AN INVERSE FREE PRECONDITIONED KRYLOV
SUBSPACE METHOD FOR SYMMETRIC
GENERALIZED EIGENVALUE PROBLEMS

Gene H. Golub * Qiang Ye †

December 14, 2000

Abstract

In this paper, we present an inverse free Krylov subspace method for finding some extreme
eigenvalues of the symmetric definite generalized eigenvalue problem \(Ax = \lambda Bx\). The basic
method takes a form of inner-outer iterations and involves no inversion of \(B\) or any shift-and-
invert matrix \(A - \lambda B\). A convergence analysis is presented that leads to a preconditioning
scheme for accelerating convergence through some equivalent transformations of the eigenvalue
problem. Numerical examples are given to illustrate the convergence properties and to demon-
strate the competitiveness of the method.

1 Introduction

Iterative methods such as the Lanczos algorithm and the Arnoldi algorithm are widely used for
solving large matrix eigenvalue problems (see [21, 22]). Effective applications of these algorithms
typically use a shift-and-invert transformation, which is sometimes called preconditioning [22] and
requires solving a linear system of equations of the original size at each iteration of the process.
For truly large problems, solving the shift-and-invert equations by a direct method such as the LU
factorization is often infeasible or inefficient. In these cases, one can employ an iterative method to
solve them approximately, resulting in two levels of iterations called inner-outer iterations. However,
methods like the Lanczos algorithm and the Arnoldi algorithm are very sensitive to perturbations
in the iterations and therefore require highly accurate solutions of these linear systems (see [8]).
Therefore, the inner-outer iterations may not offer an efficient approach for these methods.

There has recently been great interest in other iterative methods that are also based on shift-
and-invert equations but tolerate low accuracy solutions. One simple example of such methods is the
inexact inverse iteration where the linear convergence property is preserved even when the inversion
is solved to very low accuracy (see [7, 13, 14, 27]). Several more sophisticated and competitive
methods have been developed that also possess such a property. They include the Jacobi-Davidson

---

*Scientific Computing and Computational Mathematics Program, Department of Computer Science, Stanford
University, Stanford, CA 94305. E-mail: golub@scm.stanford.edu. Research supported in part by National Science
Foundation Grant DMS-9403899.

†Department of Mathematics, University of Kentucky, Lexington, KY 40506-0027. E-mail: qye@ms.uky.edu. Part
of this Research was supported by NSERC of Canada while this author was with University of Manitoba.
method [5, 24, 25], truncated RQ iterations [26, 32] and others [14, 29, 30, 31]. One difficulty with these methods is that it is not easy to determine to what accuracy the shift-and-invert equations should be solved. On the other hand, there have been several works that aim at generalizing the concept of preconditioning for linear systems to the eigenvalue problem [1, 2, 4, 11, 12, 17, 18, 15, 20, 28]. This is mostly done, however, by directly adapting a preconditioner used for inverting a certain matrix into an eigenvalue iteration and in these situations, the role of preconditioners is usually not clear, although some of them can be regarded as using inexact shift-and-invert [19]. Overall, while all these new methods have been demonstrated to work successfully in some problems, there is in general a lack of understanding of how and why they work. Furthermore, optimal eigenvalue projection methods such as the Lanczos algorithm have mostly not been incorporated in these developments.

In this paper, we shall present a variation of the Krylov subspace projection methods for computing some extreme eigenvalues of the generalized eigenvalue problem

$$Ax = \lambda Bx, \text{ called the pencil problem for } (A, B),$$

where $A, B$ are symmetric matrices and $B > 0$. The method iteratively improves an approximate eigenpair, each step of which uses either the Lanczos or the Arnoldi iteration to produce a new approximation through the Rayleigh-Ritz projection on a Krylov subspace, resulting in a form of inner-outer iterations. We shall present our theoretical and numerical findings concerning convergence properties of this method and derive bounds on asymptotic linear convergence rates. Furthermore, we shall develop from the convergence analysis some equivalent transformations of the eigenvalue problem to accelerate the convergence, which will be called preconditioning. In particular, such transformations will be based on incomplete factorization and thus generalize the preconditioning for linear systems. To the best of our knowledge, this is the first preconditioning scheme for the eigenvalue problem that is based on and can be justified by a convergence theory.

The paper is organized as follows. We present the basic algorithm in Section 2 and analyze its convergence properties in Section 3. We then give some numerical examples in Section 4 to illustrate the convergence properties. We then present a preconditioned version of the algorithm in Section 5, followed by some numerical examples on the preconditioning in Section 6. We finish with some concluding remarks in Section 7.

## 2 Basic Inverse Free Krylov Subspace Method

In this section, we present our basic algorithm for finding the smallest eigenvalue and a corresponding eigenvector $(\lambda, x)$ of a pencil $(A, B)$ where $A, B$ are symmetric with $B > 0$. We note that the method to be developed can be modified in a trivial way for finding the largest eigenvalue (or simply considering $(-A, B)$).

Given an initial approximation $(\rho_0, x_0)$, we aim at improving it through the Rayleigh-Ritz orthogonal projection on a certain subspace, i.e. by minimizing the Rayleigh quotient

$$\rho(x) \equiv \frac{x^T Ax}{x^T Bx}$$

on that subspace. Noting that the gradient of the Rayleigh quotient at $x_0$ is $\nabla \rho(x_0) = (A - \rho_0 B)x_0 / x_0^T Bx_0$, the well-known steepest descent method chooses a new approximate eigenvector $x_1 \in \text{span}\{x_0, r_0\}$.
by minimizing $\rho(x_1)$. Clearly, this can be considered as the Rayleigh-Ritz projection method on the subspace $K_1 \equiv \text{span}\{x_0, (A - \rho_0 B)x_0\}$. On the other hand, the inverse iteration constructs a new approximation by $x_1 = (A - \rho_0 B)^{-1}Bx_0$. If the inversion is solved inexactly by an iterative solver (i.e. in an inexact inverse iteration [7]), then $x_1$ is indeed chosen from a Krylov subspace as generated by $A - \rho_0 B$. Since $x_1$ is extracted from the Krylov subspace to solve the linear system, it may not be a good choice for approximating the eigenvector.

We consider here a natural extension of these two approaches that finds a new approximate eigenvector $x_1$ from the Krylov subspace

$$K_m \equiv \text{span}\{x_0, (A - \rho_0 B)x_0, \ldots, (A - \rho_0 B)^m x_0\}$$

(for some fixed $m$) by using the Rayleigh-Ritz projection method. The projection is carried out by constructing a basis for $K_m$ and then forming and solving the projection problem for the pencil $(A, B)$. Repeating the process, we arrive at the following iteration, which we call inverse free Krylov method for $(A, B)$.

**Algorithm 1:** Inverse Free Krylov Subspace Method.

Input $m \geq 1$ and an initial approximation $x_0$ with $\|x_0\| = 1$;
\[ \rho_0 = \rho(x_0); \]
For $k = 0, 1, 2, \ldots$ until convergence,
\[ K_m = \text{span}\{x_k, (A - \rho_k B)x_k, \ldots, (A - \rho_k B)^m x_k\} \]
Form $A_m = Z_m^T (A - \rho_k B) Z_m$ and $B_m = Z_m^T B Z_m$;
Find the smallest eigenpair $(\mu_1, v_1)$ for $(A_m, B_m)$;
\[ \rho_{k+1} = \rho_k + \mu_1 \text{ and } x_{k+1} = Z_m v_1. \]
End

In the algorithm, we apply the projection to the shifted pencil $(A - \rho_k B, B)$ and update the approximation accordingly, which is theoretically equivalent to using the projection of $(A, B)$ directly. This formulation, however, may improve the stability while saving matrix-vector multiplications by utilizing $(A - \rho_k B) Z_m$ which need to be computed in the construction of the basis.

In constructing a basis $z_0, z_1, \ldots, z_m$ for $K_m$, there are many possible choices and, theoretically, they are all equivalent in that the new approximate eigenpair $(\rho_{k+1}, x_{k+1})$ obtained will be the same, which is defined by

$$\rho_{k+1} = \frac{x_{k+1}^T A x_{k+1}}{x_{k+1}^T B x_{k+1}} = \min_{0 \neq u \in K_m} \frac{u^T A u}{u^T B u} \quad (1)$$

However, numerically, we will consider a basis that is orthonormal on a certain inner product. Such a basis of the Krylov subspace is typically constructed through an iterative method itself, which will be called the inner iteration. The original iteration of Algorithm 1 will be called the outer iteration. We shall discuss in the subsections later three methods for constructing the basis.

In our presentation of Algorithm 1, we have assumed for convenience that $\dim K_m = m + 1$ so that we construct a full basis $z_0, z_1, \ldots, z_m$. Generally, if $\dim K_m = p + 1 < m + 1$, we generate $z_0, z_1, \ldots, z_p$ only. Then, the Rayleigh-Ritz projection is simply carried out by replacing $Z_m$ with $Z_p = [z_0, z_1, \ldots, z_p]$ and (1) is still valid. Numerically, however, early termination at step $p$ of the inner iteration is not likely and a full basis is usually constructed even when $p < m$ in theory,
but this causes no problem as the larger space spanned by more vectors would yield a better approximation.

Given that the basic ingredients of Algorithm 1 are the projection and the Krylov subspaces, it is not surprising that some similar methods have been considered before. In [10, 11], Knyazev discussed and analyzed some very general theoretical methods and suggested several special cases, among which is the use of $K_m$ and (1). Morgan and Scott’s preconditioned Lanczos algorithm [18] takes a similar iteration but uses the smallest Ritz value of the matrix $A - \rho_k B$ rather than that of the pencil $(A - \rho_k B, B)$ to update the eigenvalue. With an $m$ varied with each iteration, it has a quadratic convergence property. We point out however that the quadratic convergence is not a desirable property because it is achieved at the cost of increasingly larger $m$ and it prevents improvement of convergence by preconditioning (there is hardly any need to accelerate a quadratic convergent algorithm). Our study will be somewhat of different nature in that we consider accelerating convergence by changing certain conditions of the problem through equivalent transformations (see section 5) as opposed to increasing $m$.

Also related to ours are methods based on inverting a shifted matrix $A - \rho_k B$ or its projection, which include the inverse iteration and the Jacobi-Davidson method [24]. When the inversion is solved approximately by an iterative method, the solution is extracted from a Krylov subspace generated by $A - \rho_k B$ (or its projection). In these cases, it is chosen to satisfy the related linear system. We note that the Jacobi-Davidson method also uses the Rayleigh-Ritz projection in the outer iteration, the cost of which increases with the iteration. By fixing the size of subspaces for projection, the cost of Algorithm 1 is fixed per outer iteration.

When $B = I$, it is easy to see that Algorithm 1 is just the standard restarted Lanczos algorithm for $A$. In this regard, our investigation is on the version with a fixed $m$ and on how $m$ affects the convergence. Furthermore, our development will lead to a preconditioning strategy that transforms $(A, I)$ to the pencil problem $(L^{-1}AL^{-T}, L^{-1}LT^{-1})$ (for some suitably chosen $L$), to which Algorithm 1 will be applied. This transformation to a more complicated problem may seem counter intuitive but an important feature of Algorithm 1 is that the case $B = I$ offers no advantage than a more general $B$.

We now discuss in details the construction of a basis for $K_m$ in Algorithm 1.

2.1 Orthonormal basis by the Lanczos algorithm

One obvious choice of the basis of the Krylov subspace $K_m$ is the orthonormal one as constructed by applying the Lanczos algorithm to $C_k = A - \rho_k B$ and $x_k$ (see Algorithm 2 below). Simultaneously with the Lanczos process, we produce a tridiagonal matrix $T = Z^T_m C_k Z_m$. The Lanczos process requires $m+1$ matrix-vector multiplications by $C_k = A - \rho_k B$. Once the basis has been constructed, we form $B_m = Z^T_m B Z_m$, which requires $m + 1$ matrix-vector multiplications by $B$. Note that $A_m = Z^T_m C_k Z_m = T$. Here we state the Lanczos algorithm.

Algorithm 2: Orthonormal basis by Lanczos.
Input: $C_k = A - \rho_k B$, an approximate eigenvector = $x_k$;
Output: $Z_m = [z_0, z_1, \ldots, z_m]$

$z_0 = x_k / \| x_k \|$; $\beta_0 = 0$; $z_{-1} = 0$

For $i = 0 : (m - 1)$,

$w = C_k z_i$

$w = w - \beta_i z_{i-1}$
\[ \alpha_i = z_i^T w \]
\[ w = w - \alpha_i z_i \]
\[ \beta_{i+1} = \|w\|_2 \]
\[ z_{i+1} = w/\beta_{i+1} \]
End

With the orthonormal basis, \( B_m \) is in general a full matrix and we need to solve a generalized eigenvalue problem for \((A_m, B_m)\). While \( A_m = T \) in the exact arithmetic, it may not be valid in a finite precision arithmetic for larger \( m \) when there could be severe loss of orthogonality among \( z_i \). This can be corrected by either computing \( A_m = Z_m^T C_k Z_m \) explicitly or using reorthogonalization [6] in the Lanczos process. We note that \( C_k Z_m \) has been computed in the Lanczos algorithm and can be stored for forming \( A_m \).

2.2 \( B \)-orthonormal basis by the Arnoldi algorithm

We can also construct a \( B \)-orthonormal basis for \( K_m \) by the modified Gram-Schmidt process in the \( B \)-inner product, which is essentially the Arnoldi algorithm. The advantage of this approach is a simpler projection problem with \( B_m = I \); but it is at the cost of a longer recurrence. We also need to compute \( A_m = Z_m^T C_k Z_m \). We state the algorithm here.

Algorithm 3: \( B \)-orthonormal basis by Arnoldi.
Input \( C_k = A - \rho_k B \), approximate eigenvector \( x_k \).
Output: \( Z_m = [z_0, z_1, \ldots, z_m] \)
\[ z_0 = x_k / \|x_k\|_B \]
For \( i = 0 : (m - 1) \)
\[ w = C_k z_i \]
For \( j = 0 : i \)
\[ h_{j,i} = z_j^T B w; \]
\[ w = w - h_{j,i} z_j \]
end
\[ z_{i+1} = w / \|w\|_B \]
End

Each step of the Arnoldi algorithm requires 2 matrix-vector multiplications with one by \( C_k \) and one by \( B \). In addition, we need to store \( B z_i \) from each iteration in order to save matrix-vector multiplications, resulting in storage cost of \( m \) vectors. We note again that, for larger \( m \), the \( B \)-orthogonality among the columns of \( Z_m \) may gradually be lost. This leads to deterioration of the equation \( B_m = I \). In that case, we need either reorthogonalization in the Arnoldi algorithm, or explicit computations of \( B_m = Z_m^T B Z_m \).

In comparing the two constructions, the computational costs associated with them are very comparable. They both require \( 2(m + 1) \) matrix-vector multiplications. The Arnoldi recurrence is more expensive in both flops and storage than the Lanczos recurrence while it produces a more compact projection matrix than the Lanczos algorithm. Clearly, these differences are very minor when \( m \) is not too large, which is the case of interest in practical implementations. In terms of numerical stability of these two theoretically equivalent processes, our testing suggests that there is very little difference. However, for the preconditioned version of Algorithm 1 that we will discuss in Section 5, the approach by the Arnoldi algorithm seems to have some advantage, see section 5.
2.3 $C_k$-orthogonal basis by a variation of the Lanczos algorithm

It is also possible to construct $Z_m$ that is $C_k$-orthogonal by a variation of the Lanczos algorithm with a three term recurrence. Then the projection $A_m = Z_m^T C_k Z_m$ will have a compact form, leading to a computationally more effective approach than the previous two. However, it is less stable owing to the indefiniteness of $C_k$. For the theoretical interest, we outline this variation of the Lanczos algorithm for $C = C_k$ in the form of a full matrix tridiagonalization.

Let $C = QTQ^T$ be the standard tridiagonalization of the Lanczos algorithm for $C$ where $T$ is tridiagonal and $Q$ is orthogonal with $x_k/\|x_k\|$ as its first column. For the sake of simplicity in presentation, we assume here that $\rho_k$ is between the first and the second eigenvalue, which implies that $C$ has exactly one negative eigenvalue. Noting that the $(1, 1)$ entry of $T$ is $x_k^T C_k x_k / x_k^T x_k = 0$, let $T = FDF^T$ be the block $LDL^T$ decomposition of $T$, where

$$D = \begin{pmatrix} D_0 & 0 \\ 0 & I \end{pmatrix} \quad \text{and} \quad D_0 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. $$

Write

$$Z = QF^{-T}. $$

It is easy to check that

$$Z^T C Z = F^{-1} Q^T C Q F^{-T} = D$$

and

$$C Z = Q F D = Q F^{-T} F^T F D = Z J $$

(2)

where $J = F^T F D$ is block tridiagonal. Now a Lanczos three term recurrence can be easily derived from (2) to construct the columns of $Z$, which still form a basis for the Krylov subspace and is essentially $C$-orthogonal. However, our tests show that this is numerically less stable. Therefore, we shall not consider this further and omit a detailed algorithm here.

3 Convergence analysis.

In this section, we study convergence properties of Algorithm 1 that include a global convergence result and a local one on the rate of linear convergence. In particular, we identify the factors that affect the speed of the convergence so as to develop preconditioning strategy to improve the convergence.

We first prove that Algorithm 1 always converges to an eigenpair. For that, we need the following proposition, the proof of which is straightforward.

**Proposition 1** Let $\lambda_1$ be the smallest eigenvalue of $(A, B)$ and $(\rho_k, x_k)$ be the eigenpair approximation obtained by Algorithm 1 at step $k$. Then

$$\lambda_1 \leq \rho_{k+1} \leq \rho_k$$

and

$$x_k^T (A - \rho_k B) x_k = 0. $$

(3)
Theorem 1 Let \((\rho_k, x_k)\) denote the eigenpair approximation obtained by Algorithm 1 at step \(k\). Then \(\rho_k\) converges to some eigenvalue \(\lambda\) of \((A, B)\) and \(\|(A - \lambda B)x_k\| \to 0\) (i.e. \(x_k\) converges in direction to a corresponding eigenvector).

Proof From Proposition 1, we obtain that \(\rho_k\) is convergent. Since \(x_k\) is bounded, there is a convergent subsequence \(x_{n_k}\). Let

\[
\lim \rho_k = \hat{\lambda}, \quad \text{and} \quad \lim x_{n_k} = \hat{x}.
\]

Write \(\hat{r} = (A - \hat{\lambda}B)\hat{x}\). Then it follows from (3) that,

\[
\hat{x}^T \hat{r} = \hat{x}^T (A - \hat{\lambda}B)\hat{x} = 0.
\]

Suppose now \(\hat{r} \neq 0\). We consider the projection of \((A, B)\) onto \(\text{span}\{\hat{x}, \hat{r}\}\) by defining

\[
\hat{A} = [\hat{x}, \hat{r}]^T A[\hat{x}, \hat{r}] \quad \text{and} \quad \hat{B} = [\hat{x}, \hat{r}]^T B[\hat{x}, \hat{r}].
\]

Noting that \(\{\hat{x}, \hat{r}\}\) is orthogonal, we have \(\hat{B} > 0\) and

\[
\hat{A} - \hat{\lambda}\hat{B} = \begin{pmatrix}
0 & \hat{r}^T \\
\hat{r}^T & \hat{r}^T (A - \hat{\lambda}B)\hat{r}
\end{pmatrix}
\]

is indefinite. Thus the smallest eigenvalue of \((\hat{A}, \hat{B})\), denoted by \(\bar{\lambda}\), is less than \(\hat{\lambda}\), i.e.

\[
\bar{\lambda} < \hat{\lambda}.
\]

Furthermore, at step \(k\), define \(r_k = (A - \rho_k B)x_k\),

\[
\hat{A}_k = [x_k, r_k]^T A[x_k, r_k] \quad \text{and} \quad \hat{B}_k = [x_k, r_k]^T B[x_k, r_k].
\]

Let \(\bar{\lambda}_{k+1}\) be the smallest eigenvalue of \((\hat{A}_k, \hat{B}_k)\). Clearly, as \(n_k \to \infty\), \(\hat{A}_{n_k} \to \hat{A}\) and \(\hat{B}_{n_k} \to \hat{B}\).

Hence by the continuity property of the eigenvalue, we have

\[
\bar{\lambda}_{n_k+1} \to \bar{\lambda}.
\]

On the other hand, \(\rho_{k+1}\) is the smallest eigenvalue of the projection of \((A, B)\) on \(K_m = \text{span}\{x_k, (A - \rho_k B)x_k, \ldots, (A - \rho_k B)^m x_k\}\), which implies

\[
\rho_{k+1} \leq \bar{\lambda}_{k+1}.
\]

Finally, combining the above together, we have obtained

\[
\bar{\lambda} = \lim \bar{\lambda}_{n_k+1} \geq \lim \rho_{n_k+1} = \hat{\lambda}\]

which is a contradiction to (4). Therefore, \(\hat{r} = (A - \hat{\lambda}B)\hat{x} = 0\), i.e. \(\bar{\lambda}\) is an eigenvalue and \(\|(A - \hat{\lambda}B)x_{n_k}\| \to 0\).

Now, to show \(\|(A - \hat{\lambda}B)x_k\| \to 0\), suppose there is a subsequence \(m_k\) such that \(\|(A - \hat{\lambda}B)x_{m_k}\| \geq \alpha > 0\). From the subsequence \(m_k\), there is a subsequence \(n_k\) for which \(x_{n_k}\) is convergent. Hence by virtue of the above proof, \(\|(A - \hat{\lambda}B)x_{n_k}\| \to 0\), which is a contradiction. This completes the proof. \(\Box\)
Next, we study the speed of convergence through a local analysis. In particular, we show that \( \rho_k \) converges at least linearly.

**Lemma 1** Let \( \lambda_1 \) be the smallest eigenvalue of \((A, B)\), \( x \) be a corresponding unit eigenvector and \((\rho_k, x_k)\) be the eigenpair approximation obtained by Algorithm 1 at step \( k \). Let \( \sigma_1 \) be the smallest eigenvalue of \( A - \rho_k B \) and \( u_1 \) be a corresponding unit eigenvector. Then

\[
(\rho_k - \lambda_1) \frac{x^T B x}{u_1^T B u_1} \leq \frac{\|\sigma_1\|}{u_1^T B u_1} \leq \rho_k - \lambda_1. \tag{5}
\]

Asymptotically, if \( \rho_k \to \lambda_1 \), we have

\[
\frac{\sigma_1}{u_1^T B u_1} = (\lambda_1 - \rho_k) + O((\lambda_1 - \rho_k)^2).
\]

**Proof** First, from the definition, we have

\[(A - \sigma_1 I - \rho_k B)u_1 = 0, \quad \text{and} \quad (A - \delta I - \lambda_1 B)x = 0.\]

Furthermore, \( A - \sigma_1 I - \rho_k B \geq 0 \) and \( A - \delta I - \lambda_1 B \geq 0 \). Therefore \((\rho_k, u_1)\) is the smallest eigenpair of \((A - \sigma_1 I, B)\) and \((\lambda_1, x)\) is the smallest eigenpair of \((A, B)\). Clearly \( A - \rho_k B \) is indefinite and hence \( \sigma_1 \leq 0 \). Now using Theorem 3 of Appendix, we have

\[
\frac{-\sigma_1}{u_1^T B u_1} \leq \rho_k - \lambda_1 \leq \frac{-\sigma_1}{x^T B x}
\]

which leads to the bound (5).

To prove the asymptotic expansion, let \( \sigma_1(t) \) be the smallest eigenvalue of \( A - tB \). Then \( \sigma_1(\lambda_1) = 0 \) and \( \sigma_1(\rho_k) = \sigma_1 \). Using the analytic perturbation theory, we obtain \( \sigma_1'(\rho_k) = -u_1^T B u_1 \) and hence

\[
\sigma_1(t) = \sigma_1(\rho_k) + \sigma_1'(\rho_k)(t - \rho_k) + O((t - \rho_k)^2)
\]

\[
= \sigma_1 - u_1^T B u_1(t - \rho_k) + O((t - \rho_k)^2).
\]

Choosing \( t = \lambda_1 \), we have

\[
0 = \sigma_1(\lambda_1) = \sigma_1 - u_1^T B u_1(\lambda_1 - \rho_k) + O((\lambda_1 - \rho_k)^2),
\]

from which the expansion follows. \( \square \)

We now present our main convergence result. We assume that \( \rho_k \) is already between the first and the second smallest eigenvalues. Then by Theorem 1, it converges to the smallest eigenvalue.

**Theorem 2** Let \( \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_n \) be the eigenvalues of \((A, B)\) and \((\rho_{k+1}, x_{k+1})\) be the approximate eigenpair obtained by Algorithm 1 from \((\rho_k, x_k)\). Let \( \sigma_1 \leq \sigma_2 \leq \cdots \leq \sigma_n \) be the eigenvalues of \( A - \rho_k B \) and \( u_1 \) be a unit eigenvector corresponding to \( \sigma_1 \). Assume \( \lambda_1 < \rho_k < \lambda_2 \). Then

\[
\rho_{k+1} - \lambda_1 \leq (\rho_k - \lambda_1)\epsilon_m + 2(\rho_k - \lambda_1)^{3/2}\epsilon_m \left( \frac{\|B\|_{\sigma_2}}{\sigma_2} \right)^{1/2} + \delta_k \tag{6}
\]
where
\[ 0 \leq \delta_k = \rho_k - \lambda_1 + \frac{\sigma_1}{u_k^T Bu_k} = O((\rho_k - \lambda_1)^2) \]

and
\[ \epsilon_m = \min_{p \in \mathcal{P}_m, p(\sigma_1) = 1} \max_{i \neq 1} |p(\sigma_i)| \]

with \( \mathcal{P}_m \) denoting the set of all polynomials of degree not greater than \( m \).

**Proof** First, write \( C = A - \rho_k B \) and \( K_m = \{ p(C)x_k, p \in \mathcal{P}_m \} \). At step \( k \) of the algorithm, we have

\[
\rho_{k+1} = \min_{0 \neq u \in K_m} \frac{u^T Au}{u^T Bu} = \rho_k + \min_{0 \neq u \in K_m} \frac{u^T (A - \rho_k B) u}{u^T Bu} = \rho_k + \min_{p \in \mathcal{P}_m, p(C)x_k \neq 0} \frac{x_k^T p(C)Bp(C)x_k}{x_k^T p(C)Bp(C)x_k}
\]

Let \( A - \rho_k B = U \Sigma U^T \) be the eigenvalue decomposition of \( A - \rho_k B \), where \( U = [u_1, u_2, \ldots, u_n] \) is orthogonal and \( \Sigma = \text{diag}\{\sigma_1, \sigma_2, \ldots, \sigma_n\} \). Let \( q \) be the minimizing polynomial in \( \epsilon_m \) with \( q(\sigma_1) = 1 \) and \( \max_{i \neq 1} |q(\sigma_i)| = \epsilon_m < 1 \). Then, it follows from \( x_k^T A \rho_k B x_k = 0 \) that \( u_k^T x_k \neq 0 \) and hence \( q(C)x_k \neq 0 \). Then

\[
\rho_{k+1} \leq \rho_k + \frac{x_k^T q(C)Bq(C)x_k}{x_k^T Uq(\Sigma)B_1q(\Sigma)U^T x_k}
= \rho_k + \frac{x_k^T Uq(\Sigma)B_1q(\Sigma)U^T x_k}{x_k^T Uq(\Sigma)B_1q(\Sigma)U^T x_k} = \rho_k + \frac{y^T q(\Sigma)\Sigma y}{y^T q(\Sigma)B_1q(\Sigma)y}
\]

where \( B_1 = U^T B U \) and \( y = U^T x_k \). Let \( y = [y_1, y_2, \ldots, y_n]^T \) and \( \hat{y} = [0, y_2, \ldots, y_n]^T \). Then,

\[
y^T q(\Sigma)B_1q(\Sigma)y = (y_1 e_1 + \hat{y})^T q(\Sigma)B_1q(\Sigma)(y_1 e_1 + \hat{y}) = y_1^2 q(\sigma_1)^2 e_1^T B_1 e_1 + 2y_1 q(\sigma_1) e_1^T B_1 q(\Sigma) \hat{y} + \hat{y}^T q(\Sigma)B_1q(\Sigma)\hat{y}
= y_1^2 \beta_1^2 + 2y_1 \beta_1 + \beta_3
\]

where

\[
\beta_1^2 = e_1^T B_1 e_1 = u_1^T Bu_1,
\beta_3 = \hat{y}^T q(\Sigma)B_1q(\Sigma)\hat{y} \leq \max_{i \neq 1} |q(\sigma_i)| \|B_1\| \|\hat{y}\|^2
= \epsilon_m^2 \|B\| \|\hat{y}\|^2
\]

and

\[ |\beta_2| = |e_1^T B_1 q(\Sigma) \hat{y}| \leq \beta_1 \beta_3. \]
Using Proposition 1, we have $y^T \Sigma y = x_k^T (A - \rho_k B)x_k = 0$, i.e. $\sum_i \sigma_i y_i^2 = 0$. Then

$$|\sigma_1| y_1^2 = \sum_{i \neq 1} \sigma_i y_i^2 \geq \sigma_2 \|y\|^2$$

and hence

$$\beta_3 \leq \epsilon_m \|B\|^{1/2} \left( \frac{|\sigma_1|}{\sigma_2} \right)^{1/2} |y_1|.$$  \hspace{1cm} (8)

On the other hand, we also have

$$y^T q^2(\Sigma) \Sigma y = \sum_i \sigma_i q^2(\sigma_i) y_i^2 \leq y^T \Sigma y = 0,$$

and

$$\hat{y}^T q^2(\Sigma) \Sigma \hat{y} = \sum_{i \neq 1} \sigma_i q^2(\sigma_i) y_i^2 \leq \epsilon_m \sum_{i \neq 1} \sigma_i y_i^2 = \epsilon_m |\sigma_1| y_1^2.$$  \hspace{1cm} (9)

where we note that $q(\sigma_1) = 1$ and $|q(\sigma_i)| \leq \epsilon_m < 1$ for $i > 1$.

Thus

$$\frac{y^T q^2(\Sigma) \Sigma y}{y^T q(\Sigma) B y q(\Sigma) y} \leq \frac{y_1^2 \sigma_1 + \hat{y}^T q(\Sigma)^2 \Sigma \hat{y}}{y_1^2 \beta_1^2 + 2|y_1| \beta_1 \beta_3 + \beta_3^2}$$

$$= \frac{\sigma_1}{\beta_1^2} \frac{2|y_1| \beta_1 \beta_3 + \beta_3^2}{y_1^2 \beta_1^2 + 2|y_1| \beta_1 \beta_3 + \beta_3^2}$$

$$\leq \frac{\sigma_1}{\beta_1^2} + 2 \left( \frac{|\sigma_1|}{\beta_1^2} \right)^{3/2} \epsilon_m \left( \frac{\|B\|}{\sigma_2} \right)^{1/2} + \frac{|\sigma_1| \epsilon_m^2}{\beta_1^4}.$$  \hspace{1cm} (10)

where we have used (8) and (9). Finally, combining (7), (10) and Lemma 1, we have

$$0 \leq \rho_{k+1} - \lambda_1 \leq \rho_k - \lambda_1 + \frac{\sigma_1}{\beta_1^2} + 2(\rho_k - \lambda_1)^{3/2} \epsilon_m \left( \frac{\|B\|}{\sigma_2} \right)^{1/2} + (\rho_k - \lambda_1) \epsilon_m^2,$$

which leads to the theorem.

\[ \Box \]

It is well known that $\epsilon_m$ in the theorem can be bounded in terms of $\sigma_i$ as (see [16, Theorem 1.64] for example)

$$\epsilon_m \leq 2 \left( \frac{1 - \sqrt[3]{\psi}}{1 + \sqrt[3]{\psi}} \right)^n \text{ with } \psi = \frac{\sigma_2 - \sigma_1}{\sigma_n - \sigma_1}.$$  \hspace{1cm} (11)

Then the speed of convergence depends on the distribution of the eigenvalues $\sigma_i$ of $A - \rho_k B$ but not those of $(A, B)$. This difference is of fundamental importance as it allows acceleration of convergence by equivalent transformations that change the eigenvalues of $A - \rho_k B$ but leave those of $(A, B)$ unchanged (see the discussion on preconditioning in Section 5). On the other hand, the bound
shows accelerated convergence when \( m \) is increased. In this regard, our numerical tests suggest that the convergence rate decreases very rapidly as \( m \) increases (see Section 4).

For the special case of \( m = 1 \) and \( B = I \), Algorithm 1 is just the steepest descent method for \( A \). It is easy to check in this case that

\[
\epsilon_1 = \frac{1 - \psi}{1 + \psi} \quad \text{with} \quad \psi = \frac{\sigma_2 - \sigma_1}{\sigma_n - \sigma_1} = \frac{\lambda_2 - \lambda_1}{\lambda_n - \lambda_1}.
\]

Using this in Theorem 2, we recover the classical convergence bound for the steepest descent method [9, p.617]. We note that there is a stronger global convergence result in this case, i.e. \( \rho_k \) is guaranteed to converge to the smallest eigenvalue if the initial vector has a nontrivial component in the smallest eigenvector (see [9, p.613]). There is no such result known for the case \( B \neq I \).

Asymptotically we can also express the bound in terms of the eigenvalues of \( A - \lambda_1 B \) instead of \( \sigma_i \) which is dependent of \( k \). We state it as the following corollary; but point out that the bound of Theorem 2 is more informative.

**Corollary 1** Let \( 0 = \gamma_1 \leq \gamma_2 \leq \cdots \leq \gamma_n \) be the eigenvalues of \( A - \lambda_1 B \). Then, we have asymptotically

\[
\rho_{k+1} - \lambda_1 \leq 4 \left( \frac{1 - \sqrt{\psi_0}}{1 + \sqrt{\psi_0}} \right)^{2m} (\rho_k - \lambda_1) + o(\rho_k - \lambda_1),
\]

where \( \psi_0 = \frac{\sigma_2}{\sigma_n} \).

The proof follows from combining (11) with \( \sigma_i - \gamma_i = O(\rho_k - \lambda_1) \) and \( \psi - \psi_0 = O(\rho_k - \lambda_1) \).

### 4 Numerical Examples - I

In this section, we present numerical examples to illustrate the convergence behavior of Algorithm 1. Here, we demonstrate the linear convergence property and the effect of \( m \) on the convergence rate.

**Example 1:** Consider the Laplace eigenvalue problem with the Dirichlet boundary condition on an L-shape region. A finite element discretization on a triangular mesh with 7585 interior nodes (using PDE toolbox of MATLAB) leads to a pencil eigenvalue problem \( Ax = \lambda Bx \). We apply Algorithm 1 to find the smallest eigenvalue with a random initial vector and the stopping criterion is set as \( \|r_k\|/\|r_0\| \leq 10^{-7} \) where \( r_k = Ax_k - \rho_k Bx_k \). We give the convergence history of the residual \( \|r_k\| \) for \( m = 2, 4, 8, 16, 32 \) (from top down resp.) in Figure 1. We present in Figure 2 (a) the number of outer iterations required to achieve convergence for each \( m \) in the range of \( 1 \leq m \leq 100 \) and, correspondingly in Figure 2 (b), the total number of inner iterations.

We observe that the residual converges linearly with the rate decreased as \( m \) increases. Furthermore, from Figure 2 (a), the number of outer iterations decreases very rapidly (quadratically or even exponentially) as \( m \) increases and it almost reaches its stationary limit for \( m \) around 70. Because of this peculiar property, we see from Figure 2 (b) that the total number of inner iterations is near minimal for a large range of \( m \) (40 < \( m < 80 \) in this case).

**Example 2:** We consider a standard eigenvalue problem \( (B = I) \) in which \( A \) is a five point finite difference discretization of the Laplace operator on the \( 32 \times 32 \) uniform mesh of the unit square. Again, we apply Algorithm 1 to find the smallest eigenvalue with a random initial vector.
In this case, it is simply a restarted Lanczos algorithm and we shall consider its comparison with the Lanczos algorithm without restart. In Figure 3, we present the convergence history of $\rho_k - \lambda_1$ where $\rho_k$ is the approximate eigenvalue obtained at each inner iteration. They are plotted in the dot lines from top down for $m = 2, 4, 8, 16, 32$ respectively. The corresponding plot for the Lanczos algorithm (without restart) is given in the solid line.

We have also considered the number of outer iterations and the total number of inner iterations as a function of $m$ and observed the same behavior as in Example 1. We omit a similar figure here. In particular, the nearly exponential decrease of the outer iteration count implies that the convergence history with a moderate $m$ (in this case $m = 32$ or even 16) will be very close to the one with very large $m$ (i.e. Lanczos without restart in the solid line).

These examples confirm the linear convergence property of Algorithm 1. Furthermore, our numerical testing has consistently shown that the number of outer iterations decreases nearly exponentially as $m$ increases. This implies that near optimal performance of the algorithm can be achieved with a moderate $m$, which is very attractive in implementations. Unfortunately we have not been able to explain this interesting behavior with our convergence results. Even for the restarted Lanczos algorithm, it seems to be a phenomenon not observed before and can not be explained by the convergence theory of the Lanczos algorithm either.

5 Preconditioning

In this section, we discuss how to accelerate the convergence of Algorithm 1 through some equivalent transformations, which we call preconditioning, and we shall present the preconditioned version of Algorithm 1.
From our convergence result (Theorem 2), the rate of convergence depends on the spectral distribution of $C_k = A - \rho_k B$, i.e. the separation of $\sigma_1$ from the rest of eigenvalues $\sigma_2, \ldots, \sigma_n$ of $C_k$. With an approximate eigenpair $(\rho_k, x_k)$, we consider for some matrix $L_k$ the transformed pencil

\[(\hat{A}_k, \hat{B}_k) \equiv (L_k^{-1} A L_k^{-T}, L_k^{-1} B L_k^{-T}),\]  

which has the same eigenvalues as $(A, B)$. Thus, applying one step of Algorithm 1 to $(\hat{A}_k, \hat{B}_k)$, we have the bound (6) of Theorem 2 with the rate of convergence $\hat{\epsilon}_m^2$ determined by $\hat{\sigma}_i$ $(1 \leq i \leq n)$, the eigenvalues of

\[\hat{C}_k = \hat{A}_k - \rho_k \hat{B}_k = L_k^{-1} (A - \rho_k B) L_k^{-T}.\]

We can now suitably choose $L_k$ to obtain a favorable distribution of $\hat{\sigma}_i$ and hence a smaller $\epsilon_m$. We shall call (13) a preconditioning transformation.

One preconditioning transformation can be constructed using the $LDL^T$ factorization of a symmetric matrix [6]. For example, if

\[A - \rho_k B = L_k D_k L_k^T\]

is the $LDL^T$ factorization with $D_k$ being a diagonal matrix of ±1, choosing this $L_k$ results in $\hat{C}_k = D_k$. Then, at the convergence stage with $\lambda_1 < \rho_k < \lambda_2$, we have $\hat{\sigma}_1 = -1$ and $\hat{\sigma}_2 = \cdots = \hat{\sigma}_n = 1$, which implies

\[\epsilon_m = 0\]

and thus by Theorem 2,

\[\rho_{k+1} - \lambda_1 \leq \delta_k = O((\rho_k - \lambda_1)^2).\]
Figure 3: **Example 2:** Eigenvalue convergence history against each inner iteration for restarted Lanczos with \( m = 2, 4, 8, 16, 32 \) in dot lines from top down and for Lanczos without restart in solid line.

![Eigenvalue convergence history](image)

We conclude that Algorithm 1, when applied to \((\hat{A}_k, \hat{B}_k)\) at step \( k \) using the exact \( LDL^T \) factorization, converges quadratically. This is even true with \( m = 1 \) (i.e. the steepest descent method).

Similarly, in light of Corollary 1, if we use a constant \( L_k = L \) obtained from the \( LDL^T \) factorization \( A - \lambda_1 B = LDL^T \) (assuming \( \lambda_1 \) is known) with \( D_k \) being a diagonal matrix of 0 and 1, Algorithm 2 also converges quadratically.

What we have described above is the ideal situations of fast quadratic convergence property achieved by using an exact \( LDL^T \) factorization. In practice, we can use an incomplete \( LDL^T \) factorization \( A - \rho_k B \approx L_k D_k L_k^T \) (through incomplete \( LU \) factorization, see [23, Chapter 10]). Then we will have a nonzero but small \( \epsilon_m \) and hence fast linear convergence. Indeed, to be efficient, we can consider a constant \( L_k = L \) as obtained from an incomplete \( LDL^T \) factorization of

\[
A - \lambda_0 B \approx LDL^T
\]

where \( \lambda_0 \) is a sufficiently good approximation of \( \lambda_1 \) and apply Algorithm 1 to (13). Then, the preconditioned algorithm converges linearly with the rate determined by the eigenvalues of

\[
\hat{C}_k = L^{-1}(A - \lambda_0 B)L^{-T} + (\lambda_0 - \rho_k)L^{-1}BL^{-T} \approx D
\]

which has a better spectral distribution as long as \((\lambda_0 - \rho_k)L^{-1}BL^{-T}\) is small relative to \( \hat{C}_k \). We note that \( \lambda_0 - \rho_k \) need not be very small if \( L^{-1}BL^{-T} \) is small (e.g. for the discretization of differential operators). It may work even when \( \lambda_0 < \lambda_1 \), for which \( A - \lambda_0 B > 0 \) and the incomplete \( LDL^T \) factorization becomes incomplete Cholesky factorization. It is also possible to construct \( L \) based on other factorization, such as approximate eigenvalue decomposition.
As in the preconditioned iterative methods for linear systems, the preconditioned iteration of Algorithm 1 can be implemented implicitly, i.e., without explicitly forming the transformed problem \((\hat{A}_k, \hat{B}_k)\) or \(\hat{C}_k\). We derive a preconditioned version of the algorithm in the rest of this section.

Let \((\rho_k, x_k)\) be the approximate eigenpair that has been obtained at step \(k\) for the pencil \((A, B)\). Then \((\rho_k, \hat{x}_k)\) with \(\hat{x}_k = L_k^T x_k\) is the corresponding approximate eigenpair for the transformed pencil (13). By applying one step of iteration to the transformed pencil, the new approximation is obtained by constructing a basis \(\hat{z}_0, \hat{z}_1, \ldots, \hat{z}_m\) for the Krylov subspace

\[
\hat{K}_m = \text{span}\{\hat{x}_k, \hat{C}_k \hat{x}_k, \hat{C}_k^2 \hat{x}_k, \ldots, \hat{C}_k^m \hat{x}_k\},
\]

and form the projection problem

\[
\hat{Z}_m^T (\hat{A}_k, \hat{B}_k) \hat{Z}_m = (\hat{Z}_m^T L_k^{-1} AL_k^{-T} \hat{Z}_m, \hat{Z}_m^T L_k^{-1} BL_k^{-T} \hat{Z}_m)
\]

(14)

where \(\hat{Z}_m = [\hat{z}_0, \hat{z}_1, \ldots, \hat{z}_m]\). If \((\rho_{k+1}, \hat{v})\) is the smallest eigenpair of the above projection problem, then \((\rho_{k+1}, L_k^{-T} \hat{Z}_m \hat{v})\) is the new approximate eigenpair for \((A, B)\).

Now, let

\[
Z_m = L_k^{-T} \hat{Z}_m = [L_k^{-T} \hat{z}_0, L_k^{-T} \hat{z}_1, \ldots, L_k^{-T} \hat{z}_m].
\]

Then, the new approximate eigenpair can be written as \((\rho_{k+1}, Z_m \hat{v})\) and the projection problem (14) is

\[
(Z_m^T A Z_m, Z_m^T B Z_m).
\]

Therefore, to complete the \(k\)-th iteration, we only need to construct \(Z_m = [z_0, z_1, \ldots, z_m]\), i.e., a basis for the subspace \(L_k^{-T} \hat{K}_m\). The actual construction of \(z_i\) depends on which method we use and will be given in details in the subsections later.

Here, we summarize the preconditioned algorithm as follows.

**Algorithm 4:** Preconditioned Inverse Free Krylov Subspace Method.

Input \(m\) and an initial approximation \(x_0\) with \(\|x_0\| = 1\);

\(\rho_0 = \rho(x_0)\);

For \(k = 0, 1, 2, \ldots \) until convergence

- Construct a preconditioner \(L_k\);
- Construct a preconditioned basis \(Z_m = [z_0, z_1, \ldots, z_m]\) for \(L_k^{-T} \hat{K}_m\);
- Form \(A_m = Z_m^T (A - \rho_k B) Z_m\) and \(B_m = Z_m^T B Z_m\);
- Find the smallest eigenvalue \(\mu_1\) and a eigenvector \(v\) for \((A_m, B_m)\);
- \(\rho_{k+1} = \rho_k + \mu_1\) and \(x_{k+1} = Z_m v\).

End

**Remark:** As in the linear system case, the above algorithm takes the same form as the original one except using a preconditioned search space \(L_k^{-T} \hat{K}_m\). In the following subsections, we discuss the construction of a preconditioned basis by the Arnoldi algorithm and the Lanczos algorithm corresponding to the construction in Sections 2.1 and 2.2. Our numerical testing suggests that the Arnoldi algorithm might be more stable than the Lanczos algorithm in some cases.
5.1 Preconditioned basis by the Arnoldi method

In the Arnoldi method, we construct $\hat{z}_0, \hat{z}_1, \ldots, \hat{z}_m$ as a $\hat{B}$-orthonormal basis for $\hat{K}_m$. Correspondingly, $z_0, z_1, \ldots, z_m$ is a $B$-orthonormal basis for $L_k^{-T}K_m$. Starting from $\hat{z}_0 = \hat{x}_k/\|\hat{x}_k\|_B$, the recurrence for $\hat{z}_i$ is

$$h_{i+1,i}\hat{z}_{i+1} = \hat{C}_k\hat{z}_i - h_{0,i}\hat{z}_0 - h_{1,i}\hat{z}_1 - \cdots - h_{i,i}\hat{z}_i,$$

where $h_{j,i} = z_j^T\hat{B}\hat{C}_k\hat{z}_i$ for $0 \leq j \leq i$. Thus, using $\hat{x}_k = L_k^T x_k$, we have

$$z_0 = L_k^{-T}\hat{z}_0 = L_k^{-T}\hat{x}_k/\|\hat{x}_k\|_B = x_k/\|x_k\|_B,$$

and

$$h_{i+1,i}z_{i+1} = L_k^{-T}L_k^{-1}C_kz_i - h_{0,i}z_0 - h_{1,i}z_1 - \cdots - h_{i,i}z_i,$$

with $h_{j,i} = z_j^T BL_kL_k^{-1}C_k z_i$. Clearly, $z_j$ is $B$-orthonormal, and $h_{j,i}$ above ensures this condition. From this, we arrive at the following algorithm.

**Algorithm 5**: Preconditioned $B$-orthonormal basis by Arnoldi

Input $C_k = A - \rho_k B$, approximate eigenvector $x_k$ and a preconditioner $L_k$.

- $z_0 = x_k/\|x_k\|_B$;
- For $i = 0 : (m - 1)$,
  - $w = L_k^{-T}L_k^{-1}C_kz_i$
  - For $j = 0 : i$
    - $h_{j,i} = z_j^T Bw$
    - $w = w - h_{j,i}z_j$
  - End
- $z_{i+1} = w/\|w\|_B$
- End

We see from the algorithm that only $L_k^{-T}L_k^{-1}$ is needed in our construction. If we use $\lambda_0 < \lambda_1$ and $L_k = L$ is an incomplete Cholesky factor, i.e. $A = \lambda_0 B \sim LL^T$, then we can use any matrix approximating $(A - \lambda_0 B)^{-1}$ for $L_k^{-T}L_k^{-1}$ without explicitly forming $L_k$. For example, for differential operators, we can use the multigrid or domain decomposition preconditioners for $A - \lambda_0 B$ directly.

5.2 Preconditioned basis by the Lanczos method

In the Lanczos method, we construct $\hat{z}_0, \hat{z}_1, \ldots, \hat{z}_m$ as an orthonormal basis for $\hat{K}_m$. Then the corresponding basis $z_0, z_1, \ldots, z_m$ is $M$-orthonormal where $M = L_kL_k^T$. Starting from $\hat{z}_0 = \hat{x}_k/\|\hat{x}_k\|_2$, the recurrence is

$$\beta_{i+1,1}\hat{z}_{i+1} = \hat{w}_i = \hat{C}_k\hat{z}_i - \alpha_i\hat{z}_i - \beta_i\hat{z}_i-1,$$

where $\alpha_i = \hat{w}_i^T\hat{C}_k\hat{z}_i$ and $\beta_{i+1} = \|\hat{w}_i\|_2$. The resulting tridiagonal matrix $T$ as constructed from $\alpha$’s and $\beta$’s satisfies $T = Z_m^T\hat{C}_kZ_m$. Thus, using $\hat{x}_k = L_k^T x_k$, we have

$$z_0 = L_k^{-T}\hat{z}_0 = x_k/\|x_k\|_M,$$

and

$$\beta_{i+1}z_{i+1} = w_i = L_k^{-T}L_k^{-1}C_kz_i - \alpha_i z_i - \beta_i z_{i-1},$$
with \( \alpha_i = z_i^T C_k z_i \), and \( \beta_{i+1} = \| L_k^T w_i \| = \| w_i \|_M \). Clearly, the formulas for \( \alpha_i \) and \( \beta_{i+1} \) ensures \( z_j \) is \( M \)-orthonormal. Thus, we have the alternative formulas

\[
\alpha_i = z_i^T M (L_k^{-T} L_k^{-1} C_k z_i - \beta_i z_{i-1}), \quad \beta_i = z_{i-1}^T C_k z_i
\]

From this, we can derive a recurrence to construct the basis. We note that this construction normalizes \( z_i \) in the \( M \)-norm. In practice, \( M \) could be nearly singular. Therefore, it is more appropriate to normalize it in the 2-norm. The following algorithm is one of several possible formulations here.

**Algorithm 6:** Preconditioned basis by Lanczos

Input \( C_k = A - \rho_k B \), approximate eigenvector \( x_k \) and a preconditioner \( L_k \).

\[
z_0 = x_k / \| x_k \|_2; \quad z_{-1} = 0;
\]

For \( i = 0 : (m - 1) \),

\[
\begin{align*}
\beta_i &= z_{i-1}^T C_k z_i / z_{i-1}^T M z_{i-1} \\
w &= w - \beta_i z_{i-1} \\
\alpha_i &= z_i^T M w / z_i^T M z_i \\
w &= w - \alpha_i z_i \\
z_{i+1} &= w / \| w \|_2
\end{align*}
\]

End

6 Numerical Examples - II

In this section, we present some numerical examples to demonstrate the effectiveness and competitiveness of the preconditioned inverse free Krylov subspace method (Algorithm 4).

Example 3: \( A \) and \( B \) are the same as in Example 1. We apply Algorithm 4 to find the smallest eigenvalue (closest to \( \lambda_0 = 0 \)). We use a constant \( L_k \) as obtained by the threshold incomplete \( LDL^T \) factorization of \( A - \lambda_0 B = A \) with the drop tolerance \( 10^{-2} \). We compare our algorithm with the Jacobi-Davidson algorithm that uses the same number of inner iterations \( (m) \) and with the same kind of preconditioner. We give in Figure 4 the convergence history of the residual \( \| r_k \| \) of Algorithm 4 in solid lines and that of the Jacobi-Davidson algorithm in dot lines from top down for \( m = 2, 4, 8, 16 \) respectively. In Figure 5, we also present the number of outer iterations and the total number of inner iterations required to reduce \( \| r_k \| / \| r_0 \| \) to \( 10^{-7} \) for each \( m \) in (a) mark for Algorithm 4 and in (o) mark for the Jacobi-Davidson algorithm.

Comparing it with Example 1, the results clearly demonstrate the acceleration effect of preconditioning by significantly reducing the number of outer iterations (Fig. 4 and Fig. 5(a)). Furthermore, the values of \( m \) at which the total number of inner iterations is near its minimum are significantly smaller with preconditioning (around \( m = 10 \), see Fig. 5). Although J-D algorithm has smaller number of total inner iteration for very small \( m \), the corresponding outer iteration count is larger, which increases its cost.

We also considered for this example the ideal preconditioning with \( L_k \) chosen as the exact \( LDL^T \) factorization of \( C_k \). In this case, we use an initial vector with \( \| Ax_0 - \rho_0 B x_0 \| < 10^{-2} \) so that \( \rho_0 \) is sufficiently close to \( \lambda_1 \). We present the residual convergence history in Figure 6 for \( m = 1 \) (the steepest descent method), 2 and 4. The result confirms the quadratic convergence property for all
Figure 4: **Example 3** residual convergence history for \( m = 2, 4, 8, 16 \) (solid lines - Algorithm 4; dot lines - Jacobi-Davidson)

\[ \| \mathbf{r}_k \| \text{ vs. outer iterations} \]

\[ \| \mathbf{r}_k \| = 10^m \]

\( m \). We have also tested the case that uses \( L_k = L \) from the exact factorization of \( A - \lambda_0 B \) and in this case it converges in just one iteration, confirming Corollary 1.

The next example is for the standard eigenvalue problem \( (B = I) \) and the preconditioned algorithm implicitly transforms it to a pencil problem.

**Example 4:** \( A \) is the same matrix as in Example 2 \( (B = I) \). We use a constant \( L_k \) as obtained by the incomplete \( L D L^T \) decompositions of \( A - \lambda_0 B = A \) with no fill-in. We compare it with the Jacobi-Davidson algorithm with the same kind of preconditioner. We also consider the shift-and-invert spectral transformed Lanczos algorithm (i.e., applying the Lanczos to \( A^{-1} \)).

We give in Figure 7 the convergence history of the residual \( \| r_k \| \) of Algorithm 4 in solid lines from top down for \( m = 2, 4, 8, 16, 32 \) and that of the Jacobi-Davidson algorithm in dot lines for \( m = 2, 4, 8, 16, 32 \) with the corresponding marks. The residual for the spectral transformed Lanczos is given in dash-dot (with +) line. Figure 8 is the number of outer iterations and the total number of inner iterations vs. \( m \).

Again, preconditioning significantly accelerates convergence and our result compares very favorably with the Jacobi-Davidson method. An interesting point here is that Algorithm 4 with \( m = 4 \) based on incomplete factorization outperforms the shift-and-invert Lanczos algorithm. Although we do not suggest this is the case in general, it does underline the effectiveness of the preconditioned algorithm.
Figure 5: Example 3 Outer and total inner iterations vs. $m$ (+ - Algorithm 4; o - Jacobi-Davidson)

7 Concluding Remarks

We have presented an inverse free Krylov subspace method that is based on the classical Krylov subspace projection methods but incorporates preconditioning for efficient implementations. A convergence theory has been developed for the method and our preliminary tests of the preconditioned version demonstrate its competitiveness. Comparing with the existing methods, it has a relatively well understood theory and simple numerical behavior. We point out that the algorithm has a fixed cost per outer iteration, which makes it easy to implement.

For the future work, we will consider generalizations in three directions, namely, an efficient block version for computing several eigenvalues simultaneously, a strategy to compute interior eigenvalues and an algorithm for the general nonsymmetric problem.

A Appendix. Perturbation Bounds for Generalized Eigenvalue Problems

We present a perturbation theorem that is used in the proof of Lemma 1 but might be of general interest as well. In the following, $A, B$, and $E$ are all symmetric.

Theorem 3 Let $\lambda_1$ be the smallest eigenvalue of $(A, B)$ with $x$ a corresponding unit eigenvector and let $\mu_1$ be the smallest eigenvalue of $(A + E, B)$ with $u$ a corresponding unit eigenvector, where $B > 0$. Then we have

$$\frac{\lambda_{\min}(E)}{u^T B u} \leq \mu_1 - \lambda_1 \leq \frac{\lambda_{\max}(E)}{x^T B x}$$
where $\lambda_{\min}(E)$ and $\lambda_{\max}(E)$ denote the smallest and the largest eigenvalues of $E$ respectively.

**Proof** Using the minimax characterization, we have

$$
\mu_1 = \min_{z \neq 0} \frac{z^T (A + E) z}{z^T B z} \\
\leq \frac{x^T (A + E) x}{x^T B x} = \frac{x^T A x}{x^T B x} + \frac{x^T E x}{x^T B x} \\
\leq \lambda_1 + \frac{\lambda_{\max}(E)}{x^T B x}.
$$

Similarly,

$$
\lambda_1 \leq \mu_1 - \frac{u^T E u}{u^T B u} \\
\leq \mu_1 - \frac{\lambda_{\min}(E)}{u^T B u}
$$

which completes the proof.

We note that $1/x^T B x$ (or $1/u^T B u$ ) is the Wilkinson condition number for $\lambda_1$ (or $\mu_1$). These bounds therefore agree with the first order analytic expansion and will be sharper than traditional bounds based on $\|B\|$.
Figure 7: **Example 4** residual convergence history for $m = 2, 4, 8, 16, 32$ (solid lines - Algorithm 4; dot lines - Jacobi-Davidson, dash-dot - shift-and-invert Lanczos)

![Graph showing residual convergence history](image)

References


Figure 8: **Example 4** Outer and total inner iterations vs. \( m \) (+ - Algorithm 4; o - Jacobi-Davidson)


http://www-math.cudenver.edu/~aknyazev


23