

A New Type of Restarted Krylov Methods

Achiya Dax

Hydrological Service, Jerusalem, Israel

Email: dax20@water.gov.il

How to cite this paper: Dax, A. (2017) A New Type of Restarted Krylov Methods. *Advances in Linear Algebra & Matrix Theory*, 7, 18-28.

<https://doi.org/10.4236/alamt.2017.71003>

Received: March 6, 2017

Accepted: March 28, 2017

Published: March 31, 2017

Copyright © 2017 by author and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

In this paper we present a new type of Restarted Krylov methods for calculating peripheral eigenvalues of symmetric matrices. 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. The Krylov matrices that we use lead to fast rate of convergence. Numerical experiments illustrate the usefulness of the proposed approach.

Keywords

Restarted Krylov Methods, Exterior Eigenvalues, Symmetric Matrices, Monotonicity, Starting Vectors

1. Introduction

In this paper we present a new type of Restarted Krylov methods. Given a symmetric matrix $G \in \mathbb{R}^{n \times n}$, the method is aimed at calculating a cluster of k exterior eigenvalues of G . Other names for such eigenvalues are “peripheral eigenvalues” and “extreme eigenvalues”. The method is best suited for handling large sparse matrices in which a matrix-vector product needs only $O(n)$ 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 [1]-[21].

The traditional restarted Krylov methods are classified into three types of restarts: “Explicit restart” [1], [9], [11], “Implicit restart” [1], [3], [12], and “Thick restart” [18], [19]. See also [7], [11], [14], [16]. When solving symmetric eigenvalue problems all these methods are carried out by 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 have recently been presented by this author in [4]. 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 matrix that we use. In this paper we introduce a modified scheme for generating the Krylov matrix. This leads to dramatic improvement in the rate of convergence, and turns the method into a powerful tool.

Let the eigenvalues of G be ordered to satisfy

$$|\hat{\lambda}_1| \geq |\hat{\lambda}_2| \geq \dots \geq |\hat{\lambda}_n| \geq 0 \quad (1.1)$$

or

$$\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \dots \geq \tilde{\lambda}_n. \quad (1.2)$$

Then the new algorithm is built to compute one of the following four types of **target clusters** that contain k extreme eigenvalues.

A dominant cluster

$$\{\hat{\lambda}_1, \dots, \hat{\lambda}_k\}. \quad (1.3)$$

A right-side cluster

$$\{\tilde{\lambda}_1, \dots, \tilde{\lambda}_k\}. \quad (1.4)$$

A left-side cluster

$$\{\tilde{\lambda}_{n+1-k}, \dots, \tilde{\lambda}_{n-1}, \tilde{\lambda}_n\}. \quad (1.5)$$

A **two-sides cluster** is a union of a right-side cluster and a left-side cluster. For example, $\{\tilde{\lambda}_1, \tilde{\lambda}_n\}$ or $\{\tilde{\lambda}_1, \tilde{\lambda}_2, \tilde{\lambda}_n\}$.

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 q th iteration, $q = 0, 1, 2, \dots$, 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

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

(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 k eigenpairs of S_q which correspond to the target cluster. (For example, if it is desired to compute a right-side cluster of G , then compute a right-side cluster 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,

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

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

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

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.

The plan of the paper is as follows. The monotonicity property that motivates the new method is established in the next section. Let $\lambda_j^{(q)}, j = 1, \dots, k$, denote the Ritz values which are computed at Step 1 of the q th 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 $\lambda_j^{(q)}, q = 1, 2, \dots$, proceeds monotonously toward the desired eigenvalue of G . The rate of convergence depends on the information matrix B_q . The method proposed in Section 3 is based on a three-term recurrence relation that leads to rapid convergence. A further improvement can be gained by Power acceleration, see Section 4. 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 [4]. Yet, in order to make this paper self-contained, we provide the proof. We start by stating two well-known interlacing theorems, e.g., [8], [10] and [20].

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

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n. \quad (2.1)$$

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

$$\eta_1 \geq \eta_2 \geq \dots \geq \eta_k \quad (2.2)$$

denote the eigenvalues of H . Then

$$\lambda_j \geq \eta_j \quad \text{for } j = 1, \dots, k, \quad (2.3)$$

and

$$\eta_{k+1-i} \geq \lambda_{n+1-i} \quad \text{for } i = 1, \dots, k. \quad (2.4)$$

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

$$\lambda_1 \geq \eta_1 \geq \lambda_2 \geq \eta_2 \geq \lambda_3 \geq \dots \geq \lambda_{n-1} \geq \eta_{n-1} \geq \lambda_n. \quad (2.5)$$

Corollary 2 (Poincarè separation theorem) Let the matrix $V \in \mathbb{R}^{n \times k}$ have k orthonormal columns. That is $V^T V = I \in \mathbb{R}^{k \times k}$. Let the matrix $H = V^T G V$ have the eigenvalues (2.2). Then the eigenvalues of H and G satisfy (2.3) and (2.4).

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

$$\{\tilde{\lambda}_1, \tilde{\lambda}_2, \dots, \tilde{\lambda}_k\},$$

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

$$\lambda_1^{(q)} \geq \lambda_2^{(q)} \geq \dots \geq \lambda_k^{(q)} \geq \dots \geq \lambda_p^{(q)}.$$

Then the Ritz values which are computed at Step 1 are

$$\lambda_1^{(q)} \geq \lambda_2^{(q)} \geq \dots \geq \lambda_k^{(q)},$$

and these values are the eigenvalues of the matrix

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

Similarly,

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

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 ,

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

On the other hand from Corollary 2 we obtain that

$$\tilde{\lambda}_j \geq \lambda_j^{(q)} \quad \text{for } j = 1, \dots, k.$$

Hence by combining these relations we see that

$$\tilde{\lambda}_j \geq \lambda_j^{(q)} \geq \lambda_j^{(q-1)} \tag{2.6}$$

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

The treatment of a left-side cluster is done in a similar way. Assume that the algorithm is aimed at computing a cluster of k left-side eigenvalues of G ,

$$\{\tilde{\lambda}_{n+1-k}, \dots, \tilde{\lambda}_{n-1}, \tilde{\lambda}_n\}.$$

Then similar arguments show that

$$\lambda_{p+1-i}^{(q-1)} \geq \lambda_{p+1-i}^{(q)} \geq \tilde{\lambda}_{n+1-i} \tag{2.7}$$

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

Recall that a two-sides cluster is the union of a right-side cluster and a left-side one. In this case the eigenvalues of S_q that correspond to the right-side satisfy (2.6) while eigenvalues of S_q that correspond to the left-side satisfy (2.7). A similar situation occurs in the computation of a dominant cluster, since a dominant cluster is either a right-side cluster, a left-side cluster, or a two-sides cluster.

3. The Basic Krylov Matrix

The basic Krylov information matrix has the form

$$B_q = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_\ell] \in \mathbb{R}^{n \times \ell}, \tag{3.1}$$

where the sequence $\mathbf{b}_1, \mathbf{b}_2, \dots$, is initialized by the starting vector \mathbf{b}_0 . The ability of a Krylov subspace to approximate a dominant subspace is characterized by the Kaniel-Paige-Saad bounds. See, for example, ([7], pp. 552-554), ([10], pp. 242-247), ([11], pp. 147-151), ([14], pp. 272-274), and the references therein. One consequence of these bounds regards the angle between \mathbf{b}_1 and the dominant subspace: The smaller the angle, the better approximation we get. This suggests that \mathbf{b}_0 should be defined as the sum of the current Ritz vectors. That is,

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

where $\mathbf{e} = (1, 1, \dots, 1)^T \in \mathbb{R}^k$ is a vector of ones. Note that there is no point in setting $\mathbf{b}_1 = V_{q+1} \mathbf{e}$, 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, while the building of these polynomials is carried out by using a three term recurrence relation, e.g. [10], [11]. 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 ([11], p. 138). Below we describe an alternative three term recurrence relation, which is based on the Modified Gram-Schmidt (MGS) orthogonalization process.

Let $\mathbf{r} \in \mathbb{R}^n$ be a given vector and let $\mathbf{q} \in \mathbb{R}^n$ be a unit length vector. That is $\|\mathbf{q}\|_2 = 1$ where $\|\cdot\|_2$ denotes the Euclidean vector norm. Then the statement “orthogonalize \mathbf{r} against \mathbf{q} ” is carried out by replacing \mathbf{r} with $\mathbf{r} - (\mathbf{r}^T \mathbf{q}) \mathbf{q}$.

Similarly, the statement “normalize \mathbf{r} ” is carried out by replacing \mathbf{r} with $\mathbf{r}/\|\mathbf{r}\|_2$. With these conventions at hand the construction of the vectors $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_\ell$, is carried out as follows.

The preparations part

a) Compute the starting vector:

$$\mathbf{b}_0 = V_{q+1} \mathbf{e} / \|V_{q+1} \mathbf{e}\|_2 \quad (3.2)$$

b) Compute \mathbf{b}_1 : Set $\mathbf{b}_1 = G\mathbf{b}_0$.

Orthogonalize \mathbf{b}_1 against \mathbf{b}_0 .

Normalize \mathbf{b}_1 .

c) Compute \mathbf{b}_2 : Set $\mathbf{b}_2 = G\mathbf{b}_1$.

Orthogonalize \mathbf{b}_2 against \mathbf{b}_0 .

Orthogonalize \mathbf{b}_2 against \mathbf{b}_1 .

Normalize \mathbf{b}_2 .

The iterative part

For $j = 3, \dots, \ell$, compute \mathbf{b}_j as follows:

a) Set $\mathbf{b}_j = G\mathbf{b}_{j-1}$.

b) Orthogonalize \mathbf{b}_j against \mathbf{b}_{j-2} .

c) Orthogonalize \mathbf{b}_j against \mathbf{b}_{j-1} .

d) Reorthogonalization: For $i = 1, \dots, j-1$, orthogonalize \mathbf{b}_j against \mathbf{b}_i .

e) Normalize \mathbf{b}_j .

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

4. The Power-Krylov Matrix

Assume for a moment that the algorithm is aimed at calculating a cluster of k dominant eigenvalues. Then the Kaniel-Paige-Saad bounds suggest that slow rate of convergence is expected when these eigenvalues are poorly separated from the other eigenvalues. Indeed, this difficulty is seen in **Table 2**, when the basic Krylov matrix is applied on problems like “Very slow geometric” and “Dense equispaced”. Let $\nu \geq 2$ be a small integer. Then the larger eigenvalues of the powered matrix, G^ν , are better separated than those of G . This suggests that a faster rate of convergence can be gained by replacing G with G^ν .

The implementation of this idea is carried out by introducing a small modification in the construction of B_q : Here the computation of \mathbf{b}_j , $j = 1, \dots, \ell$, starts with

$$\mathbf{b}_j = G^\nu \mathbf{b}_{j-1}. \quad (4.1)$$

(In our experiments $\nu = 4$.) It is important to stress that this is the only part of the algorithm that uses G^ν . All the other steps of the basic iteration remain unchanged. In particular, the Ritz values which are computed in Step 1 are those of G (not G^ν). Of course, in practice G^ν is never computed. Instead \mathbf{b}_j is computed by a sequence of ν matrix-vector multiplications.

The usefulness of the Power-Krylov approach depends on two factors: The

cost of a matrix-vector product and the distribution of the eigenvalues. As noted above, it is expected to reduce the number of iterations when the k largest eigenvalues of G are poorly separated from the rest of the spectrum. See **Table 3**. Another advantage of this approach is that ℓ is kept small. Note that although the computational effort per iteration increases, a smaller portion of time is spent on orthogonalizations and on the Rayleigh-Ritz procedure.

5. 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 $p > k$. This task can be done in the following way. Define $p = k + \ell$ and let $n \times p$ matrix

$$B_0 = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_p] \tag{5.1}$$

be generated as in Section 3, using some arbitrary starting vector \mathbf{b}_0 . Then X_0 is obtained by computing an orthonormal basis of $\text{Range}(B_0)$. A similar procedure is used in the Power-Krylov method.

In our experiments B_0 is initiated by the vector $\mathbf{b}_0 = \mathbf{e} / \|\mathbf{e}\|_2$ 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 the above choice of X_0 is often sufficient to provide accurate estimates of the desired eigenpairs.

6. Numerical Experiments

In this section we describe some experiments with the proposed methods. The test matrices have the form

$$G = \text{diag} \{ \lambda_1, \lambda_2, \dots, \lambda_n \} \in \mathbb{R}^{n \times n} \tag{6.1}$$

where

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0. \tag{6.2}$$

Recall that in Krylov methods there is no loss of generality in experimenting with diagonal matrices, e.g., ([6], p. 367). The diagonal matrices that we have used are displayed in **Table 1**. All the experiments were carried out with $\mathbf{n} = 12,000$. The experiments that we have done are aimed at computing a cluster of k dominant eigenvalues. The figures in **Table 2** and **Table 3** provide the number of iterations that were needed to satisfy the inequality

$$\left(\sum_{j=1}^k |\lambda_j - \lambda_j^{(q)}| \right) / (k |\lambda_1|) \leq 10^{-14}. \tag{6.3}$$

Thus, for example, from **Table 2** we see that only 4 iterations are needed when the algorithm computes $k = 200$ eigenvalues of the Equispaced test matrix.

The ability of the basic Krylov matrix to achieve accurate computation of the eigenvalues is illustrated in **Table 2**. We see that often the algorithm terminates within a remarkably small number of iterations. Observe that the method is highly successful in handling low-rank matrices, matrices which are nearly low-

Table 1. Types of test matrices, $n = 12000$.

Matrix type	Matrix eigenvalues $\lambda_j, j = 1, \dots, n$
Harmonic squares	$\lambda_j = (1/j)^2$
Harmonic	$\lambda_j = 1/j$
Harmonic roots	$\lambda_j = (1/j)^{1/2}$
Very fast geometric decay	$\lambda_j = (0.5)^j$
Fast geometric decay	$\lambda_j = (0.9)^j$
Geometric decay	$\lambda_j = (0.95)^j$
Moderate geometric decay	$\lambda_j = (0.99)^j$
Slow geometric decay	$\lambda_j = (0.999)^j$
Very slow geometric decay	$\lambda_j = (0.9999)^j$
Dense equispaced	$\lambda_j = n + 1 - j$
Equispaced	$\lambda_j = 1001 - j$ for $j = 1, \dots, 1000$ and $\lambda_j = 0$ for $j = 1001, \dots, n$
Low-Rank-100	$\lambda_j = 101 - j$ for $j = 1, \dots, 100$ and $\lambda_j = 0$ for $j = 101, \dots, n$
Low-Rank-50	$\lambda_j = 51 - j$ for $j = 1, \dots, 50$ and $\lambda_j = 0$ for $j = 51, \dots, n$
Low-Rank-10	$\lambda_j = n + 1 - j$ for $j = 1, \dots, 10$ and $\lambda_j = 0$ for $j = 11, \dots, n$
Harmonic Triples	$\lambda_{3j-2} = \lambda_{3j-1} = \lambda_{3j} = 1/j$ for $j = 1, \dots, n/3$
Multiple-Harmonic	$\lambda_j = 1$ for $j = 1, \dots, 10$ and $\lambda_j = 1/j$ for $j = 11, \dots, n$
Multiple-Geometric	$\lambda_j = 1$ for $j = 1, \dots, 10$ and $\lambda_j = (0.95)^j$ for $j = 11, \dots, n$
Equispaced-Geometric Gap	$\lambda_j = 101 - j$ for $j = 1, \dots, 100$ and $\lambda_j = (0.95)^j$ for $j = 101, \dots, n$

rank, like “Harmonic” or “Geometric”, and matrices with gap in the spectrum. In such matrices the initial orthonormal matrix, X_0 , is sufficient to provide accurate eigenvalues. Note also that the method has no difficulty in computing multiple eigenvalues.

As expected, a slower rate of convergence occurs when the dominant eigenvalues that we seek are poorly separated from the other eigenvalues. This situation is demonstrated in matrices like “Dense equispaced” or “Very slow geometric”. Yet, as **Table 3** shows, in such cases the Power-Krylov method leads to considerable reduction in the number of iterations.

7. Concluding Remarks

The new type of Restarted Krylov methods avoids the use of Lanczos algorithm. This simplifies the basic iteration, and clarifies the main ideas behind the

Table 2. Computing k dominant eigenvalues with the basic Krylov matrix, $\ell = k + 40$.

Matrix type	Number of iterations					
	$k = 6$	$k = 10$	$k = 20$	$k = 40$	$k = 100$	$k = 200$
Harmonic squares	0	0	0	0	0	0
Harmonic	0	0	0	0	0	0
Harmonic roots	0	0	0	1	2	3
Very fast Geometric	0	0	0	0	0	0
Fast Geometric	0	0	0	0	0	0
Geometric	0	0	0	0	0	0
Moderate Geometric	1	1	2	1	0	0
Slow Geometric	7	7	8	9	10	7
Very slow Geometric	28	27	28	23	26	23
Dense Equispaced	43	43	39	33	31	32
Equispaced	6	7	7	9	7	4
Low-Rank-100	1	1	1	0	0	0
Low-Rank-50	0	0	0	0	0	0
Low-Rank-10	0	0	0	0	0	0
Harmonic triples	2	2	2	3	3	4
Multiple-Harmonic	3	2	2	2	2	1
Multiple-Geometric	3	4	3	2	2	1
Equispaced-Geometric Gap	1	1	1	0	0	0

Table 3. Computing k dominant eigenvalues with the Power-Krylov matrix, $\ell = k + 40$.

Matrix type	Number of iterations					
	$k = 6$	$k = 10$	$k = 20$	$k = 40$	$k = 100$	$k = 200$
Harmonic squares	0	0	0	0	0	1
Harmonic	0	0	0	0	0	0
Harmonic roots	0	0	0	0	0	0
Very fast Geometric	0	0	0	0	0	0
Fast Geometric	0	0	0	0	0	0
Geometric	0	0	0	0	0	1
Moderate Geometric	0	0	0	0	0	0
Slow Geometric	3	4	4	4	4	2
Very Slow Geometric	14	17	18	15	15	12
Dense Equispaced	16	17	19	18	16	15
Equispaced-1000	3	4	3	4	3	2
Low-Rank-100	0	0	0	0	0	0
Low-Rank-50	0	0	0	0	0	0
Low-Rank-10	0	0	0	0	0	0
Harmonic triples	1	2	2	2	2	2
Multiple-Harmonic	2	2	2	1	1	1
Multiple-Geometric	1	3	1	2	0	0
Equispaced-Geometric Gap	0	0	0	0	0	0

method. The driving force that ensures convergence is the monotonicity property, which is easily concluded from the Cauchy-Poincaré interlacing theorems. The proof indicates two important points. First, there is a lot of freedom in choosing the information matrix, and that monotonicity is guaranteed as long as we achieve proper orthogonalizations. Second, the rate of convergence depends on the “quality” of the information matrix. This raises the question of how to define this matrix. Since the algorithm is aimed at computing a cluster of exterior eigenvalues, a Krylov information matrix is a good choice. In [4] we have tested the classical Krylov matrix where \mathbf{b}_j is obtained by normalizing $G\mathbf{b}_{j-1}$. However, as shown in [5], this matrix suffers from certain deficiencies. In this paper the Krylov basis is built by a three term recurrence relation, which leads to dramatic reduction in the number of iterations.

Indeed, the results of our experiments are quite encouraging. We see that the algorithm requires a remarkably small number of iterations. In particular, it efficiently handles various kinds of low-rank matrices. In these matrices the initial orthonormal matrix is often sufficient for accurate computation of the desired eigenpairs. The algorithm is also successful in computing eigenvalues of “difficult” matrices like “Dense equispaced” or “Very slow geometric decay”.

References

- [1] Bai, A., Demmel, J., Dongarra, J., Ruhe, A. and van der Vorst, H. (1999) *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*. SIAM, Philadelphia, PA.
- [2] Björck, A. (1996) *Numerical Methods for Least-Squares Problems*. SIAM, Philadelphia. <https://doi.org/10.1137/1.9781611971484>
- [3] Calvetti, D., Reichel, L. and Sorenson, D.C. (1994) An Implicitly Restarted Lanczos Method for Large Symmetric Eigenvalue Problems. *Electronic Transactions on Numerical Analysis*, **2**, 1-21.
- [4] Dax, A. (2015) A Subspace Iteration for Calculating a Cluster of Exterior Eigenvalues. *Advances in Linear Algebra and Matrix Theory*, **5**, 76-89. <https://doi.org/10.4236/alamt.2015.53008>
- [5] Dax, A. (2016) The Numerical Rank of Krylov Matrices. In: *Linear Algebra and Its Applications*.
- [6] Demmel, J.W. (1997) *Applied Numerical Linear Algebra*. SIAM, Philadelphia. <https://doi.org/10.1137/1.9781611971446>
- [7] G.H. Golub and C.F. Van Loan (2013) *Matrix Computations*. 4th Edition, Johns Hopkins University Press, Baltimore.
- [8] Horn, R.A. and Johnson, C.R. (1985) *Matrix Analysis*. Cambridge University Press, Cambridge. <https://doi.org/10.1017/CBO9780511810817>
- [9] Morgan, R.B. (1996) On Restarting the Arnoldi Method for Large Non-Symmetric Eigenvalues Problems. *Mathematics of Computation*, **65**, 1213-1230.
- [10] Parlett, B.N. (1980) *The Symmetric Eigenvalue Problem*. Prentice-Hall, Englewood Cliffs, NJ.
- [11] Saad, Y. (2011) *Numerical Methods for Large Eigenvalue Problems: Revised Edition*. SIAM, Philadelphia. <https://doi.org/10.1137/1.9781611970739>
- [12] Sorensen, D.C. (1992) Implicit Application of Polynomial Filters in a k -Step Arnoldi

di Method. *SIAM Journal on Matrix Analysis and Applications*, **13**, 357-385.

<https://doi.org/10.1137/0613025>

- [13] Stewart, G.W. (1998) *Matrix Algorithms, Vol. I: Basic Decompositions*. SIAM, Philadelphia.
- [14] Stewart, G.W. (2001) *Matrix Algorithms, Vol. II: Eigensystems*. SIAM, Philadelphia.
- [15] Trefethen, L.N. and Bau III, D. (1997) *Numerical Linear Algebra*. SIAM, Philadelphia.
- [16] Watkins, D.S. (2007) *The Matrix Eigenvalue Problem: GR and Krylov Subspace Methods*. SIAM, Philadelphia. <https://doi.org/10.1137/1.9780898717808>
- [17] Wilkinson, J.H. (1965) *The Algebraic Eigenvalue Problem*. Clarendon Press, Oxford.
- [18] Wu, K. and Simon, H. (2000) Thick-Restarted Lanczos Method for Large Symmetric Eigenvalue Problems. *SIAM Journal on Matrix Analysis and Applications*, **22**, 602-616. <https://doi.org/10.1137/S0895479898334605>
- [19] Yamazaki, I., Bai, Z., Simon, H., Wang, L. and Wu, K. (2010) Adaptive Projection Subspace Dimension for the Thick-Restart Lanczos Method. *ACM Transactions on Mathematical Software*, **37**, Article No. 27. <https://doi.org/10.1145/1824801.1824805>
- [20] Zhang, F. (1999) *Matrix Theory: Basic Results and Techniques*. Springer-Verlag, New York. <https://doi.org/10.1007/978-1-4757-5797-2>
- [21] Zhou, Y. and Saad, Y. (2008) Block Krylov-Schur Method for Large Symmetric Eigenvalue Problems. *Numerical Algorithms*, **47**, 341-359. <https://doi.org/10.1007/s11075-008-9192-9>



Submit or recommend next manuscript to SCIRP and we will provide best service for you:

Accepting pre-submission inquiries through Email, Facebook, LinkedIn, Twitter, etc.

A wide selection of journals (inclusive of 9 subjects, more than 200 journals)

Providing 24-hour high-quality service

User-friendly online submission system

Fair and swift peer-review system

Efficient typesetting and proofreading procedure

Display of the result of downloads and visits, as well as the number of cited articles

Maximum dissemination of your research work

Submit your manuscript at: <http://papersubmission.scirp.org/>

Or contact alamt@scirp.org