TRACE MINIMIZATION METHOD VIA PENALTY FOR LINEAR RESPONSE EIGENVALUE PROBLEMS

YADAN CHEN
China National Clearing Center, Beijing 100048, China
and
School of Economics and Management
University of the Chinese Academy of Sciences, Beijing 100190, China

YUAN SHEN*
School of Applied Mathematics, Nanjing University of Finance & Economics
Nanjing 210023, China

SHANSHAN LIU
Hefei 168 Middle School, Hefei 230031, China

(Communicated by Chao Xu)

Abstract. In various applications, such as the computation of energy excitation states of electrons and molecules, and the analysis of interstellar clouds, the linear response eigenvalue problem, which is a special type of the Hamiltonian eigenvalue problem, is frequently encountered. However, traditional eigen-solvers may not be applicable to this problem owing to its inherently large scale. In fact, we are usually more interested in computing some of the smallest positive eigenvalues. To this end, a trace minimum principle optimization model with orthogonality constraint has been proposed. On this basis, we propose an unconstrained surrogate model called trace minimization via penalty, and we establish its equivalence with the original constrained model, provided that the penalty parameter is larger than a certain threshold. By avoiding the orthogonality constraint, we can use a gradient-type method to solve this model. Specifically, we use the gradient descent method with Barzilai–Borwein step size. Moreover, we develop a restarting strategy for the proposed algorithm whereby higher accuracy and faster convergence can be achieved. This is verified by preliminary experimental results.

1. Introduction.

1.1. Problem description. In this study, the following linear response (LR) eigenvalue problem [14, 6] is considered:

\[
\begin{bmatrix}
0 & M \\
K & 0
\end{bmatrix}
\begin{bmatrix}
x \\
y
\end{bmatrix} = \lambda
\begin{bmatrix}
x \\
y
\end{bmatrix} \tag{1}
\]

2020 Mathematics Subject Classification. Primary: 65L15, 15A18; Secondary: 90C30.

Key words and phrases. Trace minimization, linear response eigenvalue problem, gradient descent, Barzilai-Borwein step size.

The second author is supported by the National Social Science Foundation of China under Grants 19AZD018, 20BGL028, 19BGL205, 17BTQ063.

*Corresponding author: Yuan Shen.
where $K, M \in \mathbb{R}^{n \times n}$ are both Hermitian, and one of them is required to be positive definite [19].

The LR eigenvalue problem (1), also known as the random phase approximation eigenvalue problem, arises from computing excitation states (energies) of physical systems in the study of collective motion of many-particle systems, ranging from silicon nanoparticles and nanoscale materials to interstellar clouds ([5, 12, 15]).

By letting

$$H = \begin{bmatrix} 0 & M \\ K & 0 \end{bmatrix} \in \mathbb{R}^{2n \times 2n} \quad \text{and} \quad z = \begin{bmatrix} x \\ y \end{bmatrix},$$

the LR eigenvalue problem (1) can be rewritten in the following compact form:

$$Hz = \lambda z.$$  \hspace{1cm} (3)

It is easy to observe that the matrix $H$ is generally non-symmetric, hence, problem (1) is considered significantly more difficult than the symmetric eigenvalue problem, as the underlying matrix is non-Hermitian. This problem can also be recognized as a special case of the Hamiltonian eigenvalue problem, as $H$ is a Hamiltonian matrix. The LR eigenvalue problem, which is important in the computation of energy excitation states of electrons and molecules, is usually of large scale. The matrix dimension for a molecule of a reasonable size can easily increase to tens of millions. Consequently, traditional algorithms for the Hamiltonian eigenvalue problem (e.g., [4, 9]) would be ineffective, as they are usually designed for small-scale problems. A large number of recent studies (e.g., [1, 2, 8, 13, 16, 17, 18, 19, 21]) have been concerned with the development of efficient numerical algorithms and simulation techniques for excitation-response calculations of molecules for materials design in energy science. These algorithms have been reported to be efficient for (1) (e.g., [2]). However, it is necessary to develop dedicated algorithms for (1) considering its special structure.

As shown in [1], the matrix $H$ in the form of (2) has $n$ pairs of real eigenvalues $\pm \lambda_i$ such that

$$-\lambda_n \leq -\lambda_{n-1} \leq \cdots \leq -\lambda_1 < 0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n,$$

and their squares $\lambda_i^2$, $i = 1, 2, \cdots, n$, are eigenvalues of $KM$ (or $MK$).

In particular, we are usually more interested in computing the $k$ smallest positive eigenvalues and the corresponding eigenvectors. It has been shown in [1] that this problem permits the following trace minimization principle for finding the $k$ smallest positive eigenvalues of $H$:

$$\sum_{j=1}^{k} \lambda_j(H) = \min_{X,Y \in \mathbb{R}^{n \times k}} \{ \text{tr}(X^T KX + Y^T MY) : 2X^TY = I \},$$

which implies that the minimization principle (5) for the sum of the $k$ smallest positive eigenvalues is equivalent to the original problem (1).

1.2. Method description. Based on (5), we propose an unconstrained trace minimization model and establish its equivalence to the trace minimization model with a suitably selected penalty parameter.

By introducing a Lagrange multiplier matrix $\Omega \in \mathbb{R}^{k \times k}$ into model (5), we obtain its first-order optimality condition as follows:

$$\begin{bmatrix} 0 & M \\ K & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} X \\ Y \end{bmatrix} \Omega, \quad 2X^TY = I,$$  \hspace{1cm} (6)
where $\Omega$ is assumed to be symmetric.

Let a minimizer of (5) be $(X^*, Y^*, \Omega^*)$, let $\Omega^* = Q\Lambda^*Q^\top$ be the orthogonal eigenvalue decomposition of $\Omega^*$, and let $U^* = X^*Q$ and $V^* = Y^*Q$. Then, $(U^*, V^*, \Lambda^*)$ satisfies the following relation:

$$\begin{bmatrix} 0 & M \\ K & 0 \end{bmatrix} \begin{bmatrix} U \\ V \end{bmatrix} = \begin{bmatrix} U \\ V \end{bmatrix} \Lambda, \quad 2U^\top V = I,$$

which represents a truncated eigenvalue decomposition of the matrix $H$ associated with its $k$ smallest positive eigenvalues.

Inspired by the trace minimization model described in [20], which is employed as a surrogate model for a similar eigenvalue problem, we can also apply the trace minimization model as follows to solve (5):

$$\min_{X, Y \in \mathbb{R}^{n \times k}} f_\mu(X, Y) := \frac{1}{2} \text{tr}(X^\top KX + Y^\top MY) + \frac{\mu}{4} \|2X^\top Y - I\|_F^2. \quad (8)$$

A major theoretical result in this paper is the establishment of the equivalence between problem (5) and the unconstrained optimization problem (8), provided that the penalty parameter $\mu > 0$ takes suitable finite values. Moreover, as the objective function of problem (5) is a quadratic penalty function, we can apply gradient-type methods to solve the trace minimization problem (8), as discussed later.

It is noticeable that Bai and Li proposed a locally optimal 4D CG algorithm (LO4DCG) for (1) in [2] and an improved algorithm (LOBP4dCG) [3], which are reported to be highly efficient for the LR eigenvalue problem. Teng et al. proposed two Lanczos-type methods that converge significantly faster than some classical methods [18]. However, these algorithms require costly orthogonalization or SVD operations at each iteration. As the dimension of the matrix increases, they become increasingly time consuming. Hence, these algorithms are less suitable for a large-scale setting. Even some partial SVD solvers such as LMSVD or SLRP, which only compute a few singular values [10, 11], are computationally expensive, and this is particularly true when the number $k$ of desired eigenpairs is large.

Thus, Teng et al. proposed a block Chebyshev–Davidson method [19], the per-iteration cost of which is considerably lower than that of LOBP4dCG, and it was reported to outperform LOBP4dCG significantly (see Table 2 in [19] for details). Based on the classical Golub–Kahan bidiagonalization factorization [7], a weighted Golub–Kahan–Lanczos bidiagonalization algorithm has been proposed [21]. It is highly accurate and has satisfactory convergence speed.

1.3. Contribution. The contributions of this study are as follows: 1) We establish the equivalence between our trace minimization model (8) and the trace minimization model (5) proposed in [1]. 2) We present a gradient-type algorithm for computing the $k$ smallest positive eigenvalues and the corresponding eigenvectors of a non-Hermitian matrix $H$ in an LR problem, that is, an LR eigenvalue problem. Moreover, we propose a restarting strategy that enables faster convergence and higher accuracy for our algorithm.

The main difference between our method and the EIGPEN algorithm proposed in [20] is that the LR eigenvalue problem is considerably more complicated than eigenvalue decomposition. In fact, a large number of theoretical results have been proved, and several efficient numerical algorithms have been developed for eigenvalue decomposition problems; however, the matrix in the LR eigenvalue problem
is non-Hermitian, and many results related to the properties of Hermitian matrices cannot be applied here. Avoiding the expensive eigenvalue decomposition at each iteration, our method is highly suitable for solving large-scale problems compared with existing algorithms for LR eigenvalue problems. Moreover, the computations in our algorithm are primarily matrix multiplications, thus ensuring good parallel scalability.

1.4. Organization. The rest of this paper is organized as follows. In Section 2, we describe the proposed algorithm for (8) and present the restarting technique to accelerate convergence. In Section 3, we establish the convergence properties of the proposed algorithm. Numerical experiments are presented in Section 4. The paper is concluded in the final section.

2. Description of algorithms. The trace minimization model (8) is an unconstrained nonconvex minimization problem. Several well-studied methods, such as the gradient descent method, the conjugate gradient (CG) method, and quasi-Newton method, can be applied to this problem. As we intend to increase the dimensional scalability as well as the parallelizability of the algorithm, we select a gradient-type method here.

2.1. Gradient method for trace minimization model. Let us consider the trace minimization (8):

\[
\min_{X,Y \in \mathbb{R}^{n \times k}} f_\mu(X, Y) := \frac{1}{2} \text{tr}(X^\top K X + Y^\top M Y) + \frac{\mu}{4} \|2X^\top Y - I\|_F^2. \tag{9}
\]

It is easy to obtain the gradient of \( f_\mu(X, Y) \) as follows:

\[
\partial f_\mu(X, Y) = (\partial_X f_\mu(X, Y), \partial_Y f_\mu(X, Y)), \tag{10}
\]

where

\[
\partial_X f_\mu(X, Y) = K X + \mu Y (2Y^\top X - I), \tag{11}
\]

\[
\partial_Y f_\mu(X, Y) = M Y + \mu X (2X^\top Y - I). \tag{12}
\]

Starting from \((X^0, Y^0)\), a simple steepest descent method can be formulated as the following iterative scheme:

\[
X^{j+1} = X^j - \alpha^j \partial_X f_\mu(X^j, Y^j), \tag{13}
\]

\[
Y^{j+1} = Y^j - \alpha^j \partial_Y f_\mu(X^j, Y^j), \tag{14}
\]

where \(\alpha^j\) is a suitable step size.

As the step size is critical to the performance of gradient-type methods, we present a strategy for selecting \(\alpha^j\) by using the Barzilai–Borwein (BB) step size. By letting

\[
S_X^j := X^j - X^{j-1}, \quad Z_X^j := \partial_X f_\mu(X^j, Y^j) - \partial_X f_\mu(X^{j-1}, Y^{j-1}), \tag{15}
\]

and

\[
S_Y^j := Y^j - Y^{j-1}, \quad Z_Y^j := \partial_Y f_\mu(X^j, Y^j) - \partial_Y f_\mu(X^{j-1}, Y^{j-1}), \tag{16}
\]

the BB step size is defined as

\[
\alpha^j = (\text{tr}((S_X^j)^\top Z_X^j) + \text{tr}((S_Y^j)^\top Z_Y^j))/\|Z_X^j\|_F^2 + \|Z_Y^j\|_F^2. \tag{17}
\]

It is well known that the BB step size can be quite large at some iterations, leading to nonmonotone convergence behavior. Accordingly, we apply a simple heuristic line search technique to shorten the step size. Whenever necessary, backtracking
is performed on the step size, namely, $\alpha^j = \alpha^j \delta^h$ with $\delta \in (0, 1)$, where $h$ is the smallest positive integer satisfying the following condition (Armijo rule):

$$f_\mu(X^j - \alpha^j \delta^h \partial X f_\mu(X^j, Y^j), Y^j - \alpha^j \delta^h \partial Y f_\mu(X^j, Y^j)) \leq f_\mu(X^j, Y^j) - \alpha^j \rho \|\partial f_\mu(X^0, Y^0)\|_F^2,$$

(18)

where $\rho > 0$ is a small constant.

The termination condition is

$$\|\partial X f_\mu(X^j, Y^j)\|_F + \|\partial Y f_\mu(X^j, Y^j)\|_F \leq \epsilon,$$

(19)

where $\epsilon > 0$ is a prescribed tolerance.

As the solution of (9) is not the eigenpairs of the LR eigenvalue problem (1), to approximate the eigenpairs, a Rayleigh–Ritz (RR) step should be applied to compute the Ritz pairs at the end of the algorithm. Specifically, in our context, the RR step corresponding to a given matrix $X^j, Y^j \in \mathbb{R}^{n \times k}$ is defined as follows:

1. Orthogonalize and normalize $X^j, Y^j$ to obtain $\hat{X}^j, \hat{Y}^j$.
2. Compute the projection $(\hat{X}^j)^\top K \hat{X}^j + (\hat{Y}^j)^\top M \hat{Y}^j$ and its eigenvalue decomposition $V^\top \Sigma V$.
3. Formulate the Ritz pairs as the matrix pair $(\hat{Z}, \Sigma)$, where $\hat{Z} = \begin{bmatrix} \hat{X}^j \\ \hat{Y}^j \end{bmatrix}$.

Finally, we arrive at the following algorithm:

**Algorithm 1.** Initialize $X^0, Y^0 \in \mathbb{R}^{n \times k}$, estimate $\mu \in (\lambda_j, \lambda_n)$.

Set $\epsilon > 0$ and $j = 0$.

While $\|\partial f_\mu(X^j, Y^j)\|_F > \epsilon$ and $j < \text{maxit}$ do

Step 1 Compute the step size $\alpha^j$ by (17);

Step 2 Variable update by (13) and (14);

Step 3 If (18) is not satisfied, $\alpha^j = \delta \alpha^j$, go to Step 2;

increment $j$ and continue.

end while

Execute the RR procedure $(\hat{Z}, \Sigma) = \text{RR}(X^j, Y^j)$.

return $\hat{Z}, \Sigma$.

2.2. Enhancement by restarting strategy. Empirically, Algorithm 1 is effective for LR eigenvalue problems. Generally, gradient-type methods can rapidly reduce the objective value, but the amount of reduction can become extremely small as the iterate approaches the solution.

In the trace minimization suggested in [20], it has been observed that restarting the gradient method with a modified initial point can usually accelerate convergence and achieve higher accuracy. Following this idea, we describe a restarting strategy that applies multiple RR steps for the LR trace minimization, and put forward an improved algorithm using this strategy.

Before presenting the new algorithm, we define a diagonal matrix $\hat{D}$ by

$$\hat{D}_{ii} = \max (0, \frac{\Sigma_{ii}}{\mu})^{\frac{1}{2}} \quad i = 1, 2, \ldots, k.$$

(20)

The scheme of the new algorithm consists of outer and inner loops. The inner loop is essentially the same as that of Algorithm 1, whereas the stopping rule is

$$\|\partial X f_\mu(X^j, Y^j)\|_F + \|\partial Y f_\mu(X^j, Y^j)\|_F \leq \epsilon_i,$$

(21)

where $\epsilon_i \to 0$ as the outer iteration number $i$ tends to infinity. Then, an RR step is performed to obtain the Ritz pairs $(\hat{Z}, \Sigma)$, and the termination criterion is
checked for the outer loop. If the algorithm does not stop, then a smaller tolerance
\( \epsilon_{i+1} = \delta \epsilon_i \) is set, where \( \delta \in (0, 1) \), and a better iterate is then constructed from which the algorithm restarts the next round of inner iterations. The complete algorithm with the restarting strategy is shown below. It performs better than without the restarting strategy in the numerical experiments presented in Section 4.

**Algorithm 2.** Initialize \( \bar{X}^0, \bar{Y}^0 \in \mathbb{R}^{n \times k} \).

Set \( \epsilon_0, \delta > 0 \) and \( i = j = 0 \).

while \( \| \partial X f_\mu(X^j, Y^j) \|_F^2 + \| \partial Y f_\mu(X^j, Y^j) \|_F^2 > \epsilon_{\text{out}} \) and \( i < \text{maxit}_{\text{out}} \) do

\( X^0 = X^i, Y^0 = Y^i \).

Estimate \( \mu \in (\lambda_j, \lambda_n) \) as in (42) with \( X^0, Y^0 \).

while \( \| \partial f_\mu(X^j, Y^j) \|_F > \epsilon_i \) and \( j < \text{maxit}_{\text{in}} \) do
do the same steps as in the loop of Algorithm 1.

end while

Compute \( \hat{D} \) as in (20).

Set \( \begin{bmatrix} \bar{X} \\ \bar{Y} \end{bmatrix} = \begin{bmatrix} \hat{X} \\ \hat{Y} \end{bmatrix} \hat{D} \).

Compute \( 2(\bar{X}^\top \bar{Y}) = R^\top R \) via Cholesky decomposition.

Set \( \bar{X}^{i+1} = XR^{-1}, \bar{Y}^{i+1} = YR^{-1} \).

Set \( \epsilon_{i+1} = \epsilon_i \delta \epsilon \).

Increment \( i \) and continue.

end while

return \( \hat{Z}, \Sigma \).

3. **Theoretical analysis.** Herein, we establish the equivalence between (5) and (8). If an appropriate parameter \( \mu \) is selected, that is, if \( \mu > \lambda_k \), these two problems are equivalent. The set of global minimizers of (8) spans a \( k \)-dimensional eigenspace associated with the \( k \) smallest eigenvalues of \( H \) defined in (2). We also analyze the restarting strategy and establish its theoretical soundness.

3.1. **Model equivalence.**

**Lemma 3.1.** Let \( \begin{bmatrix} X \\ Y \end{bmatrix} \) be a stationary point of (8). Then,

\[
X^\top Y = Y^\top X. \tag{22}
\]

Moreover, the first-order optimality condition of (8) can be written as

\[
\begin{bmatrix} 0 & M \\ K & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} X \\ Y \end{bmatrix} (I - 2X^\top Y) \mu. \tag{23}
\]

**Proof.** As \( \begin{bmatrix} X \\ Y \end{bmatrix} \) is a stationary point of (8), we have \( \partial X f_\mu(X, Y) = \partial Y f_\mu(X, Y) = 0 \). By using (11) and (12), we have

\[
KX = Y(I - 2Y^\top X)\mu, \tag{24}
\]

\[
MY = X(I - 2X^\top Y)\mu. \tag{25}
\]

Multiplying both sides of (24) by \( X^\top \), we obtain \( X^\top KX = \mu X^\top Y - 2\mu X^\top YY^\top X \).
After rearranging, we have

\[
X^\top Y = \frac{1}{\mu} X^\top KX + 2X^\top YY^\top X. \tag{26}
\]
As the right-hand side of (26) is symmetric, $X^TY$ is also symmetric; thus, (22) is proved. It follows from (22), (24), and (25) that (23) also holds, which completes the proof.

By Lemma 3.1, every stationary point of $f_\mu$ spans an invariant subspace of $H$. In particular, $X = Y = 0$ always yields a trivial stationary point. The next lemma shows that any stationary point of (8) can be expressed in terms of the truncated eigenpairs of $H$.

**Lemma 3.2.** Let $(U, V, D) \in \mathbb{R}^{n \times k} \times \mathbb{R}^{n \times k} \times \mathbb{R}^{k \times k}$ denote $k$ truncated eigenpairs of $H$ such that

$$
\begin{bmatrix}
0 & M \\
K & 0
\end{bmatrix}
\begin{bmatrix}
U \\
V
\end{bmatrix}
= 
\begin{bmatrix}
U \\
V
\end{bmatrix}
D,
$$

(27)

where $D$ is a diagonal matrix. A matrix $\begin{bmatrix}X \\ Y\end{bmatrix} \in \mathbb{R}^{2n \times k}$ is a stationary point of (8) if and only if

$$
\begin{bmatrix}
X \\
Y
\end{bmatrix}
= 
\begin{bmatrix}
U \\
V
\end{bmatrix}
\left[P(I-D/\mu)\right]^{1/2}Q^T,
$$

(28)

where $\mu > 0$, $Q \in \mathbb{R}^{k \times k}$ is an orthogonal matrix, and $P \in \mathbb{R}^{k \times k}$ is a diagonal projection matrix with diagonal entries

$$
P_{ii} = \begin{cases}
0, & \text{if } \mu \leq D_{ii} \\
0 \text{ or } 1, & \text{otherwise}
\end{cases}
$$

(29)

In particular, $\begin{bmatrix}X \\ Y\end{bmatrix}$ is a rank-$k$ stationary point only if $P = I$ and $\mu I - D \succ 0$.

**Proof.** We only provide a proof when $\begin{bmatrix}X \\ Y\end{bmatrix}$ is of full-rank, as the rank-deficient cases can be proved similarly although the proofs are more tedious and notationally involved.

We assume that $\begin{bmatrix}X \\ Y\end{bmatrix}$ is a full-rank stationary point that spans an invariant subspace of $H$ according to Lemma 3.1. As every $k$-dimensional invariant subspace of $H$ can be spanned by a set of $k$ truncated eigenvectors, we can write

$$
\begin{bmatrix}
X \\
Y
\end{bmatrix}
= 
\begin{bmatrix}
U \\
V
\end{bmatrix}
W,
$$

(30)

where the matrix $\begin{bmatrix}U \\ V\end{bmatrix}$ consists of $k$ unit eigenvectors of $H$, and $W \in \mathbb{R}^{k \times k}$ is a nonsingular matrix. It follows from (27) that

$$
U^TKU + V^TMV = 2U^TVD = D.
$$

(31)

Upon substituting (30) into (23) and multiplying both sides by $(Y^T, X^T)$, we obtain

$$
\begin{align*}
X^TKX + Y^TMY &= 2X^TY(I - 2X^TY)\mu \\
\iff W^TDW &= W^T(I - W^TW)\mu \\
\iff I - D/\mu &= WW^T \\
\iff W &= (I - D/\mu)^{1/2}Q^T
\end{align*}
$$

(32)
for some orthogonal $Q \in \mathbb{R}^{k \times k}$ (which can possibly hold only if $\mu > D_{ii}$ for $i = 1, 2, \cdots, k$). In the last equivalence in (32), we used the non-singularity of $W$. This completes the proof. \hfill \Box

Now, we establish the equivalence between problems (8) and (5) for suitable $\mu$ values.

**Theorem 3.3.** Problem (8) is equivalent to (5) if and only if

$$\mu > \lambda_k. \tag{33}$$

Moreover, any minimizer of $f_\mu$ defined in (8) for $\mu > \lambda_k$ has the form

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} U \\ V \end{bmatrix} (I - \Lambda_k / \mu)^{1/2} W, \tag{34}$$

where $U, V,$ and $\Lambda_k$ satisfy (7), and $W \in \mathbb{R}^{k \times k}$ is an arbitrary orthogonal matrix.

**Proof.** It readily follows from Lemma 3.1 that condition (33) is necessary for the existence of a rank-$k$ stationary point. On the other hand, assuming that $\mu$ satisfies (33) and using Lemma 3.2, it suffices to consider the representation

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} U \\ V \end{bmatrix} W,$$

where $\begin{bmatrix} U \\ V \end{bmatrix}$ consists of any $k$ truncated eigenvectors of $H$ and $W \in \mathbb{R}^{k \times k}$. Hence, we obtain

$$f_\mu(X, Y) = \frac{1}{2} \text{tr}(W^T DW) + \frac{\mu}{4} \|W^T W - I\|_F^2,$$

where $D \in \mathbb{R}^{k \times k}$ is a diagonal matrix with $k$ eigenvalues of $H$ on the diagonal corresponding to the truncated eigenvectors in $\begin{bmatrix} U \\ V \end{bmatrix}$. The rest of the proof is the same as that of Theorem 2.1 in [20]. \hfill \Box

### 3.2. Analysis of restarting strategy

Herein, we demonstrate how restarting can improve the convergence speed of Algorithm (2). Let

$$\begin{bmatrix} \tilde{X} \\ \tilde{Y} \end{bmatrix} := \arg\min_{X, Y \in \mathbb{R}^{n \times k}} \left\{ f_\mu(X, Y) : \begin{bmatrix} X \\ Y \end{bmatrix} \in \mathcal{S} \right\}, \tag{35}$$

where $\begin{bmatrix} X \\ Y \end{bmatrix} \in \mathcal{S}$ denotes that every column of $\begin{bmatrix} X \\ Y \end{bmatrix}$ is in the subspace $\mathcal{S}$. Let $\begin{bmatrix} X^J \\ Y^J \end{bmatrix}$ be the iterate generated by Algorithm 2 after $J$ iterations. If $\begin{bmatrix} X^J \\ Y^J \end{bmatrix} \in \mathcal{S}$, we obtain

$$f_\mu(\tilde{X}, \tilde{Y}) \leq f_\mu(X^J, Y^J). \tag{36}$$

Hence, $\begin{bmatrix} \tilde{X} \\ \tilde{Y} \end{bmatrix}$ is a “better point,” and we intend to construct it from the RR-step output $(\tilde{Z}, \Sigma) = RR(X^J, Y^J)$.

To this end, let us consider the following subspace trace minimization problem:

$$\begin{bmatrix} U \\ V \end{bmatrix} = \min_{X, Y \in \mathbb{R}^{n \times d}} \left\{ \text{tr}(X^T KX + Y^T MY) : 2X^T Y = I, \begin{bmatrix} X \\ Y \end{bmatrix} \in \mathcal{S} \right\}, \tag{37}$$

where $d$ is the dimension of the subspace.
Theorem 3.4. Let \( \text{span}\left\{ \begin{bmatrix} X^J \\ Y^J \end{bmatrix}, \begin{bmatrix} X^J \\ Y^J \end{bmatrix} \right\} \subseteq S \) be with dimension \( d \geq k \), and \( \begin{bmatrix} U \\ V \end{bmatrix} \) be defined in (37) so that \( U^T KU + V^T MV = \Sigma \) is diagonal with diagonal elements arranged in ascending order. Then, \( \begin{bmatrix} \tilde{X} \\ \tilde{Y} \end{bmatrix} \) in (35) has the form \( \begin{bmatrix} U_k \\ V_k \end{bmatrix} \) \( D \), where \( U_k \) and \( V_k \) consist of the first \( k \) columns of \( U \) and \( V \), respectively, and \( D \in \mathbb{R}^{k \times k} \) is a diagonal matrix, the \( i \)-th diagonal element of which is

\[
D_{ii} = \max(0, 1 - \Sigma_{ii}/\mu)^{1/2}, \quad i = 1, 2, 3, \ldots, k. \tag{38}
\]

Proof. We notice that the RR step \((\hat{Z}, \Sigma) = RR(X^J, Y^J)\), where \( \hat{Z} = \begin{bmatrix} U \\ V \end{bmatrix} \), is equivalent to solving (37) with \( S = \text{span}\left\{ \begin{bmatrix} X^J \\ Y^J \end{bmatrix}, \begin{bmatrix} X^J \\ Y^J \end{bmatrix} \right\} \). Hence, we obtain \( U^T KU + V^T MV = \Sigma \), where \( \Sigma \) is diagonal.

As \( \begin{bmatrix} U \\ V \end{bmatrix} \in \mathbb{R}^{2n \times d} \) is a basis of \( S \), the solution of (35) can be expressed as \( \begin{bmatrix} \tilde{X} \\ \tilde{Y} \end{bmatrix} = \begin{bmatrix} U \\ V \end{bmatrix} Q \) for some \( Q \in \mathbb{R}^{d \times k} \). Substituting \( \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} U \\ V \end{bmatrix} \) into (35), and noting that \( U^T KU + V^T MV = \Sigma \) and \( 2U^T V = I \), we have that the problem (35) is reduced to

\[
\min_{Q \in \mathbb{R}^{d \times k}} f_\mu(UQ, VQ) = \frac{1}{2} \text{tr}(Q^T \Sigma Q) + \frac{\mu}{4} \|Q^T Q - I\|_F^2. \tag{39}
\]

As \( \Sigma \) is a diagonal matrix, \( Q = (D \ 0)^T \) is a solution of (39). Therefore, we can construct \( Q \) as \( Q = (D \ 0)^T \), with the diagonal matrix \( D \) defined in (38) according to Theorem 2.1 in [20], and a “better point” \( \begin{bmatrix} \tilde{X} \\ \tilde{Y} \end{bmatrix} = \begin{bmatrix} U \\ V \end{bmatrix} Q = \begin{bmatrix} U_k \\ V_k \end{bmatrix} \) \( D \) is constructed. This completes the proof. \( \square \)

4. Numerical experiments. Herein, we present numerical experiments with synthetic data matrices conducted to test the performance of our algorithms as solvers for computing the \( k \)-smallest eigenvalues and the corresponding eigenvectors of the LR eigenvalue problem (1). The proposed algorithms without and with restarting (Algorithms 1 and 2, respectively) were implemented in MATLAB (R2016a), and their outstanding performance is demonstrated by the numerical results.

4.1. Experimental setting. We generate the symmetric positive definite matrices \( K \) and \( M \) as follows: (1) We first generate the diagonal matrices \( D_1, D_2 \in \mathbb{R}^{n \times n} \) with positive diagonal entries uniformly distributed in \([0,1]\). (2) Then, we generate the random matrices \( U_1, U_2 \in \mathbb{R}^{n \times n} \) with i.i.d. Gaussian entries, and perform QR factorization on them, so that their columns are orthogonal. (3) Finally, we construct the matrices by \( K = U_1 D_1 U_1^T, M = U_2 D_2 U_2^T \).

After the generation of the matrices \( K \) and \( M \), we can solve the unconstrained problem defined in (8) by the proposed algorithms. To demonstrate the numerical effectiveness of our algorithms, we compare them with LOBP4dCG [2] \(^1\) and the

\(^1\) Available at https://web.cs.ucdavis.edu/~bai/LReigsoftware/
block Chebyshev–Davidson method (Bcheb-dav) \[19\], which are popular efficient solvers expressly developed for the LR eigenvalue problem.

Two methods can be used to generate the initial points \(X^0, Y^0 \in \mathbb{R}^{n \times k}\):

- **Sparse form:** We randomly select \(k\) different number \((i_1, i_2, \ldots, i_k)\) from \([1, n]\), and set \(X^0 = [e_{i_1}, e_{i_2}, \ldots, e_{i_k}] / \sqrt{2}\) and \(Y^0 = X^0\), where \(e_{i_1}, e_{i_2}, \ldots, e_{i_k} \in \mathbb{R}^n\) and \(2(X^0)^\top Y^0 = I\).

- **Dense form:** We first generate a random matrix \(U \in \mathbb{R}^{n \times k}\), then orthogonalize \(U\) to obtain \(X^0\) such that \(2(X^0)^\top X^0 = I\), and finally set \(Y^0 = X^0\).

In our experiments, the difference between the two methods is small. For simplicity, we always use the second method for the test algorithms.

Moreover, we usually use the normalized residual norm as an optimality criterion, as suggested in \([1]\). Specifically, an alternative criterion, namely, the Frobenius norm of the gradient (NG), is also used for our algorithms as a reference.

- The normalized residual norm (NRN) for the \(i\)-th approximate eigenpair \((\lambda_i, z_i)\) is defined as

\[
\frac{\|H z_i - \lambda_i z_i\|_1}{(\|H\|_1 + \lambda_i)\|z_i\|_1},
\]

where \(z_i\) represents the \(i\)-th column of \(Z\). Unless otherwise specified, NRN always denotes the maximal value of the NRN for all eigenpairs.

- The Frobenius norm of the gradient (NG) is defined as

\[
\|\partial_X f_\mu(X^j, Y^j)\|_F^2 + \|\partial_Y f_\mu(X^j, Y^j)\|_F^2.
\]

The NRN measures the accuracy of the eigenpairs obtained by the algorithms, whereas the NG measures the accuracy of the output of our algorithm compared with the true solution of (8).

We let Bcheb-dav and our algorithms stop when either the NRN is smaller than a prescribed tolerance \(tol\) (default: \(tol = 10^{-6}\)), or a maximal number of iteration \(maxit\) (default: \(maxit = 10^4\)) is reached. It should be noticed that Bcheb-dav utilizes an incremental strategy whereby the rank of the working variables \(X\) and \(Y\) (as well as other related variables) increases with the number of iterations. Therefore, more iterations may be required to reach the prescribed rank \(k\) although the stopping criterion has already been satisfied. As LOBP4dCG is based on a different model, to make a fair comparison, we stop it using its own stopping criterion, with a properly selected tolerance (usually smaller than \(tol\), for example, \(tol = 10^{-12}\)) such that its final NRN is comparable with that of the other test algorithms.

Moreover, to realize our algorithm, it is necessary to set an appropriate value for parameter \(\mu\). In \([20]\), the authors described a tractable strategy for selecting the penalty parameter \(\mu\) such that it is reasonably close to \(\lambda_k\):

\[
\mu = \max\{c_1, c_2 \max(\text{diag}(X^0)\top K X^0 + (Y^0)\top M Y^0)\},
\]

where \(c_1 > 0\) and \(c_2 > 1\) are two suitable constants. However, \(c_1\) and \(c_2\) should be determined. For simplicity, we empirically set \(\mu = 0.5\) in most experiments. Other necessary parameters in our algorithm are set as \(\delta = 0.5\) and \(\rho = 10^{-4}\). In Bcheb-dav, the critical parameters are as follows: (1) \(nwant\), which is the number of eigenvalues to be computed, and it is set to \(k\), and (2) \(blk\), which is the block size in the blockwise filtering stage, and its optimal setting is related to \(k\). Here, we empirically set \(blk = 2\) when \(k = 5\), and \(blk = 10\) when \(k = 50\). In LOBP4dCG, the critical parameters are as follows: (1) \(precond\_one\), which determines whether
a preconditioner is used for computing the subproblems. Here, as the default preconditioner cannot boost performance under our problem settings, we do not use one. (2) \( N_e \), which is the order of the Krylov subspace for creating the extended search space, and it is always set to 2, as suggested by the authors of the code. The other parameters in Bcheb-dav and LOBP4dCG are set to their default values.

At each iteration of LOBP4dCG, orthogonalization should be performed, which can be increasingly expensive as the dimension of matrix and the number of eigenpairs to be determined increase, whereas the per-iteration cost of our algorithm has better dimensional scalability. In addition, our algorithm primarily involves matrix multiplications, which can be implemented in parallel, that is, when a parallel computation infrastructure is available, the computation time of our algorithm can be lower. By contrast, orthogonalization may not be implemented in parallel because of its inherently serial nature. For these reasons, the per-iteration cost of LOBP4dCG is usually considerably higher than that of our algorithm, and therefore the comparison of the iteration numbers may be less meaningful. Instead, we use computation time as the major performance index of the test algorithms.

4.2. Numerical results of Algorithm 1. We first present the numerical results of Algorithm 1. The iteration progress of the NRN for several different problem settings is shown in Figures 1 and 2. The iteration progress of the NG of our algorithm is shown in Figures 3 and 4 as reference. Moreover, LOBP4dCG is excluded when \( k \geq 50 \) owing to its excessively long computation time.

In Figures 3 and 4, it can be seen that the NG decreases rapidly, implying that our algorithm can rapidly obtain an accurate solution of model (8). In Figures 1 and 2, it can be seen that the convergence speed of Algorithm 1 is overall satisfactory compared with that of LOBP4dCG and Bcheb-dav. Although our algorithm is slower than LOBP4dCG in terms of iteration number, recalling that its per-iteration cost is lower, it is faster in terms of computation time, as seen in Tables 1 and 2. Bcheb-dav converges slower than our algorithm in terms of iteration number, and its NRN sequence is nonmonotonically decreasing, whereas that of our algorithm is monotonically decreasing. Moreover, our algorithm converges very slowly at the final stage of the iteration process, and it even fails to attain the prescribed tolerance \( 10^{-6} \) before the maximum iteration number is reached when \((n,k) = (1000, 50)\).

As the per-iteration costs of different algorithms are different, the number of iterations may not be a good measure of numerical efficiency. It is more informative to show the computation time rather than the iteration number. The average computation time (in s) of the test algorithms over 10 random instances under several different parameter settings is shown in Tables 1 and 2.

### Table 1. Numerical results on random problems.

| \((n, k)\) | LOBP4dCG | Bcheb-dav | Alg. 1 |
|------------|-----------|-----------|--------|
|            | NRN       | time      | NRN    | time   |
| (1000, 5)  | 3.37e-7   | 5.158     | 4.370e-7 | 1.213   |
| (2000, 5)  | 2.792e-6  | 16.156    | 5.319e-7 | 5.425   |
| (4000, 5)  | 2.052e-7  | 46.707    | 7.597e-7 | 29.476  |
| (6000, 5)  | 2.964e-7  | 219.598   | 7.230e-7 | 127.189 |

It is seen from Table 1 that when \( k = 5 \), Algorithm 1 is usually faster than LOBP4dCG and Bcheb-dav in terms of computation time, and its performance
Figure 1. Normalized residual norm vs iteration with $k = 5$ (upper left: $n = 1000$; upper right: $n = 2000$; lower left: $n = 4000$; lower right: $n = 6000$)

Table 2. Numerical results on random problems.

| $(n, k)$ | Bcheb-dav | Alg. 1 |
|---------|-----------|--------|
|         | NRN       | time   | NRN       | time   |
| (1000, 50) | 8.179e-7 | 1.924  | 6.617e-6  | 5.286  |
| (2000, 50) | 1.351e-7 | 10.493 | 9.445e-7  | 27.038 |
| (4000, 50) | 1.713e-7 | 45.975 | 9.239e-7  | 55.971 |
| (6000, 50) | 4.060e-7 | 237.142| 9.587e-7  | 189.574|

advantage becomes more pronounced when $n$ is larger. For example, our algorithm is slightly slower than Bcheb-dav when $n = 1000$, but it is considerably faster when $n = 6000$. Specifically, when the dimension $n$ increases from 1000 to 6000, and $k$ is fixed at 5, the computation time of our algorithm increases from 1.39 to 17.50 (approximately 12 times), whereas that of LOBP4dCG increases from 5.15 to 219.59 (approximately 40 times), and that of Bcheb-dav increases from 1.21 to 127.18 (over 100 times). This may indicate that our algorithm has both higher efficiency and better dimensional scalability.

Table 2 shows that Bcheb-dav may be preferable for larger $k$ ($k = 50$). For example, Bcheb-dav is approximately 175% faster than our algorithm when $n =$
1000; but its performance advantage diminishes as \( n \) grows. For example, Algorithm 1 is approximately 25% faster than Bcheb-dav when \( n = 6000 \).

The above observations indicate that \( k \) has a strong impact on performance. We now investigate the effect on performance of larger and smaller values of \( k \). We first fix \( k = 2 \), and the results are shown in Table 3. We then fix the ratio \( n/k \) at 10 and 5, and the results are shown in Table 4. Because of memory limitations, the maximal setting of the dimension \( n \) is only 2000.

The results shown in Table 3 are similar to those in Table 1. Hence, our algorithm is indeed suitable for small \( k \). For larger \( k \), the performance advantage of Bcheb-dav is conspicuous. For example, Bcheb-dav is over 10 times as fast as our algorithm when \((n, k) = (2000, 400)\). In fact, Bcheb-dav uses an incremental technique that begins with a null set, searches for eigenpairs in a small subspace, and adds the eigenpairs to the set. In contrast, the dimension of working variables in our algorithm is always \( n \times k \), which requires significantly more computation time during the iterative process.

These observations indicate that 1) the speed of our algorithm is satisfactory, and its performance advantage is more pronounced when \( k \) is relatively small, 2) the dimensional scalability of our algorithm is better than that of LOBP-ldCG and Bcheb-dav, and 3) Algorithm 1 has difficulty achieving higher accuracy.
Figure 3. Frobenius norm of gradient vs iteration with $k = 5$
(upper left: $n = 1000$; upper right: $n = 2000$; lower left: $n = 4000$; lower right: $n = 6000$)

Table 3. Numerical results on random problems with smaller $k$.

| $(n, k)$ | LOBP4dCG | Bcheb-dav | Alg. 1 |
|----------|-----------|-----------|--------|
|          | NRN   | time     | NRN   | time   | NRN   | time   |
| (1000,2) | 9.353e-10 | 1.034     | 2.347e-7 | 1.920  | 8.977e-7 | 0.522  |
| (2000,2) | 2.639e-9  | 4.906     | 8.093e-7 | 5.481  | 9.122e-7 | 0.928  |
| (4000,2) | 1.283e-9  | 21.372    | 9.518e-7 | 35.761 | 9.729e-7 | 1.159  |
| (6000,2) | 2.050e-9  | 59.513    | 9.235e-7 | 218.249| 6.868e-7 | 5.190  |

4.3. Numerical results of Algorithm 2.
As seen previously, Algorithm 1 can rapidly decrease the NRN to approximately $10^{-6}$, which is quite satisfactory for the LR eigenvalue problem. However, it is also observed that Algorithm 1 has difficulty attaining higher accuracy. Thus, we proposed the restarting strategy, and our algorithm with restarting (Algorithm 2) is expected to be more accurate. To confirm this, we now present the relevant numerical results.

The initialization and parameter settings are similar to those in the previous tests. The stopping rule is also similar, and the tolerance is $10^{-9}$. Some parameters
Figure 4. Frobenius norm of gradient vs iteration with $k = 50$
(upper left: $n = 1000$; upper right: $n = 2000$; lower left: $n = 4000$; lower right: $n = 6000$)

Table 4. Numerical results on random problems with larger $k$.

| $(n, k)$     | Bcheb-dav |          | Alg. 1 |          |
|--------------|-----------|----------|--------|----------|
|              | NRN       | time     | NRN    | time     |
| (500, 50)    | 1.721e-8  | 0.572    | 9.607e-6 | 2.462    |
| (750, 75)    | 2.133e-8  | 1.740    | 9.994e-6 | 6.769    |
| (1000,100)   | 1.299e-8  | 2.945    | 9.902e-6 | 16.409   |
| (1500,150)   | 4.218e-8  | 10.376   | 9.999e-6 | 24.900   |
| (2000,200)   | 2.782e-8  | 19.507   | 7.717e-6 | 64.872   |
| (500,100)    | 1.713e-8  | 1.557    | 9.989e-6 | 7.355    |
| (750,150)    | 1.606e-8  | 2.310    | 9.996e-6 | 20.053   |
| (1000,200)   | 2.837e-8  | 6.128    | 9.993e-6 | 46.477   |
| (1500,300)   | 2.647e-8  | 15.524   | 9.815e-6 | 85.150   |
| (2000,400)   | 1.692e-8  | 28.886   | 4.970e-6 | 415.430  |
appearing in Algorithm 2 are set as follows: $\epsilon_0 = 10^{-6}$, $\delta_\epsilon = 10^{-1}$, $\text{maxit}_{\text{in}} = 3000$, and $\text{maxit}_{\text{out}} = 5$ (which implies at most four restarting rounds).

For simplicity, we only consider three problem settings, and the results are shown in Figures 5, 6 and Table 5. First, it is observed in Figure 6 that Algorithm 2 can always decrease the NG to approximately $10^{-15}$. Figure 5 shows that Algorithm 2 can obtain a solution satisfying $\text{NRN} < 10^{-9}$ under the problem settings $(n, k) = (2000, 5), (4000, 5)$. This implies that the restarting strategy increases accuracy and accelerates the convergence of our algorithm.

Table 5. Numerical results on random problems ($tol = 10^{-9}$).

| $(n, k)$ | LOBP4dCG | Bcheb-dav | Alg. 2 |
|---------|----------|-----------|-------|
|         | NRN      | time      | NRN   | time |
| (1000,5)| 1.741e-6 | 4.556     | 5.682e-10 | 1.641  | 1.969e-8 | 29.810 |
| (2000,5)| 5.263e-8 | 13.573    | 6.082e-9  | 6.988  | 9.740e-10 | 80.980 |
| (4000,5)| 3.956e-8 | 77.154    | 6.694e-9  | 44.313 | 9.257e-10 | 315.360 |

5. Conclusions. We developed new algorithms with both dimensional scalability and parallelizability to effectively solve large-scale LR eigenvalue problems arising from a wide range of applications. Based on previous work on this problem, we reformulated the original constrained optimization model as an unconstrained optimization model called trace minimization via penalty. We theoretically established the equivalence of the proposed unconstrained model and the original constrained model, provided that the penalty parameter was appropriately selected.
A gradient-type method was applied to solve the unconstrained optimization problem, and as orthogonality was not required at each iteration, the algorithm could be implemented in parallel, thus demonstrating its applicability for large-scale problems. To accelerate the convergence of the algorithm, we proposed a restarting strategy, which markedly improved the performance of the proposed algorithm and exhibited high accuracy. Compared with previous methods, our algorithm does not require matrix factorization such as SVD at each iteration, which is crucial for a high-dimensional setting in LR eigenvalue problems. Numerical experiments demonstrated the effectiveness of our algorithms in terms of both computation time and dimensional scalability.

The performance of our algorithms can be further improved from several aspects, including convergence acceleration and reduction of the per-iteration cost, possibly through better choices of the step size and the incremental technique. Moreover, the preconditioning of the data matrix should be investigated.

Acknowledgments. The authors would like to express their sincere gratitude to Professor Yín Zhang, Yaodong Yu and Xin Liu for their valuable suggestions and comments. Special thanks are also due to Dr. Zhongming Teng who kindly provided us with the code of Bcheb-dav.

REFERENCES

[1] Z. Bai and R. Li, Minimization principles for the linear response eigenvalue problem I: Theory, *SIAM J. Matrix Anal. Appl.*, 33 (2012), 1075–1100.
[2] Z. Bai and R. Li, Minimization principles for the linear response eigenvalue problem II: Computation, *SIAM J. Matrix Anal. Appl.*, 34 (2013), 392–416.
[3] Z. Bai, R. Li and W. Lin, Linear response eigenvalue problem solved by extended locally optimal preconditioned conjugate gradient methods, Sci. China Math., 59 (2016), 1443–1460.

[4] P. Benner, H. Fassbender and M. Stoll, A Hamiltonian Krylov-Schur-type method based on the symplectic Lanczos process, Linear Algebra Appl., 435 (2011), 578–600.

[5] M. E. Casida, Time-dependent density-functional response theory for molecules, Recent Advances in Density Functional Methods, (1995), 155–192.

[6] U. Flaschka, W. Lin and J. Wu, A KQZ algorithm for solving linear-response eigenvalue equations, Linear Algebra Appl., 165 (1992), 93–123.

[7] G. Golub and W. Kahan, Calculating the singular values and pseudo-inverse of a matrix, J. Soc. Indust. Appl. Math. Ser. B Numer. Anal., 2 (1965), 205–224.

[8] M. Gruning, A. Marini and X. Gonze, Exciton-plasmon states in nanoscale materials: Breakdown of the Tamm-Dancoff approximation, Nano Letters, 9 (2009), 2820–2824.

[9] W. Lin, P. van Dooren and Q. Xu, Equivalent Characterizations of Periodical Invariant Subspaces, NCTS Preprints Series 1998-8, National Center for Theoretical Sciences, Math. Division, National Tsing Hua University, 1998.

[10] X. Liu, Z. Wen and Y. Zhang, Limited memory block Krylov subspace optimization for computing dominant singular value decompositions, SIAM J. Sci. Comput., 35 (2013), A1641–A1668.

[11] X. Liu, Z. Wen and Y. Zhang, An efficient Gauss-Newton algorithm for symmetric low-rank product matrix approximations, SIAM J. Optim., 25 (2015), 1571–1608.

[12] M. J. Lucero, A. M. N. Niklasson, S. Tretiak and M. Challacombe, Molecular-orbital-free algorithm for excited states in time-dependent perturbation theory, J. Chem. Phys., 129 (2008), 064114.

[13] M. T. Lusk and A. E. Mattsson, High-performance computing for materials design to advance energy science, MRS Bulletin, 36 (2011), 169–174.

[14] J. Olsen and P. Jorgensen, Linear and nonlinear response functions for an exact state and for an MCSCF state, J. Chem. Phys., 82 (1985), 3235–3264.

[15] G. Onida, L. Reining and A. Rubio, Electronic excitations: Density-functional versus many-body Green’s function approaches, Rev. Modern Phys., 74 (2002), 601–659.

[16] D. Rocca, D. Lu and G. Galli, Ab initio calculations of optical absorption spectra: Solution of the Bethe-Salpeter equation within density matrix perturbation theory, J. Chem. Phys., 133 (2010), 164109.

[17] Y. Saad, J. R. Chelikowsky and S. M. Shontz, Numerical methods for electronic structure calculations of materials, SIAM Rev., 52 (2010), 3–54.

[18] Z. Teng and R. Li, Convergence analysis of Lanczos-type methods for the linear response eigenvalue problem, J. Comput. Appl. Math., 247 (2013), 17–33.

[19] Z. Teng, Y. Zhou and R. Li, A block Chebyshev-Davidson method for linear response eigenvalue problems, Adv. Comput. Math., 42 (2016), 1103–1128.

[20] Z. Wen, C. Yang, X. Liu and Y. Zhang, Trace-penalty minimization for large-scale eigenspace computation, J. Sci. Comput., 66 (2016), 1175–1203.

[21] H. Zhong and H. Xu, Weighted Golub-Kahan-Lanczos bidiagonalization algorithms, Electron. Trans. Numer. Anal., 47 (2017), 153–178.

Received June 2020; 1st revision March 2021; 2nd revision August 2021; early access December 2021.

E-mail address: chenyadan@lsec.cc.ac.cn
E-mail address: ocsiban@126.com
E-mail address: 769767821@qq.com