A PROJECTED PRECONDITIONED CONJUGATE GRADIENT METHOD FOR THE LINEAR RESPONSE EIGENVALUE PROBLEM

XING LI

School of Mathematics, Shanghai University of Finance and Economics
777 Guoding Road, Yangpu District
Shanghai, 200433, People’s Republic of China

CHUNGEN SHEN

College of Science, University of Shanghai for Science and Technology
334 Jungong Road, Yangpu District
Shanghai 200093, China

LEI-HONG ZHANG*

School of Mathematics and
Shanghai Key Laboratory of Financial Information Technology
Shanghai University of Finance and Economics
777 Guoding Road, Yangpu District
Shanghai, 200433, People’s Republic of China

(Communicated by Wenyu Sun)

ABSTRACT. The linear response eigenvalue problem aims at computing a few smallest positive eigenvalues together with the associated eigenvectors of a special Hamiltonian matrix and plays an important role for estimating the excited states of physical systems. A subspace version of the Thouless minimization principle was established by Bai and Li (SIAM J. Matrix Anal. Appl., 33:1075-1100, 2012) which characterizes the desired eigenpairs as its solution. In this paper, we propose a Projected Preconditioned Conjugate Gradient (PPCG) method to solve this subspace version of Thouless’s minimization directly. We show that PPCG is an efficient implementation of the inverse power iteration and can be performed in parallel. It also enjoys several properties including the monotonicity and constraint preservation in the Thouless minimization principle. Convergence of both eigenvalues and eigenvectors are established and numerical experiences on various problems are reported.

1. Introduction. In computational quantum chemistry and physics, the so-called random phase approximation (RPA) describes the excited states of physical systems in many-particle systems [1, 3, 10, 11, 15, 16, 17, 18], which has applications in silicon nanoparticles and nanoscale materials and analysis of interstellar clouds [1, 2]. One

2010 Mathematics Subject Classification. Primary: 65F15, 65F30; Secondary: 15A18.

Key words and phrases. linear response eigenvalue problem, conjugate gradient method, inverse power iteration, minimization principle.

The third author is supported by NSF grant NSFC-11671246, NSFC-91730303 and NSFC-11371102.

* Corresponding author: Lei-Hong Zhang.
important question in RPA is to compute a few, say \(k\), eigenpairs associated with the smallest positive eigenvalues of the following eigenvalue problem:

\[
H w = \begin{bmatrix} A & B \\ -B & -A \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \lambda \begin{bmatrix} u \\ v \end{bmatrix} \tag{1.1}
\]

where \(A, B \in \mathbb{R}^{n \times n}\) are both symmetric matrices and

\[
\begin{bmatrix} A & B \\ B & A \end{bmatrix}
\]

is positive definite. \(\tag{1.2}\)

The eigenvalue problem (1.1) results from computing the excitation energies and absorption spectrum of a time-dependent Kohn-Sham (KS) system in the density functional theory \([3, 11, 12]\). The well-known time-independent (one-particle) KS equation can give properties of the corresponding many-particle system in the ground state, as long as an accurate exchange-correlation potential approximation is provided. When the underlying many-particle system is perturbed by a specific perturbation, the excitation energies and absorption spectrum are usually of interest. In this situation, the associated KS equation is time-dependent, and in the absence of external perturbations, its linearized system in the frequency domain can be expressed as an eigenvalue problem \([11, \text{eq. (14)}]\) associated with the Liouvillian super operator. The information in the eigensystem of the Liouvillian super operator provides the excited states of the underlying system where the eigenvalues represent the excitation energies and the eigenpairs can be used to estimate the absorption spectrum.

The matrix \(H\) in (1.1) is the finite dimensional \textit{batch representation} \([11]\) of the action of the Liouvillian super operator onto one-particle time-dependent KS density matrix in the frequency domain \([11]\). Note that \(H\) is a special Hamiltonian matrix and its eigenvalues are all real and appear in pairs \(\pm \lambda\) with \(\lambda \geq 0\). In quantum chemistry, the first a few eigenpairs \(\begin{bmatrix} \lambda_j \\ u_j \\ v_j \end{bmatrix}\), for \(j = 1, 2, \ldots, k\), corresponding to the smallest \(k\) positive eigenvalues

\[
0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_k
\]

of \(H\) are usually desired, where \(\lambda_j \geq 0\) (resp. \(-\lambda_j \leq 0\)) is called the \(j\)th excitation energy (resp. the deexcitation energy). The first \(k\) eigenpairs \(\begin{bmatrix} \lambda_j \\ u_j \\ v_j \end{bmatrix}\), for \(j = 1, 2, \ldots, k\), are used to compute the absorption spectrum \([3, \text{eq. (13)}]\) associated with the perturbed time-dependent KS system, and we refer to \([3, 11, 12]\) for more detailed discussion on the Time-Dependent Density Functional Theory (TDDFT) for the excited-state calculations.

The eigenvalue problem (1.1) is referred to as the \textit{Linear Response Eigenvalue Problem} in the literature, and several minimization principles and algorithms \([2, 3, 10, 11, 13, 14]\) have been proposed to obtain the first few eigenpairs of (1.1). Among them, Thouless’s minimization principle \([15]\) describes the smallest positive eigenvalue \(\lambda_1\) of \(H\) as the minimum of the following optimization problem:

\[
\lambda_1 = \min_{\|u\|_2 \neq \|v\|_2} \rho(u, v), \quad \text{where} \quad \rho(u, v) = \frac{u^T \begin{bmatrix} A & B \\ B & A \end{bmatrix} u}{\|u\|_2 - \|v\|_2}. \tag{1.3}
\]
Note that we can introduce a matrix
\[ J = \frac{1}{\sqrt{2}} \begin{bmatrix} I_n & I_n \\ I_n & -I_n \end{bmatrix}, \]
satisfying \( J^T J = J^2 = I_{2n} \) to transform similarly \([1, 2] \) \( \mathcal{H} \) to
\[ J^T \mathcal{H} J = \begin{bmatrix} A + B & A - B \\ A - B & M - K \end{bmatrix} =: \begin{bmatrix} K \\ M \end{bmatrix} =: H; \] (1.4)
thus, by the relation
\[ z := \begin{bmatrix} y \\ x \end{bmatrix} = J^T \begin{bmatrix} u \\ v \end{bmatrix}, \quad \begin{bmatrix} u \\ v \end{bmatrix} = J \begin{bmatrix} y \\ x \end{bmatrix}, \]
we can rewrite (1.1) equivalently as
\[ H z := \begin{bmatrix} K \\ M \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} = \lambda \begin{bmatrix} y \\ x \end{bmatrix}. \] (1.5)
Consequently, Thouless’s minimization principle (1.3) can be equivalently expressed as
\[ \lambda_1 = \min_{x \neq 0, y \neq 0} \rho(x, y), \quad \text{where} \quad \rho(x, y) = \frac{x^T K x + y^T M y}{2|x|^2|y|}, \] (1.6)
or equivalently,
\[ \lambda_1 = \frac{1}{2} \min_{x \neq y = 1} x^T K x + y^T M y. \]

A further step towards Thouless’s minimization principal (1.6) is made in [1] where a subspace version is developed, namely,
\[ \sum_{i=1}^{k} \lambda_i = \inf_{U^T V = I_k} \left\{ \lambda(U, V) := \frac{1}{2} \text{tr}(U^T K U + V^T M V) \right\}, \] (1.7)
where \( U, V \in \mathbb{R}^{n \times k} \).

It is true that the condition (1.2) implies that both \( K \) and \( M \) in (1.4) are symmetric and positive definite [1], and computing the first \( k \) eigenpairs of (1.5) associated with \( \lambda_j \) for \( j = 1, 2, \ldots, k \) is also referred to the Linear Response Eigenvalue Problem (LREP) in this paper.

Apart from some direct methods [5, 8] for obtaining the full eigenvalue decomposition of the matrix \( \mathcal{H} \) in (1.1), several projection methods [2, 3, 10, 13, 14] have been proposed based on this extension of Thouless’s minimization principle (1.7) for finding the desired eigenpairs. The noticeable work of Bai and Li in [2] proposes the LO4DCG method which is an efficient realization of the locally optimal block preconditioned CG by making full use of the special structure of (1.5). LO4DCG is an improvement over the block 4D steep descent method introduced in [10]. [3, 13] discuss single-vector Lanczos type methods and [14] considers a block Chebyshev-Davidson iteration to compute the desired eigenpairs of (1.5). All these algorithms follow similarly a Rayleigh-Ritz projection technique used in the traditional eigenvalue computations. In particular, different algorithm introduces specific subspace expansion procedure but projects the original LREP onto the resulted subspace to form a smaller size LREP (1.5) for the Ritz pairs.

\[ ^{1}\text{When } K \text{ are both positive definite, the “inf” in (1.7) can be replaced by “min”}. \]
In this paper, we suggest a Projected Preconditioned Conjugate Gradient (PPCG) method to solve the subspace version of Thouless’s minimization principal (1.7) directly. The proposed PPCG is indeed an efficient implementation of an alternative variables iteration for the minimization principle (1.7). PPCG iteratively computes a sequence of pairs \((U_j, V_j)\) for \(j = 1, 2, \ldots\), and preserves the bi-orthogonality \(U_j^T V_j = I_k\) and ensures the monotonicity \(\chi(U_j, V_j) \leq \chi(U_{j+1}, V_{j+1})\).

For the iteration \(j\), the \(k\) columns in either \(U_j\) or \(V_j\) can be computed by a projected preconditioned CG as a whole in a block manner, or computed separately in a parallel scheme. Moreover, if appropriate preconditioners of \(K\) and/or \(M\) are available, they can be incorporated to speed up PPCG. The linear convergence of \((U_j, V_j)\) to the minimizer of the Thouless’s minimization principal (1.7) is proved under a gap assumption \(\lambda_k < \lambda_{k+1}\), and the \(k\) approximate eigenpairs can finally be obtained via solving an LREP of size \(k\). Numerical experiences of PPCG, coded both in MATLAB and in C, on several problems with various choices of preconditioners are reported.

We organize the paper as follows. Relevant preliminary results are stated in Section 2. In Section 3, we first give the framework of an alternative variable iteration, and then describe its implementation details, i.e., the PPCG method. The convergence analysis of PPCG is made in Section 4. Our numerical evaluation of PPCG for LREP on both randomly generated problems as well as two practical problems from computational quantum chemistry is conducted in Section 5, and conclusions are drawn in Section 6.

2. Preliminary results. To facilitate our discussions, we first collect several necessary properties of the LREP in Theorem 2.1, and the reader can refer to [1, Section 2] and [15, 16, 17, 18] for more propositions.

2.1. Basic results about LREP.

**Theorem 2.1.** Suppose that \(M\) is definite. Then the following statements are true:

1. There exists a nonsingular \(Y \in \mathbb{R}^{n \times n}\) such that
   \[
   K = YA^2Y^T, \quad M = XX^T,
   \]
   where \(A = \text{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_n\}\) and \(X = Y^{-T}\).

2. The eigen-decomposition of \(KM\) and \(MK\) are
   \[
   (KM)Y = YA^2, \quad (MK)X = XA^2,
   \]
   respectively.

3. If \(K\) is also definite, then all \(\lambda_i > 0\) and \(H\) is diagonalizable:
   \[
   H \begin{bmatrix} YA & YA \\ X & -X \end{bmatrix} = \begin{bmatrix} YA & YA \\ X & -X \end{bmatrix} \begin{bmatrix} \Lambda & 0 \\ 0 & -\Lambda \end{bmatrix},
   \]

4. \(H\) is not diagonalizable if and only if \(\lambda_1 = 0\), which happens when and only when \(K\) is singular.

5. The \(i\)th column of \(Z = \begin{bmatrix} YA \\ X \end{bmatrix}\) is the eigenvector corresponding to \(\lambda_i\), and it is unique if (a) \(\lambda_i\) is a simple eigenvalue of \(H\), or (b) \(i = 1, \lambda_1 = +0 < \lambda_2\). In this case, 0 is a double eigenvalue of \(H\) but there is only one eigenvector associated with it.
The property (2.2) follows directly from (2.1), which implies that we can alternatively solve the LREP by solving any product eigenvalue problem in (2.2). It is worth noting that only the $k$ smallest positive eigenvalues together with the associated eigenvectors are of interests, and we are concerned with the extreme eigenpair of the product eigenvalue problem in (2.2).

By the Lagrangian multiplier theory, for a KKT pair $(U, V)$ of the Thouless type minimization principle (1.7), we know that there are two matrices $\Xi \in \mathbb{R}^{k\times k}$ and $\mathcal{Y} \in \mathbb{R}^{k\times k}$ such that

$$KU = V \Xi \quad \text{and} \quad MV = U \mathcal{Y}.$$  

The deflating subspace $\{U, V\}$ of $\{K, M\}$ in [1] is essentially the one spanned by the KKT pair $(U, V)$ which satisfies $KU \subseteq V$ and $MV \subseteq U$. For any deflating subspace $\{U, V\}$ with

$$\dim(U) = \dim(V) = k \quad \text{and} \quad U \oplus V^\perp = \mathbb{R}^n,$$  

there is a KKT pair $(U, V)$ so that $\mathcal{R}(U) = U$ and $\mathcal{R}(V) = V$, and vice versa. The solution of the Thouless type minimization principle (1.7) just corresponds to the extreme positive deflating subspace so that the objective function value $\chi(U, V)$ achieves the minimum.

Let $U \in \mathbb{R}^{n \times k}$ and $V \in \mathbb{R}^{n \times k}$ be the basis matrices of the subspaces $U$ and $V$ satisfying (2.3), then $W = U^T V$ is nonsingular. Factorize $W$ as $W = W_1^T W_2$ with two nonsingular $W_1$ and $W_2$ to have a structure-preserving projection matrix $H_{SR}$

$$H \begin{bmatrix} V W_1^{-1} & U W_1^{-1} \end{bmatrix} = \begin{bmatrix} V W_2^{-1} & U W_1^{-1} \end{bmatrix} H_{SR},$$  

where

$$H_{SR} = \begin{bmatrix} W_2^{-T} V^T M V W_2^{-1} & W_2^{-T} U^T K U W_1^{-1} \end{bmatrix}.$$  

Consequently, any eigenpair $\left(\lambda, \begin{bmatrix} \hat{y} \\ \hat{x} \end{bmatrix}\right)$ of $H_{SR}$ yields an eigenvalue $\lambda$ of $H$ and the corresponding eigenvector of (1.5) with $x = U W_1^{-1} \hat{x}$ and $y = V W_2^{-1} \hat{y}$.

### 2.2. Canonical angles

In our discussion of the convergece, we will use the canonical angles and angles in $M$-inner product. For two subspaces $\mathcal{A}$ and $\mathcal{B}$ of $\mathbb{R}^n$ with $k = \dim(\mathcal{A}) \leq \dim(\mathcal{B}) = \ell$, the angles $\theta_i(\mathcal{A}, \mathcal{B})$ are defined recursively for $i = 1, 2, \ldots, k$, by [7]

$$\cos \theta_i(\mathcal{A}, \mathcal{B}) = \max_{x \in \mathcal{A}} \max_{y \in \mathcal{B}} x^T y = x_i^T y_i$$  

subject to

$$\|x\|_2 = \|y\|_2 = 1, \quad x_j^T x_j = y_j^T y_j = 0, \quad j = 1, 2, \ldots, i - 1.$$  

If $A \in \mathbb{R}^{n \times k}$ and $B \in \mathbb{R}^{n \times \ell}$ are orthonormal basis matrices of $\mathcal{A}$ and $\mathcal{B}$, respectively, and $\sigma_1 \leq \cdots \leq \sigma_k$ are the singular values of $B^T A$, then the $k$ canonical angles $\theta_j(\mathcal{A}, \mathcal{B})$ from $\mathcal{A}$ to $\mathcal{B}$ are

$$0 \leq \theta_j(\mathcal{A}, \mathcal{B}) := \arccos \sigma_j \leq \frac{\pi}{2} \quad \text{for} \quad 1 \leq j \leq k.$$  

Set

$$\Theta(\mathcal{A}, \mathcal{B}) = \text{diag}\{\theta_1(\mathcal{A}, \mathcal{B}), \ldots, \theta_k(\mathcal{A}, \mathcal{B})\}.$$  

Note that $\theta_1(\mathcal{A}, \mathcal{B}) \geq \cdots \geq \theta_k(\mathcal{A}, \mathcal{B})$ and the angles are independent of the orthonormal basis matrices $A$ and $B$. Therefore, no confusion can arise to use $\Theta(\mathcal{A}, \mathcal{B})$ instead of $\Theta(\mathcal{A}, \mathcal{B})$. 
The value \( \| \sin \Theta(A, B) \|_2 \) defines a distance metric between \( A \) and \( B \), and
\[
\| \sin \Theta(A, B) \|_2 = \sin \theta_1(A, B) = \| A^T B \|_2,
\]
where \( B_\perp \) is an orthonormal basis matrix of the orthogonal complement of \( B \).

Now for the symmetric and positive definite matrix \( M \), we denote the \( M \)-inner product by \( \langle x, y \rangle_M = x^T M y \). Generalizing the canonical angles given by (2.6) and (2.7) leads to the angles in \( M \)-inner product, which we will denote by \( \theta_i(A, B)_M \) and \( \Theta(A, B)_M = \text{diag}(\theta_1(A, B)_M, \ldots, \theta_k(A, B)_M) \). The canonical angles \( \theta_i(A, B) \) and the angles \( \theta_i(A, B)_M \) in \( M \)-inner product are related as follows [7, Theorem 4.2]: if \( M = XX^T \), then the angles between subspaces \( A \) and \( B \) relative to \( M \)-inner product coincide with the canonical angles between subspaces \( X^T A \) and \( X^T B \). Based on this connection, instead of the distance metric \( \| \sin \Theta(A, B) \|_2 \) between \( A \) and \( B \), we have the distance \( \| \sin \Theta(A, B)_M \|_2 \) and
\[
\| \sin \Theta(A, B)_M \|_2 = \| \sin \Theta(X^T A, X^T B) \|_2. \quad (2.9)
\]

3. The Projected Preconditioned Conjugate Gradient Method. In this section, we first present the framework of an alternative variables iteration and then discuss an efficient Projected Preconditioned Conjugate Gradient (PPCG1rep) method for its implementation. We assume that \( K \) and \( M \) are symmetric positive definite from now on.

3.1. The framework of the alternative variables iteration. By relying on the minimization principle (1.7), a natural idea is to alternatively iterate \( U \in \mathbb{R}^{n \times k} \) and \( V \in \mathbb{R}^{n \times k} \) to improve the objective value \( \chi(U, V) \), each with the other variable fixed. In particular, starting from an initial pair \( [U_0, V_0] \in \mathbb{R}^{n \times 2k} \) satisfying \( U_0^T V_0 = I_k \), Algorithm 3.1 provides the basic iteration step.

**Algorithm 3.1** The framework of the alternative variables iteration for LREP

Given a pair \( [U_0, V_0] \in \mathbb{R}^{n \times 2k} \) with \( U_0^T V_0 = I_k \), the following iteration computes an approximate maximizer for the optimization problem (1.7).

1: \( j \leftarrow 0; \)
2: \textbf{while} not convergence \textbf{do}
3: \( U_{j+1} : = \arg \min_{U^T V_j = I_k} \frac{1}{2} \text{tr}(U^T K U) = \arg \min_{U^T V_j = I_k} \chi(U, V_j); \quad (3.1a) \)
   \( V_{j+1} : = \arg \min_{U^T V_j = I_k} \frac{1}{2} \text{tr}(V^T M V) = \arg \min_{U^T V_j = I_k} \chi(U_{j+1}, V); \quad (3.1b) \)
4: \( j \leftarrow j + 1; \)
5: \textbf{end while}

The initial pair \( [U_0, V_0] \in \mathbb{R}^{n \times 2k} \) can simply be \( U_0 = V_0 = [e_1, \ldots, e_k] \), or more generally, \( U_0 = U(U^T V)^{-1}, \; V_0 = V \) for any full column rank \( U \) and \( V \) with \( \text{rank}(U^T V) = k \). Thus it is seen that the main computational burden in each outer iteration of Algorithm 3.1 is solving the two subproblems (3.1a) and (3.1b). Taking (3.1a) for example, the Lagrangian multiplier theory says that there exists a matrix \( \Xi \in \mathbb{R}^{k \times k} \) satisfying
\[
KU - V_j \Xi = 0 \quad \text{and} \quad U^T V_j = I_k, \quad (3.2)
\]
which leads to the solution of (3.1a)

\[ U_{j+1} = K^{-1}V_j(V_j^TK^{-1}V_j)^{-1}. \] (3.3)

By a similar argument, we can easily see that the solution of (3.1b) is

\[ V_{j+1} = M^{-1}U_j(U_j^TM^{-1}U_j)^{-1}. \] (3.4)

Now, by substituting (3.3) into (3.4), we know that the iteration formulation from \( V_j \) to the next \( V_{j+1} \) is given by

\[ V_{j+1} = (KM)^{-1}V_j[V_j^T(KMK)^{-1}V_j]^{-1}(V_j^TK^{-1}V_j). \] (3.5)

One can easily realize that an exchange of the subproblems (3.1a) and (3.1b) in Algorithm 3.1 results in the iteration formula for the sequence \( \{U_j\} \) with

\[ U_{j+1} = (MK)^{-1}U_j[U_j^T(MKM)^{-1}U_j]^{-1}(U_j^TM^{-1}U_j). \] (3.6)

An interesting observation from (3.5) or (3.6) implies that our alternative variables iteration is essentially a special type of inverse power iteration for solving the product eigenvalue problems (2.2). However, we remark that the explicit iteration formula (3.5) is of little use from the computational point of view, as it would be too expensive to compute the matrix-matrix products and inverses; moreover, we will not also rely upon (3.5) to get an approximation solution (using, for example, the Krylov subspace iteration) to obtain an inexact solution \( V_{j+1} \) because the decreasing of the objective function \( \chi(U,V) \) is not guaranteed. Instead, in the next subsection, we will propose an implementation for obtaining the solution or an inexact solution which guarantees that

1. the objective function \( \chi(U,V) \) is non-increasing; that is,
   \[ \chi(U_j,V_j) \geq \chi(U_{j+1},V_{j+1}); \]

2. the constraint is preserved, i.e.,
   \[ U_j^TV_j = U_{j+1}^TV_{j+1} = I_k; \] (3.7)

3. the columns of \( U_{j+1} \) and \( V_{j+1} \) in (3.1a) and (3.1b), respectively, can be computed in parallel;
4. each column of \( U_{j+1} \) and \( V_{j+1} \) can be computed by an efficient preconditioned CG iteration and therefore, only matrix-vector products are involved.

We conclude this subsection by suggesting a stopping criterion for Algorithm 3.1 in line 2. Let \( (U_j,V_j) \) be the current iteration pair, which is an approximation (i.e., an approximate KKT pair) for the Thouless’s minimization principal (1.7). Therefore, according to (2.3), \( (U_j,V_j) \) is an approximate deflating subspace of \( \{K,M\} \), which by \( U_j^TV_j = I_k \), (2.4) and (2.5) imply that the \( k \) positive eigenvalues (the Ritz values) of

\[ H_j = \begin{bmatrix} V_j^TMV_j & U_j^TKU_j \end{bmatrix}, \] (3.8)

serve as the approximations for \( \lambda_1, \ldots, \lambda_k \). Based upon this observation, we can terminate the iteration if the relative residual error

\[ \text{Res} = \frac{\|KU_j - V_j(U_j^TKU_j)\|_1}{\|KU_j\|_1 + \|V_j(U_j^TKU_j)\|_1} + \frac{\|MV_j - U_j(V_j^TMV_j)\|_1}{\|MV_j\|_1 + \|U_j(V_j^TMV_j)\|_1} \leq \epsilon_r \] (3.9)
Therefore, we only need to consider the solutions of
$$W^{16.3}.$$ To be precise, if each of which can be computed by the (preconditioned) CG iteration [9, Section 3.2]. The Ritz pairs corresponding to the $k$ positive Ritz values of $H_j$ in (3.8) at the terminated iteration $j$ are used as the approximations for the LREP (1.5).

3.2. Solve the subproblems. We now discuss how to solve the subproblems (3.1a) and (3.1b) in an efficient way. The implementation results our Projected Preconditioned Conjugate Gradient (PPCG$\_\$lrep) method.

Take the former as an example. Using the condition $U_j^TV_j = I_k$, for the moment, we first express the constraint $U^TV_j = I_k$ as
$$U = P_vZ + U_j,$$ where $P_v \in \mathbb{R}^{n \times (n-k)}$ is a basis of Null($V_j^T$). It should be pointed out that, in our computation, $P_v$ needs not to be formed explicitly. Substituting (3.11) into (3.1a) yields
$$\chi_1(U) := \frac{1}{2} \text{tr}(U^T K U) = \frac{1}{2} \text{tr}(Z^T P_v^T K P_v Z) + \text{tr}(Z^T P_v^T K U_j) + \frac{1}{2} \text{tr}(U_j^T K U_j).$$

Now, denote $Z = [z_1, \ldots, z_k]$ and $P_v^T K U_j = [w_1, \ldots, w_k]$ and we know that (3.1a) is equivalent to
$$\min_{z_i \in \mathbb{R}^{n-k}} \sum_{i=1}^k \left( \frac{Z^T P_v^T K P_v z_i}{2} + z_i^T w_i \right) = \min_{z_i \in \mathbb{R}^{n-k}} \left\{ \varphi(z_i) := \frac{Z_i^T P_v^T K P_v z_i}{2} + z_i^T w_i \right\}.$$ Note that
$$\min_{U^TV_j = I_k} \chi(U, V_j) = \sum_{i=1}^k \min_{z_i \in \mathbb{R}^{n-k}} \varphi(z_i) + \chi(U_j, V_j).$$
Therefore, we only need to consider the solutions of
$$\min_{z_i \in \mathbb{R}^{n-k}} \left\{ \frac{Z_i^T P_v^T K P_v z_i}{2} + z_i^T w_i \right\}, \quad i = 1, \ldots, k,$$ each of which can be computed by the (preconditioned) CG iteration [9, Section 16.3]. To be precise, if $W = P_v^T K P_v$ is a precondition so that the symmetric and positive definite matrix $\tilde{K} = K$ and $W^{-\frac{1}{2}}(P_v^T K P_v)W^{-\frac{1}{2}} \approx I_{n-k}$, then the preconditioned CG iteration [9, Algorithm 16.1] for (3.13) processes as Algorithm 3.2.

There are many nice properties for the (preconditioning) CG iteration (e.g., [9, Chapter 5]), and among them, the following proposition [9, Theorem 5.2] implies the monotonic decrease of the objective value.

Proposition 1. Let $\{z_i^{(t)}\}$ be the sequence generated by Algorithm 3.2, then $z_i^{(t+1)}$ is the minimizer of $\varphi(z_i)$ over $z_i^{(t)} + \mathcal{K}_t(W^{-1}P_v^T K P_v, W^{-1}r_i^{(0)})$, where
$$\mathcal{K}_t(W^{-1}P_v^T K P_v, W^{-1}r_i^{(0)})$$ is the Krylov subspace. Therefore, $\varphi(z_i^{(t)}) \geq \varphi(z_i^{(t+1)})$. 

Algorithm 3.2 Preconditioned CG for the reduced systems w.r.t. $z_i$

Staring from an initial point $z^{(0)}_i \in \mathbb{R}^{n-k}$ and a precondition matrix $W = P_v^T \bar{K} P_v$, this preconditioned CG iteration solves inexactly (3.13).

\begin{algorithm}
\begin{algorithmic}[1]
\State $r^{(0)}_i = P_v^T \bar{K} P_v z^{(0)}_i + w_i$, $\gamma^{(0)}_i = W^{-1} r^{(0)}_i$, $\hat{d}^{(0)}_i = -\gamma^{(0)}_i$, $\ell = 0$;
\While{$||r^{(\ell)}_i||_2 > \epsilon_c$} \Do
\State $z^{(\ell+1)}_i = z^{(\ell)}_i + \hat{\alpha} \hat{d}^{(\ell)}_i$, with $\hat{\alpha} = \frac{(r^{(\ell)}_i)^T \hat{g}^{(\ell)}_i}{(d^{(\ell)}_i)^T P_v^T \bar{K} P_v d^{(\ell)}_i}$;
\State $\tilde{r}^{(\ell+1)}_i = \tilde{r}^{(\ell)}_i + \hat{\alpha} P_v^T \bar{K} P_v \hat{d}^{(\ell)}_i$;
\State $\tilde{d}^{(\ell+1)}_i = W^{-1} \tilde{r}^{(\ell+1)}_i$;
\State $\tilde{d}^{(\ell+1)}_i = -\tilde{g}^{(\ell)}_i + \hat{\beta} \tilde{d}^{(\ell)}_i$, with $\hat{\beta} = \frac{(r^{(\ell+1)}_i)^T \tilde{g}^{(\ell+1)}_i}{(\tilde{r}^{(\ell+1)}_i)^T \tilde{g}^{(\ell+1)}_i}$;
\State $\ell = \ell + 1$;
\EndWhile
\end{algorithmic}
\end{algorithm}

Indeed, Algorithm 3.2 can be taken one step further to yield a projected CG iteration (see e.g., [9, Algorithm 16.2]), which works directly on systems with respect to $u_i = P_v z_i + u^{(0)}_i$, where $U = [u^{(0)}_1, \ldots, u^{(0)}_k]$ and $U_j = [u^{(0)}_1, \ldots, u^{(0)}_k]$. This can be realized by introducing new variables $r_i, g_i$ and $d_i$ by

$$ P_v^T r_i = \tilde{r}_i, \quad g_i = P_v \tilde{g}_i, \quad d_i = P_v \tilde{d}_i, $$

respectively. With these new variables, we can work on the variable $u_i$ with the corresponding preconditioner $F_K = P_v W^{-1} P_v^T$ and with a special initial point $u^{(0)}_i$ (i.e., corresponding to $z^{(0)}_i = 0$ in Algorithm 3.2), and the iteration is summarized in Algorithm 3.3 in which the stopping criterion is $(r^{(\ell)}_i)^T g^{(\ell)}_i = (r^{(\ell)}_i)^T P_v r^{(\ell)}_i = ||r^{(\ell)}_i||_2^2$.

Algorithm 3.3 Projected preconditioned CG for the systems w.r.t. $u_i$

Given a precondition matrix $F_K = P_v (P_v^T \bar{K} P_v)^{-1} P_v^T$, this preconditioned CG iteration obtains a vector which is an approximation to the $i$th column of the solution $U$ of (3.1a).

\begin{algorithm}
\begin{algorithmic}[1]
\State $r^{(0)}_i = K u^{(0)}_i$, $g^{(0)}_i = F_K r^{(0)}_i$, $d^{(0)}_i = -g^{(0)}_i$, $\ell = 0$;
\While{$(r^{(\ell)}_i)^T g^{(\ell)}_i > \epsilon_c$} \Do
\State $u^{(\ell+1)}_i = r^{(\ell)}_i + \alpha_i d^{(\ell)}_i$, with $\alpha_i = \frac{(r^{(\ell)}_i)^T g^{(\ell)}_i}{(d^{(\ell)}_i)^T K d^{(\ell)}_i}$;
\State $r^{(\ell+1)}_i = r^{(\ell)}_i + \alpha_i K d^{(\ell)}_i$;
\State $g^{(\ell+1)}_i = F_K r^{(\ell+1)}_i$;
\State $d^{(\ell+1)}_i = -g^{(\ell+1)}_i + \beta_i d^{(\ell)}_i$, with $\beta_i = \frac{(r^{(\ell+1)}_i)^T g^{(\ell+1)}_i}{(r^{(\ell+1)}_i)^T d^{(\ell+1)}_i}$;
\State $\ell = \ell + 1$;
\EndWhile
\end{algorithmic}
\end{algorithm}

Remark 1. It can be verified that if Algorithm 3.3 generates the sequences $\{u^{(\ell)}_i\}$, $\{r^{(\ell)}_i\}$, $\{g^{(\ell)}_i\}$ and $\{d^{(\ell)}_i\}$, they uniquely determine the corresponding sequences $\{z^{(\ell)}_i\}$, $\{\tilde{r}^{(\ell)}_i\}$, $\{\tilde{g}^{(\ell)}_i\}$ and $\{\tilde{d}^{(\ell)}_i\}$, which are also the sequences generated by Algorithm 3.2 staring from the initial point $z^{(0)}_i = 0$, and it is true that

$$ u^{(\ell)}_i = P_v z^{(\ell)}_i + u^{(0)}_i. $$

(3.14)
Moreover, with the initial point \( \mathbf{u}^{(0)}_i \) and by the relation (3.14), we know that, if \( U_{j+1} = [\hat{\mathbf{u}}_1, \ldots, \hat{\mathbf{u}}_k] \) where \( \hat{\mathbf{u}}_i = P_v \mathbf{z}_i + \mathbf{u}^{(0)}_i \) is an inexact solution obtained from Algorithm 3.3, it is true that

\[
U_{j+1} = P_v [\hat{\mathbf{z}}_1, \ldots, \hat{\mathbf{z}}_k] + U_j \implies U_{j+1}^T V_j = I_k,
\]

which means that the constraint \( U^T V_j = I_k \) is preserved. In addition, by Proposition 1 and (3.14), we know that

\[
\chi_1(U_{j+1}) = \sum_{i=1}^{k} \varphi(\hat{\mathbf{z}}_i) + \chi_1(U_j) \leq \sum_{i=1}^{k} \varphi(0) + \chi_1(U_j) = \chi_1(U_j).
\]

As a result, we have

\[
\chi(U_{j+1}, V_j) \leq \chi(U_j, V_j). \tag{3.16}
\]

One can easily see that the vectors \( \mathbf{u}^{(t)}_i \) for \( i = 1, \ldots, k \) generated by Algorithm 3.3 can also be obtained simultaneously by updating the corresponding matrices

\[
R^{(t)} := [r^{(t)}_1, \ldots, r^{(t)}_k], \quad G^{(t)} := [g^{(t)}_1, \ldots, g^{(t)}_k],
\]

\[
D^{(t)} := [d^{(t)}_1, \ldots, d^{(t)}_k], \quad U^{(t)} := [u^{(t)}_1, \ldots, u^{(t)}_k]
\]
as in Algorithm 3.4.

**Algorithm 3.4** Projected preconditioned CG for (3.1a)

Given a precondition matrix \( F_K = P_v (P_v^T K P_v)^{-1} P_v^T \), this preconditioned CG iteration obtains an (inexact) solution \( U_{j+1} = U^{(t)} \) of (3.1a).

1. \( R^{(0)} = K U_j, \ G^{(0)} = F_K R^{(0)}, \ D^{(0)} = -G^{(0)}, \ \ell = 0; \)
2. \( \text{while } \max\{r^{(t)}_1^T g^{(t)}_1, \ldots, r^{(t)}_k^T g^{(t)}_k\} < \epsilon_c \) do
3. \( U^{(t+1)} = U^{(t)} + D^{(t)} \text{ diag}\{\alpha_1, \ldots, \alpha_k\}, \) with \( \alpha_i = \frac{(r^{(t)}_i)^T g^{(t)}_i}{(d^{(t)})^T K d^{(t)}}; \)
4. \( R^{(t+1)} = R^{(t)} + K D^{(t)} \text{ diag}\{\alpha_1, \ldots, \alpha_k\}; \)
5. \( G^{(t+1)} = F_K R^{(t+1)}; \)
6. \( D^{(t+1)} = -G^{(t)} + D^{(t)} \text{ diag}\{\beta_1, \ldots, \beta_k\}, \) with \( \beta_i = \frac{(r^{(t+1)}_i)^T g^{(t+1)}_i}{(r^{(t)}_i)^T g^{(t)}_i}; \)
7. \( \ell = \ell + 1; \)
8. end while

The other subproblem (3.1b) can be solved similarly. In fact, if we denote by \( P^u \) a basis of \( \text{Null}(U^T_{j+1}) \) and by \( F_M = P^u (P^u T M P^u)^{-1} P^T_u \) a precondition matrix so that \( M = M \) is symmetric and positive definite, the subproblem (3.1b) can be solved by Algorithm 3.5.

**Theorem 3.1.** Suppose \( \{U_j\} \) and \( \{V_j\} \) are the sequences generated from Algorithm 3.1 where each \( U_j \) and \( V_j \) \( (j \geq 1) \), are computed ( inexactly) by Algorithm 3.4 and Algorithm 3.5, respectively, then for \( j = 0, 1, \ldots, \)

a. \( U_j^T V_j = U_{j+1}^T V_j = I_k \) and

b. \( \chi(U_{j+1}, V_j) \geq \chi(U_j, V_j) \)

**Proof.** These conclusions follow by induction. Since \( U_0^T V_0 = I_k \), by (3.15) and (3.16), we have \( U_1^T V_0 = I_k \) and \( \chi(U_1, V_0) \leq \chi(U_0, V_0) \). Thus, according to the similar argument, the next \( V_1 \) generated by Algorithm 3.5 satisfies \( U_1^T V_1 = I_k \) and \( \chi(U_1, V_1) \leq \chi(U_1, V_0) \).
Algorithm 3.5 Projected preconditioned CG for (3.1b)

Given a precondition matrix $F_M = P_u(P_u^T M P_u)^{-1} P_u^T$, this preconditioned CG iteration obtains an (inexact) solution $V_{j+1} = V^{(\ell)}$ of (3.1b).

1: $R^{(0)} = MV_j, G^{(0)} = F_M R^{(0)}, D^{(0)} = -G^{(0)}, \ell = 0$
2: while $\max\{ (r^{(\ell)}_i)^T g^{(\ell)}_i, \ldots, (r^{(\ell)}_k)^T g^{(\ell)}_k \} > \epsilon_r$ do
3: $V^{(\ell+1)} = V^{(\ell)} + D^{(\ell)} \text{diag}\{\alpha_1, \ldots, \alpha_k\}$, with $\alpha_i = \frac{(r^{(\ell)}_i)^T g^{(\ell)}_i}{(d^{(\ell)}_i)^T M d^{(\ell)}_i}$;
4: $R^{(\ell+1)} = R^{(\ell)} + M D^{(\ell)} \text{diag}\{\alpha_1, \ldots, \alpha_k\}$;
5: $G^{(\ell+1)} = F_M R^{(\ell+1)}$;
6: $D^{(\ell+1)} = -G^{(\ell)} + D^{(\ell)} \text{diag}\{\beta_1, \ldots, \beta_k\}$, with $\beta_i = \frac{(r^{(\ell+1)}_i)^T g^{(\ell+1)}_i}{(r^{(\ell)}_i)^T g^{(\ell)}_i}$;
7: $\ell = \ell + 1$;
8: end while

For a detailed implementation of Algorithms 3.4 and 3.5, we have the following additional remarks:

1. There are many simple choices of the precondition matrices for $F_K$ and $F_M$.
   For instance, we can take $[\hat{K}, \hat{M}] = [I_n, I_n], [\hat{K}, \hat{M}] = [\text{diag}(|K_{ii}|), \text{diag}(|M_{ii}|)]$ or the block diagonal forms.

2. It should be pointed out that in computing $f_K = F_K t$ and $f_M = F_M t$ for a given vector or a matrix $t$, we do not need to form explicitly the basis matrices $P_v$ and $P_u$. Instead, they can be computed as follows (see [9, Section 16.3]):

   
   $f_K = \hat{K}^{-1}[I_n - V_j (V_j^T \hat{K}^{-1} V_j)^{-1} V_j^T \hat{K}^{-1} t]$ \\
   \quad \quad \quad = \hat{K}^{-1} t - \hat{K}^{-1} V_j (V_j^T \hat{K}^{-1} V_j)^{-1} V_j^T \hat{K}^{-1} t$ \quad \quad \quad \quad \quad \quad \quad \quad (3.17)

   $f_M = \hat{M}^{-1}[I_n - U_{j+1} (U_{j+1}^T \hat{M}^{-1} U_{j+1})^{-1} U_{j+1}^T \hat{M}^{-1} t]$ \\\n   \quad \quad \quad = \hat{M}^{-1} t - \hat{M}^{-1} U_{j+1} (U_{j+1}^T \hat{M}^{-1} U_{j+1})^{-1} U_{j+1}^T \hat{M}^{-1} t$ \quad \quad \quad \quad \quad \quad \quad \quad (3.18)

3. The columns of $U_{j+1}$ and $V_{j+1}$ can be computed in parallel in which the precondition matrices $F_K$ and $F_M$ can be used for all $i = 1, 2, \ldots, k$.

4. Convergence analysis. Assuming the subproblems (3.1a) and (3.1b) are solved exactly, we investigate the convergence of the alternative variables iteration Algorithm 3.1 in this section.

4.1. Accuracy of the deflating subspace. Relying upon the explicit updating formulation (3.5), we first estimate how fast $\mathcal{R}(V_j)$ and $\mathcal{R}(U_j)$ approach $\mathcal{R}(Y_j)$ and $\mathcal{R}(X_1)$, respectively, where $\mathcal{R}(Y_1)$ and $\mathcal{R}(X_1)$ are the eigenspaces associated with the $k$ smallest eigenvalues of $KM$ and $MK$, respectively, and $Y = [Y_1, Y_2], X = [X_1, X_2]$ and $KMY_1 = Y_1 A^2_1, MKX_1 = X_1 A^2_1$.

Taking the advantage of the equal role of $K$ and $M$ in the Thouless’s minimization principal (1.7), we can focus on the accuracy of $\mathcal{R}(V_j)$ in approximating $\mathcal{R}(Y_1)$ first, and then apply the results to $\mathcal{R}(U_j)$. To this end, we will specially use the distance $\| \sin \Theta(V_j, Y_1)_M \|_2$. For the sake of convenience, in the following discussions, the subscripts are omitted so that $V_j$ and $V_{j+1}$ are denoted simply by $V$ and $\tilde{V}$ respectively.
The values $\mu$ and $\tilde{\mu}$ in (4.1) and (4.6), respectively, can be further simplified as the following lemma shows.

By (2.9) and (2.1), for $X = [X_1, X_2]$ with $X_1 \in \mathbb{R}^{n \times k}$ and $X_2 \in \mathbb{R}^{n \times (n-k)}$, we have

$$
\mu := \| \sin \Theta(V, Y_1) \|_2 = \| \sin \Theta(X^T V, X^T Y_1) \|_2 = \| \sin \Theta\left( X^T \begin{bmatrix} I_k & 0 \end{bmatrix} \right) \|_2 = \| (V^T M V)^{-\frac{1}{2}} V^T X \begin{bmatrix} 0 \\ I_{n-k} \end{bmatrix} \|_2 = \| (V^T M V)^{-\frac{1}{2}} V^T X_2 \|_2 = \| (S^T S)^{-\frac{1}{2}} S_2 \|_2,
$$

(4.1)

where we have defined

$$
S := X^T V = \begin{bmatrix} X_1^T V \\ X_2^T V \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix}.
$$

On the other hand, it follows that

$$
\tilde{\mu} := \| \sin \Theta(\tilde{V}, Y_1) \|_2 = \| \sin \Theta(X^T \tilde{V}, X^T Y_1) \|_2 = \| \sin \Theta\left( X^T \begin{bmatrix} I_k & 0 \end{bmatrix} \right) \|_2 = \| (\tilde{V}^T M \tilde{V})^{-\frac{1}{2}} \tilde{V}^T X \begin{bmatrix} 0 \\ I_{n-k} \end{bmatrix} \|_2 = \| (\tilde{V}^T M \tilde{V})^{-\frac{1}{2}} \tilde{V}^T X_2 \|_2 = \| (\tilde{S}^T \tilde{S})^{-\frac{1}{2}} \tilde{V}^T X_2 \|_2,
$$

(4.2)

with $\tilde{S} = X^T \tilde{V}$. By (3.5) and (2.1), we have

$$
\tilde{V} = (KM)^{-1} V \Omega = X^{-T} A^{-2} X^T V (V^T X \Lambda^{-4} X^T V)^{-1}(V^T X \Lambda^{-2} X^T V) = X^{-T} A^{-2} S (S^T A^{-4} S)^{-1}(S^T A^{-2} S),
$$

(4.3)

and thus,

$$
\tilde{V}^T X_2 = (S^T A^{-2} S)(S^T A^{-4} S)^{-1} S^T A^{-2} X^{-1} X_2 = (S^T A^{-2} S)(S^T A^{-4} S)^{-1} S^T \begin{bmatrix} 0 \\ A_2^2 \end{bmatrix} = (S^T A^{-2} S)(S^T A^{-4} S)^{-1} V^T X_2 A_2^2,
$$

(4.4)

and by (4.3)

$$
\tilde{S}^T \tilde{S} = \tilde{V}^T M \tilde{V} = \Omega^T S^T A^{-2} X^{-1} M X^{-T} A^{-2} S \Omega = \Omega^T S^T A^{-4} S \Omega = (S^T A^{-2} S)(S^T A^{-4} S)^{-1}(S^T A^{-2} S).
$$

(4.5)

Consequently, we have

$$
\tilde{\mu} = \| (S^T A^{-2} S)(S^T A^{-4} S)^{-1}(S^T A^{-2} S)^{-\frac{1}{2}} (S^T A^{-2} S)(S^T A^{-4} S)^{-1} V^T X_2 A_2^2 \|_2 = \| (S^T A^{-2} S)(S^T A^{-4} S)^{-1}(S^T A^{-2} S)^{-\frac{1}{2}} (S^T A^{-2} S)^{-1} S_2^T \|_2.
$$

(4.6)
Lemma 4.1. Let $\mu$ and $\tilde{\mu}$ be defined in (4.1) and (4.6), respectively. Then
\[
\mu = \sqrt{\lambda_{\text{max}}((S_2^T S_2)(S^T S)^{-1})},
\]
\[
\tilde{\mu} = \sqrt{\lambda_{\text{max}}((S_2^T A_2^{-4} S_2)(S^T A^{-4} S)^{-1})},
\]
where $\lambda_{\text{max}}(*)$ stands for the largest eigenvalue of $*$. 

Proof. (4.7) is obvious from (4.1). For (4.8), we have from (4.6) that
\[
\tilde{\mu}^2 = \|A_2^{-2} S_2 (S^T A^{-2} S)^{-1} [(S^T A^{-2} S)(S^T A^{-4} S)^{-1} (S^T A^{-2} S)]^{\frac{1}{2}} \|_2^2
\]
\[
= \lambda_{\text{max}}\left( (S^T A^{-2} S)(S^T A^{-4} S)^{-1} (S^T A^{-2} S)^{-1} \right) 
\times \left( (S_2^T A_2^{-4} S_2)(S^T A^{-2} S)^{-1} [(S^T A^{-2} S)(S^T A^{-4} S)^{-1} (S^T A^{-2} S)]^{\frac{1}{2}} \right)
\]
\[
= \lambda_{\text{max}}\left( (S^T A^{-2} S)^{-1} (S_2^T A_2^{-4} S_2)(S^T A^{-2} S)^{-1} \right) 
\times \left( [(S^T A^{-2} S)(S^T A^{-4} S)^{-1} (S^T A^{-2} S)]^{\frac{1}{2}} \right)
\]
\[
= \lambda_{\text{max}}\left( (S_2^T A_2^{-4} S_2)(S^T A^{-4} S)^{-1} \right),
\]
as asserted. \hfill \Box

Using Lemma 4.1, we are able to give a relation between $\tilde{\mu}$ and $\mu$ in Lemma 4.2.

Lemma 4.2. Under the assumptions of Lemma 4.1, we have
\[
\tilde{\mu} \leq \frac{\mu}{\sqrt{\mu^2 + \zeta^2 (1 - \mu^2)}},
\]
where $\zeta := \frac{\lambda_{k+1}}{\lambda_k} \geq 1$.

Proof. By (4.7) in Lemma 4.1 and $S^T S = S_1^T S_1 + S_2^T S_2$, we have
\[
\mu^2 = \lambda_{\text{max}}((S_2^T S_2)(S^T S)^{-1})
\]
\[
= \lambda_{\text{max}}\left( (S_2^T S_2) \frac{1}{2} (S_1^T S_1 + S_2^T S_2)^{-1} (S_2^T S_2)^{-\frac{1}{2}} \right)
\]
\[
= \lambda_{\text{max}}\left( [I_k + (S_2^T S_2)^{-\frac{1}{2}} S_1^T S_1 (S_2^T S_2)^{-\frac{1}{2}}]^{-1} \right),
\]
and therefore,
\[
\frac{1}{\mu^2} = \lambda_{\text{min}}\left( [I_k + (S_2^T S_2)^{-\frac{1}{2}} (S_1^T S_1) (S_2^T S_2)^{-\frac{1}{2}}] \right)
\]
\[
= 1 + \lambda_{\text{min}}\left( (S_2^T S_2)^{-\frac{1}{2}} (S_1^T S_1) (S_2^T S_2)^{-\frac{1}{2}} \right). 
\]

Similarly, by
\[
S^T A^{-4} S = S_1^T A_1^{-4} S_1 + S_2^T A_2^{-4} S_2,
\]
one has
\[
\frac{1}{\tilde{\mu}^2} = \lambda_{\text{min}}\left( [I_k + (S_2^T A_2^{-4} S_2)^{-\frac{1}{2}} (S_1^T A_1^{-4} S_1) (S_2^T A_2^{-4} S_2)^{-\frac{1}{2}}] \right)
\]
\[
= 1 + \lambda_{\text{min}}\left( (S_2^T A_2^{-4} S_2)^{-\frac{1}{2}} (S_1^T A_1^{-4} S_1) (S_2^T A_2^{-4} S_2)^{-\frac{1}{2}} \right). 
\]
On the other hand, by the ascending order of \( \lambda_i > 0 \), we have

\[
(S_2^T A_2^{-4} S_2) - \frac{1}{2} (S_2^T A_1^{-4} S_1) (S_2^T A_2^{-4} S_2)^{-\frac{1}{2} - \frac{1}{2}} \geq \left( \frac{\lambda_{k+1}}{\lambda_k} \right)^4 (S_2^T S_2)^{-\frac{1}{2} - \frac{1}{2} - \frac{1}{2}},
\]
where \( A \succeq B \) means that \( A - B \) is positive semi-definite. Therefore, by (4.11) and (4.10), we have

\[
\frac{1}{\mu^2} \geq 1 + \zeta^4 \lambda_{\min} \left( (S_2^T S_2)^{-\frac{1}{2}} (S_1^T S_1) (S_2^T S_2)^{-\frac{1}{2}} \right) = 1 + \zeta^4 \cdot \left( \frac{1}{\mu^2} - 1 \right),
\]
which consequently leads to (4.9), and the proof is completed.

We can present our convergence result for the alternative variables iteration Algorithm 3.1, which also reveals the numerical behavior of the PPGC \_rep method as well.

**Theorem 4.3.** Suppose \( \{U_j\} \) and \( \{V_j\} \) are the sequences generated by Algorithm 3.1 where \( U_j \) and \( V_j \) for \( j = 1, 2, \ldots \) are the solutions of the subproblems (3.1a) and (3.1b), respectively. Assume \( \zeta = \frac{\lambda_{k+1}}{\lambda_k} > 1 \) and \( \sigma \) is arbitrary satisfying \( \frac{1}{\sigma^2} \leq \sigma < 1 \). Then

(i) if \( V_0 \) satisfies

\[
\| \sin \Theta(V_0, Y_1)_M \|_2 \leq \sqrt{\frac{\zeta^4 \sigma - 1}{\zeta^4 \sigma - \sigma}},
\]
the sequence \( \{\mathcal{R}(V_j)\} \) converges to \( \{\mathcal{R}(Y_1)\} \) at least linearly, and moreover,

\[
\| \sin \Theta(V_j, Y_1)_M \|_2 \leq \sqrt{\sigma} \| \sin \Theta(V_j, Y_1)_M \|_2.
\]

(ii) if \( U_0 \) satisfies

\[
\| \sin \Theta(U_0, X_1)_K \|_2 \leq \sqrt{\frac{\zeta^4 \sigma - 1}{\zeta^4 \sigma - \sigma}},
\]
the sequence \( \{\mathcal{R}(U_j)\} \) converges to \( \{\mathcal{R}(X_1)\} \) at least linearly, and moreover,

\[
\| \sin \Theta(U_j, X_1)_K \|_2 \leq \sqrt{\sigma} \| \sin \Theta(U_j, X_1)_K \|_2.
\]

**Proof.** For (i), by (4.9) in Lemma 4.2, for \( 0 < \sigma < 1 \), the condition \( \tilde{\mu}^2 \leq \sigma \mu^2 \) yields

\[
1 \leq \mu^2 (1 - \zeta^4) + \sigma \zeta^4 \quad \text{or} \quad 1 - \sigma \zeta^4 \leq \mu^2 (1 - \zeta^4).
\]
Because \( \mu^2 (1 - \zeta^4) < 0 \), we require \( \frac{1}{\zeta^4} \leq \sigma < 1 \) and therefore, we require

\[
\mu \leq \sqrt{\frac{\zeta^4 \sigma - 1}{\zeta^4 \sigma - \sigma}} < 1.
\]
Now, by induction, if this condition (4.13) is fulfilled at \( j = 0 \), the sequence \( \{\mathcal{R}(V_j)\} \) converges to \( \mathcal{R}(Y_1) \) and (4.14) holds, too.

Part (ii) can be proved analogously by considering the symmetric structure in (1.7).

**Remark 2.** Theorem 4.3 implies that the larger the \( \zeta \) is, the smaller the \( \sigma \) could be, and therefore, the faster convergence is. Moreover, it can be seen from the initial condition (4.13) that as \( \zeta \) gets large (i.e., \( \sigma \) gets small), the attractive basin for the local convergence becomes large, too.
Remark 3. Note that for an arbitrary basis \((U, V)\) of \((\mathcal{R}(X_1), \mathcal{R}(Y_1))\) with \(U^TV = I_k\), it is true [1, Appendix A] that \(\chi(U, V) \geq \sum_{i=1}^k \lambda_i\) and the strictly inequality is possible. But as in [1, Appendix A], we can choose another basis \((\hat{U}, \hat{V})\) of \((\mathcal{R}(X_1), \mathcal{R}(Y_1))\)

\[
\hat{U} = U\psi_1^{-1}A_k^{-\frac{1}{2}}\quad \text{and} \quad \hat{V} = V\psi_1^{-1}A_k^{-\frac{1}{2}}
\]

to have \(\chi(\hat{U}, \hat{V}) = \sum_{i=1}^k \lambda_i\), where \(\psi_1, \psi_2, A_k \in \mathbb{R}^{k \times k}\) are from the decomposition (by Theorem 2.1) \(U^TKU = \psi_1A_1^2\psi_1^T\) and \(V^TMV = \psi_2\psi_2^T\) with \(\psi_2 = \psi_1^{-1}T\) and \(A_k = \text{diag}(\lambda_1, \ldots, \lambda_k)\). In other words, an arbitrary basis \((U, V)\) of the deflating subspace \((\mathcal{R}(X_1), \mathcal{R}(Y_1))\) with \(U^TV = I_k\) may not achieve the minimum \(\sum_{i=1}^k \lambda_i\) of Thouless’s minimization principal \((1.7)\), and the solution to \((1.7)\) is only a special basis \((\mathcal{R}(X_1), \mathcal{R}(Y_1))\). Now, let \((\hat{U}, \hat{V})\) be a limit point of the sequence \(\{(U_j, V_j)\}_{j=0}^\infty\), which by Theorem 4.3 implies that \((\mathcal{R}(\hat{U}), \mathcal{R}(\hat{V})) = (\mathcal{R}(X_1), \mathcal{R}(Y_1))\). It might also be true that \(\chi(\hat{U}, \hat{V}) > \sum_{i=1}^k \lambda_i\), but this has no effect on our final task in solving LREP \((1.5)\), as the projected matrix \(\hat{V}^TM\hat{V}\) contains the desired \(k\) eigenpairs.

4.2. Accuracy of the Ritz values. With the help of Theorem 4.3, we finally consider the local convergence of the Ritz values

\[
0 < \hat{\lambda}_1 \leq \cdots \leq \hat{\lambda}_k
\]

which are the \(k\) positive eigenvalues of the matrix \(H_j\) given in \((3.8)\).

First, it has been known that [1, Theorem 4.1] \(0 \leq \lambda_i \leq \hat{\lambda}_i\) for \(i = 1, 2, \ldots, k\), and hence \(\sum_{i=1}^k \lambda_i\) is always a lower bound for \(\sum_{i=1}^k \hat{\lambda}_i\). Thus, we are interested in an upper bound for \(\sum_{i=1}^k \hat{\lambda}_i\). The following Theorem 4.5 is a Rayleigh-Ritz type perturbation theory for LREP (see [18] for other types of Rayleigh-Ritz approximations for LREP), which is of interest in its own right and also provides the accuracy of \(\sum_{i=1}^k \hat{\lambda}_i\) in approximating the Thouless’s minimum \(\sum_{i=1}^k \lambda_i\) in \((1.7)\).

Lemma 4.4. ([6, Corollary 7.7.4]) If \(A, B \in \mathbb{R}^{n \times n}\) are symmetric and positive definite, then

(i) \(A \succeq B\) if and only if \(B^{-1} \succeq A^{-1}\);

(ii) If \(A \succeq B\), then \(\sigma_i(A) \geq \sigma_i(B)\) for all \(i = 1, 2, \ldots, n\), where \(\sigma_i(A)\) for \(i = 1, 2, \ldots, n\) are eigenvalues of \(A\) in the increasing order.

Theorem 4.5. Let \(K\) and \(M\) be symmetric and positive definite and \((U, V)\) be an approximation basis for \((\mathcal{R}(X_1), \mathcal{R}(Y_1))\) satisfying \(U^TV = I_k\) and

\[
\|\sin \Theta(V, Y_1)_M\|_2 \leq \epsilon < 1 \quad \text{and} \quad \|\sin \Theta(U, X_1)_K\|_2 \leq \epsilon < 1.
\]

Let \(0 < \hat{\lambda}_1 \leq \cdots \leq \hat{\lambda}_k\) be the \(k\) positive eigenvalues of

\[
H_{SR} = \begin{bmatrix} V^TMV & U^TKU \end{bmatrix} \in \mathbb{R}^{2k \times 2k}.
\]

Then for any sufficiently small \(\epsilon \in [0, 1)\), we have

\[
0 \leq \sum_{i=1}^k \hat{\lambda}_i - \sum_{i=1}^k \lambda_i \leq \epsilon^2 \sum_{i=1}^k \left( \lambda_i + \frac{\hat{\lambda}_i^2}{\lambda_i \hat{\lambda}_{k+1}} \right) + O(\epsilon^4).
\]
Proof. Note from $M = XX^T$, $K = (YA)(YA)^T$, $Y^T X = I_n$ and $\mathcal{R}\left(\begin{bmatrix} A_1 \\ 0 \end{bmatrix}\right)$

\[
\begin{bmatrix} I_k \\ 0 \end{bmatrix}
\]

that
\[
\sin \Theta(V, Y_1)_M = \sin \Theta(X^T V, X^T Y_1) = \sin \Theta\left(X^T V, \begin{bmatrix} I_k \\ 0 \end{bmatrix}\right)
\]

and
\[
\sin \Theta(U, X_1)_K = \sin \Theta((YA)^T U, (YA)^T X_1) = \sin \Theta\left(AY^T U, \begin{bmatrix} A_1 \\ 0 \end{bmatrix}\right) = \sin \Theta\left(AY^T U, \begin{bmatrix} I_k \\ 0 \end{bmatrix}\right).
\]

Thus, assumptions (4.18) imply that
\[
X^T V (V^T M V)^{-\frac{1}{2}} = \begin{bmatrix} I_k \\ 0 \end{bmatrix} C_1 + \begin{bmatrix} 0 \\ I_{n-k} \end{bmatrix} C_2, \quad \|C_2\|_2 = \|\sin \Theta(V, Y_1)_M\|_2 \leq \epsilon
\]
\[
AY^T U (U^T K U)^{-\frac{1}{2}} = \begin{bmatrix} I_k \\ 0 \end{bmatrix} C_3 + \begin{bmatrix} 0 \\ I_{n-k} \end{bmatrix} C_4, \quad \|C_4\|_2 = \|\sin \Theta(U, X_1)_K\|_2 \leq \epsilon
\]

with $C_1^T C_1 + C_2^T C_2 = C_3^T C_3 + C_4^T C_4 = I_k$. This, by $Y^T X = I_n$, $X = [X_1, X_2]$ and $Y = [Y_1, Y_2]$, leads to
\[
V = Y_1 C_1 (V^T M V)^{\frac{1}{2}} + Y_2 C_2 (V^T M V)^{\frac{1}{2}} = Y \hat{C} (V^T M V)^{\frac{1}{2}},
\]
\[
U = X_1 A_1^{-1} C_3 (U^T K U)^{\frac{1}{2}} + X_2 A_2^{-1} C_4 (U^T K U)^{\frac{1}{2}}
\]
\[
= X A^{-1} \hat{C} (U^T K U)^{\frac{1}{2}},
\]
where $\hat{C} = [C_1^T, C_2^T]^T \in \mathbb{R}^{n \times k}$ and $\hat{C} = [C_3^T, C_4^T]^T \in \mathbb{R}^{n \times k}$, both are of orthonormal columns.

Since $U^T V = I_k$, by Thouless’s minimization principal (1.7), we have
\[
\sum_{i=1}^{k} \hat{\lambda}_i = \min_{P_1^T P_1 = I_k, P_2^T P_2 \in \mathbb{R}^{k \times k}} \frac{1}{2} \text{tr}(P_1^T U^T K U P_1 + P_2^T V^T M V P_2);
\]

on the other hand, $\sum_{i=1}^{k} \lambda_i$ is sum of the $k$ positive eigenvalues of $H_{SR}$ given in (4.19).

It has been shown [1, Theorem 2.9] that the Ritz values $\pm \hat{\lambda}_i$ for $i = 1, 2, \ldots, k$, are invariant with respect to the choice of basis $(\hat{U}, \hat{V})$ of $(\mathcal{R}(U), \mathcal{R}(V))$ as long as $\hat{U}^T \hat{V} = I_k$. Thus, we choose a new and special basis $(\hat{U}, \hat{V})$ for $(\mathcal{R}(U), \mathcal{R}(V))$ given by
\[
\hat{V} = Y \hat{C} C_1^{-1} A_1^{\frac{1}{2}}, \quad \hat{U} = X A^{-1} \hat{C} (\hat{C}^T A^{-1} \hat{C})^{-1} C_1 A_1^{-\frac{1}{2}},
\]
which satisfies $\hat{U}^T \hat{V} = I_k$ and $(\mathcal{R}(\hat{U}), \mathcal{R}(\hat{V})) = (\mathcal{R}(U), \mathcal{R}(V))$. The nonsingularity of the matrix $\hat{C}^T A^{-1} \hat{C}$ follows from (4.21), (4.22) and $U^T V = I_k$. With this choice of basis, it follows from (4.23) that
\[
\sum_{i=1}^{k} \lambda_i \leq \sum_{i=1}^{k} \hat{\lambda}_i \leq \frac{1}{2} \text{tr}(\hat{U}^T K \hat{U} + \hat{V}^T M \hat{V}),
\]
and the conclusion (4.20) follows by establishing an upper bound for $\frac{1}{2} \text{tr}(\hat{U}^T K \hat{U} + \hat{V}^T M \hat{V})$. 

To this end, we first notice by using $\hat{C}^T\hat{C} = I_k$ and $\|C_2\|_2 \leq \epsilon$ that
\[
\|(C_1 C_1^T)^{-1}\|_2 = \|(C_1^T C_1)^{-1}\|_2 = \|(I_k - C_2^T C_2)^{-1}\|_2 \leq \frac{1}{1 - \|C_2\|_2^2} \leq \frac{1}{1 - \epsilon^2}
\]
and thus,
\[
\tilde{V}^T M \tilde{V} = A_1^\frac{1}{2} C_1^{-T} \hat{C}^T Y^T M Y \hat{C} C_1^{-1} A_1^\frac{1}{2} = A_1^\frac{1}{2} (C_1 C_1^T)^{-1} A_1^\frac{1}{2} \leq \frac{1}{1 - \epsilon^2} A_1.
\]
Therefore, by Lemma 4.4,
\[
\text{tr}(\tilde{V}^T M \tilde{V}) \leq \sum_{i=1}^k \lambda_i = \sum_{i=1}^k \lambda_i + \epsilon^2 \sum_{i=1}^k \lambda_i + O(\epsilon^4). \tag{4.26}
\]
For $\text{tr}(\hat{U}^T K \hat{U})$, we have
\[
\hat{U}^T K \hat{U} = A_1^{-\frac{1}{2}} \left( C_1^{-T} \hat{C}^T A^{-1} \hat{C} \hat{C}^T A^{-1} \hat{C} \hat{C}^T A^{-1} \right)^{-1} A_1^{-\frac{1}{2}}. \tag{4.27}
\]
Consider the medium matrix in (4.27) and we have
\[
C_1^{-T} \hat{C}^T A^{-1} \hat{C} \hat{C}^T A^{-1} \hat{C} C_1^{-1} = A_1^{-1} C_3 C_3^T A_1^{-1} + C_1^{-T} C_2^T A_2^{-1} C_4 C_4^T A_2^{-1} C_2 C_1^{-1} + C_1^{-T} C_2^T A_2^{-1} C_4 C_3^T A_1^{-1} + A_1^{-1} C_3 C_4^T A_2^{-1} C_2 C_1^{-1}.
\]
Note that
\[
(1 - \epsilon^2) A_1^{-2} \leq A_1^{-1} C_3 C_3^T A_1^{-1} \leq A_1^{-2},
\]
and by $\|C_1^{-1}\|_2 \leq \frac{1}{\sqrt{1 - \epsilon^2}}$ and $\|C_3\|_2 \leq \sqrt{1 + \epsilon^2}$,
\[
\|C_1^{-T} C_2^T A_2^{-1} C_4 C_3^T A_1^{-1} + A_1^{-1} C_3 C_4^T A_2^{-1} C_2 C_1^{-1}\|_2 \leq \frac{2}{\lambda_1 \lambda_k + 1} \cdot \frac{\epsilon^2 \sqrt{1 + \epsilon^2}}{\sqrt{1 - \epsilon^2}},
\]
implying
\[
-\frac{2}{\lambda_1 \lambda_k + 1} \cdot \frac{\epsilon^2 \sqrt{1 + \epsilon^2}}{\sqrt{1 - \epsilon^2}} I_k \leq C_1^{-T} C_2^T A_2^{-1} C_4 C_3^T A_1^{-1} + A_1^{-1} C_3 C_4^T A_2^{-1} C_2 C_1^{-1}.
\]
Therefore, for any sufficiently small $\epsilon$,
\[
C_1^{-T} \hat{C}^T A^{-1} \hat{C} \hat{C}^T A^{-1} \hat{C} C_1^{-1} \geq (1 - \epsilon^2) A_1^{-2} - \frac{2}{\lambda_1 \lambda_k + 1} \cdot \frac{\epsilon^2 \sqrt{1 + \epsilon^2}}{\sqrt{1 - \epsilon^2}} I_k > 0,
\]
which by Lemma 4.4 and (4.27) yields
\[
0 \leq \hat{U}^T K \hat{U} \leq \left( (1 - \epsilon^2) A_1^{-2} - \frac{2}{\lambda_1 \lambda_k + 1} \cdot \frac{\epsilon^2 \sqrt{1 + \epsilon^2}}{\sqrt{1 - \epsilon^2}} A_1 \right)^{-1}. \tag{4.28}
\]
Note that for any sufficiently small $\epsilon$,
\[
\frac{\sqrt{1 + \epsilon^2}}{\sqrt{1 - \epsilon^2}} = 1 + \epsilon^2 + O(\epsilon^4).
\]
Then for any $i = 1, 2, \ldots, k$,
\[
\left( (1 - \epsilon^2) \lambda_i^{-1} - \frac{2}{\lambda_1 \lambda_k + 1} \cdot \frac{\epsilon^2 \sqrt{1 + \epsilon^2}}{\sqrt{1 - \epsilon^2}} \lambda_i \right)^{-1} \tag{4.29}
\]
and
\[
\lambda_i \cdot \left( 1 - \epsilon^2 \left( 1 + \frac{2 \lambda_i^2}{\lambda_1 \lambda_k + 1} \cdot \frac{\sqrt{1 + \epsilon^2}}{\sqrt{1 - \epsilon^2}} \right) \right)^{-1} \tag{4.30}
\]
\[
\begin{align*}
\lambda_i \cdot & \left(1 + \epsilon^2 \left(1 + \frac{2\lambda_i^2}{\lambda_1\lambda_{k+1}} \cdot \frac{\sqrt{1 + \epsilon^2}}{\sqrt{1 - \epsilon^2}} \right) \right) + O(\epsilon^4) \quad (4.31) \\
\lambda_i \cdot & \left(1 + \epsilon^2 \left(1 + \frac{2\lambda_i^2}{\lambda_1\lambda_{k+1}} (1 + \epsilon^2) \right) \right) + O(\epsilon^4), \quad (4.32)
\end{align*}
\]
and according to Lemma 4.4 again, (4.28) gives
\[
\text{tr}(U^TKU) \leq \sum_{i=1}^{k} \lambda_i + \epsilon^2 \sum_{i=1}^{k} \left(\lambda_i + \frac{2\lambda_i^3}{\lambda_1\lambda_{k+1}} \right) + O(\epsilon^4),
\]
which combines with (4.26) leads to (4.20).

Revealed by Theorem 4.5, we know that, similar to the Hermitian eigenvalue problem, the accuracy of the sum of the \(k\) positive Ritz values is also proportional to the square of the accuracy of the deflating subspaces. As \(\|\sin\Theta(U_j,Y_j)_M\|_2\) and \(\|\sin\Theta(U_j,X_j)_K\|_2\) converge to zero linearly with factor \(\sqrt{\sigma}\) (by Theorem 4.3), we know that \(\sum_{i=1}^{k} \hat{\lambda}_i\) converges to \(\sum_{i=1}^{k} \lambda_i\) with factor \(\sigma\).

Corollary 1. Suppose \(\{U_j\}\) and \(\{V_j\}\) are the sequences generated by Algorithm 3.1 with \(U_0\) and \(V_0\) satisfying (4.15) and (4.13), respectively. Assume \(\zeta = \frac{\lambda_{k+1}}{\lambda_k} > 1\) and \(\sigma\) is arbitrary satisfying \(\frac{1}{\sqrt{\sigma}} \leq \sigma < 1\). Then for sufficiently large \(j\), the \(k\) positive eigenvalues \(\hat{\lambda}_i\) for \(i = 1, 2, \ldots, k\) of \(H_j\) defined by (3.8) satisfy
\[
0 \leq \sum_{i=1}^{k} \hat{\lambda}_i - \sum_{i=1}^{k} \lambda_i \leq \sigma^j \sum_{i=1}^{k} \left(\lambda_i + \frac{\lambda_i^3}{\lambda_1\lambda_{k+1}} \right) + O(\sigma^j). \quad (4.33)
\]

Proof. According to Theorem 4.3, we know that
\[
\|\sin\Theta(V_j,Y_1)_M\|_2 \leq \sqrt{\sigma^j} \|\sin\Theta(V_0,Y_1)_M\|_2 \leq \sqrt{\sigma^j},
\]
\[
\|\sin\Theta(U_j,X_1)_K\|_2 \leq \sqrt{\sigma^j} \|\sin\Theta(U_0,X_1)_K\|_2 \leq \sqrt{\sigma^j}.
\]
Now use Theorem 4.5 to conclude (4.33) for any sufficient large \(j\). \(\square\)

5. Numerical experiments. In this section, we evaluate our proposed PPCG_lrep method from several aspects. In the next subsection, we first show that PPCG_lrep is a much more efficient implementation of the inverse power iteration (3.5)-(3.6) than the other straightforward one, where the involved linear systems in (3.5) and (3.6) are solved by the preconditioned CG (PCG); our next goal is to test several types of preconditioners for \(K\) and \(M\). Both of these tasks are carried out in MATLAB (R2011b) on made-up LREPs with randomly generated dense matrices and sparse matrices from University of Florida Sparse Matrix Collection [4]. Our final goal is to evaluate the parallelization capability of PPCG_lrep when columns in \(U_j\) and \(V_j\) are computed in parallel. For that purpose, we code the algorithm PPCG_lrep in C language with the inner Algorithms 3.2 and 3.3 parallelized by OpenMP, and test two practical problems arising from TDDFT. All of our tests are conducted on a PC under Linux system with Intel Core i5-3230M CPU (2.6 GHz) and 4 GB memory.

5.1. Numerical tests for dense random and sparse made-up problems. Our reported numerical experiments of PPCG_lrep (using the block implementation via Algorithms 3.4 and 3.5 to compute \((U_j, V_j)\)) are based on several types of preconditioners, namely, Iden, Diag, Chol, IChol, and SSOR. Specifically, Iden and Diag are diagonal preconditioners using the identity \(I_n\) and diagonal matrices of \(K\) and \(M\), respectively; Chol and IChol are triangular preconditioners resulting from
the Cholesky factorization for dense case and incomplete Cholesky factorization for sparse case, respectively; SSOR uses the symmetric successive over relaxation method as preconditioner. For the stopping criterion, we terminate Algorithm 3.1 whenever either (3.9) or (3.10) is satisfied, where both $\epsilon_r$ and $\epsilon_\chi$ are set to be $10^{-6}$.

The initial $U_0$ and $V_0$ are simply $[e_1, \ldots, e_k]$, and the maximum numbers of outer iteration and inner iteration are chosen to be 100 and 500, respectively. In addition, we vary the tolerance ($\text{cg.tol}$) in the inner CG from $10^{-5}$ to $10^{-10}$ in order to investigate the overall performance of PPCG_lrep with various accurate approximations of (3.1a) and (3.1b).

For our first test example, we use a pair of matrices $(K, M)$ of order $n = 500$ generated randomly by

$$K = \text{randn}(n), \quad K = K' \ast K, \quad M = \text{randn}(n), \quad M = M' \ast M$$

in MATLAB. Our goal is to compute the first 4 smallest positive eigenvalues $0 < \lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \lambda_4$. In the following Table 5.1, we compare the performance of PPCG_lrep with the straightforward way via the preconditioned CG (PCG) for solving the subproblems (3.1a) and (3.1b).

| Table 5.1. | The CPU time(s) for the case $n = 500$ and $k = 4$ |
|---|---|---|---|---|---|---|
| $\text{cg.tol}$ | Using PCG for (3.3) and (3.4) | PPCG_lrep |
| | Iden | Diag | Chol | Iden | Diag | Chol | SSOR |
| $10^{-10}$ | 78.31 | 350.15 | 3.06 | 6.60 | 5.92 | 0.28 | 27.21 |
| $10^{-9}$ | 74.53 | 342.30 | 2.61 | 5.99 | 4.74 | 0.22 | 22.09 |
| $10^{-8}$ | 74.22 | 345.08 | 3.06 | 5.31 | 3.94 | 0.21 | 17.72 |
| $10^{-7}$ | 74.38 | 339.19 | 3.48 | 4.55 | 3.12 | 0.20 | 13.35 |
| $10^{-6}$ | 73.48 | 338.52 | 3.62 | 3.84 | 2.14 | 0.19 | 9.12 |
| $10^{-5}$ | 72.76 | 334.73 | 2.94 | 3.08 | 1.49 | 0.16 | 3.53 |

The numerical results listed in Table 5.1 clearly indicate that PPCG_lrep is a much more efficient implementation for (3.3) and (3.4) than the direct way by PCG. For more information about the numerical results, in Table 5.2, we report the relative eigenvalue errors (RESeig), the total number of inner CG iterations (inner iter.) and the number of outer loop iterations (outer iter.) of PPCG_lrep. The relative error RESeig concerns the accuracy of the sum of eigenvalues associated with the approximate eigenvalues (Ritz values) $\tilde{\lambda}_i^{(j)}$ for $i = 1, \ldots, k$ at the $j$th iteration and is defined as

$$\text{RESeig} = \frac{|\sum_{i=1}^{k} \lambda_i^{\text{exact}} - \sum_{i=1}^{k} \tilde{\lambda}_i^{(j)}|}{\sum_{i=1}^{k} \tilde{\lambda}_i^{(j)}},$$

where $\lambda_i^{\text{exact}}$ for $i = 1, \ldots, k$ are computed by MATLAB function eig for the matrix $H$ given in (1.4).

We observe from Table 5.2 that

1. the outer loop iterations roughly remain the same (about 25) for different preconditioners, but the efficiency in terms of the executing CPU time and the number of inner CG steps is largely dependent on the specific preconditioner, and
2. the relative eigenvalue error RESeig (i.e., the accuracy of the outer loop) decreases when the relevant systems are computed more accurately by the
inner loop CG; in particular, for $I_{\text{den}}$ and $\text{Chol}$, the relative error $\text{RESeig}$ are roughly of the same order as the tolerance $\text{cg}_\text{tol}$ for the inner CG.

In Figure 5.1, we further demonstrate the behavior history of the relative error for individual eigenvalue $\lambda_i^{(j)}$ for $i=1, 2, 3, 4$ with respect to the outer loop iteration $j$. It is observed that various preconditioners perform differently for $\lambda_1^{(j)}$ and $\lambda_2^{(j)}$, which usually converge faster than $\lambda_3^{(j)}$ and $\lambda_4^{(j)}$.

Next, we test $\text{PPCG}_{\text{lrep}}$ on two sparse problems, where the matrices $K$ and $M$ are from University of Florida Sparse Matrix Collection [4]. The features of these matrices are presented in Table 5.3, and in the case when the two matrices from the collection have different dimensions, we extract out the leading principal submatrix of the larger one to give $K$ or $M$ of equal size. Moreover, in this test, we set the tolerance in the inner CG to be $\text{cg}_\text{tol} = 10^{-8}$ to compute the first 4 smallest positive eigenvalues. The computation outputs of $\text{PPCG}_{\text{lrep}}$ are listed in Table 5.4, and we observed that $\text{IChol}$ is a good preconditioner for $\text{PPCG}_{\text{lrep}}$ in the test problems.

### Table 5.2. Numerical results of $\text{PPCG}_{\text{lrep}}$ for $n = 500$ and $k = 4$

| $\text{cg}_\text{tol}$ | $\text{RESeig}_{\text{outer iter.}}$ | $\text{RESeig}_{\text{inner iter.}}$ | $\text{RESeig}_{\text{outer iter.}}$ | $\text{RESeig}_{\text{inner iter.}}$ |
|-------------------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|
| $10^{-10}$              | $4.29 \times 10^{-6}$        | $2.33 \times 10^{-4}$        | $1.72 \times 10^{-8}$        | $2.43 \times 10^{-7}$        |
|                         | 25                            | 25                            | 25                            | 25                            |
|                         | 14831                         | 11484                         | 25                            | 8232                         |
| $10^{-9}$               | $5.77 \times 10^{-6}$        | $2.55 \times 10^{-4}$        | $1.72 \times 10^{-8}$        | $2.37 \times 10^{-6}$        |
|                         | 25                            | 21                            | 25                            | 21                            |
|                         | 14321                         | 9500                          | 50                            | 6651                          |
| $10^{-8}$               | $5.30 \times 10^{-8}$        | $2.37 \times 10^{-5}$        | $5.39 \times 10^{-8}$        | $1.93 \times 10^{-5}$        |
|                         | 25                            | 21                            | 25                            | 21                            |
|                         | 12699                         | 7913                          | 47                            | 5337                          |
| $10^{-7}$               | $5.33 \times 10^{-7}$        | $2.68 \times 10^{-4}$        | $5.27 \times 10^{-7}$        | $1.91 \times 10^{-4}$        |
|                         | 10906                         | 6151                          | 41                            | 4004                          |
| $10^{-6}$               | $5.38 \times 10^{-6}$        | $3.60 \times 10^{-4}$        | $5.16 \times 10^{-6}$        | $2.25 \times 10^{-5}$        |
|                         | 21                            | 13                            | 25                            | 17                            |
|                         | 9114                          | 4222                          | 35                            | 2783                          |
| $10^{-5}$               | $5.87 \times 10^{-5}$        | $1.87 \times 10^{-2}$        | $5.05 \times 10^{-5}$        | $1.11 \times 10^{-4}$        |
|                         | 21                            | 13                            | 21                            | 9                             |
|                         | 7370                          | 2930                          | 29                            | 1057                          |

### Table 5.3. The sparse matrices $K$ and $M$

| Problem          | $n$   | $K$    | $M$    |
|------------------|------|-------|-------|
| SPARSE TEST 1    | 10001| blowy bq | ted\_B |
| SPARSE TEST 2    | 9604 | fv1    | finan512 |

As the final part of this subsection, we extend our numerical evaluation of $\text{PPCG}_{\text{lrep}}$ by comparing another gradient type method, namely, the block 4-D steepest descent algorithm ($\text{B4dSD}$) proposed in [10]. In this test, we set the tolerance $\text{cg}_\text{tol} = 10^{-6}$ and $\epsilon_r = \epsilon_x = 10^{-8}$ for $\text{PPCG}_{\text{lrep}}$, while we terminate $\text{B4dSD}$ whenever the number of iterations is larger than 5000, or each of the relative residual associated with (1.5) at the approximated solution $(\hat{\lambda}_i, \hat{z}_i)$ satisfies

$$}\frac{\|H \hat{z}_i - \hat{\lambda}_i \hat{\lambda}_i\|_1}{\|H\|_1 \|\hat{\lambda}_i\|_1 + \|\hat{\lambda}_i\|_1} \leq 10^{-8}$$

for $i = 1, 2, \ldots, k$. 

The relative eigenvalue error of $\hat{\lambda}_1$ w.r.t. outer loop iteration $j$ for the case $n = 500$ and $k = 4$

Table 5.4. Numerical results of PPCG$_{lrep}$ for sparse problems

| SPARSE TEST 1 | Iden | Diag | tChol | SSOR |
|---------------|------|------|-------|------|
| CPU time(s)   | 157.10 | 120.77 | 0.86  | 75.80 |
| outer iter.   | 69    | 49    | 9     | 13   |
| inter iter.   | 18926 | 12769 | 22    | 2120 |

| SPARSE TEST 2 | Iden | Diag | tChol | SSOR |
|---------------|------|------|-------|------|
| CPU time(s)   | 73.79 | 52.19 | 55.46 | 78.38 |
| outer iter.   | 473   | 477   | 473   | 473  |
| inter iter.   | 8012  | 4599  | 1076  | 2432 |

To demonstrate the performance, besides the test matrices in the form of (5.2), we use another kind of random symmetric and positive definite matrices

$$[Q,\sim] = qr(rand(n)), \quad D = \text{diag}(\text{rand}(n,1) + 0.01), \quad K = Q^*D*Q \quad (5.2)$$

for $K$ and for $M$ similarly. Tables 5.5 and 5.6 summarizes parts of our experiments for (5.1) and (5.2), respectively. Note that for B4dSD, various types of preconditioners including the identity (Iden), the Cholesky (Chol) and the CG (CG) are used for testing, and we refer to [10] for the details of B4dSD as well as the preconditioning technique. By observation, one can see that PPCG$_{lrep}$ can be more efficient for
these randomly generated problems. In particular, for the type of (5.1), the condition numbers of $K$ and $M$ are generally large and PPCG_{lrep} can achieve more accurate solutions (with smaller RESeig) than B4dSD.

**Table 5.5.** Numerical results of PPCG_{lrep} and B4dSD for the case (5.1)

| $n$ | cond($K$) | $k$ | CPU times(s) | RESeig | outer iter. | inner iter. | CPU times(s) | RESeig | outer iter. | inner iter. | CPU times(s) | RESeig | outer iter. | inner iter. | CPU times(s) | RESeig | outer iter. | inner iter. |
|-----|----------|-----|-------------|--------|-------------|-------------|-------------|--------|-------------|-------------|-------------|--------|-------------|-------------|-------------|--------|-------------|-------------|
| 1000 | 1.57 x 10$^7$ | 4   | 11.07       | 9.23   | 0.32        | 44.35       | 9.09 x 10$^{-4}$ | 7.34 x 10$^{-4}$ | 1.98 x 10$^{-4}$ |
|     |           | 10  | 56.48       | 18.94  | 1.23        | 102.43      | 5.12 x 10$^{-5}$ | 1.22 x 10$^{-2}$ | 2.54 x 10$^{-2}$ |
| 1500 | 4.98 x 10$^7$ | 4   | 31.55       | 7.26   | 0.78        | 60.22       | 4.62 x 10$^{-6}$ | 1.82 x 10$^{-3}$ | 1.94 x 10$^{-2}$ |
|     |           | 10  | 175.69      | 65.56  | 2.44        | 143.45      | 3.53 x 10$^{-5}$ | 1.53 x 10$^{-2}$ | 1.56 x 10$^{-2}$ |
| 2000 | 3.87 x 10$^7$ | 4   | 141.18      | 53.60  | 1.84        | 145.82      | 1.08 x 10$^{-5}$ | 1.95 x 10$^{-2}$ | 8.04 x 10$^{-2}$ |
|     |           | 10  | 21875       | 7925   | 13          | 6500        | 37          | 3           | 1361          |

**Table 5.6.** Numerical results of PPCG_{lrep} and B4dSD for the case (5.2)

| $n$ | cond($K$) | $k$ | CPU times(s) | RESeig | outer iter. | inner iter. | CPU times(s) | RESeig | outer iter. | inner iter. | CPU times(s) | RESeig | outer iter. | inner iter. | CPU times(s) | RESeig | outer iter. | inner iter. |
|-----|----------|-----|-------------|--------|-------------|-------------|-------------|--------|-------------|-------------|-------------|--------|-------------|-------------|-------------|--------|-------------|-------------|
| 1000 | 98.98    | 4   | 1.07        | 1.30   | 1.46        | 15.44       | 1.89 x 10$^{-4}$ | 1.89 x 10$^{-4}$ | 9.65 x 10$^{-5}$ |
|     |           | 10  | 507.43      | 130.00 | 8.05        | 241.64      | 7.65 x 10$^{-5}$ | 1.39 x 10$^{-2}$ | 3.39 x 10$^{-3}$ |
| 1500 | 99.47    | 4   | 1.41        | 1.93   | 3.89        | 27.96       | 1.12 x 10$^{-4}$ | 4.23 x 10$^{-5}$ | 4.16 x 10$^{-5}$ |
|     |           | 10  | 21875       | 7925   | 13          | 6500        | 37          | 3           | 1361          |
| 2000 | 84.74    | 4   | 2.96        | 3.54   | 12.78       | 62.44       | 5.79 x 10$^{-5}$ | 5.43 x 10$^{-3}$ | 2.69 x 10$^{-3}$ |
|     |           | 10  | 387         | 460    | 157         | 2128        | 1.55 x 10$^{-3}$ | 1.85 x 10$^{-3}$ | 1.97 x 10$^{-3}$ |
5.2. Numerical tests for two practical LREPs in parallel computation.
In this subsection, we evaluate the efficiency of the parallel implementation of PPCG\_lrep, where the columns of $U_j$ and $V_j$ are computed in parallel; this results in the implementation of PPCG\_lrep using Algorithms 3.2 and 3.3 instead of their block version Algorithms 3.4 and 3.5, respectively. For that purpose, we coded the two versions (i.e., the block version PPCGb\_lrep with Algorithms 3.4 and 3.5, and the parallel version PPCGp\_lrep using Algorithms 3.2 and 3.3) in C language. For comparison purpose, we only utilize the generic reference BLAS (REFERENCE BLAS Version 3.5.0) instead of certain optimized version like MKL or ATLAS. The block version PPCGb\_lrep mainly uses level-3 BLAS operations. For PPCGp\_lrep, we parallelize (using 4 cores) the $k$ calls of Algorithms 3.2 and 3.3 by OpenMP in each outer loop iteration.

Both algorithms are applied to solve two LREPs for computing the optical spectra of the Na$_2$ sodium clusters and silane (SiH$_4$). The test matrices are obtained from the plane wave-pseudopotential turbo TDDFT code, which is part of the Quantum ESPRESSO(QE) package. In particular, the dimensions of Na$_2$ and SiH$_4$ are 1864 and 5660, respectively, and we set the stopping criteria $\epsilon_r = \epsilon_x = 10^{-8}$ for both versions to compute the first $k = 4$ smallest positive eigenvalues. The numerical results are displayed in Table 5.7. Although these are preliminary numerical tests for PPCGp\_lrep, they show the improvements on the block version PPCGb\_lrep, which is based on level-3 BLAS operations.

| Problem | PPCGp\_lrep | PPCGb\_lrep |
|---------|-------------|-------------|
|         | Iden Diag   | Iden Diag   |
| Na$_2$  |          |             |
| CPU time(s) | 17.53 | 29.48 |
| RES(eig)  | $1.64 \times 10^{-6}$ | $1.64 \times 10^{-8}$ |
| outer iter. | 73 | 73 |
| inner iter. | 1818 | 2411 |
| SiH$_4$  |          |             |
| CPU time(s) | 105.94 | 142.84 |
| RES(eig)  | $3.20 \times 10^{-6}$ | $3.11 \times 10^{-8}$ |
| outer iter. | 121 | 121 |
| inner iter. | 1113 | 1055 |

6. Conclusions. Relying upon the subspace version of the Thouless minimization principle for the linear response eigenvalue problem, in this paper, we have introduced an alternating block (between $U$ and $V$) minimization scheme to compute the desired eigenpairs characterized by the Thouless minimization principle. The connection of this alternating scheme with the inverse power iteration facilitates us to perform the convergence analysis. To make this scheme numerically efficient, we further formulated the computation of each iteration as a projected preconditioned CG step, which can be implemented either in a block version rich in level-3 BLAS operations, or in parallel. Preliminary numerical experiments are reported and demonstrate its behaviors on make-up LREPs as well as on two practical problems from TDDFT.

Acknowledgements. The authors would like to thank Dr. Junqi Hu at Shanghai University of Finance and Economics for the discussion on the time-dependent
density functional theory for the excited-state calculations. They also thank the anonymous referees for the comments and suggestions.

REFERENCES

[1] Z. Bai and R. C. Li, Minimization principle for linear response eigenvalue problem, I: Theory, *SIAM J. Matrix Anal. Appl.*, 33 (2012), 1075–1100.
[2] Z. Bai and R. C. Li, Minimization principles for the linear response eigenvalue problem II: Computation, *SIAM J. Matrix Anal. Appl.*, 34 (2013), 392–416.
[3] J. Brabec, L. Lin, M. Shao, N. Govind, C. Yang, Y. Saad and E. G. Ng, Efficient algorithms for estimating the absorption spectrum within linear response TDDFT, *J. Chem. Theory Comput.*, 11 (2015), 5197–5208.
[4] T. Davis and Y. Hu, The University of Florida sparse matrix collection, *ACM Trans. Math. Softw.*, 38 (2011), 1–25.
[5] U. Flaschka, W.-W. Lin and J.-L. Wu, A KQZ algorithm for solving linear-response eigenvalue equations, *Linear Algebra Appl.*, 165 (1992), 93–123.
[6] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1985.
[7] A. V. Knyazev and M. E. Argentati, Principal angles between subspaces in an $A$-based scalar product: Algorithms and perturbation estimates, *SIAM J. Matrix Anal. Appl.*, 23 (2002), 2008–2040.
[8] T. Li, R.-C. Li and W.-W. Lin, A symmetric structure-preserving $IQR$ algorithm for linear response eigenvalue problems, *Linear Algebra Appl.*, 520 (2017), 191–214.
[9] J. Nocedal and S. Wright, *Numerical Optimization*, Springer, 2nd edition, 2006.
[10] D. Rocca, Z. Bai, R.-C. Li and G. Galli, A block variational procedure for the iterative diagonalization of non-Hermitian random-phase approximation matrices, *J. Chem. Phys.*, 136 (2012), 034111.
[11] D. Rocca, R. Gebauer, Y. Saad and S. Baroni, Turbo charging time-dependent density-functional theory with Lanczos chains, *J. Chem. Phys.*, 128 (2008), 928–934.
[12] E. Runge and E. K. U. Gross, Density-functional theory for time-dependent systems, *Phys. Rev. Lett.*, 52 (1984), 997–1000.
[13] Z. Teng and R.-C. Li, Convergence analysis of Lanczos-type methods for the linear response eigenvalue problem, *J. Comput. Appl. Math.*, 247 (2013), 17–33.
[14] Z. Teng, Y. Zhou and R.-C. Li, A block Chebyshev–Davidson method for linear response eigenvalue problems, *Adv. in Comput. Math.*, 42 (2016), 1103–1128.
[15] D. J. Thouless, Vibrational states of nuclei in the random phase approximation, *Nuclear Physics*, 22 (1961), 78–95.
[16] D. J. Thouless, *The Quantum Mechanics of Many-Body Systems*, Academic, 1972.
[17] L. H. Zhang, W. W. Lin and R. C. Li, Backward perturbation analysis and residual-based error bounds for the linear response eigenvalue problem, *BIT*, 55 (2015), 869–896.
[18] L. H. Zhang, J. Xue and R. C. Li, Rayleigh–Ritz approximation for the linear response eigenvalue problem, *SIAM J. Matrix Anal. Appl.*, 35 (2014), 765–782.

Received February 2017; 1st revision May 2018; final revision August 2018.

E-mail address: lxkate@163.com
E-mail address: shenchungen@usst.edu.cn
E-mail address: zhang.leihong@mail.shufe.edu.cn