PRECONDITIONED GRADIENT ITERATIONS FOR THE EIGENPROBLEM OF DEFINITE MATRIX PAIRS

MARIJA MILOLOŽA PANDUR†

Abstract. Preconditioned gradient iterations for a large and sparse Hermitian generalized eigenvalue problem $Ax = \lambda Bx$ with positive definite B, are efficient methods for computing a few extremal eigenpairs. In this paper we give a unifying framework of preconditioned gradient iterations for a definite generalized eigenvalue problem with indefinite B. More precisely, these iterations compute a few eigenvalues closest to the definiteness interval (which can be in the middle of the spectrum) and the corresponding eigenvectors of definite matrix pairs (A, B), that is, pairs having a positive definite linear combination. Sharp convergence theorems for the simplest variants are given. This framework includes an indefinite locally optimal block preconditioned conjugate gradient (LOBPCG) algorithm derived by Kressner, Miloloža Pandur, and Shao [Numer. Algorithms, 66 (2014), pp. 681–703]. We also give a generic algorithm for constructing new "indefinite extensions" of standard (with positive definite B) eigensolvers. Numerical experiments demonstrate the use of our algorithm for solving a product and a hyperbolic quadratic eigenvalue problem. With excellent preconditioners, the indefinite variant of LOBPCG is the most efficient method. Finally, we derive some ideas how to use our indefinite eigensolver to compute a few eigenvalues around *any* spectral gap and the corresponding eigenvectors of definite matrix pairs.

Key words. eigenpair, definite matrix pair, definitizing shift, definiteness interval, spectral gap, preconditioned steepest descent/ascent iteration, indefinite LOBPCG

AMS subject classifications. 65F15, 65F08, 65F50

1. Introduction.

The generalized eigenvalue problem (GEP) for given matrices $A, B \in \mathbb{C}^{n\times n}$ is to find scalars λ and nonzero vectors $x \in \mathbb{C}^n$ such that

$$Ax = \lambda Bx. \tag{1.1}$$

The pair (λ, x) is called an eigenpair, where λ is an eigenvalue and x is a corresponding eigenvector. The GEP (1.1), where A and B are both Hermitian or real symmetric, occurs in many applications of mathematics. A very important case is when B (and A) is positive definite which appears, for example, in the finite element discretization of self-adjoint and elliptic PDE-eigenvalue problems [20]. Another very important case is when B (and A) is indefinite, but the matrix pair (A, B) is *definite*, meaning that there exist real scalars α, β such that the matrix $\alpha A + \beta B$ is positive definite which appears, for example, in mechanics [57] and computational quantum chemistry [5]. Many theoretical properties (such as variational principles, perturbation theory, etc.) and eigenvalue solvers for a Hermitian matrix have been extended to definite matrix pairs [48, 54, 57].

Suppose A and B are both Hermitian and the pair (A, B) is definite; this excludes singular pairs, meaning that $\alpha A + \beta B$ is a singular matrix for each choice of the scalars α, β. In this paper, we are interested in solving the partial definite GEP (1.1), where B (and A) is indefinite. In general, when (A, B) is a definite pair, both matrices A and B can be singular, but since a definite pair is a regular pair (i.e., it is not a singular pair), the intersection of the nullspaces of such A and B must be trivial [54, Example VI.1.3]. Some of the existing eigenvalue solvers, which operate with the indefinite inner product induced by B are indefinite Jacobi algorithms [25, 56], the Rayleigh quotient method [40], and indefinite Lanczos methods [4, 40]. Specifically, in this paper, we are interested in an iterative algorithm that computes a small number of eigenvalues closest to the definiteness interval (see Definition 3.4) and the corresponding eigenvectors. These eigenvalues are themselves

†J.J. Strossmayer University of Osijek, Department of Mathematics, Trg Ljudevita Gaja 6, 31000 Osijek, Croatia. (mmiloloz@mathos.hr).

relevant in some applications such as computational quantum chemistry [5, 52]. Moreover, as we shall see, *every* spectral gap can be viewed as the definiteness interval of a related matrix pair. In this way, we devise an algorithm to compute the eigenvalues in an arbitrary interval of the spectrum. The class of algorithms we deal with are preconditioned gradient-type iterations, in a single vector and a block form, suitable for large and sparse matrices, previously studied for the case in which A and/or B are known to be positive definite; for a survey of preconditioned iterations we refer to [4, 32]. An indefinite variant of the locally optimal block preconditioned conjugate gradient method (the LOBPCG method proposed by Knyazev in [33]) is suggested by Kressner, Miloloža Pandur, and Shao in [38]. In this paper, we propose some new preconditioned eigensolvers suitable for a definite matrix pair with indefinite matrices that can be interpreted as a truncated and extended version of the indefinite LOBPCG [38]. For the truncated version of the indefinite locally optimal preconditioned conjugate gradient (LOPCG) method (this method is LOBPCG with block size one, that is, a single vector version), which we call the *indefinite preconditioned steepest descent/ascent* (PSD/A), we derive sharp convergence estimates.

This paper is organized as follows. In Section 2, we give a short review of two important preconditioned gradient iterations for a GEP $\widetilde{A}x = \widetilde{\lambda}Bx$ with $\widetilde{A}$ positive definite. In Section 3, we give a unifying framework of preconditioned gradient iterations for a definite GEP (1.1) with indefinite B called *an indefinite variant of the* (m)*-scheme*. Section 4 contains sharp convergence estimates for the simplest variants from the indefinite variant of the (m)-scheme. In Section 5, we devise some possibilities of using our algorithm to compute a modest number of eigenvalues around *any spectral gap* of a definite matrix pair. Numerical examples are given in Section 6. Section 7 contains some concluding remarks.

Notation. I_n (or simply I when the dimension is clear from the context) denotes the $n \times n$ identity matrix. $A \succ 0$ ($A \succeq 0$) means that A is a Hermitian positive (semi)definite matrix. $A \prec 0$ ($A \preceq 0$) if $-A \succ 0$ ($-A \succeq 0$). For a given $A \succ 0$, $\|\cdot\|_A$ denotes a matrix norm induced by the vector norm $\|x\|_A = \sqrt{x^H A x}$. $\mathrm{In}(B)$ denotes the inertia of a Hermitian matrix B, that is, it is defined as the triple containing the number of positive, negative, and zero eigenvalues of B. A sub- and superscript $+$, $-$ of some scalar, vector or an iteration name refer to the corresponding B-positive and B-negative property, respectively.

2. Preliminaries. Let $\widetilde{A}, B \in \mathbb{C}^{n\times n}$ be Hermitian and let $\widetilde{A}$ be positive definite. We briefly review two known preconditioned gradient iterations for finding the smallest eigenvalue and the corresponding eigenvector of the GEP

$$\widetilde{A}x = \widetilde{\lambda}Bx. \tag{2.1}$$

Let

$$\widetilde{\rho}(x) = \frac{x^H \widetilde{A}x}{x^H Bx}, \quad x^H Bx \neq 0,$$

denote the Rayleigh quotient associated with the matrix pair $(\widetilde{A}, B)$. When B is indefinite, it is convenient [33, 45] to consider the dual GEP $Bx = \widetilde{\mu}\widetilde{A}x$ with $\widetilde{\mu} = 1/\widetilde{\rho}$ and to find the largest eigenvalue and the corresponding eigenvector. Let $x^{(i)}$ denote the current approximation of the eigenvector corresponding to the smallest eigenvalue of (2.1). For a given Hermitian matrix $T \in \mathbb{C}^{n\times n}$, called the *preconditioner* (usually, $T = \widetilde{A}^{-1}$ or $T \approx \widetilde{A}^{-1}$), the preconditioned steepest descent (PSD) [21, 29, 30] iteration takes the following form

$$x^{(i+1)} = x^{(i)} - \tau^{(i)} T(\widetilde{A}x^{(i)} - \widetilde{\rho}(x^{(i)})Bx^{(i)}), \tag{2.2}$$

where the scalar iteration parameter $\tau^{(i)}$ is chosen such that $\widetilde{\rho}(x^{(i+1)})$ is minimized. The LOPCG iteration [31] takes the following form

$$x^{(i+1)} = x^{(i)} - \gamma^{(i)} x^{(i-1)} - \tau^{(i)} T(\widetilde{A}x^{(i)} - \widetilde{\rho}(x^{(i)})Bx^{(i)}), \tag{2.3}$$

where the scalar iteration parameters $\gamma^{(i)}$ and $\tau^{(i)}$ are chosen such that $\widetilde{\rho}(x^{(i+1)})$ is minimized. The preconditioned residual $w^{(i)} := Tr^{(i)}$, where $r^{(i)} := \widetilde{A}x^{(i)} - \widetilde{\rho}(x^{(i)})Bx^{(i)}$ is obtained by solving the linear system $T^{-1}w^{(i)} = r^{(i)}$. In the finite element discretization of a self-adjoint and elliptic partial differential operator, a symmetric positive definite geometric or algebraic multigrid V-cycle preconditioner is commonly used. According to Neymeyr, "typically, symmetric positive definite multigrid preconditioners can be realized with only linearly increasing computational costs (optimal complexity) and, at best, convergence rates can be guaranteed which do not depend on the mesh size" [44, p. 4]. Also; see [35].

PSD and LOPCG iterations are preconditioned gradient iterations since for the given preconditioner T they use the T -gradient of $\widetilde{\rho}$. The T-gradient reads

$$\nabla_T \widetilde{\rho}(x) := T\nabla\widetilde{\rho}(x), \quad \text{where} \quad \nabla\widetilde{\rho}(x) = \frac{2(\widetilde{A}x - \widetilde{\rho}(x)Bx)}{x^H Bx}.$$

The current residual $\widetilde{A}x - \widetilde{\rho}(x)Bx$ is collinear with the gradient $\nabla\widetilde{\rho}(x)$, which explains the term "gradient" in a gradient iteration. Therefore, the PSD iteration computes a sequence of iterates with decreasing Rayleigh quotients by successive corrections in the negative T-gradient direction of the current iterate. LOPCG additionally contains the optimal direction of the previous iterate.

The connection between a three-term recurrence of the LOPCG iteration (2.3) and a three-term recurrence of the standard linear preconditioned conjugate gradient method (PCG) [49], is pointed out in [33, p. 523]. Numerical experiments in [33, Section 7] demonstrate a similar behaviour of the two mentioned methods when the preconditioners and initial approximations are the same in both methods.

REMARK 2.1. The best possible scalar iteration parameters $\tau^{(i)}$ and $\gamma^{(i)}$ in PSD iteration (2.2) and LOPCG iteration (2.3) are not found by using some optimization method. They are given only implicitly. If x is an eigenvector, then so is αx, $\alpha \neq 0$; so what we really want to find is a direction of the next iterate. From (2.2) and (2.3) we see that the next iterate $x^{(i+1)}$ is in the subspace $\mathrm{span}[x^{(i)}, w^{(i)}]$ and $\mathrm{span}[x^{(i)}, x^{(i-1)}, w^{(i)}]$, respectively. Therefore, the best approximations for eigenpairs from the subspace are given through the Rayleigh–Ritz procedure [48]. Therefore, $(x^{(i+1)}, \widetilde{\rho}(x^{(i+1)}))$ is a *Ritz pair* of the matrix pair $(\widetilde{A}, B)$ with respect to the given subspace. As PSD (LOPCG) aims at minimizing of the Rayleigh quotient, $\widetilde{\rho}(x^{(i+1)})$ is a smaller (the smallest) Ritz value and $x^{(i+1)}$ is the associated Ritz vector.

If we want to find several smallest eigenpairs of GEP (2.1), then we can use the block (or subspace) versions of iterations (2.2) and (2.3): block preconditioned steepest descent (BPSD) and LOBPCG [32, 33], respectively. Standard preconditioned gradient iterations, such as BPSD and LOBPCG, operate with an inner product induced by a positive definite matrix and aim to compute the smallest or the largest eigenvalues (the extremal eigenvalues) and the corresponding eigenvectors. These iterations can be modified in a natural way to compute the eigenvalues around the definiteness interval (which can be in the middle of the spectrum) and the corresponding eigenvectors of a definite matrix pair with indefinite matrices. Therefore, they operate with an indefinite inner product. Indefinite variants of LOBPCG are suggested in [38]. Here we propose the whole class of indefinite variants of preconditioned gradient iterations in Section 3, which includes indefinite LOBPCG methods [38, Algorithms 1 and 2].

3. An indefinite variant of the (m)-scheme of preconditioned gradient iterations. In this section, we propose a new class of preconditioned gradient iterations for definite matrix pairs (A, B) with indefinite B. Therefore, we use the indefinite inner product induced by B. A vector $x \in \mathbb{C}^n$ is called *B-positive*, *B-negative*, and *B-neutral* if $x^H Bx > 0$, $x^H Bx < 0$, and $x^H Bx = 0$, respectively. A vector x is *B-normalized* if $|x^H Bx| = 1$. We call vectors x_i and x_j *B-orthogonal* if $x_i^H Bx_j = 0$ and *B-orthonormal* if $|x_i^H Bx_j| = \delta_{ij}$, where δ_{ij} is the Kronecker delta symbol.

DEFINITION 3.1. *Let $A, B \in \mathbb{C}^{n\times n}$. A matrix pair (A, B) is called a* Hermitian matrix pair *if both A and B are Hermitian. A Hermitian matrix pair (A, B) is called* positive (negative) definite *if there exists a real λ_0 such that $A - \lambda_0 B$ is positive (negative) definite.*

REMARK 3.2. Recall that a Hermitian matrix pair (A, B) is *definite* (e.g., [40]) if there exist real constants α, β such that $\alpha A + \beta B$ is a positive definite matrix. In this case, if $\alpha \neq 0$, then (A, B) is a positive or negative definite pair with $\lambda_0 = -\frac{\beta}{\alpha}$; if $\alpha = 0$, then $\beta \neq 0$, and B is either a positive or negative definite matrix. If the pair (A, B) is positive definite, then the pair $(-A, B)$ is negative definite, so from now on we deal only with positive definite matrix pairs.

A definite matrix pair can be diagonalized by a congruence transformation. The following theorem reveals the natural structure of a positive definite matrix pair.

THEOREM 3.3. *[39, 42] Let (A, B) be a positive definite matrix pair of order n such that B has inertia* $\mathrm{In}(B) = (n_+, n_-, n_0)$.

i) There exists a nonsingular W such that

$$W^H AW = \begin{bmatrix} \Lambda_+ & & \\ & -\Lambda_- & \\ & & I_{n_0} \end{bmatrix}, \quad W^H BW = \begin{bmatrix} I_{n_+} & & \\ & -I_{n_-} & \\ & & 0_{n_0} \end{bmatrix},$$

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

$$\lambda_{n_-}^- \leq \cdots \leq \lambda_1^- < \lambda_1^+ \leq \cdots \leq \lambda_{n_+}^+. \tag{3.1}$$

ii) (A, B) has only real finite eigenvalues and the number of finite eigenvalues is $\mathrm{rank}(B) = n_+ + n_-$. Each eigenvalue λ_j^+, λ_j^- has an eigenvector x that satisfies $x^H Bx = 1$ and $x^H Bx = -1$, respectively.

iii) $A - \lambda_0 B$ is positive definite for every $\lambda_0 \in (\lambda_1^-, \lambda_1^+)$ and nowhere else.

Note that the definiteness of the pair (A, B) precludes defective eigenvalues. Therefore, since every eigenvalue λ_j^+ has a B-positive eigenvector, λ_j^+ is called a *B-positive* eigenvalue. Similar holds for λ_j^-. Any eigenvector x belonging to a finite eigenvalue of (A, B) cannot be B-neutral, so we can always B-normalize x. Eigenvectors belonging to different eigenvalues are B-orthogonal. Theorem 3.3 justifies the following definition.

DEFINITION 3.4. *Let $A - \lambda_0 B$ be positive definite, where (A, B) is a given Hermitian pair. The set of all such λ_0 is an open interval called the* definiteness interval [57]*, and every such λ_0 is called a* definitizing shift.

The following lemma guarantees a basic property of our algorithm.

LEMMA 3.5. *[38, Section 2] 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* $\mathrm{In}(X^H BX) =: (k_+, k_-, k_0)$ *and* $\mathrm{In}(U^H BU) =: (p_+, p_-, p_0)$. *Then*

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

Now we briefly give the theoretical background for our algorithm [38, Section 2]. Let $A, B \in \mathbb{C}^{n\times n}$ be Hermitian such that B has inertia $\mathrm{In}(B) = (n_+, n_-, n_0)$ and let (A, B) be

a positive definite matrix pair with finite eigenvalues (3.1) and infinite eigenvalues

$$\lambda_i^{\infty} := \infty \quad \text{for} \quad i = 1, \ldots, n_0. \tag{3.2}$$

Now, the definiteness interval equals $(\lambda_1^-, \lambda_1^+)$. Let $U \in \mathbb{C}^{n\times p}$ have full column rank. Then the projected matrix pair $(U^H AU, U^H BU)$ is also positive definite [38, Theorem 2.3]; hence its finite eigenvalues are real and can be ordered as follows:

$$\theta_{p_-}^- \le \cdots \le \theta_1^- < \theta_1^+ \le \cdots \le \theta_{p_+}^+,$$

with $\mathrm{In}(U^H BU) = (p_+, p_-, p_0)$. The eigenvalue interlacing properties [38, Theorem 2.3] hold

$$\lambda_i^+ \le \theta_i^+ \le \lambda_{i+n-p}^+ \quad \text{for} \quad i = 1, \ldots, p_+, \tag{3.3a}$$

$$\lambda_j^- \ge \theta_j^- \ge \lambda_{j+n-p}^- \quad \text{for} \quad j = 1, \ldots, p_-, \tag{3.3b}$$

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

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

for some integers k_+, k_- satisfying $(k_+, k_-, 0) \le \mathrm{In}(B)$, where the inequality is understood elementwise. Now assume $(k_+, k_-, 0) \le \mathrm{In}(U^H BU)$. Then applying the trace minimization principle [37, 42] to (A, B) and $(U^H AU, U^H BU)$, and using the eigenvalue interlacing properties (3.3) we have

$$\min_{\substack{X \in \mathbb{C}^{n\times k} \\ X^H BX = J_k}} \operatorname{trace}(X^H AX) = \sum_{i=1}^{k_+} \lambda_i^+ - \sum_{j=1}^{k_-} \lambda_j^- \tag{3.4a}$$

$$\le \sum_{i=1}^{k_+} \theta_i^+ - \sum_{j=1}^{k_-} \theta_j^- \tag{3.4b}$$

$$= \min_{\substack{Y \in \mathbb{C}^{p\times k} \\ Y^H (U^H BU) Y = J_k}} \operatorname{trace}\left(Y^H (U^H AU) Y\right), \tag{3.4c}$$

with equality if and only if U is spanned by the eigenvectors of the pair (A, B) belonging to $\lambda_1^+, \ldots, \lambda_{k_+}^+$ and $\lambda_1^-, \ldots, \lambda_{k_-}^-$. These eigenvectors are columns in the minimizing matrix $X_{\min}$ of the function in (3.4a).

For small given integers $k_\pm$, our *aim* is to determine the minimum and the minimizing matrix $X_{\min}$ of the function given in (3.4a), that is, to find the k_+ smallest B-positive eigenvalues $\lambda_1^+, \ldots, \lambda_{k_+}^+$, the k_- largest B-negative eigenvalues $\lambda_1^-, \ldots, \lambda_{k_-}^-$, and the corresponding eigenvectors of the positive definite pair (A, B). Considering (3.4), we find approximations of the wanted eigenpairs from the chosen subspace $\mathcal{U} = \operatorname{span} U$ by using the Rayleigh–Ritz procedure. We compute all eigenpairs of the small projected pair $(U^H AU, U^H BU)$ and extract only those around its definiteness interval $\theta_j^\pm$, $j = 1, \ldots, k_\pm$ and the corresponding eigenvectors $y_j^\pm$, $j = 1, \ldots, k_\pm$. These eigenvectors are assumed to be normalized such that

$$Y^H (U^H BU) Y = J_k, \quad \text{where} \quad Y = [y_{k_+}^+, \ldots, y_1^+, y_1^-, \ldots, y_{k_-}^-].$$

The *Ritz pairs* of the matrix pair (A, B) with respect to the subspace $\mathcal{U}$ are then given by $(\theta_j^\pm, U y_j^\pm)$, $j = 1, \ldots, k_\pm$. Therefore, the matrix $X := UY$ of the Ritz vectors has B-orthonormal columns.

Algorithm 3.1 An indefinite variant of the (m)-scheme with one preconditioner, $m \geq 2$

Input: $A, B \in \mathbb{C}^{n\times n}$: coefficients of a positive definite pair (A, B) with indefinite B;
$T \in \mathbb{C}^{n\times n}$: Hermitian positive definite preconditioner;
$X^{(0)} \in \mathbb{C}^{n\times k}$: initial guess such that $(k_+, k_-, 0) \leq \operatorname{In}\big[(X^{(0)})^H B X^{(0)}\big]$.
Output: $\ell_+ \leq k_+$ smallest B-positive eigenpairs and $\ell_- \leq k_-$ largest B-negative eigenpairs.

1: B-orthonormalize $X^{(0)}$.
2: Apply the Rayleigh–Ritz procedure to (A, B) with respect to the subspace $\operatorname{span} X^{(0)}$, and let $\Theta^{(0)}$ be the diagonal matrix containing the wanted Ritz values.
3: **Initialization**: If $m \geq 3$, then compute an initial sequence of $m-2$ matrices $X^{(1)}, \ldots, X^{(m-2)}$ by executing single steps of the (j)-scheme with the initial sequence $X^{(0)}, \ldots, X^{(j-2)}$ for $j = 2, \ldots, m-1$.
4: **Iteration**:
5: **for** $i = m-2, m-1, m, \ldots$ **do**
6: Compute preconditioned residual $W^{(i)} \leftarrow T(AX^{(i)} - BX^{(i)}\Theta^{(i)})$.
7: **if** all desired eigenvalues are converged **then**
8: Exit loop.
9: **end if**
10: Set the subspace $\mathcal{U}^{(i)} \leftarrow \operatorname{span} U^{(i)} = \operatorname{span}\big[X^{(i)}, W^{(i)}, X^{(i-1)}, \ldots, X^{(i-m+2)}\big]$.
11: B-orthonormalize $U^{(i)}$.
12: Apply the Rayleigh–Ritz procedure to (A, B) with respect to the subspace $\mathcal{U}^{(i)}$ and let $X^{(i+1)}$ be the matrix of the Ritz vectors corresponding to k_+ smallest B-positive and k_- largest B-negative Ritz values and let $\Theta^{(i+1)}$ be the diagonal matrix containing the wanted Ritz values.
13: **end for**

Now we need to specify a subspace $\mathcal{U}$, which can be chosen in many ways. *An indefinite variant of the (m)-scheme with one preconditioner* is presented in Algorithm 3.1, similarly to a unifying framework suggested in [43, 44] for a class of preconditioned gradient type eigensolvers for computing the smallest eigenpair of a GEP with real symmetric positive definite matrices. Let

$$X^{(0)}, \ldots, X^{(m-2)} \in \mathbb{C}^{n\times k}, \quad k = k_+ + k_-,$$

be an initial sequence of matrices of approximations of the wanted eigenvectors with $m \geq 2$ a small fixed integer. In the ith $(i \geq m-2)$ iteration of Algorithm 3.1 we consider the subspace

$$\mathcal{U}^{(i)} = \operatorname{span} U^{(i)} := \operatorname{span}\big[X^{(i)}, W^{(i)}, X^{(i-1)}, X^{(i-2)}, \ldots, X^{(i-m+2)}\big], \tag{3.5}$$

(a matrix $X^{(-j)}$ is an empty matrix) with the *preconditioned residual matrix*

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

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

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

Notice that the dimension of the subspace $\mathcal{U}^{(i)}$ is not larger than mk for all $i = 0, 1, 2, \ldots$.

Every new subspace $\mathcal{U}^{(i)}$ in Algorithm 3.1 contains column vectors of the current iteration matrix $X^{(i)}$ containing Ritz vectors. Therefore, the eigenvalue interlacing properties guarantee

$\theta_j^{+(i+1)} \le \theta_j^{+(i)}$, $j = 1, \ldots, k_+$, that is, the B-positive Ritz values are monotonically decreasing, and $\theta_j^{-(i)} \le \theta_j^{-(i+1)}$, $j = 1, \ldots, k_-$, that is, the B-negative Ritz values are monotonically increasing for any preconditioner, not necessarily positive definite. Therefore, our iteration method is trace-reducing and robust with respect to the choice of the preconditioner.

An indefinite LOBPCG method with one preconditioner [38, Algorithm 1] coincides with Algorithm 3.1 with $m = 3$.

We now give some remarks on the use and implementation of Algorithm 3.1, similarly to [38, Section 3.1]. The user chooses $m \ge 2$, which determines the dimension of the search subspace.

Initial guess. The user needs to give an initial guess $X^{(0)}$ such that

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

holds (the inequality is understood elementwise), where ℓ_+ (ℓ_-) denotes the number of the wanted smallest B-positive (largest B-negative) eigenpairs. The inequality $k_\pm \le p_\pm$ for $\mathrm{In}\big((U^{(i)})^H B U^{(i)}\big) = (p_+, p_-, p_0)$ holds for all iterations; follows by induction from Lemma 3.5. Notice that p_+ and p_- may vary during the iterative process. Hence we need precise information about the number of eigenvalues for both types in the projected problem in order to extract the desired Ritz values. This is quite different compared to the standard LOBPCG algorithm, in which the desired Ritz values are always the smallest ones. Fortunately, we have knowledge about p_+ and p_- since the inertia of $(U^{(i)})^H B U^{(i)}$ is available as a byproduct of the B-orthogonalization procedure; see below.

When B has a particular structure, as in many applications, choosing such an initial guess is straightforward; see Examples 6.1–6.3.

We allow $X^{(i)}$ to have more than ℓ_+ B-positive and more than ℓ_- B-negative columns, which can be useful when we have a cluster of eigenvalues. In this way, the dimension of the search subspace is bigger, which can lead to faster convergence.

Initialization. The current subspace $\mathcal{U}^{(i)}$ depends on the span of $m-2$ matrices of the previous Ritz vectors. Therefore, if $m \ge 3$, we need to compute these matrices before proceeding with the iteration. For example, if $m = 4$, we need to compute an initial sequence of two matrices $X^{(1)}, X^{(2)}$ by executing single steps of the indefinite variant of the (j)-scheme with the initial sequence $X^{(0)}, X^{(j-2)}$ for $j = 2, 3$. This means that we execute only the first iteration step ($i = 0$) of the (2)-scheme to get $X^{(1)}$ from $\mathrm{span}[X^{(0)}, W^{(0)}]$. Then, we execute only the first iteration step ($i = 1$) of the (3)-scheme to get $X^{(2)}$ from $\mathrm{span}[X^{(1)}, W^{(1)}, X^{(0)}]$. Finally, we can proceed with the iteration in the (4)-scheme, where in the first iteration step ($i = 2$) we have $\mathrm{span}[X^{(2)}, W^{(2)}, X^{(1)}, X^{(0)}]$.

Choosing the basis. The natural basis $[X^{(i)}, W^{(i)}, X^{(i-1)}]$ for $\mathcal{U}^{(i)}$ in the indefinite LOBPCG method (as in the standard LOBPCG method) is ill-conditioned. To improve numerical stability a new basis is chosen [33, 38]: the matrix $X^{(i-1)}$ is replaced by the matrix $P^{(i)}$. The columns of the matrix $P^{(i)}$ are given as an implicit difference of the corresponding columns of the matrices $X^{(i)}$ and $X^{(i-1)}$. More precisely, let the $3k \times k$ matrix $Y^{(i+1)}$ containing the desired eigenvectors of the projected pair $((U^{(i)})^H A U^{(i)}, (U^{(i)})^H B U^{(i)})$ be partitioned as

$$Y^{(i+1)} = \begin{bmatrix} Y_1^{(i+1)} \\ Y_2^{(i+1)} \end{bmatrix}, \quad Y_1^{(i+1)} \in \mathbb{C}^{k\times k}, \quad Y_2^{(i+1)} \in \mathbb{C}^{2k\times k}.$$

Update ($P^{(0)} \leftarrow []$)

$$P^{(i+1)} \leftarrow [W^{(i)}, X^{(i-1)}] Y_2^{(i+1)}, \qquad X^{(i+1)} \leftarrow X^{(i)} Y_1^{(i+1)} + P^{(i+1)}.$$

Since $P^{(i+1)} = X^{(i+1)} - X^{(i)}Y_1^{(i+1)}$, then in the indefinite variant of the (m)-scheme we use the new basis $[X^{(i)}, W^{(i)}, P^{(i)}, P^{(i-1)}, \ldots, P^{(i-m+3)}]$ instead of the natural basis $[X^{(i)}, W^{(i)}, X^{(i-1)}, X^{(i-2)}, \ldots, X^{(i-m+2)}]$.

B-orthonormality. The Ritz pairs extracted from the subspace $\mathcal{U}^{(i)}$ do not depend on the choice of the basis. For the standard LOBPCG method it has been observed that choosing an orthonormal basis leads to improved numerical stability [18, 26]. In [38], the authors made a similar observation when choosing a B-orthonormal basis in the indefinite LOBPCG method. We prefer B-orthonormalization rather than standard orthonormalization (in the Euclidean inner product) in our algorithm for the following reasons. The first reason is the natural structure of the GEP we are interested in: since the eigenvectors corresponding to different eigenvalues of a definite matrix pair are B-orthogonal, we would like the approximations of the wanted eigenvectors (in our algorithm, these are Ritz vectors) to be B-orthogonal as well. Furthermore, we want to solve the trace minimization problem (3.4), where we have the B-orthonormality property $X^H BX = \mathrm{diag}(\pm 1)$. Once the basis has been constructed, we form the projected pair $((U^{(i)})^H AU^{(i)}, (U^{(i)})^H BU^{(i)})$, where, in general, $(U^{(i)})^H AU^{(i)}$ is a full matrix, and in theory $(U^{(i)})^H BU^{(i)} = \mathrm{diag}(\pm 1)$. Therefore, we can apply the J-Jacobi method of Veselić [56] to the small projected pair.

The B-orthonormalization process [38, Section 3.1] needs to be implemented carefully to avoid numerical instability. Notice that this process in not always possible due to the existence of B-neutral vectors. Further, for a vector close to a B-neutral vector, forcing B-normalization will lead to a large growth factor. To avoid that, we can use a preprocessing step [38, Section 3.1] and drop the problematic vectors from the basis; see also the discussion in [18, Section 4.1] for the standard LOBPCG. By Lemma 3.5, when orthogonalizing $U^{(i)}$ from (3.5) with $(X^{(i)})^H BX^{(i)} = \mathrm{diag}(\pm 1)$, the output $U^{(i)}_{\mathrm{drop}}$ can be chosen of the form $[X^{(i)}, Z^{(i)}]$ (consequently, we have $\mathrm{rank}(X^{(i)}) \leq \mathrm{rank}(U^{(i)}_{\mathrm{drop}}) \leq \mathrm{rank}(U^{(i)})$), which is enough for a search subspace as long as $Z^{(i)}$ contains some columns to keep the algorithm working. If too many columns must be dropped, the basis can be padded with randomly generated B-orthogonalized columns.

Deflation. A Ritz pair is deflated after it has converged to the desired accuracy. In our algorithm, we use a "deflate from the middle" strategy: a Ritz value $\theta_j^{\pm}$ (here we drop the superscript (i) for clarity) is deflatable if and only if all Ritz values $\theta_i^{\pm}$, with $1 \leq i \leq j-1$, are deflatable and*

$$\|r_j^{\pm}\|_2 := \|Ax_j^{\pm} - \theta_j^{\pm} Bx_j^{\pm}\|_2 \leq \mathsf{tol} \cdot |\theta_j^{\pm}| \, \|B\|_2 \|x_j^{\pm}\|_2, \tag{3.6}$$

where $x_j^{\pm} := Uy_j^{\pm}$ is the corresponding Ritz vector and tol is a tolerance specified by the user. Deflated Ritz vectors, as in the standard LOBPCG method [34], do not participate in the computation of $W^{(i)}$ or $P^{(j)}$, $j = i-m+3, \ldots, i$, but they still need to participate in the B-orthonormalization process to avoid repeated convergence to the same eigenvalue. The bound (3.6) is used in a backward error analysis of approximate eigenpairs [38, Section 3.4]. For large sparse matrices an estimate of the 2-norm is used. We refer to [18, Section 4] for this issue. The 1-norm of the matrix can also be used; see, e.g., [41] for a hyperbolic quadratic eigenvalue problem.

Preconditioner. The preconditioned residuals (here we drop the superscript (i) for clarity) $w_j^{\pm} := Tr_j^{\pm}$ are obtained by solving linear systems $T^{-1}w_j^{\pm} = r_j^{\pm}$ for $j = 1, \ldots, k_{\pm}$. Usually, these systems are solved only approximately by some iterative method such as the linear

*Although we use a superscript $+$ ($-$) in a residual r_j^+ (r_j^-), this does not imply that r_j^+ (r_j^-) is B-positive (B-negative). Similar holds for a preconditioned residual $w_j^{\pm}$.

conjugate gradient method (CG) [17] with a low drop tolerance. Ideally, one application of a preconditioner to a residual vector has the same costs as one matrix-vector product. We use a Hermitian positive definite preconditioner T for three reasons. Since we want to compute the interior eigenvalues around the definiteness interval, we use a definitizing shift λ_0 for which the matrix $A - \lambda_0 B$ is positive definite, and therefore we use a Hermitian positive definite preconditioner $T = (A - \lambda_0 B)^{-1}$ or $T \approx (A - \lambda_0 B)^{-1}$. The second reason is the efficiency of a positive definite preconditioner such as a symmetric positive definite multigrid preconditioner used in the discretization of an elliptic PDE-eigenproblem. The third reason lies in the fact that the convergence analysis (see Section 4.1) is given only for a symmetric positive definite preconditioner. Since our Algorithm 3.1 is robust with the choice of the preconditioner, we can even choose the shift λ_0 outside of the definiteness interval, which results in an indefinite preconditioner [33], [38, Example 5.2].

TABLE 3.1
An indefinite variant of the (m)-scheme for $m = 2$ and $m = 3$.

m	k_-	k_+	Eigensolver	m	k_-	k_+	Eigensolver
2	1	0	PSA$^-$	2	0	1	PSD$^+$
2	> 1	0	BPSA$^-$	2	0	> 1	BPSD$^+$
3	1	0	LOPCG$^-$	3	0	1	LOPCG$^+$
3	> 1	0	LOBPCG$^-$	3	0	> 1	LOBPCG$^+$
2	1	1	indefinite PSD/A	3	1	1	indefinite LOPCG
2	> 1	> 1	indefinite BPSD/A	3	> 1	> 1	indefinite LOBPCG

In Table 3.1, we appoint an eigensolver for $m = 2$ and $m = 3$ from our algorithm. The names of our methods point to a strong relationship with existing methods for the partial GEP for a matrix pair (A, B) with positive definite B. The abbreviation BPSA stands for the block (or subspace) preconditioned steepest ascent iteration. In our methods, superscripts $+$ and $-$ mean that iterations operate not on the whole Euclidean space, but only on the B-positive and B-negative subsets, respectively.

During the past few years several variants of extended (B)PSD/A and LO(B)PCG type methods were proposed for nonlinear Hermitian eigenvalue problems with variational characterizations [9, 41, 55]. The mentioned eigenvalue problems include definite matrix pairs. A so-called interval of definite type: positive[†] or negative [55, Definition 2.1, Proposition 2.4] is chosen. Then, in a single vector version, the algorithms in [9, 41, 55] compute the smallest (or the largest) eigenvalue in that interval and the corresponding eigenvector. In block versions, the algorithms compute a few smallest (or largest) eigenvalues in that interval and the corresponding eigenvectors. Our Algorithm 3.1 with $k_\pm \geq 1$ *simultaneously* computes the eigenvalues on both sides of the definiteness interval and the corresponding eigenvectors of a positive definite matrix pair. Algorithm 3.1 with $m = 2, 3$ and $k_+ \geq 1$ and $k_- = 0$, or $k_+ = 0$ and $k_- \geq 1$, coincides with the corresponding algorithms in [9, 55]. For example, our LOPCG$^+$ with $k_+ = 1$ coincides with [9, Algorithm 2.1, LOCG(1,2), with a fixed preconditioner and $F(\lambda) = \lambda B - A$]. The mentioned *extended* variants of (B)PSD/A and LO(B)PCG type methods in [9, 41], as discussed in [22, 50, 51], mean that, for example in PSD (2.2), the search subspace $\mathrm{span}[x^{(i)}, T(\widetilde{A} - \widetilde{\rho}(x^{(i)})B)x^{(i)}]$ is replaced with the m_eth order Krylov subspace (usually with a small m_e; here $\widetilde{\rho}^{(i)} := \widetilde{\rho}(x^{(i)})$)

$$\mathcal{K}^{m_e}(T(\widetilde{A} - \widetilde{\rho}^{(i)}B); x^{(i)}) := \mathrm{span}\big[x^{(i)}, T(\widetilde{A} - \widetilde{\rho}^{(i)}B)x^{(i)}, \ldots, (T(\widetilde{A} - \widetilde{\rho}^{(i)}B))^{m_e-1}x^{(i)}\big].$$

[†]For example, (λ_0, ∞) is an interval of positive type of a positive definite matrix pair, where λ_0 is a definitizing shift.

The global convergence of the method using $\mathcal{K}^{m_e}(T(\widetilde{A}-\widetilde{\rho}^{(i)}B);x^{(i)})$ as a search subspace with $T=I$ to some eigenvalue of a definite matrix pair $(\widetilde{A},B)$ with $B\succ 0$ is proven in [22, Theorem 1]. The global convergence of the LOPCG type method to the extremal eigenpair, for a general Hermitian matrix polynomial with variational characterizations, without any assumption on a preconditioner (including variable preconditioners), is proven in [9, Theorem 2.1]. In particular, an asymptotic estimate for our indefinite LOPCG$^+$ method with a fixed positive definite preconditioner is given in [9, Theorem 3.1, LOCG(1,2)].

It is natural to expect (as confirmed by our convergence analysis from Section 4.1) that the eigenvalues closest to the definitizing shift λ_0 converge first. Since we want to compute the eigenvalues around the definiteness interval, to be more efficient, we can use two definitizing shifts, i.e., λ_0^+ close to λ_1^+ and λ_0^- close to λ_1^-. Therefore, we can use two preconditioners $T_+\approx(A-\lambda_0^+B)^{-1}$ and $T_-\approx(A-\lambda_0^-B)^{-1}$, exactly as in [38, Algorithm 2], and get *an indefinite variant of the (m)-scheme with two preconditioners*. To preserve the dimension of the search subspace, we use

$$\mathcal{U}^{(i)}:=\operatorname{span}\big[X^{(i)},T_+\cdot R_+^{(i)},T_-\cdot R_-^{(i)},X^{(i-1)},\dots,X^{(i-m+2)}\big]$$

as a new subspace, where we split the residual matrix $R^{(i)}$ into two parts $R_+^{(i)}$ and $R_-^{(i)}$ associated with B-positive Ritz values and B-negative Ritz values, respectively.

4. The simplest variants: PSD$^+$ and PSA$^-$. In this section, we consider the simplest variants in Algorithm 3.1, namely PSD$^+$ and PSA$^-$. When B is positive definite, then the PSD (PSA) iteration aims to compute the smallest (the largest) eigenvalue and the corresponding eigenvector of a matrix pair (A,B). If $B\succ 0$, then PSD = PSD$^+$; if $B\prec 0$, then PSA = PSA$^-$. Therefore, when B is indefinite, as in our case, the PSD$^+$ (PSA$^-$) iteration aims to compute the smallest (the largest) B-positive (B-negative) eigenvalue and the corresponding eigenvector. To give convergence theorems for our methods PSD$^+$ and PSA$^-$, we use the convergence theorem [45] derived for the PSD iteration that is proven only for real symmetric matrices. Therefore, only in this section we assume that A and B are real and symmetric.

Let

$$\rho(x)=\frac{x^TAx}{x^TBx},\quad x^TBx\neq 0,$$

denote the Rayleigh quotient associated with a positive definite matrix pair (A,B). The single vector iteration PSD$^+$ (replacing $X^{(i)}$, $W^{(i)}$, and $X^{(i+1)}$ by x_+, w_+, and x'_+, respectively) uses the subspace $\mathcal{U}^{(i)}=\operatorname{span}[x_+,w_+]$ in the ith iteration step[‡]. A Ritz pair of (A,B) with respect to the subspace $\mathcal{U}^{(i)}$ is $(x'_+,\rho(x'_+))$, where $\rho(x'_+)$ is the smaller B-positive Ritz value. Similar holds for the PSA$^-$ iteration.

REMARK 4.1. The indefinite PSD/A iteration combines the PSD$^+$ and the PSA$^-$ iterations into one iteration method by using the subspace $\mathcal{U}^{(i)}=\operatorname{span}[X^{(i)},W^{(i)}]$, where $X^{(i)}=[x_+,x_-]$ contains the current approximations of the wanted eigenvectors corresponding to λ_1^+ and λ_1^-. The subspace $\mathcal{U}^{(i)}$ in Algorithm 3.1 with $m=3$ and $k_\pm=1$, that is, the indefinite LOPCG iteration, additionally contains the column space of the previous iterate. Therefore, the indefinite PSD/A iteration can be interpreted as a truncated version of the indefinite LOPCG iteration. Similarly, higher order schemes in Algorithm 3.1 with $m>3$ and $k_\pm=1$, can be interpreted as an extended version of the indefinite LOPCG method. Similar holds for the block versions.

[‡]If x_+ and w_+ are linearly dependent, then the iteration terminates in the current iterate x_+.

4.1. Convergence theorems for PSD$^+$ and PSA$^-$. A sharp convergence theorem for the standard PSD iteration applied to a real symmetric matrix pair $(\widetilde{A}, B)$, where $\widetilde{A}$ is positive definite, is given in [45], using the dual matrix pair $(B, \widetilde{A})$. To give corresponding convergence theorems for eigenvalues around the definiteness interval of a positive definite matrix pair (A, B) with indefinite B computed by PSD$^+$ and PSA$^-$, we need to present them as the extremal eigenvalues of some auxiliary matrix pair.

Consider a positive definite matrix pair (A, B) with eigenvalues (3.1), (3.2). Let λ_0 be a *given* definitizing shift and let

$$\widetilde{A} = A - \lambda_0 B \succ 0, \quad \widetilde{\lambda} = \lambda - \lambda_0 \neq 0.$$

Hence for the eigenproblem

$$Bx = \widetilde{\mu}\widetilde{A}x \tag{4.1}$$

the eigenvalues $\{\widetilde{\mu}\}$ are given by

$$\widetilde{\mu} = 1/\widetilde{\lambda} = 1/(\lambda - \lambda_0)$$

and arranged in the order:

$$\begin{array}{ccccccccccccc} 1/\widetilde{\lambda}_1^- & \leq & \cdots & \leq & 1/\widetilde{\lambda}_{n_-}^- & < & 0 = 1/\widetilde{\lambda}_1^\infty = \cdots = 1/\widetilde{\lambda}_{n_0}^\infty & < & 1/\widetilde{\lambda}_{n_+}^+ & \leq & \cdots & \leq & 1/\widetilde{\lambda}_1^+ \\ \| & & & & \| & & & & \| & & & & \| \\ \widetilde{\mu}_1^- & \leq & \cdots & \leq & \widetilde{\mu}_{n_-}^- & < & 0 = \widetilde{\mu}_1^\infty = \cdots = \widetilde{\mu}_{n_0}^\infty & < & \widetilde{\mu}_{n_+}^+ & \leq & \cdots & \leq & \widetilde{\mu}_1^+ \end{array}.$$

Let

$$\widetilde{\mu}(x) = \frac{x^T B x}{x^T \widetilde{A} x} = \frac{1}{\rho(x) - \lambda_0} \in \mathbb{R}, \quad x \neq 0, \tag{4.2}$$

denote the Rayleigh quotient associated with the matrix pair $(B, \widetilde{A})$ from (4.1). Therefore, the goal of the PSD$^+$ iteration is equivalent to compute the largest eigenvalue $\widetilde{\mu}_1^+$ by maximizing $\widetilde{\mu}(x)$ from (4.2), and the corresponding eigenvector. Similar holds for the PSA$^-$ iteration. Since we want to compute extremal eigenvalues $\widetilde{\mu}_1^+$ and $\widetilde{\mu}_1^-$, we can use the PSD$_{\widetilde{\mu}}$ (4.3) iteration and then apply [45, Theorem 2.2] to the matrix pair $(B, \widetilde{A})$ and $(-B, \widetilde{A})$ from (4.1), respectively. The transformation of the PSD iteration (2.2) (after multiplication by $\widetilde{\mu}(x) = 1/\widetilde{\rho}(x)$, replacing $x^{(i+1)}$, $x^{(i)}$, and $\tau^{(i)}$ by x', x, and τ_{opt}, respectively) is

$$\text{(PSD}_{\widetilde{\mu}}\text{)} \qquad \widetilde{\mu}(x)x' = \widetilde{\mu}(x)x + \tau_{\text{opt}} T(Bx - \widetilde{\mu}(x)\widetilde{A}x) \tag{4.3a}$$

with the optimal step length

$$\tau_{\text{opt}} := \arg\max_{\tau \in \mathbb{R}} \widetilde{\mu}\big(\widetilde{\mu}(x)x + \tau T(Bx - \widetilde{\mu}(x)\widetilde{A}x)\big). \tag{4.3b}$$

For a symmetric positive definite matrix T, a preconditioner, which approximates the inverse of the positive definite $\widetilde{A}$ one assumes [45, Section 1.1]

$$\|I - T\widetilde{A}\|_{\widetilde{A}} \leq \gamma, \quad \gamma \in [0, 1). \tag{4.4}$$

Neymeyr gives a convergence estimate of the poorest possible convergence of the PSD iteration (2.2) in [45, Theorem 1.2]; the proof of that theorem takes 12 pages. That estimate is sharp in the sense that an initial guess and a preconditioner T satisfying the inequality (4.4) can be chosen such that the bound is attained. [45, Theorem 1.2] guarantees the monotone

convergence of a sequence of the Ritz values to some eigenvalue; to the smallest eigenvalue only if the Ritz values have reached the final interval $[\lambda_1, \lambda_2)$, but, due to roundoff, in practice that sequence almost surely converge to the smallest eigenvalue. A similar convergence theorem for the PSD$^+$ and PSA$^-$ iteration is as follows:

THEOREM 4.2. *Let $x_+ \in \mathbb{R}^n$ and x'_+ be the PSD$^+$ iterate. The preconditioner T is assumed to satisfy (4.4). If $\lambda_i^+ \leq \rho(x_+) < \lambda_{i+1}^+$, $i = 1, \ldots, n_+ - 1$, then $\rho(x'_+) \leq \rho(x_+)$ and either $\rho(x'_+) \leq \lambda_i^+$ or*

$$\frac{\rho(x'_+) - \lambda_i^+}{\lambda_{i+1}^+ - \rho(x'_+)} \leq \sigma_{i,+}^2 \frac{\rho(x_+) - \lambda_i^+}{\lambda_{i+1}^+ - \rho(x_+)} \tag{4.5}$$

$$\text{with} \quad \sigma_{i,+} := \frac{\kappa_{i,+} + \gamma(2 - \kappa_{i,+})}{(2 - \kappa_{i,+}) + \gamma\kappa_{i,+}} \quad \text{and} \quad \kappa_{i,+} := \frac{(\lambda_i^+ - \lambda_0)(\lambda_1^- - \lambda_{i+1}^+)}{(\lambda_{i+1}^+ - \lambda_0)(\lambda_1^- - \lambda_i^+)}.$$

The estimate is sharp and can be attained for $\rho(x_+) \to \lambda_i^+$ in the three-dimensional invariant subspace associated with the eigenvalues λ_i^+, λ_{i+1}^+, and λ_1^-.

Proof. The proof follows from [45, Theorem 2.2] applied to the matrix pair $(B, \widetilde{A})$ from (4.1) with associated substitutions $\sigma \to \sigma_{i,+}$, $\mu(x) \to \widetilde{\mu}(x_+) = 1/(\rho(x_+) - \lambda_0)$, $(\mu_n, \mu_{i+1}, \mu_i, \mu_1) \to (\widetilde{\mu}_1^-, \widetilde{\mu}_{i+1}^+, \widetilde{\mu}_i^+, \widetilde{\mu}_1^+)$, $\mu_j \to \widetilde{\mu}_j^{\pm} = 1/(\lambda_j^{\pm} - \lambda_0)$, and $\kappa \to \kappa_{i,+}$. ∎

THEOREM 4.3. *Let $x_- \in \mathbb{R}^n$ and x'_- be the PSA$^-$ iterate. The preconditioner T is assumed to satisfy (4.4). If $\lambda_{i+1}^- < \rho(x_-) \leq \lambda_i^-$, $i = 1, \ldots, n_- - 1$, then $\rho(x'_-) \geq \rho(x_-)$ and either $\rho(x'_-) \geq \lambda_i^-$ or*

$$\frac{\lambda_i^- - \rho(x'_-)}{\rho(x'_-) - \lambda_{i+1}^-} \leq \sigma_{i,-}^2 \frac{\lambda_i^- - \rho(x_-)}{\rho(x_-) - \lambda_{i+1}^-} \tag{4.6}$$

$$\text{with} \quad \sigma_{i,-} := \frac{\kappa_{i,-} + \gamma(2 - \kappa_{i,-})}{(2 - \kappa_{i,-}) + \gamma\kappa_{i,-}} \quad \text{and} \quad \kappa_{i,-} := \frac{(\lambda_i^- - \lambda_0)(\lambda_1^+ - \lambda_{i+1}^-)}{(\lambda_{i+1}^- - \lambda_0)(\lambda_1^+ - \lambda_i^-)}.$$

The estimate is sharp and can be attained for $\rho(x_-) \to \lambda_i^-$ in the three-dimensional invariant subspace associated with the eigenvalues λ_i^-, λ_{i+1}^-, and λ_1^+.

Proof. The proof follows from [45, Theorem 2.2] applied to the matrix pair $(-B, \widetilde{A})$ from (4.1) with associated substitutions $\sigma \to \sigma_{i,-}$, $\mu(x) \to -\widetilde{\mu}(x_-) = -1/(\rho(x_-) - \lambda_0)$, $(\mu_n, \mu_{i+1}, \mu_i, \mu_1) \to (-\widetilde{\mu}_1^+, -\widetilde{\mu}_{i+1}^-, -\widetilde{\mu}_i^-, -\widetilde{\mu}_1^-)$, $\mu_j \to -\widetilde{\mu}_j^{\pm} = -1/(\lambda_j^{\pm} - \lambda_0)$, and $\kappa \to \kappa_{i,-}$. ∎

For κ from [45, Theorem 2.2] holds $\kappa \in (0, 1)$, which implies $\kappa_{i,\pm} \in (0, 1)$. Since $\sigma_{i,+} = \sigma_{i,+}(\gamma, \kappa_{i,+})$ is a monotone increasing function in both variables, choosing smaller γ (meaning a preconditioner T approaches to the exact inverse of $\widetilde{A}$) and/or smaller $\kappa_{i,+}$ will lead to a faster convergence of the PSD$^+$ iteration. Similar holds for the PSA$^-$ iteration.

We now give an asymptotic estimate for (4.5). Provided that $\lambda_1^+ \leq \rho(x_+) < \lambda_2^+$ and (4.4) holds, asymptotically, as $\rho(x_+) \to \lambda_1^+$, we have $(\lambda_2^+ - \rho(x'_+))/(\lambda_2^+ - \rho(x_+)) \to 1$ and therefore

$$\frac{\rho(x'_+) - \lambda_1^+}{\rho(x_+) - \lambda_1^+} \lesssim \sigma_+^2$$

$$\text{with} \quad \sigma_+ := \sigma_{1,+} = \frac{\kappa_+ + \gamma(2 - \kappa_+)}{(2 - \kappa_+) + \gamma\kappa_+} \quad \text{and} \quad \kappa_+ := \kappa_{1,+} = \frac{(\lambda_1^+ - \lambda_0)(\lambda_1^- - \lambda_2^+)}{(\lambda_2^+ - \lambda_0)(\lambda_1^- - \lambda_1^+)}.$$

Therefore, our PSD$^+$ iteration converges at least linearly with the asymptotic convergence factor σ_+^2 that depends on the gap between λ_1^+ and λ_0 relative to the gap between λ_2^+ and λ_0,

and of course on γ: the quality measure of the preconditioner T. If $\lambda_2^+ \approx \lambda_1^+$, then $\kappa_+ \approx 1$ leads to a slower convergence of the PSD$^+$ iteration. Asymptotically, when $\lambda_0 \to \lambda_1^+$, then $\kappa_+ \to 0$ and therefore $\sigma_+ \to \gamma$. Similar conclusions hold for (4.6).

Corresponding convergence theorems for the block iterations BPSD$^+$ and BPSA$^-$ can be derived from [47]. For practically important (indefinite) LO(B)PCG type methods there are still no sharp convergence estimates. Convergence theorems proven for the preconditioned inverse iteration (PINVIT) [36], PSD/A [45, 46, 47], our convergence theorems (for an indefinite case) can serve only as upper (non-sharp) estimates. Namely, the eigenvalue interlacing property implies that the Ritz value θ_1^+ (θ_1^-) computed in Algorithm 3.1 with $m = 3$ and $k_+ = 1, k_- = 0$ ($k_+ = 0, k_- = 1$) is at least as close to λ_1^+ (λ_1^-) as the Ritz value $\rho(x'_+)$ ($\rho(x'_-)$) computed by PSD$^+$ (PSA$^-$) since the subspace of PSD$^+$ and/or PSA$^-$ is contained in the subspace of Algorithm 3.1 with $m = 3$. Therefore, Algorithm 3.1 with $m = 3$ and $k_+ = 1, k_- = 0$ ($k_+ = 0, k_- = 1$) converges at least linearly with the asymptotic convergence factor σ_+^2 (σ_-^2).

5. Arbitrary spectral gaps. Algorithm 3.1 (and its extension derived at the end of Section 3) simultaneously computes a few eigenvalues around the definiteness interval and the corresponding eigenvectors of a given positive definite matrix pair. However, the definiteness interval is just one special spectral gap. In this section, we derive some ideas how to use the trace-reducing Algorithm 3.1 to compute a few eigenvalues around *any spectral gap* and the corresponding eigenvectors of a *definite pair* (A, B).

Let $\lambda_{\min}$ denote the smallest and $\lambda_{\max}$ the largest finite eigenvalue of some positive definite matrix pair (A, B) with finite eigenvalues (3.1) and let $\mathcal{I}_0$ denote its definiteness interval[§]. For the given arbitrary shift $\lambda_a \in (\lambda_{\min}, \lambda_{\max}) \setminus \mathcal{I}_0$ that is not an eigenvalue of (A, B), let $\mathcal{I}_a$ be the *spectral gap* around λ_a, that is, $\lambda_a \in \mathcal{I}_a$ and either $\mathcal{I}_a = (\lambda_i^+, \lambda_{i+1}^+)$, for some $i \in \{1, \ldots, n_+ - 1\}$ or $\mathcal{I}_a = (\lambda_{j+1}^-, \lambda_j^-)$, for some $j \in \{1, \ldots, n_- - 1\}$. The spectral gap around λ_a is defined analogously for a negative definite matrix pair.

We want to simultaneously compute a small number of eigenvalues around the given shift λ_a, more precisely, the first j_b eigenvalues that are bigger than λ_a, the first j_s eigenvalues that are smaller than λ_a, and the corresponding eigenvectors of a definite pair (A, B). We transform the pair (A, B) into some auxiliary positive definite matrix pair with the definiteness interval around zero and then use Algorithm 3.1 to compute $j_b + j_s$ eigenvalues around zero and the corresponding eigenvectors of that auxiliary pair, and consequently, the wanted eigenvalues around λ_a and the corresponding eigenvectors of the pair (A, B). We propose two ways to transform the given definite pair to some auxiliary one[¶].

Suppose first that B is positive definite and λ_a is from any desired spectral gap $\mathcal{I}_a$. Now the matrix pair $(A - \lambda_a B, B)$ has the desired spectral gap $\mathcal{I} := \mathcal{I}_a - \lambda_a$ around zero and its eigenpair (μ, x) corresponds to the eigenpair $(\mu + \lambda_a, x)$ of the pair (A, B). As is immediately verified, $\mathcal{I}$ is the definiteness interval of the positive definite pair

$$(B^{-1}, (A - \lambda_a B)^{-1}), \tag{5.1}$$

which has the same eigenvalues as the pair $(A - \lambda_a B, B)$, the eigenvectors are just multiplied by B, and Algorithm 3.1 applies with $\ell_+ = j_b$, $\ell_- = j_s$. Here, the preconditioned residual is obtained by a matrix vector product since $T = B$. This case can be extended to any definite pair with indefinite matrices and *with the known definiteness interval, or, at least part of it*. Let λ_0 be a definitizing shift of a positive definite pair (A, B) with indefinite A and B. Let λ_a be

[§] If $B \preceq 0$, then $\mathcal{I}_0 = (\lambda_{\max}, \infty)$. If $B \succeq 0$, then $\mathcal{I}_0 = (-\infty, \lambda_{\min})$.

[¶] The author is indebted to Professor Krešimir Veselić for suggesting some of the ideas in the following, in particular for using (5.1) and providing Theorem 5.1 below.

from any desired spectral gap $\mathcal{I}_a$ of (A, B). We assume that λ_0 and λ_a are not relatively close. Then $A - \lambda_0 B$ is positive definite and the eigenpair (λ, x) of the pair (A, B) corresponds to the eigenpair $(\lambda - \lambda_0, x)$ of the pair $(A - \lambda_0 B, B)$. Now we can apply the previous case to the matrix pair $(B, A - \lambda_0 B)$ in which the eigenpair $(1/(\lambda - \lambda_0), x)$ corresponds to the eigenpair (λ, x) of the pair (A, B). First, move the spectrum of $(B, A - \lambda_0 B)$ to the left by $1/(\lambda_a - \lambda_0)$, and then use the inverses. Therefore, $\mathcal{I} := 1/(\mathcal{I}_a - \lambda_0) - 1/(\lambda_a - \lambda_0)$ is the definiteness interval around zero of the positive definite pair

$$\Big((A - \lambda_0 B)^{-1}, \big(B - (\lambda_a - \lambda_0)^{-1}(A - \lambda_0 B)\big)^{-1}\Big), \tag{5.2}$$

in which the eigenpair $(\frac{1}{\lambda - \lambda_0} - \frac{1}{\lambda_a - \lambda_0}, (A - \lambda_0 B)x)$ corresponds to the eigenpair (λ, x) of the pair (A, B). Here, the preconditioned residual is obtained by a matrix vector product since $T = A - \lambda_0 B$. After running Algorithm 3.1 with $\ell_+ = j_s, \ell_- = j_b$ (notice the change of the roles of j_s and j_b with this spectral transformation) on the pair (5.2), the computed eigenpair (μ, y) of the pair (5.2) corresponds to the eigenpair

$$\Big(\frac{1}{\mu + (\lambda_a - \lambda_0)^{-1}} + \lambda_0, (A - \lambda_0 B)^{-1} y\Big)$$

of the positive definite matrix pair (A, B) with indefinite A and B. A shortcoming of this way is that we must know the inverses or at least be able to solve easily linear systems of the type

$$Bx = c \quad \text{and} \quad (A - \lambda_a B)x = c$$

for matrices in (5.1); and similarly for (5.2). There is another way, i.e., by using corresponding decompositions of the matrices appearing in (5.1) and (5.2). We assume first for simplicity $B = I$. Now make the indefinite decomposition [‖]

$$A - \lambda_a I = GJG^H, \tag{5.3}$$

where λ_a is taken from any spectral gap $\mathcal{I}_a$ of A and J is Hermitian nonsingular, that is,

$$J^H = J^{-1} = J \tag{5.4}$$

and G is a nonsingular lower block-triangular matrix with diagonal blocks of order 1 or 2. If $J = I$ or $J = -I$, then $\lambda_a < \lambda_{\min}$ and $\lambda_a > \lambda_{\max}$, respectively, that is, λ_a is not from any spectral gap. In this case, we can proceed, and at the end, compute the extremal eigenvalues, i.e., $\lambda_{\min}$ or $\lambda_{\max}$ and the ones closest to them. However, we are really interested in an indefinite $J = \operatorname{diag}(\pm 1)$.

Consider the auxiliary matrix pair $(G^H G, J)$. Due to (5.4), this pair has the same eigenvalues as the matrix $A - \lambda_a I$ and it possesses a set of J-orthonormal eigenvectors. More precisely, we have the following theorem.

THEOREM 5.1. *Let $A \in \mathbb{C}^{n\times n}$ be the given Hermitian matrix and let $U \in \mathbb{C}^{n\times p}$, $p \le n$, have orthonormal columns spanning a spectral subspace of $A_a := A - \lambda_a I$, that is,*

$$A_a U = U\Lambda_a, \tag{5.5}$$

[‖] An indefinite decomposition of a Hermitian matrix H is a decomposition of the form $PHP^H = LDL^H$, where L is a unit lower triangular matrix, $D = D^H$ is a block-diagonal matrix with diagonal blocks of order 1 or 2, and P is a permutation matrix. This decomposition is obtained by the variants of the Bunch–Parlett decomposition [3, 11, 12, 13, 14, 16]. An additional diagonalization of the diagonal blocks in D and an appropriate scaling of the columns of L results in a new decomposition $PHP^H = GJG^H$, where G is a nonsingular lower block-triangular matrix with diagonal blocks of order 1 or 2, and J is a diagonal matrix of signs of the eigenvalues of D (or, equivalently, of H) on its diagonal. We must pay attention to which pivoting strategy to use when performing an indefinite decomposition; for a banded matrix a pivoting strategy needs to preserve the bandwidth during the process [13, 15, 28].

where $\Lambda_a := \Lambda - \lambda_a I_p$, $\Lambda \in \mathbb{R}^{p\times p}$ *is a diagonal matrix containing some of the eigenvalues of* A. *Set*

$$F = G^{-1}U|\Lambda_a|^{1/2} \text{ or, equivalently, } U = GF|\Lambda_a|^{-1/2}$$

with G, J *from* (5.3). *Then*

$$F^H JF = \mathcal{J}, \tag{5.6}$$

with $\mathcal{J} := \operatorname{sign}(\Lambda_a)$ *and*

$$G^H GF = JF\Lambda_a, \tag{5.7}$$

that is, F *spans the corresponding spectral subspace of the pair* $(G^H G, J)$. *Conversely, (5.7) and (5.6) imply (5.5) and the fact that* U *has orthonormal columns.*

Proof. Premultiply $A_a U = U\Lambda_a$ by A_a^{-1} and postmultiply by Λ_a^{-1} to obtain

$$U\Lambda_a^{-1} = A_a^{-1}U. \tag{5.8}$$

Now,

$$\begin{aligned} F^H JF &= |\Lambda_a|^{1/2}U^H G^{-H} JG^{-1}U|\Lambda_a|^{1/2} \\ &= |\Lambda_a|^{1/2}U^H A_a^{-1}U|\Lambda_a|^{1/2} \\ &= |\Lambda_a|^{1/2}U^H U\Lambda_a^{-1}|\Lambda_a|^{1/2} \\ &= |\Lambda_a|^{1/2}\Lambda_a^{-1}|\Lambda_a|^{1/2} = \mathcal{J}, \end{aligned} \tag{5.9}$$

with $\mathcal{J} = \operatorname{sign}(\Lambda_a)$. (5.9) follows from (5.8). Further, premultiply (5.3) by JG^{-1} to obtain

$$G^H = JG^{-1}A_a. \tag{5.10}$$

Now,

$$\begin{aligned} G^H GF &= G^H GG^{-1}U|\Lambda_a|^{1/2} = G^H U|\Lambda_a|^{1/2} \\ &= JG^{-1}A_a U|\Lambda_a|^{1/2} = JG^{-1}U\Lambda_a|\Lambda_a|^{1/2} \\ &= JG^{-1}U|\Lambda_a|^{1/2}|\Lambda_a|^{-1/2}\Lambda_a|\Lambda_a|^{1/2} \\ &= JF\Lambda_a. \end{aligned} \tag{5.11}$$

(5.11) follows from (5.10) and (5.5).
Conversely, using (5.3), (5.7) and $J^2 = I$ we have

$$\begin{aligned} A_a U &= GJG^H GF|\Lambda_a|^{-1/2} \\ &= GJJF\Lambda_a|\Lambda_a|^{-1/2} \\ &= GF|\Lambda_a|^{-1/2}|\Lambda_a|^{1/2}\Lambda_a|\Lambda_a|^{-1/2} \\ &= U\Lambda_a. \end{aligned}$$

Now,

$$U^H U = |\Lambda_a|^{-1/2}F^H G^H GF|\Lambda_a|^{-1/2}$$

$$= |\Lambda_a|^{-1/2}F^H JF\Lambda_a|\Lambda_a|^{-1/2} \tag{5.12}$$

$$= |\Lambda_a|^{-1/2}\operatorname{sign}(\Lambda_a)\Lambda_a|\Lambda_a|^{-1/2} = I_p. \tag{5.13}$$

(5.12) follows from (5.7) and (5.13) follows from (5.6). ∎

Moreover, $\mathcal{I} := \mathcal{I}_a - \lambda_a$ is the definiteness interval around zero of the positive definite pair $(G^H G, J)$ and Algorithm 3.1 with $\ell_+ = j_b$, $\ell_- = j_s$ applies. After running Algorithm 3.1, $\Lambda = \Lambda_a + \lambda_a I_p$ contains the wanted eigenvalues around λ_a and $U = GF|\Lambda_a|^{-1/2}$ contains the corresponding eigenvectors of the Hermitian matrix A. Instead of multiplying G^H by G, we modify Algorithm 3.1 to work with factor G in a way that $G^H G x$ is implemented by $G^H(Gx)$ and a preconditioned residual $w = Tr$ with $T = (G^H G)^{-1}$ is implemented by solving two linear systems $G^H z = r$ and $Gw = z$. The first matrix $(U^{(i)})^H G^H G U^{(i)}$ in a projected pair can be implemented by $(S^{(i)})^H S^{(i)}$, where $S^{(i)} := GU^{(i)}$.

If we have a pair (A, B) with B positive definite, then by making the Cholesky decomposition $B = LL^H$ the matrix G in the preceding theorem has to be replaced by $L^{-1}G$. More precisely, we have the following proposition.

PROPOSITION 5.2. *Let (A, B) be the given Hermitian matrix pair with $B \in \mathbb{C}^{n\times n}$ positive definite. Consider the indefinite decomposition*

$$A - \lambda_a B = CJC^H, \tag{5.14}$$

where λ_a is taken from any spectral gap $\mathcal{I}_a$ of the pair (A, B); J is as in (5.4) and the Cholesky decomposition $B = LL^H$. Let $V \in \mathbb{C}^{n\times p}$, $p \le n$, be B-orthonormal: $V^H BV = I_p$, spanning a spectral subspace of the pair $(A - \lambda_a B, B)$, that is,

$$(A - \lambda_a B)V = BV\Lambda_a, \tag{5.15}$$

where

$$\Lambda_a := \Lambda - \lambda_a I_p, \tag{5.16}$$

Λ is a diagonal matrix containing some of the eigenvalues of the pair (A, B). Set

$$F = C^{-1}LL^H V|\Lambda_a|^{1/2} \text{ or, equivalently, } V = L^{-H}L^{-1}CF|\Lambda_a|^{-1/2}.$$

Then

$$F^H JF = \mathcal{J}, \tag{5.17}$$

with $\mathcal{J} := \operatorname{sign}(\Lambda_a)$ and

$$(L^{-1}C)^H(L^{-1}C)F = JF\Lambda_a, \tag{5.18}$$

that is, F spans the corresponding spectral subspace of the pair $((L^{-1}C)^H(L^{-1}C), J)$. Conversely, (5.18) and (5.17) imply (5.15) and the fact that V is B-orthonormal.

Proof. Set $A_1 = L^{-1}AL^{-H}$, $U = L^H V$ and $G = L^{-1}C$. Premultiply (5.14) by L^{-1} and postmultiply by L^{-H}, use $G = L^{-1}C$ to obtain

$$A_1 - \lambda_a I = GJG^H.$$

Premultiply (5.15) by L^{-1} and use $U = L^H V$ to obtain

$$(A_1 - \lambda_a I)U = U\Lambda_a.$$

Now, $U^H U = V^H L^H LV = V^H BV = I_p$, set $F = G^{-1}U|\Lambda_a|^{1/2}$ and apply Theorem 5.1 with Λ_a from (5.16) and substituting $A \to A_1$. ∎

Moreover, $\mathcal{I} := \mathcal{I}_a - \lambda_a$ is the definiteness interval around zero of the positive definite pair $((L^{-1}C)^H(L^{-1}C), J)$ and Algorithm 3.1 with $\ell_+ = j_b$, $\ell_- = j_s$ applies. After running Algorithm 3.1, $\Lambda = \Lambda_a + \lambda_a I_p$ contains the wanted eigenvalues around λ_a and $V = L^{-H}L^{-1}CF|\Lambda_a|^{-1/2}$ contains the corresponding eigenvectors of the matrix pair (A, B) with B positive definite. Here we modify Algorithm 3.1 to work with factors L and C in a way that $t = (L^{-1}C)^H(L^{-1}C)x$ is implemented by

$$\begin{array}{ll} y = Cx & \text{matrix-vector multiplication,} \\ Lz = y & \text{solving a linear system,} \\ L^H v = z & \text{solving a linear system,} \\ C^H v = t & \text{matrix-vector multiplication;} \end{array}$$

and similarly for a preconditioned residual with the preconditioner $T = \left((L^{-1}C)^H(L^{-1}C)\right)^{-1}$.

REMARK 5.3. If we have a matrix pair (A, B) with B negative definite, then an analogous proposition holds; consider the auxiliary pair $(A - \lambda_a B, -B)$.

Finally, we give the proposition for a definite matrix pair (A, B) with indefinite B and λ_a from any spectral gap.

PROPOSITION 5.4. *Let (A, B) be the given positive definite matrix pair with $B \in \mathbb{C}^{n\times n}$ indefinite. Let λ_0 be an arbitrary definitizing shift and set $\widetilde{A} = A - \lambda_0 B$, which is positive definite. Consider the indefinite decomposition $B - (\lambda_a - \lambda_0)^{-1}\widetilde{A} = CJC^H$, where λ_a is taken from any spectral gap $\mathcal{I}_a$, of the pair (A, B); J is as in (5.4) and the Cholesky decomposition $\widetilde{A} = LL^H$. Let $V \in \mathbb{C}^{n\times p}$, $p \le n$, be $\widetilde{A}$-orthonormal: $V^H\widetilde{A}V = I_p$, spanning a spectral subspace of the pair $(B - (\lambda_a - \lambda_0)^{-1}\widetilde{A}, \widetilde{A})$, that is,*

$$(B - (\lambda_a - \lambda_0)^{-1}\widetilde{A})V = \widetilde{A}V\Lambda_{0a}, \tag{5.19}$$

where

$$\Lambda_{0a} := (\Lambda - \lambda_0 I_p)^{-1} - (\lambda_a - \lambda_0)^{-1} I_p,$$

$\Lambda \in \mathbb{R}^{p\times p}$ is a diagonal matrix containing some of the eigenvalues of the pair (A, B). Set

$$F = C^{-1}LL^HV|\Lambda_{0a}|^{1/2} \text{ or, equivalently, } V = L^{-H}L^{-1}CF|\Lambda_{0a}|^{-1/2}.$$

Then

$$F^H JF = \mathcal{J}, \tag{5.20}$$

with $\mathcal{J} := \operatorname{sign}(\Lambda_{0a})$ and

$$(L^{-1}C)^H(L^{-1}C)F = JF\Lambda_{0a}, \tag{5.21}$$

that is, F spans the corresponding spectral subspace of the pair $((L^{-1}C)^H(L^{-1}C), J)$. Conversely, (5.21) and (5.20) imply (5.19) and the fact that V is $\widetilde{A}$-orthonormal.

Proof. The proof follows immediately from Theorem 5.2 by substituting $A - \lambda_a B \to B - (\lambda_a - \lambda_0)^{-1}\widetilde{A}$, $B \to \widetilde{A}$ and $\Lambda_a \to \Lambda_{0a}$. ∎

Moreover, $\mathcal{I} := 1/(\mathcal{I}_a - \lambda_0) - 1/(\lambda_a - \lambda_0)$ is the definiteness interval around zero of the positive definite pair $((L^{-1}C)^H(L^{-1}C), J)$ and Algorithm 3.1 with $\ell_+ = j_s, \ell_- = j_b$ applies; notice the change of the indices $\pm$ compared to the previous cases. After running Algorithm 3.1, $\Lambda = (\Lambda_{0a} + (\lambda_a - \lambda_0)^{-1}I_p)^{-1} + \lambda_0 I_p$ contains the wanted eigenvalues around λ_a and $V = L^{-H}L^{-1}CF|\Lambda_{0a}|^{-1/2}$ contains the corresponding eigenvectors of the positive definite matrix pair (A, B) with indefinite B. Although $V^H\widetilde{A}V = I_p$, after

premultiplication (5.19) by V^H we get $V^HBV = \Lambda_{0a} + (\lambda_a - \lambda_0)^{-1}I_p$. Therefore, V is B-orthogonal and after normalization it becomes B-orthonormal.

REMARK 5.5. If a matrix pair (A, B) is negative definite with B indefinite, then an analogous proposition holds; consider the auxiliary pair $(B - (\lambda_a - \lambda_0)^{-1}\widetilde{A}, -\widetilde{A})$.

REMARK 5.6. When applying Algorithm 3.1 to the corresponding pair from (5.7) or (5.21), an initial guess $X^{(0)}$ needs to have at least k_+ J-positive columns and k_- J-negative columns. This is easily achieved by choosing appropriate columns of the identity matrix. However, when using inverses, the initial guess needs to have at least k_+ $\widetilde{B}^{-1}$-positive columns and k_- $\widetilde{B}^{-1}$-negative columns. For the corresponding pair (5.1), $\widetilde{B} = A - \lambda_a B$, and for (5.2), $\widetilde{B} = B - (\lambda_a - \lambda_0)^{-1}(A - \lambda_0 B)$. Assume $Y^HJY = J_k$ for some matrix $Y \in \mathbb{C}^{n\times k}$, where $J_k = \operatorname{diag}(\pm 1)$ and $k = k_+ + k_-$. Then for $X := CY$ we have $X^H\widetilde{B}^{-1}X = J_k$, where $\widetilde{B} = CJC^H$. Therefore, $CX^{(0)}$ can be used as an initial guess for our algorithm applied to the corresponding pair (5.1) or (5.2).

6. Numerical experiments. In this section, we consider some numerical experiments illustrating the performance of Algorithm 3.1 for different $k_\pm = \ell_\pm$ (a number of the wanted eigenpairs) and different m; the dimension of the search subspace is $(k_+ + k_-)m$. In some experiments we compare different preconditioners for the same initial guess, $k_\pm$, and m. In all experiments we use $\mathsf{tol} = 10^{-7}$ in the convergence criterion (3.6); unless otherwise stated. Sometimes our algorithm fails to converge within the allowed number of iterations; we mark this failure as ∞ in our results. We have two sets of our experiments. In the first set we apply Algorithm 3.1 to the original matrix pair to compute eigenpairs around the definiteness interval. In the second set we apply Algorithm 3.1 to the transformed matrix pair to compute eigenpairs around some arbitrary spectral gap. All experiments have been performed in MATLAB R2014a on Intel Core i3-4150 CPU 3.50GHz, 6 GB RAM, with the exception of Example 6.2, which has been performed in MATLAB R2014a on Intel i5 760 @ 2.80GHz, 8 GB RAM.

6.1. The definiteness interval. In this subsection, we consider one product eigenvalue problem and two hyperbolic quadratic eigenvalue problems. We apply Algorithm 3.1 to the corresponding definite matrix pair. A comparison is made between Algorithm 3.1 for several values of m and the corresponding algorithm that uses all previous iteration matrices $X^{(j)}$, that is, $P^{(j)}$ for all $j \le i$ denoted by "w.h." in the results; meaning the whole history.

EXAMPLE 6.1. Consider the product eigenvalue problem

$$MKx = \lambda^2 x, \quad KMy = \lambda^2 y, \quad 0 \neq x, y \in \mathbb{C}^n, \tag{6.1}$$

where $M, K \in \mathbb{C}^{n\times n}$ are Hermitian positive semidefinite and one of them is positive definite. This problem appears in computational quantum chemistry [5], where it is of interest to find a few smallest eigenvalues and the corresponding eigenvectors. Bai and Li developed the corresponding theory (Cauchy-like interlacing inequalities, the trace minimization principle) for this problem in [5]. The BPSD-like method for the product eigenvalue problem is proposed in [52] and LOBPCG-like methods in [6, 7, 8]. It is well known that (6.1) is equivalent to the GEP for the pair (A, B), where

$$A = \begin{bmatrix} K & 0 \\ 0 & M \end{bmatrix}, \qquad B = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}. \tag{6.2}$$

Therefore, λ^2 is an eigenvalue of MK if and only if $\pm\lambda$ are the eigenvalues of (A, B), the corresponding eigenvectors of (A, B) are given by

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

where x, y are from (6.1). If both K and M are positive definite, then the pair (A, B) from (6.2) is trivially positive definite, and finding the smallest eigenvalues of the product eigenvalue problem (6.1) is equivalent to finding the eigenvalues around the definiteness interval of the pair (A, B). The direct application of our Algorithm 3.1 to the pair (A, B) would imply working with matrices of double order. Therefore, a specialized LOBPCG-type algorithm has been proposed in [38, Algorithm 3]. This algorithm is mathematically equivalent to [6, Algorithm 4.1] and the experiment [38, Example 5.1] demonstrates that these two algorithms have a similar convergence behavior. Here we want to compare the behavior of the specialized Algorithm 3.1 for different values of m. This specialized algorithm reduces memory requirements, computational cost, and CPU time, and preserves the symmetry of the computed eigenpairs, compared with the direct application of Algorithm 3.1 to the product eigenvalue problem. For brevity, we will not write down this algorithm, but just point out that for $m = 3$ this specialized algorithm is [38, Algorithm 3]. As a concrete example from the linear response analysis of the density matrix in electronic structure calculations, we use the same matrices as in [8], [38, Example 5.1]. Therefore, the matrices K, M are real, symmetric positive definite of order 5660. We refer to [8, Section 5] for more information about these matrices and the natural preconditioner for this concrete example.

TABLE 6.1
Product eigenvalue problem from Example 6.1 with a zero shift using exact and CG preconditioners.

	m	2	3	4	5	w.h.
			$k_+ = 1$			
exact	# iter	107	41	35	35	35
CG	# iter	108	41	38	37	35
	# in.iter	725, 737	355, 353	327, 325	319, 319	302, 298
	av.# in.iter	6.78, 6.89	8.88, 8.23	8.84, 8.78	8.86, 8.86	8.88, 8.76
			$k_+ = 2$			
exact	# iter	97, 116	38, 47	34, 39	34, 34	33, 33
CG	# iter	98, 119	38, 48	35, 40	34, 34	32, 33
	# in.iter	1411, 1440	741, 728	642, 625	578, 561	550, 538
	av.# in.iter	6.56, 6.7	8.82, 8.67	8.79, 8.56	8.76, 8.50	8.73, 8.54
			$k_+ = 3$			
exact	# iter	96, 103, 107	37, 38, 47	34, 36, 40	33, 34, 36	30, 32, 33
CG	# iter	96, 103, 120	37, 39, 49	35, 36, 42	34, 35, 37	29, 31, 34
	# in.iter	2042, 1990	1063, 1039	972, 959	912, 904	794, 782
	av.# in.iter	6.46, 6.3	8.71, 8.52	8.84, 8.72	8.85, 8.78	8.73, 8.59
			$k_+ = 4$			
exact	# iter	95, 96, 104, 227	36, 37, 39, 70	33, 34, 35, 69	32, 33, 33, 67	29, 29, 30, 36
CG	# iter	98, 98, 102, 222	35, 38, 40, 67	34, 35, 35, 71	33, 33, 34, 65	29, 30, 30, 36
	# in.iter	3355, 3306	1549, 1506	1518, 1479	1424, 1406	1029, 1018
	av.# in.iter	6.50, 6.41	8.80, 8.56	8.88, 8.65	8.84, 8.73	8.50, 8.41

The four smallest positive eigenvalues are given by

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

A comparison is made between the specialized Algorithm 3.1 with $m = 2, 3, 4, 5$, and the corresponding algorithm in which the whole history is included. Results are reported in Table 6.1. The rows in this table correspond to one value of k_+ for $k_+ = 1, 2, 3, 4$, and the columns correspond to one value of m. Here we use zero as a definitizing shift and two different ways for computing preconditioned residuals, i.e., the MATLAB `backslash` operator, and

the linear conjugate gradient method with stopping tolerance 10^{-2} and a maximum of 20 iterations; the rows in the table starting with "exact" and "CG", respectively. Since we use a zero shift, the preconditioner T equals A^{-1} for A from (6.2), so we apply the CG method twice: once for K and once for M. The total number of CG iterations is denoted by "# in.iter" in Table 6.1; the first number corresponds to the total number for K and the second for M. We also put the average number of CG iterations per iteration of the specialized Algorithm 3.1 denoted by "av.# in.iter", first for K and then for M. The total number of required iterations of the specialized Algorithm 3.1 until all k_+ desired eigenvalues have converged is denoted by "# iter". For example, we explain the field by $m = 2$ and $k_+ = 3$ from Table 6.1: All three wanted eigenvalues converge after 107 iterations of Algorithm 3.1 when using the exact preconditioner, and after 120 iterations when using CG-based preconditioners. The first smallest positive eigenvalue converges after 96 iterations, and the second smallest positive converges after 103 iterations. The total number of inner CG iterations for K and M is 2042 and 1990, respectively. The average number of inner CG iterations per iteration of the specialized Algorithm 3.1 for K and M is 6.46 and 6.3, respectively.

The findings of Table 6.1 are as follows. The total number of required iterations of the specialized Algorithm 3.1, and also the total number of inner CG iterations, is reduced by increasing the dimension of the search subspace. The most important difference is between $m = 2$ and $m = 3$; specialized indefinite BPSD/A and specialized indefinite LOBPCG, respectively. The total number of required iterations of the specialized variant of Algorithm 3.1 for $m = 3, 4, 5$, and when using the whole history is very similar, the largest difference is for the fourth Ritz value. The first three positive eigenvalues are clustered, therefore the convergence for the first three Ritz pairs is much faster than for the fourth one. Moreover, by enlarging the block size k_+ we can get a faster convergence to the smallest eigenpair, but with more numerical cost per iteration of the specialized Algorithm 3.1. It is very important to note that the results for CG-based preconditioners with very crude approximations of the preconditioned residuals are very similar, in some cases the same, to the results for exact preconditioned residuals. The maximum number of inner CG iterations per iteration of the specialized Algorithm 3.1 is 10 for all values of k_+ and m including the case in which the whole history is included. We have repeated the experiment with CG-based preconditioners with stopping tolerance 10^{-1} and 10^{-3}; the total number of required iterations of the specialized Algorithm 3.1 is quite similar to the presented case.

TABLE 6.2
Product eigenvalue problem from Example 6.1 with a nonzero shift using CG preconditioners.

	$k_+ = 1$		$k_+ = 2$		$k_+ = 3$	
m	2	3, 4, 5, w.h.	2	3, 4, 5, w.h.	2	3, 4, 5, w.h.
# iter	6	6	6, 6	6, 6	6, 6, 6	6, 6, 6
# in.iter	158	161	310	311	471	476
av.# in.iter	31.6	32.2	31.0	31.1	31.4	31.73
	$k_+ = 4$					
m	2	3	4	5		w.h.
# iter	6, 6, 6, 46	6, 6, 6, 18	6, 6, 6, 18	6, 6, 6, 18		6, 6, 6, 17
# in.iter	1, 251	908	908	911		881
av.# in.iter	20.85	28.38	28.38	28.47		28.42

Now we choose a relatively close definitizing shift $\lambda_0 = 0.54$ so we make use of the preconditioners $T_\pm = (A \mp \lambda_0 B)^{-1}$. We apply the linear CG method with stopping tolerance 10^{-2} and a maximum of 50 iterations. Table 6.2 gives the total number of required iterations of the specialized Algorithm 3.1 until all $k_+ = 1, 2, 3, 4$ desired eigenpairs have converged,

including the total and the average number of CG iterations for $m = 2, 3, 4, 5$, and when using the whole history. For example, the total number of inner CG iterations for $k_+ = 1$ is 158 for $m = 2$ and in all other cases it is 161. The average number of inner CG iterations per iteration of the specialized Algorithm 3.1 for $k_+ = 1$ is 31.6 for $m = 2$ and it is 32.2 in all other cases. The maximum number of inner CG iterations per iteration of the specialized Algorithm 3.1 is reached in some cases. When an excellent shift is used, there is no significant difference in the convergence of the fourth Ritz value by enlarging the dimension of the search subspace.

EXAMPLE 6.2. Consider a quadratic eigenvalue problem (QEP)

$$(\lambda^2 M + \lambda D + K)x = 0, \quad 0 \neq x \in \mathbb{C}^n, \tag{6.3}$$

where M, D, $K \in \mathbb{C}^{n\times n}$ are Hermitian and M is positive definite. A Hermitian linearization of (6.3) yields the Hermitian pair (A, B) with

$$A = \begin{bmatrix} M & 0 \\ 0 & -K \end{bmatrix}, \qquad B = \begin{bmatrix} 0 & M \\ M & D \end{bmatrix}. \tag{6.4}$$

The positive definiteness of this matrix pair is equivalent to the hyperbolicity of the original QEP [27, 56].

Here we consider a simple scalable example as in [38, Example 5.2]:

$$K = (n+1)^2 \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix}, \quad M = I_n, \quad D = 2K. \tag{6.5}$$

This QEP is hyperbolic; its eigenvalues are given by

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

for $j = 1, \ldots, n$. As n increases, the definiteness interval of (A, B) converges to around $(-19.2258, -0.5134)$. Considering that $\|D\|_2 = 2\|K\|_2 = O(n^2)$, while $\|M\|_2 = 1$, we propose to rescale the pair (A, B) as follows:

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

We aim at computing the eigenvalues $\lambda_j^{\pm}$ for $j = 1, 2, 3$, that is, $k_{\pm} = 3$ in Algorithm 3.1 and the variant of Algorithm 3.1 with two preconditioners; see the end of Section 3. A comparison is made between Algorithm 3.1 with $m = 2, 3, 4, 5$, and the corresponding algorithm in which the whole history is included. The number of total iterations for B-positive and B-negative eigenpairs for $n = 2000$, 4000, 6000, 8000 is reported in Table 6.3. Notice that the order of the linearized pair (A, B) is doubled, so $n = 8000$ means that we work with matrices A, B of order $16,000$. The maximum number of allowed iterations of Algorithm 3.1 is 300 for one shift and 100 for two shifts. The CPU time is given in brackets in Table 6.3. An algebraic multigrid (AMG) V-cycle preconditioner can be used as *a black box* for solving linear systems for the preconditioned residuals in this example since we consider the scalable pair (A, B). Therefore, we use the implementation `HSL_MI20` [2] with the default settings. Since D in (6.5) is positive definite, for fixed n and B given in (6.4), the initial B-positive

TABLE 6.3
QEP from Example 6.2 with AMG preconditioners.

m	2	3	4	5	w.h.
n	one shift, without deflation				
2000	∞ (7.36)	221 (7.18)	153 (7.79)	172 (11.25)	214 (6020)
4000	∞ (12.73)	∞ (18.55)	245 (25.65)	192 (23.23)	16 (4.06)
6000	∞ (19.09)	240 (21.32)	206 (29.81)	132 (22.17)	17 (7.91)
8000	∞ (24.78)	234 (26.77)	196 (36.72)	63 (12.89)	12 (3.57)
n	two shifts, without deflation				
2000	81 (1.69)	21 (0.55)	21 (0.94)	19 (1.07)	14 (1.86)
4000	68 (2.70)	22 (1.15)	23 (1.60)	23 (2.35)	24 (14.1)
6000	53 (3.16)	26 (2.29)	20 (2.28)	18 (2.20)	14 (4.83)
8000	37 (3.12)	18 (2.09)	20 (2.65)	21 (3.64)	61 (335)
n	two shifts, with deflation				
2000	∞ (1.69)	∞ (1.41)	24 (0.58)	17 (0.63)	18 (1.63)
4000	∞ (2.83)	19 (0.70)	∞ (3.25)	22 (1.11)	16 (2.21)
6000	∞ (3.43)	25 (1.27)	27 (1.53)	21 (1.55)	15 (2.77)
8000	∞ (4.63)	20 (1.44)	20 (1.87)	21 (2.10)	17 (6.37)

and B-negative vectors are chosen as corresponding columns from $[0;\ I]$ and $[M^{-1}D;\ -I]$, respectively. We use the same initial guess for fixed n and fixed shifts.

The findings of Table 6.3 are as follows. Enlarging the dimension of the search subspace significantly reduces, in almost all cases, the number of total iterations of Algorithm 3.1 when the shift is not good (in this case, $\lambda_0 = -5$), but when using two excellent shifts (here, $\lambda_0^+ = -0.514$ and $\lambda_0^- = -19.22$), there is no such significant reduction in the number of total iterations in the indefinite (m)-scheme when m is increased. Although in almost all cases the use of all previous iteration matrices gives the smallest number of total iterations, there is an increase in the numerical cost and memory requirements per iteration. When the dimension of the search subspace is fixed, like in Algorithm 3.1, there is a fixed numerical cost and memory requirements per iteration. When two excellent shifts are used, we see that enlarging the dimension of the search subspace increases CPU time. Again, the indefinite (m)-scheme with $m = 3$, i.e., indefinite LOBPCG, is more efficient than the indefinite (m)-scheme with $m = 2$, i.e., indefinite BPSD/A. We notice that the number of total iterations of Algorithm 3.1 with two shifts for fixed m is very similar for different values of n.

EXAMPLE 6.3. Consider another hyperbolic quadratic eigenvalue problem with matrices

$$K = \begin{bmatrix} 15 & -5 & & \\ -5 & \ddots & \ddots & \\ & \ddots & \ddots & -5 \\ & & -5 & 15 \end{bmatrix}, \quad M = I_n, \quad D = 2K. \tag{6.6}$$

These matrices can be produced by the command `nlevp('spring',n,1,10,5,10,5)` from a collection of NLEVP [10] and the eigenvalues are given by

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

for $j = 1, \ldots, n$. As n increases, the definiteness interval of the linearized pair (A, B) from (6.4) converges to around $(-9.4721, -0.52786)$. Moreover, the gaps between eigenvalues become arbitrarily small as $n \to \infty$. The illustration for $n = 1000$ is given in Fig. 6.1.

The inefficiency of Algorithm 3.1 with $m = 3$ (since it uses only one preconditioner) is illustrated in [38, Example 5.3] caused by a decrease in the eigenvalue gaps; illustration

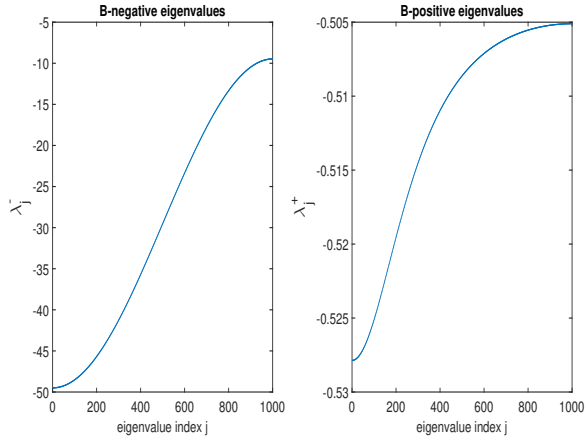

FIG. 6.1. *Eigenvalues of Example 6.3 for* $n = 1000$. *The definiteness interval of the pair* (A, B), *when* n *increases, is around* $(-9.4721, -0.52786)$. 1000 B*-negative eigenvalues are in the interval around* $(-49.4948, -9.4721)$ *and* 1000 B*-positive eigenvalues are in the interval around* $(-0.52786, -0.5051)$.

was made using the exact preconditioner with shift $\lambda_0 = -5$, nearly in the middle of the definiteness interval. However, exact tailored preconditioners $T_\pm$ perform very well. Here we want to compare Algorithm 3.1 with two shifts and with $m = 2, 3, 4, 5, 10$, and when using the whole history. Here we use $\lambda_0^- = -9.47$ and $\lambda_0^+ = -0.528$ as definitizing shifts and two different ways for computing preconditioned residuals: the MATLAB `backslash` operator and the linear CG method with stopping tolerance 10^{-2} and a maximum of 50 iterations. The obtained results for exact preconditioners and CG-based preconditioners are reported in Table 6.4 and Table 6.5, respectively. We aim at computing the eigenvalues $\lambda_j^\pm$ for $j = 1, 2, 3,$

TABLE 6.4
QEP from Example 6.3 with two shifts and exact inverse preconditioners.

m	2	3	4	5	10	w.h.
			$n = 1000$			
# iter B-pos	227	37	30	28	23	20
# iter B-neg	19	10	10	9	9	9
CPU time	(1.74)	(0.35)	(0.36)	(0.39)	(0.48)	(0.84)
			$n = 2000$			
# iter B-pos	720	73	62	61	49	36
# iter B-neg	54	17	16	15	14	14
CPU time	(11.9)	(1.30)	(1.63)	(1.76)	(2.62)	(7.35)

that is, $k_\pm = 3$ in the variant of Algorithm 3.1 with two preconditioners. We list the number of required iterations for B-positive and B-negative eigenpairs to converge separately, denoted by "# iter B-pos" and "# iter B-neg", respectively, in Tables 6.4 and 6.5 for $n = 1000$ and $n = 2000$; note that the order of the pair (A, B) is doubled. The CPU time is also given in both tables. The total number of inner CG-iterations is given in Table 6.5, separately for preconditioned residuals corresponding to B-positive and B-negative Ritz vectors, denoted by "# in.iter B-pos" and "# in.iter B-neg", respectively.

The findings of Tables 6.4 and 6.5 are as follows. The total number of required iterations of Algorithm 3.1 with two shifts, and also the total number of inner CG iterations, are reduced by increasing the dimension of the search subspace. The most important difference is again

TABLE 6.5
QEP from Example 6.3 with two shifts and CG-based preconditioners.

m	2	3	4	5	10	w.h.
	$n = 1000$					
# iter B-pos	239	51	47	42	37	35
# iter B-neg	288	79	79	79	80	77
# in.iter B-pos	23,942	6950	6400	6000	5300	5100
# in.iter B-neg	34,350	11,550	11,500	11,500	11,600	11,350
CPU time	(8.06)	(1.22)	(1.54)	(1.70)	(2.55)	(7.38)
	$n = 2000$					
# iter B-pos	716	89	86	77	72	67
# iter B-neg	941	151	151	151	151	113
# in.iter B-pos	71,696	12,750	11,800	11,200	10,500	9900
# in.iter B-neg	90,050	24,450	22,450	22,450	22,400	16,800
CPU time	(26.56)	(5.93)	(8.39)	(8.70)	(14.38)	(271.9)

between $m = 2$ and $m = 3$. Significant savings in CPU time are achieved by switching from $m = 2$ to $m = 3$. Algorithm 3.1 needs more CPU time by a further increase in the dimension of the search subspace, especially when using the whole history. The effectiveness of the tailored preconditioners $(A - \lambda_0^{\pm} B)^{-1}$ deteriorates as n increases, due to the decrease in the eigenvalue gaps. In this example, in contrast to Example 6.1, the exact preconditioners (note that matrices A, B are sparse) have outperformed the CG-based preconditioners.

Finally, we demonstrate that using a shift λ_0^+ that is very close to the eigenvalue λ_1^+ accelerates the convergence of Algorithm 3.1, as observed in Section 4.1. We list the required number of iterations for all six eigenpairs separately in Table 6.6.

TABLE 6.6
QEP from Example 6.3 with two exact inverse preconditioners with shifts $\lambda_0^- = -9.47$ *and* $\lambda_0^+ = -0.528$, $(\lambda_0^+ = -0.5279)$ *for* $n = 1000$.

m	λ_3^-	λ_2^-	λ_1^-	λ_1^+	λ_2^+	λ_3^+	CPU time
2	19	13	11	133 (41)	157 (49)	227 (71)	1.74 (0.66)
3	10	9	8	35 (19)	36 (21)	37 (24)	0.35 (0.27)
10	9	8	8	22 (14)	22 (15)	23 (15)	0.48 (0.37)

6.2. The arbitrary gaps. In this subsection, we consider ordinary eigenvalue problems with indefinite and positive definite matrices, as well as a definite generalized eigenvalue problem with both matrices indefinite. A comparison is made between Algorithm 3.1 applied to the corresponding pair (5.1) or (5.2), and to the corresponding pair from (5.7) or (5.21). In our results, the former is denoted by "using (an) inverse(s)" and the latter by "using (a) decomposition(s)" . An indefinite decomposition of a particular matrix is obtained by [53, pp. 1–2]. A maximum number of allowed iterations is 100 in all experiments. All our results are reported in tables, where for a fixed $k_{\pm}$:

(i) the first column corresponds to the total number of iterations of Algorithm 3.1,
(ii) the second column corresponds to the 2-norm of the absolute error $AV - BV\Lambda$, where Λ is a diagonal matrix of approximations of the wanted eigenvalues, and V is a matrix of approximations of the corresponding eigenvectors,
(iii) the third column corresponds to CPU time; the execution time of Algorithm 3.1 applied to the transformed matrix pair.

Although the initial guesses when using inverses and decompositions are connected (see Remark 5.6), due to roundoff, we can expect a different total number of iterations of our algorithm when using inverses and decompositions. Notice that $T = I$, when applying our

algorithm to the pair $(I, (A-\lambda_a I)^{-1})$, while $T = (G^H G)^{-1}$, when applying our algorithm to the pair $(G^H G, J)$.

EXAMPLE 6.4. We first consider two sparse ordinary eigenvalue problems with real symmetric indefinite matrices coming from Platzman's oceanographic models. These problems are part of the Harwell-Boeing collection [19]. We obtain them from the Matrix Market [1]. All eigenvalues occur in pairs, with the exception of an isolated singleton at zero. The eigenvalues of interest are interior eigenvalues, located in the interval $(0.0001, 0.024)$. Numerical experiments with these matrices can be found, for example, in [23, Section 5.1] and [50, Example 4.4.2].

TABLE 6.7
The Atlantic Ocean model from Example 6.4 with the convergence criterion (3.6) and $\mathsf{tol} = 10^{-7}$.

m	$k_\pm = 2$			$k_\pm = 6$			$k_+ = 0, k_- = 6$		
	using an inverse								
3	25	10^{-13}	(0.30)	28	10^{-12}	(0.78)	34	10^{-12}	(0.61)
5	15	10^{-13}	(0.22)	21	10^{-12}	(0.96)	31	10^{-12}	(0.76)
10	15	10^{-12}	(0.30)	21	10^{-11}	(2.88)	21	10^{-12}	(1.15)
	using a decomposition								
3	15	10^{-10}	(0.14)	22	10^{-9}	(0.40)	58	10^{-9}	(0.62)
5	13	10^{-10}	(0.15)	∞	10^{-1}	(3.30)	∞	10^{-2}	(2.08)
10	11	10^{-9}	(0.17)	∞	10^{-2}	(7.10)	∞	10^{-2}	(3.50)

TABLE 6.8
The Atlantic Ocean model from Example 6.4 with $m = 10$ *and the convergence criterion (6.7).*

tol	$k_\pm = 6$			$k_+ = 0, k_- = 6$		
	using an inverse					
10^{-7}	14	10^{-10}	(1.51)	16	10^{-10}	(0.32)
10^{-8}	16	10^{-11}	(1.62)	19	10^{-11}	(0.40)
	using a decomposition					
10^{-7}	8	10^{-4}	(0.22)	10	10^{-5}	(0.14)
10^{-8}	11	10^{-5}	(0.34)	11	10^{-6}	(0.16)

First, we consider a finite-difference model for the shallow wave equations for the Atlantic Ocean. The corresponding matrix A is of order 362 with $\lambda_{\min} \approx -3.55 \cdot 10^{-12}$ and $\lambda_{\max} \approx 0.77$. Here we want to compute the eigenpairs around the shift $\lambda_a = 0.024$. Therefore, we apply Algorithm 3.1 to the pair $(I, (A-\lambda_a I)^{-1})$ and to the pair $(G^H G, J)$, where $A - \lambda_a I = GJG^H$. In our implementation we have $\|(A-\lambda_a I) - GJG^H\|_2 \approx 10^{-15}$. The results are reported in Table 6.7. For example, Algorithm 3.1 with $m = 10$ applied to the matrix pair $(I, (A-\lambda_a I)^{-1})$ needs 21 iterations to compute approximations for the first 6 eigenpairs on both sides of λ_a. The corresponding absolute error is of order 10^{-11} and Algorithm 3.1 runs 2.88 seconds. However, Algorithm 3.1 applied to the decomposition of $A - \lambda_a I$ does not converge within the allowed number of iterations. A possible way to avoid such a non-convergence is to use another convergence criterion (see the discussion in [18, Section 4]), such as

$$\|r_j^{\pm}\|_2 \le \mathsf{tol} \cdot \left(\|A\|_2 + |\theta_j^{\pm}|\,\|B\|_2\right)\|x_j^{\pm}\|_2. \tag{6.7}$$

For large sparse matrices estimates of the 2-norms are used; in this example $B = I$. The results with $m = 10$ and the new convergence criterion are reported in Table 6.8. The convergence is now achieved in all cases when using a decomposition. When applying Algorithm 3.1

to the pair $(I, (A-\lambda_a I)^{-1})$, the algorithm takes fewer iterations by using the convergence criterion (6.7) than by using (3.6), but with larger absolute error. By decreasing the tolerance tol (for example, from 10^{-7} to 10^{-8}) we can achieve a smaller absolute error.

TABLE 6.9
The Atlantic and Indian Ocean model from Example 6.4. For every fixed m, the first row refers to the convergence criterion (3.6) and the second row to the convergence criterion (6.7).

m	$k_\pm = 2$			$k_\pm = 6$			$k\pm = 20$		
	using an inverse								
3	33	10^{-12}	(0.91)	41	10^{-12}	(2.23)	43	10^{-11}	(10.89)
	20	10^{-10}	(0.55)	25	10^{-9}	(1.31)	36	10^{-9}	(9.06)
5	24	10^{-13}	(0.79)	25	10^{-11}	(2.58)	41	10^{-11}	(31.16)
	18	10^{-10}	(0.49)	18	10^{-9}	(1.53)	31	10^{-9}	(23.51)
10	21	10^{-13}	(1.06)	21	10^{-13}	(4.88)	38	10^{-12}	(133.7)
	14	10^{-10}	(0.49)	14	10^{-10}	(2.31)	27	10^{-9}	(88.49)
	using a decomposition								
3	21	10^{-9}	(1.36)	∞	10^{-2}	(8.70)	∞	10^{-1}	(30.36)
	9	10^{-3}	(0.59)	12	10^{-4}	(0.98)	25	10^{-4}	(7.15)
5	∞	10^{-2}	(7.53)	∞	10^{-2}	(13.91)	∞	10^{-1}	(60.41)
	8	10^{-3}	(0.47)	9	10^{-4}	(1.01)	∞	10^{-1}	(60.21)
10	21	10^{-9}	(1.44)	∞	10^{-2}	(26.60)	∞	10^{-2}	(184.7)
	8	10^{-3}	(0.47)	8	10^{-4}	(0.95)	11	10^{-4}	(10.04)

Next, we consider a finite-difference model for the shallow wave equations for the Atlantic and Indian Ocean. The corresponding matrix A is of order 1919 with $\lambda_{\min} \approx -3.40 \cdot 10^{-16}$ and $\lambda_{\max} \approx 2.92$. Here we want to compute the eigenpairs around the shift 0.0121 in the middle of the interval of interest. In our implementation we have $\|(A - 0.0121I) - GJG^H\|_2 \approx 10^{-14}$. The results are reported in Table 6.9. For $m = 10$ and $k\pm = 2$ by using the convergence criterion (3.6), Algorithm 3.1 applied to the inverse and Algorithm 3.1 applied to the decomposition terminate after 21 iterations, but the former is faster and more accurate than the latter. In all cases when our algorithm terminates within the allowed number of iterations, it takes fewer iterations by using the convergence criterion (6.7) than by using (3.6), but with larger absolute error. Algorithm 3.1 applied to the decomposition is quite sensitive to a convergence criterion. In many cases, it does not converge within the allowed number of iterations when using (3.6).

By observing the results from Tables 6.7–6.9, we can see that Algorithm 3.1 applied to the inverse is more accurate than Algorithm 3.1 applied to the decomposition regardless of a convergence criterion.

EXAMPLE 6.5. In this example, we consider a sparse ordinary eigenvalue problem with a real symmetric positive definite matrix. More precisely, we consider a five-point finite difference discretization of the Laplace operator on a 115×115 uniform mesh of the unit square without the circle with radius 0.5 centered in the left vertex. The corresponding matrix A is of order $10,279$. We want to detect first $k_\pm = 1, 5, 10$ eigenvalues around the shift $\lambda_a = 7$. Therefore, we apply Algorithm 3.1 to the pair $(I, (A-\lambda_a I)^{-1})$ and to the pair $(G^H G, J)$, where $A - \lambda_a I = GJG^H$. In our implementation we have $\|(A - \lambda_a I) - GJG^H\|_2 \approx 10^{-14}$. The results are reported in Table 6.10.

As in the previous example, in all cases when our algorithm terminates within the allowed number of iterations, it takes fewer iterations by using the convergence criterion (6.7) than by using (3.6), but with larger absolute error. When using the decomposition the absolute error is satisfactory with (3.6) (in the cases when the algorithm terminated within the allowed number of iterations), but is very unsatisfactory with (6.7). In the latter case the algorithm terminates

TABLE 6.10

The Laplacian eigenvalue problem on the 115×115 *uniform mesh from Example 6.5. For every fixed* m*, the first row refers to the convergence criterion (3.6) and the second row to the convergence criterion (6.7).*

m	$k_\pm = 1$			$k_\pm = 5$			$k\pm = 10$		
	using an inverse								
3	10	10^{-14}	(0.77)	27	10^{-12}	(7.99)	41	10^{-10}	(24.68)
	8	10^{-11}	(0.51)	19	10^{-9}	(4.51)	33	10^{-9}	(19.26)
5	10	10^{-15}	(0.82)	19	10^{-12}	(9.10)	41	10^{-10}	(118.7)
	8	10^{-11}	(0.63)	15	10^{-10}	(5.55)	22	10^{-9}	(47.72)
10	10	10^{-14}	(0.99)	20	10^{-12}	(21.24)	21	10^{-11}	(100.0)
	8	10^{-11}	(0.53)	12	10^{-9}	(10.35)	15	10^{-10}	(49.37)
	using a decomposition								
3	9	10^{-11}	(1.70)	52	10^{-9}	(17.81)	∞	10^{-1}	(64.61)
	3	10^{-2}	(0.62)	6	10^{-2}	(1.92)	7	10^{-2}	(3.99)
5	9	10^{-11}	(1.89)	∞	10^{-1}	(55.28)	∞	10^{-1}	(114.5)
	3	10^{-2}	(0.67)	41	10^{-2}	(20.41)	7	10^{-3}	(5.28)
10	9	10^{-11}	(1.72)	∞	10^{-1}	(96.29)	∞	10^{-1}	(265.2)
	3	10^{-2}	(0.56)	6	10^{-2}	(2.30)	7	10^{-3}	(5.93)

too quickly with very poor approximations of the wanted eigenpais.

EXAMPLE 6.6. In the last example, we consider a quadratic eigenvalue problem (6.3) with matrices M, D, K from the spring example (6.6). The positive definiteness of M, D, K implies both A, B from (6.4) to be indefinite. Therefore, in detecting an arbitrary spectral gap we need to use the pair (5.2) or Proposition 5.4. First, we apply "perfect shuffling" [57, p. 104] to the matrix pair (A, B), that is, we use auxiliary matrices $A_1 = P^H A P$, $B_1 = P^H B P$, where P is a permutation matrix given by $Pe_{2j-1} = e_j$, $Pe_{2j} = e_{j+n}$, $j = 1, \ldots, n$ and e_j is the jth column of the identity matrix I_{2n}. This is a trick that transforms both matrices A, B to five-diagonal matrices A_1, B_1:

$$A_1 = \begin{bmatrix} 1 & 0 & 0 & & & & & & \\ & -15 & 0 & 5 & & & & & \\ & & 1 & 0 & 0 & & & & \\ & & & -15 & 0 & 5 & & & \\ & & & & \ddots & \ddots & \ddots & & \\ & & & \text{sym} & & & & 5 & \\ & & & & & & & 0 & \\ & & & & & & & -15 & \end{bmatrix}, B_1 = \begin{bmatrix} 0 & 1 & 0 & & & & & & \\ & 30 & 0 & -10 & & & & & \\ & & 0 & 1 & 0 & & & & \\ & & & 30 & 0 & -10 & & & \\ & & & & \ddots & \ddots & \ddots & & \\ & & & \text{sym} & & & & -10 & \\ & & & & & & & 1 & \\ & & & & & & & 30 & \end{bmatrix}.$$

The matrix pair (A, B) is congruent to (A_1, B_1): the eigenvalues are the same, and the eigenvectors are multiplied by P. Let λ_0 be an arbitrary definitizing shift and λ_a the given shift from any spectral gap of the positive definite matrix pair (A, B) from (6.4). Set $\widehat{A}_1 = A_1 - \lambda_0 B_1 \succ 0$.

TABLE 6.11

QEP from Example 6.6 using inverses. For every fixed m*, the first row refers to the convergence criterion (3.6) and the second row to the convergence criterion (6.7).*

m	$k_\pm = 1$			$k_\pm = 5$			$k_\pm = 10$		
3	15	10^{-11}	(0.37)	24	10^{-10}	(0.89)	28	10^{-10}	(1.72)
	11	10^{-9}	(0.36)	19	10^{-8}	(0.63)	22	10^{-8}	(1.30)
5	14	10^{-12}	(0.41)	55	10^{-10}	(4.04)	25	10^{-9}	(4.94)
	10	10^{-8}	(0.31)	21	10^{-8}	(1.33)	21	10^{-8}	(3.39)
10	14	10^{-12}	(0.49)	31	10^{-10}	(7.34)	30	10^{-11}	(18.2)
	12	10^{-9}	(0.39)	19	10^{-8}	(3.91)	17	10^{-8}	(8.70)

Consider the indefinite decomposition $\widetilde{B}_1 = B_1 - (\lambda_a - \lambda_0)^{-1}\widetilde{A}_1 = CJC^H$ obtained by [53], but with a pivoting strategy given in [13, Algorithm E]. That pivoting strategy is designed for five-diagonal real symmetric or Hermitian matrices and preserves the five-diagonal structure during the process. In our implementation we have $\|\widetilde{B}_1 - CJC^H\|_2 \approx 10^{-14}$. Let L be the Cholesky factor of $\widetilde{A}_1$; cf. Theorem 5.4. Set $n = 1000$ in (6.6) and $\lambda_0 = -5$. We want to detect first $k_\pm = 1, 5, 10$ eigenvalues around the shift $\lambda_a = -30$; in the middle of B-negative eigenvalues: see Figure 6.1. Therefore, we apply Algorithm 3.1 to the pair $(\widetilde{A}_1^{-1}, \widetilde{B}_1^{-1})$ and to the pair (C^HC, J). The results are reported in Table 6.11. We omit the results when using decompositions since, in almost all cases, either the algorithm does not converge within the allowed number of iterations or the absolute error is $O(10^{-3})$ or larger.

7. Conclusions. Numerical experiments [35] for a generalized eigenvalue problem with both symmetric positive definite matrices demonstrate that a (k)-scheme of preconditioned gradient iterations [43, 44] for $k \geq 4$ is of minor importance and that the locally optimal block preconditioned conjugate gradient (LOBPCG) iteration is an optimal eigensolver within that (k)-scheme. In our experiments, we noticed that indefinite LOBPCG is much faster than indefinite block preconditioned steepest descent/ascent (BPSD/A). A further increase in the dimension of the search subspaces often leads to a decrease in the total number of iterations of our Algorithm 3.1; it is especially significant when a preconditioner is poor (when a definitizing shift is not close to the boundaries of the definiteness interval), although it is more memory- and time-consuming. When excellent preconditioners are used, we can conclude that indefinite LOBPCG is an optimal eigensolver within our indefinite (m)-scheme, that is, Algorithm 3.1. In some cases, very crude approximations of the preconditioned residuals can be used.

There is an important difference between LOBPCG type algorithms designed for a hyperbolic quadratic eigenvalue problem [9, 41, 55] and our indefinite LOBPCG type algorithm applied to the matrix pair from the linearization: the former can use any initial guess, the latter needs to contain at least as many B-positive and B-negative vectors as we want to compute. Many new indefinite eigensolvers can be proposed using some appropriate search subspace and applying the Rayleigh–Ritz projection method that extracts interior eigenpairs. The generic indefinite algorithm (cf. [50, Algorithm 2.1]) is given in Algorithm 7.1. If the column space of every iteration matrix of the Ritz vectors is included in the subspace $\mathcal{U}^{(i)}$, then choosing a valid initial guess will provide enough B-positive and B-negative vectors in any basis matrix $U^{(i)}$ of $\mathcal{U}^{(i)}$ for all $i = 1, 2, \ldots$. Some choices of the subspace $\mathcal{U}^{(i)}$ of Algorithm 7.1 can, with a good separation of the desired eigenvalues, lead to the fast convergence without preconditioning. However, if preconditioning is used, then we propose to add the column space of the preconditioned residual of the current iteration matrix to the subspace $\mathcal{U}^{(i)}$. The crucial thing is to know at least one definitizing shift; it is trivial if A is positive definite. For finding such a shift, we can use the algorithm given in [24]. For some examples in applications, knowing only one definitizing shift is not enough, two are necessary. Not just any two definitizing shifts, but those shifts that are close to the boundaries of the definiteness interval. Finding an algorithm that can give such two definitizing shifts is a matter of the future work.

Finally, we derived some ideas how to use an indefinite eigensolver (for example, Algorithm 3.1) to compute a few eigenvalues around *any spectral gap* and the corresponding eigenvectors of a definite matrix pair. The proposed spectral transformations can be applied to those definite matrix pairs for which solving the corresponding linear systems is easy or performing decompositions is not expensive and can be done very accurately. Our experiments demonstrate that the application of Algorithm 3.1 to a transformed pair with inverses is more efficient than the application to a transformed pair with decompositions. Enlarging the

Algorithm 7.1 Generic indefinite eigensolver

Input: $A, B \in \mathbb{C}^{n\times n}$: coefficients of positive definite pair (A, B) with indefinite B;
Output: k_+ smallest B-positive eigenpairs and k_- largest B-negative eigenpairs.

1: **for** $i = 1, 2, \ldots$ **do**
2: Construct a subspace $\mathcal{U}^{(i)}$ of dimension $m \ll n$ such that $(k_+, k_-, 0) \leq \mathrm{In}\left[(U^{(i)})^H B U^{(i)}\right]$ holds for any basis matrix $U^{(i)}$.
3: Apply the Rayleigh–Ritz procedure to (A, B) with respect to the subspace $\mathcal{U}^{(i)}$ and extract k_+ smallest B-positive and k_- largest B-negative Ritz pairs.
4: **end for**

dimension of the search subspaces can reduce the total number of iterations of Algorithm 3.1, but often increases CPU time. Therefore, we can recommend to use $m = 3$ in our indefinite m-scheme, that is, to use the indefinite LOBPCG iteration.

Acknowledgments. Some basic ideas of this paper emerged during the author study visit to the ANCHP, MATHICSE, EPF Lausanne, Switzerland, under the direction of Professor Daniel Kressner to whom the author is indebted for his illuminating discussions and comments, as well as for kind hospitality. The author also thanks Dr. Meiyue Shao for all the patience and support during the formation of paper [38], the predecessor of this paper. The author thanks Professors Krešimir Veselić and Ninoslav Truhar for valuable discussions and Professor Klaus Neymeyr for clarifying some details in the proof of [45, Theorem 2.2].

REFERENCES

[1] *MATRIX MARKET*. http://math.nist.gov/MatrixMarket/.
[2] *HSL. A collection of Fortran codes for large scale scientific computation*, 2011. http://www.hsl.rl.ac.uk.
[3] C. ASHCRAFT, R. G. GRIMES, AND J. G. LEWIS, *Accurate symmetric indefinite linear equation solvers*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 513–561.
[4] Z. BAI, J. W. DEMMEL, J. J. DONGARRA, A. RUHE, AND H. VAN DER VORST, eds., *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, SIAM, Philadelphia, 2000. http://web.eecs.utk.edu/~dongarra/etemplates/book.html.
[5] Z. BAI AND R.-C. LI, *Minimization principles for the linear response eigenvalue problem I: Theory*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 1075–1100.
[6] ———, *Minimization principles for the linear response eigenvalue problem II: Computation*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 392–416.
[7] ———, *Minimization principles and computation for the generalized linear response eigenvalue problem*, BIT, 54 (2014), pp. 31–54.
[8] Z. J. BAI, R. C. LI, AND W. W. LIN, *Linear response eigenvalue problem solved by extended locally optimal preconditioned conjugate gradient methods*, Sci. China Math., 59 (2016), pp. 1443–1460.
[9] P. BENNER AND X. LIANG, *Convergence analysis of extended LOBPCG for the extreme eigenvalue*, in preparation.
[10] T. BETCKE, N. J. HIGHAM, V. MEHRMANN, C. SCHRÖDER, AND F. TISSEUR, *NLEVP: A collection of Nonlinear Eigenvalue Problems*. http://www.mims.manchester.ac.uk/research/numerical-analysis/nlevp.html.
[11] J. R. BUNCH, *Analysis of the diagonal pivoting method*, SIAM J. Numer. Anal., 8 (1971), pp. 656–680.
[12] ———, *Partial pivoting strategies for symmetric matrices*, SIAM J. Numer. Anal., 11 (1974), pp. 521–528.
[13] J. R. BUNCH AND L. KAUFMAN, *Some stable methods for calculating inertia and solving symmetric linear systems*, Math. Comp., 31 (1977), pp. 163–179.
[14] J. R. BUNCH, L. KAUFMAN, AND B. N. PARLETT, *Decomposition of a symmetric matrix*, Numer. Math., 27 (1976), pp. 95–109.
[15] J. R. BUNCH AND R. F. MARCIA, *A simplified pivoting strategy for symmetric tridiagonal matrices*, Numer. Linear Algebra Appl., 13 (2006), pp. 865–867.
[16] J. R. BUNCH AND B. N. PARLETT, *Direct methods for solving symmetric indefinite systems of linear equations*, SIAM J. Numer. Anal., 8 (1971), pp. 639–655.

669 [17] J. W. DEMMEL, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, PA, 1997.
670 [18] J. A. DUERSCH, M. SHAO, C. YANG, AND M. GU, *A robust and efficient implementation of LOBPCG*, SIAM
671 J. Sci. Comput., 40 (2018), pp. C655–C676.
672 [19] I. DUFF, R. G. GRIMES, AND J. G. LEWIS, *Users' guide for the Harwell-Boeing sparse matrix collection*,
673 Technical report TR/PA/92/86, CERFACS, 1992.
674 [20] E. G. D'YAKONOV, *Optimization in Solving Elliptic Problems*, CRC Press, Boca Raton, Florida, 1996.
675 [21] E. G. D'YAKONOV AND M. Y. OREKHOV, *Minimization of the computational labor in determining the first*
676 *eigenvalues of differential operators*, Math. Notes, 27 (1980), pp. 382–391.
677 [22] G. H. GOLUB AND Q. YE, *An inverse free preconditioned Krylov subspace method for symmetric generalized*
678 *eigenvalue problems*, SIAM J. Sci. Comput., 24 (2002), pp. 312–334.
679 [23] R. G. GRIMES, J. G. LEWIS, AND H. D. SIMON, *A shifted block Lanczos algorithm for solving sparse*
680 *symmetric generalized eigenproblems*, SIAM J. Matrix. Anal. Appl., 15 (1994), pp. 228–272.
681 [24] C.-H. GUO, N. J. HIGHAM, AND F. TISSEUR, *An improved arc algorithm for detecting definite Hermitian*
682 *pairs*, SIAM. J. Matrix Anal. Appl., 31 (2009), pp. 1131–1151.
683 [25] V. HARI, S. SINGER, AND S. SINGER, *Block-oriented J-Jacobi methods for Hermitian matrices*, Linear
684 Algebra Appl., 433 (2010), pp. 1491–1512.
685 [26] U. HETMANIUK AND R. LEHOUCQ, *Basis selection in LOBPCG*, J. Comput. Phys., 218 (2006), pp. 324–332.
686 [27] N. J. HIGHAM, F. TISSEUR, AND P. M. VAN DOOREN, *Detecting a definite Hermitian pair and a hyperbolic*
687 *or elliptic quadratic eigenvalue problem, and associated nearness problems*, Linear Algebra Appl.,
688 351/352 (2002), pp. 455–474.
689 [28] L. KAUFMAN, *The retraction algorithm for factoring banded symmetric matrices*, Numer. Linear Algebra
690 Appl., 14 (2007), pp. 237–254.
691 [29] A. V. KNYAZEV, *Calculation of eigenvalues and eigenvectors in mesh problems: algorithms and error*
692 *estimates*, Dept. Numer. Math, USSR Acad. Sci., Moscow, (1986). (in Russian).
693 [30] ———, *Convergence rate estimates for iterative methods for a mesh symmetric eigenvalue problem*, Sov. J.
694 Numer. Anal. Math. Modelling, 2 (1987), pp. 371–396.
695 [31] ———, *A preconditioned conjugate gradient method for eigenvalue problems and its implementation in a*
696 *subspace*, in Numerical Treatment of Eigenvalue Problems, Vol. 5, J. Albrecht, L. Collatz, P. Hagedorn,
697 and W. Velte, eds., vol. 96 of Internat. Ser. Numer. Math., Birkhäuser, Basel, 1991, pp. 143–154.
698 [32] ———, *Preconditioned eigensolvers - an oxymoron?*, Electron. Trans. Numer. Anal., 7 (1998), pp. 104–
699 123. http://etna.mcs.kent.edu/volumes/1993-2000/vol7/abstract.php?vol=
700 7&pages=104-123.
701 [33] ———, *Toward the optimal preconditioned eigensolver: locally optimal block preconditioned conjugate*
702 *gradient method*, SIAM J. Sci. Comput., 23 (2001), pp. 517–541.
703 [34] A. V. KNYAZEV, M. E. ARGENTATI, I. LASHUK, AND E. E. OVTCHINNIKOV, *Block locally optimal*
704 *preconditioned eigenvalue xolvers (BLOPEX) in hypre and PETSc*, SIAM J. Sci. Comput., 29 (2007),
705 pp. 2224–2239.
706 [35] A. V. KNYAZEV AND K. NEYMEYR, *Efficient solution of symmetric eigenvalue problems using multi-*
707 *grid preconditioners in the locally optimal block conjugate gradient method*, Electron. Trans. Numer.
708 Anal., 15 (2003), pp. 38–55. http://etna.mcs.kent.edu/volumes/2001-2010/vol15/
709 abstract.php?vol=15&pages=38-55.
710 [36] ———, *Gradient flow approach to geometric convergence analysis of preconditioned eigensolvers*, SIAM J.
711 Matrix Anal. Appl., 31 (2009), pp. 621–628.
712 [37] J. KOVAČ-STRIKO AND K. VESELIĆ, *Trace minimization and definiteness of symmetric pencils*, Linear
713 Algebra Appl., 216 (1995), pp. 139–158.
714 [38] D. KRESSNER, M. MILOLOŽA PANDUR, AND M. SHAO, *An indefinite variant of LOBPCG for definite*
715 *matrix pencils*, Numer. Algorithms, 66 (2014), pp. 681–703.
716 [39] P. LANCASTER AND L. RODMAN, *Canonical forms for Hermitian matrix pairs under strict equivalence and*
717 *congruence*, SIAM Rev., 47 (2005), pp. 407–443.
718 [40] P. LANCASTER AND Q. YE, *Variational and numerical methods for symmetric matrix pencils*, Bull. Austral.
719 Math. Soc., 43 (1991), pp. 1–17.
720 [41] X. LIANG AND R.-C. LI, *The hyperbolic quadratic eigenvalue problem*, Forum Math. Sigma, 3 (2015). e13,
721 (93 pages).
722 [42] X. LIANG, R.-C. LI, AND Z. BAI, *Trace minimization principles for positive semi-definite pencils*, Linear
723 Algebra Appl., 438 (2013), pp. 3085–3106.
724 [43] K. NEYMEYR, *A Hierarchy of Preconditioned Eigensolvers for Elliptic Differential Operators*, Hab. Thesis,
725 Mathematisches Institut, Universität Tübingen, 2001.
726 [44] ———, *On preconditioned eigensolvers and invert-Lanczos processes*, Linear Algebra Appl., 430 (2009),
727 pp. 1039–1056.
728 [45] ———, *A geometric convergence theory for the preconditioned steepest descent iteration*, SIAM J. Numer.
729 Anal., 50 (2012), pp. 3188–3207.

[46] K. NEYMEYR, E. OVTCHINNIKOV, AND M. ZHOU, *Convergence analysis of gradient iterations for the symmetric eigenvalue problem*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 443–456.
[47] K. NEYMEYR AND M. ZHOU, *The block preconditioned steepest descent iteration for elliptic operator eigenvalue problems*, Electron. Trans. Numer. Anal., 41 (2014), pp. 93–108. http://etna.mcs.kent.edu/volumes/2011-2020/vol41/abstract.php?vol=41&pages=93-108.
[48] B. N. PARLETT, *The Symmetric Eigenvalue Problem*, vol. 20 of Classics in Applied Mathematics, SIAM, Philadelphia, PA, 1998.
[49] B. T. POLYAK, *Introduction to optimization*, Optimization Software Inc. Publications Division, New York, 1987.
[50] P. D. QUILLEN, *Generalizations of an Inverse Free Krylov Subspace Method for the Symmetric Generalized Eigenvalue Problem*, Ph.D. Thesis, Department of Mathematics, University of Kentucky Doctoral Dissertations. 380., 2005.
[51] P. D. QUILLEN AND Q. YE, *A block inverse-free preconditioned Krylov subspace method for symmetric generalized eigenvalue problems*, J. Comput. Appl. Math., 233 (2010), pp. 1298–1313.
[52] D. ROCCA, Z. BAI, R.-C. LI, AND G. GALLI, *A block variational procedure for the iterative diagonalization of non-Hermitian random-phase approximation matrices*, J. Chem. Phys., 136 (2012). 034111 (9 pages).
[53] I. SLAPNIČAR, *Componentwise analysis of direct factorization of real symmetric and Hermitian matrices*, Linear Algebra Appl., 272 (1998), pp. 227–275.
[54] G. W. STEWART AND J. G. SUN, *Matrix Perturbation Theory*, Academic Press, Inc., Boston, MA, 1990.
[55] D. B. SZYLD AND F. XUE, *Preconditioned eigensolvers for large-scale nonlinear Hermitian eigenproblems with variational characterizations. I. Extreme eigenvalues*, Math. Comp., 85 (2016), pp. 2887–2918.
[56] K. VESELIĆ, *A Jacobi eigenreduction algorithm for definite matrix pairs*, Numer. Math., 64 (1993), pp. 241–269.
[57] ———, *Damped Oscillations of Linear Systems. A Mathematical Introduction*, vol. 2023 of Lecture Notes in Mathematics, Springer, Heidelberg, 2011.