ACCELERATING CONVERGENCE BY AUGMENTED RAYLEIGH-RITZ PROJECTIONS FOR LARGE-SCALE EIGENPAIR COMPUTATION

ZAIWEN WEN[‡] AND YIN ZHANG[§]

Abstract.
Iterative algorithms for large-scale eigenpair computation are mostly based subspace projections consisting of two main steps: a subspace update (SU) step that generates bases for approximate eigenspaces, followed by a Rayleigh-Ritz (RR) projection step that extracts approximate eigenpairs. A predominant methodology for the SU step makes use of Krylov subspaces that builds orthonormal bases piece by piece in a sequential manner. On the other hand, block methods such as the classic (simultaneous) subspace iteration, allow higher levels of concurrency than what is reachable by Krylov subspace methods, but may suffer from slow convergence. In this work, we analyze the rate of convergence for a simple block algorithmic framework that combines an augmented Rayleigh-Ritz (ARR) procedure with the subspace iteration. Our main results are Theorem 4.5 and its corollaries which show that the ARR procedure can provide significant accelerations to convergence speed. Our analysis will offer useful guidelines for designing and implementing practical algorithms from this framework.

1. Introduction. For a given real symmetric matrix $A \in \mathbb{R}^{n\times n}$, let $\lambda_1, \lambda_2, \cdots, \lambda_n$ be the eigenvalues of A sorted in an descending order: $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$, and $u_1, \ldots, u_n \in \mathbb{R}^n$ be corresponding eigenvectors such that $Au_i = \lambda_i u_i$, $\|u_i\|_2 = 1$, $i = 1, \ldots, n$ and $u_i^T u_j = 0$ for $i \neq j$. An eigen-decomposition of A is then $A = U_n \Lambda_n U_n^T$, where for any integer $i \in [1, n]$

$$U_i = [u_1,\ u_2,\ \ldots, u_i] \in \mathbb{R}^{n\times i}, \quad \Lambda_i = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_i) \in \mathbb{R}^{i\times i}, \tag{1.1}$$

and $\mathrm{diag}(\cdot)$ denotes a diagonal matrix with its arguments on the diagonal. For simplicity, we also write $A = U\Lambda U^T$ where $U = U_n$ and $\Lambda = \Lambda_n$. In this paper, we consider A to be large-scale, which usually implies that A is sparse. Since eigenvectors are generally dense, instead of computing all n eigenpairs of A, in applications it is only realistic to compute $k \ll n$ eigenpairs corresponding to k largest or smallest eigenvalues of A. Fortunately, these so-called exterior (or extreme) eigenpairs of A often contain the most relevant information about the underlying system or dataset represented by the matrix A. Unlike in many previous works, in this work we concentrate on the cases where k is far larger than a few. As the problem size n becomes ever larger, the scalability of algorithms with respect to k has become a critical issue even though k remains a small portion of n.

Most algorithms for computing a subset of eigenpairs of large matrices are iterative in which each iteration consists of two main steps: a subspace update (SU) step and a projection step. The subspace update step varies from method to method but with a common goal in finding a matrix $X \in \mathbb{R}^{n\times k}$ so that its column space is a good approximation to the k-dimensional eigenspace spanned by k desired eigenvectors. Once X is obtained and orthonormalized, the projection step aims to extract from $X^T AX$ a set of approximate eigenpairs that are optimal in a sense. The method of choice for this projection step is the Rayleigh-Ritz (RR) procedure, as will be detailed in Section 2. More complete treatments of iterative algorithms for computing subsets of eigenpairs can be found in several books, for example [1, 15, 19, 4, 23].

For decades, the predominant methodology for subspace update had been (and arguably still is) Krylov subspace methods, as represented by Lanczos type methods [9, 12] for real symmetric matrices. These methods generate an orthonormal matrix X one column (or a few

[‡]Beijing International Center for Mathematical Research, Peking University, Beijing, CHINA (wenzw@pku.edu.cn). Research supported in part by NSFC grants 11322109 and 11421101, and by the National Basic Research Project under the grant 2015CB856000.

[§]Department of Computational and Applied Mathematics, Rice University, Houston, UNITED STATES (yzhang@rice.edu). Research supported in part by NSF DMS-1115950 and NSF DMS-1418724.

columns) at a time in a sequential mode. Along the way, each column (or group of columns) is multiplied by the matrix A and made orthogonal to all the previous ones. In contrast to Krylov subspace methods, block methods, as represented by the classic simultaneous subspace iteration method [16], carry out the multiplications of A to all columns of X at the same time in a batch mode. As such, block methods generally require a lower level of communication intensity.

The operation of the sparse matrix A multiplying a vector, or SpMV, used to be the most relevant complexity measure for algorithm efficiency. As Krylov subspace methods generally require considerably fewer SpMVs than block methods do, they had become the methodology of choice for the past few decades even up to date. However, the evolution of modern computer architectures, particularly the emergence of multi/many-core architectures, has seriously eroded the relevance of SpMV (and arithmetic operations in general) as a leading complexity measure, as communication costs have, gradually but surely, become more and more predominant.

The purpose of this work is to analyze a simple block algorithmic framework for computing a relatively large number of exterior eigenpairs. It is widely accepted that a key shortcoming of block methods is that their convergence can become excessively slow when the decay rate in relevant eigenvalues is too flat. A central effort of our algorithm construction is to rectify this issue of slow convergence. Our framework starts with an outer iteration loop that features an enhanced RR step called augmented Rayleigh-Ritz (ARR) projection which can provably accelerate convergence under mild conditions. For the SU step, we consider the classic power method applied to multiple vectors without frequent or periodic orthogonalizations. The well-known technique of polynomial acceleration can also be incorporated into the framework, but will not be studied in any detail. Our main contribution is an analysis of the proposed framework that reveals the rate of convergence by the ARR projection, and provides guidelines for the construction of practical algorithms within the framework.

The rest of this paper is organized as follows. A brief overview of relevant iterative algorithms for eigenpair computation is presented in Section 2. The ARR procedure and our algorithm framework are proposed in Section 3. We analyze the ARR procedure in Section 4. Numerical results are presented in Section 5. Finally, we conclude the paper in Section 6.

2. Overview of Iterative Algorithms for Eigenpair Computation. Algorithms for the eigenvalue problem have been extensively studied for decades. We will only briefly review a small subset of them that are most closely related to the present work.

Without loss of generality, we assume for convenience that A is positive definite (after a shift if necessary). Our task is to compute k largest eigenpairs (U_k, Λ_k) for some $k \ll n$ where by definition $AU_k = U_k\Lambda_k$ and $U_k^T U_k = I \in \mathbb{R}^{k\times k}$. Replacing A by a suitable function of A, say $\lambda_1 I - A$, one can also in principle apply the same algorithms to finding k smallest eigenpairs as well.

An RR step is to extract approximate eigenpairs, called Ritz-pairs, from a given matrix $Z \in \mathbb{R}^{n\times m}$ whose range space, $\mathcal{R}(Z)$, is supposedly an approximation to a desired m-dimensional eigenspace of A. Let $\mathbf{orth}(Z)$ be the set of orthonormal bases for the range space of Z. The RR procedure is described as Algorithm 1 below, which is also denoted by a map $(Y, \Sigma) = \mathrm{RR}(A, Z)$ where the output (Y, Σ) is a Ritz pair block.

2.1. Krylov Subspace Methods. Krylov subspaces are the foundation of several state-of-the-art solvers for large-scale eigenvalue calculations. By definition, for given matrix $A \in \mathbb{R}^{n\times n}$ and vector $v \in \mathbb{R}^n$, the Krylov subspace of order $p \geq 0$ associated with A and v is

$$\mathcal{K}_p(A, v) = \mathrm{span}\{v, Av, A^2v, \ldots, A^pv\}. \tag{2.1}$$

Algorithm 1: Rayleigh-Ritz procedure: $(Y, \Sigma) = \mathrm{RR}(A, Z)$

1 Given $Z \in \mathbb{R}^{n\times m}$, orthonormalize Z (if necessary) to obtain $U \in \mathbf{orth}(Z)$.
2 Compute $H = U^T AU \in \mathbb{R}^{m\times m}$, the projection of A onto the range space of U.
3 Compute the eigen-decomposition $H = V^T \Sigma V$, where $V^T V = I$ and Σ is diagonal.
4 Assemble the Ritz pairs (Y, Σ) where $Y = UV \in \mathbb{R}^{n\times m}$ satisfies $Y^T Y = I$.

Typical Krylov subspace methods include Arnoldi algorithm for general matrices (e.g., [12, 11]) and Lanczos algorithm for symmetric (or Hermitian) matrices (e.g., [20, 10]). In either algorithm, orthonormal bases for Krylov subspaces are generated through a Gram-Schmidt type process. Some variants of Jacobi-Davidson methods (e.g., [2, 21]) are based on a different framework, but they too rely on Krylov subspace methodologies to solve linear systems at every iteration.

Direct extensions of Krylov methods lead to so-called block Krylov methods [5, 26, 3, 7] that replace a single vector $v \in \mathbb{R}^n$ by a block matrix $V \in \mathbb{R}^{n\times b}$, $b > 1$, for the purpose of either improving convergence or enhancing parallelism. Starting from $V \in \mathbb{R}^{n\times b}$, block Krylov methods generate an orthonormal basis for the block Krylov subspace

$$\mathcal{K}_p(A, V) = \mathrm{span}\{V, AV, A^2V, \ldots, A^pV\}, \tag{2.2}$$

and then apply the RR procedure to compute approximate eigenpairs of A. The dimension of $\mathcal{K}_p(A, V)$ can be up to b times larger than that of $\mathcal{K}_p(A, v)$.

2.2. Classic Subspace Iteration. The simple (or simultaneous) subspace iteration (SSI) method (see [16, 17, 22, 24, 23], for example) extends the idea of the power method which computes a single eigenpair corresponding to the largest eigenvalue (in magnitude). Starting from an initial (random) matrix U, SSI performs repeated matrix multiplications AU, followed by periodic orthogonalizations and RR projections. The main purpose of orthogonalization is to prevent the iterate matrix U from losing rank numerically. In addition, since the rates of convergence for different eigenpairs are uneven, numerically converged eigenvectors can be deflated after each RR projection. A main advantage of SSI is the use of simultaneous matrix-block multiplications instead of individual matrix-vector multiplications. It enables fast memory access and highly parallelizable computation on modern computer architectures. Suppose that the eigenvalues of A are ordered into a descending order in absolute value and there is a gap between the k-th and the $(k+1)$-th eigenvalues. Then the SSI method is guaranteed to converge to the eigenspace corresponding to the k largest eigenvalues from any generic starting point. However, a severe shortcoming of the SSI method is that its convergence speed depends critically on eigenvalue distributions that can, and often does, become intolerably slow in the face of unfavorable eigenvalue distributions.

2.3. Trace Maximization Methods. Computing a k-dimensional eigenspace associated with k largest eigenvalues of A is equivalent to solving:

$$\max_{X\in\mathbb{R}^{n\times k}} \mathrm{tr}(X^T AX), \ \mathbf{s.t.}\ X^T X = I. \tag{2.3}$$

Some block algorithms have been developed based on solving (2.3) or its minimization counterpart. Projection-type methods include the locally optimal block preconditioned conjugate gradient method (LOBPCG) [8] and more recently the limited memory block Krylov subspace optimization method (LMSVD) [14]. At each iteration, these methods solve a subspace

trace maximization problem of the form

$$Y = \underset{X \in \mathbb{R}^{n\times k}}{\arg\max} \left\{ \mathrm{tr}(X^T A X) : X^{\mathrm{T}} X = I,\ X \in \mathcal{S} \right\}, \tag{2.4}$$

where $X \in \mathcal{S}$ means that each column of X is in the given subspace $\mathcal{S}$. LOBPCG [8] constructs $\mathcal{S}$ as the span of the two most recent iterates $X^{(i-1)}$ and $X^{(i)}$, and the residual at $X^{(i)}$. In LMSVD [14], the subspace $\mathcal{S}$ is spanned by the current i-th iterate and the previous p iterates. For a given $\mathcal{S}$, problem (2.4) is solved by calling Algorithm 1 (i.e., the RR procedure) with input Z being a basis for $\mathcal{S}$.

3. An Algorithmic Framework with Augmented Rayleigh-Ritz Projections. It is easy to see that the Rayleigh-Ritz procedure in Algorithm 1 solves the trace-maximization subproblem (2.4) with the subspace $\mathcal{S} = \mathcal{R}(Z)$, while the solution Y is such that $Y^{\mathrm{T}} A Y$ is a diagonal matrix Σ. Naturally, for a fixed number k the larger the subspace $\mathcal{R}(Z)$ is, the greater chances there are to extract k Ritz pairs of better quality. The classic SSI always sets Z to the current iterate $X^{(i)}$, while both LOBPCG [8] and LMSVD [14] augment $X^{(i)}$ by additional blocks. Not surprisingly, using such augmented RR projections is the main reason why algorithms like LOGPCG and LMSVD generally converge faster SSI does.

In this work, we focus on using an augmented RR procedure where the augmentation is based on a block Krylov subspace structure as in (2.2). Specifically, for integer $p \geq 0$ we let

$$\mathcal{S} = \mathcal{K}_p(A, X) \equiv \mathrm{span}\{X, AX, A^2X, \ldots, A^pX\}. \tag{3.1}$$

With the above subspace $\mathcal{S}$ and a basis Z, we solve the trace maximization problem (2.4) via the RR procedure. We call this procedure the augmented RR (or ARR) procedure, which is formally presented as Algorithm 2.

Algorithm 2: ARR: $(Y, \Sigma) = \textsc{Arr}(A, X, p)$

1 Input $X \in \mathbb{R}^{n\times k}$ and $p \geq 0$ so that $(p+1)k < n$.
2 Construct matrix $K_p = [X \ \ AX \ \ A^2X \ \ \cdots \ \ A^pX]$.
3 Perform an RR step using $(\hat{Y}, \hat{\Sigma}) = \mathrm{RR}(A, K_p)$.
4 Extract k leading Ritz pairs (Y, Σ) from $(\hat{Y}, \hat{\Sigma})$.

We next introduce a prototype algorithmic framework that is equipped with ARR projections coupled with a block method for subspace update. We will call this prototype framework ARRABIT, standing for ARR and block iteration. In this framework, at each outer iteration a subspace update (SU) step is performed, and then an ARR step follows.

In principle, the SU step can be fulfilled by any reasonable block scheme that should not necessarily require orthogonalizations. In this paper, we consider the classic power iteration as our main updating scheme, i.e., for $X_0 = [x_1\ x_2\ \cdots, x_k] \in \mathbb{R}^{n\times k}$, we set $X = \rho(A)^q X_0$, where $q > 0$ is an integer parameter, and $\rho(t)$ is a polynomial including $\rho(t) = t$ (i.e., no acceleration). Formally, we state our algorithmic framework as Algorithm 3. Although far away from numerically viable, this prototype algorithm will allow us to carry out a theoretical convergence analysis in the setting of exact arithmetic.

3.1. Convergence Rate of Subspace Iteration. It is clear that when there is no augmentation (i.e., $p = 0$), Algorithm 3 reduces essentially to the classic subspace iteration where orthonormalization is done every q power iterations. Let $\{|\rho(\lambda_j)|\}_{j=1}^n$ be ordered in a descending order and

$$X = \rho(A)^q X_0 \in \mathbb{R}^{n\times k}, \tag{3.2}$$

Algorithm 3: ARRABIT (prototype)

Input matrix $A \in \mathbb{R}^{n\times n}$, integers $k, p, q > 0$ and polynomial $\rho(t)$.
Initialize X to a random matrix $X_0 \in \mathbb{R}^{n\times k}$.
while *"not converged",* **do**
    Power iteration: $X = \rho(A)^q X$.
    ARR projection: $(X, \Sigma) = \textsc{Arr}(A, X, p)$ as in Algorithm 2.

where X_0 is a generic initial matrix. Then it is well known (see [23] for example) that the rate of convergence of $\mathcal{R}(X)$ to the eigenspace $\mathcal{R}(U_k)$ is given by

$$\langle \mathcal{R}(U_k), \mathcal{R}(X)\rangle = O\left(\left|\frac{\rho(\lambda_{k+1})}{\rho(\lambda_k)}\right|^q\right), \tag{3.3}$$

where $\langle\cdot,\cdot\rangle$ is the angle between two subspaces, provided that there is a gap between $|\rho(\lambda_k)|$ and $|\rho(\lambda_{k+1})|$. However, it is a common occurrence that $|\lambda_k|$ and $|\lambda_{k+1}|$ are so close to each other that using polynomial filtering alone can hardly separate them, making the convergence speed of subspace iteration too slow to be practical in many situations.

To accelerate convergence, one could use more than k columns to compute k eigenpairs. For instance, if $X_0, X \in \mathbb{R}^{n\times rk}$ in (3.2) for some $r \geq 1$, then the convergence rate will be improved to

$$\langle \mathcal{R}(U_k), \mathcal{R}(X)\rangle = O\left(\left|\frac{\rho(\lambda_{rk+1})}{\rho(\lambda_k)}\right|^q\right). \tag{3.4}$$

However, the amount of computation in each power iteration will be increased about r times, making such a strategy unattractive when k is relatively large.

A main result of this paper is to show that with augmentation in Algorithm 3, i.e., $p > 0$ in the ARR procedure, the faster rate of convergence in (3.4) can be achieved under reasonable conditions. The main computational cost of achieving such an acceleration is to perform an RR projection onto an rk-dimensional subspace instead of a k-dimensional one.

3.2. Relations to Block Lanczos Methods. On the surface, the ARR procedure, presented as Algorithm 2, is mathematically equivalent to block Lanczos methods. Both apply RR projections to A onto Krylov subspaces: ARR onto $\mathcal{K}_p(A, X)$ for $X \in \mathbb{R}^{n\times k}$ and the block Lanczos onto $\mathcal{K}_p(A, V)$ for $V \in \mathbb{R}^{n\times b}$. When $k = b$, the two subspaces are indeed mathematically identical provided that $X = V$. However, this apparent equivalence is only an empty proposition.

Extending the Lanczos iterations from a single vector to a few, block Lanczos methods operate under the implicit condition of $b \ll p$. In fact, existing convergence results for block Lanczos methods require $b \leq p + 1$ (see the next paragraph). On the contrary, our ARRABIT framework is primarily constructed to compute a relatively large number of eigenpairs (say, $k = 500$) while using only a few augmentation blocks in the ARR procedure (say, $p \leq 5$); that is, we are interested in cases of $k \gg p$. This implies that $k \leq p + 1$ would never hold for the cases of our interest. Consequently, the existing convergence results for block Lanczos methods are not applicable to the cases of our interest.

The convergence rate of either single-vector or block Lanczos methods has been analyzed in [18, 15, 6, 13]. All the rate of convergence results developed so far, to the best of our knowledge, rely on Chebyshev polynomials of the first kind. Specifically, when k eigenpairs are computed, the error bound for the i-th eigenvector, $i = 1, 2, \cdots, k$, requires evaluating

the $(p+1-i)$-th degree Chebyshev polynomials of the first kind at a point greater than 1. For $i = k$, obviously $p \geq k$ is necessary in order to ensure the existence of a meaningful error bound. For the cases of our interest where $k \gg p$, none of the existing theoretical error bounds is applicable, which necessitates an analysis of a different kind.

4. Analysis of the Augmented Rayleigh-Ritz Procedure. In this section, we provide new understanding on the convergence of the ARR procedure from a different perspective than the existing results. To facilitate our analysis, we first propose a new measure of accuracy for approximations of eigenspaces.

4.1. A Measure of Accuracy. Recall that $A = U\Lambda U^T$ is an eigen-decomposition of $A \in \mathbb{R}^{n\times n}$. For integer $k \in [1, n)$, we introduce the partition $U = [U_k\ U_{k+}]$ where

$$U_k = [u_1\ \ u_2\ \ \cdots u_k] \quad \text{and} \quad U_{k+} = [u_{k+1}\ \ u_{k+2}\ \cdots\ u_n]. \tag{4.1}$$

Let $X \in \mathbb{R}^{n\times k}$ be an approximate basis for $\mathcal{R}(U_k)$, the range space of U_k. It is desirable for X to have a large projection in $\mathcal{R}(U_k)$ relative to that in $\mathcal{R}(U_{k+})$. We will measure the accuracy of X based on the numbers in $\{\|u_i^T X\|\}_{i=1}^n$, where $\|u_i^T X\| = \|(u_i u_i^T)X\|$ is the projection of X onto the one-dimensional subspace $\mathcal{R}(u_i)$.

For a fixed X, however, the number $\|u_i^T X\|$ is unique if and only if the multiplicity of λ_i, the i-th eigenvalue of A, is one; otherwise different orthonormal bases can give rise to different values of $\|u_i^T X\|$. For a reason that will soon becomes clear, we first introduce the following technical assumption without loss of generality.

Let $X \in \mathbb{R}^{n\times k}$ be a given nonzero matrix. For an index $i \in [1, k]$, if $\lambda_i = \lambda_{i+\ell}$ is a multiple eigenvalue whose multiplicity equals $\ell + 1$ for some $\ell > 0$, then without loss of generality we will always assume that an orthonormal basis, $\{u_j\}_{j=i}^{i+\ell}$, for the eigenspace of λ_i is so constructed that the smallest value in $\{\|u_j^T X\|\}_{j=i}^{i+\ell}$ is maximized over the set of all orthonormal bases for the eigenspace of λ_i. That is, the set $\{u_j\}_{j=i}^{i+\ell}$ solves the following problem

$$\max_{\{v_0,\cdots,v_\ell\}} \min_{j\in\{0,\cdots,\ell\}} \|v_j^T X\| \tag{4.2}$$

where $\{v_0, \cdots, v_\ell\} \subset \mathbb{R}^n$ represents any orthonormal basis for the eigenspace of λ_i.

The optimal value in (4.2) is always positive unless $V^T X = 0$ for $V = [v_0, \cdots, v_\ell]$, meaning that all columns of X are perpendicular to the eigenspace of λ_i. As long as $V^T X$ has a single nonzero column, one can rotate it into the first orthant so that the rotated matrix, say $R(V^T X)$, has no zero rows, while the columns of VR^T remain an orthonormal basis for the eigenspace of λ_i that guarantees a nonzero objective value in (4.2).

Now we define a measure for the relative accuracy of X to be the ratio

$$\delta_k(X) \triangleq \frac{\max_{i>k} \|u_i^T X\|}{\min_{i\leq k} \|u_i^T X\|}. \tag{4.3}$$

Clearly, $\delta_k(X) = 0$ means that all the columns of X are in $\mathcal{R}(U_k)$. In general, the smaller $\delta_k(X)$ is, the better is X as an approximate basis for $\mathcal{R}(U_k)$. By our technical assumption above, the denominator in the ratio is zero if and only if the columns of X are all in $\mathcal{R}(U_{k+})$ — the orthogonal complement of $\mathcal{R}(U_k)$.

Let $Y \in \mathbb{R}^{n\times k}$ be another approximate basis for the eigenspace $\mathcal{R}(U_k)$ which is constructed from X. To compare Y with X, we naturally compare $\delta_k(Y)$ with $\delta_k(X)$. More precisely, we will try to estimate the ratio $\delta_k(Y)/\delta_k(X)$ and show that under reasonable conditions, it can be made much less than the unity.

4.2. Technical Results. Before calling the ARR procedure, we have an iterate matrix of rank k, $X \in \mathbb{R}^{n\times k}$, from which we construct the augmented matrix

$$K_p = [X \;\; AX \;\; \cdots \;\; A^pX]$$

for a given $p \geq 0$. In view of $A = U\Lambda U^T$, we rewrite U^TK_p as

$$U^TK_p = [U^TX \;\; \Lambda U^TX \;\; \cdots \;\; \Lambda^pU^TX] \in \mathbb{R}^{n\times(p+1)k}. \tag{4.4}$$

We next normalize the rows of U^TK_p. Let $D = \mathrm{diag}(d_{11}, \ldots, d_{nn})$ be the diagonal matrix whose diagonal consists of the row norms of U^TK_p. From the structure of U^TK_p in (4.4), we see that

$$d_{ii} = \|e_i^TU^TK_p\| = \|u_i^TX\|\|e_i^TV\|, \;\; i = 1, 2, \cdots, n, \tag{4.5}$$

where e_i is the i-th column of the $n \times n$ identity matrix and V is the following Vandermonde matrix constructed from the spectrum of A:

$$V = \begin{pmatrix} 1 & \lambda_1 & \lambda_1^2 & \cdots & \lambda_1^p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \lambda_n & \lambda_n^2 & \cdots & \lambda_n^p \end{pmatrix} \in \mathbb{R}^{n\times(p+1)}, \tag{4.6}$$

where $\lambda_1, \cdots, \lambda_n$ are the eigenvalues of A.

Let $D^\dagger$ be the pseudo-inverse of D, that is, $D^\dagger$ is a diagonal matrix with $(D^\dagger)_{ii} = 1/d_{ii}$ if $d_{ii} > 0$ and zero otherwise. The normalization of the rows of U^TK_p in (4.4) leads to the matrix

$$G = D^\dagger U^TK_p = [C \;\; \Lambda C \;\; \cdots \;\; \Lambda^pC] \;\text{ for }\; C = D^\dagger U^TX, \tag{4.7}$$

so that the nonzero rows of G all have unit norm. Now we can rewrite

$$K_p = UDD^\dagger U^TK_p = UDG. \tag{4.8}$$

For $p \geq 0$ so that $k + pk < n$, let m be an integer parameter so that $m \in [k, k + pk]$. Consider the partition

$$K_p = [U_m \; U_{m+}] \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix} \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} = [U_m \; U_{m+}] \begin{bmatrix} D_1G_1 \\ D_2G_2 \end{bmatrix}, \tag{4.9}$$

where D and G are partitioned following that of U. In particular, $G_1 \in \mathbb{R}^{m\times(p+1)k}$ consists of the first m rows of G. In the sequel, we will make use of an important assumption on G which we will call the G-Assumption:

ASSUMPTION 4.1 (G-Assumption). *The first $m \in [k, k + pk]$ rows of G in* (4.7) *are linearly independent; i.e., $G_1 \in \mathbb{R}^{m\times(p+1)k}$ in* (4.9) *has full row rank.*

The G-Assumption implies that (i) $D_1 > 0$, and (ii) the pseudo-inverse $G_1^\dagger$ exists such that $G_1G_1^\dagger = I_{m\times m}$. In view of (4.9), let us define

$$Y = K_pG_1^\dagger D_1^{-1}E_k = [U_m \; U_{m+}] \begin{bmatrix} I_{m\times m} \\ D_2G_2G_1^\dagger D^{-1} \end{bmatrix} E_k, \tag{4.10}$$

where $E_k \in \mathbb{R}^{m\times k}$ consists of the first k columns of the $m \times m$ identity matrix; i.e., Y consists of the first k columns of the matrix in front of E_k. Another implication of the G-Assumption is that U_m^TX must not have zero rows; otherwise the rank of the first m rows of C in (4.7) would be less than m, contradicting the G-Assumption.

We summarize what we already have for Y into the following lemma which directly follows from (4.10).

LEMMA 4.2. *Let $A = U\Lambda U^T$ be the eigen-decomposition of $A = A^T \in \mathbb{R}^{n\times n}$. For integers $k > 0$ and $p \geq 0$ satisfying $(p+1)k < n$, and $m \in [k, k+pk]$, let G and K_p be defined as in* (4.7) *and* (4.8)*, respectively, for a rank-k matrix $X \in \mathbb{R}^{n\times k}$. Under the G-Assumption, Y in* (4.10) *has the expression*

$$Y = U_m E_k + U_{m+} S E_k \in \mathbb{R}^{n\times k}, \tag{4.11}$$

where $S = D_2 G_2 G_1^\dagger D_1^{-1}$ and $E_k \in \mathbb{R}^{m\times k}$ consists of the first k columns of $I_{m\times m}$.

Since Y is extracted from the subspace $\mathcal{R}(K_p)$ constructed from X, a central question is how much improvement Y can provide over X as an approximate basis for $\mathcal{R}(U_k)$. We study this question by comparing the accuracy measure $\delta_k(Y)$ relative to $\delta_k(X)$.

LEMMA 4.3. *Let d_{ii} be defined in* (4.5)*. Under the conditions of Lemma 4.2,*

$$\delta_k(Y) \leq \frac{\max_{i>m} d_{ii}}{\min_{i\leq k} d_{ii}} \max_{1\leq i\leq n-m} \|e_i^T G_2 G_1^\dagger E_k\|. \tag{4.12}$$

Proof. It follows from (4.11) that

$$u_i^T Y = \begin{cases} e_i^T, & i \in [1,k] \\ \mathbf{0}^T, & i \in [k+1, m] \\ e_{i-m}^T S E_k, & i \in [m+1, n] \end{cases}$$

where $e_i \in \mathbb{R}^k$, $\mathbf{0} \in \mathbb{R}^k$ and $e_{i-m} \in \mathbb{R}^{n-m}$. These formulas imply that in $\delta_k(Y)$ (see definition (4.3)) the denominator is $\min_{i\leq k} \|u_i^T Y\| = 1$; thus

$$\delta_k(Y) = \max_{i>k} \|u_i^T Y\| = \max_{i>m} \|u_i^T Y\|. \tag{4.13}$$

In view of the formula $S = D_2 G_2 G_1^\dagger D_1^{-1}$, and the definition of D in (4.5), we have

$$u_i^T Y = d_{ii} e_{i-m}^T G_2 G_1^\dagger D_1^{-1} E_k, \quad i \in [m+1, n].$$

Therefore, for $i \in [m+1, n]$, $\|u_i^T Y\| \leq (d_{ii}/\min_{j\leq k} d_{jj}) \|e_{i-m}^T G_2 G_1^\dagger E_k\|$. It follows that

$$\max_{i>m} \|u_i^T Y\| \leq \frac{\max_{i>m} d_{ii}}{\min_{i\leq k} d_{ii}} \max_{1\leq i\leq n-m} \|e_i^T G_2 G_1^\dagger E_k\|,$$

which, together with (4.13), establishes (4.12). ∎

4.3. Main Results. For any matrix M of n rows, for $m \in [k, k+pk]$ we define

$$\Gamma_{k,m}(M) \triangleq \frac{\max_{j>m} \|e_j^T M\|}{\min_{j\leq k} \|e_j^T M\|}. \tag{4.14}$$

By this definition, $\delta_k(X) = \Gamma_{k,k}(U^T X)$. It is worth observing that (i) $\Gamma_{k,m}(M)$ is monotonically non-increasing with respect to m for fixed k and M; (ii) $\Gamma_{k,m}(M)$ is small if the first k rows of M are much larger in magnitude than the last $n-m$; (iii) if $\{\|e_i^T M\|\}$ is non-increasing, then $\Gamma_{k,m}(M) \leq 1$. Specifically, since the eigenvalues of A are ordered in a descending order in absolute value, for the matrix V in (4.6) we have

$$\Gamma_{k,m}(V) = \frac{\|e_{m+1}^T V\|}{\|e_k^T V\|} = \left(\frac{1+\lambda_{m+1}^2 + \cdots + \lambda_{m+1}^{2p}}{1+\lambda_k^2 + \cdots + \lambda_k^{2p}}\right)^{1/2} \leq 1; \tag{4.15}$$

and the faster the decay is between λ_k and λ_{m+1}, the smaller is $\Gamma_{k,m}(V)$.

Moreover, when $M = z \in \mathbb{R}^n$ is a vector which is in turn the element-wise product of two other vectors $x, y \in \mathbb{R}^n$, i.e., $z_i = x_i y_i$ for $i = 1, \cdots, n$, then it holds that

$$\Gamma_{k,m}(z) \le \Gamma_{k,m}(x)\,\Gamma_{k,m}(y). \tag{4.16}$$

In our first main result, we refine the estimation of $\delta_k(Y)$ and compare it to $\delta_k(X)$.

LEMMA 4.4. *Under the conditions of Lemma 4.2,*

$$\delta_k(Y) \le \Gamma_{k,m}(U^T X)\Gamma_{k,m}(V) \left\| G_1^\dagger E_k \right\|_2, \tag{4.17}$$

$$\frac{\delta_k(Y)}{\delta_k(X)} \le \frac{\max_{j>m} \|u_j^T X\|}{\max_{j>k} \|u_j^T X\|} \Gamma_{k,m}(V) \left\| G_1^\dagger E_k \right\|_2. \tag{4.18}$$

Proof. Observe that the ratio in the right-hand side of (4.12) is none other than $\Gamma_{k,m}(d)$. Applying (4.16) to $M = d$ where $d = \mathrm{diag}(D)$ with D_{ii} defined in (4.5), $x_i = \|u_i^T X\|$ and $y_i = \|e_i^T V\|$, we derive $\Gamma_{k,m}(d) \le \Gamma_{k,m}(U^T X)\Gamma_{k,m}(V)$. In addition, we observe that $\|e_i^T G_2 G_1^\dagger E_k\| \le \|G_1^\dagger E_k\|_2$ for all $i \in [1, n-m]$, since the row vectors $e_i^T G_2$ are either zero or unit vectors. Substituting the above two inequalities into (4.12), we arrive at (4.17). To derive (4.18), we simply observe that

$$\Gamma_{k,m}(U^T X) = \frac{\max_{j>m} \|u_j^T X\|}{\min_{j\le k} \|u_j^T X\|} = \delta_k(X) \frac{\max_{j>m} \|u_j^T X\|}{\max_{j>k} \|u_j^T X\|}.$$

Now (4.18) follows from substituting the above into (4.17) and dividing by $\delta_k(X)$. ∎

Next we consider the case where $X \in \mathbb{R}^{n\times k}$ is the result of applying a block power iteration q times to an initial random matrix $X_0 \in \mathbb{R}^{n\times k}$. In this case,

$$X = \rho(A)^q X_0 = U \rho(\Lambda)^q U^T X_0, \quad U^T X = \rho(\Lambda)^q U^T X_0, \tag{4.19}$$

where $\rho(A)$ is a polynomial or rational matrix function accelerator (or filter). Without loss of generality, we can assume that X_0 has rank k and $\delta_k(X_0) < \infty$.

We make the following assumption about the filtered spectrum:

$$\min_{1\le j\le k} |\rho(\lambda_j)| = |\rho(\lambda_k)| \ge |\rho(\lambda_{k+1})| \ge \cdots \ge |\rho(\lambda_{m+1})| = \max_{m<j\le n} |\rho(\lambda_j)|. \tag{4.20}$$

If there is a significant decay in $\{|\rho(\lambda_j)|\}$ from the index $k+1$ on, then it is likely that

$$\max_{j>k} |\rho(\lambda_j)|^q \|u_j^T X_0\| = |\rho(\lambda_{k+1})|^q \|u_{k+1}^T X_0\|, \tag{4.21}$$

especially when a similar decay also exists in $\{\|u_j^T X_0\|\}$.

In view of (4.4) and (4.19), we have

$$U^T K_p = \left[U^T X \;\; \Lambda U^T X \;\; \cdots \;\; \Lambda^p U^T X\right] = \rho(A)^q \left[U^T X_0 \;\; \Lambda U^T X_0 \;\; \cdots \;\; \Lambda^p U^T X_0\right].$$

Recall that G_1 consists of the first m normalized rows of $U^T K_p$. Hence,

$$G_1 = \mathrm{diag}\,(d_{11}, \cdots, d_{mm})^{-1} \left[U_m^T X_0 \;\; \Lambda_m U_m^T X_0 \;\; \cdots \;\; \Lambda_m^p U_m^T X_0\right], \tag{4.22}$$

where U_m is formed by the first m columns of U, $\Lambda_m = \mathrm{diag}\,(\lambda_1, \cdots, \lambda_m)$, and d_{ii} are defined by (4.5) with X replaced by X_0.

When $m = k + pk$, G_1 is square and

$$G_1^{-1} = \left[U_m^T X_0 \;\; \Lambda_m U_m^T X_0 \;\; \cdots \;\; \Lambda_m^p U_m^T X_0\right]^{-1} \operatorname{diag}(d_{11}, \cdots, d_{mm}). \tag{4.23}$$

THEOREM 4.5. *Let X be defined in (4.19) from an initial matrix $X_0 \in \mathbb{R}^{n\times k}$, $\Gamma_{k,m}(V)$ be defined by* (4.15)*, and $G_1^\dagger E_k$ be the first k columns of the pseudo-inverse of G_1 defined in* (4.22)*. Assume that the conditions of Lemma 4.2 hold. Then*

$$\delta_k(Y) \le c_m \left|\frac{\rho(\lambda_{m+1})}{\rho(\lambda_k)}\right|^q, \tag{4.24}$$

$$\frac{\delta_k(Y)}{\delta_k(X)} \le c'_m \left|\frac{\rho(\lambda_{m+1})}{\rho(\lambda_{k+1})}\right|^q, \tag{4.25}$$

where

$$c_m = \Gamma_{k,m}(U^T X_0)\Gamma_{k,m}(V) \left\|G_1^\dagger E_k\right\|_2, \tag{4.26}$$

$$c'_m = \Theta_{k,m}(U^T X_0)\Gamma_{k,m}(V) \left\|G_1^\dagger E_k\right\|_2, \tag{4.27}$$

and

$$\Theta_{k,m} = \begin{cases} \frac{\max_{j>m}\|u_j^T X_0\|}{\min_{j>k}\|u_j^T X_0\|}, & \textit{in general,} \\ \frac{\max_{j>m}\|u_j^T X_0\|}{\|u_{k+1}^T X_0\|}, & \textit{when (4.21) holds.} \end{cases} \tag{4.28}$$

Proof. Since $U^T X = \rho(\Lambda)^q U^T X_0$,

$$\|u_i^T X\| = |\rho(\lambda_i)|^q \|u_i^T X_0\|, \;\; i = 1, \cdots, n. \tag{4.29}$$

By (4.16) and (4.20),

$$\Gamma_{k,m}(U^T X) \le \Gamma_{k,m}(\rho(\Lambda)^q)\Gamma_{k,m}(U^T X_0) = \left|\frac{\rho(\lambda_{m+1})}{\rho(\lambda_k)}\right|^q \Gamma_{k,m}(U^T X_0).$$

Substituting the above into (4.17) yields (4.24) and (4.26).

To prove (4.25), we utilize (4.29) and (4.20) to derive the inequality

$$\frac{\max_{j>m}\|u_j^T X\|}{\max_{j>k}\|u_j^T X\|} = \frac{\max_{j>m}|\rho(\lambda_j)|^q\|u_j^T X_0\|}{\max_{j>k}|\rho(\lambda_j)|^q\|u_j^T X_0\|} \le \left|\frac{\rho(\lambda_{m+1})}{\rho(\lambda_{k+1})}\right|^q \frac{\max_{j>m}\|u_j^T X_0\|}{\min_{j>k}\|u_j^T X_0\|}.$$

Substituting the above into (4.18) yields (4.25) and (4.27) for the general case of (4.28). The second case of (4.28) is obvious when (4.21) holds true. ∎

Finally, let us state a few special cases that are of particular interest.

COROLLARY 4.6. *If the G-Assumption holds for $m = rk$ where $r = p + 1$, then*

$$\delta_k(Y) \le c_{rk} \left|\frac{\rho(\lambda_{rk+1})}{\rho(\lambda_k)}\right|^q, \tag{4.30}$$

$$\frac{\delta_k(Y)}{\delta_k(X)} \le c'_{rk} \left|\frac{\rho(\lambda_{rk+1})}{\rho(\lambda_{k+1})}\right|^q, \tag{4.31}$$

where c_{rk} and c'_{rk} are defined in (4.26) *and* (4.27)*, respectively, in which $m = rk$ and $G_1^\dagger$ reduces to G_1^{-1} defined in* (4.23)*.*

When $p = 0$ (no augmentation) and $\rho(t) = t$ (no polynomial acceleration), inequality (4.30) *reduces to*

$$\delta_k(Y) \le c_k \left| \frac{\lambda_{k+1}}{\lambda_k} \right|^q, \tag{4.32}$$

which recovers the convergence rate of the classic subspace iteration method.

All the above results give asymptotic rates of convergence in exact arithmetic. We note that both constant c_m and c'_m depend on the size of $\|G_1^\dagger E_k\|$ which tends to increase with m. In finite precisions, the term $|\rho(\lambda_{rk+1})/\rho(\lambda_{k+1})|^q$ cannot be made smaller than roundoff errors (in fact, this term may be much larger than roundoff errors). Therefore, an excessively large c_{rk} (or c'_{rk}), which could occur when X is badly conditioned, may render the right-hand of (4.30) (or (4.31)) numerically irrelevant. The corollary below should be more meaningful in finite-precision situations, whose proof follows directly from (4.24) and (4.25).

COROLLARY 4.7. *If the G-Assumption holds for $m = rk$ where $r = p + 1$, then*

$$\delta_k(Y) \le \Psi_k(p,q) \equiv \min_{m \in [k,rk]} c_m \left| \frac{\rho(\lambda_{m+1})}{\rho(\lambda_k)} \right|^q, \tag{4.33}$$

$$\frac{\delta_k(Y)}{\delta_k(X)} \le \Psi'_k(p,q) \equiv \min_{m \in [k,rk]} c'_m \left| \frac{\rho(\lambda_{m+1})}{\rho(\lambda_{k+1})} \right|^q, \tag{4.34}$$

where c_m and c'_m are defined in (4.26) *and* (4.27)*, respectively.*

4.4. Interpretation of results. To put our results into perspective, let us examine the results and make several remarks on points of interest. Unless otherwise specified, our discussion is under the assumption of exact arithmetic by default. The second point below is of particular importance.

1. Without augmentation ($p = 0$), the obtained convergence rate of $\delta_k(Y)$, see (4.24) for $m = k$, reduces to $|\rho(\lambda_{k+1})/\rho(\lambda_k)|$ which is the same rate of the classic power iteration applied to $\rho(A)$ (see (3.3)).
2. With augmentation and $m = (p+1)k = rk$, the convergence rate of $\delta_k(Y)$, see (4.30), increases to $|\rho(\lambda_{rk+1})/\rho(\lambda_k)|$ — the same rate as if k is increased to rk during the power iteration (see (3.4)). This is particularly important since the only extra work required for such an acceleration is an RR on $(p+1)k$ vectors in place of an RR on k vectors.
3. The error bound (4.25) indicates that for appropriate values of p and q, Y can be made better than X, while X is the result of applying a q-step subspace iteration to an initial matrix. Since the subspace iteration itself is convergent under mild conditions, (4.25) guarantees, in exact arithmetic, a faster convergence of ARRABIT (i.e., Algorithm 3) under suitable conditions.

To improve the performance of Algorithm 3, we may choose a suitable polynomial accelerator ρ to enlarge the gap between $|\rho(\lambda_k)|$ and $|\rho(\lambda_{rk+1})|$, and select q to be as large as permissible by numerical stability. Ideally, such parameters should be chosen adaptively. These practical issues, however, will be left to be studied in another work along with many other practical issues.

Let us now take a close look at the two constants c_m and c'_m in (4.26) and (4.27), respectively, both taking the form of a three-term product in which only the first terms differ.

1. For fixed k, p and m, c_m and c'_m are solely determined by A and X_0, but independent of q — the number of power iterations applied to X_0 to produce X, see (4.19).

2. The first terms, $\Gamma_{k,m}(U^T X_0)$ and $\Theta_{k,m}(U^T X_0)$, should have reasonable sizes in generic cases when X_0 is randomly chosen. In the case where X_0 is already a good approximate basis for $\mathcal{R}(U_k)$, one can expect a significant decay in $\{\|u_j^T X_0\|\}$. In this case, most likely (4.21) holds and the second case of (4.28) applies.
3. When the eigenvalues of A are ordered in a descending order in absolute value, the second term $\Gamma_{k,m}(V)$ is less than one, see (4.15).
4. The third term $\|G_1^\dagger E_k\|_2$, however, presents a complicating factor. How this term behaves for $p > 0$ requires a scrutiny which will be the topic of Section 4.5.

Finally, we remark that all of our results point out that there exists a matrix $Y \in \mathbb{R}^{n\times k}$ in the augmented subspace $\mathcal{R}(K_p)$ (which is constructed from the matrix X) that, under reasonable conditions, will be a better approximate basis for $\mathcal{R}(U_k)$ than X is. It is known that the Ritz pairs produced by the RR procedure are optimal approximations to the eigenpairs of A from the input subspace (see [15] for example). Therefore, the derived bounds in this section should be attainable by the Ritz pairs generated by the ARR procedure.

4.5. Validity of G-Assumption. A key condition for our results is the so-called G-Assumption in (4.1), that requires the first m rows of G in (4.7) to be linearly independent. The larger m is, the better the convergence rate will be.

Let us examine the matrix G_1 defined in (4.22). To simplify notation, we redefine

$$C = \operatorname{diag}(d_{11}, \cdots, d_{mm})^{-1} U_m^T X_0$$

and rewrite

$$G_1 = [C \;\; \Lambda_m C \;\; \cdots \;\; \Lambda_m^p C] \in \mathbb{R}^{m\times(p+1)k}, \tag{4.35}$$

where Λ_m is the $m \times m$ leading block of Λ.

We first give a necessary condition for the m rows of G_1 to be linearly independent.

PROPOSITION 4.8. *Let integer $m \in [k+1, k+pk]$ for $p > 0$. The matrix G_1 defined in* (4.35) *has full rank m only if Λ_m has no more than k equal diagonal elements (i.e., Λ_m contains no eigenvalue of multiplicity greater than k).*

Proof. Without loss of generality, suppose that the first $k+1$ diagonal elements of Λ_m are all equal, i.e., $\lambda_1 = \lambda_2 = \cdots = \lambda_{k+1} = \alpha$. Then the first $k+1$ rows of G_1 is of the form $[C' \;\; \alpha C' \;\; \cdots \;\; \alpha^p C']$, where C' consists of the first $k+1$ rows of C. Since all column blocks are scalar multiples of C' which has k columns, the rank of G_1 is at most k. □

The fact that G_1 is built from C which has only k columns dictates that for the rank of G_1 to be greater than k, it is necessary that the maximum multiplicity in Λ_m must not exceed k. An interesting question then is the following: what happens if the maximum multiplicity in Λ_m is exactly k? For this question we present an answer for the case of $p = 1$ and $m = 2k$. In this case, when the maximum multiplicity in Λ_m is exactly k, then G_1 is nonsingular in a generic sense.

Let $m = 2k$, and let us do the partitioning

$$C = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}, \quad \Lambda_m = \begin{bmatrix} \Lambda_1 & \\ & \Lambda_2 \end{bmatrix}, \quad G_1 = \begin{bmatrix} C_1 & \Lambda_1 C_1 \\ C_2 & \Lambda_2 C_2 \end{bmatrix}, \tag{4.36}$$

where C_j, Λ_j, $j = 1, 2$, are all $k \times k$ submatrices. Recall that Λ_1 consists of the first k eigenvalues of A and Λ_2 the next k eigenvalues.

PROPOSITION 4.9. *Let $p = 1$, $m = 2k$, and C, Λ_m and G_1 be defined as in* (4.36)*. Let r be the maximum multiplicity in Λ_m. Assume that any $k \times k$ submatrix of C is nonsingular. Then G_1 is nonsingular for $r = k$.*

Proof. We will show that when λ_1 or λ_{k+1} has multiplicity k, then G_1 is nonsingular. All the other cases can be similarly proven with appropriate permutations before partitioning (4.36) is done. First, the nonsingularity of G_1 is equivalent to that of

$$\begin{bmatrix} C_1 & \Lambda_1 C_1 \\ C_2 & \Lambda_2 C_2 \end{bmatrix} \begin{bmatrix} C_1^{-1} & \\ & C_1^{-1} \end{bmatrix} = \begin{bmatrix} I & \Lambda_1 \\ C_2 C^{-1} & \Lambda_2 C_2 C^{-1} \end{bmatrix} = \begin{bmatrix} I & \Lambda_1 \\ F & \Lambda_2 F \end{bmatrix},$$

where $F = C_2 C_1^{-1}$ is nonsingular by our assumption. By eliminating the (2,1)-block, we obtain a block upper triangular matrix in which the (2,2)-block is $\Lambda_2 F - F\Lambda_1$. Hence, the nonsingularity of G_1 is equivalent to that of $F\Lambda_1 - \Lambda_2 F$, or in turn equivalent to that of

$$K = \Lambda_1 - F^{-1}\Lambda_2 F. \tag{4.37}$$

If the multiplicity of λ_1 is k (implying that $\Lambda_1 = \lambda_k I$), then $K = F^{-1}(\lambda_k I - \Lambda_2)F$. On the other hand, if the multiplicity of λ_{k+1} is k (implying that $\Lambda_2 = \lambda_{k+1} I$), then $K = \Lambda_1 - \lambda_{k+1} I$. In either case, K is nonsingular since $\lambda_{k+1} < \lambda_k$; hence, so is G_1. ∎

In Proposition 4.9, we assume that every $k \times k$ submatrix of C is nonsingular. It is well-known that for a generic random matrix C, this assumption holds with high probability. Therefore, in a generic setting G_1 is nonsingular with high probability.

Intuitively, the more variance exists in Λ_m, the more likely that G_1 will have full row rank m. However, this remains unproven for the case of maximum multiplicity $r < k$. To examine this case, let us rewrite K in (4.37) into a sum of two matrices,

$$K = (\Lambda_1 - \lambda_k I) + F^{-1}(\lambda_k I - \Lambda_2)F.$$

The first is diagonal and positive semidefinite, and the second has positive eigenvalues when $\lambda_k > \lambda_{k+1}$, but is generally asymmetric. So far, we have not been able to find a result that guarantees nonsingularity for such a matrix K. However, in a generic setting where C and diagonal Λ are random matrices, nonsingularity should be expected with high probability (which has been empirically confirmed by our numerical experiments).

It should be noted that G_1 being nonsingular with $m = (p+1)k$ represents the best scenario where the acceleration potential of p-block augmentation is fully realized. However, $m < (p+1)k$ does not represent a failure, considering the fact that as long as $m > k$, an acceleration is still realized to some extent. Practically speaking, what is really relevant is the condition number of G_1 rather than its nonsingularity.

Once it is established for $p = 1$ and $m = 2k$ that in a generic setting G_1 is nonsingular whenever the maximum multiplicity of Λ_m is less than or equal to k, the same result can in principle be extended to the case of $p = 3$ by considering

$$G_1 = \begin{bmatrix} C & \Lambda C & \Lambda^2 C & \Lambda^3 C \end{bmatrix} = \begin{bmatrix} [C & \Lambda C] & \Lambda^2 [C & \Lambda C] \end{bmatrix} = [\hat{C} \;\; \hat{\Lambda}\hat{C}],$$

where $\hat{C} = [C \;\; \Lambda C]$ and $\hat{\Lambda} = \Lambda^2$, which has the same form as for the case $p = 1$. It will also cover the case of $p = 2$ where the matrix involved is a submatrix of the one for $p = 3$.

5. Numerical Results. In this section, we conduct proof-of-concept numerical experiments on Algorithm 3 (ARRABIT) to examine the tightness of inequality (4.33), that is, $\delta_k(Y) \leq \Psi_k(p, q)$,with various parameter values on both random and deterministic matrices. Our measure $\delta_k(Y)$ is also compared with two other standard measures

$$\pi_k(Y) = \tan\langle \mathcal{R}(U_k), \mathcal{R}(Y) \rangle \quad \text{and} \quad \nu_k(Y) = \frac{\|U_{k+}^T Y\|_2}{\|U_k^T Y\|_2},$$

where $\langle \cdot, \cdot \rangle$ is the angle between two subspaces.

For simplicity of experiments, we apply a simple polynomial function

$$\rho(A) = A^5, \tag{5.1}$$

to test matrices A that are chosen to be positive definite unless otherwise specified. Since $\rho(A)^q = A^{5q}$, in this case the effect of polynomial filtering can be absorbed into the power. For the sake of generality, however, we choose to keep these two items separate. Indeed, the performance of ARRABIT can be made much better if more sophisticated polynomials such as Chebyshev polynomials are judiciously used.

It is well-known that too large of a q-value can make $(A^{5q})X$ lose numerical rank. In our experiments, we choose the power q in Step 4 of Algorithm 3 after doing some trial and error in advance to avoid numerical difficulties. In addition, we normalize each column of X once it is multiplied by A to help enhance numerical stability.

Let (x_i, μ_i), $i = 1, 2, \cdots, k$, be computed Ritz pairs where $x_i^T x_j = \delta_{ij}$. We terminate the algorithm when the following maximum relative residual norm becomes smaller than a tolerance 10^{-12}, i.e.,

$$\texttt{maxres} \triangleq \max_{i=1,\ldots,k} \left\{ \frac{\|Ax_i - \mu_i x_i\|_2}{\max(1, |\mu_i|)} \right\} \leq 10^{-12}. \tag{5.2}$$

All numerical experiments are performed in MATLAB on a MacBook Pro computer with a Intel Core i7 (2.5 GHZ) CPU and 16GB memory.

5.1. Experiments on Random Matrices. We first examine the inequalities (4.24) and (4.25), specifically, the following five quantities:

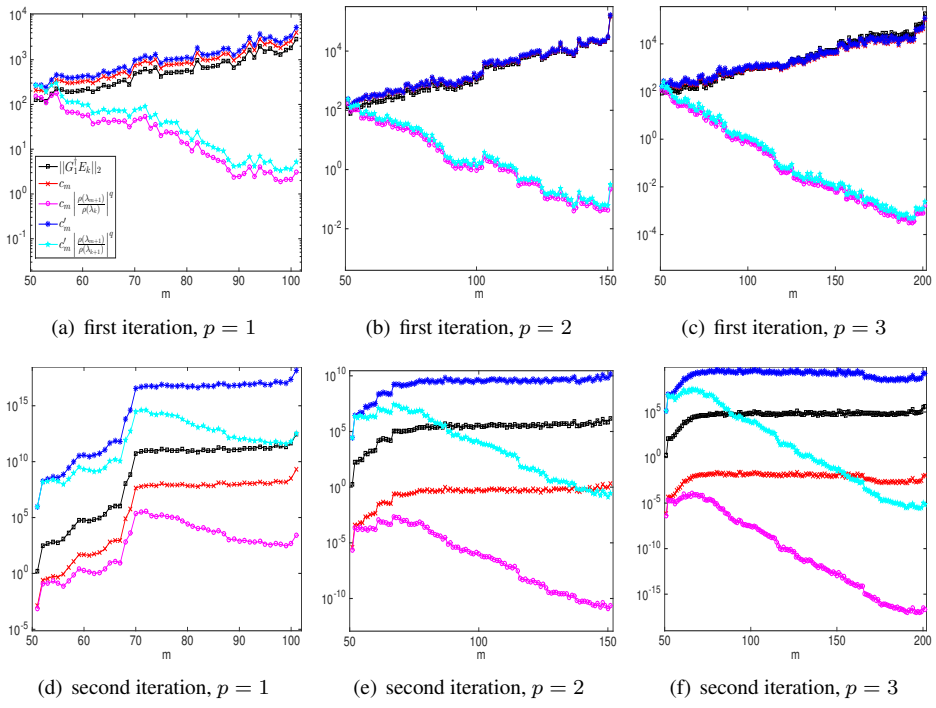
$$\|G_1^\dagger E_k\|_2, \;\; c_m, \;\; c'_m, \;\; c_m \left| \frac{\rho(\lambda_{m+1})}{\rho(\lambda_k)} \right|^q, \;\; c'_m \left| \frac{\rho(\lambda_{m+1})}{\rho(\lambda_{k+1})} \right|^q, \tag{5.3}$$

at either the first or the second iteration of ARRABIT. We note that all five quantities are m-dependent (though not explicit in the first one); and the last two are the right-hand sides of (4.24) and (4.25), respectively.

In this set of experiments, we generate matrices of the form $A = V \operatorname{diag}(s) V^\top$ where V is an orthonormalization of an $n \times n$ random matrix whose entries are i.i.d. standard Gaussian, and $s \in \mathbb{R}^n$ is also i.i.d. standard Gaussian whose elements are sorted into a descending order. Throughout the tests, we set $n = 1000$, $k = 50$ (the number of eigenpairs), and $q = 15$ (the number of power steps), and vary p (the number of augmentation blocks) from 1 to 3. Figure 5.1 shows the values of the above five quantities on a typical random instance for $m = k, k+1, \ldots, (p+1)k$. The following observations are now in order:

- The top three plots in Figure 5.1 indicate that at the first iteration, which starts from a random initial X_0, the coefficients c_m and c'_m are basically dominated by the term $\|G_1^\dagger E_k\|_2$ which tends to increase as m increases. On the other hand, the two right-hand sides tend to decrease as m increases, and the decay rate improves as the p value increases.
- At the second iteration, where the initial X_0 is no longer random, c_m is smaller than and c'_m larger than $\|G_1^\dagger E_k\|_2$, by approximately a uniform factor for all m values in either case. Consequently, the right-hand side of (4.24) is smaller than that of (4.25) by approximately a constant factor all m values. These results suggest that it is more difficult to make improvements at the second iteration than at the first one, which appears intuitively reasonable.

- For the case of $p = 1$, error bound (4.25) loses its meaningfulness since c'_m values are large and the right-hand side becomes greater than 1 (and similar situation occurs to (4.24) as well for most m values). Once p is increased to 2 or 3, the right-hand sides of both (4.24) and (4.25) behave as expected.
- Normally, the minima of the right-hand sides of both (4.24) and (4.25) occur at or near the end where $m = k + pk$, which validates the rate of convergence results (4.24) and (4.25) in Corollary 4.6.



(a) first iteration, $p = 1$ (b) first iteration, $p = 2$ (c) first iteration, $p = 3$

(d) second iteration, $p = 1$ (e) second iteration, $p = 2$ (f) second iteration, $p = 3$

FIG. 5.1. *The five quantities in* (5.3) *for* $m \in \{k, k+1, \cdots, (p+1)k\}$ *on a typical random matrix.*

In order to see the distributions of relevant quantities involved in the right-hand side of (4.33), i.e., $\Psi_k(p, q)$, we run ARRABIT on 1000 random instances and present statistics for 9 quantities given in Table 5.1. Three of these quantities are the 3 factors that define c_m, see (4.26). Recall that $\Psi_k(p, q)$ is the minimum value over $m \in [k, rk]$. The values of the first five m-dependent quantities in Table 5.1 are corresponding to the m that gives $\Psi_k(p, q)$. The last three quantities in Table 5.1, $\delta_k(Y)$, $\nu_k(Y)$ and $\pi_k(Y)$, are the three accuracy measures of Y as an approximate basis for $\mathcal{R}(U_k)$. Finally, X_0 and Y refer to the input and output matrices, respectively, at each outer iteration of ARRABIT.

Table 5.1 gives the minimum, mean and maximum values of the 9 quantities at the first and second ARRABIT iterations over 1000 replications for $p = 1, 2, 3$. In addition, Figure 5.2 presents histograms of $\log_{10}(\|G_1^\dagger E_k\|_2)$ and $\log_{10}(c_m)$ over these 1000 replications. From these results, we can make several observations:

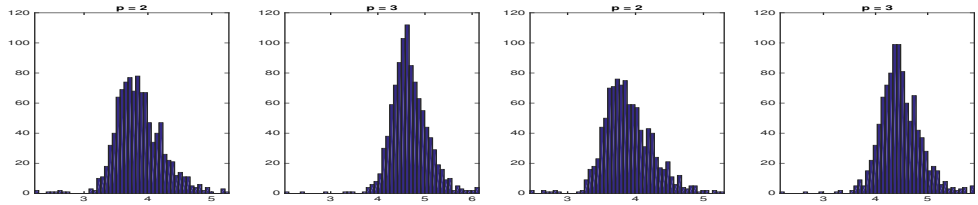
- the constants c_m remain moderate in size at the first two ARRABIT iterations;
- the error bound (4.33) becomes tighter as approximate solutions become more accurate (but before the effects of roundoff errors kick in);
- the two accuracy measures $\delta_k(Y)$ and $\nu_k(Y)$ are of the same order; on the other hand, $\pi_k(Y)$ is larger than $\nu_k(Y)$ when Y is far from $\mathcal{R}(U_k)$, but essentially coin-

cides with $\nu_k(Y)$ as soon as Y gets closer to $\mathcal{R}(U_k)$;

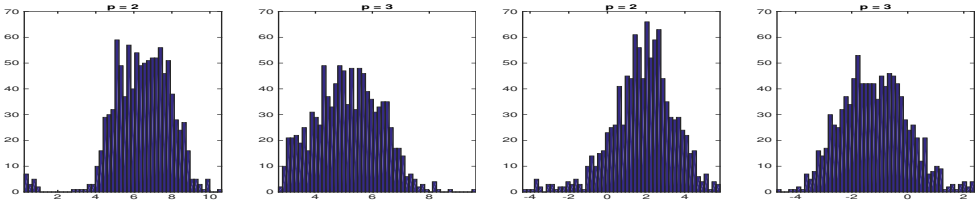
- for $p = 3$, on average two iterations are enough for ARRABIT to achieve an accuracy of $\delta_k(Y) < 6 \times 10^{-7}$; in favorable cases one iteration is sufficient to reach an accuracy of $\delta_k(Y) < 6 \times 10^{-8}$.

TABLE 5.1
Statistics of 9 quantities over 1000 *random replications. (In some cases* $q < 15$ *is used due to numerical issues.)*

	$\|G_1^\dagger E_k\|_2$	$\Gamma_{k,m}(U^T X_0)$	$\Gamma_{k,m}(V)$	c_m	$\left\lvert\frac{\rho(\lambda_{m+1})}{\rho(\lambda_k)}\right\rvert^q$	$\Psi_k(p,q)$	$\delta_k(Y)$	$\nu_k(Y)$	$\pi_k(Y)$
first iteration, $p = 1$									
min	5.0e+01	1.4e+00	8.3e-01	8.8e+01	2.6e-05	2.8e-02	4.0e-05	7.6e-05	7.6e-05
mean	1.4e+03	1.7e+00	8.8e-01	2.2e+03	1.9e-02	3.3e+01	2.0e-02	2.8e-02	3.3e-02
max	5.7e+04	2.2e+00	1.0e+00	8.3e+04	9.5e-01	4.5e+03	3.8e+00	9.9e-01	6.1e+00
first iteration, $p = 2$									
min	1.7e+02	1.5e+00	5.3e-01	2.8e+02	7.5e-09	7.1e-05	1.4e-07	4.6e-07	4.6e-07
mean	1.1e+04	1.7e+00	6.3e-01	1.2e+04	2.5e-03	1.1e+01	2.5e-04	6.9e-04	6.9e-04
max	1.9e+05	2.1e+00	1.0e+00	2.0e+05	9.7e-01	2.6e+03	1.9e-02	3.8e-02	3.8e-02
first iteration, $p = 3$									
min	1.1e+02	1.5e+00	2.8e-01	1.5e+02	1.7e-11	2.8e-08	5.7e-08	1.7e-07	1.7e-07
mean	7.6e+04	1.7e+00	3.7e-01	4.8e+04	1.5e-03	2.9e+00	1.0e-05	3.6e-05	3.6e-05
max	1.4e+06	2.2e+00	9.9e-01	7.9e+05	9.5e-01	4.2e+02	1.0e-03	3.2e-03	3.2e-03
second iteration, $p = 1$									
min	1.4e+00	4.0e-05	8.2e-01	5.7e-05	5.6e-09	1.6e-06	1.2e-09	2.2e-09	2.2e-09
mean	1.2e+06	1.7e-02	9.7e-01	2.8e+03	6.0e-01	1.5e-01	1.8e-03	1.9e-03	1.9e-03
max	3.6e+08	3.8e+00	1.0e+00	2.1e+05	1.0e+00	3.7e+01	2.1e-01	2.1e-01	2.1e-01
second iteration, $p = 2$									
min	1.7e+00	3.8e-08	5.3e-01	4.2e-05	7.2e-14	2.0e-16	4.1e-13	8.0e-13	8.0e-13
mean	1.5e+08	1.9e-04	6.2e-01	6.4e+03	1.5e-02	4.6e-05	1.9e-05	2.0e-05	2.0e-05
max	4.1e+10	6.5e-03	1.0e+00	6.5e+05	9.9e-01	1.1e-02	5.3e-03	5.3e-03	5.3e-03
second iteration, $p = 3$									
min	5.1e+02	1.2e-08	2.8e-01	2.2e-05	1.7e-17	5.1e-22	7.3e-14	9.3e-14	8.9e-14
mean	7.5e+06	9.1e-06	3.6e-01	2.7e+00	1.2e-12	1.8e-10	5.2e-07	7.4e-07	7.4e-07
max	4.4e+09	1.0e-03	4.6e-01	2.4e+02	7.1e-10	1.7e-07	1.9e-04	4.0e-04	4.0e-04



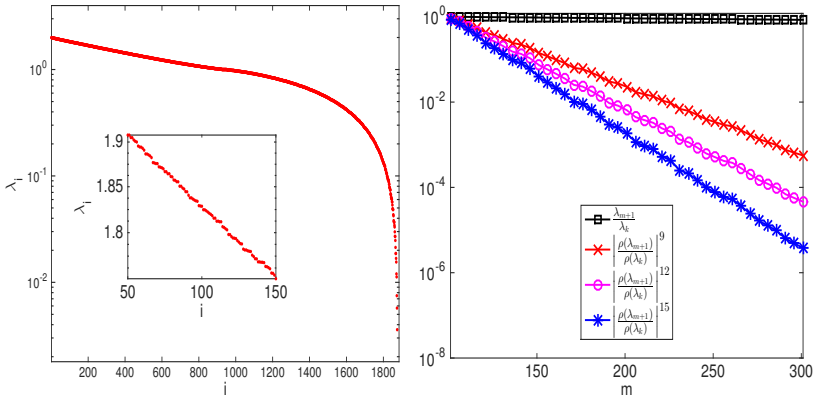
(a) $\log_{10}(\|G_1^\dagger E_k\|_2)$, first iteration (b) $\log_{10}(c_m)$, first iteration



(c) $\log_{10}(\|G_1^\dagger E_k\|_2)$, second iteration (d) $\log_{10}(c_m)$, second iteration

FIG. 5.2. *Histograms of* $\log_{10}(\|G_1^\dagger E_k\|_2)$ *and* $\log(c_m)$ *where* m *is corresponding to* $\Psi_k(p,q)$.

5.2. Experiments on A Deterministic Matrix. In this subsection, we use a test matrix that is the finite difference Laplacian on an L-shaped domain generated by the MATLAB command: `A = delsq(numgrid('L',52))`. The resulting A is symmetric positive definite of dimensionality $n = 1875$. We show the spectrum of A in Figure 5.3(a), and plot four types of spectral ratios in Figure 5.3(b): $|\lambda_{m+1}/\lambda_k|$ and $|\rho(\lambda_{m+1})/\rho(\lambda_k)|^q$ for $q = 9, 12, 15$ and $k = 100$. Since there is no significant decay in the first a few hundred eigenvalues, the first spectral ratio $|\lambda_{m+1}/\lambda_k|$ is close to 1 as m varies from 100 to 300. The other three ratios, after polynomial transformations and with a suitable power q, can be made much smaller than 1 as m increases.



(a) Spectrum of (A) (b) Spectral ratios with $k = 100$ and $m \geq k$

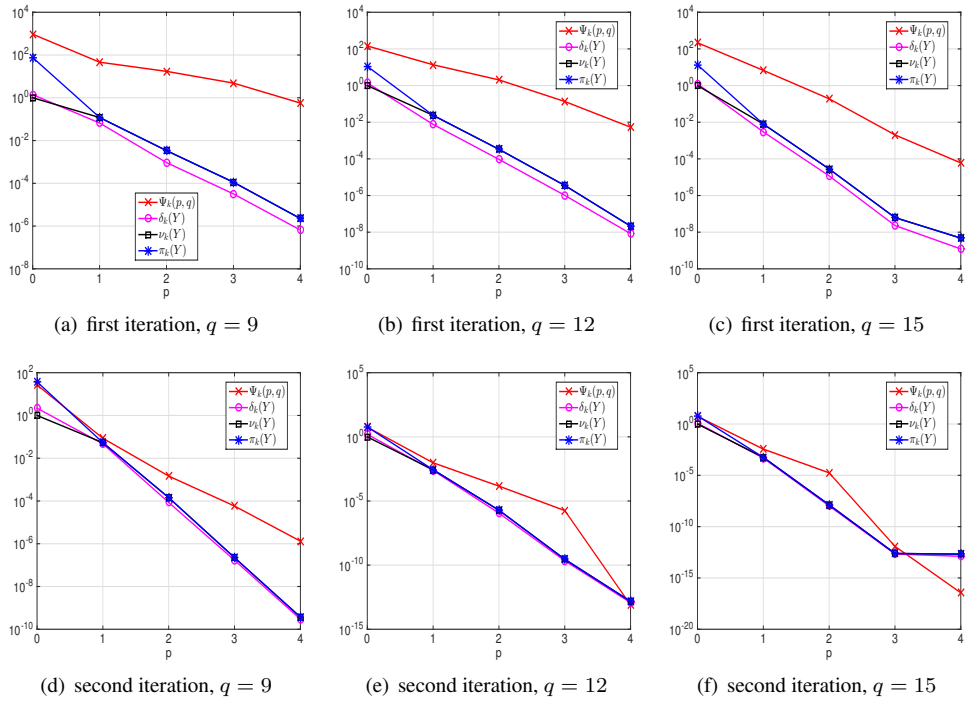
FIG. 5.3. *spectral information of the delsq matrix with* $n = 1875$

In our experiments, we focus our attention to investigating the tightness of the error bound (4.33), that is, $\delta_k(Y) \leq \Psi_k(p, q)$. We plot the left and the right hand sides with different parameter values p, q, k at different iterations. Specifically, the parameter ranges are $p \in \{0, 1, \ldots, 4\}$, $q \in \{3, 6, 9, 12, 15\}$ and $k \in \{50, 100, 150, \ldots, 300\}$, although only a subset of the combinations are tested with results given in Figures 5.4-5.8.

We first mention a special case in Figure 5.4 where two other accuracy measures $\pi_k(Y)$ and $\nu_k(Y)$ are included in addition to $\delta_k(Y)$. The results show that the three measures are very close to each other, especially when Y is close to $\mathcal{R}(U_k)$. For this reason, we exclude $\pi_k(Y)$ and $\nu_k(Y)$ from all subsequent tests.

Now we make several observations based on Figures 5.4-5.8.

- The error bound (4.33) holds in all the tests except in two cases where roundoff errors appear to have prevented $\delta_k(Y)$ from going below its corresponding $\Psi_k(p, q)$ value that is near the machine epsilon.
- In all the tests, the error bound (4.33) becomes considerably tighter at the second iteration than at the first one.
- When there is augmentation (i.e., $p > 0$), the bound $\Psi_k(p, q)$ in (4.33) always decreases as any one of the parameters p, q and k increases within its range.
- When p, q or k are suitably chosen, a single ARRABIT iteration is often sufficient for $\delta_k(Y)$ to reach an accuracy of 10^{-6} or below on this particular test problem.
- The acceleration provided by ARR over the plain RR is best illustrated in Figure 5.6 (just compare the two plots with $p = 0$ with the other four plots with $p > 0$).



(a) first iteration, $q = 9$ (b) first iteration, $q = 12$ (c) first iteration, $q = 15$

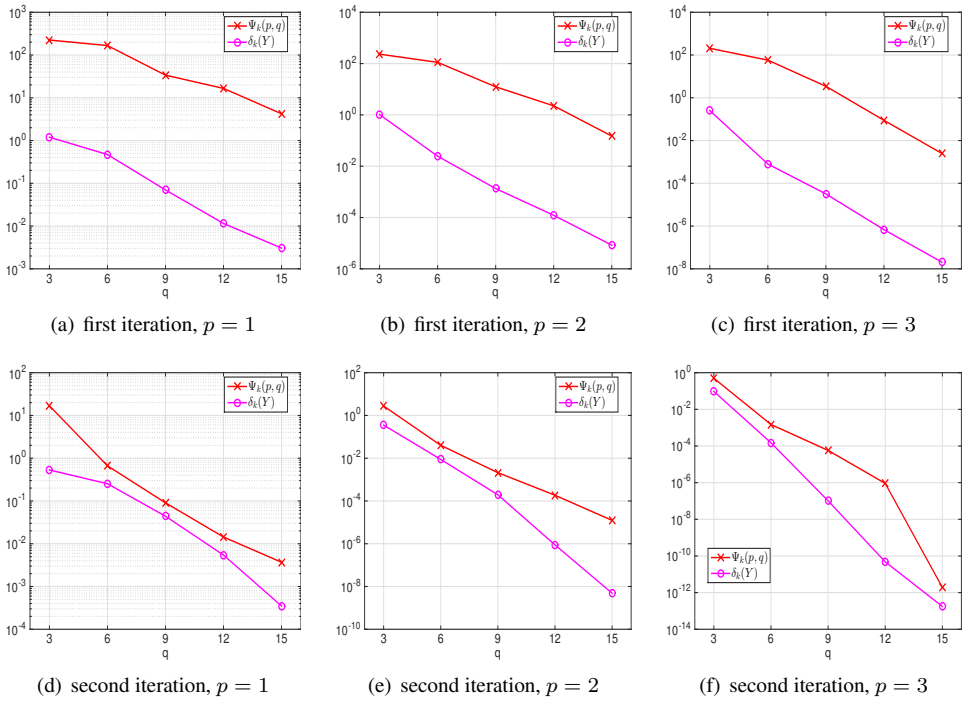
(d) second iteration, $q = 9$ (e) second iteration, $q = 12$ (f) second iteration, $q = 15$

FIG. 5.4. *$\Psi_k(p,q)$ and $\delta_k(Y)$ versus p at two iterations with $q = 9, 12, 15$ and $k = 100$*

5.3. Parameter Selections. Our results reveal that the convergence rate of ARRABIT is tightly bounded by the spectral ratio $(|\rho(\lambda_{k+1+pk})|/|\rho(\lambda_k)|)^q$ and a few constants that are not controllable by us. The smaller the ratio is, the faster is the convergence in exact arithmetic. For a given k, the selectable parameters are the polynomial function $\rho(\cdot)$, the number of augmentation block p and the power q. All these parameters need to be chosen and synthesized carefully, and ideally in an adaptive manner. Once $\rho(\cdot)$ and p are chosen, to make the spectral ratio as small as permissible by numerical stability, a sensible scheme for selecting q is to keep increasing q until the matrix $\rho(A)^q X$ becomes sufficiently badly conditioned, implying that the size of the spectral ratio is near the level of roundoff errors.

6. Concluding Remarks. This paper is a first step towards constructing a block algorithm of high scalability suitable for computing relatively large numbers of exterior eigenpairs for large-scale matrices on modern computers. Our strategy is simple: to reduce as much as possible the number of Rayleigh-Ritz projections (RR calls) or, in other words, to shift as much as possible computation burdens to subspace update (SU) steps. This strategy is based on the considerations that RR steps perform small dense eigenvalue decompositions, as well as basis orthogonalizations, thus possessing limited concurrency; on the other hand, SU steps can be accomplished by block operations like A times X, thus more scalable.

To reach for maximal concurrency, we choose the classic subspace iteration for subspace updating. It is well known that the convergence of the subspace iteration can be excessively slow, preventing it from being widely used to drive general-purpose eigensolvers. Therefore, the key to success reduces to whether one could accelerate the subspace iteration sufficiently and reliably to an extent so that it can compete in speed with Krylov subspace methods in general. In our analysis, we show that an effective acceleration is provably accomplishable



(a) first iteration, $p = 1$ (b) first iteration, $p = 2$ (c) first iteration, $p = 3$

(d) second iteration, $p = 1$ (e) second iteration, $p = 2$ (f) second iteration, $p = 3$

FIG. 5.5. *$\Psi_k(p,q)$ and $\delta_k(Y)$ versus q at two iteration with $p = 1, 2, 3$ and $k = 100$*

through the use of an augmented Rayleigh-Ritz (ARR) procedure, preferably coupled with polynomial accelerations in practice. The resulting prototype algorithm combining ARR and subspace iteration is named ARRABIT, which uses A only in matrix multiplications. Our main theoretical results appear in Theorem 4.5 and its corollaries. Numerical tests are performed to check the tightness of the derived error bounds on random and deterministic matrices. Among other things, the tests indicate that it is possible for ARRABIT to use only two or three ARR projections to reach a good solution accuracy, even when the number of augmentation blocks is limited to only 1 or 2.

There are a number of future directions worth pursuing from this point on. The foremost is a comprehensive implementation of ARRABIT and its numerical verifications (see [25] for an initial work in this direction). Software development is also important. The present work has laid a solid foundation for these and other future activities.

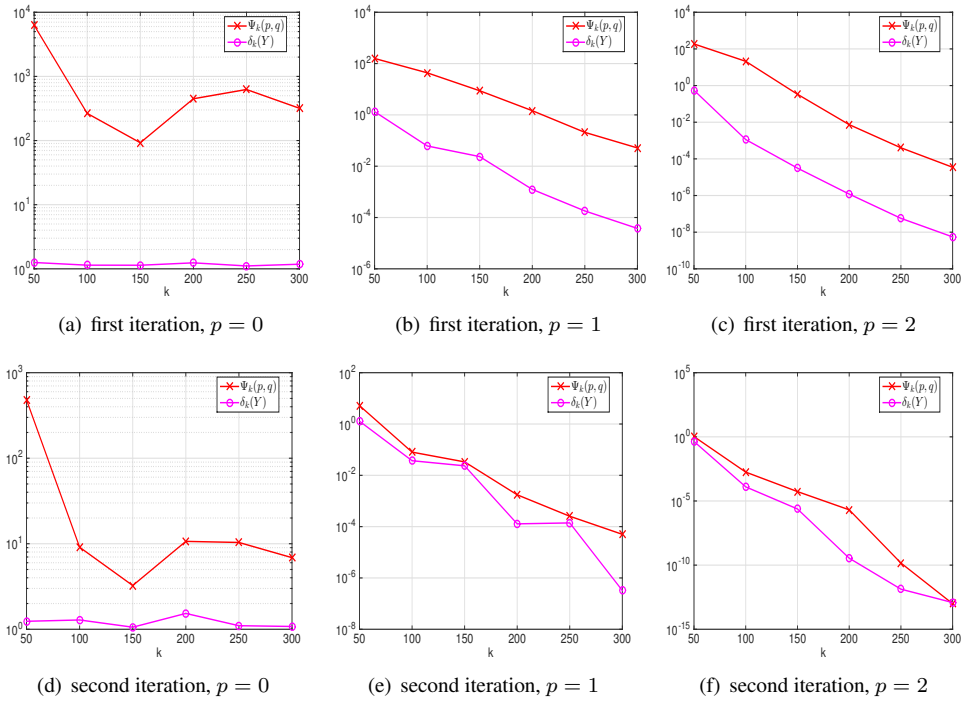
Acknowledgements. The authors would like to thank Chao Yang, Zhaojun Bai and Daniel Kressner for valuable discussions on eigenvalue computation.

REFERENCES

[1] Z. BAI, J. DEMMEL, J. DONGARRA, A. RUHE, AND H. VAN DER VORST, *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.

[2] M. BOLLHÖFER AND Y. NOTAY, *JADAMILU: a software code for computing selected eigenvalues of large sparse symmetric matrices*, Comput. Phys. Comm., 177 (2007), pp. 951–964.

[3] J. CULLUM AND W. DONATH, *A block lanczos algorithm for computing the q algebraically largest eigenvalues and a corresponding eigenspace of large, sparse, real symmetric matrices*, in Decision and Control



(a) first iteration, $p = 0$ (b) first iteration, $p = 1$ (c) first iteration, $p = 2$

(d) second iteration, $p = 0$ (e) second iteration, $p = 1$ (f) second iteration, $p = 2$

FIG. 5.6. *$\Psi_k(p, q)$ and $\delta_k(Y)$ versus k at two iterations with $p = 0, 1, 2$ and $q = 9$*

including the 13th Symposium on Adaptive Processes, 1974 IEEE Conference on, Nov 1974, pp. 505–509.

[4] J. W. DEMMEL, *Applied Numerical Linear Algebra*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997.

[5] G. H. GOLUB AND R. UNDERWOOD, *The block Lanczos method for computing eigenvalues*, in Mathematical software, III (Proc. Sympos., Math. Res. Center, Univ. Wisconsin, Madison, Wis., 1977), no. 39, Academic Press, New York, 1977, pp. 361–377.

[6] G. H. GOLUB AND C. F. VAN LOAN, *Matrix computations*, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, third ed., 1996.

[7] R. G. GRIMES, J. G. LEWIS, AND H. D. SIMON, *A shifted block Lanczos algorithm for solving sparse symmetric generalized eigenproblems*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 228–272.

[8] A. V. KNYAZEV, *Toward the optimal preconditioned eigensolver: locally optimal block preconditioned conjugate gradient method*, SIAM J. Sci. Comput., 23 (2001), pp. 517–541.

[9] C. LANCZOS, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Nat'l Bur. Std., 45 (1950), pp. 225–282.

[10] R. M. LARSEN, *Lanczos bidiagonalization with partial reorthogonalization*, Aarhus University, Technical report, DAIMI PB-357, September 1998.

[11] R. B. LEHOUCQ, *Implicitly restarted Arnoldi methods and subspace iteration*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 551–562.

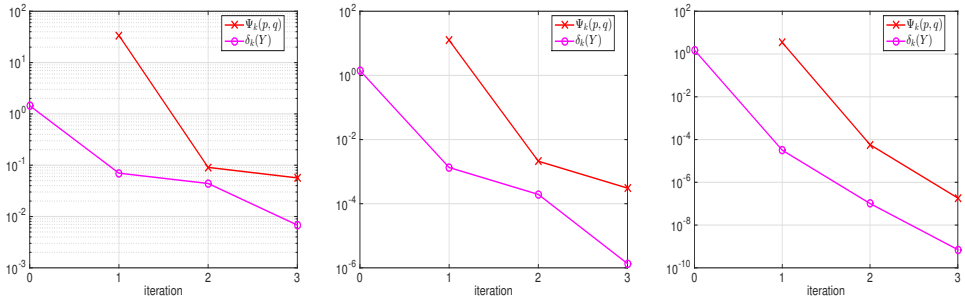
[12] R. B. LEHOUCQ, D. C. SORENSEN, AND C. YANG, *ARPACK users' guide: Solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*, vol. 6 of Software, Environments, and Tools, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1998.

[13] R.-C. LI AND L.-H. ZHANG, *Convergence of the block lanczos method for eigenvalue clusters*, Numer. Math., 131 (2015), pp. 83–113.

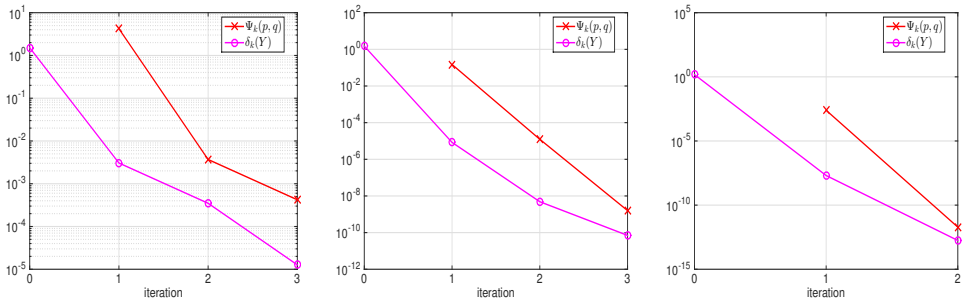
[14] X. LIU, Z. WEN, AND Y. ZHANG, *Limited memory block krylov subspace optimization for computing dominant singular value decompositions*, SIAM Journal on Scientific Computing, 35-3 (2013), pp. A1641–A1668.

[15] B. PARLETT, *The Symmetric Eigenvalue Problem*, Prentice-Hall, 1980.

[16] H. RUTISHAUSER, *Computational aspects of F. L. Bauer's simultaneous iteration method*, Numer. Math., 13 (1969), pp. 4–13.



(a) $p = 1, q = 9$ (b) $p = 2, q = 9$ (c) $p = 3, q = 9$



(d) $p = 1, q = 15$ (e) $p = 2, q = 15$ (f) $p = 3, q = 15$

FIG. 5.7. *Iteration history of* $\Psi_k(p,q)$ *and* $\delta_k(Y)$ *with various* p, q *and* $k = 100$

[17] H. RUTISHAUSER, *Simultaneous iteration method for symmetric matrices*, Numer. Math., 16 (1970), pp. 205–223.

[18] Y. SAAD, *On the rates of convergence of the lanczos and the block-lanczos methods*, SIAM J. Numer. Anal., 17 (1980), pp. 687–706.

[19] Y. SAAD, *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press, 1992.

[20] D. C. SORENSEN, *Implicitly restarted Arnoldi/Lanczos methods for large scale eigenvalue calculations*, in Parallel numerical algorithms (Hampton, VA, 1994), vol. 4 of ICASE/LaRC Interdiscip. Ser. Sci. Eng., Kluwer Acad. Publ., 1996, pp. 119–165.

[21] A. STATHOPOULOS AND C. F. FISCHER, *A Davidson program for finding a few selected extreme eigenpairs of a large, sparse, real, symmetric matrix*, Computer Physics Communications, 79 (1994), pp. 268–290.

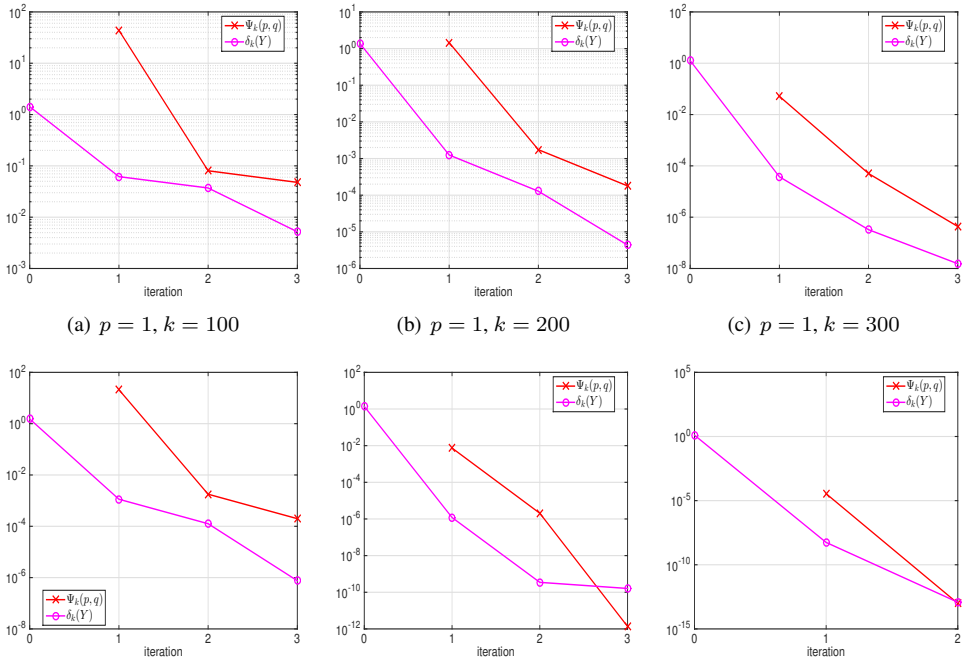
[22] G. W. STEWART, *Simultaneous iteration for computing invariant subspaces of non-Hermitian matrices*, Numer. Math., 25 (1975/76), pp. 123–136.

[23] ———, *Matrix algorithms Vol. II: Eigensystems*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2001.

[24] W. J. STEWART AND A. JENNINGS, *A simultaneous iteration algorithm for real matrices*, ACM Trans. Math. Software, 7 (1981), pp. 184–198.

[25] Z. WEN AND Y. ZHANG, *Block algorithms with augmented rayleigh-ritz projections for large-scale eigenpair computation*, tech. rep., arxiv: 1507.06078, 2015.

[26] Q. YE, *An adaptive block lanczos algorithm*, Numer. Algorithms, 12 (1996), pp. 97–110.



(a) $p = 1, k = 100$ (b) $p = 1, k = 200$ (c) $p = 1, k = 300$

(d) $p = 2, k = 100$ (e) $p = 2, k = 200$ (f) $p = 2, k = 300$

FIG. 5.8. *Iteration history of* $\Psi_k(p,q)$ *and* $\delta_k(Y)$ *with various* p, k *and* $q = 9$