

**PRECONDITIONED EIGENSOLVERS FOR LARGE-SCALE  
NONLINEAR HERMITIAN EIGENPROBLEMS  
WITH VARIATIONAL CHARACTERIZATIONS.  
I. EXTREME EIGENVALUES**

DANIEL B. SZYLD AND FEI XUE

**ABSTRACT.** Efficient computation of extreme eigenvalues of large-scale linear Hermitian eigenproblems can be achieved by preconditioned conjugate gradient (PCG) methods. In this paper, we study PCG methods for computing 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 eigenvalue. The efficiency of variants of PCG is illustrated by numerical experiments. Overall, the locally optimal block preconditioned conjugate gradient (LOBPCG) is the most efficient method, as in the linear setting.

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 [7], time-harmonic acoustic wave equation in bounded domains [9], delay differential equations [20], and modeling of vibrations of certain fluid-solid structures [10], loaded strings [30], and wiresaws [36]. 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,

---

Received by the editor August 27, 2014, and in revised form, April 28, 2015 and June 9, 2015.  
2010 *Mathematics Subject Classification.* Primary 65F15, 65F50, 15A18, 15A22.

*Key words and phrases.* Nonlinear Hermitian eigenproblems, variational principle, preconditioned conjugate gradient, convergence analysis.

The first author was supported by NSF under grants DMS-1115520 and DMS-1418882.

The second author was supported by NSF under grants DMS-1115520 and DMS-1419100.

and linear Hermitian eigenproblems; see, e.g., [3, 11, 14–16, 19, 23–25], 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  $P : \mathbb{R} \rightarrow \mathbb{C}^{n \times n}$  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 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 eigenvalue 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 completed in Part II of our study [32].

The rest of this paper is organized as follows. In Section 2, we review the basics of the nonlinear Hermitian eigenproblem, and of 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  $T(\cdot) : J \subset \mathbb{R} \rightarrow \mathbb{C}^{n \times n}$  maps a real scalar  $\mu$  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  $\text{null } T(\lambda)$ . To establish a nonlinear variational principle and other relevant properties, we begin the discussion with several definitions.

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

$$(2.1) \quad x^* T(\rho(x)) x = 0.$$

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

**Definition 2.3.** The scalar  $\lambda$  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  $\lambda_k$ .

To simplify our analysis, we make a nonrestrictive assumption that  $T(\cdot)$  does not have both zero and infinite eigenvalues on the interval  $J$ . If this is not the case, choose  $\sigma \in \mathbb{R}$  that is not an eigenvalue and define  $\tilde{T}(\mu) = T(\mu + \sigma)$  such that zero is not an eigenvalue of  $\tilde{T}(\cdot)$ . 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 let  $D = \mathbb{C}^n \setminus \{0\}$  be the domain of the Rayleigh functional  $\rho : D \rightarrow J$ . Then  $J$  is an interval of positive (negative) type for  $T(\cdot)$  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 for  $T(\cdot)$  if  $x^* T'(\rho(x))x > 0$  ( $< 0$ ) for all  $x \in D$ . If, in addition,  $T(\cdot)$  is twice continuously differentiable and  $x^* T''(\rho(x))x \neq 0$  for all  $x \in D$ , then  $J$  is of positive (negative) type for  $T(\cdot)$  if and only if  $x^* T'(\rho(x))x > 0$  ( $< 0$ ) for all  $x \in D$ .

*Proof.* For simplicity and clarity, we only give the proof for  $J$  of positive type for  $T(\cdot)$ , as the case of negative type is very similar.

We first establish the relation between the type of  $J$  for  $T(\cdot)$  and the definiteness of  $T(a)$  and  $T(b)$ . From Definition 2.2,  $J$  is an interval of positive type for  $T(\cdot)$  if and only if for any  $x \in D$  and  $\mu \in [a, \rho(x)) \cup (\rho(x), b]$ ,  $\text{sign}(\mu - \rho(x)) = \text{sign}(x^*T(\mu)x)$ , or equivalently,  $\text{sign}(x^*T(\mu)x)$  is constantly  $-1$  on  $[a, \rho(x))$ , and constantly  $+1$  on  $(\rho(x), b]$ . By the continuity of  $T(\cdot)$ ,  $x^*T(\mu)x$  is continuous with respect to  $\mu$ ; in addition,  $\rho(x)$  is the unique solution of  $x^*T(\mu)x = 0$  on  $(a, b)$ . These conditions are thus equivalent to  $x^*T(a)x < 0$  and  $x^*T(b)x > 0$  for all  $x \in D$ , i.e., the negative and positive definiteness of  $T(a)$  and  $T(b)$ , respectively.

If  $T(\cdot)$  is continuously differentiable, then  $x^*T'(\mu)x$  is continuous with respect to  $\mu$  on  $J$  for all  $x \in D$ . Therefore, recalling (2.1), if  $x^*T'(\rho(x))x = \lim_{\mu \rightarrow \rho(x)} \frac{x^*T(\mu)x - x^*T(\rho(x))x}{\mu - \rho(x)} = \lim_{\mu \rightarrow \rho(x)} \frac{x^*T(\mu)x}{\mu - \rho(x)} > 0$ , then there is a small  $\delta > 0$  such that  $\text{sign}(x^*T(\mu)x) = \text{sign}(\mu - \rho(x))$  in  $(\rho(x) - \delta, \rho(x)) \cup (\rho(x), \rho(x) + \delta)$ . By continuity of  $x^*T(\mu)x$  with respect to  $\mu$  and the fact that  $\rho(x)$  is the unique solution of  $x^*T(\mu)x = 0$ , we have  $\text{sign}(x^*T(\mu)x) = \text{sign}(\mu - \rho(x))$  on  $[a, \rho(x)) \cup (\rho(x), b]$ , or equivalently,  $J$  is of positive type for  $T(\cdot)$ .

Finally, assume in addition that  $T(\cdot)$  is twice continuously differentiable, and that for all  $x \in D$ ,  $x^*T''(\rho(x))x \neq 0$ . To complete the proof, we only need to show that if  $J$  is of positive type for  $T(\cdot)$ , then  $x^*T'(\rho(x))x > 0$ . In this case, we have  $\text{sign}(\mu - \rho(x)) = \text{sign}(x^*T(\mu)x)$ , and thus  $x^*T'(\rho(x))x = \lim_{\mu \rightarrow \rho(x)} \frac{x^*T(\mu)x - x^*T(\rho(x))x}{\mu - \rho(x)} = \lim_{\mu \rightarrow \rho(x)} \frac{x^*T(\mu)x}{\mu - \rho(x)} \geq 0$ . Therefore we simply need to show that  $x^*T'(\rho(x))x \neq 0$  for all  $x \in D$ .

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  $\xi \in (\rho(x) - |\delta\rho|, \rho(x) + |\delta\rho|)$ . Since  $x^*T(\rho(x))x = 0$ , and  $x^*T'(\rho(x))x = 0$  by assumption, we have  $\text{sign}(\mu - \rho(x)) = \text{sign}(x^*T(\mu)x) = \text{sign}(x^*T''(\xi)x) = \text{sign}(x^*T''(\rho(x))x) \neq 0$  for all  $\mu \in (\rho(x) - |\delta\rho|, \rho(x)) \cup (\rho(x), \rho(x) + |\delta\rho|)$  with sufficiently small  $|\delta\rho|$ , due to the continuity of  $T''(\cdot)$ . But this is impossible as  $\text{sign}(\mu - \rho(x)) = 1$  on  $(\rho(x) - |\delta\rho|, \rho(x))$  and  $-1$  on  $(\rho(x), \rho(x) + |\delta\rho|)$ . Therefore,  $x^*T'(\rho(x))x \neq 0$  for all  $x \in D$ , and the proposition is established.  $\square$

*Remark.* For linear generalized Hermitian eigenproblems  $Av = \lambda Bv$  with a positive definite  $B$  ( $B = I$  for the standard eigenproblem  $Av = \lambda v$ ), since  $T(\mu) = \mu B - A$ , we have  $x^*T'(\rho(x))x = x^*Bx > 0$  satisfied for any nonzero vector  $x$ . Therefore any open interval  $J$  containing the whole spectrum of  $(A, B)$  is of positive type for this eigenproblem.

The conditions given in Proposition 2.4 can be used to determine whether  $J$  is or can be of definite type for  $T(\cdot)$ . On such an interval of definite type for  $T(\cdot)$ , we have a variational characterization of eigenvalues of  $T(\cdot)$  and the orthogonality of eigenvectors [18, 33, 35]. 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 for  $T(\cdot)$ , 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 for  $T(\cdot)$ , then

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

if  $J$  is of negative type for  $T(\cdot)$ , then

$$(2.3) \quad \begin{aligned} \lambda_k &= \max\{\min\{\rho(x) \mid x \in S, x \neq 0\} \mid \dim(S) = k\} \quad \text{and} \\ \lambda_k &= \min\{\max\{\rho(x) \mid x \in S, x \neq 0\} \mid \dim(S) = n - k + 1\}. \end{aligned}$$

Moreover, there exist  $n$  corresponding eigenvectors  $\{v_k\}_{k=1}^n$  that form a basis for  $\mathbb{C}^n$ , and they satisfy  $\chi(v_i, v_i) \neq 0$  and  $\chi(v_i, v_j) = 0$  for all  $i \neq j$ ,  $1 \leq i, j \leq n$ , where the scalar-valued function  $\chi(\cdot, \cdot)$  is defined as

$$(2.4) \quad \chi(x, y) = \begin{cases} y^* \frac{T(\rho(y)) - T(\rho(x))}{\rho(y) - \rho(x)} x & (\rho(x) \neq \rho(y)), \\ y^* T'(\rho(x)) x & (\rho(x) = \rho(y)). \end{cases}$$

It follows immediately from Theorem 2.5 that if  $J$  is of positive type for  $T(\cdot)$ , then  $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ , and if  $J$  is of negative type for  $T(\cdot)$ , then  $\lambda_n \leq \lambda_{n-1} \leq \dots \leq \lambda_1$ . Such a natural ordering of eigenvalues is important in stability assessment applications. In addition, the variational principle 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  $(x^* T(\rho(x)) x) v = 0 v = 0$ , where  $v$  is a nonzero scalar.

**Theorem 2.6.** *Let  $J = (a, b)$  be finite and of definite type for a continuous  $T(\cdot)$ . For any  $x \in \mathbb{C}^n \setminus \{0\}$ , let  $x = \sum_{i=1}^m c_i v_{j_i}$  ( $1 \leq m \leq n$ ,  $1 \leq j_1 < j_2 < \dots < j_m \leq n$ ) be the eigenvector decomposition of  $x$ , where  $c_i \neq 0$  for all  $1 \leq i \leq m$ . If  $J$  is of positive type for  $T(\cdot)$ , then  $\lambda_{j_1} \leq \rho(x) \leq \lambda_{j_m}$ ; if, in addition,  $\lambda_{j_1} < \lambda_{j_m}$ , then  $\lambda_{j_1} < \rho(x) < \lambda_{j_m}$ ; if  $J$  is of negative type for  $T(\cdot)$ , then  $\lambda_{j_m} \leq \rho(x) \leq \lambda_{j_1}$ ; 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 for  $T(\cdot)$ ,  $T(\cdot)$  be continuously differentiable on  $J$ , and let  $U \in \mathbb{C}^{n \times m}$  contain  $m$  linearly independent columns. Then the projected eigenproblem  $T_p(\nu)w \equiv U^* T(\nu) U w = 0$  has exactly  $m$  eigenpairs  $\{(\nu_j, w_j)\}_{j=1}^m$  satisfying the nonlinear variational principle (2.2) or (2.3), where  $\nu_j$  is a  $j$ -th eigenvalue of  $T_p(\cdot)$ . In addition, if  $J$  is of positive type for  $T(\cdot)$ , then  $\lambda_j \leq \nu_j \leq \lambda_{n-m+j}$ ; if  $J$  is of negative type for  $T(\cdot)$ , then  $\lambda_{n-m+j} \leq \nu_j \leq \lambda_j$  ( $1 \leq j \leq m$ ).*

*Proof.* First, note that for any  $x \in \mathbb{C}^m \setminus \{0\}$ , the solution  $\rho \in J$  of the scalar equation  $x^* T_p(\rho) x = (Ux)^* T(\rho) (Ux) = 0$  is unique due to the uniqueness of the Rayleigh functional value for  $T(\cdot)$ . The Rayleigh functional  $\rho(\cdot) : \mathbb{C}^m \setminus \{0\} \rightarrow \mathbb{R}$  is thus well defined for  $T_p(\cdot)$ . Without loss of generality, assume that  $J$  is of positive type for  $T(\cdot)$ , so that  $T(a) < 0$  and  $T(b) > 0$ .<sup>1</sup> Therefore,  $T_p(a) = U^* T(a) U < 0$  and  $T_p(b) = U^* T(b) U > 0$ . By Proposition 2.4,  $J$  is of positive type for  $T_p(\cdot)$ . In addition, since  $T(\cdot)$  is continuously differentiable on  $J$ , so is  $T_p(\cdot) = U^* T(\cdot) U$ .

<sup>1</sup>Here we adopt the conventional notation  $M > 0$  and  $M < 0$  for positive and negative definite matrices  $M$ , respectively.

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 (2.2) or (2.3).

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  $W_j = \text{span}\{w_1, \dots, w_j\}$  ( $1 \leq j \leq m$ ). Then

$$(2.5) \quad \begin{aligned} \nu_j &= \max\{\rho_p(x) \mid x \in W_j, x \neq 0\} = \max\{\rho(z) \mid z \in UW_j, z \neq 0\} \\ &\geq \min\{\max\{\rho(z) \mid z \in S, z \neq 0\} \mid \dim(S) = j\} = \lambda_j. \end{aligned}$$

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

$$(2.6) \quad \begin{aligned} \nu_j &= \min\{\rho_p(x) \mid x \in W_{m-j+1}, x \neq 0\} = \min\{\rho(z) \mid z \in UW_{m-j+1}, z \neq 0\} \\ &\leq \max\{\min\{\rho(z) \mid z \in S, z \neq 0\} \mid \dim(S) = m - j + 1\} \\ &= \lambda_{n-(m-j+1)+1} = \lambda_{n-m+j}. \end{aligned}$$

The proof for  $J$  of negative type for  $T(\cdot)$  is very similar and is thus omitted.  $\square$

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 (2.2) or (2.3). 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 [37] 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  $\alpha$  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  $\alpha x$ , 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 [12]. 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  $\rho$ ,  $\nabla\rho$  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  $T(\cdot) : \mathbb{R} \rightarrow \mathbb{R}^{n \times n}$  be a real symmetric matrix-valued function, and  $\rho(\cdot) : \mathbb{R}^n \setminus \{0\} \rightarrow J \subset \mathbb{R}$  the Rayleigh functional, where  $J$  is of definite type for  $T(\cdot)$ . Assume that  $T(\cdot)$  and  $\rho(\cdot)$  are twice continuously differentiable, and that  $x^T T'(\rho)x \neq 0$  for all  $x \in \mathbb{R}^n \setminus \{0\}$ . Then*

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

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

*Proof.* See the Appendix.  $\square$

*Remark.* The assumption that  $x^T T'(\rho)x \neq 0$  for all  $x \neq 0$ , which guarantees that  $\nabla\rho$  and  $\nabla^2\rho$  are well defined, is generally not restrictive. As we explained, it holds for the linear Hermitian eigenproblem  $Av = \lambda Bv$  with a positive definite  $B$ . In addition, for truly nonlinear eigenproblems, if  $T''(\cdot)$  is continuous and  $x^T T''(\rho)x \neq 0$  for all  $x \in \mathbb{R}^n \setminus \{0\}$ , then  $x^T T'(\rho)x \neq 0$  if and only if  $J$  is an interval of definite type for  $T(\cdot)$ ; see 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  $\lambda_\ell$  be an eigenvalue of  $T(\cdot)$ , and  $v_\ell \in \text{null } T(\lambda_\ell)$  a unit eigenvector. Let  $x = \gamma(v_\ell \cos \theta + g \sin \theta)$  be an eigenvector approximation, where  $g \perp \text{null } T(\lambda_\ell)$  is a unit vector. Then for all  $\theta$  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  $\theta$ , then  $\|\nabla\rho(x)\| = \mathcal{O}(\sin \theta)$ .*

*Proof.* See the Appendix.  $\square$

**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 (2.2) or (2.3). 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.

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

---

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  $k = 0, 1, \dots$ , 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
- 

need to mimic the CG constructed on Grassmann manifold for linear Hermitian eigenproblems [12], which takes great effort to tune CG to proceed in a manner consistent with the geometry of the unit sphere. This is because by construction, both  $\beta_k$  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  $x_0, x_1, \dots$ ; 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  $\|\|x_1\| \nabla \rho(x_1) - \|x_2\| \nabla \rho(x_2)\| \leq L\alpha$  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

$$\angle(x_1, x_2) = \alpha, \quad \sin \alpha = \min \left\{ \frac{\|x_1 - x_2\|}{\|x_1\|} \mid \angle(x_1, x_2) = \alpha \right\} \leq \frac{\|x_1 - x_2\|}{\|x_1\|},$$

and similarly  $\sin \alpha \leq \frac{\|x_1 - x_2\|}{\|x_2\|}$ . Thus, if  $\nabla \rho$  is Lipschitz continuous in direction, then

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

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

**Theorem 3.4.** Let  $J = (a, b)$  be finite and of definite type for  $T(\cdot)$ , and  $T(\cdot)$  be a real symmetric matrix-valued function continuously differentiable on  $J$  for which the variational principle (2.2) or (2.3) holds. Consider the iteration  $x_{k+1} = x_k + \alpha_k p_k$ , where  $p_k$  is a descent direction for the Rayleigh functional  $\rho(\cdot) : \mathbb{R}^n \rightarrow J \subset \mathbb{R}$ , and  $\alpha_k$  satisfies the strong Wolfe conditions

$$(3.4) \quad \rho(x_{k+1}) \leq \rho(x_k) + c_1 \alpha_k \nabla \rho(x_k)^T p_k, \quad \text{and}$$

$$(3.5) \quad \|x_{k+1}\| \|\nabla \rho(x_{k+1})^T p_k\| \leq -c_2 \|x_k\| \nabla \rho(x_k)^T p_k,$$



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

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

*Proof.* See the Appendix.  $\square$

**Lemma 3.5.** Suppose that Algorithm 1 is implemented with a step length  $\alpha_k$  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  $k$ :

$$(3.7) \quad -\frac{1}{1-c_2} \leq \frac{\nabla\rho(x_k)^T p_k}{\|\nabla\rho(x_k)\|^2 \|x_k\|} \leq \frac{2c_2-1}{1-c_2}.$$

*Proof.* See the Appendix.  $\square$

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 for  $T(\cdot)$ , and let  $T(\cdot)$  be a real symmetric matrix-valued function continuously differentiable on  $J$ , for which the variational principle (2.2) or (2.3) 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  $S = \left\{ \frac{x}{\|x\|} \mid x \neq 0, \rho(x) \leq \rho(x_0) \right\}$ , and that  $\sup_{x \in S} \frac{\|x\|^2}{|x^T T'(\rho)x|} = F > 0$ . Then there exists  $\lambda_\ell \in \{\lambda_i\}$  such that  $\lim_{k \rightarrow \infty} \rho(x_k) = \lambda_\ell$ , and there is a subsequence  $\{x_{k_j}\}$  such that  $\lim_{j \rightarrow \infty} \angle(x_{k_j}, \text{null } T(\lambda_\ell)) = 0$ .

*Proof.* The proof is lengthy and technical, so we provide an outline of it. First, we show the crucial property that there exists a subsequence  $\{x_{k_j}\}$  of the CG iterates such that  $\lim_{j \rightarrow \infty} \|x_{k_j}\| \|\nabla\rho(x_{k_j})\| = 0$ . The idea we use closely follows the proof of the global convergence of the classical Fletcher-Reeves CG that satisfies the strong Wolfe conditions; see, e.g., [24, Section 5.2]. Then we show that  $\lim_{k \rightarrow \infty} \rho(x_k)$  must be an eigenvalue of  $T(\cdot)$ , and that the subsequence  $\{x_{k_j}\}$  satisfying  $\lim_{j \rightarrow \infty} \|x_{k_j}\| \|\nabla\rho(x_{k_j})\| = 0$  converges towards the desired eigenspace.

We begin with a trivial case. If  $x_0$  is an eigenvector corresponding to the eigenvalue  $\lambda_\ell$ , i.e.,  $x_0 \in \text{null } T(\lambda_\ell)$ , then  $\rho(x_0) = \lambda_\ell$ , and thus

$$\nabla\rho(x_0) = -\frac{2}{x_0^T T'(\lambda_\ell) x_0} T(\lambda_\ell) x_0 = 0.$$

The theorem holds straightforwardly.

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  $\|T(\cdot)\| : \mathbb{R} \rightarrow \mathbb{R}$  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  $\|T(\rho(x))\| \leq M$  for all  $x \neq 0$ . By assumption,  $\sup_{x \in S} \frac{\|x\|^2}{|x^T T'(\rho)x|} = F > 0$ , and it follows from (3.1) that  $\|x\| \|\nabla\rho\|_{x \in S} \leq \frac{2\|T(\rho)\| \|x\|^2}{|x^T T'(\rho)x|} \leq 2MF := \Gamma < \infty$ .

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

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

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

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

and therefore, from Step 3 of Algorithm 1,

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

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  $\|p_{k-1}\|^2$ ,  $\|p_{k-2}\|^2$ ,  $\dots$  and  $\|p_1\|^2$ . We have

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

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  $\beta_k$  (Step 3 of Algorithm 1) that  $\beta_k^2 \beta_{k-1}^2 \dots \beta_{k-i}^2 = \frac{\|\nabla\rho(x_k)\|^4 \|x_k\|^4}{\|\nabla\rho(x_{k-i-1})\|^4 \|x_{k-i-1}\|^4}$ , and therefore

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

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

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

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  $\lim_{k \rightarrow \infty} \inf \|x_k\| \|\nabla \rho(x_k)\| = 0$ . This means there exists a subsequence of  $\{x_k\}$ , denoted as  $\{x_{k_j}\}$ , such that

$$\lim_{j \rightarrow \infty} \|x_{k_j}\| \|\nabla \rho(x_{k_j})\| = 0.$$

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  $\lim_{k \rightarrow \infty} \rho(x_k) = \lambda^*$ . Consequently, we also have  $\lim_{j \rightarrow \infty} \rho(x_{k_j}) = \lambda^*$ . Our goal is to show that  $\lambda^*$  is an eigenvalue. Assume by contradiction that  $\lambda^*$  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

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

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

Finally, we show the eigenvector convergence of a subsequence of the iterates. Let  $\{x_{k_j}\}$  be a subsequence such that  $\lim_{j \rightarrow \infty} \|x_{k_j}\| \|\nabla \rho(x_{k_j})\| = 0$ . Assume by contradiction that there is a  $\delta > 0$  independent of the iteration count, such that for any  $M > 0$ , there exists an  $m > M$  such that  $\angle(x_{k_m}, \text{null } T(\lambda_\ell)) \geq \delta$ . It follows that

$$\lim_{j \rightarrow \infty} \|x_{k_j}\| \|\nabla \rho(x_{k_j})\| = \lim_{j \rightarrow \infty} \frac{2 \|x_{k_j}\| \|T(\lambda_\ell) x_{k_j}\|}{|x_{k_j}^* T'(\lambda_\ell) x_{k_j}|} \neq 0,$$

contradicting the established result. Therefore there exists a subsequence  $\{x_{k_j}\}$  of the CG iterates, such that  $\lim_{j \rightarrow \infty} \angle(x_{k_j}, \text{null } T(\lambda_\ell)) = 0$ . The theorem is thus established.  $\square$

*Remark.* Note in particular that if  $x_0$  is such that  $\rho(x_0)$  is smaller than the second lowest eigenvalue, then Algorithm 1 converges to the lowest eigenvalue.

**3.1.2. CG with exact line search (Rayleigh-Ritz 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, however, 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.

**Theorem 3.7.** *Algorithm 2 performs exact line search. That is, for each step  $k$ , the new iterate  $x_{k+1} = x_k + \alpha_k p_k$  is achieved with the optimal step size  $\alpha_k = \arg \min_{\alpha} \rho(x_k + \alpha p_k)$ .*

---

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 $k = 0, 1, \dots$ , 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 $(\nu_1^{(k)}, w_1^{(k)})$ .
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

---

*Proof.* In Step 4, we project  $T(\cdot)$  onto  $\text{range}(U_k) = \text{span}\{x_k, p_k\}$ , solve the projected eigenproblem  $T_p^{(k)}(\nu)w \equiv U_k^T T(\nu) U_k w = 0$  of size  $2 \times 2$  for the lowest primitive Ritz pair  $(\nu_1^{(k)}, w_1^{(k)})$ , 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  $\rho(x_{k+1}) = \nu_1^{(k)}$ . 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  $(x_{k+1}^T T(\rho) x_{k+1}) v = ([1 \ \alpha_k] T_p(\rho) [1 \ \alpha_k]^T) v = 0$ . By the Cauchy Interlacing Theorem,  $\nu_1^{(k)} \leq \rho(x_k + \alpha p_k) \leq \nu_2^{(k)}$  for any step size  $\alpha$ , i.e.,  $\nu_1^{(k)} = \min_{\alpha} \rho(x_k + \alpha p_k)$ . Since  $\nu_1^{(k)} = \rho(x_{k+1})$  is achieved, Algorithm 2 must perform exact line search, which gives the optimal step size  $\alpha_k^* = (e_2^T w_1^{(k)}) / (e_1^T w_1^{(k)})$  from the Rayleigh-Ritz projection.  $\square$

Algorithm 2 is highly favorable from the optimization point of view, as it performs exact line search achieved by the Rayleigh-Ritz projection for modest cost. Such a nice property is guaranteed by the nonlinear variational principle and the Cauchy Interlacing Theorem. In the general setting of unconstrained optimization, by contrast, the inexact line search satisfying variants of Wolfe conditions is the most time-consuming computation for CG methods.

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

- (i)  $p_k^T \nabla \rho(x_{k+1}) = 0$ ,
- (ii)  $p_k^T \nabla \rho(x_k) = -\|x_k\| \|\nabla \rho(x_k)\|^2$ ,
- (iii)  $\|p_k\|^2 = \|x_k\|^2 \|\nabla \rho(x_k)\|^2 + \beta_k^2 \|p_{k-1}\|^2$ ,
- (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  $\alpha_k$  is obtained by minimizing  $\rho(x_k + \alpha p_k)$  with respect to  $\alpha$ . 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  $\beta_k$ .  $\square$

Note that the only difference between Algorithms 1 and 2 is that the latter replaces the inexact line search with the exact line search, which automatically satisfies the strong Wolfe conditions. Therefore the global convergence of Algorithm 2 can be established directly from that of Algorithm 1 (Theorem 3.6), as follows.

**Theorem 3.9.** *Under the same assumptions as in Theorem 3.6, there is a  $\lambda_\ell$  ( $1 \leq \ell \leq n$ ) such that Algorithm 2 generates a sequence of iterates  $\{x_k\}$  satisfying  $\lim_{k \rightarrow \infty} \rho(x_k) = \lambda_\ell$ , and there is a subsequence  $\{x_{k_j}\}$  satisfying*

$$\lim_{j \rightarrow \infty} \angle(x_{k_j}, \text{null } T(\lambda_\ell)) = 0.$$

*Remark.* In practice, we find that Algorithm 2 consistently converges more robustly than Theorem 3.9 indicates. That is, it does not stagnate at a higher eigenvalue (a saddle point of the Rayleigh functional  $\rho(x)$ ) and always successfully converges to the lowest one. We do not have a complete understanding of this favorable behavior, as a saddle point may still be a point of attraction for CG-type methods for a general nonlinear optimization problem. We speculate that this is due to the use of exact line search, but to the best of our knowledge, no convincing argument has been given for the strong global convergence to the minimum of Rayleigh quotient or functional  $\rho(x)$ .

**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 for  $T(\cdot)$ , on which the  $n$  eigenvalues satisfy the variational principle (2.2), and that  $\lambda_1$  is simple, i.e.,  $\lambda_1 < \lambda_2 \leq \dots \leq \lambda_n$ . Let  $v_1$  be the eigenvector associated with  $\lambda_1$ ,  $x_k$  be an approximation to  $v_1$ , and  $v_1$  and  $x_k$  are normalized such that  $v_1^T T'(\lambda_1) v_1 = 1$  and  $x_k^T T'(\rho_k) x_k = 1$ , where  $\rho_k = \rho(x_k)$ .

The main point we want to show is that the asymptotic behavior of nonlinear CG for computing  $(\lambda_1, v_1)$  is very similar to linear CG for solving a positive semi-definite linear system. Therefore, the well-known convergence rate estimate of linear CG is asymptotically descriptive of nonlinear CG performing Rayleigh functional minimization. The intuition for this observation has been discussed in [2, Chapter 12.3] for linear Hermitian eigenproblems.

We show our results in two steps. First, we derive a quadratic function  $Q_\beta(\cdot)$  centered at the eigenvector approximation  $x_k \approx v_1$ , which provides a good approximation to the Rayleigh functional near  $x_k$ . Then we compare one step of nonlinear GC for minimizing  $\rho(\cdot)$  and linear CG for minimizing  $Q_\beta(\cdot)$ , starting at  $x_k$  and using the same search direction  $p_k$ , and show that the new iterates  $x_{k+1}$  generated by the two methods are very close to each other.

Let us first find a quadratic function that approximates  $\rho(\cdot)$  near  $x_k$ . To this end, it is natural to consider the second order Taylor expansion of  $\rho(x)$  at  $x_k$ , that is,

$$\rho(x) \approx Q_\alpha(\Delta x) = \frac{1}{2} \Delta x^T \nabla^2 \rho(x_k) \Delta x + \nabla \rho(x_k)^T \Delta x + \rho(x_k),$$

where  $\Delta x = x - x_k$ . The minimizer of  $Q_\alpha(\cdot)$ ,  $x^* = x_k + \Delta x^*$ , satisfies  $\nabla^2 \rho(x_k) \Delta x^* = -\nabla \rho(x_k)$ . However, from the expressions (3.1) and (3.2) for  $\nabla \rho$  and  $\nabla^2 \rho$ , it is not hard to show that the solution for this equation is  $\Delta x^* = -x_k$ , and thus  $x^* = x_k + \Delta x^* = 0$ , which does not yield any improvement in the eigenvector direction. Therefore, we have to consider a different quadratic function to approximate  $\rho(\cdot)$ .

This difficulty can be overcome by approximating the Hessian  $\nabla^2 \rho(x_k)$  and the gradient  $\nabla \rho(x_k)$  on an  $(n-1)$ -dimensional space. The motivation for such an approximation is to enforce a meaningful correction  $\Delta x$  by requiring that  $\Delta x$  not be very close to  $x_k$  in direction. This is precisely the idea for constructing the Jacobi-Davidson correction equation [28, 29].

**Lemma 3.10.** *Consider the quadratic function*

$$(3.14) \quad Q_\beta(x) = -\frac{2}{x_k^T T'(\rho_k) x_k} \left( \frac{1}{2} \Delta x^T P_k^T T(\rho_k) P_k \Delta x + x_k^T T(\rho_k) P_k \Delta x \right) + \rho(x_k),$$

where  $\Delta x = x - x_k$ , and  $P_k = I - \frac{x_k x_k^T T'(\rho_k)}{x_k^T T'(\rho_k) x_k}$  is an oblique projection. Then,

$$\begin{aligned} Q_\beta(x_k) &= \rho(x_k), & \nabla Q_\beta(x_k) &= \nabla \rho(x_k), & \text{and} \\ \nabla^2 Q_\beta(x_k) &= P_k^T \left( \nabla^2 \rho(x_k) + \frac{x_k^T T''(\rho_k) x_k}{x_k^T T'(\rho_k) x_k} \nabla \rho(x_k) \nabla \rho(x_k)^T \right) P_k. \end{aligned}$$

*Proof.* First, it is very easy to see that  $Q_\beta(x_k) = \rho(x_k)$ . Also,  $\nabla Q_\beta(x_k)$  can be obtained by differentiating  $Q_\beta(x)$  with respect to  $\Delta x$  and letting  $\Delta x = 0$ . Straightforward derivation shows that  $\nabla Q_\beta(x_k) = P_k^T \nabla \rho(x_k) = \nabla \rho(x_k)$ .

To derive  $\nabla^2 Q_\beta(x_k)$ , we differentiate  $Q_\beta(x)$  twice with respect to  $\Delta x$  and let  $\Delta x = 0$ . Note that  $x_k^T T'(\rho_k) P_k = 0$  and  $P_k^T T'(\rho_k) x_k = 0$ . Consequently,

$$\begin{aligned} \nabla^2 Q_\beta(x_k) &= -\frac{2}{x_k^T T'(\rho_k) x_k} P_k^T T(\rho_k) P_k \\ &= -\frac{2}{x_k^T T'(\rho_k) x_k} P_k^T (T(\rho_k) + \nabla \rho(x_k) x_k^T T'(\rho_k) + T'(\rho_k) x_k \nabla \rho(x_k)^T) P_k \\ &= P_k^T \left( \nabla^2 \rho(x_k) + \frac{x_k^T T''(\rho_k) x_k}{x_k^T T'(\rho_k) x_k} \nabla \rho(x_k) \nabla \rho(x_k)^T \right) P_k \quad (\text{see (3.2)}), \end{aligned}$$

which establishes the desired result. In addition, note that  $\|\nabla \rho(x)\| = \mathcal{O}(\sin \theta_k)$  (see Proposition 3.2), and we have  $\nabla^2 Q_\beta(x_k) = P_k^T \nabla^2 \rho(x_k) P_k + \mathcal{O}(\sin^2 \theta_k)$ .  $\square$

Lemma 3.10 shows that  $Q_\beta(x)$  is a good approximation to  $\rho(x)$  near  $x_k$  for all  $\Delta x = x - x_k$  lying in  $\text{range}(P_k) = \text{span}\{(T'(\rho_k) x_k)^\perp\}$ . Thus the minimizer for  $Q_\beta(\cdot)$  can be obtained by solving  $\nabla Q_\beta(x) = 0$ , i.e.,  $\nabla^2 Q_\beta(x_k) \Delta x + \nabla Q_\beta(x_k) = 0$ . This equation is equivalent to

$$(3.15) \quad H_k \Delta x \equiv - \left( I - \frac{T'(\rho_k) x_k x_k^T}{x_k^T T'(\rho_k) x_k} \right) T(\rho_k) \left( I - \frac{x_k x_k^T T'(\rho_k)}{x_k^T T'(\rho_k) x_k} \right) \Delta x = T(\rho_k) x_k.$$

Note that (3.15) is a Jacobi-Davidson correction equation, to which the exact solution is

$$(3.16) \quad \Delta x^* = -x_k + \frac{x_k^T T'(\rho_k) x_k}{x_k^T T'(\rho_k) T(\rho_k)^{-1} T'(\rho_k) x_k} T(\rho_k)^{-1} T'(\rho_k) x_k.$$

That is, the minimizer of  $Q_\beta(\cdot)$  is  $x^* = x_k + \Delta x^* = T(\rho_k)^{-1} T'(\rho_k) x_k$  up to a scaling factor, which is exactly the new iterate computed by one step of Rayleigh functional iteration (RFI) starting with  $x_k$  [26, Chapter 4.3]. For Hermitian eigenproblems, the local convergence of RFI is cubic; that is,  $\sin \angle(x^*, v_1) = \mathcal{O}(\sin^3 \angle(x_k, v_1))$  for all  $\angle(x_k, v_1)$  sufficiently small. In our setting, equation (3.15) is solved by CG approximately, which generates a sequence of iterates  $\Delta x_1, \Delta x_2, \dots$  that converges towards the exact solution  $\Delta x^*$  (3.16).



Our main interest is to explore the connection between the behaviors of CG for the minimization of  $Q_\beta(\cdot)$  and that of  $\rho(\cdot)$ . To this end, we first present a lemma on the positive semi-definiteness of the matrix  $H_k$  (3.15).

**Lemma 3.11.** *Let  $J = (a, b)$  be finite and of positive type for  $T(\cdot)$ ,  $\lambda_1$  be the lowest simple eigenvalue of  $T(\cdot)$ , and  $v_1$  be the corresponding eigenvector. If  $\angle(x_k, v_1)$  is sufficiently small, then the coefficient matrix  $H_k$  of (3.15) is positive semi-definite.*

*Proof.* To show the semi-definiteness of

$$H_k = - \left( I - \frac{T'(\rho_k)x_k x_k^T}{x_k^T T'(\rho_k)x_k} \right) T(\rho_k) \left( I - \frac{x_k x_k^T T'(\rho_k)}{x_k^T T'(\rho_k)x_k} \right),$$

note that  $\mathbb{R}^n = \text{span}\{x_k\} \oplus \text{span}\{(T'(\rho_k)x_k)^\perp\}$ . Since  $x_k \in \text{null}(P_k)$ ,  $x_k^T H_k x_k = 0$ , it is thus sufficient to show that  $x^T H_k x > 0$  for any nonzero  $x \in \text{span}\{(T'(\rho_k)x_k)^\perp\}$ .

Since  $x_k \not\perp T'(\rho_k)x_k$  (in fact  $x_k^T T'(\rho_k)x_k = 1$  by assumption), we define

$$\varphi_{\min} = \angle(x_k, \text{span}\{(T'(\rho_k)x_k)^\perp\})$$

such that for any nonzero  $x \in \text{span}\{(T'(\rho_k)x_k)^\perp\}$ ,  $0 < \varphi_{\min} \leq \varphi = \angle(x_k, x)$ . Suppose that  $x_k$  is sufficiently close to  $v_1$  in direction, i.e.,  $\theta_k = \angle(x_k, v_1) < \varphi_{\min}/C$  for a sufficiently large constant  $C > 0$ , such that  $|\rho_k - \lambda_1| = \mathcal{O}(\tan^2 \theta_k)$ ; see [27]. It follows that for any  $x \in \text{span}\{(T'(\rho_k)x_k)^\perp\}$ ,

$$0 < \varphi_{\min} < \frac{C(\varphi_{\min} - \theta_k)}{C - 1} \leq \frac{C(\angle(x, x_k) - \angle(x_k, v_1))}{C - 1} \leq \frac{C\angle(x, v_1)}{C - 1},$$

and therefore

$$\begin{aligned} |\rho_k - \lambda_1| &= \mathcal{O}(\tan^2 \theta_k) < \mathcal{O}(\tan^2 \frac{\varphi_{\min}}{C}) \\ &\leq \mathcal{O}\left(\tan^2 \frac{\angle(x, v_1)}{C-1}\right) \leq |\rho(x) - \lambda_1| = \mathcal{O}(\tan^2 \angle(x, v_1)), \end{aligned}$$

because  $C$  is sufficiently large. Since  $\rho(x) \geq \lambda_1$ , we must have  $\rho(x) > \rho_k > \lambda_1$ , and thus  $x^T H_k x = -x^T T(\rho_k)x = -\frac{(\rho_k - \rho(x))x^T T(\rho_k)x}{\rho_k - \rho(x)} > 0$  for all nonzero  $x \in \text{span}\{(T'(\rho_k)x_k)^\perp\}$  because  $J$  is of positive type for  $T(\cdot)$ . This shows that  $H_k$  is positive semi-definite.  $\square$

Our final step is to compare one step of nonlinear CG for minimizing  $\rho(\cdot)$  and linear CG for minimizing  $Q_\beta(\cdot)$ , starting at  $x_k$  and using the same search direction  $p_k$ . Our main result in this section is presented in the following theorem.

**Theorem 3.12.** *Let  $J = (a, b)$  be finite and of positive type for  $T(\cdot)$ ,  $\lambda_1$  be the lowest simple eigenvalue of  $T(\cdot)$ , and  $v_1$  be the corresponding eigenvector. Assume that  $x_k$  is a good approximation to  $v_1$  such that  $\sin \theta_k = \angle(x_k, v_1)$  is sufficiently small. Consider one step of linear CG for minimizing  $Q_\beta(\cdot)$  and nonlinear CG for minimizing  $\rho(\cdot)$ , starting at  $x_k$ , going along the same search direction  $p_k$ , and using exact line search. Assume that the search direction is such that  $\angle(x_k, p_k) \geq \omega_k$  for some  $\omega_k \gg \theta_k > 0$ . Then, the new iterates  $x_{k+1}^{LCG}$  and  $x_{k+1}^{NCG}$  generated by linear and nonlinear CG, respectively, satisfy  $\|x_{k+1}^{NCG} - x_k\| / \|x_{k+1}^{LCG} - x_k\| = 1 + \mathcal{O}(\sin \theta_k)$ .*

*Proof.* First, we explore how linear CG performs in one step for the minimization of  $Q_\beta(\cdot)$ . From (3.15), it is not hard to show that

$$H_k = - \left( T(\rho_k) + \frac{T'(\rho_k)x_k}{2} \nabla \rho(x_k) + \nabla \rho(x_k) \frac{x_k^T T'(\rho_k)}{2} \right) = -T(\rho_k) + \mathcal{O}(\sin \theta_k),$$

which is a small perturbation of  $-T(\rho_k)$  if  $\theta_k = \angle(x_k, v_1)$  is small. Let  $x_k$  be the current CG iterate, and  $p_k$  be the *normalized* search direction to minimize  $Q_\beta(\cdot)$ . By assumption,  $\angle(x_k, p_k) \geq \omega_k$  for some  $\omega_k \gg \theta_k > 0$ . Following the idea with which  $\rho(x) > \rho_k$  is established in the proof of the semi-definiteness of  $H_k$ , one can show similarly that  $\rho(p_k) > \rho_k$  and  $\rho(p_k)$  is bounded away from  $\rho_k$ . Thus, it is not difficult to see that  $p_k^T T(\rho_k) p_k = \frac{(\rho_k - \rho(p_k)) p_k^T T(\rho_k) p_k}{\rho_k - \rho(p_k)}$  is negative and is bounded away from zero. Let the new CG iterate be  $x_{k+1} = x_k + \alpha_k p_k$ . From (3.15), the exact line search condition for this step is  $T(\rho_k)x_k - H_k(\alpha_k p_k) \perp p_k$ , and therefore

$$\begin{aligned} \alpha_k \|p_k\| &= \frac{p_k^T T(\rho_k) x_k \|p_k\|}{p_k^T H_k p_k} = \frac{p_k^T T(\rho_k) x_k \|p_k\|}{p_k^T (-T(\rho_k) + \mathcal{O}(\sin \theta_k)) p_k} \\ (3.17) \quad &= -\frac{p_k^T T(\rho_k) x_k \|p_k\|}{p_k^T T(\rho_k) p_k + \mathcal{O}(\sin \theta_k)} = -\frac{p_k^T T(\rho_k) x_k \|p_k\|}{p_k^T T(\rho_k) p_k} (1 + \mathcal{O}(\sin \theta_k)). \end{aligned}$$

From (3.17), we see that the step size  $\alpha_k \|p_k\| = \|x_{k+1}^{LCG} - x_k\|$  is independent of the scaling of  $p_k$ , and it is proportional to the eigenresidual norm  $\|T(\rho_k)x_k\| = \|x_k\| \mathcal{O}(\sin \theta_k)$ ; see the proof of Proposition 3.2. Therefore, as long as  $\|x_k\|$  remains bounded, the search step size  $\alpha_k \|p_k\| \rightarrow 0$  as the initial iterate  $x_k$  approaches  $v_1$  in direction, i.e., as  $\theta_k = \angle(x_k, v_1) \rightarrow 0$ .

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

$$\begin{aligned} (3.18) \quad 0 &= -\frac{x_{k+1}^T T'(\rho_{k+1}) x_{k+1}}{2} p_k^T \nabla \rho(x_{k+1}) = p_k^T T(\rho(x_k + \alpha_k p_k)) (x_k + \alpha_k p_k) \\ &= p_k^T T \left( \rho_k + \nabla \rho(x_k)^T (\alpha_k p_k) + \frac{1}{2} (\alpha_k p_k)^T \nabla^2 \rho(x_k) (\alpha_k p_k) \right. \\ &\quad \left. + \mathcal{O}((\alpha_k \|p_k\|)^3) \right) (x_k + \alpha_k p_k) \\ &= p_k^T T(\rho_k) x_k + (p_k^T T'(\rho_k) x_k) \nabla \rho(x_k)^T (\alpha_k p_k) + p_k^T T(\rho_k) (\alpha_k p_k) \\ &\quad + \frac{1}{2} (\alpha_k p_k)^T ((p_k^T T'(\rho_k) x_k) \nabla^2 \rho(x_k) + (p_k^T T''(\rho_k) x_k) \nabla \rho(x_k) \nabla \rho(x_k)^T) (\alpha_k p_k) \\ &\quad + p_k^T T'(\rho_k) (\alpha_k p_k) \nabla \rho(x_k)^T (\alpha_k p_k) + \mathcal{O}((\alpha_k \|p_k\|)^3) \\ &= \|p_k\| (C_2 (\alpha_k \|p_k\|)^2 + C_1 (\alpha_k \|p_k\|) + C_0) + \mathcal{O}((\alpha_k \|p_k\|)^3), \end{aligned}$$

where

$$\begin{aligned} C_0 &= \frac{p_k^T T(\rho_k) x_k}{\|p_k\|} = \|x_k\| \mathcal{O}(\sin \theta_k), \\ C_1 &= \frac{p_k^T T(\rho_k) p_k}{\|p_k\|^2} + \frac{p_k^T T'(\rho_k) x_k \nabla \rho(x_k)^T p_k}{\|p_k\|^2} = \frac{p_k^T T(\rho_k) p_k}{\|p_k\|^2} + \mathcal{O}(\sin \theta_k), \\ C_2 &= \frac{(p_k^T T'(\rho_k) x_k) p_k^T \nabla^2 \rho(x_k) p_k}{2 \|p_k\|^3} + \frac{(p_k^T T''(\rho_k) x_k) p_k^T \nabla \rho(x_k) \nabla \rho(x_k)^T p_k}{2 \|p_k\|^3} \\ &\quad + \frac{p_k^T T'(\rho_k) p_k \nabla \rho(x_k)^T p_k}{\|p_k\|^3} = \frac{(p_k^T T'(\rho_k) x_k) p_k^T \nabla^2 \rho(x_k) p_k}{2 \|p_k\|^3} + \frac{1}{\|x_k\|} \mathcal{O}(\sin \theta_k). \end{aligned}$$

Note that the coefficients  $C_0 = \|x_k\| \mathcal{O}(\sin \theta_k)$ ,  $C_1 = \mathcal{O}(1)$  and  $C_2 = \mathcal{O}(1)$ , and they do not depend on the scaling of  $p_k$ . Neglecting the cubic term of  $\alpha_k \|p_k\|$  in (3.18), and using the root formula for quadratic equations, we have the step size

$$(3.19) \quad \alpha_k \|p_k\| = \frac{-C_1 + \sqrt{C_1^2 - 4C_0C_2}}{2C_2} = \frac{-C_1 + C_1 \left(1 - \frac{2C_0C_2}{C_1^2} + \mathcal{O}\left(\frac{C_0^2C_2^2}{C_1^4}\right)\right)}{2C_2} \\ = -\frac{C_0}{C_1} \left(1 + \mathcal{O}\left(\frac{C_0C_2}{C_1^2}\right)\right) = -\frac{p_k^T T(\rho_k) x_k \|p_k\|}{p_k^T T(\rho_k) p_k} (1 + \mathcal{O}(\sin \theta_k)),$$

for  $\theta_k$  sufficiently small. Again, the step size  $\alpha_k \|p_k\| = \|x_{k+1}^{NCG} - x_k\|$  is independent of the scaling of  $p_k$ .

Therefore, the dominant terms of the search step size shown in (3.17) for linear CG and in (3.19) for nonlinear CG are identically  $-\frac{p_k^T T(\rho_k) x_k \|p_k\|}{p_k^T T(\rho_k) p_k}$ , and thus the conclusion of the theorem,  $\|x_{k+1}^{NCG} - x_k\| / \|x_{k+1}^{LCG} - x_k\| = 1 + \mathcal{O}(\sin \theta_k)$ , is established.  $\square$

In other words, under the conditions described in Theorem 3.12, the new iterates  $x_{k+1}^{LCG}$  and  $x_{k+1}^{NCG}$  generated by linear and nonlinear CG become increasingly close to each other asymptotically as  $\theta_k \rightarrow 0$ . This suggests that nonlinear CG for computing the lowest eigenvalue could converge as rapidly as linear CG for solving the semi-definite system (3.15). In particular, CG converges much more quickly than the steepest descent method for solving Hermitian eigenproblems. Such a superiority has been observed extensively in practice.

#### 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 (2.2) or (2.3). Without loss of generality, assume that  $J = (a, b)$  is of positive type for  $T(\cdot)$  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, 22, 23] 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  $\beta_k$  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., [19, 24]. 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-valued function  $\chi(\cdot, \cdot)$  defined in (2.4). One would consider performing Gram-Schmidt orthogonalization with respect to this function, but this is not viable as  $\chi(\cdot, \cdot)$  is generally *not bilinear*. 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  $T(\lambda)v = 0$  [13].

---

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  $x_0 \notin \text{span}\{v_1, \dots, v_j\}$ .
  4. For  $k = 0, 1, \dots$ , 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  $U_k = [v_1 \dots v_j \ x_k \ p_k] \in \mathbb{C}^{n \times (j+2)}$ , 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  $(\nu_{j+1}^{(k)}, w_{j+1}^{(k)})$ .
    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  $(\nu_{j+1}^{(k)}, w_{j+1}^{(k)})$  in Step 7, and finally sets  $x_{k+1}$  as the scaled  $(j+1)$ -st lowest Ritz vector  $U_k w_{j+1}^{(k)} / (e_{j+1}^T w_{j+1}^{(k)})$  in Step 8.

The convergence of PCG may be enhanced by enlarging the space for projection. To see this improvement, assume that the lowest eigenvalue  $\lambda_1$  is of interest. The Rayleigh-Ritz projection is now performed onto a three-dimensional space  $U_k^{\text{LOPCG}} = [x_k \ g_k \ p_{k-1}]$ , 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 [21] and discussed in detail in [22]. 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

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

---

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  $x_0 \notin \text{span}\{v_1, \dots, v_j\}$ .
  4. For  $k = 0, 1, \dots$ , 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  $U_k = [v_1 \dots v_j \ x_k \ g_k]$ ; otherwise, form  $U_k = [v_1 \dots v_j \ x_k \ g_k \ p_{k-1}]$ . Normalize each column of  $U_k$ , and set  $r = \text{number of columns of } U_k$ .
    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  $(\nu_{j+1}^{(k)}, w_{j+1}^{(k)})$ .
    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

---

(see Step 5 of Algorithm 3), and thus

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

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  $\text{span}\{g_k, p_{k-1}\}$ , which contains the traditional search direction  $p_k = g_k + \beta_k p_{k-1}$  independently of the value of  $\beta_k$ . 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 needs 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.

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 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  $k = 0, 1, \dots$ , 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  $U_k = [v_1 \dots v_j \ X_k e_{j+1} \dots X_k e_m \ P_k e_{j+1} \dots P_k e_m] \in \mathbb{C}^{n \times (2m-j)}$ , 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  $(\nu_{j+1}^{(k)}, w_{j+1}^{(k)}), \dots, (\nu_m^{(k)}, w_m^{(k)})$ .
6. Update  $X_{k+1}$  s.t.  $X_{k+1} e_\ell = U_k \frac{w_\ell^{(k)}}{e_\ell^T w_\ell^{(k)}} e_\ell$  ( $j+1 \leq \ell \leq m$ ); 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

---



---

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  $k = 0, 1, \dots$ , until convergence, i.e.,  $j = m$ .
3. Set  $G_k \in \mathbb{R}^{n \times m}$  s.t.  $G_k e_\ell = -\|X_k e_\ell\| M^{-1} \nabla \rho(X_k e_\ell)$  ( $j+1 \leq \ell \leq m$ ).
4. If  $k=0$ , form  $U_k = [X_k \ G_k] \in \mathbb{R}^{n \times 2m}$ ; otherwise form  $U_k = [v_1 \dots v_j \ X_k e_{j+1} \dots X_k e_m \ G_k e_{j+1} \dots G_k e_m \ P_{k-1} e_{j+1} \dots P_{k-1} e_m] \in \mathbb{R}^{n \times (3m-2j)}$ . 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  $(\nu_{j+1}^{(k)}, w_{j+1}^{(k)}), \dots, (\nu_m^{(k)}, w_m^{(k)})$ .
6. Update  $P_k$  s.t.  $P_k e_\ell = \sum_{i=j+1}^r (e_i^T w_\ell^{(k)}) U_k e_i$  ( $j+1 \leq \ell \leq m$ ).
7. Update  $X_{k+1}$  s.t.  $X_{k+1} e_\ell = U_k w_\ell^{(k)} = \sum_{i=1}^m (e_i^T w_\ell^{(k)}) U_k e_i + P_k e_\ell$  ( $j+1 \leq \ell \leq m$ ); normalize each column of  $X_{k+1}$ .
8. 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  $j = \ell$ .
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 [8] 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  $v_\ell$  corresponding to a high eigenvalue  $\lambda_\ell$  ( $\ell > 1$ ) must be indefinite, and it is highly indefinite if  $\ell \gg 1$ , i.e.,  $\lambda_\ell$  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 semi-definiteness of the Hessian  $H_0 \approx \nabla^2 \rho(x_0)$  defined in (3.15), 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 [8]. 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  $1 \leq i \leq m$ . The experiments are performed on an iMac desktop computer running Mac OS X 10.8.5, MATLAB R2014b, with a 2.9 GHz Intel Core i5 processor and 16GB 1600 MHz DDR3 memory.

TABLE 1. Description of the test problems

name	type	dimension	interval	end of interest
<i>wiresaw</i>	quadratic	1024	(0, 3250)	lowest
<i>gen_hyper2</i>	quadratic	4096	(−843, 0.3943)	highest
<i>sleep</i>	quadratic	16384	(−16.33, −1.61)	lowest
<i>loaded_string</i>	rational	10000	(4.4, $1.2 \times 10^9$ )	lowest
<i>pdde</i>	nonlinear	39601	(−20.87, 4.08)	highest
<i>artificial</i>	nonlinear	16129	(−0.43, 3.34)	highest

We choose three quadratic, one rational, and two truly nonlinear problems, and summarize them in Table 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 (2.2) or (2.3) in its original form, but it can be easily transformed to a Hermitian eigenproblem satisfying the variational principle by substituting  $\lambda$  with  $i\lambda$ . The transformed problem has 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.hyper2* 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  $I_\ell$  and  $I_r$ . We aim at solving the highest eigenvalues close to the right boundary of the left interval  $I_\ell$ . 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 (2.2) or (2.3) 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 (2.2) or (2.3) 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  $200 \times 200$  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  $128 \times 128$  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 (2.2) or (2.3) 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  $\sigma$  is the endpoint of the interval described in Table 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  $\mu$  is the midpoint of the last two newly computed distinct eigenvalues.

**5.1. Comparison between variants of PCG methods.** To compare different variants of PCG, we use the same starting vectors generated by the MATLAB `randn(n,m)` function initialized with the `RandStream` subroutine using seed 1. The Rayleigh functional  $\rho = \rho(x)$ , which is the unique solution of  $x^*T(\rho)x = 0$  on the interval  $J = (a, b)$ , is computed by the MATLAB `roots` function for polynomial  $T(\cdot)$ , or by `fzero(fun, [a,b])` otherwise, where `fun` is the functional handle to the mapping  $\mu \rightarrow x^*T(\mu)x$ .

The performance of PCG methods is presented in Table 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 4.76 seconds to find the lowest 10 eigenvalues. Similarly, PCG takes 319, 910, and 2682 preconditioned matrix-vector products, and 15.29, 55.10, and 229.84 seconds to compute the lowest 20, 40, and 80 eigenvalues, respectively. Using variable preconditioning, it takes fewer preconditioned matrix-vector products, but also incurs mild additional cost for updating preconditioners. Nevertheless, it takes only 1412 preconditioned matrix-vector products and 145.09 seconds (about 40% faster) to find the lowest 80 eigenvalues. Note that variable preconditioning is not enabled for the computation of the 10 extreme eigenvalues, and therefore the performance is not given. We also note that some methods failed to converge for certain problems in the maximum number of iterations, and such a failure is marked as  $\infty$ . For example, with fixed preconditioning, single-vector and block methods did not find the 80 highest eigenvalues of the problem *gen.hyper2* in  $500 \times 80 = 40000$  and 500 iterations, respectively. Variable preconditioning is not enabled for the problem *sleepy*, 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 for solving linear eigenproblems.

- Block PCG methods are significantly more competitive for computing multiple eigenvalues than single-vector methods in arithmetic cost and CPU time. 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.

TABLE 2. Performance of PCG methods

$m$	10	20	40	80
<i>wiresaw</i>				
PCG (fixed pcd)	113 (4.76s)	319 (15.29s)	910 (55.10s)	2682 (229.84s)
PCG (variable pcd)	—	282 (17.85s)	639 (52.75s)	1412 (145.09s)
LOPCG (fixed pcd)	109 (3.32s)	309 (11.05s)	891 (42.74s)	2729 (202.25s)
LOPCG (variable pcd)	—	262 (12.45s)	590 (37.47s)	1317 (115.26s)
BPCG (fixed pcd)	82 (3.63s)	182 (7.70s)	413 (19.12s)	897 (47.89s)
BPCG (variable pcd)	—	176 (8.10s)	379 (19.03s)	774 (39.63s)
LOBPCG (fixed pcd)	65 (3.07s)	138 (6.63s)	298 (16.65s)	629 (44.54s)
LOBPCG (variable pcd)	—	132 (6.82s)	276 (16.68s)	565 (36.66s)
<i>gen_hyper2</i>				
PCG (fixed pcd)	2813 (5.44s)	7434 (18.61s)	20521 (92.60s)	$\infty$
PCG (variable pcd)	—	3154 (6.47s)	3584 (9.04s)	5813 (42.93s)
LOPCG (fixed pcd)	2963 (5.41s)	9208 (22.66)	27648 (120.72s)	$\infty$
LOPCG (variable pcd)	—	3269 (6.34s)	3664 (8.75s)	5472 (38.30s)
BPCG (fixed pcd)	684 (1.30s)	2146 (4.77s)	3089 (10.32s)	$\infty$
BPCG (variable pcd)	—	1645 (3.30s)	1967 (4.79s)	8806 (33.96s)
LOBPCG (fixed pcd)	451 (1.02s)	1030 (2.60s)	1563 (6.46s)	$\infty$
LOBPCG (variable pcd)	—	859 (2.10s)	1093 (3.85s)	3547 (34.30s)
<i>sleepier</i>				
PCG (fixed pcd)	847 (7.45s)	1367 (14.38s)	2028 (28.21s)	3317 (101.50s)
LOPCG (fixed pcd)	942 (7.20s)	1438 (13.12s)	2086 (32.27s)	3379 (111.32s)
BPCG (fixed pcd)	457 (3.68s)	558 (5.33s)	633 (8.39s)	893 (16.37s)
LOBPCG (fixed pcd)	251 (2.24s)	305 (3.97s)	397 (8.07s)	675 (21.67s)
<i>loaded_string</i>				
PCG (fixed pcd)	90 (0.61s)	513 (3.72s)	1028 (11.64s)	3543 (102.52s)
PCG (variable pcd)	—	208 (2.14s)	502 (7.61s)	1104 (31.35s)
LOPCG (fixed pcd)	83 (0.52s)	240 (1.96s)	733 (10.50s)	2251 (69.77s)
LOPCG (variable pcd)	—	196 (1.76s)	465 (6.87s)	1038 (31.95s)
BPCG (fixed pcd)	59 (0.62s)	136 (2.50s)	304 (8.48s)	676 (41.26s)
BPCG (variable pcd)	—	132 (2.45s)	290 (7.42s)	609 (30.74s)
LOBPCG (fixed pcd)	52 (0.50s)	109 (1.10s)	239 (3.15s)	518 (12.32s)
LOBPCG (variable pcd)	—	105 (1.09s)	225 (3.05s)	464 (10.32s)
<i>pdde</i>				
PCG (fixed pcd)	899 (61.80s)	1270 (118.93s)	2185 (372.62s)	4661 (1995.16s)
PCG (variable pcd)	—	1253 (111.12s)	1986 (314.10s)	2974 (969.48s)
LOPCG (fixed pcd)	169 (12.91s)	512 (64.05s)	1397 (304.54s)	4298 (2189.74s)
LOPCG (variable pcd)	—	353 (44.58s)	685 (143.42s)	1424 (653.56s)
BPCG (fixed pcd)	501 (19.09s)	937 (32.55s)	1769 (75.96s)	1992 (147.40s)
BPCG (variable pcd)	—	937 (33.61s)	1765 (77.73s)	1958 (141.18s)
LOBPCG (fixed pcd)	109 (6.05s)	200 (13.36s)	436 (40.87s)	823 (99.41s)
LOBPCG (variable pcd)	—	191 (12.98s)	377 (33.45s)	701 (75.53s)
<i>artificial</i>				
PCG (fixed pcd)	145 (5.78s)	503 (36.18s)	1777 (243.13s)	$\infty$
PCG (variable pcd)	—	455 (37.02s)	839 (103.17s)	$\infty$
LOPCG (fixed pcd)	135 (5.16s)	523 (37.65s)	1837 (254.58s)	$\infty$
LOPCG (variable pcd)	—	356 (26.07s)	722 (88.09s)	$\infty$
BPCG (fixed pcd)	102 (2.08s)	325 (8.06s)	696 (19.03s)	$\infty$
BPCG (variable pcd)	—	309 (7.44s)	647 (17.43s)	$\infty$
LOBPCG (fixed pcd)	91 (2.39s)	239 (8.17s)	468 (17.69s)	$\infty$
LOBPCG (variable pcd)	—	214 (7.04s)	401 (14.32s)	$\infty$

- 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  $\mathcal{O}(9m^3)$  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 *sleeper*.
- 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 *gen.hyper2*, the failure of convergence was fixed by variable preconditioning.
- Overall, 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 prominent advantage of LOPCG over PCG. In fact, with fixed preconditioning, LOPCG converges slower than PCG for the two quadratic problems *gen.hyper2* 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.

**5.2. Comparison between PCG and PCG+ $\ell$  methods.** To further demonstrate the performance of the new PCG methods, we compare them with PCG+ $\ell$  algorithms, which include all search directions constructed in the previous  $\ell$  iterations into the search subspace. As we explained, the Rayleigh-Ritz procedure essentially performs exact search for the minimizer of the Rayleigh functional value in the whole search subspace. Therefore, with extended search subspaces, PCG+ $\ell$  methods are expected to outperform the regular PCG methods in terms of total counts of preconditioned matrix vector products.

We compare LOPCG, PCG+3, PCG+9, PCG+15 and LOBPCG, for computing the 10 extreme eigenvalues of each problem described in Table 1. One sees from Table 3 that PCG+ $\ell$  methods exhibit accelerated convergence, taking fewer preconditioned matrix-vector product than LOPCG. The improvement is particularly obvious for problems *gen.hyper2* and *sleeper*, for which LOPCG converges slowly. The slow convergence is caused by our choice of the preconditioner  $M = T(\sigma)$  with a  $\sigma$  relatively far from the desired eigenvalues. It seems that PCG+ $\ell$  variants are most attractive when single-vector PCG methods working with a weak preconditioner take many iterations to converge. On the other hand, we note that using larger  $\ell$  reduces the iteration counts, but not necessarily the CPU time. In fact, PCG+15 always runs slower than PCG+9 for our test problems. MATLAB profiling shows that this is primarily because the former takes more CPU time managing the extended search subspace and solving the larger Rayleigh-Ritz projected eigenproblems. We also see from Table 3 that single-vector PCG+ $\ell$  methods are not

as competitive as LOBPCG in terms of CPU time, but note that they are more memory-efficient.

TABLE 3. Comparison between PCG+ $\ell$  methods and LOBPCG

problem	LOPCG	PCG+3	PCG+9	PCG+15	LOBPCG
<i>wiresaw</i>	109 (3.32s)	93 (3.01s)	93 (3.08s)	93 (3.14s)	65 (3.07s)
<i>gen_hyper2</i>	2963 (5.41s)	703 (1.62s)	416 (1.21s)	355 (1.30s)	451 (1.02s)
<i>sleeper</i>	942 (7.20s)	487 (4.50s)	418 (4.75s)	410 (5.56s)	251 (2.24s)
<i>loaded_string</i>	83 (0.52s)	74 (0.45s)	74 (0.46s)	74 (0.50s)	52 (0.50s)
<i>pdde</i>	169 (12.91s)	144 (11.41s)	142 (11.83s)	144 (12.52s)	109 (6.05s)
<i>artificial</i>	135 (5.16s)	124 (4.99s)	124 (5.18s)	124 (5.38s)	91 (2.39s)

We also tested block PCG+ $\ell$  methods on these eigenproblems, but found them no more attractive than LOBPCG. Specifically, compared to LOBPCG, block PCG+ $\ell$  with  $\ell = 2$  or 3 reduce the preconditioned matrix-vector product counts marginally or mildly, but they run essentially as rapidly as LOBPCG at best; further increasing  $\ell$  no longer decreases the iteration counts, but incurs considerably more CPU time. This is again because of the rapidly increasing arithmetic cost associated with managing a larger block search subspace and solving larger projected eigenproblems. Nevertheless, if available preconditioners are *very weak* such that LOBPCG converges fairly slowly, then there may be some chance for block PCG+ $\ell$  methods to be more attractive. Overall, we consider PCG+ $\ell$  methods as complements, not competitors, to the variants of PCG methods we studied, and LOBPCG in particular.

**5.3. Comparison between PCG and JD methods.** In this section, we compare variants of PCG methods and the Jacobi-Davidson (JD) methods for computing the extreme 10 eigenvalues of the test problems. This comparison provides a convincing illustration of the superior efficiency of our proposed methods, since JD is a highly competitive algorithm in this problem setting; see, e.g., [34] and references therein.

The implementation of the JD method used in our experiments is based on that described in [6]. The only difference is that our JD correction is constructed as

$$(5.1) \quad \left( I - \frac{T'(\rho_k)x_kx_k^*}{x_k^*T'(\rho_k)x_k} \right) T(\rho_k) \left( I - \frac{x_kx_k^*T'(\rho_k)}{x_k^*T'(\rho_k)x_k} \right) \Delta x_k = -T(\rho_k)x_k,$$

where  $x_k$  is the eigenvector approximation in the  $k$ -th JD iteration, and  $\rho_k = \rho(x_k)$  is the corresponding Rayleigh functional value. Note that (5.1) is mathematically equivalent to the linear system defined in the Rayleigh functional iteration (RFI); see [26] and [31]. The right-preconditioned GMRES(5) is used to solve (5.1), with the same LDL<sup>T</sup> preconditioners constructed in Section 5.1. We use a search subspace of dimension 5 and 15 (excluding the storage for converged eigenvectors), respectively, for the Rayleigh-Ritz projection performed in the outer iteration of JD. The two variants of JD are thus referred to as JD-GMRES(5) + RR(5) and JD-GMRES(5) + RR(15), respectively.

Similar to the experiments done in Section 5.1, all methods use the same starting eigenvector approximations generated by the MATLAB function `randn(n,m)` initialized with the `RandStream` function with seed 1. The relative tolerance for each desired eigenpair approximation is  $10^{-10}$  as in Section 5.1, and the same fixed definite preconditioners are used.



The experiment results are summarized in Table 4. We use *pdde* as an example to see the performance. For this problem, it takes PCG and LOPCG, respectively, 899 and 169 preconditioned matrix-vector products, and 61.80 and 12.91 seconds; similarly, it takes BPCG and LOBPCG, respectively, 501 and 109 preconditioned matrix-vector products, and 19.06 and 6.05 seconds. By contrast, it takes the two variants of JD 1350 and 275 preconditioned matrix-vector products, and 66.05 and 16.13 seconds, respectively. Note that the storage needed by the two variants of JD is about  $5 + 5 + 10 = 20$  and  $5 + 15 + 10 = 30$  vectors, respectively, the total amount needed for GMRES, the Rayleigh-Ritz procedure, and the converged eigenvectors. These numbers are considerably higher than those required in the single-vector PCG and LOPCG methods (slightly over 12 and 13 vectors), and are close to those needed for the BPCG and LOBPCG methods, depending on the implementation.

TABLE 4. Comparison between PCG methods and JD-GMRES

problem	PCG	LOPCG	BPCG	LOBPCG	JD-GMRES(5) + RR(5)	JD-GMRES(5) + RR(15)
<i>wiresaw</i>	113 (4.76s)	109 (3.32s)	82 (3.63s)	65 (3.07s)	438 (6.15s)	143 (2.62s)
<i>gen_hyper2</i>	2813 (5.44s)	2963 (5.41s)	684 (1.30s)	451 (1.02s)	$\infty$	1020 (2.05s)
<i>sleepier</i>	847 (7.45s)	942 (7.20s)	457 (3.68s)	251 (2.24s)	1185 (7.66s)	460 (3.53s)
<i>loaded_string</i>	90 (0.61s)	83 (0.52s)	59 (0.62s)	52 (0.50s)	295 (1.15s)	120 (0.62s)
<i>pdde</i>	899 (62.80s)	169 (12.91s)	501 (19.09s)	109 (6.05s)	1350 (66.05s)	275 (16.13s)
<i>artificial</i>	145 (5.78s)	135 (5.16s)	102 (2.08s)	91 (2.39s)	670 (13.74s)	200 (5.32s)

Table 4 shows that JD-GMRES(5)+RR(5) is the least efficient method, always taking the most preconditioned matrix-vector products and also the longest CPU time. This clearly indicates that JD is not efficient when working with a small search subspace for the outer iteration. This weakness is fixed in JD-GMRES(5)+RR(15), which exhibits significantly improved performance. Nevertheless, it is outperformed by LOBPCG, except for *wiresaw* in terms of CPU time. Note that this problem involves rather dense matrices, and thus the evaluation of the matrix  $T(\rho)$  takes much longer CPU time than other problems. In this case, JD-GMRES has some advantage because, in each outer iteration, the construction and solution of the correction equation (5.1) only require one evaluation of  $T(\rho)$ ; by contrast, LOBPCG requires an evaluation of  $T(\rho)$  for each eigenvector approximation in each iteration step. Overall, still, LOBPCG is clearly superior to JD-GMRES(5)+RR(15).

## 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 (2.2) or (2.3) 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, which considerably outperforms JD-GMRES methods with similar storage cost.

## APPENDIX

*Proof of Proposition 2.6.* We give the proof by mathematical induction. Assume without loss of generality that  $J$  is of positive type for  $T(\cdot)$ , so that  $T(\cdot)$  has  $n$  eigenvalues  $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ . 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 for  $T(\cdot)$ ,  $(\lambda_{k_1} - \rho(x))(x^* T(\lambda_{k_1})x) > 0$ , 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,  $(\lambda_{k_1} - \rho(v_{k_2}))(v_{k_2}^* T(\lambda_{k_1}) v_{k_2}) < 0$ , where  $\rho(v_{k_2}) = \lambda_{k_2} > \lambda_{k_1}$ . This is contradictory to the fact that  $J$  is of positive type for  $T(\cdot)$ . 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  $(\lambda_{k_{m+1}} - \rho(\tilde{x}))(\tilde{x}^* T(\lambda_{k_{m+1}})\tilde{x}) > 0$ , since  $J$  is of positive type, and thus  $0 > \tilde{x}^* T(\lambda_{k_{m+1}})\tilde{x} = (\tilde{x} - c_{m+1} v_{k_{m+1}})^* T(\lambda_{k_{m+1}})(\tilde{x} - c_{m+1} v_{k_{m+1}}) = x^* T(\lambda_{k_{m+1}})x$ . 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  $(\lambda_{k_{m+1}} - \rho(x))(x^* T(\lambda_{k_{m+1}})x) < 0$ . This is impossible since  $J$  is of positive type for  $T(\cdot)$ . 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

$$0 = \tilde{x}^* T(\rho(\tilde{x}))\tilde{x} = (\tilde{x} - c_{m+1} v_{k_{m+1}})^* T(\lambda_{k_{m+1}})(\tilde{x} - c_{m+1} v_{k_{m+1}}) = x^* T(\lambda_{k_{m+1}})x.$$

That is,  $\rho(x) = \lambda_{k_{m+1}}$ . However, by the inductive hypothesis,  $\lambda_{k_1} \leq \rho(x) \leq \lambda_{k_m}$ , and  $\lambda_{k_1} < \rho(x) < \lambda_{k_m}$  if  $\lambda_{k_1} < \lambda_{k_m}$ ,  $\rho(x) \neq \lambda_{k_{m+1}}$  unless  $\lambda_{k_1} = \lambda_{k_2} = \dots = \lambda_{k_{m+1}}$ , 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 for  $T(\cdot)$ .

The proof for  $J$  of negative type for  $T(\cdot)$  is analogous and thus is omitted.  $\square$

*Proof of Proposition 3.1.* From the definition of  $\rho(\cdot)$ ,  $x^T T(\rho)x = 0$ , and  $(x + \Delta x)^T T(\rho(x + \Delta x))(x + \Delta x) = 0$  for all  $\Delta x \in \mathbb{C}^n$ . Since  $T(\cdot)$  and  $\rho(\cdot)$  are twice continuously differentiable, we have

(6.1)

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

where

$$(6.2) \quad Q_2(\Delta x; x) = \Delta x^T T(\rho) \Delta x + \Delta x^T T'(\rho) x \nabla \rho^T \Delta x + \Delta x^T \nabla \rho x^T T'(\rho) \Delta x \\ + \frac{x^T T'(\rho) x}{2} \Delta x^T \nabla^2 \rho \Delta x + \frac{x^T T''(\rho) x}{2} \Delta x^T \nabla \rho \nabla \rho^T \Delta x$$

contains all the second order terms of  $\Delta x$ . Since (6.1) holds for all small  $\Delta x$ , both the first order terms and the second order terms of  $\Delta x$  must be identically zero. Therefore (3.1) and (3.2) follow immediately from (6.1) and (6.2), as  $x^T T'(\rho) x \neq 0$  by assumption.  $\square$

*Proof of Proposition 3.2.* First note that  $\rho(v_\ell) = \lambda_\ell$ ,  $\nabla \rho(v_\ell) = -\frac{2}{v_\ell^T T'(\lambda_\ell) v_\ell} T(\lambda_\ell) v_\ell = 0$ , and  $\nabla^2 \rho(v_\ell) = -\frac{2}{v_\ell^T T'(\lambda_\ell) v_\ell} T(\lambda_\ell)$ . Consider the Taylor expansion of  $\rho(x)$  at  $v_\ell$ ,

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

where  $v_\ell^T T'(\lambda_\ell) v_\ell \neq 0$  because  $J$  is of definite type for  $T(\cdot)$ . It follows that  $|\rho(x) - \lambda_\ell| = \mathcal{O}(\tan^2 \theta)$ .

To analyze  $\|\nabla \rho(x)\|$ , note that since  $v_\ell^T T'(\lambda_\ell) v_\ell \neq 0$ ,  $v_\ell^T T'(\lambda_\ell) v_\ell \cos^2 \theta + \mathcal{O}(\sin \theta)$  is bounded away from zero for sufficiently small  $\theta$ . Therefore,

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

*Proof of Theorem 3.4.* 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

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

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

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

It follows that

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

Then, it follows from (3.4) that

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

By the variational principle (2.2) or (2.3),  $\rho(x)$  is bounded below by  $\lambda_1$  or  $\lambda_n$  for all  $x \in \mathbb{R}^n \setminus \{0\}$ , where  $\lambda_1, \lambda_n \in J$  are finite, and (3.6) is thus established.  $\square$

*Proof of Lemma 3.5.* 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

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

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

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

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,

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

which completes the proof.  $\square$

#### ACKNOWLEDGEMENT

The authors thank the referees for their comments and suggestions, which helped the authors to improve their presentation.

#### REFERENCES

- [1] M. Al-Ammari and F. Tisseur, *Hermitian matrix polynomials with real eigenvalues of definite type. Part I: Classification*, Linear Algebra Appl. **436** (2012), no. 10, 3954–3973, DOI 10.1016/j.laa.2010.08.035. MR2914557
- [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 refinement of invariant pairs for matrix polynomials*, Linear Algebra Appl. **435** (2011), no. 3, 574–536, DOI 10.1016/j.laa.2010.06.029. MR2794589 (2012h:15012)

- [5] T. Betcke, N. J. Higham, V. Mehrmann, C. Schröder, and F. Tisseur, *NLEVP: a collection of nonlinear eigenvalue problems*, ACM Trans. Math. Software **39** (2013), no. 2, Art. 7, 28, DOI 10.1145/2427023.2427024. MR3031626
- [6] T. Betcke and H. Voss, *A Jacobi-Davidson type projection method for NEPs*, Future Generation Computer Systems, Vol. 20 (2004), pp. 363–372.
- [7] 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, October, 2011.
- [8] Y. Cai, Z. Bai, J. E. Pask, and N. Sukumar, *Hybrid preconditioning for iterative diagonalization of ill-conditioned generalized eigenvalue problems in electronic structure calculations*, J. Comput. Phys. **255** (2013), 16–30, DOI 10.1016/j.jcp.2013.07.020. MR3109776
- [9] 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. **13** (2006), no. 6, 487–512, DOI 10.1002/nla.484. MR2247445 (2007c:65031)
- [10] C. Conca, J. Planchard, and M. Vanninathan, *Fluids and Periodic Structures*, RAM: Research in Applied Mathematics, vol. 38, John Wiley & Sons, Ltd., Chichester; Masson, Paris, 1995. MR1652238 (99k:73094)
- [11] Y.H. Dai, *Nonlinear Conjugate Gradient Methods*, Wiley Encyclopedia of Operations Research and Management Science (2011), DOI:10.1002/9780470400531.eorms0183.
- [12] A. Edelman, T. A. Arias, and S. T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM J. Matrix Anal. Appl. **20** (1999), no. 2, 303–353, DOI 10.1137/S0895479895290954. MR1646856 (99h:65075)
- [13] C. Effenberger, *Robust successive computation of eigenpairs for nonlinear eigenvalue problems*, SIAM J. Matrix Anal. Appl. **34** (2013), no. 3, 1231–1256, DOI 10.1137/120885644. MR3092750
- [14] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*, 2nd ed., Numerical Mathematics and Scientific Computation, Oxford University Press, Oxford, 2014. MR3235759
- [15] G. Gambolati, F. Sartoretto, and P. Florian, *An orthogonal accelerated deflation technique for large symmetric eigenproblems*, Comput. Methods Appl. Mech. Engrg. **94** (1992), no. 1, 13–23, DOI 10.1016/0045-7825(92)90154-C. MR1148708 (92i:65072)
- [16] N. Gould, D. Orban, and P. Toint, *Numerical methods for large-scale nonlinear optimization*, Acta Numer. **14** (2005), 299–361, DOI 10.1017/S0962492904000248. MR2168345 (2006h:90068)
- [17] C.-H. Guo, N. J. Higham, and F. Tisseur, *Detecting and solving hyperbolic quadratic eigenvalue problems*, SIAM J. Matrix Anal. Appl. **30** (2008/09), no. 4, 1593–1613, DOI 10.1137/070704058. MR2486855 (2010b:15021)
- [18] K. P. Hadeler, *Variationsprinzipien bei nichtlinearen Eigenwertaufgaben* (German), Arch. Rational Mech. Anal. **30** (1968), 297–307. MR0234322 (38 #2639)
- [19] W. W. Hager and H. Zhang, *A survey of nonlinear conjugate gradient methods*, Pac. J. Optim. **2** (2006), no. 1, 35–58. MR2548208 (2010h:90095)
- [20] E. Jarlebring, *The Spectrum of Delay-Differential Equations: Numerical Methods, Stability and Perturbation*, PhD thesis, Inst. Comp. Math, TU Braunschweig, 2008.
- [21] A. V. Knyazev, *A preconditioned conjugate gradient method for eigenvalue problems and its implementation in a subspace*, Numerical treatment of eigenvalue problems, Vol. 5 (Oberwolfach, 1990), Internat. Ser. Numer. Math., vol. 96, Birkhäuser, Basel, 1991, pp. 143–154, DOI 10.1007/978-3-0348-6332-2\_11. MR1109101
- [22] A. V. Knyazev, *Preconditioned eigensolvers—an oxymoron?*, Electron. Trans. Numer. Anal. **7** (1998), 104–123 (electronic). Large scale eigenvalue problems (Argonne, IL, 1997). MR1667642 (99h:65068)
- [23] A. V. Knyazev, *Toward the optimal preconditioned eigensolver: locally optimal block preconditioned conjugate gradient method*, SIAM J. Sci. Comput. **23** (2001), no. 2, 517–541 (electronic), DOI 10.1137/S1064827500366124. Copper Mountain Conference (2000). MR1861263 (2003g:65050)
- [24] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed., Springer Series in Operations Research and Financial Engineering, Springer, New York, 2006. MR2244940 (2007a:90001)
- [25] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003. MR1990645 (2004h:65002)

- [26] K. Schreiber, *Nonlinear Eigenvalue Problems: Newton-type Methods and Nonlinear Rayleigh Functionals*, Ph.D thesis, Department of Mathematics, TU Berlin, 2008.
- [27] H. Schwetlick and K. Schreiber, *Nonlinear Rayleigh functionals*, Linear Algebra Appl. **436** (2012), no. 10, 3991–4016, DOI 10.1016/j.laa.2010.06.048. MR2914559
- [28] G. L. G. Sleijpen and H. A. Van der Vorst, *A Jacobi-Davidson iteration method for linear eigenvalue problems*, SIAM J. Matrix Anal. Appl. **17** (1996), no. 2, 401–425, DOI 10.1137/S0895479894270427. MR1384515 (96m:65042)
- [29] G. L. G. Sleijpen, A. G. L. Booten, D. R. Fokkema, and H. A. Van der Vorst, *Jacobi-Davidson type methods for generalized eigenproblems and polynomial eigenproblems*, BIT **36** (1996), no. 3, 595–633, DOI 10.1007/BF01731936. International Linear Algebra Year (Toulouse, 1995). MR1410100 (97k:65095)
- [30] S. I. Solov'ev, *Preconditioned iterative methods for a class of nonlinear eigenvalue problems*, Linear Algebra Appl. **415** (2006), no. 1, 210–229, DOI 10.1016/j.laa.2005.03.034. MR2214753 (2007c:65036)
- [31] D. B. Szyld and F. Xue, *Local convergence analysis of several inexact Newton-type algorithms for general nonlinear eigenvalue problems*, Numer. Math. **123** (2013), no. 2, 333–362, DOI 10.1007/s00211-012-0489-1. MR3010183
- [32] D. B. Szyld, E. Vecharynski, and F. Xue, *Preconditioned eigensolvers for large-scale nonlinear Hermitian eigenproblems with variational characterizations. II. Interior eigenvalues*, SIAM J. Sci. Comput. **37** (2015), no. 6, A2969–A2997, DOI 10.1137/15M1016096. MR3435092
- [33] H. Voss, *A minmax principle for nonlinear eigenproblems depending continuously on the eigenparameter*, Numer. Linear Algebra Appl. **16** (2009), no. 11–12, 899–913, DOI 10.1002/nla.670. MR2567141 (2010k:49101)
- [34] H. Voss, *Nonlinear Eigenvalue Problems*, Chapter 60 in L. Hogben (ed.), Handbook of Linear Algebra, CRC Press, Boca Raton 2014.
- [35] H. Voss and B. Werner, *A minimax principle for nonlinear eigenvalue problems with applications to nonoverdamped systems*, Math. Methods Appl. Sci. **4** (1982), no. 3, 415–424, DOI 10.1002/mma.1670040126. MR669135 (84a:49063)
- [36] 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.
- [37] H. Yang, *Conjugate gradient methods for the Rayleigh quotient minimization of generalized eigenvalue problems* (English, with English and German summaries), Computing **51** (1993), no. 1, 79–94, DOI 10.1007/BF02243830. MR1242660 (94h:65040)

DEPARTMENT OF MATHEMATICS, TEMPLE UNIVERSITY (038-16), 1805 N. BROAD STREET,  
 PHILADELPHIA, PENNSYLVANIA 19122-6094  
*E-mail address:* szyld@temple.edu

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF LOUISIANA AT LAFAYETTE, P.O. Box 41010,  
 LAFAYETTE, LOUISIANA 70504-1010  
*E-mail address:* fxue@louisiana.edu