PRECONDITIONED EIGENSOLVERS FOR LARGE-SCALE NONLINEAR HERMITIAN EIGENPROBLEMS WITH VARIATIONAL CHARACTERIZATIONS. I. CONJUGATE GRADIENT METHODS

Daniel B. Szyld and Fei Xue

Report 14-08-26
August 2014

PRECONDITIONED EIGENSOLVERS FOR LARGE-SCALE NONLINEAR HERMITIAN EIGENPROBLEMS WITH VARIATIONAL CHARACTERIZATIONS. I. CONJUGATE GRADIENT METHODS*

DANIEL B. SZYLD* AND FEI XUE†

Abstract. Preconditioned conjugate gradient (PCG) methods have been widely used for computing a few extreme eigenvalues of large-scale linear Hermitian eigenproblems. In this paper, we study PCG methods to compute extreme eigenvalues of nonlinear Hermitian eigenproblems of the form $T(\lambda)v = 0$ that admit a nonlinear variational principle. We investigate some theoretical properties of a basic CG method, including its global and asymptotic convergence. We propose several variants of single-vector and block PCG methods with deflation for computing multiple eigenvalues, and compare them in arithmetic and memory cost. Variable indefinite preconditioning is shown to be effective to accelerate convergence when some desired eigenvalues are not close to the lowest or highest one. Efficiency of these algorithms is illustrated by numerical experiments.

AMS subject classifications. 65F15, 65F10, 65F50, 15A18, 15A22.

1. Introduction.

Nonlinear Hermitian algebraic eigenproblems of the form $T(\lambda)v = 0$ arise naturally in a variety of scientific and engineering applications, such as simulations of the sound radiation from rolling tires [6], time-harmonic acoustic wave equation in bounded domains [8], delay differential equations [19], and modeling of vibrations of certain fluid-solid structures [9], loaded strings [29], and wiresaws [32]. Most of these nonlinear eigenproblems, similar to their linear counterparts, allow for a variational characterization (min-max principle) of some eigenvalues on certain intervals. Many desirable properties of these eigenvalues can be derived from the variational principle, and special numerical methods can be developed to compute them efficiently. In Part I of this study, we explore several variants of preconditioned conjugate gradient (PCG) methods for fast computation of a few extreme eigenvalues of large-scale nonlinear Hermitian eigenproblems that follow a nonlinear variational principle.

PCG methods are very efficient and widely used for solving unconstrained optimization of generic smooth functions, Hermitian positive definite linear systems, and linear Hermitian eigenproblems; see, e.g., [3, 10, 13, 14, 15, 18, 22, 23, 24], and many references therein. For linear Hermitian eigenproblems $Av = \lambda Bv$, these algorithms are directly based on the variational principle that characterizes the extreme eigenvalues as minimizers (or maximizers) of the Rayleigh quotient $\rho(x) = \frac{x^*Ax}{x^*Bx}$ over certain subspaces of proper dimensions. Inspired by the success of PCG methods for solving linear Hermitian eigenproblems, our motivation here is to extend the efficiency of these methods to the setting of solving nonlinear Hermitian eigenproblems of the form $T(\lambda)v = 0$ for extreme eigenvalues.

In particular, PCG methods are expected to outperform linearization-based methods for large hyperbolic or definite Hermitian polynomial eigenproblems [1]. Consider a hyperbolic Hermitian polynomial eigenproblem $P(\lambda)v = 0$, where $P : \mathbb{R} \to \mathbb{C}^{n\times n}$ is a Hermitian matrix-valued polynomial of degree m. There are m intervals on which the real eigenvalues of $P(\cdot)$ are characterized by a variational principle. Structure-preserving linearization of $P(\cdot)$ leads to a linear eigenproblem m times larger in dimension; in addition, most extreme eigenvalues on each of the m intervals, except for those on the outer ends of the left-most and right-most ones, become interior eigenvalues of the linearized eigenproblem, and they are thus more difficult to compute than extreme ones. Moreover, development of preconditioners for the linearized

*This report is dated 26 August 2014. This work was supported by the National Science Foundation under grants DMS-1115520, DMS-1418882 and DMS-1419100.

[1]Department of Mathematics, Temple University (038-16), 1805 N. Broad Street, Philadelphia, PA 19122-6094, USA (szyld@temple.edu)

[2]Department of Mathematics, University of Louisiana at Lafayette, P.O. Box 41010, Lafayette, LA 70504-1010, USA (fxue@louisiana.edu)

eigenproblem with block structures compatible with the companion forms necessarily involves considerable complications. By contrast, PCG methods are more competitive in this setting since they work on the original form of the problem of order n, for which preconditioners are typically easier to develop, and all extreme eigenvalues can be computed efficiently thanks to the variational principle. Moreover, PCG methods can solve truly nonlinear Hermitian eigenproblems that cannot be directly tackled by linearization.

In this paper, we obtain a new understanding of the global and asymptotic convergence of a basic CG method for computing one extreme eigenvalue of $T(\lambda)v = 0$, and we develop several variants of PCG methods for computing multiple extreme eigenvalues. We shall see that the global convergence of CG towards an eigenvalue can be guaranteed under mild assumptions. In addition, asymptotically, the behavior of CG for solving the lowest eigenpair is similar to that of CG for solving a corresponding semi-definite linear system of equations. This observation suggests that the well-known convergence rate of CG for positive definite linear systems could be descriptive of CG for solving Hermitian eigenproblems. In terms of efficiency, the standard PCG methods generally require more iterations to converge than the locally optimal variants. Moreover, we show that the use of variable indefinite preconditioning can accelerate the convergence considerably if a majority of the wanted extreme eigenvalues are not very close to the lowest or highest one of interest.

If many extreme eigenvalues are desired, PCG methods become less attractive since they are based on variational principles and thus require full deflation of all converged eigenvectors. Therefore, the storage cost of PCG is proportional to p, and the total arithmetic cost is proportional to p^2 (hard deflation) or p^4 (soft deflation), where p is the number of desired eigenvalues. To keep the costs under control for large p, we will investigate alternative preconditioned eigensolvers to compute interior eigenvalues using partial deflations. If one is interested in eigenvalues alone, these solvers require a *fixed* amount of memory, and the arithmetic cost is proportional to p. This work is to be completed in Part II of our study.

The rest of this paper is organized as follows. In Section 2, we review the basics of the nonlinear Hermitian eigenproblem, and a nonlinear variational principle in particular. In Section 3, we prove the global convergence of a CG method, and we explore the asymptotic behavior of CG for computing the lowest eigenvalue. In Section 4, we propose variants of PCG for computing multiple eigenvalues, and we highlight the implementation of deflation and the use of variable indefinite preconditioning. Numerical results are given in Section 5 to illustrate the efficiency of PCG methods. Section 6 is the conclusion of this paper.

2. Problem description and the variational principle. In this section, we describe the nonlinear Hermitian eigenproblem of interest and its basic properties. These preliminary results are used later to develop variants of PCG methods and to obtain insight into their numerical behavior. Consider the nonlinear algebraic eigenproblem $T(\lambda)v = 0$, where $T(\cdot) : J \subset \mathbb{R} \to \mathbb{C}^{n\times n}$ maps a real scalar μ in an open interval J continuously to the Hermitian matrix $T(\mu)$. Here the eigenvalue $\lambda \in J$ is a scalar for which $T(\lambda)$ is singular, and the corresponding eigenvector $v \in \mathbb{C}^n \setminus \{0\}$ lies in $\operatorname{null} T(\lambda)$. To establish a nonlinear variational principle and other relevant properties, we begin the discussion with several definitions.

Definition 2.1. *The* Rayleigh functional $\rho(\cdot) : D \to J$ $(D \subset \mathbb{C}^n \setminus \{0\})$ *is a continuous mapping of $x \in D$ to the unique solution $\rho(x) \in J$ of the scalar equation $x^*T(\rho(x))x = 0$.*

Definition 2.2. *For a given $T(\cdot)$, $J \subset \mathbb{R}$ is called an interval of* positive or negative type, *if $(\mu - \rho(x))(x^*T(\mu)x)$ is constantly positive or constantly negative, respectively, for all $x \in D$ and all $\mu \in J$, $\mu \neq \rho(x)$. Both positive and negative type are called* definite type.

Definition 2.3. *The scalar λ is* a k-th eigenvalue *of $T(\cdot)$ if zero is the k-th largest eigenvalue of the matrix $T(\lambda)$. Unless specified otherwise, a k-th eigenvalue is denoted as λ_k.*

To simplify our analysis, we make the non-restrictive assumption that $T(\cdot)$ does not have both zero and infinite eigenvalues on the inverval J. If this is not the case, choose $\sigma \in \mathbb{R}$ that is not an eigenvalue and define $\widetilde{T}(\mu) = T(\mu + \sigma)$ such that zero is not an eigenvalue of $\widetilde{T}(\cdot)$. If $T(\cdot)$ has infinite but not zero eigenvalues, consider the problem $S(\lambda)v = T(\frac{1}{\lambda})v = 0$ that maps infinite eigenvalues of $T(\cdot)$ to zero eigenvalues of $S(\cdot)$. Therefore, we can assume that $J = (a, b)$ is finite, and that a and b are not eigenvalues of $T(\cdot)$. The following proposition gives sufficient and necessary conditions for J to be of definite type.

Proposition 2.4. *Let $J = (a, b) \subset \mathbb{R}$ be finite, where a, b are not eigenvalues of $T(\cdot)$, and $D = \mathbb{C}^n \setminus \{0\}$ be the domain of the Rayleigh functional $\rho : D \to J$. Then J is an interval of positive (negative) type if and only if $T(a)$ is negative (positive) definite and $T(b)$ is positive (negative) definite. Assume that $T(\cdot)$ is continuously differentiable. Then J is of positive (negative) type if $x^*T'(\rho(x))x > 0$ (< 0) for all $x \in D$. If, in addition, $T(\cdot)$ is twice continuously differentiable and $x^*T''(\rho(x))x \neq 0$, then J is of positive (negative) type if and only if $x^*T'(\rho(x))x > 0$ (< 0) for all $x \in D$.*

Proof. We first establish the relation between the type of J and the definiteness of $T(a)$ and $T(b)$. Let $J = (a, b)$ be an interval of positive type; that is, $(\mu - \rho(x))(x^*T(\mu)x) > 0$ for all nonzero $x \in \mathbb{C}^n$ and all $\mu \in (a, b)$, $\mu \neq \rho(x)$. By the continuity of $T(\cdot)$, we have $(a - \rho(x))(x^*T(a)x) \geq 0$, i.e., $x^*T(a)x \leq 0$ for all nonzero vector x. By our assumption that a is not an eigenvalue of $T(\cdot)$, $T(a)$ is nonsingular, and thus $T(a) < 0$. Similarly, one can show that $T(b) > 0$. If J is of negative type, it is clear that the conclusion on the sign of the definiteness of $T(a)$ and $T(b)$ should be reversed.

If $T(a) < 0$ and $T(b) > 0$, then $(a - \rho(x))(x^*T(a)x) > 0$, and $(b - \rho(x))(x^*T(b)x) > 0$ for all $x \neq 0$. For any given $x \neq 0$, by the continuity of $T(\cdot)$, $(\mu - \rho(x))(x^*T(\mu)x) > 0$ for all $\mu \in (a, \rho(x)) \cup (\rho(x), b)$; otherwise, there is a $\mu_0 \in (a, \rho(x)) \cup (\rho(x), b)$ such that $(\mu_0 - \rho(x))(x^*T(\mu_0)x) = 0$, which is contradictory to the uniqueness of $\rho(x)$ as the solution of $x^*T(\rho(x))x = 0$. Therefore $J = (a, b)$ is an interval of positive type. Similarly, if $T(a) > 0$ and $T(b) < 0$, it is easy to show that $J = (a, b)$ is of negative type.

We now study the connection between the type of J and the sign of $x^*T(\rho(x))x$. Assume that $x^*T'(\rho(x))x = \lim_{\mu \to \rho(x)} \frac{x^*T(\mu)x - x^*T(\rho(x))x}{\mu - \rho(x)} > 0$ for all $x \in D$. Since $T(\cdot)$ is continuous, there is a $\delta > 0$, such that $\frac{x^*T(\mu)x - x^*T(\rho(x))x}{\mu - \rho(x)} > 0$ for all $\mu \in (\rho(x) - \delta, \rho(x) + \delta)$, $\mu \neq \rho(x)$; in particular, $x^*T(\mu)x < 0$ for $\mu \in (\rho(x) - \delta, \rho(x))$. For any $\mu \in [a, \rho(x))$, note that $\mu - \rho(x) < 0$, $T(\cdot)$ is continuous, and $\rho(x)$ is the unique zero of $x^*T(\rho)x = 0$. Therefore $x^*T(\mu)x < 0$ for all $\mu \in [a, \rho(x))$. Since this observation holds for any $x \in D$, $T(a) < 0$ is established. One can show similarly that $T(b) > 0$. Clearly, if $x^*T'(\rho(x))x < 0$ for all $x \in D$, then the sign of the definiteness of $T(a)$ and $T(b)$ should be reversed.

Assume in addition that $T(\cdot)$ is twice continuously differentiable. To complete the proof, we simply need to show that J is of positive type only if $x^*T'(\rho(x))x > 0$ for all $x \in D$. In this case, $(\mu - \rho(x))(x^*T(\mu)x) > 0$, i.e., $\frac{x^*T(\mu)x - x^*T(\rho(x))x}{\mu - \rho(x)} > 0$ for all $x \in D$ and all $\mu \in J$, $\mu \neq \rho(x)$. Thus $x^*T'(\rho(x))x = \lim_{\mu \to \rho(x)} = \frac{x^*T(\mu)x - x^*T(\rho(x))x}{\mu - \rho(x)} \geq 0$. Therefore, we only need to show that $x^*T'(\rho(x))x \neq 0$. Assume by contradiction that $x^*T'(\rho(x))x = 0$ for some $x \in D$. Let $\delta\rho = \mu - \rho(x)$, and consider $x^*T(\mu)x = x^*T(\rho(x) + \delta\rho)x = x^*T(\rho(x))x + (x^*T'(\rho(x))x)\delta\rho + \frac{1}{2}(x^*T''(\xi)x)(\delta\rho)^2$, where $\xi \in (\rho(x) - |\delta\rho|, \rho(x) + |\delta\rho|)$. For $|\delta\rho|$ sufficiently small, since $T(\cdot)$ is twice continuously differentiable, $x^*T''(\xi)x$ is nonzero and is of the same sign as $x^*T''(\rho(x))x$. Note that $x^*T(\rho(x))x = 0$, and $x^*T'(\rho(x))x = 0$ by assumption, and thus $x^*T(\mu)x$ is of the same sign as $x^*T''(\rho(x))x$ for all μ close to $\rho(x)$. As a result, $(\mu - \rho(x))(x^*T(\mu)x)$ changes sign as μ goes across $\rho(x)$. However, this is contradictory to the fact that $(\mu - \rho(x))(x^*T(\mu)x)$ is constantly positive or constantly negative for all $\mu \in J$, $\mu \neq \rho(x)$. This completes the proof that $x^*T'(\rho(x))x > 0$. Similarly, if J is of negative type, then $x^*T'(\rho(x))x < 0$. □

The sufficient and necessary conditions given in Proposition 2.4 can be used to determine whether J is or can be of definite type. On an interval of definite type, we have a variational characterization of eigenvalues of $T(\cdot)$ and the orthogonality of eigenvectors [17, 30, 31]. This result, as reiterated below, plays a central role for the development of CG methods.

THEOREM 2.5 (Nonlinear Variational Principle). *Let $J \subset \mathbb{R}$ be finite and of definite type, and $T(\cdot)$ be continuously differentiable on J. Then there exist exactly n eigenvalues $\{\lambda_k\}_{k=1}^n$ of $T(\cdot)$ on J that satisfy a variational principle. Specifically, if J is of positive type, then*

$$\begin{aligned} \lambda_k &= \min\{\max\{\rho(x) \,|\, x \in S, x \neq 0\} \,|\, \dim(S) = k\} \qquad \text{and} \\ \lambda_k &= \max\{\min\{\rho(x) \,|\, x \in S, x \neq 0\} \,|\, \dim(S) = n-k+1\}; \end{aligned} \tag{2.1}$$

if J is of negative type, then

$$\begin{aligned} \lambda_k &= \max\{\min\{\rho(x) \,|\, x \in S, x \neq 0\} \,|\, \dim(S) = k\} \qquad \text{and} \\ \lambda_k &= \min\{\max\{\rho(x) \,|\, x \in S, x \neq 0\} \,|\, \dim(S) = n-k+1\}. \end{aligned} \tag{2.2}$$

Moreover, there exist n corresponding eigenvectors $\{v_k\}_{k=1}^n$ that form a basis for $\mathbb{C}^n$, and they are orthogonal with respect to the scalar product $[\cdot,\cdot]$ defined as

$$[x,y] = \left\{ \begin{array}{c} y^* \frac{T(\rho(y)) - T(\rho(x))}{\rho(y)-\rho(x)} x \quad (\rho(x) \neq \rho(y)) \\ y^* T'(\rho(x)) x \quad (\rho(x) = \rho(y)) \end{array} \right. . \tag{2.3}$$

The variational principle stated in Theorem 2.5 can be used to derive several interesting properties of eigenvalue approximations obtained by projecting $T(\cdot)$ onto low dimensional spaces. For example, the following two theorems give some bounds on the Ritz values. Note in particular that the Rayleigh functional value $\rho(x)$ is the Ritz value obtained from the one-dimensional projected problem $(x^* T(\rho(x)) x)\, v = 0v = 0$, where v is a nonzero scalar.

THEOREM 2.6. *Let $J = (a,b)$ be finite and of definite type, and $T(\cdot)$ be continuous. For any $x \in \mathbb{C}^n \setminus \{0\}$, let $x = \sum_{i=1}^m c_i v_{j_i}$ $(1 \le m \le n,\ 1 \le j_1 < j_2 < \ldots < j_m \le n)$ be the eigenvector decomposition of x, where $c_i \neq 0$ for all $1 \le i \le m$. If J is of positive type, then $\lambda_{j_1} \le \rho(x) \le \lambda_{j_m}$; if in addition $\lambda_{j_1} < \lambda_{j_m}$, then $\lambda_{j_1} < \rho(x) < \lambda_{j_m}$; if J is of negative type, then $\lambda_{j_m} \le \rho(x) \le \lambda_{j_1}$; if in addition $\lambda_{j_m} < \lambda_{j_1}$, then $\lambda_{j_m} < \rho(x) < \lambda_{j_1}$.*

Proof. See the Appendix. □

THEOREM 2.7 (Nonlinear Cauchy interlacing theorem). *Let $J = (a,b)$ be finite and of definite type, $T(\cdot)$ be continuously differentiable on J, and $U \in \mathbb{C}^{n\times m}$ contain m linearly independent columns. Then the projected eigenproblem $T_p(\nu)w \equiv U^*T(\nu)Uw = 0$ has exactly m eigenpairs $\{(\nu_j, w_j)\}_{j=1}^m$ satisfying the nonlinear variational principle* (2.1) or (2.2)*, where ν_j is a j-th eigenvalue of $T_p(\cdot)$. In addition, if J is of positive type, then $\lambda_j \le \nu_j \le \lambda_{n-m+j}$; if J is of negative type, then $\lambda_{n-m+j} \le \nu_j \le \lambda_j$ $(1 \le j \le m)$.*

Proof. First, note that for any given nonzero $x \in \mathbb{C}^m$, the solution $\rho \in J$ of the scalar equation $x^*T_p(\rho)x = (Ux)^*T(\rho)(Ux) = 0$ is unique because of the uniqueness of the Rayleigh functional value for $T(\cdot)$. Therefore the Rayleigh functional $\rho(\cdot) : \mathbb{C}^m \backslash \{0\} \to \mathbb{R}$ is well defined for $T_p(\cdot)$. Without loss of generality, assume that J is of positive type, so that $T(a) < 0$ and $T(b) > 0$. Consequently, $T_p(a) = U^*T(a)U < 0$, $T_p(b) = U^*T(b)U > 0$. By Proposition 2.4, J is also of positive type for $T_p(\cdot)$. In addition, since $T(\cdot)$ is continuously differentiable on J, so is $T_p(\cdot) = U^*T(\cdot)U$. It follows from Theorem 2.5 that there are exactly m eigenvalues of $T_p(\cdot)$ on J, denoted as $\{\nu_j\}_{j=1}^m$, that satisfy the variational principle (2.1) or (2.2).

To establish the interlacing result, let the Rayleigh functional of the projected operator $T_p(\cdot)$ be $\rho_p(x)$. Since $x^*T_p(\rho_p(x))x = 0 = (Ux)^*T(\rho(Ux))(Ux)$, $\rho_p(x) \equiv \rho(Ux)$. Let $W_j = \operatorname{span}\{w_1, \ldots, w_j\}$ $(1 \le j \le m)$. Then

$$\begin{aligned} \nu_j &= \max\{\rho_p(x) \,|\, x \in W_j, x \neq 0\} = \max\{\rho(z) \,|\, z \in UW_j, z \neq 0\} \\ &\ge \min\{\max\{\rho(z) \,|\, z \in S, z \neq 0\} \,|\, \dim(S) = j\} = \lambda_j. \end{aligned} \tag{2.4}$$

Now let $W_{m-j+1} = \text{span}\{w_j, \ldots, w_m\}$. Then

$$\begin{aligned}\nu_j &= \min\{\rho_p(x) \mid x \in W_{m-j+1}, x \neq 0\} = \min\{\rho(z) \mid z \in UW_{m-j+1}, z \neq 0\} \qquad (2.5)\\ &\leq \max\{\min\{\rho(z) \mid z \in S, z \neq 0\} \mid \dim(S) = m-j+1\} = \lambda_{n-(m-j+1)+1} = \lambda_{n-m+j}.\end{aligned}$$

The proof for J of negative type is very similar and is thus omitted. □

We will use Theorem 2.7 to show that the exact line search in CG for the optimization of the Rayleigh functional can be achieved by the Rayleigh-Ritz projection, and that the locally optimal variant of CG converges more rapidly than regular CG in one iteration step if both methods start with the same iterate.

3. Global and asymptotic convergence of CG.

In this section, we explore the convergence of a CG method for computing one eigenvalue of the nonlinear Hermitian eigenproblem $T(\lambda)v = 0$ admitting the nonlinear variational principle (2.1) or (2.2). By this principle, the computation of extreme eigenvalues can be achieved by the optimization of the Rayleigh functional, for which CG methods are naturally suitable and efficient.

3.1. Global convergence.

It has been observed in extensive numerical experiments that single-vector CG methods with deflation converge very robustly towards extreme eigenvalues of linear Hermitian eigenproblems $Av = \lambda v$ or $Av = \lambda Bv$. Such a robustness can be explained by the global convergence established in [33] for the standard CG method with the exact line search and the Fletcher-Reeves formula. Here, we give a similar analysis of the global convergence of a special variant of CG in the new setting of nonlinear eigenproblems.

Before we study the global convergence, one should realize that the standard global convergence analysis of CG in the whole Euclidean space is not directly applicable to the optimization of Rayleigh functionals. This is because the Rayleigh functional $\rho(x)$ depends on the direction of x alone and not on its scaling, i.e., $\rho(x) = \rho(\alpha x)$ for all nonzero scalars α and vectors x. As a result, the gradient of $\rho(\cdot)$ satisfies $\frac{1}{\alpha}\nabla\rho(x) = \nabla\rho(\alpha x)$; that is, if x in $\nabla\rho(x)$ is replaced with αx, the gradient vector is scaled by a factor of $\frac{1}{\alpha}$. Therefore, the traditional stopping criterion of CG in the general setting of optimization – $\|\nabla\rho(x)\| \leq \delta$ for some small $\delta > 0$ – cannot guarantee that x approximates the desired eigenvector in direction. In fact, due to the dependence on scaling, $\nabla\rho(x)$ does not satisfy the Lipschitz condition $\|\nabla\rho(x_1) - \nabla\rho(x_2)\| \leq L\|x_1 - x_2\|$ for all $x_1, x_2 \in \mathbb{C}^n \setminus \{0\}$. One approach to get around this issue is to develop CG on Grassman manifolds; see [11]. However, the understanding of theoretical properties of these CG methods, especially their convergence, remains far from complete. Instead, we propose a special variant of CG that works independently of the scaling of any iterate x_k, for which the global convergence can be established.

From now on, for the sake of simplicity, we denote the values of the Rayleigh functional $\rho(x)$, its gradient $\nabla\rho(x)$ and its Hessian $\nabla^2\rho(x)$ corresponding to the vector x, as ρ, $\nabla\rho$ and $\nabla^2\rho$, respectively, when there is no danger of confusion. To facilitate the study of global convergence, we first give expressions of $\nabla\rho$ and $\nabla^2\rho$. To simplify our analysis, we assume in this section that $T(\cdot)$ is real symmetric.

PROPOSITION 3.1. *Let $T(\cdot) : \mathbb{R} \to \mathbb{R}^{n\times n}$ be a real symmetric matrix-valued function, and $\rho(\cdot) : \mathbb{C}^n \setminus \{0\} \to J \subset \mathbb{R}$ be the Rayleigh functional, where J is of definite type. Assume that $T(\cdot)$ and $\rho(\cdot)$ are twice continuously differentiable, and $x^T T'(\rho)x \neq 0$ for all $x \neq 0$. Then*

$$\nabla\rho = -\frac{2}{x^T T'(\rho)x} T(\rho)x, \qquad \textit{and} \qquad (3.1)$$

$$\nabla^2\rho = -\frac{2}{x^T T'(\rho)x}\left(T(\rho) + \nabla\rho\, x^T T'(\rho) + T'(\rho)x\,\nabla\rho^T + \frac{x^T T''(\rho)x}{2}\nabla\rho\,\nabla\rho^T\right). \qquad (3.2)$$

Proof. See the Appendix. □

Remark. The assumption that $x^T T'(\rho)x \neq 0$ for all $x \neq 0$ holds for the linear Hermitian eigenproblem $Av = \lambda Bv$ with a positive definite B, as $x^T T'(\rho)x = x^T Bx > 0$. In addition, note that if $x^T T''(\rho)x \neq 0$ for all $x \neq 0$, then the assumption $x^T T'(\rho)x \neq 0$ holds from Proposition 2.4.

Having the expressions of the gradient and Hessian of $\rho(\cdot)$, we can estimate the accuracy of $\rho(x)$ as an eigenvalue approximation and the magnitude of $\nabla\rho$ if x is a good eigenvector approximation. This result will be used in the study of the asymptotic convergence of CG.

PROPOSITION 3.2. *Suppose that the assumptions of* Proposition 3.1 *hold. Let λ_ℓ be an eigenvalue of $T(\cdot)$, and $v_\ell \in \operatorname{null} T(\lambda_\ell)$ be a unit eigenvector. Let $x = \gamma\,(v_\ell \cos\theta + g\sin\theta)$ be an eigenvector approximation, where $g \perp \operatorname{null} T(\lambda_\ell)$ is a unit vector. Then for all θ sufficiently small, $|\rho(x) - \lambda_\ell| = \mathcal{O}(\tan^2\theta)$. If, in addition, $\gamma = \|x\|$ satisfies $c \leq \gamma \leq C$ for some constants $c, C > 0$ independently of θ, then $\|\nabla\rho(x)\| = \mathcal{O}(\sin\theta)$.*

Proof. See the Appendix. □

3.1.1. CG with inexact line search (Wolfe conditions). In this section, we present and analyze a special Fletcher-Reeves variant of CG algorithm with inexact line search for computing the extreme eigenvalue of a real symmetric eigenproblem $T(\lambda)v = 0$ satisfying the variational principle (2.1) or (2.2). Without loss of generality, assume that the lowest eigenvalue is of interest. Note that this method uses the *scaling-invariant* gradient $\|x\|\nabla\rho$ to construct search directions and test convergence.

```
Algorithm 1 : CG with inexact line search for computing the lowest eigenvalue of $T(\lambda)v = 0$
1. Choose $x_0 \neq 0$.
2. For $k = 0, 1, \ldots,$ until convergence, i.e., $\|x_k\|\|\nabla\rho(x_k)\| \leq \delta$
3.     If $k = 0$, set $p_0 = -\|x_0\|\nabla\rho(x_0)$;
       otherwise, set $\beta_k = \frac{\nabla\rho(x_k)^T\nabla\rho(x_k)\|x_k\|^2}{\nabla\rho(x_{k-1})^T\nabla\rho(x_{k-1})\|x_{k-1}\|^2}$, and $p_k = -\|x_k\|\nabla\rho(x_k) + \beta_k p_{k-1}$.
4.     Perform a line search satisfying the strong Wolfe conditions, i.e., find $\alpha_k > 0$ such that
       $\rho(x_{k+1}) \leq \rho(x_k) + c_1\alpha_k\nabla\rho(x_k)^T p_k$  and  $\|x_{k+1}\||\nabla\rho(x_{k+1})^T p_k| \leq -c_2\|x_k\|\nabla\rho(x_k)^T p_k$,
       where $0 < c_1 < c_2 < 1$.
5.     Set $x_{k+1} = x_k + \alpha_k p_k$, and normalize $x_{k+1}$ if necessary.
6. End For
```

An interesting and attractive feature of Algorithm 1 is that we are free to scale each new iterate x_{k+1} by any nonzero factor in Step 5. In particular, there is no need to mimic the CG constructed on Grassmann manifold for linear Hermitian eigenproblems [11], which takes great efforts to tune CG to proceed in a manner consistent with the geometry of the unit sphere. This is because by construction, both β_k and p_k in Algorithm 1 are *scaling-invariant* of the CG iterates; that is, they depend only on the directions, instead of the scalings, of $x_0, x_1, \ldots$; see (3.1) and Step 3. Therefore, one can normalize x_k in any convenient manner after each CG step without being concerned about any potential geometric constraints.

Our main interest here is to prove the global convergence of Algorithm 1 towards some (ideally the lowest) eigenpair. To this end, we need to establish several intermediate results.

DEFINITION 3.3. *The gradient $\nabla\rho$ as given in* (3.1) *is called* Lipschitz continuous in direction *if there is a constant $L > 0$ such that $\big\|\|x_1\|\nabla\rho(x_1) - \|x_2\|\nabla\rho(x_2)\big\| \leq L\alpha$ for all $x_1, x_2 \in \mathbb{C}^n \setminus \{0\}$ that satisfy $\alpha = \angle(x_1, x_2) \leq \frac{\pi}{2}$.*

To establish the global convergence, we derive a useful inequality for $\nabla\rho$ that is Lipschitz continuous in direction. Since $\alpha = \angle(x_1, x_2) \leq \frac{\pi}{2}$ by definition, $\alpha \leq \frac{\pi}{2}\sin\alpha$. Also, for any $x_1, x_2 \neq 0$ such that $\angle(x_1, x_2) = \alpha$, $\sin\alpha = \min\left\{\frac{\|x_1 - x_2\|}{\|x_1\|}\,\middle|\,\angle(x_1, x_2) = \alpha\right\} \leq \frac{\|x_1 - x_2\|}{\|x_1\|}$, and similarly $\sin\alpha \le \frac{\|x_1-x_2\|}{\|x_2\|}$. Thus, if $\nabla\rho$ is Lipschitz continuous in direction, then

$$\big\|\|x_1\|\nabla\rho(x_1) - \|x_2\|\nabla\rho(x_2)\big\| \le L\alpha \le \frac{\pi L}{2}\sin\alpha \le \frac{\pi L\|x_1-x_2\|}{2\max(\|x_1\|,\|x_2\|)}, \tag{3.3}$$

for all $x_1, x_2 \neq 0$, $\angle(x_1,x_2) \le \frac{\pi}{2}$. This inequality will be used to prove the following theorem.

Theorem 3.4. *Let $J=(a,b)$ be finite and of definite type, and $T(\cdot)$ be a real symmetric matrix-valued function continuously differentiable on J for which the variational principle* (2.1) *or* (2.2) *holds. Consider the iteration $x_{k+1}=x_k+\alpha_k p_k$, where p_k is a descent direction for the Rayleigh functional $\rho(\cdot):\mathbb{R}^n\to J\subset\mathbb{R}$, and α_k satisfies the* strong Wolfe conditions

$$\rho(x_{k+1}) \le \rho(x_k) + c_1\alpha_k\nabla\rho(x_k)^Tp_k, \qquad \textit{and} \tag{3.4}$$

$$\|x_{k+1}\||\nabla\rho(x_{k+1})^Tp_k| \le -c_2\|x_k\|\nabla\rho(x_k)^Tp_k, \tag{3.5}$$

with $0<c_1<c_2<1$. Let $\theta_k=\angle(-\nabla\rho(x_k),p_k)$, such that $\cos\theta_k=\frac{-\nabla\rho(x_k)^Tp_k}{\|\nabla\rho(x_k)\|\|p_k\|}>0$. Assume that $\rho(\cdot)$ is continuously differentiable in an open set $G\subset\mathbb{R}^n\setminus\{0\}$ containing the level set $\{x\,|\,x\neq 0,\rho(x)\le\rho(x_0)\}$, and that $\nabla\rho$ is Lipschitz continuous in direction on G. Then

$$\sum_{k=0}^{\infty}\|x_k\|^2\|\nabla\rho(x_k)\|^2\cos^2\theta_k < \infty. \tag{3.6}$$

Proof. See the Appendix. □

Lemma 3.5. *Suppose that* Algorithm 1 *is implemented with a step length α_k that satisfies the strong Wolfe conditions* (3.4)-(3.5) *with $0<c_2<\frac{1}{2}$. Then the method generates descent directions p_k that satisfy the following inequality for all k*

$$-\frac{1}{1-c_2} \le \frac{\nabla\rho(x_k)^Tp_k}{\|\nabla\rho(x_k)\|^2\|x_k\|} \le \frac{2c_2-1}{1-c_2} \tag{3.7}$$

Proof. See the Appendix. □

With the above preliminary results, we are ready to establish the global convergence of Algorithm 1.

Theorem 3.6. *Let $J=(a,b)$ be finite and of definite type, and $T(\cdot)$ be a real symmetric matrix-valued function continuously differentiable on J, for which the variational principle* (2.1) *or* (2.2) *holds. Let $x_0\neq 0$ be the initial iterate of* Algorithm 1, *which is implemented with an inexact line search satisfying the Strong Wolfe conditions* (3.4)-(3.5) *where $0<c_1<c_2<\frac{1}{2}$. Assume that the gradient $\nabla\rho$ is Lipschitz continuous in direction in a neighborhood of $S=\left\{\frac{x}{\|x\|}\,\middle|\,x\neq 0,\rho(x)\le\rho(x_0)\right\}$, and that $\sup_{x\in S}\frac{\|x\|^2}{|x^TT'(\rho)x|}=F>0$. Then there exists $\lambda_\ell\in\{\lambda_i\}$ such that* $\lim_{k\to\infty}\rho(x_k)=\lambda_\ell$, *and* $\lim_{k\to\infty}\angle\left(x_k,\operatorname{null}T(\lambda_\ell)\right)=0$.

Proof. If x_0 is an eigenvector corresponding to the eigenvalue λ_ℓ, e.g., $x_0\in\operatorname{null}T(\lambda_\ell)$, then $\rho(x_0)=\lambda_\ell$, and thus $\nabla\rho(x_0)=-\frac{2}{x_0^TT'(\lambda_\ell)x_0}T(\lambda_\ell)x_0=0$. The theorem holds trivially.

Otherwise, note that each iteration of Algorithm 1 generates a new iterate x_k such that $\rho(x_k)<\rho(x_{k-1})$. Therefore all iterates of the algorithm belong to the level set S. Also, since $\|T(\cdot)\|:\mathbb{R}\to\mathbb{R}$ is a continuous function defined on a closed interval $[\lambda_1,\lambda_n]$ or $[\lambda_n,\lambda_1]$, there exists an $M>0$ such that $\|T(\rho(x))\|\le M$ for all $x\neq 0$. By assumption, $\sup_{x\in S}\frac{\|x\|^2}{|x^TT'(\rho)x|}=F>0$, and it follows from (3.1) that $\|x\|\|\nabla\rho\|_{x\in S}\le\frac{2\|T(\rho)\|\|x\|^2}{|x^TT'(\rho)x|}\le 2MF:=\Gamma<\infty$.

For $\theta_k = \angle(-\nabla\rho(x_k), p_k)$, $\cos\theta_k = \frac{-\nabla\rho(x_k)^T p_k}{\|\nabla\rho(x_k)\|\|p_k\|} = \frac{-\nabla\rho(x_k)^T p_k}{\|\nabla\rho(x_k)\|^2\|x_k\|}\frac{\|\nabla\rho(x_k)\|\|x_k\|}{\|p_k\|}$. It follows from (3.7) in Lemma 3.5 that $0 \leq \frac{1-2c_2}{1-c_2}\frac{\|\nabla\rho(x_k)\|\|x_k\|}{\|p_k\|} \leq \cos\theta_k$. By Theorem 3.4, we have

$$\sum_{k=0}^{\infty}\left(\frac{1-2c_2}{1-c_2}\frac{\|\nabla\rho(x_k)\|\|x_k\|}{\|p_k\|}\right)^2\|\nabla\rho(x_k)\|^2\|x_k\|^2 \leq \sum_{k=0}^{\infty}\|\nabla\rho(x_k)\|^2\|x_k\|^2\cos^2\theta_k < \infty.$$

Since $0 < \frac{1-2c_2}{1-c_2} < 1$, it follows that

$$\sum_{k=0}^{\infty}\frac{\|\nabla\rho(x_k)\|^4\|x_k\|^4}{\|p_k\|^2} < \infty. \tag{3.8}$$

Meanwhile, note from the second Wolfe condition (3.5) and Lemma 3.5 that

$$\|x_k\|\left|\nabla\rho(x_k)^T p_{k-1}\right| \leq -c_2\|x_{k-1}\|\nabla\rho(x_{k-1})^T p_{k-1} \leq \frac{c_2}{1-c_2}\|\nabla\rho(x_{k-1})\|^2\|x_{k-1}\|^2, \tag{3.9}$$

and therefore, from Step 3 of Algorithm 1,

$$\begin{aligned}\|p_k\|^2 &\leq \|x_k\|^2\|\nabla\rho(x_k)\|^2 + 2\beta_k\|x_k\|\left|\nabla\rho(x_k)^T p_{k-1}\right| + \beta_k^2\|p_{k-1}\|^2 \\ &\leq \|x_k\|^2\|\nabla\rho(x_k)\|^2 + 2\beta_k\frac{c_2}{1-c_2}\|\nabla\rho(x_{k-1})\|^2\|x_{k-1}\|^2 + \beta_k^2\|p_{k-1}\|^2 \\ &= \left(1+\frac{2c_2}{1-c_2}\right)\|x_k\|^2\|\nabla\rho(x_k)\|^2 + \beta_k^2\|p_{k-1}\|^2.\end{aligned} \tag{3.10}$$

Let $c_3 = 1 + \frac{2c_2}{1-c_2}$ such that $1 < c_3 < 3$. To expand the upper bound for $\|p_k\|^2$, the above inequality can be applied step by step to bound $\|p_{k-1}\|^2$, $\|p_{k-2}\|^2$, ... and $\|p_1\|^2$. We have

$$\begin{aligned}\|p_k\|^2 &\leq c_3\|x_k\|^2\|\nabla\rho(x_k)\|^2 + \beta_k^2\left(c_3\|x_{k-1}\|^2\|\nabla\rho(x_{k-1})\|^2 + \beta_{k-1}^2\|p_{k-2}\|^2\right) \\ &\leq c_3\Bigg(\|x_k\|^2\|\nabla\rho(x_k)\|^2 + \beta_k^2\|x_{k-1}\|^2\|\nabla\rho(x_{k-1})\|^2 + \\ &\qquad \beta_k^2\beta_{k-1}^2\|x_{k-2}\|^2\|\nabla\rho(x_{k-2})\|^2 + \ldots + \prod_{i=1}^{k}\beta_i^2\|x_0\|^2\|\nabla\rho(x_0)\|^2\Bigg).\end{aligned} \tag{3.11}$$

To establish the theorem, we now assume by contradiction that there exists $\gamma > 0$ such that $\|x_k\|\|\nabla\rho(x_k)\| \geq \gamma$ for all k. Note from the definition of β_k (Step 3 of Algorithm 1) that $\beta_k^2\beta_{k-1}^2\ldots\beta_{k-i}^2 = \frac{\|\nabla\rho(x_k)\|^4\|x_k\|^4}{\|\nabla\rho(x_{k-i-1})\|^4\|x_{k-i-1}\|^4}$, and therefore

$$\begin{aligned}\|p_k\|^2 &\leq c_3\left(\|\nabla\rho(x_k)\|^2\|x_k\|^2 + \frac{\|\nabla\rho(x_k)\|^4\|x_k\|^4}{\|\nabla\rho(x_{k-1})\|^2\|x_{k-1}\|^2} + \ldots + \frac{\|\nabla\rho(x_k)\|^4\|x_k\|^4}{\|\nabla\rho(x_0)\|^2\|x_0\|^2}\right) \\ &= c_3\|\nabla\rho(x_k)\|^4\|x_k\|^4\sum_{i=0}^{k}\frac{1}{\|\nabla\rho(x_i)\|^2\|x_i\|^2} \leq c_3\gamma_k^4\frac{k+1}{\gamma^2},\end{aligned} \tag{3.12}$$

where $\gamma_k = \|\nabla\rho(x_k)\|\|x_k\|$ satisfies $0 < \gamma \leq \gamma_k \leq \Gamma < \infty$. It follows that

$$\sum_{k=0}^{m}\frac{1}{\|p_k\|^2} \geq \frac{\gamma^2}{c_3}\sum_{k=0}^{m}\frac{1}{\gamma_k^4(k+1)} \geq \frac{\gamma^2}{c_3\Gamma^4}\sum_{k=0}^{m}\frac{1}{k+1},$$

and thus

$$\sum_{k=0}^{\infty} \frac{1}{\|p_k\|^2} \geq \frac{\gamma^2}{c_3\Gamma^4} \sum_{k=0}^{\infty} \frac{1}{(k+1)} = \infty. \tag{3.13}$$

However, since $\sum_{k=0}^{\infty} \frac{\|\nabla\rho(x_k)\|^4\|x_k\|^4}{\|p_k\|^2} < \infty$ (see (3.8)), and $\|\nabla\rho(x_k)\|\|x_k\| \geq \gamma > 0$ for all k by assumption, we have $\sum_{k=0}^{\infty} \frac{1}{\|p_k\|^2} < \infty$, contradicting (3.13). Therefore such $\gamma > 0$ does not exist, and thus $\lim_{k\to\infty} \inf \|x_k\|\|\nabla\rho(x_k)\| = 0$. This means there exists a subsequence of $\{x_k\}$, denoted as $\{x_{k_j}\}$, such that $\lim_{j\to\infty} \|x_{k_j}\|\|\nabla\rho(x_{k_j})\| = 0$.

To establish the convergence, note that Algorithm 1 generates a sequence of iterates $\{x_k\}$ such that $\rho(x_{k+1}) \leq \rho(x_k)$. Since $\rho(x) \in [\lambda_1, \lambda_n] \subset J$ and J is finite, there exists $\lambda^* \in [\lambda_1, \lambda_n]$ such that $\lim_{k\to\infty} \rho(x_k) = \lambda^*$. Consequently, we also have $\lim_{j\to\infty} \rho(x_{k_j}) = \lambda^*$. Our goal is to show that λ^* is an eigenvalue. Assume by contradiction that λ^* is not an eigenvalue, such that $T(\lambda^*)$ is nonsingular. Let $\sigma^*_{\min} > 0$ be the smallest singular value of $T(\lambda^*)$ such that $\|T(\lambda^*)x\| \geq \sigma^*_{\min}\|x\|$ for all $x \neq 0$. It follows from (3.1) that

$$0 = \lim_{j\to\infty} \|x_{k_j}\|\|\nabla\rho(x_{k_j})\| = \lim_{j\to\infty} \frac{2\left\|T(\rho(x_{k_j}))\frac{x_{k_j}}{\|x_{k_j}\|}\right\|}{\left|\frac{x_{k_j}^T}{\|x_{k_j}\|}T'\left(\rho(x_{k_j})\right)\frac{x_{k_j}}{\|x_{k_j}\|}\right|} \geq \frac{2\sigma^*_{\min}}{\max_{\rho\in J}\|T'(\rho)\|} > 0,$$

an obvious contradiction. Therefore $\lambda^* = \lim_{k\to\infty} \rho(x_k)$ is an eigenvalue, and we let it be λ_ℓ.

Finally, we show that $\lim_{k\to\infty} \angle\left(x_k, \operatorname{null} T(\lambda_\ell)\right) = 0$. Assume by contradiction that this is not the case, i.e., there exists $\delta_1 > 0$ independent of the iteration count, such that for any $M > 0$, there exists an $m > M$ such that $\angle\left(x_m, \operatorname{null} T(\lambda_\ell)\right) \geq \delta_1$. By the continuity of the Rayleigh functional $\rho(\cdot)$, there exits $\delta_2 > 0$, which depends on δ_1 and is independent of the iteration count, such that $|\rho(x_m) - \lambda_\ell| \geq \delta_2$. This contradicts the fact that $\lim_{k\to\infty} \rho(x_k) = \lambda_\ell$. Thus the convergence of $\{x_k\}$ towards the eigenspace is established. □

3.1.2. CG with exact line search (projection). Inexact line search is widely used for CG in the general setting of unconstrained nonlinear optimization. For the solution of extreme eigenvalues of Hermitian eigenproblems, essentially all variants of CG used in practice perform exact line search achieved by the Rayleigh-Ritz projection. We replace inexact line search in Algorithm 1 with exact line search and obtain Algorithm 2.

```
Algorithm 2 : CG with exact line search for computing the lowest eigenvalue of $T(\lambda)v = 0$
1. Choose $x_0 \neq 0$.
2. For $k = 0, 1, \ldots,$ until convergence, i.e., $\|x_k\|\|\nabla\rho(x_k)\| \leq \delta$
3.     If $k = 0$, set $p_0 = -\|x_0\|\nabla\rho(x_0)$;
       otherwise, set $\beta_k = \frac{\nabla\rho(x_k)^T\nabla\rho(x_k)\|x_k\|^2}{\nabla\rho(x_{k-1})^T\nabla\rho(x_{k-1})\|x_{k-1}\|^2}$, and $p_k = -\|x_k\|\nabla\rho(x_k) + \beta_k p_{k-1}$.
4.     Form $U_k = [x_k\ \ p_k]$, perform the Rayleigh-Ritz projection and solve the projected
       eigenproblem $U_k^T T(\nu)U_k w = 0$ for the lowest primitive Ritz pair $\left(\nu_1^{(k)}, w_1^{(k)}\right)$.
5.     Set $x_{k+1} = U_k\frac{w_1^{(k)}}{e_1^T w_1^{(k)}} = x_k + \frac{e_2^T w_1^{(k)}}{e_1^T w_1^{(k)}}p_k$, and normalize $x_{k+1}$ if necessary.
6. End For
```

Specifically, to find the optimal step size, we project $T(\cdot)$ onto $U_k = [x_k\ \ p_k]$, solve the projected 2×2 Hermitian eigenproblem $T_p^{(k)}(\nu)w \equiv U_k^T T(\nu)U_k w = 0$ for the lowest primitive Ritz pair $\left(\nu_1^{(k)}, w_1^{(k)}\right)$, and set $x_{k+1} = U_k w_1^{(k)}/(e_1^T w_1^{(k)}) = x_k + (e_2^T w_1^{(k)})/(e_1^T w_1^{(k)})p_k$ such that $\rho(x_{k+1}) = \nu_1^{(k)}$. By the Cauchy interlacing theorem, the two Ritz values $\nu_1^{(k)}, \nu_2^{(k)}$ satisfy the variational principle (2.1) or (2.2). Also, since $x_{k+1} = x_k + \alpha_k p_k = U_k[1\ \alpha_k]^T$, $\rho(x_{k+1})$ is the eigenvalue of the one-dimensional problem $\left(x_{k+1}^T T(\rho) x_{k+1}\right) v = \left([1\ \alpha_k] T_p(\rho)[1\ \alpha_k]^T\right) v = 0$. It follows from the interlacing theorem that for any α_k, $\nu_1^{(k)} \leq \rho(x_{k+1}) \leq \nu_2^{(k)}$; that is, $\nu_1^{(k)} = \min_{\alpha_k \in \mathbb{R}} \rho(x_k + \alpha_k p_k)$. Thus the optimal step size $\alpha_k^* = (e_2^T w_1^{(k)})/(e_1^T w_1^{(k)})$ is found by the Rayleigh-Ritz projection, which essentially realizes the exact line search.

Note that for Algorithm 2, we are also free to scale x_{k+1} in Step 5 by any nonzero factor, as is the case for Algorithm 1. In addition, we have the following properties of Algorithm 2.

Proposition 3.7. Algorithm 2 *generates* $\{p_k\}$ *and* $\{\nabla\rho(x_k)\}$ *satisfying*

$$\text{(i) } p_k^T \nabla\rho(x_{k+1}) = 0, \qquad \text{(ii) } p_k^T \nabla\rho(x_k) = -\|x_k\|\|\nabla\rho(x_k)\|^2,$$
$$\text{(iii) } \|p_k\|^2 = \|x_k\|^2\|\nabla\rho(x_k)\|^2 + \beta_k^2\|p_{k-1}\|^2, \qquad \text{(iv) } \|x_k\|\|\nabla\rho(x_k)\| \leq \|p_k\|.$$

Proof. For property (i), note that Algorithm 2 performs exact line search in the direction of p_k; that is, the optimal step size α_k is obtained by minimizing $\rho(x_k + \alpha p_k)$ with respect to α. Therefore, $\frac{\partial}{\partial\alpha}\rho(x_k + \alpha p_k)\big|_{\alpha=\alpha_k} = p_k^T \nabla\rho(x_k + \alpha_k p_k) = p_k^T \nabla\rho(x_{k+1}) = 0$.

In Step 3 of Algorithm 2, $p_k = -\|x_k\|\nabla\rho(x_k) + \beta_k p_{k-1}$. Taking the transpose of p_k and postmultiplying by $\nabla\rho(x_k)$, we have $p_k^T \nabla\rho(x_k) = -\|x_k\|\|\nabla\rho(x_k)\|^2$ (property (ii)). Property (iii) is obvious from Step 3 and the fact that $\nabla\rho(x_k)$ is orthogonal to p_{k-1}, and property (iv) follows from property (iii). Note that all properties hold independently of the choice of β_k. □

Since Algorithm 2 is a special variant of Algorithm 1, its global convergence can be established directly from Theorem 3.6. We summarize the result as follows.

Theorem 3.8. *Under the same assumptions as in* Theorem 3.6*, there is a* λ_ℓ $(1 \leq \ell \leq n)$ *such that* Algorithm 2 *generates a sequence of iterates* $\{x_k\}$ *satisfying* $\lim_{k\to\infty} \rho(x_k) = \lambda_\ell$, *and* $\lim_{k\to\infty} \angle\left(x_k, \operatorname{null} T(\lambda_\ell)\right) = 0$.

Remark. In practice, Algorithm 2 almost surely converges to the lowest eigenvalue. This is because the eigenvectors corresponding to all other eigenvalues are saddle points of the Rayleigh functional $\rho(\cdot)$. Unless x_k is close to a saddle point and this saddle point is indeed the minimizer along the search direction p_k, it is unlikely to prevent Algorithm 2 from moving towards new iterates near eigenvectors corresponding to lower eigenvalues. Note in particular that if x_0 is such that $\rho(x_0)$ is smaller than the second lowest eigenvalue, then Algorithm 2 converges to the lowest eigenvalue.

3.2. Asymptotic convergence.

In this section, we study the asymptotic convergence of CG for computing the lowest eigenvalue of the nonlinear Hermitian eigenproblem. Without loss of generality, we assume that $J = (a, b)$ is of positive type, on which the n eigenvalues satisfy the variational principle (2.1) or (2.2), and that λ_1 is simple, i.e., $\lambda_1 < \lambda_2 \leq \ldots \leq \lambda_n$. Let v_1 be the eigenvector associated with λ_1, x_0 be an approximation to v_1, and v_1 and x_0 are normalized such that $v_1 T'(\lambda_1) v_1 = 1$ and $x_0 T'(\rho_0) x_0 = 1$, where $\rho_0 = \rho(x_0)$.

The main point we want to show is that the asymptotic behavior of CG for computing (λ_1, v_1) is very similar to CG for solving a positive semi-definite linear system. Therefore, the well-known convergence rate estimate of CG as a linear solver is likely to be descriptive of CG performing Rayleigh functional minimization. The intuition for this observation has been discussed in [2, Chapter 12.3] for linear Hermitian eigenproblems $Av = \lambda Bv$.

To gain a better understanding of the connections between the two settings in which CG is used, it is natural to note that a general nonlinear scalar-valued smooth function can be approximated by a quadratic function with positive definite Hessian near the desired minimizer. Following this intuition, we have $\rho(x) \approx Q_\alpha(\Delta x) = \frac{1}{2}\Delta x^T \nabla^2\rho(x_0)\Delta x + \nabla\rho(x_0)^T \Delta x + \rho(x_0)$ (second order Taylor expansion of $\rho(x)$ at x_0), where $\Delta x = x - x_0$. The minimizer of $Q_\alpha(\cdot)$, $x^* = x_0 + \Delta x^*$, satisfies $\nabla^2\rho(x_0)\Delta x^* = -\nabla\rho(x_0)$. However, from the expressions (3.1) and

(3.2) for $\nabla\rho$ and $\nabla^2\rho$, we see that the solution for this equation is $\Delta x^* = -x_0$, and thus $x^* = x_0 + \Delta x^* = 0$, which does not yield any improvement in the eigenvector direction.

This difficulty can be overcome by approximating the Hessian $\nabla^2\rho(x_0)$ and the gradient $\nabla\rho(x_0)$ on an $(n-1)$-dimensional space. The motivation for such an approximation is to enforce a meaningful correction Δx by requiring that Δx not be very close to x_0 in direction. Note that this is precisely the idea for constructing the Jacobi-Davidson correction equation [27, 28]. In fact, an example of such a new quadratic function is

$$\begin{aligned} Q_\beta(\Delta x) &= \frac{1}{2}\Delta x^T P_0^T\left(\nabla^2\rho(x_0) + \frac{x_0^T T''(\rho_0)x_0}{x_0^T T'(\rho_0)x_0}\nabla\rho(x_0)\nabla\rho(x_0)^T\right)P_0\Delta x + \nabla\rho(x_0)^T P_0\Delta x + \rho_0 \\ &= -\frac{2}{x_0^T T'(\rho_0)x_0}\left(\frac{1}{2}\Delta x^T P_0^T T(\rho_0)P_0\Delta x + x_0^T T(\rho_0)P_0\Delta x\right) + \rho_0, \end{aligned}$$

where $P_0 = I - \frac{x_0 x_0^T T'(\rho_0)}{x_0^T T'(\rho_0)x_0}$ is the oblique projection onto span $\left\{(T'(\rho_0)x_0)^\perp\right\}$ along span$\{x_0\}$. The major difference between $Q_\alpha(\cdot)$ and $Q_\beta(\cdot)$ is that $Q_\beta(\cdot)$ uses (i) the oblique projector P_0 to filter out the component of Δx that lies in span$\{x_0\}$, and (ii) the Hessian $\nabla^2\rho(x_0)$ with a small perturbation $\frac{x_0^T T''(\rho_0)x_0}{x_0^T T'(\rho_0)x_0}\nabla\rho(x_0)\nabla\rho(x_0)^T = \mathcal{O}(\sin^2\theta_0)$ (see the estimate of $\|\nabla\rho(x)\|$ in Proposition 3.2). Therefore, $Q_\beta(\Delta x)$ is a good approximation to $\rho(x)$ near x_0 for all $\Delta x = x - x_0$ lying in span $\left\{(T'(\rho_0)x_0)^\perp\right\}$. Thus the minimizer for $Q_\beta(\cdot)$ can be obtained by solving $\nabla Q_\beta(\Delta x) = 0$, i.e.,

$$H_0\Delta x \equiv -\left(I - \frac{T'(\rho_0)x_0x_0^T}{x_0^T T'(\rho_0)x_0}\right)T(\rho_0)\left(I - \frac{x_0x_0^T T'(\rho_0)}{x_0^T T'(\rho_0)x_0}\right)\Delta x = T(\rho_0)x_0. \qquad (3.14)$$

Note that (3.14) is a Jacobi-Davidson correction equation, to which the exact solution is

$$\Delta x^* = -x_0 + \frac{x_0^T T'(\rho_0)x_0}{x_0^T T'(\rho_0)T(\rho_0)^{-1}T'(\rho_0)x_0}T(\rho_0)^{-1}T'(\rho_0)x_0. \qquad (3.15)$$

That is, the minimizer of $Q_\beta(\cdot)$ is $x^* = x_0 + \Delta x^* = T(\rho_0)^{-1}T'(\rho_0)x_0$ up to a scaling factor, which is exactly the new iterate computed by one step of Rayleigh functional iteration (RFI) starting with x_0 [25, Chapter 4.3]. For Hermitian eigenproblems, the local convergence of RFI is cubic; that is, $\sin\angle(x^*, v_1) = \mathcal{O}\left(\sin^3\angle(x_0, v_1)\right)$ for all $\angle(x_0, v_1)$ sufficiently small. In our setting, equation (3.14) is solved by CG approximately, which generates a sequence of iterates $\Delta x_1, \Delta x_2, \ldots$ that converges towards the exact solution Δx^* (3.15).

Our main interest is to explore the connection between the behaviors of CG for the minimization of $Q_\beta(\cdot)$ and $\rho(\cdot)$. To this end, we first show that the coefficient matrix H_0 of (3.14) is positive semi-definite if $\angle(x_0, v_1)$ is sufficiently small. Then we apply one step of CG to minimize $Q_\beta(\cdot)$ and compare its behavior with one step of CG minimizing $\rho(\cdot)$.

To show the semi-definiteness of $H_0 = -\left(I - \frac{T'(\rho_0)x_0x_0^T}{x_0^T T'(\rho_0)x_0}\right)T(\rho_0)\left(I - \frac{x_0x_0^T T'(\rho_0)}{x_0^T T'(\rho_0)x_0}\right)$, note that $\mathbb{R}^n = \text{span}\{x_0\} \oplus \text{span}\left\{(T'(\rho_0)x_0)^\perp\right\}$. Since $x_0^T H_0 x_0 = 0$, it is sufficient to show that $x^T H_0 x > 0$ for any nonzero $x \in \text{span}\left\{(T'(\rho_0)x_0)^\perp\right\}$. Considering $x_0 \not\perp T'(\rho_0)x_0$ (in fact $x_0^T T'(\rho_0)x_0 = 1$ by assumption), we define $\varphi_{\min} = \angle\left(x_0, \text{span}\left\{(T'(\rho_0)x_0)^\perp\right\}\right)$ such that for any nonzero $x \in \text{span}\left\{(T'(\rho_0)x_0)^\perp\right\}$, $0 < \varphi_{\min} \le \varphi = \angle(x_0, x)$. Suppose that x_0 is sufficiently close to v_1 in direction, i.e., $\theta_0 = \angle(x_0, v_1) \ll \varphi_{\min}$, and thus $|\rho_0 - \lambda_1| = \mathcal{O}(\tan^2\theta_0)$ [26]. It follows that for any $x \in \text{span}\left\{(T'(\rho_0)x_0)^\perp\right\}$, $0 < \varphi_{\min} \approx \varphi_{\min} - \theta_0 \le \angle(x, x_0) - \angle(x_0, v_1) \le \angle(x, v_1)$, and therefore

$$\begin{aligned} |\rho_0 - \lambda_1| &= \mathcal{O}(\tan^2\theta_0) \ll \mathcal{O}(\tan^2\varphi_{\min}) \approx \mathcal{O}(\tan^2(\varphi_{\min} - \theta_0)) \\ &\le \mathcal{O}(\tan^2\angle(x, v_1)) = |\rho(x) - \lambda_1|. \end{aligned}$$

Since $\rho(x) \ge \lambda_1$, we must have $\rho(x) > \rho_0 > \lambda_1$, and consequently $x^T H_0 x = -x^T T(\rho_0) x = -\frac{(\rho_0-\rho(x))x^T T(\rho_0)x}{\rho_0-\rho(x)} > 0$ as J is of positive type. Thus H_0 is positive semi-definite.

To see how CG performs in one step for the minimization of $Q_\beta(\cdot)$, note from (3.14) that

$$H_0 = -\left(T(\rho_0) + \frac{T'(\rho_0)x_0}{2}\nabla\rho(x_0) + \nabla\rho(x_0)\frac{x_0^T T'(\rho_0)}{2}\right) = -T(\rho_0) + \mathcal{O}(\sin\theta_0)$$

is a small perturbation of $-T(\rho_0)$ if $\theta_0 = \angle(x_0, v_1)$ is small. Let x_0 be the current CG iterate, and p_0 be the *normalized* search direction to minimize $Q_\beta(\cdot)$. Assume that $\angle(x_0, p_0) \ge \omega_0$ for some $\omega_0 \gg \theta_0 > 0$. Following the idea with which $\rho(x) > \rho_0$ is established in the proof of the semi-definiteness of H_0, one can show similarly that $\rho(p_0) > \rho_0$ and $\rho(p_0)$ is bounded away from ρ_0. It is then not difficult to see that $p_0^T T(\rho_0) p_0 = \frac{(\rho_0-\rho(p_0))p_0^T T(\rho_0)p_0}{\rho_0-\rho(p_0)}$ is negative and is bounded away from zero. Let the new CG iterate be $x_1 = x_0 + \alpha_0 p_0$. From (3.14), the exact line search condition for this CG step is $T(\rho_0)x_0 - H_0(\alpha_0 p_0) \perp p_0$, and it follows that

$$\begin{aligned}\alpha_0\|p_0\| &= \frac{p_0^T T(\rho_0)x_0\|p_0\|}{p_0^T H_0 p_0} = \frac{p_0^T T(\rho_0)x_0\|p_0\|}{p_0^T\left(-T(\rho_0) + \mathcal{O}(\sin\theta_0)\right)p_0} \\ &= -\frac{p_0^T T(\rho_0)x_0\|p_0\|}{p_0^T T(\rho_0)p_0 + \mathcal{O}(\sin\theta_0)} = -\frac{p_0^T T(\rho_0)x_0\|p_0\|}{p_0^T T(\rho_0)p_0}\left(1 + \mathcal{O}(\sin\theta_0)\right). \qquad (3.16)\end{aligned}$$

From (3.16), we see that the step size $\alpha_0\|p_0\| = \|x_1 - x_0\|$ is independent of the scaling of p_0, and it is proportional to the eigenresidual norm $\|T(\rho_0)x_0\| = \|x_0\|\mathcal{O}(\sin\theta_0)$; see the proof of Proposition 3.2. Therefore, as long as $\|x_0\|$ remains bounded, the search step size $\alpha_0\|p_0\| \to 0$ as the initial iterate x_0 approaches v_1 in direction, i.e., $\theta_0 = \angle(x_0, v_1) \to 0$.

Finally, we explore the behavior of CG in one step for the minimization of $\rho(\cdot)$, with the same initial iterate x_0 and search direction p_0 as discussed above. The exact line search condition in this case is $p_0^T\nabla\rho(x_1) = 0$, for which the Taylor expansion at x_0 is

$$\begin{aligned}0 = -\frac{x_1^T T'(\rho_1)x_1}{2}p_0^T\nabla\rho(x_1) &= p_0^T T\left(\rho(x_0+\alpha_0 p_0)\right)(x_0+\alpha_0 p_0) \qquad (3.17)\\ = p_0^T T\left(\rho_0 + \nabla\rho(x_0)^T(\alpha_0 p_0) + \frac{1}{2}(\alpha_0 p_0)^T\nabla^2\rho(x_0)(\alpha_0 p_0) + \mathcal{O}\left((\alpha_0\|p_0\|)^3\right)\right)&(x_0+\alpha_0 p_0)\\ = p_0^T T(\rho_0)x_0 + \left(p_0^T T'(\rho_0)x_0\right)\nabla\rho(x_0)^T(\alpha_0 p_0) + p_0^T T(\rho_0)(\alpha_0 p_0) + &\\ \frac{1}{2}(\alpha_0 p_0)^T\left(\left(p_0^T T'(\rho_0)x_0\right)\nabla^2\rho(x_0) + (p_0^T T''(\rho_0)x_0)\nabla\rho(x_0)\nabla\rho(x_0)^T\right)(\alpha_0 p_0) + &\\ p_0^T T'(\rho_0)(\alpha_0 p_0)\nabla\rho(x_0)^T(\alpha_0 p_0) + \mathcal{O}\left((\alpha_0\|p_0\|)^3\right)&\\ = \|p_0\|\left(C_2(\alpha_0\|p_0\|)^2 + C_1(\alpha_0\|p_0\|) + C_0\right) + \mathcal{O}\left((\alpha_0\|p_0\|)^3\right),&\end{aligned}$$

where

$$\begin{aligned}C_0 &= \frac{p_0^T T(\rho_0)x_0}{\|p_0\|} = \|x_0\|\mathcal{O}(\sin\theta_0), \qquad (3.18)\\ C_1 &= \frac{p_0^T T(\rho_0)p_0}{\|p_0\|^2} + \frac{p_0^T T'(\rho_0)x_0\nabla\rho(x_0)^T p_0}{\|p_0\|^2} = \frac{p_0^T T(\rho_0)p_0}{\|p_0\|^2} + \mathcal{O}(\sin\theta_0), \qquad \text{and}\\ C_2 &= \frac{\left(p_0^T T'(\rho_0)x_0\right)p_0^T\nabla^2\rho(x_0)p_0}{2\|p_0\|^3} + \frac{\left(p_0^T T''(\rho_0)x_0\right)p_0^T\nabla\rho(x_0)\nabla\rho(x_0)^T p_0}{2\|p_0\|^3} + \\ &\quad \frac{p_0^T T'(\rho_0)p_0\nabla\rho(x_0)^T p_0}{\|p_0\|^3} = \frac{\left(p_0^T T'(\rho_0)x_0\right)p_0^T\nabla^2\rho(x_0)p_0}{2\|p_0\|^3} + \frac{1}{\|x_0\|}\mathcal{O}(\sin\theta_0).\end{aligned}$$

Note that the coefficients $C_0 = \|x_0\|\mathcal{O}(\sin\theta_0)$, $C_1 = \mathcal{O}(1)$ and $C_2 = \mathcal{O}(1)$, and they do not depend on the scaling of p_0. Neglecting the cubic term of $\alpha_0\|p_0\|$ in (3.17), and using the root formula for quadratic equations, we have

$$\begin{aligned}\alpha_0\|p_0\| &= \frac{-C_1 + \sqrt{C_1^2 - 4C_0C_2}}{2C_2} = \frac{-C_1 + C_1\left(1 - \frac{2C_0C_2}{C_1^2} + \mathcal{O}\left(\frac{C_0^2C_2^2}{C_1^4}\right)\right)}{2C_2} \\ &= -\frac{C_0}{C_1}\left(1 + \mathcal{O}\left(\frac{C_0C_2}{C_1^2}\right)\right) = -\frac{p_0^T T(\rho_0)x_0\|p_0\|}{p_0^T T(\rho_0)p_0}\left(1 + \mathcal{O}(\sin\theta_0)\right)\end{aligned} \tag{3.19}$$

for all θ_0 sufficiently small. Therefore the dominant terms of the search step size shown in (3.16) and (3.19) are identically $-\frac{p_0^T T(\rho_0)x_0\|p_0\|}{p_0^T T(\rho_0)p_0}$.

In other words, given the same initial iterate $x_0 \approx v_1$ and search direction p_0 for which $\angle(p_0, x_0) \geq \omega_0 > 0$, CG with exact line search yields an almost identical new iterate x_1 for the minimization of the quadratic function $Q_\beta(\cdot)$ and the Rayleigh functional $\rho(\cdot)$. Our observation suggests that CG for computing the lowest eigenvalue could potentially converge as rapidly as CG for solving the semi-definite system (3.14). In particular, CG is expected to converge much more quickly than the steepest descent method for solving Hermitian eigenproblems. Such a superiority has been observed extensively in numerical experiments.

4. Variants of PCG-type methods for computing multiple eigenpairs.

In this section, we study several variants of the preconditioned conjugate gradient (PCG) method for computing multiple extreme eigenvalues of nonlinear Hermitian eigenproblems satisfying the variational principle (2.1) or (2.2). Without loss of generality, assume that $J = (a, b)$ is of positive type on which the lowest m eigenvalues $\{\lambda_i\}_{i=1}^m$ are desired. We discuss single-vector PCG with the Fletcher-Reeves formula, the locally optimal PCG (LOPCG), and their block versions BPCG, and LOBPCG. These methods are developed as direct extensions of their well-known counterparts for linear Hermitian eigenproblems (see [3, 21, 22] and references therein) to the nonlinear setting.

4.1. Single-vector methods.

We study a single-vector PCG as given in Algorithm 3. This method is obtained by incorporating preconditioning and deflation to Algorithm 2. We construct a symmetric positive definite (SPD) preconditioner $M \approx -T(\sigma)$ (where $\sigma < \lambda_1$) and compute the scaled preconditioned gradient $\|x_k\|M^{-1}\nabla\rho(x_k)$ to form β_k and the search direction p_k in Step 5 of the algorithm. The way CG is preconditioned here is the same as in the general setting of solving unconstrained nonlinear optimizations; see, e.g., [18, 23]. In particular, $M \approx -T(\sigma)$ is an approximation to the Hessian $\nabla^2\rho(v_1) = -\frac{2}{v_1^T T'(\lambda_1)v_1}T(\lambda_1)$ (up to a scaling factor) at the desired minimizer $x^* = v_1$ of the Rayleigh functional. PCG invokes the preconditioner M directly, without assuming the availability of any factorizations of M.

To compute multiple eigenpairs, deflation is needed to avoid repeated convergence. For linear Hermitian problems $Av = \lambda Bv$ with an SPD B, deflation is done by the Gram-Schmidt orthogonalization against converged eigenvectors with respect to the B-inner product. For nonlinear Hermitian problems, the eigenvectors $\{v_i\}_{i=1}^n$ are pairwise orthogonal with respect to the scalar product $[\cdot,\cdot]$ defined in (2.3). One would consider performing orthogonalization with respect to this scalar product, but this is not viable as $[\cdot,\cdot]$ is generally *not bilinear*. Therefore, a Gram-Schimidt step orthogonalizing a vector u against v_i, i.e., $\hat{u} = u - \frac{[u,v_i]}{[v_i,v_i]}v_i$, does not lead to $[\hat{u}, v_i] = 0$, because $\left[u - \frac{[u,v_i]}{[v_i,v_i]}v_i, v_i\right] \neq [u, v_i] - \frac{[u,v_i]}{[v_i,v_i]}[v_i, v_i] = 0$. In addition, even if we could generate a set of vectors orthogonal to all converged eigenvectors $\{v_i\}_{i=1}^j$ with respect to $[\cdot,\cdot]$, a linear combination of these vectors may not be orthogonal to any v_i $(1 \leq i \leq j)$. Consequently, the orthogonalization-based 'hard deflation' cannot be done for nonlinear eigenproblems as for linear problems. Instead, we include all converged eigenvectors in the space for the Rayleigh-Ritz projection and then select unconverged Ritz pairs as approximations for new eigenpairs. Such a strategy is called ‘soft deflation’, and it is also used to expand invariant pairs [4] in a robust manner for solving general (non-Hermitian) nonlinear eigenproblems of the form $T(\lambda)v = 0$ [12].

```
Algorithm 3 : PCG for computing the lowest $m$ eigenpairs $\{(\lambda_i, v_i)\}_{i=1}^{m}$ of $T(\lambda)v = 0$
1. Construct an SPD preconditioner $M \approx -T(\sigma)$ where $\sigma < \lambda_1$. Set $j = 0$.
2. While $j < m$, do
3.    Choose a nonzero normalized vector $x_0 \notin \mathrm{span}\{v_1, \ldots, v_j\}$.
4.    For $k = 0, 1, \ldots$, until the convergence of the $(j+1)$st eigenpair, i.e., $\|x_k\|\|\nabla\rho(x_k)\| \leq \delta$
5.       If $k = 0$, set $p_k = -\|x_k\| M^{-1}\nabla\rho(x_k)$;
         otherwise, set $\beta_k = \frac{\|x_k\|^2}{\|x_{k-1}\|^2}\frac{\nabla\rho(x_k)^T M^{-1}\nabla\rho(x_k)}{\nabla\rho(x_{k-1})^T M^{-1}\nabla\rho(x_{k-1})}$, $p_k = -\|x_k\| M^{-1}\nabla\rho(x_k) + \beta_k p_{k-1}$.
6.       Form $U_k = [v_1 \ldots v_j \;\; x_k \;\; p_k] \in \mathbb{C}^{n\times(j+2)}$, and normalize each column of $U_k$
         (the search direction vector $p_k$ itself remains unnormalized).
7.       Perform the Rayleigh-Ritz projection and solve the projected eigenproblem
         $U_k^T T(\nu) U_k w = 0$ for the $(j+1)$st lowest primitive Ritz pair $\big(\nu_{j+1}^{(k)}, w_{j+1}^{(k)}\big)$.
8.       Set $x_{k+1} = U_k \frac{w_{j+1}^{(k)}}{e_{j+1}^T w_{j+1}^{(k)}}$, and normalize $x_{k+1}$.
9.    End For
10.  Set $(\lambda_{j+1}, v_{j+1}) = (\rho(x_k), x_k)$, and $j = j + 1$.
11. End While
```

With soft deflation, Algorithm 3 (PCG) proceeds as follows. It computes the m lowest eigenpairs sequentially, one eigenpair at a time. Assume that the lowest j eigenpairs have converged. To approximate the $(j+1)$st eigenpair, PCG starts with an initial iterate x_0 in Step 3. In each iteration, it computes the search direction p_k in Step 5, forms the projection space U_k including all converged eigenvectors in Step 6, then performs the Rayleigh-Ritz projection and solves for the $(j+1)$st lowest primitive Ritz pair $\big(\nu_{j+1}^{(k)}, w_{j+1}^{(k)}\big)$ in Step 7, and finally sets x_{k+1} as the scaled $(j+1)$st lowest Ritz pair $U_k w_{j+1}^{(k)}/(e_{j+1}^T w_{j+1}^{(k)})$ in Step 8.

```
Algorithm 4 : LOPCG for computing the lowest $m$ eigenpairs $\{(\lambda_i, v_i)\}_{i=1}^{m}$ of $T(\lambda)v = 0$
1. Choose a SPD preconditioner $M \approx -T(\sigma)$ where $\sigma < \lambda_1$. Set $j = 0$.
2. While $j < m$, do
3.    Choose a nonzero normalized vector $x_0 \notin \mathrm{span}\{v_1, \ldots, v_j\}$.
4.    For $k = 0, 1, \ldots$, until the convergence of the $(j+1)$st eigenpair, i.e., $\|x_k\|\|\nabla\rho(x_k)\| \leq \delta$
5.       Set $g_k = -\|x_k\| M^{-1}\nabla\rho(x_k)$.
6.       If $k = 0$, form $U_k = [v_1 \ldots v_j \;\; x_k \;\; g_k]$; otherwise, form $U_k = [v_1 \ldots v_j \;\; x_k \;\; g_k \;\; p_{k-1}]$.
         Normalize each column of $U_k$, and set $r$ = number of columns of $U_k$.
7.       Perform the Rayleigh-Ritz projection and solve the projected eigenproblem
         $U_k^T T(\nu) U_k w = 0$ for the $(j+1)$st lowest primitive Ritz pair $\big(\nu_{j+1}^{(k)}, w_{j+1}^{(k)}\big)$.
8.       Set $p_k = \sum_{i=1, i\neq j+1}^{r} \frac{e_i^T w_{j+1}^{(k)}}{e_{j+1}^T w_{j+1}^{(k)}} U_k e_i$, $x_{k+1} = U_k \frac{w_{j+1}^{(k)}}{e_{j+1}^T w_{j+1}^{(k)}} = x_k + p_k$, and normalize $x_{k+1}$.
9.    End For
10.  Set $(\lambda_{j+1}, v_{j+1}) = (\rho(x_k), x_k)$, and $j = j + 1$.
11. End While
```

The convergence of PCG may be enhanced by enlarging the space for projection. To see this improvement, assume that the lowest eigenvalue λ_1 is of interest. The Rayleigh-Ritz projection is now performed onto a three-dimensional space $U_k^{LOPCG} = [x_k \;\; g_k \;\; p_{k-1}]$, where $g_k = -\|x_k\| M^{-1}\nabla\rho(x_k)$ is the preconditioned gradient, and p_{k-1} is the search direction in the previous PCG step. This idea was proposed in [20] and discussed in detail in [21]. The new algorithm is referred to as the locally optimal preconditioned conjugate gradient (LOPCG) method. The convergence rate of PCG is improved thanks to the enlarged space, since

$$\mathrm{range}(U_k^{PCG}) = \mathrm{span}\{x_k,\, p_k\} \subset \mathrm{span}\{x_k,\, g_k,\, p_{k-1}\} = \mathrm{range}(U_k^{LOPCG})$$

(see Step 5 of Algorithm 3), and thus

$$\rho(x_{k+1}^{LOPCG}) = \min_{x \in \mathrm{range}(U_k^{LOPCG})} \rho(x) \leq \min_{x \in \mathrm{range}(U_k^{PCG})} \rho(x) = \rho(x_{k+1}^{PCG})$$

by the Cauchy interlacing theorem. In fact, LOPCG performs an exact search for the optimal new iterate x_{k+1} along the two-dimensional space $\mathrm{span}\{g_k,\, p_{k-1}\}$, which contains the traditional search direction $p_k = g_k + \beta_k p_{k-1}$ independently of the value of β_k. Such an exact search is almost impossible to achieve in the general setting of unconstrained optimization, but it can be done easily by Rayleigh-Ritz projection, thanks to the interlacing theorem.

One need to note that the superiority of LOPCG over PCG is based on the assumption that both methods have the same previous search direction p_{k-1} and the current iterate x_k. Such an assumption actually holds only in the first iteration. In practice, LOPCG may take more iterations to converge than PCG, as shown in our numerical experiments in Section 5.

4.2. Block methods. Single-vector PCG and LOPCG can be extended to block versions, which are referred to as BPCG and LOBPCG and described in detail in Algorithms 5 and 6, respectively. Here, we assume that the initial block size of the two algorithms equals the number of desired lowest eigenpairs m. In practice, we may choose a block size slightly larger than m to accelerate convergence.

```
Algorithm 5 : BPCG for computing the lowest $m$ eigenpairs $\{(\lambda_i, v_i)\}_{i=1}^m$ of $T(\lambda)v = 0$
1. Choose an SPD preconditioner $M \approx -T(\sigma)$ where $\sigma < \lambda_1$, $X_0 \in \mathbb{R}^{n\times m}$ of rank $m$. Set $j = 0$.
2. For $k = 0, 1, \ldots,$ until convergence, i.e., $j = m$
3.    If $k = 0$, set $P_k \in \mathbb{R}^{n\times m}$ s.t. $P_k e_\ell = -\|X_k e_\ell\| M^{-1}\nabla\rho(X_k e_\ell)$ $(1 \leq \ell \leq m)$; otherwise,
      set $B_k \in \mathbb{R}^{m\times m}$ s.t. $(B_k)_{\ell\ell} = \frac{\|X_k e_\ell\|^2}{\|X_{k-1} e_\ell\|^2} \frac{\nabla\rho(X_k e_\ell)^T M^{-1} \nabla\rho(X_k e_\ell)}{\nabla\rho(X_{k-1} e_\ell)^T M^{-1} \nabla\rho(X_{k-1} e_\ell)}$ $(j+1 \leq \ell \leq m)$,
      and set $P_k \in \mathbb{R}^{n\times m}$ s.t. $P_k e_\ell = -\|X_k e_\ell\| M^{-1}\nabla\rho(X_k e_\ell) + (B_k)_{\ell\ell} P_{k-1} e_\ell$ $(j+1 \leq \ell \leq m)$.
4.    Form $U_k = [v_1 \ldots v_j \;\; X_k e_{j+1} \, \ldots \, X_k e_m \;\; P_k e_{j+1} \, \ldots \, P_k e_m] \in \mathbb{C}^{n\times(2m-j)}$, and normalize
      each column of $U_k$ (the search direction $P_k$ itself remains unnormalized).
5.    Perform Rayleigh-Ritz projection and solve the projected eigenproblem $U_k^T T(\nu) U_k w = 0$
      for the $(j+1)$st through $m$th lowest primitive Ritz pair $\left(\nu_{j+1}^{(k)}, w_{j+1}^{(k)}\right), \ldots, \left(\nu_m^{(k)}, w_m^{(k)}\right)$.
6.    Update $X_{k+1}$ s.t. $X_{k+1} e_\ell = U_k \frac{w_\ell^{(k)}}{e_\ell^T w_\ell^{(k)}}$ $(j+1 \leq \ell \leq m)$; normalize each column of $X_{k+1}$.
7.    Find the largest $\ell$ $(j+1 \leq \ell \leq m)$ s.t. for each $q$, $j+1 \leq q \leq \ell$, $\|\nabla\rho(X_{k+1} e_q)\| \leq \frac{\delta}{\|X_{k+1} e_q\|}$.
      If such $\ell$ exists, set $v_q = X_{k+1} e_q$, $\lambda_q = \rho(v_q)$ for $j+1 \leq q \leq \ell$, and set $j = \ell$.
8. End For
```

The framework of BPCG is very similar to PCG. Briefly, BPCG keeps a block iterate X_k of m columns that columnwise approximates the lowest eigenvectors $\{v_i\}_{i=1}^m$. It performs the Rayleigh-Ritz projection using the space incorporating the converged eigenvectors and the two-dimensional projection spaces generated in PCG for each unconverged column. BPCG then updates the active columns of the new iterate X_{k+1} as the lowest unconverged Ritz vectors. We check the convergence for each active column, and deflate converged columns as the algorithm proceeds, so that the block size decreases as more eigenpairs converge. BPCG with such a deflation strategy is widely used, and is also referred to as the block deflation-accelerated (preconditioned) conjugate gradient (BDACG) for linear eigenproblems [3].

```
Algorithm 6 : LOBPCG for computing the lowest $m$ eigenpairs $\{(\lambda_i, v_i)\}_{i=1}^m$ of $T(\lambda)v = 0$
1. Choose an SPD preconditioner $M \approx -T(\sigma)$ where $\sigma < \lambda_1$, $X_0 \in \mathbb{R}^{n\times m}$ of rank $m$. Set $j = 0$.
2. For $k = 0, 1, \ldots,$ until convergence, i.e., $j = m$
3.     Set $G_k \in \mathbb{R}^{n\times m}$ s.t. $G_k e_\ell = -\|X_k e_\ell\| M^{-1}\nabla\rho(X_k e_\ell)$ $(j+1 \le \ell \le m)$.
4.     If $k = 0$, form $U_k = [X_k\ G_k] \in \mathbb{R}^{n\times 2m}$; otherwise form
       $U_k = [v_1 \ldots v_j\ \ X_k e_{j+1}\ \ldots\ X_k e_m\ \ G_k e_{j+1}\ \ldots\ G_k e_m\ \ P_{k-1} e_{j+1}\ \ldots\ P_{k-1} e_m] \in \mathbb{R}^{n\times(3m-2j)}$.
       Normalize each column of $U_k$, and set $r$ = number of columns of $U_k$.
5.     Perform Rayleigh-Ritz projection and solve the projected eigenproblem $U_k^T T(\nu) U_k w = 0$
       for the $(j+1)$st through $m$th lowest primitive Ritz pair $\left(\nu_{j+1}^{(k)}, w_{j+1}^{(k)}\right), \ldots, \left(\nu_m^{(k)}, w_m^{(k)}\right)$.
6.     Update $P_k$ s.t. $P_k e_\ell = \sum_{i=m+1}^{r}(e_i^T w_\ell^{(k)}) U_k e_i$ $(j+1 \le \ell \le m)$.
7.     Update $X_{k+1}$ s.t. $X_{k+1} e_\ell = U_k w_\ell^{(k)} = \sum_{i=1}^{m}(e_i^T w_\ell^{(k)}) U_k e_i + P_k e_\ell$ $(j+1 \le \ell \le m)$;
       normalize each column of $X_{k+1}$.
8.     Find the largest $\ell$ $(j+1 \le \ell \le m)$ s.t. for each $q$, $j+1 \le q \le \ell$, $\|\nabla\rho(X_{k+1}e_q)\| \le \frac{\delta}{\|X_{k+1}e_q\|}$.
       If such $\ell$ exists, set $v_q = X_{k+1}e_q$, $\lambda_q = \rho(v_q)$ for $j+1 \le q \le \ell$, and set $j = \ell$.
9. End For
```

Similar to the single-vector methods, the convergence of BPCG may be enhanced by performing the Rayleigh-Ritz projection onto an enlarged space incorporating the converged eigenvectors, the active columns of X_k, the scaled preconditioned gradient G_k and the previous search direction P_{k-1} (all in block forms). The enhanced algorithm is called LOBPCG (Algorithm 6), which also uses the progressive deflation as done for BPCG. Again, thanks to the enlarged projection spaces and the interlacing theorem, the convergence rate of LOBPCG is expected to be faster than that of BPCG. We shall see in Section 5 that LOBPCG is the most efficient PCG-type method for all test problems.

4.3. Variable and indefinite preconditioning. The performance of all the PCG methods we studied can be improved by using variable and indefinite preconditioning, if a relatively large number of low eigenvalues are desired and most of them are not very close to the lowest one. This preconditioning strategy has been used in [7] with a block preconditioned steepest descent method for solving high eigenpairs of linear Hermitian eigenproblems arising from the self-consistent field (SCF) iteration for discretized Kohn-Sham equations.

The seemingly untraditional use of indefinite preconditioning is warranted by the motivation of preconditioning for CG. For the solution of SPD linear systems $Ax = b$, the Hessian of the quadratic $\varphi(x) = \frac{1}{2}x^T Ax - b^T x$ is A, and thus it is natural to use SPD preconditioning to approximate the Hessian. By contrast, for Hermitian eigenproblems, we see from (3.2) that the Hessian of the Rayleigh function $\rho(\cdot)$ near an eigenvector v_ℓ corresponding to a high eigenvalue λ_ℓ $(\ell > 1)$ must be indefinite, and it is highly indefinite if $\ell \gg 1$, i.e., λ_ℓ that is deep in the interior of the spectrum is computed. Thus an indefinite preconditioner $M \approx T(\sigma)$ with $\sigma \approx \lambda_\ell$ is expected to provide more accurate approximation to the Hessian in this situation. In fact, by using deflation, and following our proof of the positive semidefiniteness of the Hessian $H_0 \approx \nabla^2\rho(x_0)$ defined in (3.14), it is reasonable to conjecture that the *restriction* of the Hessian on the orthogonal complement of the converged eigenspace is positive definite, and so is the restriction of the indefinite preconditioner. Such a definiteness of the restricted operator is called 'effective positive definiteness' in [7]. The performance improvement achieved by using variable preconditioning is shown in the next section.

5. Numerical Experiments. In this section, we illustrate the performance of PCG methods on a few nonlinear Hermitian eigenproblems satisfying the variational characterization. Our goal here is to compare different methods in the rate and robustness of convergence, as well as their arithmetic and memory cost. We consider single-vector and block versions of regular PCG and the locally optimal variants, with fixed and variable preconditioning, for computing $m = 10$, 20, 40 and 80 extreme eigenvalues of the test problems to a relative tolerance $\frac{\|T(\lambda_i)v_i\|}{\|T(\lambda_i)\|_F\|v_i\|} \leq 10^{-10}$ for $1 \leq i \leq m$. In each experiment, all methods use the same starting eigenvector approximations generated by the MATLAB function `randn(n,m)` initialized with the `RandStream` function using seed 1. The experiments were performed on an iMac desktop computer running Mac OS X 10.8.5, MATLAB R2012b, with a 2.9 GHz Intel Core i5 processor and 16GB 1600 MHz DDR3 memory.

name	type	dimension	interval	end of interest
wiresaw	quadratic	1024	$(0, 3250)$	lowest
*gen_hyper*2	quadratic	4096	$(-843, 0.3943)$	highest
sleeper	quadratic	16384	$(-16.33, -1.61)$	lowest
loaded_string	rational	10000	$(4.4, 1.2 \times 10^9)$	lowest
artificial	nonlinear	16129	$(-0.43, 3.34)$	lowest
pdde	nonlinear	39601	$(-20.87, 4.08)$	highest

Table 5.1
Description of the test problems

We choose three quadratic, one rational, and two truly nonlinear problems, and summarize them in Table 5.1. The quadratic and rational problems come from the NLEVP toolbox [5]. Specifically, the gyroscopic quadratic problem *wiresaw* of dimension 1024 arises in the vibration analysis of a wiresaw, which is constructed by the command `nlevp('wiresaw1',1024)`. Since all the eigenvalues are purely imaginary, this problem does not satisfy the variational principle (2.1) or (2.2) in its original form, but it can be easily transformed to a Hermitian eigenproblem satisfying the variational principle by substituting λ with $i\lambda$. The transformed problem have 1024 pairs of real eigenvalues $\{\lambda_i^{\pm}\}$, where $\lambda_i^- = -\lambda_i^+$, and $\{\lambda_i^-\}$ and $\{\lambda_i^+\}$ lie in $I_\ell = (-3250, 0)$ and $I_r = (0, 3250)$, respectively. The variational principle holds on each of the intervals. We are interested in computing the lowest eigenvalues close to the left boundary of the right interval I_r. The hyperbolic quadratic problem *gen_hyper*2 of dimension 4096 is obtained by the command `nlevp('gen_hyper2',ev,[eye(4096) eye(4096)])`, where `ev` is a vector consisting of the reciprocals of 8192 random numbers generated by `randn` function initialized with a zero seed. The eigenvalues of this quadratic problem are set to be the elements of `ev`, 4096 of which are distributed on the left interval $I_\ell = (-843, 0.3943)$, and the rest lie in the right interval $I_r = (0.3943, 20061)$. The variational principle is satisfied on both I_ℓ and I_r. We aim at solving the highest eigenvalues close to the right boundary of the left interval I_ℓ. The third quadratic problem *sleeper* of the form $T(\lambda) = A_0 + \lambda A_1 + \lambda^2 A_2$ of dimension 16384 describes the oscillations of a rail track resting on sleepers. We construct the problem by the command `nlevp('sleeper',128)`, and then change the matrix associated with the constant term from A_0 to $A_0 - 2I$, so that the modified problem satisfies the variational principle (2.1) or (2.2) on $(-16.33, -1.61)$. We compute the lowest eigenvalues close to the left boundary. The rational problem *loaded_string* of the form $T(\lambda) = A - \lambda B + \frac{\lambda}{\lambda-1}C$ of dimension 10000 arises in the finite element discretization of a boundary problem describing the eigenvibration of a string attached with a spring. It is generated by the command `nlevp('loaded_string',10000)`. We are interested in the lowest eigenvalues lying on the interval $(4.4, 1.2 \times 10^9)$ where the variational principle (2.1) or (2.2) holds. More details of these test problems can be found in [5].

Two truly nonlinear test problems are described as follows. One arises from the modeling of a partial delay differential equations (PDDE), and another one is artificially constructed. The PDDE is $u_t(x,t) = \Delta u(x,t) + a(x)u(x,t) + b(x)u(x,t-2)$ defined on $\Omega = [0,\pi] \times [0,\pi]$ for $t \geq 0$, where $a(x) = 8\sin(x_1)\sin(x_2)$ and $b(x) = 100|\sin(x_1 + x_2)|$, with Dirichlet boundary condition $u(x,t) = 0$ for all $x \in \partial\Omega$ and $t \geq 0$. Assume that the solution is in the form of

$u(x,t) = e^{\lambda t}v(x)$. Using the standard 5-point stencil finite difference approximation to the Laplacian operator on a 200×200 uniform grid, we have the algebraic eigenproblem of the form $T(\lambda) = \lambda I + (M + A) + e^{-2\lambda}B$, where the matrices M, A and B of order 39601 are the discretized form of the Laplacian operator, $a(x)$ and $b(x)$, respectively. The variational principle is satisfied on the interval $(-20.87, 4.08)$, and the highest eigenvalues are of interest. The artificial problem of order 16129 is of the form $T(\lambda) = -\sin\frac{\lambda}{5}A + \sqrt{\lambda+1}B + e^{-\lambda/\sqrt{\pi}}C$, where $A = I$, $B = \text{tridiag}[1; -2; 1]$, and C forms the standard 5-point stencil finite difference discretization of the Laplacian, based on a 128×128 uniform grid on the unit square, without scaling by the mesh size factor $\frac{1}{h^2} = 128^2$ as is done for the PDDE problem. We seek the lowest eigenvalues on $(-0.43, 3.34)$ where the variational principle (2.1) or (2.2) holds.

As we discussed in Section 4.3, the use of variable indefinite preconditioning may accelerate the convergence of PCG methods for computing eigenvalues not very close to the lowest or highest one. To illustrate such a performance improvement, we let the initial preconditioner be the LDL decomposition of $T(\sigma)$, where σ is the one of the two endpoints of the intervals described in Table 5.1 near which the eigenvalues are of interest. For the fixed preconditioning strategy, the initial preconditioner is used throughout the computation. To enable variable preconditioning, as the algorithms proceed, we update the preconditioner as the LDL decomposition of $T(\mu)$ once 10 or more new eigenpairs are found, where μ is the midpoint of the last two newly computed distinct eigenvalues.

The performance of PCG methods is presented in Table 5.2. We assess each method by the number of preconditioned matrix-vector products and the CPU time. Let us take the problem *wiresaw* as an example to see results. It takes single-vector PCG with fixed preconditioning 113 preconditioned matrix-vector products and 6.43 seconds to find the lowest 10 eigenvalues. Similarly, this method takes 319, 910 and 2682 preconditioned matrix-vector products, and 18.37, 54.95 and 189.93 seconds to compute the lowest 20, 40 and 80 eigenvalues, respectively. Using variable preconditioning, the algorithm runs considerably faster; in particular, it takes only 1412 preconditioned matrix-vector products and 116.43 seconds (about 40−50% less expensive) to compute the lowest 80 eigenvalues. Note that variable preconditioning is not enabled for the computation of 10 extreme eigenvalues, and therefore the performance is not given. In addition, some methods failed to converge for certain problems in the maximum number of iterations, and such a failure is marked as ∞. For example, with fixed preconditioning, single-vector and block methods did not find the 80 highest eigenvalues of the problem *gen_hyper*2 in $500 \times 80 = 40000$ and 500 iterations, respectively. Variable preconditioning is not enabled for the problem *sleeper*, because all the desired eigenvalues are very tightly clustered, and fixed preconditioning is sufficient to achieve rapid convergence.

Considering the overall performance of different PCG methods, we have the following observations, most of which are similar to those obtained in the setting of solving linear Hermitian eigenproblems.

- Block PCG methods are significantly more competitive than single-vector methods in arithmetic cost and CPU time for computing multiple eigenvalues. Consequently, one should choose block methods whenever memory permits. If the memory is not sufficient for block methods to compute all eigenvalues simultaneously, we can run block methods with a smaller block size to compute desired eigenvalues partitioned in groups sequentially. This strategy is widely used in practice to compute dozens to hundreds of eigenvalues of linear Hermitian eigenproblems.
- Using an excessively large block size m also hampers the arithmetic efficiency of block methods and could lead to a significant increase in CPU time. This is because solving the dense eigenproblem of order $2m$ (BPCG) or $3m$ (LOBPCG) arising from the Rayleigh-Ritz projection for m extreme Ritz values takes at least $\mathcal{O}(4m^3)$ or $\mathcal{O}(9m^3)$

m	10	20	40	80
wiresaw				
PCG (fixed pcd)	113 (6.43s)	319 (18.37s)	910 (54.95s)	2682 (189.93s)
PCG (variable pcd)	–	282 (18.10s)	639 (45.94s)	1412 (116.43s)
LOPCG (fixed pcd)	109 (6.12s)	309 (17.27s)	891 (53.70s)	2729 (190.78s)
LOPCG (variable pcd)	–	262 (16.65s)	590 (41.54s)	1317 (106.77s)
BPCG (fixed pcd)	82 (4.32s)	182 (8.94s)	413 (20.01s)	897 (44.67s)
BPCG (variable pcd)	–	176 (8.79s)	379 (19.27s)	774 (38.64s)
LOBPCG (fixed pcd)	65 (3.56s)	138 (6.79s)	298 (14.34s)	629 (31.26s)
LOBPCG (variable pcd)	–	132 (6.75s)	276 (13.98s)	558 (28.46s)
*gen_hyper*2				
PCG (fixed pcd)	1520 (3.64s)	6501 (20.79s)	∞	∞
PCG (variable pcd)	–	2052 (5.56s)	2834 (10.49s)	∞
LOPCG (fixed pcd)	1829 (4.23s)	10029 (34.11s)	∞	∞
LOPCG (variable pcd)	–	2162 (5.50s)	2584 (8.34s)	5137 (46.77s)
BPCG (fixed pcd)	644 (1.41s)	2074 (5.83s)	2708 (8.51s)	∞
BPCG (variable pcd)	–	1405 (3.08s)	2020 (5.19s)	9014 (36.08s)
LOBPCG (fixed pcd)	547 (1.15s)	895 (2.07s)	1607 (7.56s)	∞
LOBPCG (variable pcd)	–	801 (1.90s)	1101 (3.69s)	3826 (48.68s)
sleeper				
PCG (fixed pcd)	1382 (12.64s)	2284 (24.99s)	3395 (52.10s)	6041 (187.69s)
LOPCG (fixed pcd)	1806 (16.92s)	2990 (33.83s)	4482 (74.74s)	7029 (217.09s)
BPCG (fixed pcd)	1086 (8.91s)	1248 (11.62s)	1393 (15.92s)	1596 (24.12s)
LOBPCG (fixed pcd)	608 (5.08s)	651 (6.32s)	736 (9.24s)	917 (20.33s)
loaded_string				
PCG (fixed pcd)	85 (0.58s)	241 (1.98s)	711 (9.21s)	2566 (71.59s)
PCG (variable pcd)	–	207 (1.69s)	480 (6.03s)	1117 (30.12s)
LOPCG (fixed pcd)	83 (0.57s)	232 (1.90s)	697 (9.88s)	2181 (66.77s)
LOPCG (variable pcd)	–	203 (1.67s)	456 (5.51s)	1016 (29.15s)
BPCG (fixed pcd)	55 (0.36s)	128 (0.87s)	298 (2.31s)	663 (7.32s)
BPCG (variable pcd)	–	125 (0.85s)	278 (2.12s)	606 (6.48s)
LOBPCG (fixed pcd)	48 (0.32s)	109 (0.77s)	233 (1.88s)	508 (7.01s)
LOBPCG (variable pcd)	–	105 (0.75s)	212 (1.73s)	447 (6.37s)
pdde				
PCG (fixed pcd)	349 (39.84s)	796 (141.93s)	1892 (713.83s)	6193 (7075.84s)
PCG (variable pcd)	–	777 (140.57s)	1487 (514.87s)	2563 (2718.64s)
LOPCG (fixed pcd)	195 (24.38s)	515 (105.32s)	1343 (546.62s)	4261 (5544.76s)
LOPCG (variable pcd)	–	363 (68.78s)	693 (247.44s)	1371 (1512.81s)
BPCG (fixed pcd)	563 (27.79s)	1149 (48.72s)	1183 (101.41s)	2430 (355.40s)
BPCG (variable pcd)	–	1149 (48.72s)	1167 (93.81s)	2407 (333.74s)
LOBPCG (fixed pcd)	103 (7.16s)	199 (14.71s)	444 (75.74s)	833 (238.79s)
LOBPCG (variable pcd)	–	186 (13.27s)	371 (50.55s)	703 (176.81s)
artificial				
PCG (fixed pcd)	194 (8.20s)	471 (35.33s)	1484 (238.21s)	∞
PCG (variable pcd)	–	388 (29.11s)	1003 (162.23s)	∞
LOPCG (fixed pcd)	183 (7.54s)	449 (33.71s)	1259 (194.07s)	∞
LOPCG (variable pcd)	–	370 (27.73s)	755 (109.87s)	∞
BPCG (fixed pcd)	128 (2.77s)	295 (7.17s)	867 (31.27s)	∞
BPCG (variable pcd)	–	290 (7.10s)	834 (30.35s)	∞
LOBPCG (fixed pcd)	103 (1.96s)	211 (5.07s)	641 (27.30s)	∞
LOBPCG (variable pcd)	–	199 (4.83s)	529 (21.86s)	∞

Table 5.2
Performance of PCG methods

floating point operations. For a sufficiently large m, the total arithmetic cost of block methods is dominated by the cost for solving the projected eigenproblems. In this situation, though LOBPCG takes fewer preconditioned matrix-vector products than BPCG, it may require more CPU time, as is the case for computing the 80 highest

eigenvalues of the problem *gen_hyper*2.
- The use of variable preconditioning is most helpful to improve the efficiency of single-vector methods. This also indicates that block methods are more favorable, because their performance depends less on the quality of preconditioners, and thus preconditioning can be updated less frequently if many eigenvalues are desired. Moreover, variable preconditioning may also enhance the robustness of convergence. As shown for the problem *gen_hyper*2, the failure of convergence was fixed partially or completely by using variable preconditioning.
- LOBPCG is the most robust and efficient PCG-type method. In particular, it takes fewer preconditioned matrix-vector products than BPCG for all test problems. For single-vector methods, there is no such a prominent advantage of LOPCG over PCG. In fact, with fixed preconditioning, LOPCG converges slower than PCG for the two quadratic problems *gen_hyper*2 and *sleeper*, and it performs almost as well as PCG for the problem *wiresaw*. As we discussed, choosing a modest block size is necessary for the locally optimal variants to achieve optimal efficiency.

6. Conclusion. We studied PCG methods for solving extreme eigenvalues of large-scale nonlinear Hermitian eigenproblems of the form $T(\lambda)v = 0$ that admit a variational characterization of eigenvalues. Conditions and consequences of the variational principle (2.1) or (2.2) are discussed. Taking an optimization perspective, we established the global convergence of a basic CG method, and we obtained a better understanding of the asymptotic behavior of CG performing Rayleigh functional minimization. Several variants of single-vector and block PCG methods with soft deflation were proposed to compute multiple eigenvalues. Variable and indefinite preconditioning is shown effective to accelerate the convergence of PCG methods. Similar to the case for linear eigenproblems, numerical experiments show that LOBPCG is the most efficient and robust method in the nonlinear setting.

Appendix

1. Proof of Proposition 2.6.

Proof. We give the proof by mathematical induction. Assume without loss of generality that J is of positive type, so that $T(\cdot)$ has n eigenvalues $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$. First, let $m = 2$ and consider $x = c_1 v_{k_1} + c_2 v_{k_2}$ for $c_1, c_2 \neq 0$ such that $\lambda_{k_1} \leq \lambda_{k_2}$. Assume by contradiction that $\rho(x) < \lambda_{k_1}$, for example. Since J is of positive type, $(\lambda_{k_1} - \rho(x))\,(x^* T(\lambda_{k_1})x) > 0$, and thus $0 < x^* T(\lambda_{k_1})x = (\bar{c}_1 v_{k_1}^* + \bar{c}_2 v_{k_2}^*) T(\lambda_{k_1})(c_1 v_{k_1} + c_2 v_{k_2}) = |c_2|^2 v_{k_2}^* T(\lambda_{k_1}) v_{k_2}$. Since $c_2 \neq 0$, $v_{k_2}^* T(\lambda_{k_1}) v_{k_2} > 0$, and thus $\lambda_{k_1} < \lambda_{k_2}$ (otherwise, $\lambda_{k_1} = \lambda_{k_2}$, and $v_{k_2}^* T(\lambda_{k_1}) v_{k_2} = 0$). Therefore $(\lambda_{k_1} - \rho(v_{k_2}))\left(v_{k_2}^* T(\lambda_{k_1}) v_{k_2}\right) < 0$, where $\rho(v_{k_2}) = \lambda_{k_2} > \lambda_{k_1}$. This is contradictory to the fact that J is of positive type. Therefore, $\rho(x) \geq \lambda_{k_1}$. Similarly, $\rho(x) \leq \lambda_{k_2}$.

If, in addition, $\lambda_{k_1} < \lambda_{k_2}$, then we can show easily that $\rho(x) \neq \lambda_{k_1}, \lambda_{k_2}$. In fact, assume by contradiction that $\rho(x) = \lambda_{k_1}$. Then $0 = x^* T(\rho(x))x = |c_2|^2 v_{k_2}^* T(\lambda_{k_1}) v_{k_2}$, which means that $\lambda_{k_1} = \rho(v_{k_2}) = \lambda_{k_2}$. This is contradictory to the assumption that $\lambda_{k_1} < \lambda_{k_2}$. Similarly, we can show that $\rho(x) \neq \lambda_{k_2}$, and thus $\lambda_{k_1} < \rho(x) < \lambda_{k_2}$.

Assume that the conclusion holds for $m \geq 2$. Consider $x = \sum_{i=1}^{m} c_i v_{k_i}$, and $\tilde{x} = x + c_{m+1} v_{k_{m+1}}$ ($c_i \neq 0$, $1 \leq i \leq m+1$). By assumption, $\lambda_{k_1} \leq \rho(x) \leq \lambda_{k_m} \leq \lambda_{k_{m+1}}$. Assume by contradiction that $\rho(\tilde{x}) > \lambda_{k_{m+1}}$. Note that $(\lambda_{k_{m+1}} - \rho(\tilde{x}))\left(\tilde{x}^* T(\lambda_{k_{m+1}})\tilde{x}\right) > 0$ since J is of positive type, and thus $0 > \tilde{x}^* T(\lambda_{k_{m+1}})\tilde{x} = (\tilde{x} - c_{m+1} v_{m+1})^* \, T(\lambda_{k_{m+1}})\,(\tilde{x} - c_{m+1} v_{m+1}) = x^* T(\lambda_{k_{m+1}})x$. Therefore, $\rho(x) < \lambda_{k_{m+1}}$ (otherwise, $\rho(x) = \lambda_{k_{m+1}}$, and $x^* T(\lambda_{k_{m+1}})x = 0$), and it follows that $(\lambda_{k_{m+1}} - \rho(x))\left(x^* T(\lambda_{k_{m+1}})x\right) < 0$. This is impossible since J is of positive type. Therefore, $\rho(\tilde{x}) \leq \lambda_{k_{m+1}}$. Similarly, $\rho(\tilde{x}) \geq \lambda_{k_1}$ can be shown.

If, in addition, $\lambda_{k_1} < \lambda_{k_{m+1}}$, then $\rho(\tilde{x}) \neq \lambda_{k_1}, \lambda_{k_{m+1}}$. In fact, assume that $\rho(\tilde{x}) = \lambda_{k_{m+1}}$ by contradiction. Then $0 = \tilde{x}^* T(\rho(\tilde{x}))\tilde{x} = \left(\tilde{x} - c_{m+1} v_{k_{m+1}}\right)^* T(\lambda_{k_{m+1}})\left(\tilde{x} - c_{m+1} v_{k_{m+1}}\right) =$

$x^*T(\lambda_{k_{m+1}})x$. That is, $\rho(x) = \lambda_{k_{m+1}}$. However, by the inductive hypothesis, $\lambda_{k_1} \le \rho(x) \le \lambda_{k_m}$, and $\lambda_{k_1} < \rho(x) < \lambda_{k_m}$ if $\lambda_{k_1} < \lambda_{k_m}$, $\rho(x) \neq \lambda_{k_{m+1}}$ unless $\lambda_{k_1} = \lambda_{k_2} = \ldots = \lambda_{k_{m+1}}$, which is contradictory to the assumption that $\lambda_{k_1} < \lambda_{k_{m+1}}$. Therefore, $\rho(\tilde{x}) \neq \lambda_{k_{m+1}}$. One can show similarly that $\rho(\tilde{x}) \neq \lambda_{k_1}$, and thus $\lambda_{k_1} < \rho(\tilde{x}) < \lambda_{k_{m+1}}$. This completes the proof for J of positive type. The proof for J of negative type is analogous. □

2. Proof of Proposition 3.1.

Proof. From the definition of $\rho(\cdot)$, $x^TT(\rho)x = 0$, and $(x+\Delta x)^TT(\rho(x+\Delta x))(x+\Delta x) = 0$ for all $\Delta x \in \mathbb{C}^n$. Since $T(\cdot)$ and $\rho(\cdot)$ are twice continuously differentiable, we have

$$\begin{aligned}
&(x+\Delta x)^T T(\rho(x+\Delta x))(x+\Delta x) \\
&= (x+\Delta x)^T T\left(\rho(x) + \nabla\rho^T\Delta x + \frac{1}{2}\Delta x^T\nabla^2\rho\Delta x + \mathcal{O}(\Delta x^3)\right)(x+\Delta x) \\
&= x^TT(\rho)x + 2x^TT(\rho)\Delta x + \left(x^TT'(\rho)x\right)\nabla\rho^T\Delta x + Q_2(\Delta x; x) + \mathcal{O}(\Delta x^3) \\
&= \left(2x^TT(\rho) + \left(x^TT'(\rho)x\right)\nabla\rho^T\right)\Delta x + Q_2(\Delta x; x) + \mathcal{O}(\Delta x^3) = 0,
\end{aligned} \tag{6.1}$$

where

$$\begin{aligned}
Q_2(\Delta x; x) = \Delta x^TT(\rho)\Delta x + \Delta x^TT'(\rho)x\nabla\rho^T\,\Delta x + \Delta x^T\nabla\rho\, x^TT'(\rho)\Delta x + \\
\frac{x^TT'(\rho)x}{2}\Delta x^T\nabla^2\rho\,\Delta x + \frac{x^TT''(\rho)x}{2}\Delta x^T\nabla\rho\,\nabla\rho^T\Delta x
\end{aligned} \tag{6.2}$$

contains all the second order terms of Δx. Since (6.1) holds for all small Δx, both the first order terms and the second order terms of Δx must be identically zero. Therefore (3.1) and (3.2) follow immediately from (6.1) and (6.2), as $x^TT'(\rho)x \neq 0$ by assumption. □

3. Proof of Proposition 3.2.

Proof. First note that $\rho(v_\ell) = \lambda_\ell$, $\nabla\rho(v_\ell) = -\frac{2}{v_\ell^TT'(\lambda_\ell)v_\ell}T(\lambda_\ell)v_\ell = 0$, and $\nabla^2\rho(v_\ell) = -\frac{2}{v_\ell^TT'(\lambda_\ell)v_\ell}T(\lambda_\ell)$. Consider the Taylor expansion of $\rho(x)$ at v_ℓ

$$\begin{aligned}
\rho(x) &= \rho\left(\frac{x}{\gamma\cos\theta}\right) = \rho(v_\ell + g\tan\theta) \\
&= \rho(v_\ell) + \nabla\rho(v_\ell)(g\tan\theta) + \frac{1}{2}(g\tan\theta)^T\nabla^2\rho(v_\ell)(g\tan\theta) + \mathcal{O}(\|g\tan\theta\|^3) \\
&= \lambda_\ell - \frac{g^TT(\lambda_\ell)g}{v_\ell^TT'(\lambda_\ell)v_\ell}\tan^2\theta + \mathcal{O}(\tan^3\theta),
\end{aligned} \tag{6.3}$$

where $v_\ell^TT'(\lambda_\ell)v_\ell \neq 0$ because J is of definite type. It follows that $|\rho(x) - \lambda_\ell| = \mathcal{O}(\tan^2\theta)$.

To analyze $\|\nabla\rho(x)\|$, note that since $v_\ell^TT'(\lambda_\ell)v_\ell \neq 0$, $v_\ell^TT'(\lambda_\ell)v_\ell\cos^2\theta + \mathcal{O}(\sin\theta)$ is bounded away from zero for sufficiently small θ. Therefore,

$$\begin{aligned}
\|\nabla\rho(x)\| &= \frac{2\|T(\rho)x\|}{|x^TT'(\rho)x|} = \frac{2\left\|\left(T(\lambda_\ell) + \mathcal{O}(|\rho-\lambda_\ell|)\right)\gamma(v_\ell\cos\theta + g\sin\theta)\right\|}{|x^TT'(\rho)x|} \\
&= \frac{2\|T(\lambda_\ell)g\|\sin\theta}{\gamma\left|(v_\ell\cos\theta + g\sin\theta)^T(T(\lambda_\ell) + \mathcal{O}(|\rho-\lambda_\ell|)(v_\ell\cos\theta + g\sin\theta)\right|} \\
&= \frac{2\|T(\lambda_\ell)g\|\sin\theta}{\gamma\left|v_\ell^TT'(\lambda_\ell)v_\ell\cos^2\theta + \mathcal{O}(\sin\theta)\right|} = \mathcal{O}(\sin\theta). \quad \square
\end{aligned} \tag{6.4}$$

4. Proof of Theorem 3.4.

Proof. From the second Wolfe condition (3.5), we have

$$\|x_{k+1}\|\nabla\rho(x_{k+1})^T p_k \geq -\|x_{k+1}\||\nabla\rho(x_{k+1})^T p_k| \geq c_2\|x_k\|\nabla\rho(x_k)^T p_k,$$

and thus

$$\left(\|x_{k+1}\|\nabla\rho(x_{k+1})^T - \|x_k\|\nabla\rho(x_k)^T\right) p_k \geq (c_2-1)\|x_k\|\nabla\rho(x_k)^T p_k.$$

From the Lipschitz continuity in direction (3.3), we have

$$\left(\|x_{k+1}\|\nabla\rho(x_{k+1})^T - \|x_k\|\nabla\rho(x_k)^T\right) p_k \leq \frac{\pi L\|x_{k+1}-x_k\|}{2\|x_k\|}\|p_k\| = \frac{\alpha_k \pi L\|p_k\|^2}{2\|x_k\|}.$$

It follows that

$$\frac{\alpha_k \pi L\|p_k\|^2}{2\|x_k\|} \geq (c_2-1)\|x_k\|\nabla\rho(x_k)^T p_k > 0, \qquad \text{or} \qquad \alpha_k \geq \frac{2(c_2-1)}{\pi L}\frac{\|x_k\|^2}{\|p_k\|^2}\nabla\rho(x_k)^T p_k > 0.$$

Then, it follows from (3.4) that

$$\begin{aligned}
\rho(x_{k+1}) &\leq \rho(x_k) - c_1(-\alpha_k\nabla\rho(x_k)^T p_k) \\
&\leq \rho(x_k) - c_1\frac{2(1-c_2)}{\pi L}\frac{\|x_k\|^2}{\|p_k\|^2}\left(\nabla\rho(x_k)^T p_k\right)^2 = \rho(x_k) - C\|x_k\|^2\|\nabla\rho(x_k)\|^2\cos^2\theta_k \\
&\leq \rho(x_{k-1}) - C\|x_{k-1}\|^2\|\nabla\rho(x_{k-1})\|^2\cos^2\theta_{k-1} - C\|x_k\|^2\|\nabla\rho(x_k)\|^2\cos^2\theta_k \\
&\leq \ldots \leq \rho(x_0) - C\sum_{i=0}^{k}\|x_i\|^2\|\nabla\rho(x_i)\|^2\cos^2\theta_i, \qquad \text{where } C = 2c_1(1-c_2)/\pi L > 0.
\end{aligned}$$

By the variational principle (2.1) or (2.2), $\rho(x)$ is bounded below by λ_1 or λ_n for all $x \in \mathbb{R}^n \setminus \{0\}$, where $\lambda_1, \lambda_n \in J$ are finite, and (3.6) is thus established. □

5. Proof of Lemma 3.5

Proof. First, it is easy to see that for $0 < c_2 < \frac{1}{2}$, $-2 < -\frac{1}{1-c_2} < -1$ and $-1 < \frac{2c_2-1}{1-c_2} < 0$. From Algorithm 1, $p_0 = -\|x_0\|\nabla\rho(x_0)$, and thus $-\frac{1}{1-c_2} \leq \frac{\nabla\rho(x_0)^T p_0}{\|\nabla\rho(x_0)\|^2\|x_0\|} = -1 \leq \frac{2c_2-1}{1-c_2}$.

Assume that (3.7) holds for some $k \geq 0$. Our aim is to establish this inequality for $k+1$. In fact, from Step 3 of Algorithm 1, it follows that

$$\frac{\nabla\rho(x_{k+1})^T p_{k+1}}{\|x_{k+1}\|\|\nabla\rho(x_{k+1})\|^2} = -1 + \beta_{k+1}\frac{\nabla\rho(x_{k+1})^T p_k}{\|x_{k+1}\|\|\nabla\rho(x_{k+1})\|^2} = -1 + \frac{\|x_{k+1}\|\nabla\rho(x_{k+1})^T p_k}{\|\nabla\rho(x_k)\|^2\|x_k\|^2}.$$

Applying the second Wolfe condition (3.5) to the numerator of the last term above, we have

$$-1 + c_2\frac{\nabla\rho(x_k)^T p_k}{\|x_k\|\|\nabla\rho(x_k)\|^2} \leq \frac{\nabla\rho(x_{k+1})^T p_{k+1}}{\|x_{k+1}\|\|\nabla\rho(x_{k+1})\|^2} \leq -1 - c_2\frac{\nabla\rho(x_k)^T p_k}{\|x_k\|\|\nabla\rho(x_k)\|^2}, \tag{6.5}$$

where $-\frac{1}{1-c_2} \leq \frac{\nabla\rho(x_k)^T p_k}{\|x_k\|\|\nabla\rho(x_k)\|^2} \leq \frac{2c_2-1}{1-c_2} < 0$ by induction hypothesis. Therefore,

$$-\frac{1}{1-c_2} = -1 - c_2\frac{1}{1-c_2} \leq \frac{\nabla\rho(x_{k+1})^T p_{k+1}}{\|x_{k+1}\|\|\nabla\rho(x_{k+1})\|^2} \leq -1 + c_2\frac{1}{1-c_2} = \frac{2c_2-1}{1-c_2}, \tag{6.6}$$

which completes the proof. □

REFERENCES

[1] M. Al-Ammari and F. Tisseur, *Hermitian matrix polynomials with real eigenvalues of definite type. Part I: Classification*, Linear Algebra and Its Applications, Vol. 436 (2012), pp. 3954–3973.

[2] P. Arbenz, *Lecture Notes on Solving Large Scale Eigenvalue Problems*, ETH Zürich, 2012.

[3] P. Arbenz, U. L. Hetmaniuk, R. B. Lehoucq and R. S. Tuminaro, *A comparison of eigensolvers for large-scale 3D modal analysis using AMG-preconditioned iterative methods*, International Journal for Numerical Methods in Engineering, Vol. 64 (2005), pp. 204–236.

[4] T. Betcke and D. Kressner, *Perturbation, extraction and renement of invariant pairs for matrix polynomials*, Linear Algebra and its Applications, Vol. 435 (2011), pp. 514–536.

[5] T. Betcke, N. J. Higham, V. Mehrmann, C. Schröder and F. Tisseur, *NLEVP: A Collection of Nonlinear Eigenvalue Problems*, ACM Transactions on Mathematical Software, Vol. 39, Article 7, 2013.

[6] M. M. Betcke and H. Voss, *Restarting iterative projection methods for Hermitian nonlinear eigenvalue problems with minmax property*, Report 157, Institute of Mathematics, Hamburg University of Technology.

[7] Y. Cai, Z. Bai, J. E. Paskd and N. Sukumar, *Hybrid preconditioning for iterative diagonalization of ill-conditioned generalized eigenvalue problems in electronic structure calculations*, Journal of Computational Physics, Vol. 255 (2013), pp. 16–30.

[8] F. Chaitin-Chatelin and M. B. van Gijzen, *Analysis of parameterized quadratic eigenvalue problems in computational acoustics with homotopic deviation theory*, Numer. Linear Algebra Appl. Vol. 13 (2006), pp. 487–512.

[9] C. Conca, J. Planchard and M. Vanninathan, *Fluid and Periodic Structures*, Vol. 38 of Research in Applied Mathematics, J. Wiley, Chichester, Masson, Paris 1995.

[10] Y.H. Dai, *Nonlinear Conjugate Gradient Methods*, Wiley Encyclopedia of Operations Research and Management Science (2011), DOI:10.1002/9780470400531.eorms0183.

[11] A. Edelman, T. A. Arias and S. T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM Journal on Matrix Analysis and Applications, Vol. 20 (1998), pp. 303–353.

[12] C. Effenberger, *Robust successive computation of eigenpairs for nonlinear eigenvalue problems*, SIAM Journal on Matrix Analysis and Applications, Vol. 34 (2013), pp. 1231–1256.

[13] H. C. Elman, D. J. Silvester and A. J. Wathen, *Finite Elements and Fast Iterative Solvers, Second Edition*, Oxford University Press, New York, 2014.

[14] G. Gambolati, F. Sartoretto and P. Florian, *An orthogonal accelerated deflation technique for large symmetric eigenproblems*, Comp. Methods App. Mech. Eng., Vol. 94 (1992), pp. 13–23.

[15] N. I. M. Gould, D. Orban and P. L. Toint, *Numerical methods for large-scale nonlinear optimization*, Acta Numerica, Vol. 14 (2005), pp. 299–361.

[16] C.-H. Guo, N. Higham and F. Tisseur, *Detecting and solving hyperbolic quadratic eigenvalue problems*, SIAM Journal on Matrix Analysis and Applications, Vol. (2008), pp. 1593–1613.

[17] K. P. Hadeler, *Variationsprinzipien bei nichtlinearen Eigenwertaufgaben*, Archive for Rational Mechanics and Analysis, Vol. 30 (1968), pp. 297–307.

[18] W. W. Hager and H. Zhang, *A survey of nonlinear conjugate gradient methods*, Pacific Journal of Optimization, Vol. 2 (2006), pp. 35–58.

[19] E. Jarlebring, *The Spectrum of Delay-Differential Equations: Numerical Methods, Stability and Perturbation*, PhD thesis, Inst. Comp. Math, TU Braunschweig, 2008.

[20] A. V. Knyazev, *A preconditioned conjugate gradient method for eigenvalue problems and its implementation in a subspace*, in Numerical Treatment of Eigenvalue Problems Vol. 5 (Oberwolfach, 1990), International Series of Numerical Mathematics, Vol. 96, pp. 143–154, Birkh´auser, Basel, 1991.

[21] A. V. Knyazev, *Preconditioned eigensolvers – an oxymoron?*, Electronic Transactions on Numerical Analysis, Vol. 7 (1998), pp. 104–123.

[22] A. V. Knyazev, *Toward the optimal preconditioned eigensolver: locally optimal block preconditioned conjugate gradient method*, SIAM Journal on Scientific Computing, Vol. 23 (2001), pp. 517–541.

[23] J. Nocedal and S. J. Wright, *Numerical Optimization, 2nd edition*, Springer Series in Operations Research, Springer, New York, 2006.

[24] Y. Saad, *Iterative Methods for Sparse Linear Systems, 2nd edition*, SIAM, Philadelphia, 2003.

[25] K. Schreiber, *Nonlinear Eigenvalue Problems: Newton-type Methods and Nonlinear Rayleigh Functionals*, Ph.D thesis, Department of Mathematics, TU Berlin, 2008.

[26] H. Schwetlicka and K. Schreiber, *Nonlinear Rayleigh functionals*, Linear Algebra and Its Applications, Vol. 436 (2012), pp. 3991–4016.

[27] G. L. G. Sleijpen and H. A. van der Vorst, *A Jacobi-Davidson iteration method for linear eigenvalue problems*, SIAM Journal on Matrix Analysis and Applications, Vol. 17 (1996), pp. 401–425.

[28] G. Sleijpen, A. Booten, D. Fokkema and H. Van der Vorst, *Jacobi-Davidson type methods for generalized eigenproblems and polynomial eigenproblems*, BIT, Vol. 36 (1996), pp. 595–633.

[29] S. I. Solov'ëv, *Preconditioned iterative methods for a class of nonlinear eigenvalue problems*, Linear Algebra and Its Applications. Vol. 415 (2006), pp. 210–229.

[30] H. Voss, *A minmax principle for nonlinear eigenproblems depending continuously on the eigenparameter*, Numer. Linear Algebra Appl., Vol. 16 (2009), pp. 899–913.

[31] H. Voss and B. Werner, *A minimax principle for nonlinear eigenvalue problems with applications to nonoverdamped systems*, Mathematical Methods in the Applied Sciences, Vol. 4 (1982), pp. 415–424.

[32] S. Wei and I. Kao, *Vibration analysis of wire and frequency response in the modern wiresaw manufacturing process*, Journal of Sound and Vibration, Vol. 231 (2000), pp. 1383–1395.

[33] H. Yang, *Conjugate gradient methods for the Rayleigh quotient minimization of generalized eigenvalue problems*, Computing, Vol. 51 (1993), pp. 79–94.