Received: 20 July 2017 | Revised: 8 August 2018 | Accepted: 21 August 2018

DOI: 10.1002/nla.2213

RESEARCH ARTICLE

WILEY

A restarted Krylov method with inexact inversions

Achiya Dax

Israel Hydrological Service, P.O. Box
36118, Jerusalem 91360, Israel

Correspondence
Achiya Dax, Israel Hydrological Service,
P.O. Box 36118, Jerusalem 91360, Israel.
Email: dax20@water.gov.il

AMS Subject Classification: 65F15, 65F25, 65F50

Summary

In this paper, we present a new type of restarted Krylov method for calculating the smallest eigenvalues of a symmetric positive definite matrix G. The new framework avoids the Lanczos tridiagonalization process and the use of polynomial filtering. This simplifies the restarting mechanism and allows the introduction of several modifications. Convergence is assured by a monotonicity property that pushes the eigenvalues toward their limits. Another innovation is the use of inexact inversions of G to generate the Krylov matrices. In this approach, the inverse of G is approximated by using an iterative method to solve the related linear system. Numerical experiments illustrate the usefulness of the proposed approach.

KEYWORDS

inexact inversions, inner–outer iterations, monotonicity, restarted Krylov methods, smallest eigenvalues, symmetric matrices

1 | INTRODUCTION

In this paper, we present a new type of restarted Krylov method. Given a symmetric positive definite matrix $G \in \mathbb{R}^{n\times n}$, the method is aimed at calculating the k smallest eigenvalues of G. Being a Krylov method, it is best suited for handling large sparse matrices in which a matrix–vector product needs only $0(n)$ flops. Another underlying assumption is that k^2 is considerably smaller than n. The need for computing a few extreme eigenvalues of such a matrix arises in many applications (see the works of Calvetti et al.,[1] Saad,[2] and Watkins[3]).

The traditional restarted Krylov methods are classified into three types of restart: "explicit restart,"[2,4,5] "implicit restart,"[1,4,6] and "thick restart"[7,8] (see also the works of Saad,[2] Watkins,[3] Golub and Van Loan,[9] and Stewart[10]). When solving symmetric eigenvalue problems, these methods are carried out by the repeated use of the Lanczos tridiagonalization process and the use of polynomial filtering to determine the related starting vectors. This way, each iteration generates a new tridiagonal matrix and computes its eigensystem. The method proposed in this paper is based on a different framework, one that avoids these techniques. The basic iteration of the new method was recently proposed by this author in a prior work.[11] The driving force is a monotonicity property that pushes the estimated eigenvalues toward their limits. The rate of convergence depends on the quality of the Krylov matrices that we use. In this paper, we introduce a further modification: the use of inexact inversions to generate the Krylov matrices.

A simple way to construct a Krylov subspace that provides good estimates of the smallest eigenvalues is to build this subspace with G^{-1} instead of G. In practice, G^{-1} is almost never available in explicit form, and matrix–vector products of the form $G^{-1}\boldsymbol{b}$ are computed by solving the linear system $G\boldsymbol{x} = \boldsymbol{b}$. The solution of this system is often achieved by computing a factorization of G and using this factorization to solve the related linear systems. Another alternative is to use an iterative solver. In this way, the system $G\boldsymbol{x} = \boldsymbol{b}$ is solved by applying some iterative method. The choice of an iterative solver depends on the nature of the matrix at hand, and the computation of an accurate solution may require many iterations. It is common, therefore, to save time by stopping the iterative process before reaching an exact solution. The origin of this tactic, which is called **inexact inversion**, lies in the fields of Newton's method and Krylov subspace methods for solving large linear systems.[12–14] Recently, this technique has been found helpful in several iterative methods for solving large eigenvalue problems, for example, inverse iterations,[15–19] Rayleigh quotient iterations,[19–21] subspace iterations,[22,23] the Lanczos process,[24] explicitly restarted Krylov methods,[25] and implicitly restarted Arnoldi methods.[26–28] In these methods, the iterations of the linear system solver are called **inner** iterations, whereas the iterations of the eigenvalue method are called **outer** iterations. Consequently, this type of method is often called **inner–outer** iterative methods.[12,24,26] However, as we shall see in Sections 4 and 5, there are substantial differences between our implementation of inexact inversion and the way it is used in former implicitly restarted Arnoldi (or Lanczos) methods.

Other efficient methods for calculating small eigenvalues of large matrices are the Jacobi–Davidson method[29] and the locally optimal block preconditioned conjugate gradient method.[30,31] These methods are not using the Arnoldi–Lanczos process and are based on different ideas.

The main idea behind the new method is clarified by inspecting its basic iteration. In this paper, we concentrate on the smallest eigenvalues, but the algorithm can compute any cluster of k exterior eigenvalues. Let the eigenvalues of G be ordered to satisfy

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n > 0. \tag{1}$$

Then, the term “exterior eigenvalues” refers to a set of k largest eigenvalues, a set of k smallest eigenvalues, or any set that is combined from a number of the largest eigenvalues plus a number of the smallest ones. Other names for such eigenvalues are “peripheral eigenvalues“ and “extreme eigenvalues.”

Note that although the above definitions refer to clusters of eigenvalues, the algorithm is carried out by computing the corresponding k eigenvectors of G. The subspace that is spanned by these eigenvectors is called the **target space**.

The basic iteration

The qth iteration, $q = 0, 1, 2, \ldots$, is composed of the following five steps. The first step starts with a matrix $V_q \in \mathbb{R}^{n\times k}$ that contains “old” information on the target space, a matrix $Y_q \in \mathbb{R}^{n\times \ell}$ that contains “new” information, and a matrix $X_q = [V_q, Y_q] \in \mathbb{R}^{n\times(k+\ell)}$ that includes all the known information. The matrix X_q has $p = k + \ell$ orthonormal columns. That is, we have

$$X_q^T X_q = I \in \mathbb{R}^{p\times p}.$$

(Typical values for ℓ lie between k to $2k$.)

Step 1: Eigenvalues extraction. First, compute the Rayleigh quotient matrix

$$S_q = X_q^T G X_q.$$

Then, compute the k smallest eigenvalues of S_q. The corresponding k eigenvectors of S_q are assembled into a matrix

$$U_q \in \mathbb{R}^{p\times k}, \quad U_q^T U_q = I \in \mathbb{R}^{k\times k},$$

which is used to compute the related matrix of Ritz vectors, that is,

$$V_{q+1} = X_q U_q.$$

Note that both X_q and U_q have orthonormal columns, and V_{q+1} inherits this property.

Step 2: Collecting new information. Compute a Krylov matrix $B_q \in \mathbb{R}^{n\times \ell}$ that contains new information on the target space.

Step 3: Orthogonalize the columns of B_q against the columns of V_{q+1}. There are several ways to achieve this task. In exact arithmetic, the resulting matrix, Z_q, satisfies the Gram–Schmidt formula

$$Z_q = B_q - V_{q+1}\left(V_{q+1}^T B_q\right).$$

Step 4: Build an orthonormal basis of Range(Z_q). Compute a matrix

$$Y_{q+1} \in \mathbb{R}^{n\times \ell}, \quad Y_{q+1}^T Y_{q+1} = I \in \mathbb{R}^{\ell\times \ell},$$

whose columns form an orthonormal basis of Range(Z_q). This can be done by a QR factorization of Z_q. (If rank(Z_q) is smaller than ℓ, then ℓ is temporarily reduced to be rank(Z_q).)

Step 5: Define X_{q+1} by the rule

$$X_{q+1} = [V_{q+1}, Y_{q+1}],$$

which ensures that

$$X_{q+1}^T X_{q+1} = I \in \mathbb{R}^{p\times p}.$$

At this point, we are not concerned with efficiency issues, and the above description is mainly aimed to clarify the purpose of each step. Hence, there might be better ways to carry out the basic iteration. Note also that the basic iteration is easily modified to provide the k largest eigenvalues of G.

The plan of the paper is as follows. The monotonicity property that motivates the new method is established in the next section. Let $\lambda_{n+1-j}^{(q)}$, $j = 1, \dots, k$, denote the Ritz values that are computed at Step 1 of the qth iteration. Then, it is shown that each iteration gives a better approximation of the target cluster. Moreover, for each $j, j = 1, \dots, k$, the sequence $\lambda_{n+1-j}^{(q)}$, $q = 1, 2, \dots$, proceeds monotonically toward the desired eigenvalue of G. The rate of convergence depends on the Krylov information matrix B_q. The basic Krylov scheme is presented in Section 3. It is based on a three-term recurrence relation that provides a good Krylov matrix of G. Yet, the theory of Krylov subspaces tells us that a slow rate of convergence is expected whenever the smallest eigenvalues of G are clustered together. This difficulty is resolved in Section 4, in which we discuss the use of exact inversions and inexact inversions. The paper ends with numerical experiments that illustrate the usefulness of the proposed methods.

2 | THE MONOTONICITY PROPERTY

The monotonicity property is an important feature of the new iteration, whose proof is given in a prior work.[11] Yet, in order to make this paper self-contained, we provide the proof. We start by stating two well-known interlacing theorems.[32,33,34]

Theorem 1. (Cauchy interlace theorem)
Let $G \in \mathbb{R}^{n\times n}$ be a symmetric matrix with eigenvalues

$$\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_n. \tag{2}$$

Let the symmetric matrix $H \in \mathbb{R}^{k\times k}$ be obtained from G by deleting $n - k$ rows and the corresponding $n - k$ columns. Let

$$\eta_1 \ge \eta_2 \ge \cdots \ge \eta_k \tag{3}$$

denote the eigenvalues of H. Then, we have

$$\lambda_j \ge \eta_j \ \ for \ \ j = 1, \dots, k \tag{4}$$

and

$$\eta_{k+1-i} \ge \lambda_{n+1-i} \ \ for \ \ i = 1, \dots, k. \tag{5}$$

In particular, for $k = n - 1$, we have the interlacing relations

$$\lambda_1 \ge \eta_1 \ge \lambda_2 \ge \eta_2 \ge \lambda_3 \ge \cdots \ge \lambda_{n-1} \ge \eta_{n-1} \ge \lambda_n. \tag{6}$$

Corollary 1. (Poincaré separation theorem)
Let the matrix $V \in \mathbb{R}^{n\times k}$ have k orthonormal columns. That is, $V^TV = I \in \mathbb{R}^{k\times k}$. Let the matrix $H = V^TGV$ have the eigenvalues (3). Then, the eigenvalues of H and G satisfy (4) and (5).

Let us turn now to consider the qth iteration of the new method, $q = 1, 2, 3, \dots$. Assume first that the algorithm is aimed at computing the k largest eigenvalues of G, that is,

$$\{\lambda_1, \lambda_2, \dots, \lambda_k\},$$

and let the eigenvalues of the matrix

$$S_q = X_q^T G X_q = [V_q, Y_q]^T G [V_q, Y_q]$$

be denoted as

$$\lambda_1^{(q)} \ge \lambda_2^{(q)} \ge \cdots \ge \lambda_k^{(q)} \ge \cdots \ge \lambda_p^{(q)}.$$

Then, the Ritz values that are computed at Step 1 are

$$\lambda_1^{(q)} \ge \lambda_2^{(q)} \ge \cdots \ge \lambda_k^{(q)},$$

and these values are the eigenvalues of the matrix

$$V_{q+1}^T G V_{q+1}.$$

Similarly,

$$\lambda_1^{(q-1)} \geq \lambda_2^{(q-1)} \geq \cdots \geq \lambda_k^{(q-1)}$$

are the eigenvalues of the matrix

$$V_q^T G V_q.$$

Therefore, since the columns of V_q are the first k columns of X_q, we have

$$\lambda_j^{(q)} \geq \lambda_j^{(q-1)} \quad \text{for} \quad j = 1, \ldots, k.$$

On the other hand, from Corollary 1, we obtain that

$$\lambda_j \geq \lambda_j^{(q)} \quad \text{for} \quad j = 1, \ldots, k.$$

Hence, by combining these relations, we see that

$$\lambda_j \geq \lambda_j^{(q)} \geq \lambda_j^{(q-1)} \tag{7}$$

for $j = 1, \ldots, k$ and $q = 1, 2, 3, \ldots$.

Next, we assume that the algorithm is aimed at computing the k smallest eigenvalues of G, that is,

$$\{\lambda_{n+1-k}, \ldots, \lambda_{n-1}, \lambda_n\}.$$

Then, similar arguments show that

$$\lambda_{n+1-i}^{(q-1)} \geq \lambda_{n+1-i}^{(q)} \geq \lambda_{n+1-i} \tag{8}$$

for $i = 1, \ldots, k$ and $q = 1, 2, 3, \ldots$.

3 | THE BASIC KRYLOV MATRIX

The basic Krylov information matrix has the form

$$B_q = [\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_\ell] \in \mathbb{R}^{n \times \ell}, \tag{9}$$

where the sequence $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots$ is initialized by the starting vector $\boldsymbol{b}_0$. The ability of a Krylov subspace to approximate exterior eigenpairs is characterized by the Kaniel–Paige–Saad bounds (see, e.g., the works of Golub and Van Loan,[9(pp.552–554)] Parlett,[33(pp.242–247)] Saad,[2(pp.147–151)] and Stewart[10(pp.272–274)] as well as the references therein). One consequence of these bounds regards the angle between $\boldsymbol{b}_1$ and the target subspace: The smaller the angle, the better approximation we get. This suggests that $\boldsymbol{b}_0$ should be defined as the sum of the current Ritz vectors. That is, we have

$$\boldsymbol{b}_0 = V_{q+1}\mathbf{e},$$

where $\mathbf{e} = (1, 1, \ldots, 1)^T \in \mathbb{R}^k$ is a vector of ones. Note that there is no point in setting $\boldsymbol{b}_1 = V_{q+1}\mathbf{e}$, since in the next step, B_q is orthogonalized against V_{q+1}.

The proof of the Kaniel–Paige–Saad bounds relies on the properties of Chebyshev polynomials, whereas the building of these polynomials is carried out by using a three-term recurrence relation (see, e.g., Saad[2] and Parlett[33]). This observation suggests that in order to achieve these bounds, the algorithm for generating our Krylov sequence should use a "similar" three-term recurrence relation. Indeed, this feature is one of the reasons that make the Lanczos recurrence so successful (see Saad[2(p.138)]). Below, we describe an alternative three-term recurrence relation, which is based on the modified Gram–Schmidt (MGS) orthogonalization process.

Let $\boldsymbol{r} \in \mathbb{R}^n$ be a given vector, and let $\boldsymbol{q} \in \mathbb{R}^n$ be a unit length vector. That is, $\|\boldsymbol{q}\|_2 = 1$, where $\|\cdot\|_2$ denotes the Euclidean vector norm. Then, the statement "orthogonalize $\boldsymbol{r}$ against $\boldsymbol{q}$" is carried out by replacing $\boldsymbol{r}$ with $\boldsymbol{r} - (\boldsymbol{r}^T\boldsymbol{q})\boldsymbol{q}$. Similarly, the statement "normalize $\boldsymbol{r}$" is carried out by replacing $\boldsymbol{r}$ with $\boldsymbol{r}/\|\boldsymbol{r}\|_2$. With these conventions at hand, the construction of the vectors $\boldsymbol{b}_0, \boldsymbol{b}_1, \ldots, \boldsymbol{b}_\ell$ is carried out as follows.

The preparations part

1. Compute the starting vector:

$$\boldsymbol{b}_0 = V_{q+1}\mathbf{e}/\|V_{q+1}\mathbf{e}\|_2. \tag{10}$$

2. Compute $\boldsymbol{b}_1$: Set $\boldsymbol{b}_1 = G\boldsymbol{b}_0$.
   Orthogonalize $\boldsymbol{b}_1$ against $\boldsymbol{b}_0$.
   Normalize $\boldsymbol{b}_1$.
3. Compute $\boldsymbol{b}_2$: Set $\boldsymbol{b}_2 = G\boldsymbol{b}_1$.
   Orthogonalize $\boldsymbol{b}_2$ against $\boldsymbol{b}_0$.
   Orthogonalize $\boldsymbol{b}_2$ against $\boldsymbol{b}_1$.
   Normalize $\boldsymbol{b}_2$.

The iterative part
For $j = 3, \ldots, \ell$, compute $\boldsymbol{b}_j$ as follows.

1. Set $\boldsymbol{b}_j = G\boldsymbol{b}_{j-1}$.
2. Orthogonalize $\boldsymbol{b}_j$ against $\boldsymbol{b}_{j-2}$.
3. Orthogonalize $\boldsymbol{b}_j$ against $\boldsymbol{b}_{j-1}$.
4. Reorthogonalization: For $i = 1, \ldots, j-1$, orthogonalize $\boldsymbol{b}_j$ against $\boldsymbol{b}_i$.
5. Normalize $\boldsymbol{b}_j$.

The reorthogonalization step is aimed at ensuring that the numerical rank of B_q will stay close to ℓ. Yet, for small values of ℓ, it is not quite essential.

Note that although $\boldsymbol{b}_0$ is defined in an “explicit” way, there is a major difference between our method and former explicitly restarted Krylov methods. That is, in our method, the Krylov matrix B_q is orthogonalized against V_{q+1}, and the resulting matrix, Z_q, is used to construct an orthogonal extension of V_{q+1}. This important ingredient is missing in the former methods.

4 | COMPUTING THE SMALLEST EIGENVALUES: EXACT INVERSIONS VERSUS INEXACT INVERSIONS

The basic Krylov matrix that is defined in the previous section is using matrix–vector products of the form

$$\boldsymbol{b}_j = G\boldsymbol{b}_{j-1}, \quad j = 1, 2, \ldots, \ell, \tag{11}$$

and the columns of this matrix are aimed at providing a basis of the Krylov subspace that is spanned by the vectors

$$G\boldsymbol{b}_0, G^2\boldsymbol{b}_0, \ldots, G^{\ell}\boldsymbol{b}_0.$$

The methods proposed in this section are aimed at computing the smallest eigenvalues of G. In this context, the Kaniel–Paige–Saad bounds anticipate the following difficulty. If the eigenvalues of G converge toward zero, then a Krylov subspace that is generated by G does not provide good estimates of the smallest eigenpairs, which leads to a slow rate of convergence. Yet, this difficulty can be resolved by building the Krylov subspace with G^{-1} instead of G, since the inverse matrix has the same eigenvectors as G with the eigenvalues $1/\lambda_i,\ i = 1, \ldots, n$. The implementation of this idea is carried out by replacing (11) with

$$\boldsymbol{b}_j = G^{-1}\boldsymbol{b}_{j-1}, \ j = 1, 2, \ldots, \ell. \tag{12}$$

All the other operations in the construction of B_q (the choice of $\boldsymbol{b}_0$, orthogonalizations, and normalizations) remain unchanged. In this way, the columns of B_q provide a basis for the Krylov subspace that is spanned by the vectors

$$G^{-1}\boldsymbol{b}_0, \ \ G^{-2}\boldsymbol{b}_0, \ldots, G^{-\ell}\boldsymbol{b}_0.$$

In practice, however, there are many cases in which G^{-1} is not available in explicit form. Thus, usually, $\boldsymbol{b}_j$ is obtained from $\boldsymbol{b}_{j-1}$ by solving the related system of linear equations, that is,

$$G\boldsymbol{x} = \boldsymbol{b}_{j-1}. \tag{13}$$

Basically, there are two types of methods for solving large sparse linear systems: direct methods and iterative methods. Below, we offer a few words on each option.

In direct methods, we compute a factorization of G and use this factorization to solve the related linear systems (see, e.g., the works of Stewart[35], Dongarra et al.[36] and Duff et al.[37]). The factorization is done only once, before starting the restarted Krylov method. This feature turns to be an important advantage of the direct approach.

Note also that the basic iteration of the restarted Krylov method remains unchanged. The only changes occur in the computation of the information matrix, B_q, by replacing (11) with a direct solution of (13). Thus, in particular, in Step 1 of the basic iteration, we compute the Ritz values and Ritz vectors of G. This feature distinguishes our method from former restarted Krylov methods: Recall that each iteration of these methods achieves a new Lanczos tridiagonalization. Consequently, the replacement of G by G^{-1} results in a tridiagonal matrix, T, whose eigenpairs provide Ritz pairs of G^{-1}.

The second option is to solve the linear systems by applying an iterative method. Indeed, there is a plethora of such methods and several textbooks that discuss this topic (see, e.g., the works[9,36,38–45] and the references therein). Yet, the structure of inner–outer eigenvalue methods invites special methods.[16,25–27,46] Once we choose a method, we need to specify a proper stopping condition. This can be done in the following way. Assume for simplicity that the iterative algorithm is aimed at solving the linear system

$$G\boldsymbol{x} = \boldsymbol{b}, \tag{14}$$

where $\boldsymbol{b}$ is a given unit vector. Let $\boldsymbol{x}_i$ denote the current solution at the end of the ith iteration, $i = 1, 2, \dots$, and let $\boldsymbol{r}_i = \boldsymbol{b} - G\boldsymbol{x}_i$ denote the related residual vector. Then, since $\|\boldsymbol{b}\|_2 = 1$, a reasonable stopping condition is

$$\|\boldsymbol{r}_i\|_2 \leq \varepsilon, \tag{15}$$

where ε is a prescribed small positive constant.

If ε is sufficiently small or if the linear system is solved by a direct method, then we say that we have **exact inversion**. Otherwise, when the linear system is solved by an iterative method and ε is not very small, we say that we have **inexact inversion**. In this case, the computed solution of (14) approximates the vector $G^{-1}\boldsymbol{b}$, and the quality of the approximation depends on ε. The smaller ε is, the better approximation we get.

The use of inexact inversions emphasizes the differences between our method and restarted Krylov methods that are based on Lanczos tridiagonalization. In these methods, each outer iteration generates a tridiagonal matrix, $\tilde{T}$, whose Ritz pairs approximate those of G^{-1}. Consequently, the accuracy of the computed solutions of the linear systems determines the accuracy of the computed Ritz pairs.[17,19,26,27] In contrast, the outer iterations in our method compute Ritz pairs of G, and this task is carried out by extracting the eigenpairs of S_q, the relevant Rayleigh quotient matrix. Moreover, the monotonicity property ensures improvement regardless of the quality of B_q as basis for the desired Krylov subspace. Indeed, the experiments in Section 7 show that with moderate values of ε, the effect of inexact inversions is similar to that of exact inversions.

5 | THE USE OF SHIFT-AND-INVERT

The former discussions concentrate on the smallest eigenvalues of a positive definite matrix. However, the positive definiteness assumption is not essential. It is only aimed at clarifying the basic ideas and simplifying the presentation. Using a shift-and-invert technique, the new method is easily modified to compute a cluster of k eigenvalues on the right side (or the left side) of the spectrum of any symmetric matrix. The shift, α, can be updated every outer iteration, using the current Ritz values.

The implementation of this technique requires a minor change in the basic iteration: In this case, $\mathbf{b}_j$ is obtained from $\mathbf{b}_{j-1}$ by solving

$$(G - \alpha I)\mathbf{x} = \mathbf{b}_{j-1} \tag{16}$$

instead of (13). As before, this can be done by either a direct method or an indirect one. The main point is that all the other parts of the basic iteration remain unchanged. Thus, in particular, in Step 1, the Ritz pairs are extracted from the matrix $S_q = X_q^T G X_q$. (Note, again, the difference between our method and the Lanczos approach, which computes Ritz pairs of $(G - \alpha I)^{-1}$.) The shift-and-invert technique can also be used to compute interior clusters of eigenvalues, but in this case, the monotonicity property is not guaranteed.

6 | THE INITIAL ORTHONORMAL MATRIX

To start the algorithm, we need to supply an "initial" orthonormal matrix, $X_0 \in \mathbb{R}^{n\times p}$, where $p > k$. In the basic scheme, this is done in the following way. Define $p = k + \ell$ and let the $n \times p$ matrix

$$B_0 = [\boldsymbol{b}_1, \boldsymbol{b}_2, \dots, \boldsymbol{b}_p] \tag{17}$$

be generated as in Section 3, using some arbitrary starting vector $\boldsymbol{b}_0$. Then, X_0 is obtained by computing an orthonormal basis of Range(B_0). Similar procedures are used in exact inversions and inexact inversions.

In our experiments, B_0 is initiated by the vector $\boldsymbol{b}_0 = \mathbf{e}/\|\mathbf{e}\|_2$, where $\mathbf{e} = (1, 1, \dots, 1)^T \in \mathbb{R}^n$. Yet, a random starting vector is equally good. In the next section, we shall see that in some cases, the above choice of X_0 is sufficient to provide accurate estimates of the desired eigenpairs.

7 | NUMERICAL EXPERIMENTS

The former sections introduce four types of restarted Krylov methods: the basic scheme, exact inversions, inexact inversions, and shift-and-invert. In this section, we describe some numerical experiments that illustrate the behavior of these methods. The first type of test matrices has the form

$$G = HDH, \tag{18}$$

where D is a diagonal matrix, that is,

$$D = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_n\} \in \mathbb{R}^{n\times n}, \tag{19}$$

and H is a random householder matrix, that is,

$$H = I - 2\boldsymbol{h}\boldsymbol{h}^T/\boldsymbol{h}^T\boldsymbol{h}, \tag{20}$$

where $\boldsymbol{h}$ is a random vector. The term "random vector" means that each entry of $\boldsymbol{h}$ is a random number from the interval $[-1, 1]$. The random number generator is of uniform distribution. Note that (18) generates a symmetric $n \times n$ matrix whose eigenvalues are the diagonal entries of D. The diagonal matrices that we have used are displayed in Table 1.

In the **basic scheme**, the Krylov information matrix, B_q, is generated as prescribed in Section 3. The other methods are carried out as outlined in Sections 4 and 5. In **exact inversions**, we have used the relation $G^{-1}\boldsymbol{b} = HD^{-1}H\boldsymbol{b}$. The **inexact inversion** method was implemented by applying the **conjugate gradient** method to solve the related linear systems. We have used a basic version of this method, one that is described in several textbooks (see, e.g., the works of Demmel,[39(pp.311–312)] Greenbaum,[40(p.35)] Trefethen and Bau,[42(p.294)] and van der Vorst[43(p.42)]). The starting point of the conjugate gradient algorithm is always the origin point, $\mathbf{0} \in \mathbb{R}^n$, and the stopping condition is (15). We have used two values of ε: $\varepsilon = 10^{-10}$ and $\varepsilon = 10^{-5}$ (see Tables 4 and 5, respectively). The rate of convergence of the conjugate gradient method slows down as the condition number of G increases (see, e.g., the works of Demmel,[39(pp.312–314)] Saad,[41(pp.203–205)]

TABLE 1 The test matrices

Matrix type	**Matrix eigenvalues** λ_j, $j = 1, \dots, n$
Chebyshev zeros in $(0, 1]$	$\lambda_j = \cos((\pi/2)(n - j)/n)$
Equispaced	$\lambda_j = j$
Equispaced roots	$\lambda_j = j^{1/2}$
Harmonic roots	$\lambda_j = (1/j)^{1/2}$
Harmonic	$\lambda_j = 1/j$
Harmonic powers	$\lambda_j = (1/j)^{3/2}$
Harmonic squares	$\lambda_j = (1/j)^2$
Poisson	$\lambda_j = 2[1 - \cos(j\pi/(n + 1))]$

TABLE 2 Computing the $k = 12$ smallest eigenvalues with the basic scheme

Matrix type	**Number of iterations**					
	$n = 1{,}000$	$n = 2{,}000$	$n = 3{,}000$	$n = 10{,}000$	$n = 20{,}000$	$n = 30{,}000$
Chebyshev zeros	5	9	11	25	48	62
Equispaced	7	12	15	36	66	83
Equispaced roots	3	3	4	5	7	8
Harmonic roots	53	155	235	1322	4094	> 9999
Harmonic	175	667	1357	>9999		
Harmonic powers	705	3226	9892	>9999		
Harmonic squares	2743	>9999				
Poisson	54	201	443	4274	>9999	

TABLE 3 Computing the $k = 12$ smallest eigenvalues with exact inversions

Matrix type	**Number of iterations**					
	$n = 1{,}000$	$n = 2{,}000$	$n = 3{,}000$	$n = 10{,}000$	$n = 20{,}000$	$n = 30{,}000$
Chebyshev	0	0	0	0	0	0
Equispaced	0	0	0	0	0	0
Equispaced roots	0	0	0	0	0	0
Harmonic roots	9	15	24	49	98	122
Harmonic	6	9	12	24	43	52
Harmonic powers	4	7	8	15	22	28
Harmonic squares	3	5	6	10	13	15
Poisson	0	0	0	0	0	0

TABLE 4 Computing the $k = 12$ smallest eigenvalues with inexact inversions, $\varepsilon = 10^{-10}$

Matrix type	**Number of iterations**					
	$n = 1{,}000$	$n = 2{,}000$	$n = 3{,}000$	$n = 10{,}000$	$n = 20{,}000$	$n = 30{,}000$
Chebyshev zeros	0	0	0	0	0	0
Equispaced	0	0	0	0	0	0
Equispaced roots	0	0	0	0	0	0
Harmonic roots	10	17	22	53	95	128
Harmonic	7	9	12	26	39	55
Harmonic powers	4	7	8	15	24	28
Harmonic squares	3	5	6	10	13	16
Poisson	0	0	0	0	0	0

TABLE 5 Computing the $k = 12$ smallest eigenvalues with inexact inversions, $\varepsilon = 10^{-5}$

Matrix type	**Number of iterations**					
	$n = 1{,}000$	$n = 2{,}000$	$n = 3{,}000$	$n = 10{,}000$	$n = 20{,}000$	$n = 30{,}000$
Chebyshev zeros	1	0	0	0	0	0
Equispaced	1	0	0	0	0	0
Equispaced roots	1	1	2	1	1	1
Harmonic roots	17	27	32	62	104	127
Harmonic	11	13	14	27	49	56
Harmonic powers	6	7	9	16	22	31
Harmonic squares	4	5	7	12	14	16
Poisson	0	0	0	0	0	0

and Trefethen and Bau[42(pp.299–300)]). A common remedy for this difficulty is the use of a preconditioning technique, but we have not used this option (see the next section).

In our experiments, $\boldsymbol{\ell} = \boldsymbol{k} + \mathbf{40}$, where k denotes the number of desired eigenvalues. This ad hoc choice ensures that ℓ increases with k, but as k exceeds 40, the value of ℓ remains smaller than $2k$. The figures in Tables 2–5 provide the number of iterations (the number of restarts) that are needed to achieve the following accuracy criterion:

$$\left(\sum_{i=1}^{k}\left|\lambda_{n+1-i}^{(q)} - \lambda_{n+1-i}\right|\right)/(k\lambda_1) \leq 10^{-14}. \tag{21}$$

As before, the eigenvalues of G are assumed to satisfy (1),

$$\lambda_{n+1-k} \geq \cdots \geq \lambda_n \tag{22}$$

are the k smallest eigenvalues of G, and

$$\lambda_{n+1-k}^{(q)} \geq \cdots \geq \lambda_n^{(q)} \tag{23}$$

denote the computed values of these eigenvalues at the qth iteration.

The ability of the basic scheme to compute small eigenvalues is illustrated in Table 2. If the small eigenvalues are well separated, then the basic Krylov matrix is able to gain valuable information on the target space, and the restarted Krylov method terminates within a small number of iterations. This feature is seen in matrices like "Chebyshev zeros," "Equispaced," and "Equispaced roots." However, as implied by the Kaniel–Paige–Saad bounds, if the small eigenvalues cluster together, then the basic Krylov matrix is unable to provide useful information, which leads to a slow rate of convergence. This difficulty is seen in the "Harmonic" matrices.

The use of "exact inversions" enables us to overcome the above difficulty. Since G is positive definite, the smallest eigenvalues of G correspond to the largest eigenvalues of G^{-1}. If the latter eigenvalues are well separated, then the Krylov matrix that is built by using exact inversions contains valuable information on the target space, which leads to a fast rate of convergence. This property is demonstrated in Table 3. Note that in some cases, the initial Krylov matrix is sufficient to provide the required accuracy.

The usefulness of inexact inversions is illustrated in Tables 4 and 5. As expected, if ε is sufficiently small, then the behavior of this method resembles that of exact inversions. Indeed, a comparison of Table 4 with Table 3 shows that both methods require nearly the same number of iterations. A similar behavior is seen in Table 5, in which we have used a larger value of ε.

The second part of our experiments provides timing results that compare the new approach with MATLAB's "eigs" function. Tables 6 and 7 describe experiments with **"Sparse Random"** matrices that are defined as follows. Here, we have

$$G = A^T A \in \mathbb{R}^{n\times n}, \tag{24}$$

where A is an $m \times n$ matrix and $m = 8n$. Each column of A has only eight nonzero entries. The values of these entries are random numbers from the interval $[-1, 1]$, whereas the row indices of the nonzero entries are random integers between 1 and m. As before, the random number generator is of uniform distribution. Note that the diagonal entries of G are strictly positive. In addition, G has nearly $8n$ nonzero entries with random locations outside the main diagonal. Consequently, the nonzeros/zeros ratio is close to $9/n$, and G is stored as a sparse matrix. The dimension, n, of the tested matrices is quite large, which prevents us from achieving "exact inversion." Another feature that resembles practical situations is that the eigenvalues of G are not known in advance. Hence, it is not possible to use (21) as a stopping condition. Instead, the iterative process is terminated as soon as it satisfies

$$\left(\sum_{j=1}^{k}\left\|G\mathbf{v}_j - \left(\mathbf{v}_j^T G\mathbf{v}_j\right)\mathbf{v}_j\right\|_2\right)/k \leq 10^{-12}, \tag{25}$$

where $\mathbf{v}_j$, $j = 1, \ldots, k$, are the current Ritz vectors, which are mutually orthogonal unit vectors. The justification for this criterion can be found in the work of Parlett[33(p.69)] or Saad[2(p.61)]. (We have done some experiments to fix a suitable right-hand side for this bound. The accuracy of (25) is similar to (21) and close to that of "eigs.") Table 6 demonstrates how the computation time increases with n, whereas Table 7 explores the effect of k, the number of computed eigenpairs. The inexact inversion method applies the conjugate gradient method with $\varepsilon = 10^{-10}$. The results in Tables 6 and 7 clearly illustrate that the new approach competes favorably with MATLAB's "eigs" function.

The experiments in Table 8 were carried out on a tridiagonal Toeplitz matrix, $T = (t_{ij}) \in \mathbb{R}^{n\times n}$, that has the following structure: $t_{ii} = 3$ for $i = 1, \ldots, n$, $t_{i,i+1} = t_{i+1,i} = 1$ for $i = 1, \ldots, n-1$, and $t_{ij} = 0$ otherwise. The eigenvalues of T are known to satisfy

$$\lambda_j = 3 + 2\cos j\pi/(n+1), \quad j = 1, \cdots, n. \tag{26}$$

TABLE 6 Computing the $k = 40$ smallest eigenvalues of a sparse random matrix

Matrix dimension n	Computation time (in seconds)		
	eigs	Basic	Inexact
10,000	8.1	12.8	13.9
20,000	30.5	33.8	35.0
30,000	75.3	47.4	36.9
35,000	115.5	106.0	73.4
40,000	318.8	103.5	53.9
42,000	369.1	64.8	50.1
45,000	735.4	84.7	48.7
48,000	1656.7	107.4	95.9
50,000		167.7	95.2
100,000		186.2	122.3
200,000		482.3	311.4

TABLE 7 Computing the k smallest eigenvalues of a sparse random matrix: the effect of k on the computation time, $n = 40,000$

Number of eigenpairs k	Computation time (in seconds)		
	eigs	Basic	Inexact
5	381.9	4.75	3.96
10	306.2	20.6	15.7
20	250.9	23.8	24.8
30	268.5	59.3	35.4
40	318.8	103.4	53.9
50	346.6	129.2	115.1
60	360.2	136.9	106.0
70	340.2	240.5	167.7
80	398.3	254.2	158.8

That is, the smallest eigenvalues are clustered near 1, which invites the use of shift-and-invert with $\alpha = 1$. Furthermore, since T is tridiagonal, the shift-and-invert scheme is using “exact inversions,” which are based on the LU factorization of $T - I$. (Recall that the LU factorization is computed only once; the Krylov subspaces that are built in this way are those of $(T - I)^{-1}$, whereas the Ritz pairs are computed from T.) The usefulness of this approach is illustrated in Table 8. Since the eigenvalues of T are known, the new algorithms terminate by using (21). The term “fail“ means that “eigs” ends with a warning that the requested eigenvalues did not converge.

Table 9 describes experiments with $n \times n$ pentadiagonal matrices of the form

$$P = H^T H, \tag{27}$$

where here $H = (h_{ij}) \in \mathbb{R}^{n \times n}$ is a Hankle matrix. The entries of H are defined as follows: $h_{n+1-j,j-1} = 0.5$ for $j = 2, \dots, n$, $h_{n+1-j,j} = 2$ for $j = 1, \dots, n$, and $h_{n+2-j,j} = 2$ for $j = 2, \dots, n$. All the other entries of H are zeros. The eigenvalues of P are not known in advance, and therefore, the new methods terminate by (25). However, the experiments with “eigs“ and “Basic” show that P has one zero eigenvalue, whereas the other small eigenvalues are clustered near 0.25. This suggests the use of a shift-and-invert scheme with $\alpha = 0.2$ and LU factorization of $P - 0.2I$. (As before, the LU factorization is computed only once; the Krylov subspaces are built from $(P - 0.2I)^{-1}$, but the Ritz pairs are extracted from P.) The usefulness of this approach is illustrated in Table 9.

Finally, we note that the timing results were measured by applying MATLAB's “tic“ and “toc” functions. This technique has a small deficiency: Repeated runs of the same program on the same data may give different results. The difference is negligible for large running times (larger than 10 seconds say). Yet, it can be meaningful for small running times (smaller than 0.5 seconds say). In such cases, the figures in Tables 6–9 are median values that are obtained from a number of runs.

TABLE 8 Computing the $k = 12$ smallest eigenvalues of a tridiagonal Toeplitz matrix

Matrix dimension n	Computation time (in seconds)		
	eigs	Basic	Shift-and-invert
1,000	0.199	0.875	0.0311
1,500	0.276	1.57	0.0431
1,800	0.348	2.46	0.0429
1,900	0.378	3.91	0.0455
2,000	fail	2.79	0.0489
3,000	fail	13.3	0.0635
4,000	fail	38.5	0.0793
5,000	fail	79.9	0.0804
8,000	fail	168.3	0.0969
10,000		431.0	0.134
20,000			0.202
50,000			0.451
100,000			0.935
200,000			3.47

TABLE 9 Computing the $k = 12$ smallest eigenvalues of a pentadiagonal matrix

Matrix dimension n	Computation time (in seconds)		
	eigs	Basic	Shift-and-invert
500	0.170	2.75	0.062
1,000	0.215	11.1	0.185
2,000	0.335	51.6	0.404
3,000	0.466	215.9	0.956
4,000	fail	465.2	1.81
5,000	fail	821.6	2.86
10,000	fail		15.5
20,000	fail		71.3
30,000			225.9

8 | CONCLUDING REMARKS

The simplicity of the new restarting procedure allows the introduction of useful modifications in the building of B_q. One innovation is the rule for determining the starting vector. This rule is much simpler and more intuitive than rules that are based on polynomial filtering. Another modification is a three-term recurrence relation that is based on the modified Gram–Schmidt orthogonalization. In this paper, we focus the attention on the use of exact and inexact inversions. The new method implements this idea in a simple robust way, which differs substantially from former methods. The experiments that we have done illustrate the viability of these ideas. We see that for moderate values of ε, such as $\varepsilon = 10^{-10}$ or $\varepsilon = 10^{-5}$, the effect of inexact inversions is similar to that of exact inversions. In some cases, the new method computes the required eigenvalues within a remarkably small number of restarts.

The new approach opens the way for the effective use of shift-and-invert, which is quite different from that of Lanczos methods. Using this technique, the new method is easily modified to compute any cluster of k eigenvalues of any symmetric matrix. The experiments that we have done are quite encouraging.

The current presentation of the new method is aimed at clarifying the main ideas and keeps the basic iteration as simple as possible. However, it is important to note that the basic algorithm can be improved in several ways. As in other subspace methods, the modifications may include effective orthogonalization schemes, locking, optimizing ("tuning") the values of ℓ and ε, and the use of block Krylov methods. The adaptation of the new method to compute small singular values is considered in the work of Dax.[47]

The efficiency of an "inner–outer" iterative eigenvalue method depends on the linear solver. In many cases, one uses a standard sparse solver (like CG, MINRES, or GMRES) and equip the solver with a suitable preconditioner (such as incomplete Cholesky factorization). In some cases, the preconditioner is modified to take advantage of the fact that the right-hand side vector, $\boldsymbol{b}$, is close to a certain eigenspace of G. Another improvement regards the use of variable stopping criteria. That is, the value of ϵ in (15) changes during the inner iterations or during the outer iterations. For detailed discussions of these issues, see, for example, the works of Golub and Ye,[25] Freitag,[26] and Freitag and Spence.[16,27,45] Yet, in these issues, there is no difference between our algorithm and other inner–outer methods, and any advanced solver can be adapted. Note also that in inner–outer methods, most of the computation time is often spent on the inner iterations. This feature increases the appeal of the new approach, as the extra time that it pays for deserting the Lanczos tridiagonalization process becomes negligible.

ORCID

Achiya Dax http://orcid.org/0000-0002-6403-1020

REFERENCES

1. Calvetti D, Reichel L, Sorensen DC. An implicitly restarted Lanczos method for large symmetric eigenvalue problems. Elec Trans Numer Anal. 1994;2:1–21.
2. Saad Y. Numerical methods for large eigenvalue problems: Revised edition. Philadelphia, PA: Society for Industrial and Applied Mathematics; 2011.
3. Watkins DS. The matrix eigenvalue problem: GR and Krylov subspace methods. Philadelphia, PA: Society for Industrial and Applied Mathematics; 2007.
4. Bai Z, Demmel J, Dongarra J, Ruhe A, van der Vorst H. Templates for the solution of algebraic eigenvalue problems: A practical guide. Philadelphia, PA: Society for Industrial and Applied Mathematics; 1999.
5. Morgan RB. On restarting the Arnoldi method for large non-symmetric eigenvalues problems. Math Comput. 1996;65:1213–1230.
6. Sorensen DC. Implicit application of polynomial filters in a *k*-step Arnoldi method. SIAM J Matrix Anal Appl. 1992;13:357–385.
7. Wu K, Simon H. Thick-restarted Lanczos method for large symmetric eigenvalue problems. SIAM J Matrix Anal Appl. 2000;22:602–616.
8. Yamazaki I, Bai Z, Simon H, Wang L-W, Wu K. Adaptive projection subspace dimension for the thick-restart Lanczos method. ACM Trans Math Softw. 2010;37(3). Article No. 27.
9. Golub GH, Van Loan CF. Matrix computations. Fourth edition. Baltimore, MD: Johns Hopkins University Press; 2013.
10. Stewart GW. Matrix algorithms: Volume II: eigensystems. Philadelphia, PA: Society for Industrial and Applied Mathematics; 1998.
11. Dax A. A new type of restarted Krylov methods. Adv Linear Algebra Matrix Theory. 2017;7:18–28.
12. Giladi E, Golub GH, Keller JB. Inner and outer iterations for the Chebyshev algorithm. SIAM J Numer Anal. 1998;35:300–319.
13. Simoncini V, Szyld DB. Theory of inexact Krylov subspace methods and applications to scientific computing. SIAM J Sci Comput. 2003;25:454–477.
14. van den Eshof J, Sleijpen GLG. Inexact Krylov subspace methods for linear systems. SIAM J Matrix Anal Appl. 2004;26:125–153.
15. Berns-Müller J, Graham IG, Spence A. Inexact inverse iteration for symmetric matrices. Linear Algebra Appl. 2006;416:389–413.
16. Freitag MA, Spence A. A tuned preconditioner for inexact inverse iteration applied to Hermitian eigenvalue problems. IMA J Numer Anal. 2008;28:522–551.
17. Golub GH, Ye Q. Inexact inverse iteration for generalized eigenvalue problems. BIT Numer Math. 2000;40:671–684.
18. Lai Y-L, Lin K-Y, Lin W-W. An inexact inverse iteration for large sparse eigenvalue problems. Numer Linear Algebra Appl. 1997;4:425–437.
19. Smit P, Paardekooper MHC. The effects on inexact solvers in algorithms for symmetric eigenvalue problems. Lin Alg Appl. 1999;287:337–357.
20. Simoncini V, Eldén L. Inexact Rayleigh quotient-type methods for eigenvalue computations. BIT Numer Math. 2002;42:159–182.
21. Xue F, Elman HC. Convergence analysis of iterative solvers in inexact Rayleigh quotient iteration. SIAM J Matrix Anal Appl. 2009;31:877–899.
22. Xue F, Elman HC. Fast inexact subspace iteration for generalized eigenvalue problems with spectral transformation. Linear Algebra Appl. 2011;435:601–622.
23. Ye Q, Zhang P. Inexact inverse subspace iteration for generalized eigenvalue problems. Linear Algebra Appl. 2011;434:1697–1715.
24. Golub GH, Zhang Z, Zha H. Large sparse symmetric eigenvalue problems with homogeneous linear constraints: the Lanczos process with inner–outer iterations. Linear Algebra Appl. 2000;309:289–306.
25. Golub GH, Ye Q. An inverse free preconditioned Krylov subspace method for symmetric generalized eigenvalue problems. SIAM J Sci Comput. 2002;24:312–334.
26. Freitag MA. Inner-outer iterative methods for eigenvalue problems - convergence and preconditioning. PhD [dissertation]. Bath, UK: Department of Mathematical Sciences, University of Bath; 2007.

27. Freitag MA, Spence A. Shift-invert Arnoldi's method with preconditioned iterative solves. SIAM J Matrix Anal Appl. 2009;31:942–969.
28. Xue F, Elman HC. Fast inexact implicitly restarted Arnoldi method for generalized eigenvalue problems with spectral transformation. SIAM J Matrix Anal Appl. 2012;33:433–459.
29. Sleijpen GLG, Van der Vorst HA. A Jacobi–Davidson iteration method for linear eigenvalue problems. SIAM Review. 2000;42:267–293.
30. Knyazev AV. Toward the optimal preconditioned eigensolver: locally optimal block preconditioned conjugate gradient method. SIAM J Sci Comput. 2001;23:517–541.
31. Knyazev AV, Argentati ME, Lashuk I, Ovtchinnikov EE. Block locally optimal preconditioned eigenvalue solvers (BLOPEX) in Hypre and PETSc. SIAM J Sci Comput. 2007;29:2224–2239.
32. Horn RA, Johnson CR. Matrix analysis. Cambridge, UK: Cambridge University Press; 1985.
33. Parlett BN. The symmetric eigenvalue problem. Englewood Cliffs, NJ: Prentice-Hall; 1980.
34. Zhang F. Matrix theory: Basic results and techniques. New York, NY: Springer-Verlag New York; 1999.
35. Stewart GW. Matrix algorithms: Volume I: basic decompositions. Philadelphia, PA: Society for Industrial and Applied Mathematics; 2001.
36. Dongarra JJ, Duff IS, Sorensen DC, ven der Vorst HA. Numerical linear algebra for high-performance computers. Philadelphia, PA: Society for Industrial and Applied Mathematics; 1998.
37. Duff IS, Erisman AM, Reid JK. Direct methods for sparse matrices. London, UK: Oxford University Press; 1986.
38. Axelsson O. Iterative solution methods. Cambridge, UK: Cambridge University Press; 1994.
39. Demmel JW. Applied numerical linear algebra. Philadelphia, PA: Society for Industrial and Applied Mathematics; 1997.
40. Greenbaum A. Iterative methods for solving linear systems. Philadelphia, PA: Society for Industrial and Applied Mathematics; 1997.
41. Saad Y. Iterative methods for sparse linear systems: Second edition. Philadelphia, PA: Society for Industrial and Applied Mathematics; 2003.
42. Trefethen LN, Bau D III. Numerical linear algebra. Philadelphia, PA: Society for Industrial and Applied Mathematics; 1997.
43. van der Vorst HA. Iterative Krylov methods for large linear systems. Cambridge, UK: Cambridge University Press; 2003.
44. Varga RS. Matrix iterative analysis. Englewood Cliffs, NJ: Prentice-Hall; 1962.
45. Young DM. Iterative solution of large linear systems. Orlando, FL: Academic Press; 1971.
46. Freitag MA, Spence A. Rayleigh quotient iteration and simplified Jacobi-Davidson method with preconditioned iterative solves. Linear Algebra Appl. 2008;428:2049–2060.
47. Dax A. A new algorithm for the smallest singular triplets of a large matrix. Hydrological Service of Israel. 2018. Technical Report. In preparation.

How to cite this article: Dax A. A restarted Krylov method with inexact inversions. *Numer Linear Algebra Appl.* 2019;26:e2213. https://doi.org/10.1002/nla.2213