can be chosen as

$$Z_* = \begin{bmatrix} X_* & X_* \\ Y_* & -Y_* \end{bmatrix}. \tag{23}$$

Hence, we can additionally require the matrix Z in (22) to have the same block structure. The B-orthogonality condition $Z^H BZ = J_k$ then becomes $X^H Y = \frac{1}{2} I_{k_+}$ for the blocks X, Y of Z. To summarize, we arrive at a trace minimization principle of the form

$$2 \sum_{i=1}^{k_+} \lambda_i^+ = \min_{X^H Y = \frac{1}{2} I_{k_+}} 2 \operatorname{trace}(X^H KX + Y^H MY) = \min_{X^H Y = I_{k_+}} \operatorname{trace}(X^H KX + Y^H MY). \tag{24}$$

This turns out to be identical with the minimization principle in [5], from which LOBP4DCG is derived. A similar observation has been made in [4, Remark A.1]. This already strongly indicates that LOBP4DCG can be viewed as a special case of Algorithm 2 applied to (20). To draw this conclusion, we need to impose additional requirements on the initial guess and the preconditioner in Algorithm 2.

Initial guess Taking into account (23), it is sensible to choose an initial guess having the same block structure. For instance,

$$\begin{bmatrix} X^{(0)} & X^{(0)} \\ Y^{(0)} & -Y^{(0)} \end{bmatrix} = \begin{bmatrix} E_{k_+} & E_{k_+} \\ E_{k_+} & -E_{k_+} \end{bmatrix} \tag{25}$$

is a valid initial guess, where $E_{k_+} = [e_1, \ldots, e_{k_+}]$ contains the first k_+ columns of I_m.

Preconditioners Given two approximate eigenpairs $(\hat{\lambda}, [\hat{x};\ \hat{y}])$ and $(-\hat{\lambda}, [\hat{x};\ -\hat{y}])$, the corresponding residual vectors inherit the structure:

$$\begin{bmatrix} r_x \\ r_y \end{bmatrix} = (A - \hat{\lambda} B) \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} \quad \Leftrightarrow \quad \begin{bmatrix} r_x \\ -r_y \end{bmatrix} = (A + \hat{\lambda} B) \begin{bmatrix} \hat{x} \\ -\hat{y} \end{bmatrix}.$$

Again, it is sensible to require that this structure is preserved for the preconditioned residuals $T_+ \left[\begin{smallmatrix} r_x \\ r_y \end{smallmatrix} \right]$ and $T_- \left[\begin{smallmatrix} r_x \\ -r_y \end{smallmatrix} \right]$. This is the case in general if and only if

$$T_- = \begin{bmatrix} I & \\ & -I \end{bmatrix} T_+ \begin{bmatrix} I & \\ & -I \end{bmatrix}.$$

For example, this property holds if $T_+ = (A - \lambda_0 B)^{-1}$ and $T_- = (A + \lambda_0 B)^{-1}$ for some definitizing shift λ_0.

B-orthonormalization When imposing the requirements on the initial guess and the preconditioned residual, the basis before performing B-orthonormalization in Algorithm 2 takes the form

$$U = \begin{bmatrix} U_X & U_X \\ U_Y & -U_Y \end{bmatrix}, \qquad U_X, U_Y \in \mathbb{C}^{m \times p}. \tag{26}$$

It is important that B-orthonormalization does not destroy this structure. For this purpose, let us consider a perfect shuffle permutation of the columns of U:

$$\left[\begin{array}{cc|cc|c|cc} u_{X,1} & u_{X,1} & u_{X,2} & u_{X,2} & \cdots\ \cdots & u_{X,p} & u_{X,p} \\ u_{Y,1} & -u_{Y,1} & u_{Y,2} & -u_{Y,2} & \cdots\ \cdots & u_{Y,p} & -u_{Y,p} \end{array} \right] \tag{27}$$

If U_X, U_Y are real matrices then

$$\begin{bmatrix} u_{X,j} \\ u_{Y,j} \end{bmatrix}^T B \begin{bmatrix} u_{X,j} \\ -u_{Y,j} \end{bmatrix} = 0.$$

Hence, the individual block columns in (27) are already B-orthogonal. The B-orthogonalization procedure (e.g., Gram-Schmidt) can therefore proceed in blocks of 2 columns, which preserves the structure of (27). By shuffling back, the obtained B-orthonormal basis inherits the structure of U in (26). Even when U_X and U_Y are not real, we can proceed in the same way. Note, however, that the obtained basis will not be B-orthogonal within the block columns of (27).

Summary Based on the observations above, Algorithm 2 applied to (20) with proper initial guess and preconditioners completely preserves the block structure (23) in the iterates. Hence, the algorithm can be formulated in terms of $m \times p$ matrices $X^{(i)}$, $Y^{(i)}$ instead of a $2m \times 2p$ matrix. This reduces both, memory requirements and computational cost. More importantly, this strategy preserves the symmetry of the computed eigenvalues and eigenvectors. As a result of this specialization and careful reorganization of Algorithm 2, we obtain Algorithm 3. Because Algorithm 3 and the LOBP4DCG algorithm proposed in [5] operate with the same projection subspace, they are mathematically equivalent.

Algorithm 3 Indefinite LOBPCG method for product eigenvalue problem

```
Input: $K, M \in \mathbb{C}^{m\times m}$: Hermitian positive definite matrices;
  $T_+ \in \mathbb{C}^{2m\times 2m}$: Hermitian positive definite preconditioner;
  $\ell_+ \in \mathbb{N}$: #desired positive eigenvalues;
  $X^{(0)}, Y^{(0)} \in \mathbb{C}^{m\times k_+}$: initial guess s.t. $k_+ \geq \ell_+$ and $(X^{(0)})^H Y^{(0)}$ is nonsingular.
Output: $\ell$ smallest positive eigenpairs of $KM$.

$B$-orthonormalize $[X^{(0)}, X^{(0)}; Y^{(0)}, -Y^{(0)}]$.
$(\Theta^{(0)}, V^{(0)}) \leftarrow \mathrm{RR}([X^{(0)}, X^{(0)}; Y^{(0)}, -Y^{(0)}], A, B)$.
Update $[X^{(i+1)}; Y^{(i+1)}]$; $P_X^{(0)} = P_Y^{(0)} \leftarrow []$.
for $i = 0, 1, \ldots$ do
  $R_X^{(i)} = KX^{(i)} - Y^{(i)}\Theta^{(i)}$, $R_Y^{(i)} = MY^{(i)} - X^{(i)}\Theta^{(i)}$.
  (optional) Deflate converged eigenvalues.
  if all desired eigenvalues are converged then
    Exit loop.
  end if
  $[W_X^{(i)}; W_Y^{(i)}] \leftarrow T_+ \cdot [R_X^{(i)}; R_Y^{(i)}]$,
  $U_X^{(i)} \leftarrow [X^{(i)}, W_X^{(i)}, P_X^{(i)}]$, $U_Y^{(i)} \leftarrow [Y^{(i)}, W_Y^{(i)}, P_Y^{(i)}]$.
  $B$-orthonormalize $[U_X^{(i)}, U_X^{(i)}; U_Y^{(i)}, -U_Y^{(i)}]$.
  $(\Theta^{(i+1)}, V^{(i+1)}) \leftarrow \mathrm{RR}([U_X^{(i)}, U_X^{(i)}; U_Y^{(i)}, -U_Y^{(i)}], A, B)$.
  $P_X^{(i+1)} \leftarrow U_{X_2}^{(i)}(V_{X_2}^{(i+1)} + V_{Y_2}^{(i+1)})$, $P_Y^{(i+1)} \leftarrow U_{Y_2}^{(i)}(V_{X_2}^{(i+1)} - V_{Y_2}^{(i+1)})$.[a]
  $X^{(i+1)} \leftarrow U_{X_1}^{(i)}(V_{X_1}^{(i+1)} + V_{Y_1}^{(i+1)}) + P_X^{(i+1)}$, $Y^{(i+1)} \leftarrow U_{Y_1}^{(i)}(V_{X_1}^{(i+1)} - V_{Y_1}^{(i+1)}) + P_Y^{(i+1)}$.
end for
$(\Theta, [X; Y]) \leftarrow (\Theta^{(i)}, [X^{(i)}; Y^{(i)}])$.
```

[a] $V^{(i+1)}$ is partitioned into $[V_{X_1}, V_{X_2}; V_{Y_1}, V_{Y_2}]$, where $V_{X_1}, V_{Y_1} \in \mathbb{C}^{m\times k_+}$.

5 Numerical experiments

In the following, we present numerical experiments for three different examples. The main purpose of these examples is to give some insight into the performance and limitations of the proposed algorithms. All experiments have been performed in MATLAB R2012a on an Intel Xeon quadcore 2.66 GHz CPU.

Example 1 We first consider a pencil $A - \lambda B$ of the form (20) corresponding to a product eigenvalue problem KM. In our example, the matrices K, M are real,

5660×5660 and symmetric positive definite. As the examples considered in [5], these matrices arise in the linear response analysis of the density matrix in electronic structure calculations. We aim at computing the four smallest positive eigenvalues, which are given by

$$\lambda_1^+ \approx 0.541812517132466, \qquad \lambda_2^+ \approx 0.541812517132473,$$
$$\lambda_3^+ \approx 0.541812517132498, \qquad \lambda_4^+ \approx 0.615143209274579.$$

Comparisons are made between our algorithms and the LOBP4DCG algorithm proposed in [5].

We make use of the preconditioners

$$T_0 = A^{-1}, \quad T_{\pm\lambda_0} = (A \mp \lambda_0 B)^{-1}, \tag{28}$$

where $\lambda_0 = 0.54$ is a relatively close shift. We use $\mathsf{tol} = 10^{-7}$. The initial guess proposed in (25) with $k = 4$ is used for Algorithms 1 and 2, while the initial guess $\left[\begin{smallmatrix} E_4 \\ E_4 \end{smallmatrix}\right]$ is used for LOBP4DCG and Algorithm 3.

Table 1 contains the number of required iterations until all four desired eigenvalues have converged. As expected from the discussion in Section 4, all algorithms have a similar convergence behavior when using the same preconditioner. Not surprisingly, using the preconditioner $T_{\pm\lambda_0}$ instead of T_0 accelerates convergence. The execution times are mainly provided as a reference; we have not attempted to optimize any of the algorithms. Nevertheless, they nicely reflect that Algorithm 3 and LOBP4DCG gain efficiency from exploiting the structure of A, B.

Figure 1 contains plots of the relative errors for the smallest four positive Ritz values produced in the ith iteration of LOBP4DCG and Algorithm 3. Note that the convergence observed for the first three Ritz values is much faster than for the fourth. Moreover, we clearly see the effect of the quadratic residual bound (19); the errors in the deflated Ritz values ($\sim 10^{-13}$) are much smaller than $\mathsf{tol} = 10^{-7}$.

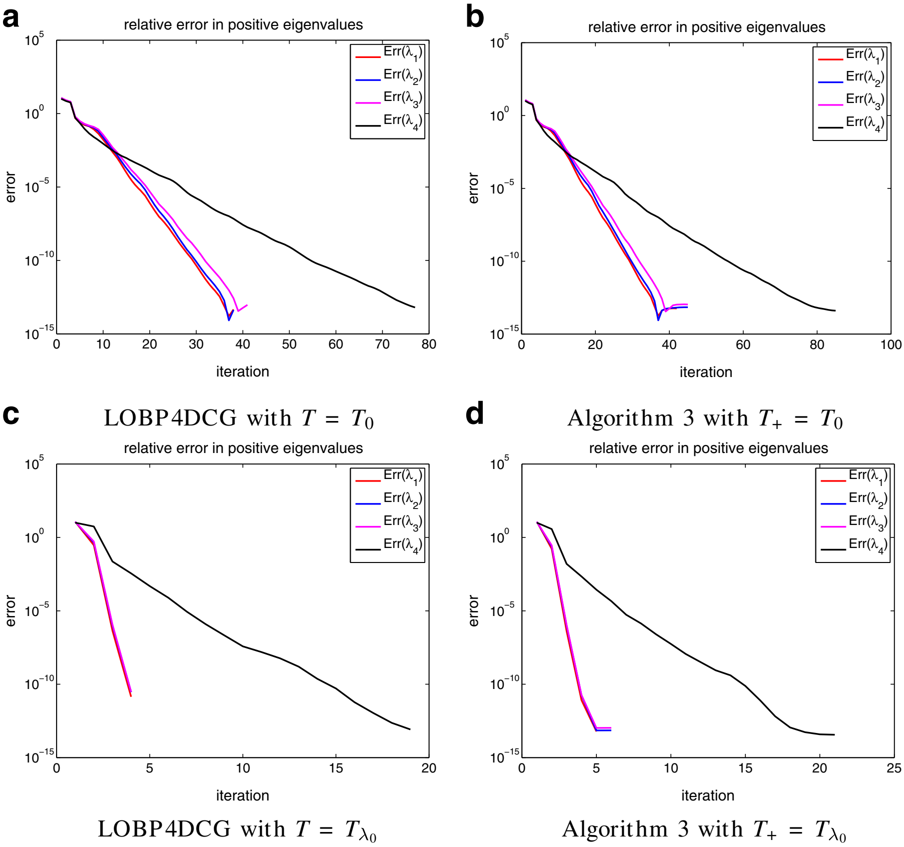
Next, we investigate the effect of replacing the exact inverse preconditioners (28) with a few iterations of the CG method applied to A and $A \mp \lambda_0 B$, respectively. The stopping tolerance for the inner CG iterations is set to 10^{-2} and we allow for 50 iterations at most. Table 2 shows the obtained results, including the total number of inner CG iterations required until convergence of the Ritz values.

Example 2 Now we consider a quadratic eigenvalue problem (QEP)

$$Q(\lambda)x = (\lambda^2 M + \lambda D + K)x = 0, \tag{29}$$

Table 1 Product eigenvalue problem with exact inverse preconditioners

Solver	#(iter)	CPU time (sec.)
LOBP4DCG with $T = T_0$	77	45.2
Algorithm 3 with $T_+ = T_0$	85	67.1
Algorithm 1 with $T = T_0$	77	88.1
LOBP4DCG with $T = T_{\lambda_0}$	19	27.7
Algorithm 3 with $T_+ = T_{\lambda_0}$	21	29.7
Algorithm 2 with $T_\pm = T_{\pm\lambda_0}$	20	53.8



LOBP4DCG with $T = T_0$

Algorithm 3 with $T_+ = T_0$

LOBP4DCG with $T = T_{\lambda_0}$

Algorithm 3 with $T_+ = T_{\lambda_0}$

Fig. 1 Relative errors of computed eigenvalues in LOBP4DCG and Algorithm 3

where M, D, $K \in \mathbb{C}^{n\times n}$ are Hermitian and M is positive definite. Such eigenvalue problems have motivated much of the research on definite pencils, see, e.g., [37]. A linearization of (29) yields the Hermitian pencil $A - \lambda B$ with

$$A = \begin{bmatrix} M & \\ & -K \end{bmatrix}, \qquad B = \begin{bmatrix} & M \\ M & D \end{bmatrix}. \tag{30}$$

The positive definiteness of this pencil is equivalent to requiring the original QEP (29) to be hyperbolic [15, 36]. When D is positive definite, the B-positive and B-negative eigenvectors are contained in the subspaces spanned by columns of $\left[\begin{smallmatrix} 0 \\ I \end{smallmatrix}\right]$ and $\left[\begin{smallmatrix} M^{-1}D \\ -I \end{smallmatrix}\right]$, respectively. This simplifies the task of choosing appropriate initial guesses in our algorithms.

Table 2 Product eigenvalue problem with CG-based preconditioners

Solver	#(iter)	#(inner iter)	CPU time (sec.)
LOBP4DCG with $T = T_0$	76	2716	284.9
Algorithm 3 with $T_\pm = T_0$	94	1965	243.6
Algorithm 1 with $T = T_0$	78	2722	571.0
LOBP4DCG with $T = T_{\lambda_0}$	20	2806	262.6
Algorithm 3 with $T_+ = T_{\lambda_0}$	27	1141	120.8
Algorithm 2 with $T_\pm = T_{\pm\lambda_0}$	20	1914	363.7

As a simple scalable example, we consider

$$K = (n+1)^2 \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix}, \quad M = I_n, \quad D = 2K.$$

In this case, the QEP can be easily verified to be hyperbolic; its eigenvalues are given by

$$\lambda_j^\pm = -\alpha_j \pm \sqrt{\alpha_j^2 - \alpha_j}, \quad \text{where} \quad \alpha_j = 4(n+1)^2 \sin^2 \frac{j\pi}{2(n+1)}, \tag{31}$$

for $j = 1, \dots, n$. As n increases, the definiteness interval of $A - \lambda B$ converges to around $(-19.2258, -0.5134)$.

It is well known that numerical algorithms applied to linearizations of QEPs are sensitive to the scaling of the coefficients [10], and our algorithms are no exception. Considering that $\|D\|_2 = 2\|K\|_2 = O(n^2)$, while $\|M\|_2 = 1$, we propose to rescale the pencil $A - \lambda B$ as follows:

$$A - \lambda B \leftarrow \begin{bmatrix} I & \\ & \frac{1}{n+1} I \end{bmatrix} (A - \lambda B) \begin{bmatrix} I & \\ & \frac{1}{n+1} I \end{bmatrix}.$$

We make use of the preconditioners

$$T_0 = (A - \lambda_0 B)^{-1}, \qquad T_1^\pm = (A - \lambda_0^\pm B)^{-1}, \tag{32}$$

where $\lambda_0 = -9$ is a reasonable choice nearly in the middle of the definiteness interval, while $\lambda_0^+ = -0.514$ and $\lambda_0^- = -19.22$ are very close to its boundaries. We use again $\mathsf{tol} = 10^{-7}$ and aim at computing the eigenvalues $\lambda_j^\pm$ for $j = 1, \dots, 3$, that is, $\ell_+ = \ell_- = 3$ in Algorithms 1 and 2. The obtained results are reported in Table 3. Note that we now list the number of total iterations required for B-positive and B-negative eigenvalues separately. For example, in Algorithm 1 with $k_+ = k_- = 3$ all desired B-negative eigenvalues have converged already after 36 iterations, while a total number of 198 iterations is required until also the desired B-positive eigenvalues have converged. This large difference is caused by the fact that the B-positive eigenvalues are more clustered. Only when using Algorithm 2 with the tailored preconditioners $T_1^\pm$, this effect disappears.

Table 3 QEP from Example 5.2 with exact inverse preconditioners

solver	$k_+ = k_-$	$n = 1000$		$n = 2000$	
		#(iter) B-pos.	#(iter) B-neg.	#(iter) B-pos.	#(iter) B-neg.
Algorithm 1					
with $T = T_0$	3	198	36	121	25
	4	137	28	184	27
	5	173	22	178	24
Algorithm 2					
with $T_\pm = T_1^\pm$	3	14	21	11	16
	4	11	15	14	19
	5	11	17	11	15
Algorithm 2					
with $T_\pm = T_2^\pm$	3	12	20	10	17
	4	14	20	10	18
	5	10	16	8	15

It is comforting to see that the number of iterations does not increase significantly when n is doubled. This observation remains true when replacing the exact inverses in (32) by algebraic multigrid (AMG) v-cycle preconditioners, see Table 4. For this purpose, we have used the implementation `HSL_MI20` [1] with the default settings.

It is interesting to note what happens when choosing $T_2^\pm = (A - \lambda_i^\pm B)^{-1}$ with shifts $\lambda_i^+ = -0.51$ and $\lambda_i^- = -20$, which are both *outside* the definiteness interval. Hence, $T_2^\pm$ are indefinite and the assumptions in the convergence analysis of Section 3.2 are not satisfied. Nevertheless, Table 3 clearly demonstrates that the convergence behavior of Algorithm 2 changes little when replacing the definite preconditioners $T_1^\pm$ by $T_2^\pm$. Note that the use of indefinite preconditioners was already discussed in [16] for the standard LOBPCG method.

Table 4 QEP from Example 5.2 with AMG preconditioners

solver	$k_+ = k_-$	$n = 1000$		$n = 2000$	
		#(iter) B-pos.	#(iter) B-neg.	#(iter) B-pos.	#(iter) B-neg.
Algorithm 1					
with $T = T_0$	3	215	45	198	45
	4	185	40	214	39
	5	182	37	188	36
Algorithm 2					
with $T_\pm = T_1^\pm$	3	19	26	20	18
	4	18	23	20	25
	5	19	24	16	23

Finally, we remark that choosing $k_\pm > \ell_\pm$ *can* have a positive effect on the convergence, but the findings of Tables 3 and 4 are not conclusive.

Example 3 Based on the convergence analysis in Section 3.2, we expect our algorithms to perform poorly if the gaps between the desired eigenvalues and the rest of the spectrum are tiny. The example `spring` from the collection NLEVP [8] is a particularly bad instance of such a problem. Specifically, consider the matrices produced by the command `nlevp('spring',n,1,10,5,10,5)`:

$$K = \begin{bmatrix} 15 & -5 & & \\ -5 & \ddots & \ddots & \\ & \ddots & \ddots & -5 \\ & & -5 & 15 \end{bmatrix}, \quad M = I_n, \quad D = 2K.$$

Then the corresponding QEP is hyperbolic and its eigenvalues are given by

$$\lambda_j^\pm = -\alpha_j \pm \sqrt{\alpha_j^2 - \alpha_j}, \quad \text{where} \quad \alpha_j = 5\Big(3 - 2\cos\frac{j\pi}{n+1}\Big), \tag{33}$$

for $j = 1, \ldots, n$. As n increases, the definiteness interval of $A - \lambda B$ converges to around $(-9.472, -0.527)$. Moreover, the gaps between eigenvalues become arbitrarily small as $n \to \infty$, see also Fig. 2.

We make use of the preconditioners

$$T_0 = (A - \lambda_0 B)^{-1}, \qquad T_1^\pm = (A - \lambda_0^\pm B)^{-1}, \tag{34}$$

where $\lambda_0 = -5$ is nearly in the middle of the definiteness interval, while $\lambda_0^+ = -0.528$ and $\lambda_0^- = -9.47$ are very close to its boundaries. We use again $\mathsf{tol} = 10^{-7}$ and aim at computing the eigenvalues $\lambda_j^\pm$ for $j = 1, \ldots, 3$, that is, $\ell_+ = \ell_- = 3$ in Algorithms 1 and 2. The obtained results are reported in Table 5. It can be observed

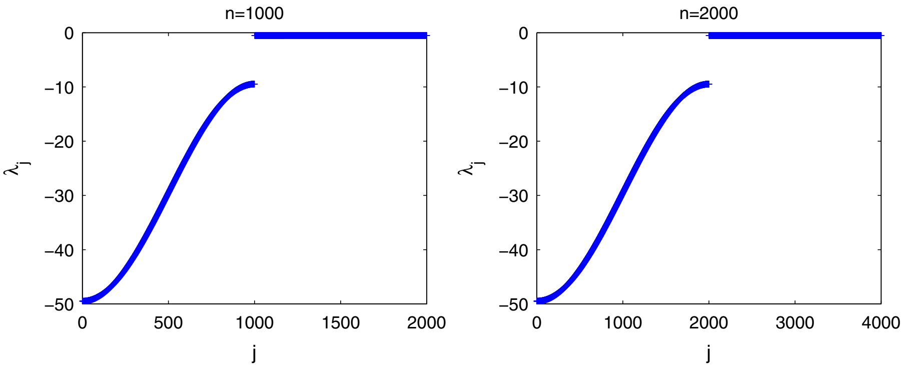


Fig. 2 Eigenvalues of Example 5.3 for $n = 1000$ and $n = 2000$

Table 5 QEP from Example 5.3 with exact inverse preconditioners

		$n = 1000$		$n = 2000$	
solver	$k_+ = k_-$	#(iter) B-pos.	#(iter) B-neg.	#(iter) B-pos.	#(iter) B-neg.
Algorithm 1					
with $T = T_0$	3	> 1500	544	> 1500	974
	4	> 1500	526	> 1500	960
	5	> 1500	513	> 1500	1167
Algorithm 2					
with $T_\pm = T_1^\pm$	3	37	10	74	17
	4	27	9	65	15
	5	24	9	53	15

that the preconditioner T_0 is not effective at all. Even worse, the effectiveness of the excellent preconditioners $T_1^\pm$ deteriorates as n increases, due to the decrease of the eigenvalue gaps.

6 Conclusions

We have shown that the LOBPCG method extends in a natural way to positive definite pencils $A - \lambda B$. Apart from the work by Bai and Li [5], which turns out to be a special case of our method, we are not aware of any other extension of LOBPCG beyond positive definite B.

Finally, it is natural to ask whether the methods developed in this paper can be extended to semi-definite pencils $A - \lambda B$, in view of the trace minimization principle established in [20, 24]. More specifically, consider the case that $A - \lambda_0 B$ is semi-definite for some λ_0 but there is no definitizing shift. In this situation, it appears to be difficult to construct effective preconditioners. Quite likely, information on the nullspace of $A - \lambda_0 B$ needs to be incorporated, similar to the techniques discussed in [2].

Acknowledgments The authors thank Professor Ren-Cang Li for providing the matrices used in Example 5.1 and Cedric Effenberger for helpful discussions. They are grateful to the referees, in particular for pointing out a mistake in the convergence discussion of Section 3.2. Part of this work was performed while the second author was staying at EPFL. She also thanks Professors Krešimir Veselić and Ninoslav Truhar for their guidance.

References

1. HSL.: A collection of Fortran codes for large scale scientific computation. Available from http://www.hsl.rl.ac.uk. (2011)
2. Arbenz, P., Drmač, Z.: On positive semidefinite matrices with known null space. SIAM J. Matrix Anal. Appl. **24**(1), 132–149 (2002)

3. Bai, Z., Demmel, J.W., Dongarra, J.J., Ruhe, A., van der Vorst, H. (eds.): Templates for the solution of algebraic eigenvalue problems. Software, Environments, and Tools. SIAM, Philadelphia (2000)
4. Bai, Z., Li, R.-C.: Minimization principles for the linear response eigenvalue problem I: theory. SIAM J. Matrix Anal. Appl. **33**(4), 1075–1100 (2012)
5. Bai, Z., Li, R.-C.: Minimization principles for the linear response eigenvalue problem II: computation. SIAM J. Matrix Anal. Appl. **34**(2), 392–416 (2013)
6. Bai, Z., Li, R.-C.: Minimization principles for the linear response eigenvalue problem III: general case. Mathematics preprint series. The University of Texas, Arlington (2013)
7. Benner, P., Kressner, D., Mehrmann, V.: Skew-Hamiltonian and Hamiltonian eigenvalue problems: theory, algorithms and applications. In: Drmač, Z., Marušić, M., Tutek, Z. (eds.) Proceedings of the Conference on Applied Mathematics and Scientific Computing, Brijuni (Croatia), June 23-27, 2003, pp. 3–39. Springer-Verlag (2005)
8. Betcke, T., Higham, N.J., Mehrmann, V., Schröder, C., Tisseur, F.: NLEVP: a collection of nonlinear eigenvalue problems. ACM Trans. Math. Software **39**(2), 7:1–7:28 (2013). Also available from http://www.mims.manchester.ac.uk/research/numerical-analysis/nlevp.html
9. D′yakonov, E.G.: Optimization in Solving Elliptic Problems. CRC Press, Boca Raton (1996)
10. Fan, H.-Y., Lin, W.-W., Van Dooren, P.: Normwise scaling of second order polynomial matrices. SIAM J. Matrix Anal. Appl. **26**(1), 252–256 (2004)
11. Gohberg, I., Lancaster, P., Rodman, L.: Matrices and indefinite scalar products. Operator Theory: Advances and Applications, vol. 8. Birkhäuser Verlag, Basel (1983)
12. Hansen, P.C., Yalamov, P.Y.: Symmetric rank revealing factorizations. In: Recent Advances in Numerical Methods and Applications, II (Sofia, 1998), pp. 687–695. World Sci. Publ., River Edge (1999)
13. Hari, V., Singer, S., Singer, S.: Block-oriented J-Jacobi methods for Hermitian matrices. Linear Algebra Appl. **433**(8–10), 1491–1512 (2010)
14. Hetmaniuk, U., Lehoucq, R.: Basis selection in LOBPCG. J. Comput. Phys. **218**(1), 324–332 (2006)
15. Higham, N.J., Tisseur, F., Van Dooren, P.: Detecting a definite Hermitian pair and a hyperbolic or elliptic quadratic eigenvalue problem, and associated nearness problems. Linear Algebra Appl. **351/352**, 455–474 (2002)
16. Knyazev, A.V.: Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. SIAM J. Sci. Comput. **23**(2), 517–541 (2001)
17. Knyazev, A.V., Neymeyr, K.: Efficient solution of symmetric eigenvalue problems using multigrid preconditioners in the locally optimal block conjugate gradient method. Electron. Trans. Numer. Anal. **15**, 38–55 (2003)
18. Knyazev, A.V., Neymeyr, K.: A geometric theory for preconditioned inverse iteration. III: a short and sharp convergence estimate for generalized eigenvalue problems. Linear Algebra Appl. **358**, 95–114 (2003)
19. Knyazev, A.V., Neymeyr, K.: Gradient flow approach to geometric convergence analysis of preconditioned eigensolvers. SIAM J. Matrix Anal. Appl. **31**(2), 621–628 (2009)
20. Kovač-Striko, J., Veselić, K.: Trace minimization and definiteness of symmetric pencils. Linear Algebra Appl. **216**, 139–158 (1995)
21. Lancaster, P., Rodman, L.: Canonical forms for Hermitian matrix pairs under strict equivalence and congurence. SIAM Rev. **47**(3), 407–443 (2005)
22. Lancaster, P., Ye, Q.: Variational and numerical methods for symmetric matrix pencils. Bull. Austral. Math. Soc. **43**(1), 1–17 (1991)
23. Lashuk, I., Argentati, M., Ovtchinnikov, E., Knyazev, A.: Preconditioned eigensolver LOBPCG in hypre and PETSc. In: Widlund, O., Keyes, D. (eds.) Domain Decomposition Methods in Science and Engineering XVI, vol. 55, pp. 635–642. Lecture Notes Computation Science Engineering, (2007)
24. Liang, X., Li, R.-C., Bai, Z.: Trace minimization principles for positive semi-definite pencils. Linear Algebra Appl. **438**(7), 3085–3106 (2013)
25. Mathias, R.: Quadratic residual bounds for the Hermitian eigenvalue problem. SIAM J. Matrix Anal. Appl. **19**(2), 541–550 (1998)
26. Miloloža Pandur, M.: Some iterative methods for solving the symmetric generalized eigenvelue problem. PhD thesis, Department of Mathematics, University of Zagreb, in preparation
27. Neymeyr, K.: A geometric theory for preconditioned inverse iteration. I: extrema of the Rayleigh quotient. Linear Algebra Appl. **322**(1–3), 61–85 (2001)

28. Neymeyr, K.: A geometric theory for preconditioned inverse iteration. II: convergence estimates. Linear Algebra Appl. **322**(1–3), 87–104 (2001)
29. Neymeyr, K.: A geometric theory for preconditioned inverse iteration applied to a subspace. Math. Comp. **71**(237), 197–216 (2002)
30. Neymeyr, K.: A geometric convergence theory for the preconditioned steepest descent iteration. SIAM Numer. Anal. **50**(6), 3188–3207 (2012)
31. Neymeyr, K., Ovtchinnikov, E., Zhou, M.: Convergence analysis of gradient iterations for the symmetric eigenvalue problem. SIAM J. Matrix Anal. Appl. **32**(2), 443–456 (2011)
32. Parlett, B.N.: The Symmetric Eigenvalue Problem, Classics in Applied Mathematics, vol. 20. Corrected reprint of the 1980 original. SIAM, Philadelphia (1998)
33. Stewart, G.W.: Basic decompositions. Matrix Algorithms, vol. I. SIAM, Philadelphia (1998)
34. Stewart, G.W., Sun, J.-G.: Matrix Perturbation Theory. Academic, New York (1990)
35. Truhar, N.: Relative Perturbation Theory for Matrix Spectral Decompositions. PhD thesis, Department of Mathematics, University of Zagreb (2000)
36. Veselić, K.: A Jacobi eigenreduction algorithm for definite matrix pairs. Numer. Math. **64**(2), 241–269 (1993)
37. Veselić, K.: A mathematical introduction. Damped Oscillations of Linear Systems, vol. 2023. Lecture Notes in Mathematics. Springer, Heidelberg (2011)