An Implicitly Restarted Refined Bidiagonalization Lanczos Method for Computing a Partial Singular Value Decomposition.

ARTICLE *in* SIAM JOURNAL ON MATRIX ANALYSIS AND APPLICATIONS · JANUARY 2003

Impact Factor: 1.81 · DOI: 10.1137/S0895479802404192 · Source: DBLP

2 AUTHORS, INCLUDING:

Zhongxiao Jia
Tsinghua University
65 PUBLICATIONS **619** CITATIONS

SIAM J. MATRIX ANAL. APPL.
Vol. 25, No. 1, pp. 246–265

AN IMPLICITLY RESTARTED REFINED BIDIAGONALIZATION LANCZOS METHOD FOR COMPUTING A PARTIAL SINGULAR VALUE DECOMPOSITION*

ZHONGXIAO JIA† AND DATIAN NIU‡

Abstract. The bidiagonalization Lanczos method can be used for computing a few of the largest or smallest singular values and corresponding singular vectors of a large matrix, but the method may encounter some convergence problems. In this paper the convergence of the method is analyzed, showing why it may converge erratically and perhaps fail to converge. To correct this possible nonconvergence and improve the method, a refined bidiagonalization Lanczos method is proposed. The implicitly restarting technique due to Sorensen is applied to the method, and an implicitly restarted refined bidiagonalization Lanczos algorithm (IRRBL) is developed. A new selection of shifts is proposed for use within IRRBL, called refined shifts, and a reliable and efficient algorithm is developed for computing the refined shifts. Numerical experiments show that IRRBL can perform better than the implicitly restarted bidiagonalization Lanczos algorithm (IRBL) proposed by Larsen, in particular when the smallest singular triplets are desired.

Key words. singular value, singular vector, the bidiagonalization Lanczos method, Ritz value, Ritz vector, refined Ritz vector, the refined bidiagonalization Lanczos method, implicit restart, exact shifts, refined shifts, convergence

AMS subject classifications. 65F15, 15A18

PII. S0895479802404192

1. Introduction.

We are concerned with the following problem.

PROBLEM 1. *Compute numerically the k largest or smallest singular values and corresponding left and right singular vectors of a large real $M \times N$ matrix $A \in \mathcal{R}^{M\times N}$, where k is much smaller than M and N.*

Such a problem arises from many applications, e.g., total least squares problems, determination of numerical rank of a matrix, regression analysis, and image processing and pattern recognitions.

Without loss of generality, we assume that $M \geq N$ (otherwise we work on A^{T}, the transpose of A). Let σ_i, $i = 1, 2, \ldots, N$, be the singular values of A, labeled in decreasing or increasing order, and u_i and v_i the corresponding left and right singular vectors. The triplets (σ_i, u_i, v_i) are called the singular triplets of A. We then have the singular value decomposition (SVD) of A:

$$A = U \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} V^{\mathrm{T}} = U_1 \Sigma V^{\mathrm{T}}, \tag{1.1}$$

where $U = (u_1, u_2, \ldots, u_M) = (U_1, U_2)$ is orthogonal with $U_1 = (u_1, u_2, \ldots, u_N)$, $V = (v_1, v_2, \ldots, v_N)$ orthogonal, and $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_N)$.

Consider the augmented matrix

$$\tilde{A} = \begin{pmatrix} 0 & A \\ A^{\mathrm{T}} & 0 \end{pmatrix}. \tag{1.2}$$

*Received by the editors March 18, 2002; accepted for publication (in revised form) by H. van der Vorst December 17, 2002; published electronically May 29, 2003. This work was supported by Special Funds for State Major Basic Research Projects (G19990328).

http://www.siam.org/journals/simax/25-1/40419.html

†Department of Mathematical Sciences, Tsinghua University, Beijing 100084, People's Republic of China (zjia@math.tsinghua.edu.cn).

‡Department of Applied Mathematics, Dalian University of Technology, Dalian 116024, People's Republic of China.

It is easily verified that $\tilde{A}$ has the $2N$ eigenvalues $\pm\sigma_1, \ldots, \pm\sigma_N$ and $M - N$ eigenvalues zero. The eigenvectors associated with σ_i and $-\sigma_i$ are $\frac{1}{\sqrt{2}} \left(u_i^{\mathrm{T}}, v_i^{\mathrm{T}}\right)^{\mathrm{T}}$ and $\frac{1}{\sqrt{2}} \left(u_i^{\mathrm{T}}, -v_i^{\mathrm{T}}\right)^{\mathrm{T}}$, respectively, and the eigenvectors associated with the eigenvalues zero are $\left(u^{\mathrm{T}}, 0^{\mathrm{T}}\right)^{\mathrm{T}}$, where the u's are orthogonal to $u_1, \ldots, u_N$. Therefore, we get an eigenproblem equivalent to (1.1).

PROBLEM 2. *Compute numerically the k largest or smallest positive eigenvalues of $\tilde{A}$ and the associated eigenvectors.*

Since M and N are assumed to be large and the dimension of $\tilde{A}$ is $M + N$, only projection methods are reasonable to solve Problem 2. A typical method is the symmetric Lanczos method [26]. However, if the method is applied to solve Problem 2 directly and explicitly, then the computational complexity and the memory requirement will be greatly increased. So it is not preferable to work on $\tilde{A}$ directly. Another consequence of using $\tilde{A}$ explicitly is that the smallest positive eigenvalues of $\tilde{A}$ are now interior ones, while they are the leftmost (extreme) singular values of A. Note that the symmetric Lanczos method usually favors the extreme eigenvalues and the associated eigenvectors, and it is very difficult to compute interior eigenpairs [26]. Therefore, we should not work on $\tilde{A}$ directly for computing the smallest singular values of A.

Because of the mentioned drawbacks, we attempt to solve Problem 1 by working on $\tilde{A}$ implicitly. It will turn out that the bidiagonalization Lanczos method [4, 5, 9] and its refined version to be proposed in this paper can settle these problems elegantly.

Over the past decade, the implicit restarting technique due to Sorensen [27] has proven to be a powerful and efficient tool for restarting a Krylov subspace algorithm. It has been used in various contexts, e.g., [2, 3, 9, 14, 17, 28, 29, 30]. It may save computational cost considerably at each restart and maintain numerical stability. However, it should be kept in mind that for an overall performance one of the keys for the success of an implicitly restarted Krylov algorithm is reasonable selection of shifts involved [14, 17]. Other applications of the technique are possible. Björck, Grimme, and van Dooren [3] successfully applied the implicit restarting technique to the lower bidiagonalization Lanczos method for ill-posed least squares problems. Wang and Zha [30] proposed a variant of their algorithm for computing a few largest singular values of A. Both algorithms take zeros as shifts. Larsen [22] developed an implicitly restarted bidiagonalization Lanczos algorithm and discussed many issues, including selection of shifts and the maintenance of semiorthogonality of Lanczos vectors. A few packages are now available for computing a partial SVD of A, e.g., PROPACK and LANSO [21, 22] and ARPACK [23]. PROPACK works on A directly, and LANSO is a symmetric Lanczos algorithm with selective orthogonalization and solves the eigenproblem of $A^T A$ or $\tilde{A}$. Both packages work without restarting until the desired singular values and/or singular vectors have been found, while ARPACK solves the eigenproblems of $A^T A$ and $\tilde{A}$ whose MATLAB counterparts are eigs.m and svds.m, respectively.

The paper is organized as follows. In section 2, we describe the bidiagonalization Lanczos process, and we show how the process can be combined with the Rayleigh–Ritz procedure for computing a partial SVD of A. We then make a convergence analysis of approximate singular values (Ritz values) and approximate singular vectors (Ritz vectors). We show that, under the natural hypothesis that the deviations of a desired singular vector from a sequence of Krylov subspaces tend to zero, there is a Ritz value that converges to the desired singular value, while, on the other hand, the

associated Ritz vectors may converge erratically and even may fail to converge to the desired left and right singular vectors. In section 3, based on the refined projection methods for large matrix eigenproblems [28, 29] proposed by Jia [10, 12, 13, 15, 16], by exploiting the bidiagonalization Lanczos process we propose a refined bidiagonalization Lanczos method for Problem 1. The refined method has a different background from the standard method. The fundamental difference between the refined method and the standard method is that rather than using Ritz approximations, the former seeks new approximate singular vectors, called refined singular vector approximations or simply refined Ritz approximations, from certain Krylov subspaces that minimize the norms of certain residuals and use them to approximate the desired singular vectors. We analyze the convergence of refined Ritz approximations and show that they always converge, provided that the deviations tend to zero. In section 4, we review an implicitly restarted bidiagonalization Lanczos algorithm (IRBL) for Problem 1, in which the shifts are often selected as those unwanted approximate singular values (Ritz values) [21, 22], called exact shifts. In order to compute the large close singular values and improve performance, Larsen [22] proposed a simple adaptive shifting strategy that replaces bad shifts by zero. This strategy often appears to be quite effective. In section 5, motivated by Jia's work [14, 17], we discuss the selection of shifts involved in an implicitly restarted algorithm, and we propose a new shifts scheme, called refined shifts, for use within the implicitly restarted refined bidiagonalization Lanczos algorithm (IRRBL). Still, we exploit Larsen's adaptive shifting strategy to compute the large close singular values. We show qualitatively that the refined shifts are better than the exact shifts for use within IRBL. We discuss how to compute the refined shifts efficiently and reliably. However, Larsen's adaptive shifting strategy cannot work for computing the smallest close singular values. To this end, we give a heuristic analysis and propose to replace bad shifts by the largest Ritz value at the current cycle. In section 6 we make numerical experiments on several real-world problems, indicating that IRRBL can be more efficient than IRBL, in particular for computing the smallest singular triplets. To be complete, we also compare our algorithm with PROPACK, LANSO, and ARPACK and show the superiority of IRRBL. Finally, in section 7 we draw some conclusions.

Some notation to be used is introduced now. Throughout the paper, denote by $||\cdot||$ the Euclidean norm, by $\mathcal{K}_m(C, w_1) = span\{w_1, Cw_1, \ldots, C^{m-1}w_1\}$ the m-dimensional Krylov subspace generated by C and a unit length vector w_1, and by e_m the mth coordinate vector of dimension m.

2. The bidiagonalization Lanczos process and method.

2.1. The bidiagonalization Lanczos process.

We first describe the lower bidiagonalization Lanczos process due to Paige and Saunders [25], which is a variant of the upper bidiagonalization Lanczos process due to Golub and Kahan [7].

ALGORITHM 1. THE m-STEP BIDIAGONALIZATION LANCZOS PROCESS.

1. Start: Choose a unit length vector p_1 of dimension M, $\beta_1 = 1$ and let $q_0 = 0$.
2. For $i = 1, 2, \ldots, m$
   (a) $r_i = A^{\mathrm{T}} p_i - \beta_i q_{i-1}$
   $\alpha_i = ||r_i||, q_i = r_i/\alpha_i$
   (b) $z_i = Aq_i - \alpha_i p_i$
   $\beta_{i+1} = ||z_i||, p_{i+1} = z_i/\beta_{i+1}$
   Endfor

Define $Q_m = (q_1, q_2, \ldots, q_m)$ and $P_{m+1} = (p_1, p_2, \ldots, p_{m+1})$. Then Algorithm 1 can be written in matrix form

$$AQ_m = P_{m+1}B_m, \tag{2.1}$$

$$A^T P_{m+1} = Q_m B_m^{\mathrm{T}} + \alpha_{m+1} q_{m+1} e_{m+1}^{\mathrm{T}}. \tag{2.2}$$

Therefore, we have

$$P_{m+1}^T A Q_m = B_m, \tag{2.3}$$

where

$$B_m = \begin{pmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \beta_3 & \ddots & \\ & & \ddots & \alpha_m \\ & & & \beta_{m+1} \end{pmatrix} \in \mathcal{R}^{(m+1)\times m}$$

is called the projection matrix of A with the left subspace $span\{P_{m+1}\}$ and the right subspace $span\{Q_m\}$.

Note that the above three relations can also be written as

$$\begin{aligned} \tilde{A} \begin{pmatrix} P_{m+1} & 0 \\ 0 & Q_m \end{pmatrix} &= \begin{pmatrix} P_{m+1} & 0 \\ 0 & Q_m \end{pmatrix} \begin{pmatrix} 0 & B_m \\ B_m^{\mathrm{T}} & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ r_{m+1} e_m^{\mathrm{T}} & 0 \end{pmatrix} \\ &= \begin{pmatrix} P_{m+1} & 0 & 0 \\ 0 & Q_m & q_{m+1} \end{pmatrix} \begin{pmatrix} 0 & B_m \\ B_m^{\mathrm{T}} & 0 \\ \alpha_{m+1} e_m^{\mathrm{T}} & 0 \end{pmatrix}. \end{aligned} \tag{2.4}$$

In finite precision arithmetic, it is well known [26] that the orthogonality of P_{m+1} and Q_m, Lanczos basis vectors, may lose soon. In order to maintain numerical (semi)orthogonality, an efficient approach is to use a partial reorthogonalization. For details, refer to Larsen [21, 22].

It is known that there is a close relationship between the above bidiagonalization process and the symmetric Lanczos process applied to $A^T A$ and AA^{T}, both of which have the same nonzero eigenvalues σ_i^2, $i = 1, 2, \ldots, N$, as well as $\tilde{A}$. For details, see [4, 8, 21].

2.2. The bidiagonalization Lanczos method. Let θ_i, $i = 1, 2, \ldots, m$, be the singular values of B_m, and let w_i and s_i be the corresponding left and right singular vectors. Define

$$\tilde{u}_i = P_{m+1} w_i, \qquad \tilde{v}_i = Q_m s_i.$$

It follows from (2.1) and (2.2) that

$$A\tilde{v}_i = \theta_i \tilde{u}_i, \tag{2.5}$$

$$A^{\mathrm{T}} \tilde{u}_i = \theta_i \tilde{v}_i + \alpha_{m+1} q_{m+1} e_{m+1}^{\mathrm{T}} w_i. \tag{2.6}$$

Therefore, if $\alpha_{m+1} = 0$, then $(\theta_i, \tilde{u}_i, \tilde{v}_i)$, $i = 1, 2, \ldots, m$, are exact singular triplets of A. The bidiagonalization Lanczos method uses the triplets $(\theta_i, \tilde{u}_i, \tilde{v}_i)$ as approximate singular triplets of A. This is the way of achieving the Ritz–Galerkin process on

the Krylov subspaces $\mathcal{K}_m(A^TA, A^Tq_1)$ and $\mathcal{K}_{m+1}(AA^{\mathrm{T}}, q_1)$. So, the triplets $(\theta_i, \tilde{u}_i, \tilde{v}_i)$ are simply called Ritz approximations of singular triplets. Similar to the symmetric Lanczos method, the largest and smallest singular values of B_m converge usually rapidly to the largest and smallest singular values of A [4, 8, 21].

We claim an approximate triplet $(\theta_i, \tilde{u}_i, \tilde{v}_i)$ to have converged if

$$\sqrt{\|A\tilde{v}_i - \theta_i\tilde{u}_i\|^2 + \|A^{\mathrm{T}}\tilde{u}_i - \theta_i\tilde{v}_i\|^2} = \alpha_{m+1} \mid e_{m+1}^T w_i \mid \leq tol, \tag{2.7}$$

where *tol* is a user-prescribed tolerance. Therefore, we do not need to form the Ritz approximations $\tilde{u}_i, \tilde{v}_i$ explicitly until the convergence occurs.

We next show that the method is an orthogonal projection method that projects $\tilde{A}$ onto a suitable subspace. Define the subspace

$$E = span\left\{\begin{pmatrix} P_{m+1} & 0 \\ 0 & Q_m \end{pmatrix}\right\}. \tag{2.8}$$

Then it follows from (2.4), (2.5), and (2.6) that the pairs

$$(\theta_i, \tilde{\varphi}_i) = \left(\theta_i, \frac{1}{\sqrt{2}}\begin{pmatrix}\tilde{u}_i \\ \tilde{v}_i\end{pmatrix}\right), \qquad i = 1, 2, \ldots, m,$$

satisfy the orthogonal projection (Rayleigh–Ritz approximation)

$$\begin{cases} \tilde{\varphi}_i \in E, \\ \tilde{A}\tilde{\varphi}_i - \theta_i\tilde{\varphi}_i \perp E, \end{cases} \tag{2.9}$$

and the projection matrix is $\tilde{B} = \left(\begin{smallmatrix} 0 & B_m \\ B_m^{\mathrm{T}} & 0 \end{smallmatrix}\right)$. The $(\theta_i, \tilde{\varphi}_i)$ are part of the Ritz pairs of $\tilde{A}$ with respect to E.

Jia [11, 15] and Jia and Stewart [18, 19] have proved that, for a general matrix and a general projection subspace, the Ritz vectors may fail to converge. In the context of this paper, note that the spectral condition number of $\tilde{B}$ is always one. Then from Theorem 2.1 of [19], we can get the following simplified result.

THEOREM 2.1. *Define* $\varepsilon = \sin\angle\left(\left(\begin{smallmatrix} u \\ v \end{smallmatrix}\right), E\right)$ *and assume that* ε *is small enough. Then there is a matrix* F *satisfying*

$$||F|| \leq \frac{\varepsilon}{\sqrt{1-\varepsilon^2}}||A|| \tag{2.10}$$

such that σ *is an exact eigenvalue of*

$$\tilde{B}_m + F = \begin{pmatrix} 0 & B_m \\ B_m^{\mathrm{T}} & 0 \end{pmatrix} + F.$$

Furthermore, there exists a positive eigenvalue θ *of* $\tilde{B}_m$ *such that*

$$|\sigma - \theta| \leq ||F||. \tag{2.11}$$

This theorem shows that there is always a Ritz value θ that converges to a desired σ once the deviation ε of $(u^{\mathrm{T}}, v^{\mathrm{T}})^{\mathrm{T}}$ from E tends to zero.

Theorem 3.2 in [19] reduces to the following result.

THEOREM 2.2. *Let* $(\theta, \tilde{w}, \tilde{s})$ *be a singular triplet of* B_m*, and let* $(\tilde{w}, \tilde{W}_\perp)$ *and* $(\tilde{s}, \tilde{S}_\perp)$ *be orthogonal matrices such that*

$$\begin{pmatrix} \tilde{w}^{\mathrm{T}} \\ \tilde{W}_\perp^{\mathrm{T}} \end{pmatrix} B_m \left(\tilde{s}, \tilde{S}_\perp\right) = \begin{pmatrix} \theta & 0 \\ 0 & C \end{pmatrix}. \tag{2.12}$$

Define the matrix $\tilde{C} = \left(\begin{smallmatrix} 0 & C \\ C^{\mathrm{T}} & 0 \end{smallmatrix}\right)$, *and assume that* $\sigma I - \tilde{C}$ *is nonsingular. Let the separation of* σ *and the spectra of* C *be defined by*

$$\mathrm{sep}(\sigma, \tilde{C}) = ||(\sigma I - \tilde{C})^{-1}||^{-1}. \tag{2.13}$$

Then if

$$\mathrm{sep}(\sigma, \tilde{C}) \geq \mathrm{sep}(\theta, \tilde{C}) - \mid \theta - \sigma \mid > 0, \tag{2.14}$$

we have

$$\begin{aligned} \sin \angle \left(\begin{pmatrix} u \\ v \end{pmatrix}, \begin{pmatrix} \tilde{u} \\ \tilde{v} \end{pmatrix} \right) &\leq \left(1 + \frac{||A||}{\sqrt{1-\varepsilon^2}\mathrm{sep}(\sigma, \tilde{C})} \right) \varepsilon \\ &\leq \left(1 + \frac{||A||}{\sqrt{1-\varepsilon^2}(\mathrm{sep}(\theta, \tilde{C}) - |\theta - \sigma|)} \right) \varepsilon. \end{aligned} \tag{2.15}$$

Suppose that Algorithm 1 does not break down, i.e., $\alpha_{m+1} \neq 0$. Then B_m only has simple singular values, i.e., θ is different from the singular values of C in (2.12). As a consequence, assumption (2.14) holds with $\varepsilon \to 0$ as $\theta \to \sigma$. However, we must point out that $\mathrm{sep}(\theta, \tilde{C}) - \mid \theta - \sigma \mid$ can be arbitrarily near zero because C may have a singular value that is arbitrarily close to σ, though it is different from σ. Thus, the right-hand side of (2.15) may converge to zero erratically and even may not approach zero although $\varepsilon \to 0$, which means that the Ritz vector $(\tilde{u}^{\mathrm{T}}, \tilde{v}^{\mathrm{T}})^{\mathrm{T}}$ may converge erratically and even may not converge to $(u^{\mathrm{T}}, v^{\mathrm{T}})^{\mathrm{T}}$.

Next we establish an inequality on approximate left and right singular vectors $\tilde{u}$ and $\tilde{v}$.

THEOREM 2.3. *We have*

$$\sin^2 \angle(u, \tilde{u}) + \sin^2 \angle(v, \tilde{v}) \leq 2 \sin^2 \angle \left(\begin{pmatrix} u \\ v \end{pmatrix}, \begin{pmatrix} \tilde{u} \\ \tilde{v} \end{pmatrix} \right). \tag{2.16}$$

Proof. By definition, we obtain

$$\begin{aligned} \sin^2 \angle(u, \tilde{u}) + \sin^2 \angle(v, \tilde{v}) &= \min_{\alpha} ||u - \alpha\tilde{u}||^2 + \min_{\alpha} ||v - \alpha\tilde{v}||^2 \\ &\leq \min_{\alpha} (||u - \alpha\tilde{u}||^2 + ||v - \alpha\tilde{v}||^2) \\ &= \min_{\alpha} \left\| \begin{pmatrix} u \\ v \end{pmatrix} - \alpha \begin{pmatrix} \tilde{u} \\ \tilde{v} \end{pmatrix} \right\|^2 \\ &= 2 \min_{\alpha} \left\| \frac{1}{\sqrt{2}} \begin{pmatrix} u \\ v \end{pmatrix} - \frac{1}{\sqrt{2}} \alpha \begin{pmatrix} \tilde{u} \\ \tilde{v} \end{pmatrix} \right\|^2 \\ &= 2 \sin^2 \angle \left(\begin{pmatrix} u \\ v \end{pmatrix}, \begin{pmatrix} \tilde{u} \\ \tilde{v} \end{pmatrix} \right), \end{aligned}$$

which completes the proof. ∎

Combining Theorems 2.1–2.3, we conclude that under the natural hypothesis that $\varepsilon \to 0$ there is a Ritz value θ that converges to the desired singular value unconditionally, while the corresponding $\tilde{u}$ and $\tilde{v}$ may converge erratically and may even fail to converge to the desired left and right singular vectors u and v.

3. The refined bidiagonalization Lanczos method. As was seen previously, the bidiagonalization Lanczos method may have convergence problems for computing singular vectors. In order to correct this deficiency, we apply the principle of the refined eigenvector approximation advocated by Jia [10, 12] and popularized by Jia [13, 15, 16, 17] (also see [2, 28, 29]) to the bidiagonalization Lanczos method, and we propose a refined bidiagonalization Lanczos method. For $\tilde{A}$, a refined projection method seeks for each θ_i, $i = 1, 2, \ldots, k$, a unit length vector $\tilde{\psi}_i \in E$ satisfying the optimality property

$$||\tilde{A}\tilde{\psi}_i - \theta_i\tilde{\psi}_i|| = \min_{\psi\in E, ||\psi||=1} ||\tilde{A}\psi - \theta_i\psi|| \tag{3.1}$$

and uses them as new approximations to the desired eigenvectors $\frac{1}{\sqrt{2}}(u_i^{\mathrm{T}}, v_i^{\mathrm{T}})^{\mathrm{T}}$, $i = 1, 2, \ldots, k$. We call $\tilde{\psi}_i$ a refined eigenvector approximation or simply a refined Ritz vector of $\tilde{A}$ with respect to θ_i and the spectral norm. Partition

$$\tilde{\psi}_i = (\tilde{\psi}_{i1}^{\mathrm{T}}, \tilde{\psi}_{i2}^{\mathrm{T}})^{\mathrm{T}}, \tag{3.2}$$

with $\tilde{\psi}_{i1}$ and $\tilde{\psi}_{i2}$ being $m+1$- and m-dimensional, respectively, and take

$$\hat{u}_i = \frac{\tilde{\psi}_{i1}}{\|\tilde{\psi}_{i1}\|}, \qquad \hat{v}_i = \frac{\tilde{\psi}_{i2}}{\|\tilde{\psi}_{i2}\|}. \tag{3.3}$$

Then accordingly, we call the triplet $(\theta, \hat{u}_i, \hat{v}_i)$ a refined Ritz triplet for approximating the singular triplet (σ_i, u_i, v_i) of A.

Based on Theorem 3.2 of Jia [12], we have the following result.

THEOREM 3.1. *Let $z_i = (x_i^{\mathrm{T}}, y_i^{\mathrm{T}})^{\mathrm{T}}$ be the right singular vector of the matrix*

$$\begin{pmatrix} 0 & B_m \\ B_m^{\mathrm{T}} & 0 \\ \alpha_{m+1}e_m^{\mathrm{T}} & 0 \end{pmatrix} - \theta_i \begin{pmatrix} I & 0 \\ 0 & I \\ 0 & 0 \end{pmatrix}$$

associated with its smallest singular value $\sigma_{\min}$, where x_i and y_i are $m+1$- and m-dimensional, respectively. Then

$$\tilde{\psi}_i = \begin{pmatrix} P_{m+1} & 0 \\ 0 & Q_m \end{pmatrix} z_i, \tag{3.4}$$

$$\hat{u}_i = \frac{P_{m+1}x_i}{\|x_i\|}, \qquad \hat{v}_i = \frac{Q_m y_i}{\|y_i\|}, \tag{3.5}$$

$$||\tilde{A}\tilde{\psi}_i - \theta_i\tilde{\psi}_i|| = \sigma_{\min}. \tag{3.6}$$

The computational cost of each z_i is $O(m^3)$ flops. So if k is small, the extra cost of the refined bidiagonalization Lanczos method is very low, compared with the bidiagonalization Lanczos method. So, we can compute the refined approximate singular triplets efficiently and accurately.

Write

$$\hat{x}_i = \frac{x_i}{\|x_i\|}, \qquad \hat{y}_i = \frac{y_i}{\|y_i\|}.$$

Then it follows from (2.1) and (2.2) that

$$\begin{aligned} \|A\hat{v}_i - \theta_i\hat{u}_i\| &= \|AQ_m\hat{y}_i - \theta_i P_{m+1}\hat{x}_i\| \\ &= \|P_{m+1}B_m\hat{y}_i - \theta_i P_{m+1}\hat{x}_i\| \\ &= \|B_m\hat{y}_i - \theta_i\hat{x}_i\| \end{aligned} \tag{3.7}$$

and

$$\|A^{\mathrm{T}}\hat{u}_i - \theta_i\hat{v}_i\| = \sqrt{\|B_m^{\mathrm{T}}\hat{x}_i - \theta_i\hat{y}_i\|^2 + \alpha_{m+1}^2 \mid e_{m+1}^{\mathrm{T}}\hat{x}_i \mid^2}. \tag{3.8}$$

Therefore, we can claim a refined Ritz triplet $(\theta_i, \hat{u}_i, \hat{v}_i)$ to have converged if

$$\sqrt{\|B_m\hat{y}_i - \theta_i\hat{x}_i\|^2 + \|B_m^{\mathrm{T}}\hat{x}_i - \theta_i\hat{y}_i\|^2 + \alpha_{m+1}^2 \mid e_{m+1}^{\mathrm{T}}\hat{x}_i \mid^2} \le tol, \tag{3.9}$$

where *tol* is a user-prescribed tolerance. This important relation means that, similar to the bidiagonalization Lanczos method (cf. (2.7)), we do not need to form the refined Ritz approximations $\hat{u}_i$ and $\hat{v}_i$ explicitly before they converge.

Jia [20] proved that if $||\tilde{A}\tilde{\psi}_i - \theta_i\tilde{\psi}_i|| \neq 0$, i.e., the refined Ritz triplet $(\theta_i, \hat{u}_i, \hat{v}_i)$ is not an exact singular triplet of A, then $\tilde{\psi}_i \neq \tilde{\varphi}_i$, i.e., the refined approximations $\hat{u}_i$ and $\hat{v}_i$ are different from the Ritz approximations $\tilde{u}_i$ and $\tilde{v}_i$. Moreover, if $||\tilde{A}\tilde{\varphi}_i - \theta_i\tilde{\varphi}_i|| \neq 0$, then $||\tilde{A}\tilde{\psi}_i - \theta_i\tilde{\psi}_i|| < ||\tilde{A}\tilde{\varphi}_i - \theta_i\tilde{\varphi}_i||$. Furthermore, if θ_i is very close to one of the other distinct Ritz values θ_j, $j \neq i$, then it may happen that $||\tilde{A}\tilde{\psi}_i - \theta_i\tilde{\psi}_i|| \ll ||\tilde{A}\tilde{\varphi}_i - \theta_i\tilde{\varphi}_i||$. Therefore, $\hat{u}_i$ and $\hat{v}_i$ is more accurate and may be much more accurate than $\tilde{u}_i$ and $\tilde{v}_i$.

Jia and Stewart [18] derived a priori error bounds on the refined Ritz vector. The following result is a direct corollary of Theorem 4.1 of [18].

THEOREM 3.2. *Let (σ, u, v) be a singular triplet of A, and let $(u, U_\perp)$ and $(v, V_\perp)$ be orthogonal matrices such that*

$$\begin{pmatrix} u^{\mathrm{T}} \\ U_\perp^{\mathrm{T}} \end{pmatrix} A(v, V_\perp) = \begin{pmatrix} \sigma & 0 \\ 0 & L \end{pmatrix}, \tag{3.10}$$

where $L = U_\perp^T A V_\perp$. Define $\tilde{L} = \left(\begin{smallmatrix} 0 & L \\ L^{\mathrm{T}} & 0 \end{smallmatrix}\right)$. Assume that $(\theta, \tilde{\psi})$ is the refined Ritz pair approximating $(\sigma, \frac{1}{\sqrt{2}}(u^{\mathrm{T}}, v^{\mathrm{T}})^{\mathrm{T}})$. Then if

$$\mathrm{sep}(\theta, \tilde{L}) \ge \mathrm{sep}(\sigma, \tilde{L}) - |\theta - \sigma| > 0, \tag{3.11}$$

then

$$\sin\angle\left(\tilde{\psi}, \begin{pmatrix} u \\ v \end{pmatrix}\right) \le \frac{||\tilde{A} - \theta I||\varepsilon + |\theta - \sigma|}{\sqrt{1 - \varepsilon^2}(\mathrm{sep}(\sigma, \tilde{L}) - |\theta - \sigma|)}. \tag{3.12}$$

Recall that Theorem 2.1 shows $\theta \to \sigma$ as $\varepsilon \to 0$. Note that $\mathrm{sep}(\sigma, \tilde{L})$ is a positive constant independent of ε, assuming that A has only simple singular values. Therefore, Theorem 3.2 indicates that the refined Ritz approximations $\hat{u}$ and $\hat{v}$ converge to the left and right singular vectors u and v, respectively, as $\varepsilon \to 0$. Generally, they can be expected to be more accurate than the corresponding Ritz approximations $\tilde{u}$ and $\tilde{v}$. Hence the refined bidiagonalization Lanczos method corrects the possible nonconvergence of the standard bidiagonalization Lanczos method.

4. Implicit restart. In practice, due to the limitation of memory and computational complexity, m should not be large. However, for a small m, it is often the case that ε is not small enough, so that it cannot guarantee the convergence of the bidiagonalization Lanczos method and its refined counterpart. Therefore, we usually have to restart the methods in order to compute the desired singular triplets with prescribed accuracy. Over the past decade, the implicit restarting technique due to Sorensen [27] has proven to be a very successful and powerful restarting scheme and has been used either trivially or nontrivially in various contexts. In what follows, we review the technique and show how it is applied to the bidiagonalization Lanczos method and its refined counterpart.

For a general matrix C whose eigenpairs are (λ_i, φ_i), the m-step Arnoldi process [27] is

$$CV_m = V_m H_m + r_m e_m^{\mathrm{T}}. \tag{4.1}$$

Assume that the eigenpairs $(\lambda_i, \varphi_i), i = 1, 2, \ldots, k$, are desired. Given $m-k$ shifts μ_j, $j = 1, 2, \ldots, m-k$, for the $m \times m$ upper Hessenberg matrix H_m, we successively apply QR iterations to the shifted $H_m - \mu_j I$, deriving

$$(H_m - \mu_1 I)(H_m - \mu_2 I) \cdots (H_m - \mu_{m-k}) = QR, \tag{4.2}$$

where Q is orthogonal (unitary) and R is upper triangular. Define $H_m^+ = Q^* H_m Q$, $V_m^+ = V_m Q$, and H_k^+ to be the $k \times k$ leading principal matrix of H_m^+ and V_k^+ the first k columns of V_m^+. Then it holds by the k-step Arnoldi process that

$$CV_k^+ = V_k^+ H_k^+ + r_k^+ e_k^{\mathrm{T}}. \tag{4.3}$$

It has been shown [27] that the new initial vector

$$v_1^+ = p(C) v_1 \tag{4.4}$$

with $p(\lambda) = \alpha \prod_{j=1}^{m-k} (\lambda - \mu_j)$ and α a normalizing factor. Furthermore, it is shown [27] that

$$r_k^+ = 0 \quad \text{if and only if} \quad v_1^+ \in span\{\varphi_1, \varphi_2, \ldots, \varphi_k\}. \tag{4.5}$$

In this case the Arnoldi process breaks down at step k, V_k^+ spans an invariant subspace of C associated with $\lambda_1, \lambda_2, \ldots, \lambda_k$, and the eigenvalues of H_k^+ are just $\lambda_1, \lambda_2, \ldots, \lambda_k$. If r_k^+ is approximately zero, V_k^+ spans an approximate invariant subspace of C, and the eigenvalues of H_k^+ are accepted to have converged to $\lambda_1, \ldots, \lambda_k$.

The implicit restarting technique can be adapted to the bidiagonalization Lanczos process, as was done in [3, 22, 30]. They work in the following way: given the $m-k$ shifts $\mu_1, \ldots, \mu_{m-k}$, the implicit restarting technique leads to

$$\begin{cases} (B_m B_m^{\mathrm{T}} - \mu_1^2 I) \cdots (B_m B_m^{\mathrm{T}} - \mu_{m-k}^2 I) = \tilde{Q} R, \\ \tilde{P}^T B_m \tilde{Q} \quad \text{still (lower) bidiagonal}, \end{cases} \tag{4.6}$$

where $\tilde{P}$ and $\tilde{Q}$ are the accumulation matrices of Givens rotations applied to B_m from the left and right, respectively. Define $P_{m+1}^+ = P_{m+1}\tilde{P}$, $Q_m^+ = Q_m \tilde{Q}$, and $B_m^+ = \tilde{P}^T B_m \tilde{Q}$. The process is achieved implicitly from $B_m B_m^{\mathrm{T}}$ to $B_m^+ (B_m^+)^{\mathrm{T}}$ by working directly on B_m. This is a typical step of the Golub–Kahan SVD algorithm [7]

for a lower bidiagonal B_m. Then by exploiting the special structure of $\tilde{P}$ we obtain by manipulation

$$\tilde{A}\begin{pmatrix} P_{k+1}^+ & 0 \\ 0 & Q_k^+ \end{pmatrix} = \begin{pmatrix} P_{k+1}^+ & 0 \\ 0 & Q_k^+ \end{pmatrix}\begin{pmatrix} 0 & B_k^+ \\ B_k^{+\mathrm{T}} & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ (\alpha_{k+1}\tilde{p}_{m+1,k+1}q_{m+1} + \alpha_{k+1}^+ q_{k+1}^+)e_{k+1}^{\mathrm{T}} & 0 \end{pmatrix}, \tag{4.7}$$

with $\tilde{p}_{m+1,k+1}$ being the $(m+1, k+1)$ entry of $\tilde{P}$. Since $\alpha_{k+1}\tilde{p}_{m+1,k+1}q_{m+1}+\alpha_{k+1}^+ q_{k+1}^+$ is orthogonal to Q_k^+, we get a k-step bidiagonalization Lanczos process (Algorithm 1). It is then extended to an m-step bidiagonalization Lanczos process in a standard way. So we avoid restarting Algorithm 1 from scratch and doing it from step $k+1$ upwards. This way saves computational cost of the first k steps of Algorithm 1 by performing a sequence of implicit shift SVD steps on the small B_m at low cost. As a result, we have formally sketched an implicitly restarted bidiagonalization Lanczos algorithm (IRBL) and an implicitly restarted refined bidiagonalization Lanczos algorithm (IRRBL) for computing a partial SVD of a large matrix, which will be detailed later.

5. Selection of shifts. As was seen previously, we can run IRBL and IRRBL once the shifts μ_j, $j = 1, 2, \ldots, m-k$, are given. However, in order to make them work as efficiently as possible, we must select the best possible shifts available for each of them. For an implicitly restarted Krylov subspace algorithm for the eigenproblem, it has been shown [14, 17] that if the shifts are more accurate approximations to some of the unwanted eigenvalues of the original matrix, then the resulting new Krylov subspace will contain more accurate eigenvectors to the desired eigenvectors, so that the algorithm may converge faster. For IRBL and IRRBL, the same conclusion still holds. In an ideal case, similar to Theorem 3 of [17], we can prove the following result.

THEOREM 5.1. *Assume that the sets $\{\sigma_1, \ldots, \sigma_k\}$ and $\{\sigma_{k+1}, \ldots, \sigma_N\}$ are disjoint and A has only simple singular values. Then if $m-k$ distinct ones among σ_j, $j = k+1, \ldots, N$, are selected as shifts at each restart, then IRBL and IRRBL converge after at most $\lceil \frac{N-k}{m-k} \rceil$ restarts.*

Note that p_1^+ can be expressed as

$$\gamma p_1^+ = \prod_{i=1}^{m-k} (AA^{\mathrm{T}} - \mu_i^2 I)p_1, \tag{5.1}$$

with γ being a normalizing factor. Then by a continuity argument of polynomials, it is seen from this theorem and the above relation that the better μ_j approximates an unwanted singular value σ_{j_i} with $j_i > k$, the smaller the component of p_1^+ is in the direction of u_{j_i}, so that $\mathcal{K}_m(A^TA, A^Tp_1)$ and $\mathcal{K}_{m+1}(AA^{\mathrm{T}}, p_1)$ contain more accurate approximations to $v_1, v_2, \ldots, v_k$ and $u_1, u_2, \ldots, u_k$. As a consequence, IRBL and IRRBL usually converges faster.

For the implicitly restarted Arnoldi algorithm (IRA), Sorensen [27] proposed to select those unwanted Ritz values as shifts, called exact shifts. In some sense, this selection scheme is best for the algorithm as the exact shifts are the best approximations available obtained by the algorithm to some unwanted eigenvalues. So, for IRBL, we still use the exact shifts θ_j, $j = k+1, \ldots, m$, as they are the best approximations to some unwanted singular values obtained by IRBL at the current cycle. However, these exact shifts are *not* best for IRRBL as we can find better possible shifts than them based on the refined approximations $\hat{u}_i, \hat{v}_i$, $i = 1, 2, \ldots, k$, as shown below.

Note that the refined Ritz approximations $\hat{u}_i, \hat{v}_i$ are more accurate than the corresponding Ritz approximations $\tilde{u}_i, \tilde{v}_i$, $i = 1, 2, \ldots, k$. This motivates us to seek better possible shifts than θ_j, $j = k+1, \ldots, m$ based on $\hat{u}_i, \hat{v}_i$, $i = 1, 2, \ldots, k$. Let us make the orthogonal direct sum decompositions

$$span\{P_{m+1}\} = span\{\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_k\} \oplus span\{\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_k\}^{\perp} \tag{5.2}$$

$$= span\{\tilde{u}_1, \tilde{u}_2, \ldots, \tilde{u}_k\} \oplus span\{\tilde{u}_1, \tilde{u}_2, \ldots, \tilde{u}_k\}^{\perp}, \tag{5.3}$$

$$span\{Q_m\} = span\{\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_k\} \oplus span\{\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_k\}^{\perp} \tag{5.4}$$

$$= span\{\tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_k\} \oplus span\{\tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_k\}^{\perp}, \tag{5.5}$$

where $\oplus$ denotes the direct sum. Then clearly

$$span\{\tilde{u}_1, \tilde{u}_2, \ldots, \tilde{u}_k\}^{\perp} = span\{\tilde{u}_{k+1}, \ldots, \tilde{u}_{m+1}\},$$
$$span\{\tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_k\}^{\perp} = span\{\tilde{v}_{k+1}, \ldots, \tilde{v}_m\}.$$

Define

$$\tilde{U}_k = (\tilde{u}_1, \tilde{u}_1, \ldots, \tilde{u}_k), \qquad \tilde{V}_k = (\tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_k)$$

and

$$\tilde{U}_{m-k} = (\tilde{u}_{k+1}, \ldots, \tilde{u}_{m+1}), \qquad \tilde{V}_{m-k} = (\tilde{v}_{k+1}, \ldots, \tilde{v}_m).$$

Then it is easily justified from the bidiagonalization Lanczos method that the *wanted* Ritz values $\theta_1, \theta_2, \ldots, \theta_k$ are the singular values of A with respect to the left and right subspaces $span\{\tilde{u}_1, \tilde{u}_2, \ldots, \tilde{u}_k\}$ and $span\{\tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_k\}$, that is, they are the singular values of the projection matrix

$$\tilde{U}_k^T A \tilde{V}_k,$$

while on the other hand the *unwanted* Ritz values $\theta_{k+1}, \ldots, \theta_m$ are the singular values of A with respect to the left and right subspaces $span\{\tilde{u}_1, \tilde{u}_2, \ldots, \tilde{u}_k\}^{\perp}$ and $span\{\tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_k\}^{\perp}$, that is, they are the singular values of the projection matrix

$$\tilde{U}_{m-k}^T A \tilde{V}_{m-k}.$$

Keep in mind that $\hat{u}_i, \hat{v}_i$ are generally more accurate than $\tilde{u}_i, \tilde{v}_i$, $i = 1, 2, \ldots, k$, respectively. Then it is clear that $span\{\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_k\}^{\perp}$ and $span\{\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_k\}^{\perp}$ contain more accurate approximations to the unwanted left and right singular vectors $u_{k+1}, \ldots, u_N$ and $v_{k+1}, \ldots, v_N$ than $span\{\tilde{u}_1, \tilde{u}_2, \ldots, \tilde{u}_k\}^{\perp}$ and $span\{\tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_k\}^{\perp}$, respectively. As a consequence, the Ritz values ξ_i, $i = 1, 2, \ldots, m-k$, of A with respect to the left and right subspaces $span\{\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_k\}^{\perp}$ and $span\{\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_k\}^{\perp}$ should be generally more accurate approximations to some $m-k$ unwanted singular values than the unwanted Ritz values $\theta_{k+1}, \ldots, \theta_m$ of A with respect to the left and right subspaces $span\{\tilde{u}_1, \tilde{u}_2, \ldots, \tilde{u}_k\}^{\perp}$ and $span\{\tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_k\}^{\perp}$. Therefore, this suggests that we take the ξ_i's as shifts for use within IRRBL. In terms of Jia's terminology [14, 17], they are called the refined shifts. Jia [14, 17] presented very efficient and reliable algorithms to compute the refined shifts for use within the implicitly restarted refined Arnoldi algorithm and the implicitly restarted refined harmonic Arnoldi algorithm, respectively. Adapted from Jia's trick [14], we can propose an efficient algorithm to compute the refined shifts ξ_i's for IRRBL as follows.

Note that $\hat{u}_i = P_{m+1}\hat{x}_i$, $\hat{v}_i = Q_m\hat{y}_i$, $i = 1, 2, \ldots, k$. Define

$$\hat{U}_k = (\hat{u}_1, \ldots, \hat{u}_k) = P_{m+1}(\hat{x}_1, \ldots, \hat{x}_k) = P_{m+1}\hat{X}_k$$

and

$$\hat{V}_k = (\hat{v}_1, \ldots, \hat{v}_k) = Q_m(\hat{y}_1, \ldots, \hat{y}_k) = Q_m\hat{Y}_k.$$

Then we compute the full QR decompositions

$$\hat{X}_k = WR_1, \qquad \hat{Y}_k = SR_2$$

and partition

$$W = (W_k, W_{m-k}), \qquad S = (S_k, S_{m-k}),$$

from which it can be proved, similarly to Jia [14], that

$$span\{\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_k\}^{\perp} = span\{P_{m+1}W_{m-k}\}, \tag{5.6}$$

$$span\{\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_k\}^{\perp} = span\{Q_mS_{m-k}\}. \tag{5.7}$$

Recall from (2.3) that $P_{m+1}^TAQ_m = B_m$. Then it is known that the projection matrix of A with respect to the left subspace $span\{\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_k\}^{\perp}$ and the right subspace $span\{\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_k\}^{\perp}$ is

$$G = (P_{m+1}W_{m-k})^TA(Q_mS_{m-k}) = W_{m-k}^{\mathrm{T}}(P_{m+1}^TAQ_m)S_{m-k} = W_{m-k}^TB_mS_{m-k},$$

which can be formed at cost of $(m-k)^2m$ flops. So we have exploited the relation $P_{m+1}^TAQ_m = B_m$ to form G, which avoids computing $G = (P_{m+1}W_{m-k})^TA(Q_mS_{m-k})$ directly and reduces the computational cost considerably.

Based on the above arguments and the algorithms for computing the refined shifts [14, 17], we are now able to present the following algorithm.

ALGORITHM 2. THE COMPUTATION OF REFINED SHIFTS ξ_i'S.

1. Form the projection matrix

$$G = W_{m-k}^TB_mS_{m-k}.$$

2. Compute the $m-k$ singular values ξ_j, $j = 1, 2, \ldots, m-k$, of G.
3. Take the ξ_j's as the refined shifts for use within IRRBL.

Thus, starting with the refined Ritz approximations $\hat{u}_i, \hat{v}_i$, $i = 1, 2, \ldots, k$, we can compute the refined shifts ξ_j's using $O(m^3)$ flops, which is negligible compared with one cycle of IRBL.

As Larsen [22] noted, when large close singular values are present, IRBL with exact shifts may have very poor performance and even stagnation. IRRBL inherits the same deficiency. This is explained as follows: By inspecting the relation (5.1), we see the component along the desired kth singular vector u_k is greatly damped if a shift μ_i is very close to σ_k, so that θ_k converges to σ_k very slowly. Since $\mu_1 = \theta_{k+1}$ in the exact shifts and it is an approximation to σ_{k+1}, it is a bad shift when σ_k and σ_{k+1} are close and θ_{k+1} is approximating σ_{k+1}. For this case, the refined shifts have the same deficiency as there is a refined shift that is approaching σ_{k+1}.

To correct this problem, Larsen [22], for IRBL with the exact shifts $\mu_i = \theta_{k+i}$, $i = 1, 2, \ldots, m-k$, proposed the adaptive shifting strategy that required that the relative gaps

$$\mathrm{relgap}_{ki} = \frac{(\theta_k - \epsilon_k) - \mu_i}{\theta_k} \tag{5.8}$$

between the smallest Ritz value θ_k (i.e., the desired kth largest singular value) and all the shifts μ_i, $i = 1, 2, \ldots, m-k$, be larger than some prescribed tolerance, where ϵ_i is the residual norm (2.7). Since $\theta_k - \epsilon_k$ is an approximation to σ_k, relgap_{ki} can be considered to be an approximation of the relative gap of σ_k and the shift μ_i.

However, there is an oversight in (5.8), as relgap_{ki} is only guaranteed to be positive when θ_k is approaching σ_k, i.e., the IRBL is starting to converge. Clearly, if θ_k is still not converging, then ϵ_k is not small. In this case, relgap_{ki} can be negative, so that the strategy cannot work. A simple correction we propose is to replace relgap_{ki} by its absolute value. As in Larsen [22], if

$$| \text{ relgap}_{ki} | \leq 10^{-3},$$

we claim μ_i to be a bad shift and set it to zero. Zero shifts will amplify the component along u_k in p_1^+ and thus overcome the drawback of the exact shifts.

So the combination of the exact shifts and zero shifts will amplify the components along u_i, $i = 1, 2, \ldots, k$, in p_1^+ and at the same time dampen those along the unwanted u_i, $i = k+1, \ldots, N$. It holds to the refined shifts. So we combine the refined shifts with zero shifts for use within IRRBL when computing the largest singular triplets. However, we must point out that the above adaptive strategy works only for computing the largest singular values σ_i, $i = 1, 2, \ldots, k$. It *cannot* be adapted to compute the smallest close singular values of A.

To see why, suppose that we are required to compute the k smallest singular values $\sigma_1 < \sigma_2 < \cdots < \sigma_k$, and we use the k smallest Ritz values θ_i, $i = 1, 2, \ldots, k$, to approximate them. Now the exact shifts are the remaining $m-k$ unwanted large Ritz values $\mu_i = \theta_{k+i}$, $i = 1, 2, \ldots, m-k$, as shifts. Expand q_1 in the left singular basis vectors $\{u_j\}_{j=1}^M$ as

$$p_1 = \sum_{j=1}^{N} \alpha_j u_j + \sum_{j=N+1}^{M} \alpha_j u_j.$$

Then

$$\gamma p_1^+ = \sum_{j=1}^{k} \alpha_j \prod_{i=k+1}^{m} (\sigma_j^2 - \theta_i^2) u_j + \sum_{j=k+1}^{N} \alpha_j \prod_{i=k+1}^{m} (\sigma_j^2 - \theta_i^2) u_j + \sum_{j=N+1}^{M} \alpha_j \prod_{i=k+1}^{m} (-\theta_i^2) u_j.$$

It is clear that if θ_{k+1} is close to σ_k, then the component of p_1^+ in u_k is very small relative to the others. A good cure for this is to replace such a θ_i by the largest Ritz value θ_{m-k}. This way will amplify the components of p_1^+ along u_i, $i = 1, 2, \ldots, k$, and meanwhile possibly dampen those along u_i, $i = k+1, \ldots, N$.

Obviously, the above adaptive shifting strategy can be combined with the refined shifts. The differences are now that ϵ_k is the residual norm (3.9) and bad shifts are replaced by the largest refined shift. Having done the above, we now come to the following practical algorithm.

ALGORITHM 3. IRRBL WITH THE REFINED SHIFTS.

1. Assume a unit length vector p_1 of dimension M and the steps m, the number k of the desired largest or smallest singular triplets (σ_i, u_i, v_i), $i = 1, 2, \ldots, k$, and a user-prescribed tolerance *tol*.
2. Run Algorithm 1 to construct B_m, P_{m+1}, and Q_{m+1}.
3. Compute the singular values θ_i, $j = 1, 2, \ldots, m$, and take the first k ones as approximations to the desired σ_i, $i = 1, \ldots, k$. For each θ_i, $i = 1, 2, \ldots, k$, compute z_i satisfying (3.4).

4. Check if (3.9) for $i = 1, 2, \ldots, k$ is below *tol*. If yes, stop and explicitly compute the refined Ritz approximations $\hat{u}_i = P_{m+1}x_i/\|x_i\|$ and $\hat{v}_i = Q_m y_i/\|y_i\|$, where x_i and y_i are the vectors consisting of the first $m+1$ components and the last m components of z_i, respectively (see (3.5)); otherwise, continue.
5. Use Algorithm 2 to compute the refined shifts ξ_i, $i = 1, 2, \ldots, m-k$.
6. Go to step 2 and implicitly restart combined with the adaptive shifting strategy.

We see it is not necessary to explicitly form the refined Ritz approximations $\hat{u}_i, \hat{v}_i$, $i = 1, 2, \ldots, k$, before the algorithm converges. This way saves some computational work.

6. Numerical experiments. We have tested IRBL, IRRBl, PROPACK, LANSO, and ARPACK, whose MATLAB counterparts are lansvd.m, laneig.m (downloaded from [22]), and eigs.m, respectively. We ran experiments on an Intel Celeron 1700 MHz with main memory 256MB using MATLAB 5.3 with machine precision $\mathbf{u} = 2.22 \times 10^{-16}$. Recall (2.7) and (3.9). The stopping criterion for IRBL and IRRBL is

$$stopcrit = \max_{1\le i\le k} \sqrt{\|A\hat{v}_i - \theta_i\hat{u}_i\|^2 + \|A^{\mathrm{T}}\hat{u}_i - \theta_i\hat{v}_i\|^2}.$$

If

$$stopcrit \le tol \times \max\{\|B_m\|, 1\},$$

then

$$stopcrit \Leftarrow \max_{1\le i\le k} \frac{stopcrit}{||A||_1}.$$

If $stopcrit < tol$, stop.

By taking $m = 2k$ we intend to make all the restarted algorithms as black-box solvers for computing the largest singular values. To make a fair comparison, we used the same starting vector generated randomly in a uniform distribution whenever possible for all the restarted algorithms. In experiments, we took $tol = 10^{-6}$. In all the tables, "iter" denotes the number of restarts, "CPU" the CPU timings in second, and $m > 1000$ denotes no convergence of LANSO or PROPACK when the steps m (i.e., the subspace dimension) exceeded 1000. We terminated LANSO and PROPACK and counted CPU timings when $m > 1000$.

Example 1. We took some test matrices from [1, 6] for our purpose. Keep $\tilde{A} = [0, A; A', 0]$ in mind. IRBL and IRRBL used the same initial vector p_1, eigs($\tilde{A}$) used $(p_1^{\mathrm{T}}, 0)^{\mathrm{T}}$, and eigs($A^TA$) used A^Tp_1 as initial vectors.

From Tables 6.1–6.3, we see that IRRBL works at least as efficiently as IRBL in terms of restarts. For $k = 50$ and well1850, illc1850, and tols4000, it consumed significantly more CPU time than IRBL for some of the test matrices. This is because we had to compute k small SVDs to obtain refined Ritz vectors. In all the other cases, IRRBL was as good as IRBL and could be significantly better than IRBL both in terms of restarts and CPU timings. In particular, for can1054, saylr4, and add32, IRRBL was much faster than IRBL. Both algorithms were significantly better than ARPACK applied to $\tilde{A}$. ARPACK applied to $A^{\mathrm{T}}A$ was faster than IRRBL for five of the eight test matrices but was considerably slower than IRRBL for af23560, saylr4, and add32. However, ARPACK applied to $A^{\mathrm{T}}A$ is not able to compute the left

TABLE 6.1
Computing 10 *largest singular triplets.*

Matrix	well1850		illc1850		tols4000		af23560	
Program	steps	time	steps	time	steps	time	steps	time
lansvd	70	0.47	70	0.53	21	0.16	47	9.00
laneig(A^TA)	75	0.27	75	0.27	155	2.80	51	7.30
laneig($\tilde{A}$)	139	1.45	115	1.14	225	11.3	155	67.7
	iter	time	iter	time	iter	time	iter	time
eigs(A^TA)	18	1.02	11	0.77	22	11.4	8	50.2
eigs($\tilde{A}$)	55	10.13	25	5.84	36	31.5	22	112.1
IRBL	7	2.16	7	2.23	19	16.5	5	23.5
IRRBL	7	2.41	7	2.59	17	15.5	5	23.6
Matrix	can1054		dwt1242		saylr4		add32	
Program	steps	time	steps	time	steps	time	steps	time
lansvd	45	0.27	70	0.41	369	14.1	349	28.4
laneig(A^TA)	67	0.30	115	0.66	401	18.8	371	29.8
laneig($\tilde{A}$)	129	2.23	183	4.84	807	349	531	208
	iter	time	iter	time	iter	time	iter	time
eigs(A^TA)	6	1.41	7	2.30	42	43.4	72	152
eigs($\tilde{A}$)	14	6.34	16	9.67	107	207.3	81	417
IRBL	43	11.3	27	7.98	n.c.	-	79	56.8
IRRBL	8	2.50	15	5.33	48	41.9	44	39.0

TABLE 6.2
Computing 20 *largest singular triplets.*

Matrix	well1850		illc1850		tols4000		af23560	
Program	steps	time	steps	time	steps	time	steps	time
lansvd	150	2.38	141	3.28	41	0.33	83	17.0
laneig(A^TA)	143	0.84	141	1.05	167	3.55	85	14.7
laneig($\tilde{A}$)	139	8.20	279	7.91	303	22.8	173	102.7
	iter	time	iter	time	iter	time	iter	time
eigs(A^TA)	14	2.89	20	4.05	9	21	6	82.9
eigs($\tilde{A}$)	32	27.2	53	38.7	15	59.8	17	299
IRBL	7	7.41	11	13.0	9	27.8	4	64.2
IRRBL	7	10.9	8	13.3	8	28.6	4	66.0
Matrix	can1054		dwt1242		saylr4		add32	
Program	steps	time	steps	time	steps	time	steps	time
lansvd	74	0.66	122	1.19	445	21.5	467	49
laneig(A^TA)	83	0.42	145	1.09	575	43.2	505	63
laneig($\tilde{A}$)	167	4.22	259	12.4	>1000	862	>1000	1844
	iter	time	iter	time	iter	time	iter	time
eigs(A^TA)	4	2.70	8	5.83	29	82.7	72	152
eigs($\tilde{A}$)	10	14.2	18	26.7	77	352	81	417
IRBL	4	3.23	8	8.17	33	93.5	38	116
IRRBL	4	5.48	8	12.5	31	103	12	43

singular vectors simultaneously and is less preferable, as it can lead to severe loss of accuracy of small singular values. LANSO failed in some cases when m exceeded 1000. It could be faster than IRBL and IRRBL in some cases but required (much) more memory to save Lanczos basis vectors for computing singular vectors. LANSO applied to $\tilde{A}$ could be much slower than IRRBL and meanwhile used much more memory. PROPACK was faster than IRRBL in most cases but used much more memory.

TABLE 6.3
Computing 50 *largest singular triplets.*

Matrix	well1850		illc1850		tols4000		af23560	
Program	steps	time	steps	time	steps	time	steps	time
lansvd	422	38.5	315	12.4	101	1.47	154	75.8
laneig(A^TA)	423	13.0	319	4.52	215	6.16	167	52.1
laneig($\tilde{A}$)	847	129	635	47.1	473	59.1	333	14340
	iter	time	iter	time	iter	time	iter	time
eigs(A^TA)	21	29	13	18	2	40	5	302
eigs($\tilde{A}$)	48	236	31	173	9	206	15	1560
IRBL	9	64	6	40	4	67	3	287
IRRBL	9	219	6	142	4	135	3	319
Matrix	can1054		dwt1242		saylr4		add32	
Program	steps	time	steps	time	steps	time	steps	time
lansvd	135	1.69	223	4.25	808	123	505	64.5
laneig(A^TA)	139	1.08	213	2.59	>1000	277	469	51.6
laneig($\tilde{A}$)	281	12.8	423	30.9	>1000	641	>1000	2459
	iter	time	iter	time	iter	time	iter	time
eigs(A^TA)	3	8.69	5	20.8	37	489	19	288
eigs($\tilde{A}$)	8	52.3	12	97.8	81	1680	27	1042
IRBL	2	7.83	4	22.8	1019	38774	13	339
IRRBL	2	43.6	4	92.9	139	6126	7	291

Example 2. We now report some test results for computing a few of the smallest singular triplets by IRBL and IRRBL. In contrast to Example 1, it appears that the computation of smallest singular triplets is much more difficult. It turns out that it is hard to use them as black-box solvers. So we test each case for several m. Since LANSO, PROPACK, and ARPACK exploit shift-and-invert to compute smallest singular triplets, we are not able to compare IRBL and IRRBL with them now and can only give a comparison between IRRBL and IRBL. The test matrices are from [1, 6]. In the tables, "n.c." denotes no convergence after 2000 restarts are used. Tables 6.4–6.13 list the results obtained.

We see that in contrast to Tables 6.1–6.3 it was much more difficult to compute the smallest singular triplets. We could use neither IRBL nor IRRBL as a black-box solver. Performance of IRBL and IRRBL depended heavily on m. However, it is clearly seen from Tables 6.4–6.13 that IRRBL was much more efficient than IRBL, and the latter often failed but the former solved a problem quite successfully.

TABLE 6.4
*well*1850*, computing the* k *smallest singular triplets.*

	$k=3$				$k=5$			
	IRBL		IRRBL		IRBL		IRRBL	
m	iter	time	iter	time	iter	time	iter	time
10	1077	204	697	138	1351	209	933	153
15	372	152	294	125	347	128	190	76
20	193	138	132	98	161	107	71	49
25	116	129	74	84	91	96	60	66

TABLE 6.5
*dw*2048, *computing the k smallest singular triplets.*

	$k=3$				$k=5$			
	IRBL		IRRBL		IRBL		IRRBL	
m	iter	time	iter	time	iter	time	iter	time
20	n.c.	-	n.c.	-	n.c.	-	1393	1067
30	n.c.	-	1716	3835	955	1566	667	1140
40	1516	6121	806	3350	493	1449	285	882
50	929	4470	481	2394	301	1433	209	1042

TABLE 6.6
*lshp*2233, *computing the k smallest singular triplets.*

	$k=3$				$k=5$			
	IRBL		IRRBL		IRBL		IRRBL	
m	iter	time	iter	time	iter	time	iter	time
20	1475	1574	734	808	n.c.	-	1587	1638
30	602	1440	311	756	949	2130	581	1348
40	328	1402	214	933	499	2084	284	1237
50	207	1380	165	1163	309	2197	207	1491

TABLE 6.7
*bcspwr*06, *computing the k smallest singular triplets.*

	$k=3$				$k=5$			
	IRBL		IRRBL		IRBL		IRRBL	
m	iter	time	iter	time	iter	time	iter	time
15	271	86.3	158	51.1	1179	338	829	250
20	137	72.3	68	36.7	521	251	419	221
25	85	69.8	48	40.7	293	224	192	156

TABLE 6.8
*bcspwr*07, *computing the k smallest singular triplets.*

	$k=3$				$k=5$			
	IRBL		IRRBL		IRBL		IRRBL	
m	iter	time	iter	time	iter	time	iter	time
10	n.c.	-	1231	199	n.c.	-	n.c.	-
15	709	246	417	149	n.c.	-	1685	548
20	361	211	265	160	1069	557	615	346
25	215	196	124	115	595	538	394	350

TABLE 6.9
*bcspwr*08, *computing the k smallest singular triplets.*

	$k=3$				$k=5$			
	IRBL		IRRBL		IRBL		IRRBL	
m	iter	time	iter	time	iter	time	iter	time
10	1385	237	691	113	n.c.	-	n.c.	-
15	485	182	287	102	n.c.	-	1691	561
20	245	144	153	93.3	1563	843	1067	619
25	149	137	116	110	885	786	582	547

TABLE 6.10
*bcspwr*09, *computing the k smallest singular triplets.*

	$k=3$				$k=5$			
	IRBL		IRRBL		IRBL		IRRBL	
m	iter	time	iter	time	iter	time	iter	time
10	1063	183	693	121	n.c.	-	n.c.	-
15	371	139	319	121	1041	347	497	169
20	189	117	165	102	463	262	274	159
25	113	110	83	82	263	236	186	171

TABLE 6.11
*pde*900, *computing the k smallest singular triplets.*

	$k=3$				$k=5$			
	IRBL		IRRBL		IRBL		IRRBL	
m	iter	time	iter	time	iter	time	iter	time
10	n.c.	-	1431	146	n.c.	-	1827	155
15	913	201	649	143	797	151	398	79.1
20	458	177	311	121	355	138	285	118
25	164	204	199	122	204	124	108	64.3

TABLE 6.12
*jpwh*991, *computing the k smallest singular triplets.*

	$k=3$				$k=5$			
	IRBL		IRRBL		IRBL		IRRBL	
m	iter	time	iter	time	iter	time	iter	time
15	1371	343	1039	255	n.c.	-	1627	380
20	665	279	421	176	968	397	746	317
25	381	252	284	194	527	322	349	221
30	265	246	193	181	325	280	234	214

TABLE 6.13
*plat*1919, *computing the k smallest singular triplets.*

	$k=3$				$k=5$			
	IRBL		IRRBL		IRBL		IRRBL	
m	iter	time	iter	time	iter	time	iter	time
15	1569	671	785	335	n.c.	-	n.c.	-
20	763	560	307	223	n.c.	-	n.c.	-
25	451	489	244	267	n.c.	-	1526	1642
30	145	179	117	183	n.c.	-	947	1450

7. Conclusion. Both IRRBL and IRBL can be used to compute a partial SVD of a large matrix. But IRRBL is much more efficient than IRBL for computing the smallest singular triplets; in some cases, it can be significantly better than IRBL for computing the largest singular triplets. In comparison with IRBL, it is safer to use IRRBL as a black-box solver for computing the largest singular triplets. For computing the smallest singular triplets, IRBL and IRRBL still cannot perform as black-box solvers, and their performance depends heavily on m. Numerical experiments have demonstrated that (1) the refined Ritz approximations can be much more accurate than the Ritz approximations and (2) the refined shifts can be much better than the exact shifts. For the effect of the refined approximations and the refined shifts on a refined restarted algorithm, see [14, 17] for more analysis.

Note the difficulty of computing the smallest singular triplets. It may be good

to combine IRBL and IRRBL with shift-and-invert. As is well known, however, each step may be very costly and even unacceptable since one has to solve a large linear system each step. Another possibly promising approach to settling the issue is to develop harmonic versions of IRBL and IRRBL, avoiding explicit shift-and-invert, as was done in [17, 24].

Acknowledgments. We thank two referees very much for their very helpful comments and suggestions, which enabled us to improve the presentation and numerical experiments of the paper considerably.

REFERENCES

[1] Z. Bai, R. Barret, D. Day, J. Demmel, and J. Dongarra, *Test Matrix Collection for Non-Hermitian Eigenvalue Problems*, Technical Report CS-97-355, University of Tennessee, Knoxville, 1997. LAPACK Note #123. Data available online from http://math.nist.gov/MarketMatrix.

[2] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. A. van der Vorst, *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, SIAM, Philadelphia, PA, 2000.

[3] A. Björck, E. Grimme, and P. van Dooren, *An implicit shift bidiagonalization algorithm for ill-posed systems*, BIT, 34 (1994), pp. 510–534.

[4] Å. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, PA, 1996.

[5] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, PA, 1997.

[6] I. S. Duff, R. G. Grimes, and J. G. Lewis, *Sparse matrix test problems*, ACM Trans. Math. Software, 15 (1989), pp. 1–14.

[7] G. H. Golub and W. Kahan, *Calculating the singular values and pseudo-inverse of a matrix*, J. Soc. Indust. Appl. Math. Ser. B Numer. Anal., 2 (1965), pp. 205–224.

[8] G. H. Golub, F. T. Luk, and M. L. Overton, *A block Lanczos method for computing the singular values and singular vectors of a matrix*, ACM Trans. Math. Software, 7 (1981), pp. 149–169.

[9] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.

[10] Z. Jia, *Some Numerical Methods for Large Unsymmetric Eigenproblems*, Ph.D. thesis, Department of Mathematics, Bielefeld University, Bielefeld, Germany, 1994.

[11] Z. Jia, *The convergence of generalized Lanczos methods for large unsymmetric eigenproblems*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 843–862.

[12] Z. Jia, *Refined iterative algorithms based on Arnoldi's process for large unsymmetric eigenproblems*, Linear Algebra Appl., 259 (1997), pp. 1–23.

[13] Z. Jia, *A refined iterative algorithm based on the block Arnoldi process for large unsymmetric eigenproblems*, Linear Algebra Appl., 270 (1998), pp. 170–189.

[14] Z. Jia, *Polynomial characterizations of the approximate eigenvectors by the refined Arnoldi method and an implicitly restarted refined Arnoldi algorithm*, Linear Algebra Appl., 287 (1999), pp. 191–214.

[15] Z. Jia, *Composite orthogonal projection methods for large matrix eigenproblems*, Sci. China Ser. A, 42 (1999), pp. 577–585.

[16] Z. Jia, *A refined subspace iteration algorithm for large sparse eigenproblems*, Appl. Numer. Math., 32 (2000), pp. 35–52.

[17] Z. Jia, *The refined harmonic Arnoldi method and an implicitly restarted refined algorithm for computing interior eigenpairs of large matrices*, Appl. Numer. Appl., 42 (2002), pp. 489–512.

[18] Z. Jia and G. W. Stewart, *An analysis of the Rayleigh-Ritz method for approximating eigenspaces*, Math. Comp., 70 (2001), pp. 637–647.

[19] Z. Jia and G. W. Stewart, *On the Convergence of Ritz Values, Ritz Vectors and Refined Ritz Vectors*, Technical Report TR-3986, Department of Computer Science, University of Maryland, College Park, MD, 1999. Also available online from http://www.cs.umd.edu/~stewart.

[20] Z. Jia, *A Theoretical Comparison of Two Classes of Projection Methods*, technical report, Department of Applied Mathematics, Dalian University of Technology, Dalian, People's Republic of China, 2001.

[21] R. M. Larsen, *Lanczos Bidiagonalization with Partial Reorthogonalization*, technical report, Department of Computer Science, University of Aarhus, Aarhus, Denmark, 1998. Also

available online from http://soi.stanford. edu/˜rmunk/PROPACK.
[22] R. M. Larsen, *Combining Implicit Restarts and Partial Reorthogonalization in Lanczos Bidiagonalization*, available online from http://soi.stanford.edu/˜rmunk/PROPACK.
[23] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, SIAM, Philadelphia, PA, 1998.
[24] R. B. Morgan, *Implicitly restarted GMRES and Arnoldi methods for nonsymmetric systems of equations*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1112–1135.
[25] C. C. Paige and M. A. Saunders, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Software, 8 (1982), pp. 43–71.
[26] B. N. Parlett, *The Symmetric Eigenvalue Problem*, SIAM, Philadelphia, PA, 1998.
[27] D. C. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357–385.
[28] G. W. Stewart, *Matrix Algorithms. Volume* II: *Eigensystems*, SIAM, Philadelphia, PA, 2001.
[29] H. A. van der Vorst, *Computational methods for large eigenvalue problems*, in Handbook of Numerical Analysis, Vol. III, P. G. Ciarlet and J. L. Lions, eds., Elsevier, North-Holland, Amsterdam, 2002, pp. 3–179.
[30] X. Wang and H. Zha, *An Implicitly Restarted Bidiagonal Lanczos Method for Large-Scale Singular Value Problems*, Technical Report 42472, Scientific Computing Division, Lawrence Berkeley National Laboratory, Berkeley, CA, 1998.