Advances in Applied Mathematics and Mechanics
Adv. Appl. Math. Mech., Vol. **18**, No. 1, pp. 93-108

DOI: 10.4208/aamm.OA-2023-0024
February 2026

Inexact Iterative WYD Method for Eigenvalue Problems

Fangyi Zheng, Jianyang Luo and Shuli Sun*

LTCS, Department of Mechanics and Engineering Science, College of Engineering, Peking University, Beijing 100871, China

Received 15 January 2023; Accepted (in revised version) 6 May 2025

Abstract. In this paper, we propose a new inexact iterative subspace projection method for real symmetric positive definite generalized eigenvalue problems based on the WYD (an abbreviation of the initials of the proposers: Wilson, Yuan, and Dickens [1]) method. Firstly an analysis of the convergence condition of the approximate eigenvectors is given when $\mathbf{A}\mathbf{r}_{k+1}=\lambda\mathbf{B}\mathbf{r}_k$ is inexactly solved by the Krylov subspace methods. Then the inexact iterative WYD (IIWYD) method is constructed, which utilizes the approximate Ritz subspace generated by the WYD method with an inexact solver as the main search space. The IIWYD method improves the quality of the search space during the iterative process, significantly reduces the number of iteration steps, and improves the overall computational efficiency and stability. The results of numerical experiments show that the IIWYD method is more efficient and stable compared to the locally optimal block preconditioned conjugate gradient (LOBPCG) method and the Jacobi-Davidson (JD) method. In addition, we also discuss the effects of the refined strategy and the conjugate strategy in our method.

AMS subject classifications: 65F15, 65F10

Key words: Eigenvalue problems, iterative method, WYD, LOBPCG, Jacobi-Davidson.

1 Introduction

For a given real symmetric positive definite (SPD) matrix $\mathbf{A}$ and $\mathbf{B}$, there exists a set of real numbers λ with vectors $\mathbf{x}$ satisfying $\mathbf{A}\mathbf{x}=\lambda\mathbf{B}\mathbf{x}$, where λ and $\mathbf{x}$ are called the eigenvalues and the corresponding eigenvectors. For the SPD generalized eigenvalue problem, all eigenvalues are positive.

For small-scale eigenvalue problems, classical algorithms such as the Jacobi method and the QR decomposition method are wildly used. For larger-scale sparse problems,

*Corresponding author.
Emails: zhengfy1@pku.edu.cn (F. Zheng), luojianyang@pku.edu.cn (J. Luo), sunsl@mech.pku.edu.cn (S. Sun)

these algorithms are unsatisfactory in computational efficiency, because their time complexity is related to the cube of the dimension of matrices. Lanczos [2] and Arnoldi [3] gave efficient methods for large-scale symmetric and general eigenvalue problems, respectively. Since then many effective subspace projection methods have been developed [4–7]. Yuan proposed the WYD method [8] as an eigenvalue solver which is particularly effective for structural dynamic analysis problems, and showed the relationship between the WYD method and the Lanczos method.

The basic process of the subspace projection method is to form a search space firstly, then project the large-scale matrix to this subspace, and transform the original problem into a small-scale eigenvalue problem. Based on the eigenvalues and eigenvectors of this small-scale problem, a set of corresponding approximate eigenvalues and approximate eigenvectors of the original problem can be obtained in the search space [9], this process is often called the Rayleigh-Ritz process [10]. The subspace iterative method combines the Rayleigh-Ritz process with the vector simultaneous iterations to achieve much faster convergence for multiple eigenvalues and eigenvectors [11]. Earlier, methods for generating the search spaces were usually based on exact solvers. However, some more efficient eigenvalue solvers have since emerged, which use inexact linear solvers to generate the search spaces while still allowing the eigenvectors to converge. The most representative of such eigenvalue solvers include the Jacobi-Davidson method (JD) [12, 13], the locally optimal block preconditioned conjugate gradient (LOBPCG) method [14], etc.

We introduce the inexact solver and iterative strategy into the WYD method and propose the inexact iterative WYD (IIWYD) method, which can efficiently solve large-scale SPD eigenvalue problems. In Section 2, the original WYD method is introduced. Then the complete IIWYD method is constructed and analyzed in Section 3. In Section 4, numerical experiments are conducted to demonstrate the convergence and computational efficiency of the IIWYD method. In Section 5, we give brief conclusions on the IIWYD method and discussions on the relationship between different acceleration strategies.

2 Original WYD method and optimization strategies

The WYD method [8] is based on the load-dependent Ritz vector (LDRV) method [1, 15], which is proposed to be used for dynamic analysis originally. Yuan proposed that the Ritz vectors can be regarded as the approximation to the eigenvectors when the random vector is taken as the initial vector. When the number of Ritz vectors increases, the error in approximate eigenvectors will also decrease. Yuan also proved that, for the N-order eigenvalue problem, the projection matrix $\mathbf{A}^*$ of the WYD method and coefficient matrix $\mathbf{T}$ of the Lanczos method exactly satisfy $\mathbf{A}^* = \mathbf{T}^{-1}$ when both the WYD and Lanczos method perform N steps. Since the Lanczos vectors depend only on the previous two terms, and the Ritz vectors of the WYD method remain orthogonal, usually the Ritz vectors of the WYD method have better numerical stability and clearer physical meaning [8].

Chen [16] gives a more practical iterative form of the WYD method, which reduces the

dimensions of the search space that is required to be expanded. Meanwhile, the added iterations also improve the accuracy of the eigenvalues and eigenvectors greatly. For the sake of discussion, we give here the WYD method in the iterative form, see Algorithm 2.1.

Algorithm 2.1 Iterated WYD method.

Input: $\mathbf{A},\mathbf{B}\in\mathbb{R}^{N\times N}$ be SPD, needed number of smallest eigenvalues n_{ev}, maximum order of Ritz vectors n_r, initial $\mathbf{B}$-unit vectors $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3,\cdots,\mathbf{x}_{n_{ev}}]$

repeat

for i = 1 to n_{ev} **do**

$\mathbf{r}_{i0}=\mathbf{x}_i$

for j = 1 to n_r **do**

Solve the iterative equations as follows

$$\mathbf{A}\mathbf{r}'_{ij}=\mathbf{B}\mathbf{r}_{i(j-1)} \tag{2.1}$$

$\mathbf{B}$-orthogonalize $\mathbf{r}'_{ij}$ to all the previous Ritz vectors, then normalize it to get $\mathbf{r}_{ij}$

end for

Let $\mathbf{R}_i=\{\mathbf{r}_{i1},\mathbf{r}_{i2},\mathbf{r}_{i3},\cdots,\mathbf{r}_{in_r}\}$

end for

Perform Rayleigh-Ritz process with $[\mathbf{R}_1,\mathbf{R}_2,\cdots,\mathbf{R}_{n_{ev}}]$, get n_{ev} new approximate eigenvectors $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3,\cdots,\mathbf{x}_{n_{ev}}]$ with n_{ev} smallest approximate eigenvalues $\mathbf{\Lambda}=\mathrm{diag}\{\lambda_1,\lambda_2,\lambda_3,\cdots,\lambda_{n_{ev}}\}$

until the error of $\mathbf{AX}=\mathbf{BX\Lambda}$ is small enough

Output: n_{ev} smallest eigenvalues $\mathbf{\Lambda}=\mathrm{diag}\{\lambda_1,\lambda_2,\lambda_3,\cdots,\lambda_{n_{ev}}\}$ with eigenvectors $\mathbf{X}=\{\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3,\cdots,\mathbf{x}_{n_{ev}}\}$

When applied to dynamic analysis problems, $\mathbf{x}$ can be taken as the equivalent distribution of external and inertial forces, so $\mathbf{R}$ is called the "load-dependent Ritz vector".

Since (2.1) needs to be solved repeatedly, it is usually possible to perform a cholesky decomposition on $\mathbf{A}$ to accelerate. However, the time-space costs of matrix decomposition are potentially too large for extremely large-scale problems, and distributed computing does not accelerate matrix decomposition as well as the iterative solvers. Therefore it is still necessary to further optimize the WYD method with iterative solvers. There are two approaches to overcome this dilemma.

One of these is to solve (2.1) with an inexact linear solver to reduce the computational cost of each iteration, which is mentioned in several methods such as the JD method [12] and the LOBPCG method [14, 17]. The other is to introduce additional search spaces to maintain the stability of the algorithm and achieve accelerated convergence. We next describe these two methods in detail, in particular the impact of inexact linear solvers.

3 Inexact Iterative WYD method

3.1 Introducing inexact Krylov subspace solvers into the WYD method and its convergence analysis

In the WYD method, the orthogonalization is processed instantly alongside the computation of the Ritz vectors, but the orthogonalization process can also be merged and executed after the inner loop, which would not change the search space. To illustrate this, let $\mathbf{V}=\text{span}\{\mathbf{q}_1,\mathbf{q}_2,\mathbf{q}_3,\cdots,\mathbf{q}_n\}$, in which $\mathbf{q}_j$ is generated by (2.1) from a given $\mathbf{q}_0$. From Algorithm 2.1 the following relationship can be found

$$\mathbf{q}_j'=\mathbf{A}^{-1}\mathbf{B}\mathbf{q}_{j-1}, \tag{3.1a}$$

$$\mathbf{q}_j=\frac{1}{\eta_j}\Big(\mathbf{q}_j'-\sum_{k=1}^{j-1}\theta_{jk}\mathbf{q}_k\Big), \tag{3.1b}$$

in which θ_{jk} are orthogonalization factors and η_j are normalization factors. From (3.1), $\mathbf{q}_j'$ can also be expressed in reverse by $\mathbf{q}_j$ as

$$\begin{aligned}\mathbf{q}_j'&=\sum_{k=1}^{j-1}\theta_{jk}\mathbf{q}_k+\eta_j\mathbf{q}_j\\&=[\mathbf{q}_1,\mathbf{q}_2,\mathbf{q}_3,\cdots,\mathbf{q}_n]\mathbf{w}_j,\end{aligned} \tag{3.2}$$

in which $\mathbf{w}_j=[\theta_{j1},\theta_{j2},\cdots,\theta_{j(j-1)},\eta_j,0,0,\cdots,0]^\top$.

Assume that there is a set of vectors $[\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_3,\cdots,\mathbf{p}_n]$ generated by

$$\mathbf{p}_j=\lambda\mathbf{A}^{-1}\mathbf{B}\mathbf{p}_{j-1} \tag{3.3}$$

with the initial vector $\mathbf{p}_0=\mathbf{q}_0$, λ is a constant number. Next we prove that if $\mathbf{p}_j$ can be expressed as a linear combination of $[\mathbf{q}_1,\mathbf{q}_2,\mathbf{q}_3,\cdots,\mathbf{q}_j]$, or $\mathbf{p}_j=[\mathbf{q}_1,\mathbf{q}_2,\mathbf{q}_3,\cdots,\mathbf{q}_n]\mathbf{v}_j$, in which $\mathbf{v}_j=[v_{j1},v_{j2},\cdots,v_{jj},0,0,\cdots,0]^\top$, then $\mathbf{p}_{j+1}$ can also be expressed as

$$\mathbf{p}_{j+1}=[\mathbf{q}_1,\mathbf{q}_2,\mathbf{q}_3,\cdots,\mathbf{q}_n]\mathbf{v}_{j+1}.$$

From (3.2) and (3.3) we can get

$$\begin{aligned}\mathbf{p}_{j+1}&=\lambda\mathbf{A}^{-1}\mathbf{B}\mathbf{p}_j\\&=\lambda\mathbf{A}^{-1}\mathbf{B}[\mathbf{q}_1,\mathbf{q}_2,\mathbf{q}_3,\cdots,\mathbf{q}_j]\mathbf{v}_j\\&=\lambda[\mathbf{A}^{-1}\mathbf{B}\mathbf{q}_1,\mathbf{A}^{-1}\mathbf{B}\mathbf{q}_2,\mathbf{A}^{-1}\mathbf{B}\mathbf{q}_3,\cdots,\mathbf{A}^{-1}\mathbf{B}\mathbf{q}_j]\mathbf{v}_j\\&=\lambda[\mathbf{q}_2',\mathbf{q}_3',\mathbf{q}_4',\cdots,\mathbf{q}_{j+1}']\mathbf{v}_j\\&=\lambda[\mathbf{q}_1,\mathbf{q}_2,\mathbf{q}_3,\cdots,\mathbf{q}_n][\mathbf{w}_2,\mathbf{w}_3,\cdots,\mathbf{w}_{j+1}]\mathbf{v}_j\\&=[\mathbf{q}_1,\mathbf{q}_2,\mathbf{q}_3,\cdots,\mathbf{q}_n]\mathbf{v}_{j+1}.\end{aligned} \tag{3.4}$$

Since $\mathbf{p}_0=\mathbf{q}_0$, $\mathbf{p}_1=\eta_1\mathbf{q}_1$, (3.4) holds for any j. So there must be

$$\mathbf{V}=\text{span}\{\mathbf{q}_1,\mathbf{q}_2,\mathbf{q}_3,\cdots,\mathbf{q}_n\}=\text{span}\{\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_3,\cdots,\mathbf{p}_n\},$$

in other words, the orthogonalization process can executed after the inner loop, which would not change the search space. Therefore, in the following analysis, for the sake of simplicity, we can change (2.1) to an equivalent form

$$\mathbf{A}\mathbf{r}_k=\lambda\mathbf{B}\mathbf{r}_{k-1}. \tag{3.5}$$

The search space will not change after the subsequent orthogonalization, so the Rayleigh-Ritz process will also output the same result. Then we try to apply an inexact solver to (3.5).

Among the various types of inexact solvers, the Krylov subspace iterative methods (e.g., CG, MINRES, GMRES, etc.) with a limited number of iterations are popular choices for many inexact eigenvalue solvers [12, 17]. If we take an appropriate initial value for $\mathbf{r}_{k+1}$, we can obtain a new vector that is sufficient to improve the eigenvector in only a few iterations. However, to the authors' knowledge, the reliability of this approach has not been thoroughly analyzed. Here we give a convergence condition based on an error decomposition and estimation approach. Using the properties of the residuals of Krylov subspace methods, we can derive the conditions that such an inexact solver should satisfy to give a sufficiently good output.

For N-order symmetric positive definite matrices $\mathbf{A}$ and $\mathbf{B}$, there are N positive real numbers $\{\lambda_1,\lambda_2,\cdots,\lambda_N\}$ and $\mathbf{B}$-unit vectors $\{\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_N\}$, such that $\mathbf{A}\mathbf{x}_i=\lambda_i\mathbf{B}\mathbf{x}_i$, and $\mathbf{x}_i^\top\mathbf{B}\mathbf{x}_j=\delta_{ij}$, where δ_{ij} is the Kronecker symbol.

Then for the N-dimensional $\mathbf{B}$-unit vector $\mathbf{r}_k$, it can always be uniquely represented as

$$\mathbf{r}_k=\sum_{i=1}^{N}\alpha_i\mathbf{x}_i \tag{3.6}$$

and

$$\|\mathbf{r}_k\|_{\mathbf{B}}=\sqrt{\mathbf{r}_k^\top\mathbf{B}\mathbf{r}_k}=\sqrt{\sum_{i=1}^{N}\alpha_i^2}=1. \tag{3.7}$$

For simplicity, we firstly consider the eigenvector corresponding to the smallest eigenvalue λ_1. Although the exact eigenvalues are not known until the computation is over, however, since the eigenvalues converge much faster than the eigenvectors in the WYD method, we can actually assume that we have obtained eigenvalues with sufficient accuracy when analyzing the convergence of the eigenvectors. The feasibility of this assumption is also reflected in the numerical experiments. When solving equation

$$\mathbf{A}\mathbf{r}_{k+1}=\lambda_1\mathbf{B}\mathbf{r}_k,$$

using Krylov subspace solvers with initial value $\mathbf{r}_{k+1}^{(0)} \Leftarrow \mathbf{r}_k$ and performing m steps, we have

$$\mathbf{r}_{k+1} = \lambda_1 \mathbf{A}^{-1}\mathbf{B}\mathbf{r}_k + \mathbf{e}_m, \tag{3.8}$$

where the error vector $\mathbf{e}_m$ can be decomposed by

$$\mathbf{e}_m = \sum_{i=1}^{N} \beta_i \mathbf{x}_i. \tag{3.9}$$

Substituting (3.6) and (3.9) into (3.8), we get

$$\begin{aligned}\mathbf{r}_{k+1} =& \lambda_1 \mathbf{A}^{-1}\mathbf{B}\Big(\sum_{i=1}^{N} \alpha_i \mathbf{x}_i\Big) + \sum_{i=1}^{N} \beta_i \mathbf{x_i} \\ =& (\alpha_1+\beta_1)\mathbf{x}_1 + \sum_{i=2}^{N}\Big(\alpha_i \frac{\lambda_1}{\lambda_i} + \beta_i\Big)\mathbf{x}_i.\end{aligned} \tag{3.10}$$

Let

$$\langle \mathbf{a}, \mathbf{b} \rangle = \arccos \frac{\mathbf{a}^\top \mathbf{B}\mathbf{b}}{\|\mathbf{a}\|_{\mathbf{B}} \|\mathbf{b}\|_{\mathbf{B}}}$$

be the angle between $\mathbf{a}$ and $\mathbf{b}$. Since both $\mathbf{x}_1$ and $-\mathbf{x}_1$ are eigenvectors of $\mathbf{A}$ corresponding to λ_1, it can be assumed that $\langle \mathbf{r}_k, \mathbf{x}_1 \rangle < \pi/2$ (if not, just replace $\mathbf{x}_1$ with $-\mathbf{x}_1$). Thus

$$\langle \mathbf{r}_k, \mathbf{x}_1 \rangle = \arccos \frac{\mathbf{r}_k^\top \mathbf{B}\mathbf{x}_1}{\|\mathbf{r}_k\|_{\mathbf{B}} \|\mathbf{x}_1\|_{\mathbf{B}}} = \arcsin \sqrt{1 - \left(\frac{\mathbf{r}_k^\top \mathbf{B}\mathbf{x}_1}{\|\mathbf{r}_k\|_{\mathbf{B}} \|\mathbf{x}_1\|_{\mathbf{B}}} \right)^2}.$$

If $\mathbf{r}_{k+1}$ is expected to be closer to the eigenvector $\mathbf{x}_1$ than $\mathbf{r}_k$, which is expressed by the inequality

$$\langle \mathbf{r}_{k+1}, \mathbf{x}_1 \rangle < \langle \mathbf{r}_k, \mathbf{x}_1 \rangle, \tag{3.11}$$

substituting (3.6) and (3.10) into (3.11), then

$$1 - \frac{(\alpha_1+\beta_1)^2}{\sum_{i=1}^{N}(\alpha_i \frac{\lambda_1}{\lambda_i} + \beta_i)^2} < 1 - \frac{\alpha_1^2}{\sum_{i=1}^{N} \alpha_i^2} = 1 - \alpha_1^2. \tag{3.12}$$

Furthermore, if the sequence $\{\mathbf{r}_k\}$ is expected to converge strictly to $\mathbf{x}_1$, which means

$$\lim_{k \to +\infty} \langle \mathbf{r}_k, \mathbf{x}_1 \rangle = 0, \tag{3.13}$$

then we can try to find a constant $0 < \gamma < 1$ such that

$$1 - \frac{(\alpha_1+\beta_1)^2}{\sum_{i=1}^{N}(\alpha_i \frac{\lambda_1}{\lambda_i} + \beta_i)^2} < \gamma(1 - \alpha_1^2). \tag{3.14}$$

When Krylov subspace method is applied to solve (3.5) and $\mathbf{B}$ is taken as the preconditioner, the initial residual vector is

$$\boldsymbol{\xi}^{(0)}=\lambda_1\mathbf{r}_k-\mathbf{B}^{-1}\mathbf{A}\mathbf{r}_k=\sum_{i=1}^{N}\alpha_i(\lambda_1-\lambda_i)\mathbf{x}_i,$$

therefore $\mathbf{x}_1^\top\mathbf{B}\boldsymbol{\xi}^{(0)}=0$. Thus the Krylov subspace

$$U=\text{span}\{\boldsymbol{\xi}^{(0)},\mathbf{B}^{-1}\mathbf{A}\boldsymbol{\xi}^{(0)},(\mathbf{B}^{-1}\mathbf{A})^2\boldsymbol{\xi}^{(0)},\cdots\}$$

will always be perpendicular to $\mathbf{x}_1$ during the calculation, so we have $\beta_1=0$ in (3.14). Then (3.14) is equivalent to

$$\sum_{i=1}^{N}\left(\alpha_i\frac{\lambda_1}{\lambda_i}+\beta_i\right)^2<\frac{\alpha_1^2}{1-\gamma(1-\alpha_1^2)}=\Omega(\gamma,\alpha_1), \tag{3.15}$$

while

$$\begin{aligned}\sum_{i=1}^{N}\left(\alpha_i\frac{\lambda_1}{\lambda_i}+\beta_i\right)^2&=\sum_{i=1}^{N}\left(\alpha_i\frac{\lambda_1}{\lambda_i}\right)^2+\sum_{i=1}^{N}\beta_i^2+2\sum_{i=1}^{N}\alpha_i\beta_i\frac{\lambda_1}{\lambda_i}\\&\le\sum_{i=1}^{N}\left(\alpha_i\frac{\lambda_1}{\lambda_i}\right)^2+\sum_{i=1}^{N}\beta_i^2+2\sum_{i=1}^{N}|\alpha_i\beta_i|\frac{\lambda_1}{\lambda_i}\\&\le\sum_{i=1}^{N}\left(\alpha_i\frac{\lambda_1}{\lambda_i}\right)^2+\sum_{i=1}^{N}\beta_i^2+2\Big[\sum_{i=1}^{N}\left(\alpha_i\frac{\lambda_1}{\lambda_i}\right)^2\sum_{i=1}^{N}\beta_i^2\Big]^{\frac{1}{2}}.\end{aligned} \tag{3.16}$$

Let

$$C(\mathbf{r}_k)=C(\alpha_1,\alpha_2,\cdots,\alpha_N)=\sum_{i=1}^{N}\left(\alpha_i\frac{\lambda_1}{\lambda_i}\right)^2\quad\text{and}\quad\beta=\sqrt{\sum_{i=1}^{N}\beta_i^2}=\|\mathbf{e}_m\|_{\mathbf{B}}$$

substitute into (3.16), we get

$$\sum_{i=1}^{N}\left(\alpha_i\frac{\lambda_1}{\lambda_i}+\beta_i\right)^2\le\beta^2+2\sqrt{C}\beta+C. \tag{3.17}$$

Therefore, as long as

$$\beta^2+2\sqrt{C}\beta+C<\Omega(\gamma,\alpha_1) \tag{3.18}$$

has positive real solutions, (3.15) can hold. Thus there must be $C(\mathbf{r}_k)<\Omega(\gamma,\alpha_1)$ constantly.

For the convenience of the analysis, we provisionally disregard the case of multiple eigenvalues and assume that $\lambda_1<\lambda_2$. From (3.6) we know that $0<C(\mathbf{r}_k)\le C(\mathbf{x}_1)=1$, the equality sign holds if and only if $\alpha_1=1$ and $\alpha_i=0$ for all $2\le i\le N$, which means $\mathbf{r}_k=\mathbf{x}_1$.

Since $\mathbf{r}_k$ belongs to the $(N-1)$-dimensional curved surface $\mathcal{S}$ composed of all the $\mathbf{B}$-unit vectors in N-dimensional space and $C(\alpha_1,\alpha_2,\cdots,\alpha_N)$ is a positive definite quadratic

form, then $\forall 0<\epsilon<1$, $\exists$ the closed domain $\mathcal{D}\subseteq\mathcal{S}$, $\forall\mathbf{x}\in\mathcal{D}$, $1-\epsilon\leq C(\mathbf{x})\leq 1$, and $\forall\mathbf{x}\in\complement_{\mathcal{S}}\mathcal{D}$, $0<C(\mathbf{x})<1-\epsilon$. If $\mathbf{r}_k\in\mathcal{D}$, because ϵ can be arbitrarily small, it is always possible to take a proper ϵ such that $\mathbf{r}_k$ satisfies the termination condition of the algorithm. Therefore we focus on the case where $\mathbf{r}_k\in\complement_{\mathcal{S}}\mathcal{D}$. In this case, since $0<C(\mathbf{r}_k)<1-\epsilon$, there is $C(\mathbf{r}_k)<\Omega(\gamma,\alpha_1)$ as long as $\Omega(\gamma,\alpha_1)\geq 1-\epsilon$ always holds. Because $\Omega(\gamma,\alpha_1)=\alpha_1^2/(1-\gamma(1-\alpha_1^2))$ decreases as γ increases, we should take

$$\gamma=\sup_{\alpha_1}\Big\{\frac{1}{1-\alpha_1^2}\Big(1-\frac{\alpha_1^2}{1-\epsilon}\Big)\Big\}. \tag{3.19}$$

If and only if $\alpha_1=0$, which means $\mathbf{r}_k$ does not contain any component in the direction of $\mathbf{x}_1$, then γ can only be 1, there is no legitimate γ that satisfies $0<\gamma<1$. However, if the initial vector is a random vector, then α_1 can hardly be 0, so practically there is always a constant γ such that both $0<\gamma<1$ and $C(\mathbf{r}_k)<\Omega(\gamma,\alpha_1)<1$ hold constantly.

When there exists a legitimate γ, the condition that makes (3.18) hold is

$$\beta<\sqrt{\Omega}-\sqrt{C}. \tag{3.20}$$

In most Krylov subspace methods, the modulus of the residual vector is theoretically upper bounded. For example, the modulus of the residual vector at the m-th step of the CG method satisfies

$$\|\mathbf{e}_m\|\leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^m\|(\lambda_1\mathbf{B}-\mathbf{A})\mathbf{r}_k\|,$$

in which $\kappa=\lambda_N/\lambda_1$ is the spectral condition number [18]. So there must exist m such that (3.20) holds. Thus Theorem 3.1 can be obtained.

Theorem 3.1. *If the initial vector $\mathbf{r}_0$ is a random vector, the system of linear equations $\mathbf{A}\mathbf{r}_{k+1}=\lambda_1\mathbf{B}\mathbf{r}_k$ is solved by Krylov subspace methods with an initial value $\mathbf{r}_{k+1}^{(0)}\Leftarrow\mathbf{r}_k$, and there are no multiple eigenvalues, as long as the modulus β of the residual vector satisfies $\beta<\sqrt{\Omega}-\sqrt{C}$ after m steps, in which*

$$C=\sum_{i=1}^{N}(\alpha_i\lambda_1/\lambda_i)^2,\quad \Omega(\gamma,\alpha_1)=\alpha_1^2/(1-\gamma(1-\alpha_1^2)),$$

γ is a constant related to the convergence requirement and the initial vector $\mathbf{r}_0$, then

$$\lim_{k\to+\infty}\langle\mathbf{r}_k,\mathbf{x}_1\rangle=0$$

holds inevitably.

The convergence conditions of the other eigenvectors can be obtained similarly on the space $\mathbf{V}/\mathbf{x}_1$ in turn, without further elaboration. This theorem gives a relatively strict convergence condition for all eigenvalue algorithms using inexact Krylov subspace solvers.

3.2 Optimization of the search space

By taking an appropriate number of steps for the linear solver in the WYD method, we can find a sequence that converges to the eigenvector through the search space span$\{\mathbf{R}\}$. However, since the convergence condition given by Theorem 3.1 is related to both the approximate eigenvectors $\mathbf{X}$ and the spectral condition number of the matrix $\mathbf{A}$, it is usually not easy to find a constant m that always makes this condition satisfied. Moreover, although the convergence of the approximate eigenvalues is much faster than that of the eigenvectors, the approximate eigenvalues themselves are still subject to errors. A simpler way to formally maintain the convergence of the eigenvalue sequence is to include the current approximate eigenvector $\mathbf{X}$ into the search space, which ensures that the eigenvalue sequence is monotonically non-increasing.

To further speed up the convergence, we can continue to expand the search space. Jia proposed the refined vectors $\mathbf{F}$, which is obtained by solving the following small-scale eigenvalue problem [19]

$$\mathbf{V}^\top(\mathbf{A}-\lambda_i\mathbf{B})^\top(\mathbf{A}-\lambda_i\mathbf{B})\mathbf{V}\mathbf{f}=\sigma\mathbf{f}, \tag{3.21a}$$

$$\mathbf{F}=\mathbf{V}\mathbf{f}, \tag{3.21b}$$

in which λ_i is the approximate eigenvalue of the original problem $\mathbf{A}\mathbf{x}=\lambda\mathbf{B}\mathbf{x}$, $\mathbf{f}$ is the eigenvector corresponding to the smallest eigenvalue in the small-scale eigenvalue problem. The refined vectors minimize the absolute errors

$$e_a=\|\mathbf{A}\mathbf{x}-\lambda_i\mathbf{B}\mathbf{x}\|$$

in the search space $\mathbf{V}$, so the refined vectors can be used as a decent approximation to the eigenvectors to expand the search space of the next iteration and named as the refined strategy.

The LOBPCG method provides a different thought [14]. The LOBPCG method uses a linear combination of the current approximate eigenvectors and the approximate eigenvectors of the previous iteration, i.e., the conjugate vectors $\mathbf{P}$, to expand the search space. Therefore the search space of the LOBPCG method can be expressed as

$$\mathbf{V}=\text{span}\{\mathbf{X}_i,\mathbf{P},\mathbf{W}\}=\text{span}\{\mathbf{X}_i,\mathbf{X}_{i-1},\mathbf{W}\}, \tag{3.22}$$

in which $\mathbf{X}_i$ is composed of the approximate eigenvectors obtained in the i-th iteration and $\mathbf{W}$ is given by the following equation

$$\mathbf{W}=\mathbf{T}(\mathbf{A}\mathbf{X}_i-\mathbf{B}\mathbf{X}_i\mathbf{\Lambda}), \tag{3.23}$$

where $\mathbf{T}$ is a constant preconditioner or an inexact linear solver and $\mathbf{\Lambda}$ is a diagonal matrix composed of the approximate eigenvalues. If the set of conjugate vectors $\mathbf{P}$ is considered as an acceleration strategy as well, it can also be used to expand the search space of the WYD method and named as the conjugate strategy.

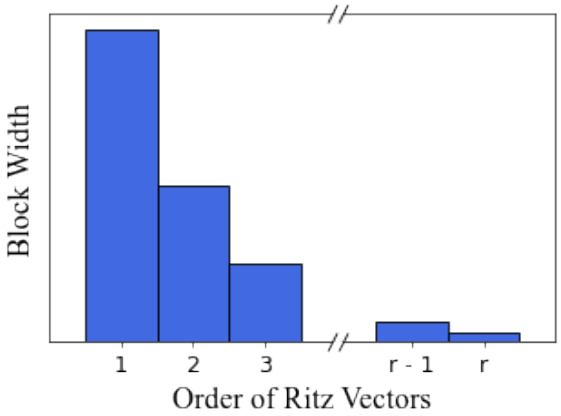


Figure 1: Variation of the block width with the order of Ritz vectors.

The above discussion focuses on the calculation of a single eigenvalue, and it can be simply extended as a block method like most eigenvalue solvers [4, 20].

Usually the convergence of higher-order eigenvalues and eigenvectors is generally slower than that of lower-order eigenvalues and corresponding eigenvectors [21], which means that the relative errors of higher-order eigenvalues and eigenvectors are usually larger. The errors will accumulate in the calculation process of Ritz vectors, resulting in a decrease in the effectiveness of higher-order Ritz vectors. To reduce the cost of the invalid calculation, we propose a block shrinkage strategy for optimization, i.e., as the order i of Ritz vectors increases, the width of each block of Ritz vectors $\mathbf{R}_i$ is reduced to a fixed percentage s. For example, if the block size used in the calculation is 16, the maximum order of the Ritz vectors is 4, and the shrinkage rate is 0.5, then the number of Ritz vectors of order 1 to 4 is 16, 8, 4, 2 respectively. This strategy can also be expressed visually in Fig. 1. The block shrinkage strategy reduces the calculation of the parts with larger errors, and also prevents the total dimension of the Ritz vectors from growing linearly with the maximum order of Ritz vectors, but instead, there is always a well-defined upper bound.

3.3 Complete algorithm for the IIWYD method

The above optimization strategies are combined to obtain the complete algorithm of the IIWYD method, see Algorithm 3.1.

4 Numerical experiments and results

We implemented the IIWYD method based on the PETSc [22, 23] and SLEPc [24, 25] library and compared the computational efficiency with the LOBPCG method and the JD

Algorithm 3.1 Inexact iterated WYD method.

Input: $\mathbf{A},\mathbf{B}\in\mathbb{R}^{N\times N}$ be SPD, needed number of smallest eigenvalues n_{ev}, maximum order of Ritz vectors n_r, initial $\mathbf{B}$-unit vectors $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3,\cdots,\mathbf{x}_{n_{ev}}]$, vectors $\mathbf{F}$ generated by a certain acceleration strategy, shrinking rate s, maximum steps m for the Krylov subspace linear solver

repeat

for i = 1 to n_{ev} **do**

$\mathbf{r}_{i0}=\mathbf{x}_i$

for j=1 to $n_{ri}=\min\{\lceil\log_s(\frac{i}{n_{ev}})\rceil,n_r\}$ **do**

Solve for the current inexact Ritz vector (at most m steps)

$$\mathbf{A}\mathbf{r}_{ij}=\lambda_i\mathbf{B}\mathbf{r}_{i(j-1)}$$

end for

Let $\mathbf{R}_i=\{\mathbf{r}_{i1},\mathbf{r}_{i2},\mathbf{r}_{i3},\cdots,\mathbf{r}_{in_{ri}}\}$

end for

Compute $\mathbf{F}$ according to the selected acceleration strategy

Construct the search space $\mathbf{V}\Leftarrow\text{span}\{\mathbf{R}_1,\mathbf{R}_2,\cdots,\mathbf{R}_{n_{ev}},\mathbf{X},\mathbf{F}\}$

Perform Rayleigh-Ritz process with $\mathbf{V}$, get n_{ev} new approximate eigenvectors $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3,\cdots,\mathbf{x}_{n_{ev}}]$ with n_{ev} smallest approximate eigenvalues $\mathbf{\Lambda}=\text{diag}\{\lambda_1,\lambda_2,\lambda_3,\cdots,\lambda_{n_{ev}}\}$

until the error of $\mathbf{AX}=\mathbf{BX\Lambda}$ is small enough

Output: n_{ev} smallest eigenvalues $\mathbf{\Lambda}=\text{diag}\{\lambda_1,\lambda_2,\lambda_3,\cdots,\lambda_{n_{ev}}\}$ with eigenvectors $\mathbf{X}=\{\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3,\cdots,\mathbf{x}_{n_{ev}}\}$

method provided by SLEPc. In the JD method, we use the GMRES method to solve the correction equations [12]. In the LOBPCG method, Knyazev [17] suggests using the CG method with the BoomerAMG [26] preconditioner as the solver, and in this case, we can execute only one CG step to get a good enough search space, so we can quit the CG process after the BoomerAMG preconditioner to save more computational cost. However, this approach is poor in stability in some cases. So for each test case, we preferably try to use BoomerAMG as the preconditioner and then for cases where the BoomerAMG preconditioner does not produce good results, we use the simpler Jacobi (Diagonal) preconditioner, and set the maximum number of steps of the CG method to 60. In all tests for each case, the preconditioners of the IIWYD method and the LOBPCG method are always the same. In the IIWYD method, we take the depth of the Ritz vectors $n_r=3$ and the shrinking rate $s=0.5$. For the acceleration strategy, we implement two schemes, the refined strategy and the conjugate strategy, which are both mentioned in Section 3.2. All other computational parameters are appropriately tuned by trial calculations for smaller-scale problems. The convergence boundaries of all eigenvectors are taken as relative

Table 1: Information about the matrices used for the numerical experiments.

Matrix Name	Order/Rank(N)	Nonzeros(nnz)	λ_1	λ_{100}
apache1	80,800	542,184	0.090919	25.842
apache2	715,176	4,817,870	0.029949	8.9220
bmwcra_1	148,770	10,641,602	0.087225	7976.0
crankseg_1	52,804	10,614,210	997.03	40813.
crankseg_2	63,838	14,148,858	1015.4	37351.
hood	220,542	9,895,422	1.0000	1.0000
StocF-1465	1,465,137	21,005,389	0.025781	0.41085
Hook_1498	1,498,023	59,374,451	0.076955	12.425
ct20stif	52,329	2,600,295	-[1]	-[1]
s3dkq4m2	90,449	4,427,725	-[1]	-[1]
x104	108,384	8,713,602	-[1]	-[1]

[1] All solvers did not produce converged results within a limited time.

errors, which satisfy

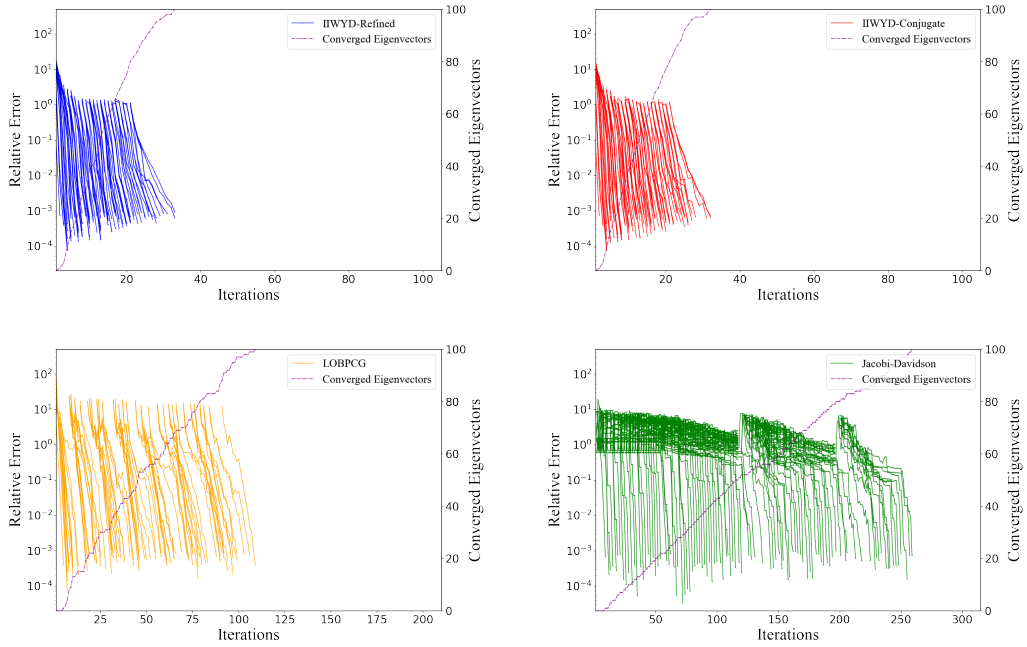
$$e_r = \frac{\|\mathbf{A}\mathbf{x}_i - \lambda_i \mathbf{B}\mathbf{x}_i\|}{\|\mathbf{A}\mathbf{x}_i\|} \approx \frac{\|\mathbf{A}\mathbf{x}_i - \lambda_i \mathbf{B}\mathbf{x}_i\|}{\lambda_i \|\mathbf{B}\mathbf{x}_i\|} < 1\times 10^{-3}. \tag{4.1}$$

In practice, when e_r satisfies the convergence condition, the relative error of the eigenvalue has usually narrowed to less than 1×10^{-6}, so that this condition is already an acceptable bound. In addition, since the approximate eigenvalues appear in the practical form of (4.1), there are unexpected variations in e_r when the eigenvalues are far from converging. However, as the eigenvalues converge, the results of e_r will be sufficiently accurate.

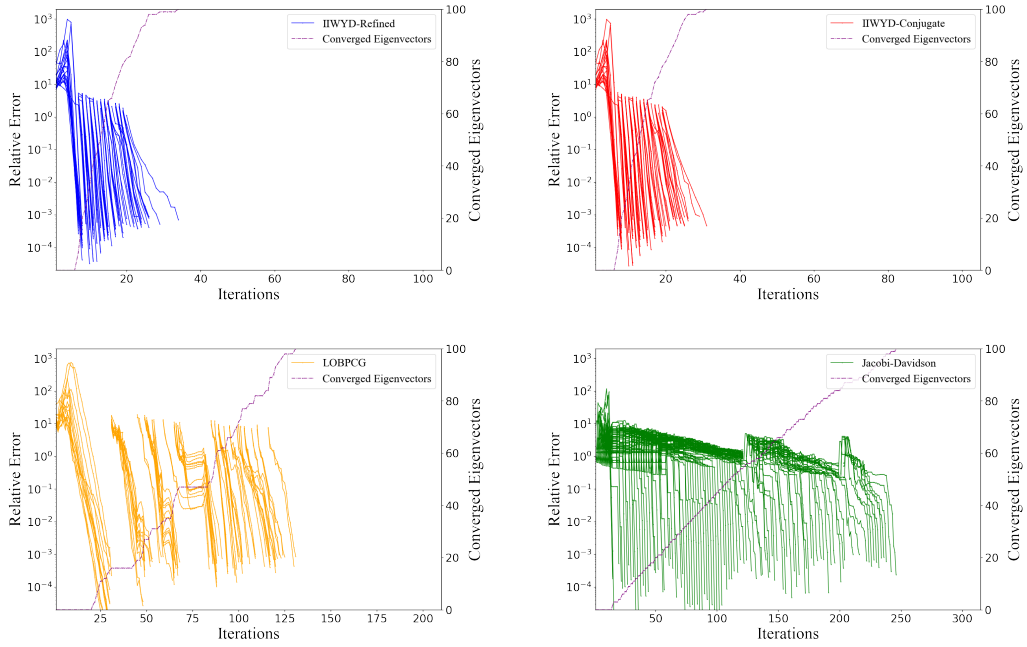
The matrices $\mathbf{A}$ used in the experiments are obtained from the SuiteSparse Matrix Collection [27] and are all real symmetric positive definite matrices, the basic information of these matrices is shown in Table 1. Also for simplicity, we take $\mathbf{B} = \mathbf{I}$ for all cases.

All algorithms solve for the first 100 minimum eigenvalues of $\mathbf{A}\mathbf{x} = \lambda \mathbf{B}\mathbf{x}$ and their corresponding eigenvectors. The experiment environment is a personal laptop with an Intel Core i7-9750H CPU (Turbo Boost Technology off), maximum available memory of 16 GB, Ubuntu 16.04 operating system, PETSc and SLEPc version 3.15.1. Table 2 shows the preconditioner (PC, J for Jacobi, A for BoomerAMG) used by the methods to be tested in each test case, the elapsed time (in seconds), and the number of iterations (iters) required. Fig. 2 shows the trend of the relative errors of the eigenvectors in each of the four methods, taking matrices *apache*2 and *bmwcra*_1 as examples.

According to the experiment results, the JD method has relatively better numerical stability compared to the LOBPCG method, but its average computational efficiency is not satisfactory. Since the JD method is based on constructing and solving the correction equations, it is quite different from other methods, so the number of iterations of the JD



(a) apache2



(b) bmwcra_1

Figure 2: Variation of relative errors with the number of iterations.

Table 2: Comparison of IIWYD method with refined strategy and conjugate strategy, LOBPCG method, and Jacobi-Davidson method.

Matrix Name	PC	IIWYD				LOBPCG		Jacobi-Davidson	
		Refined		Conjugate					
		iters	time(s)	iters	time(s)	iters	time(s)	iters	time(s)
apache1	A	34	13.8	32	11.8	74	29.7	339	164.9
apache2	A	34	214.0	33	201.7	74	523.3	263	1207.2
bmwcra_1	J	35	392.1	32	366.5	103	1046.2	228	630.1
crankseg_1	J	50	474.4	43	445.0	23	418.5[1]	487	1271.0
crankseg_2	J	48	690.4	43	657.7	-	-[2]	522	1836.9
hood	J	163	2860.1	135	2474.0	162	2030.3	-	-[2]
StocF-1465	A	75	875.5	59	767.5	-	-[2]	-	>10000
Hook_1498	J	30	2296.5	26	2245.3	104	7754.1	250	4991.5

[1] Terminated unexpectedly after 31 eigenvectors have converged.
[2] Terminated due to fatal errors during the calculations.

method is not comparable with other methods. It can be seen from Fig. 2 that the solving strategy of the JD method converges the eigenvectors near the target one by one, so it has some disadvantages in the computation of a large batch of eigenvectors.

In contrast, the convergence properties of the IIWYD method are more similar to those of the LOBPCG method. Since the IIWYD method accommodates multiple levels of Ritz vectors in one iteration, the search space formed by the Ritz vectors contains more effective information of eigenvectors, which makes the convergence efficiency of the IIWYD method higher than that of the LOBPCG method in each iteration, and the total number of iterations of the IIWYD method is also less. In addition, the expansion of the search space in the IIWYD method also improves the numerical stability of the algorithm.

In addition to the matrices above, we also tested other cases which are more difficult for eigenvalue solving, such as $ct20stif$, $s3dkq4m2$, $x104$, etc. However, all four methods failed to obtain any of the eigenvectors within the time limitation. This indicates that the existing eigenvalue solvers still have significant shortcomings when facing some special cases.

5 Conclusions and discussion

The iterative inexact WYD method is proposed based on the WYD method, which is efficient for solving large-scale symmetric positive definite generalized eigenvalue problems. We extend the original WYD method by introducing the iterative strategy, the inexact Krylov subspace solvers, and additional acceleration strategies. The results of numerical experiments show that the IIWYD method has better computational efficiency and numerical stability compared with several commonly used inexact eigenvalue solvers.

The condition is briefly analyzed, which should be satisfied when inexact Krylov subspace methods are applied in eigenvalue solvers to generate Ritz vectors that converge

to the eigenvectors. This contributes to the improvement of various inexact eigenvalue methods as well as the convergence analysis and the error estimation of these algorithms.

Another important result obtained from the experiments is that the IIWYD method has very similar efficiency when using the refined strategy and the conjugate strategy. If the additional time cost incurred in computing the refined vector is ignored, the performance of the two strategies will be much closer. The refined vectors have explicit algebraic properties, for any approximate eigenvalue λ_i and its corresponding refined vector

$$\mathbf{f}_i \in \mathbf{V}, \quad \forall \mathbf{x} \in \mathbf{V}, \quad \|\mathbf{A}\mathbf{f}_i - \lambda_i \mathbf{B}\mathbf{f}_i\| \leq \|\mathbf{A}\mathbf{x} - \lambda_i \mathbf{B}\mathbf{x}\|.$$

The conjugate vectors are not generated in this way, it can be obtained directly using the coefficients generated by the Rayleigh-Ritz process. This result suggests that there are closer algebraic relationship between the conjugate vectors and the refined vectors, or in other words, the conjugate vectors serve to introduce the residual minimization process numerically. We further conjecture that if an eigenvalue solver takes the results of the Rayleigh-Ritz process as the approximate eigenvectors, then introducing the search space that minimizes the residuals $\|\mathbf{A}\mathbf{x} - \lambda_i \mathbf{B}\mathbf{x}\|$ may obtain a good acceleration; conversely, if we take the refined vectors as the approximate eigenvectors, then introducing the Rayleigh-Ritz process to expand the search space may also obtain a good acceleration. The verification and extension of this conjecture may serve as an idea for further optimization of the eigenvalue solvers in the future.

Acknowledgements

This work is supported by National Key R&D Program of China (No. 2021YFB2401700) and National Natural Science Foundation of China (No. 11672362). With many thanks to Prof. Pu Chen, for his guidance during the research.

References

[1] E. WILSON, M., YUAN AND J. DICKENS, *Dynamic analysis by direct superposition of ritz vectors,* Earthquake Eng. Struct. Dyn., 10 (1982), pp. 813–821.

[2] C. LANCZOS, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators,* J. Res. National Bureau Standards, 45 (1950), pp. 255–282.

[3] W. ARNOLDI, *The principle of minimized iterations in the solution of the matrix eigenvalue problem,* Quart. Appl. Math., 9 (1951), pp. 17–29.

[4] G. GOLUB, AND R. UNDERWOOD, *The block lanczos method for computing eigenvalues,* Math. Software, (1977), pp. 361–377.

[5] D. CALVETTI, L. REICHEL AND D. SORENSEN, *An implicitly restarted lanczos method for large symmetric eigenvalue problems,* Electr. Trans. Numer. Anal., 2 (1994), pp. 1–21.

[6] K. WU, AND H. SIMON, *Thick-restart lanczos method for large symmetric eigenvalue problems,* SIAM J. Matrix Anal. Appl., 22 (2000), pp. 602–616.

[7] Y. SAAD, *Variations on arnoldi's method for computing eigenelements of large unsymmetric matrices,* Linear Algebra and Its Applications, 34 (1980a), pp. 269–295.
[8] M. YUAN, ET AL., *The WYD method in large eigenvalue problems,* Eng. Comput., 6 (1989), pp. 49–57.
[9] M. KADALBAJOO, AND A. GUPTA, *An overview on the eigenvalue computation for matrices,* Neural, Parallel and Scientific Computations, 19 (2011), pp. 129–164.
[10] K. BATHE, AND E. WILSON, Numerical Methods in Finite Element Analysis, Prentice Hall, 1976.
[11] K. BATHE, AND E. WILSON, *Large eigenvalue problems in dynamic analysis,* ASCE J. Eng. Mech. Division, 98 (1972), pp. 1471–1485.
[12] G. SLEIJPEN, AND D. VAN, *A jacobi-davidson iteration method for linear eigenvalue problems,* SIAM J. Matrix Anal. Appl, 17 (1996), pp. 401–425.
[13] X. FAN, ET AL., *Parallel computing study for the large-scale generalized eigenvalue problems in modal analysis,* Science China Physics, Mechanics and Astronomy, 57 (2014), pp. 477–489.
[14] A. KNYAZEV, *Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method,* SIAM J. Sci. Comput., 23 (2001), pp. 517–541.
[15] P. LÉGER, AND E. WILSON, *Generation of load dependent ritz transformation vectors in structural dynamics,* Eng. Comput., 4 (1987), pp. 309–318.
[16] P. CHEN, ET AL., *Advances in solution of classical generalized eigenvalue problem,* Interact. Multiscale Mech., 1 (2008), pp. 211–230.
[17] A. KNYAZEV, ET AL., *Block locally optimal preconditioned eigenvalue xolvers (blopex) in hypre and petsc,* SIAM J. Sci. Comput., 29 (2007), pp. 2224–2239.
[18] Y. SAAD, Iterative Methods for Sparse Linear Systems, SIAM, (2003).
[19] Z. JIA, *Refined iterative algorithms based on arnoldi's process for large unsymmetric eigenproblems,* Linear algebra and Its Applications, 259 (1997), pp. 1–23.
[20] D. BOLEY, AND G. GOLUB, *The lanczos-arnoldi algorithm and controllability,* Systems and Control Letters, 4 (1984), pp. 317–324.
[21] Y. SAAD, *On the rates of convergence of the lanczos and the block-lanczos methods,* SIAM J. Numer. Anal., 17 (1980b), pp. 687–706.
[22] S. BALAY, ET AL., Petsc, (2001), see `http://www.mcs.anl.gov/petsc`.
[23] S. BALAY, ET AL., Petsc Users Manual, (2022), see `https://petsc.org/release/docs/manual`.
[24] V. HERNANDEZ, J. ROMAN, AND V. VIDAL, *Slepc: A scalable and flexible toolkit for the solution of eigenvalue problems,* ACM Trans. Math. Software (TOMS), 31 (2005), pp. 351–362.
[25] J. ROMAN, ET AL., *Slepc users manual,* D. Sistemes Informàtics i Computació, Universitat Politècnica de València, TR DSIC-II/24/02, 3 (2016).
[26] R. FALGOUT, J. JONES, AND U. YANG, *The design and implementation of hypre, a library of parallel high performance preconditioners,* Numerical Solution of Partial Differential Equations on Parallel Computers, (2006), pp. 267–294.
[27] S. KOLODZIEJ, ET AL., *The suitesparse matrix collection website interface,* J. Open Sourre Software, 4 (2019), 1244.