TAIWANESE JOURNAL OF MATHEMATICS
Vol. 20, No. 4, pp. 943–956, August 2016
DOI: 10.11650/tjm.20.2016.6194
This paper is available online at http://journal.tms.org.tw

Improving Approximate Singular Triplets in Lanczos Bidiagonalization Method

Datian Niu* and Jiana Meng

Abstract. Lanczos bidiagonalization method is the most popular method for computing some largest singular triplets of large matrices. In this method, $2m+1$ base vectors are generated from the m-step Lanczos bidiagonalization process, but only $2m$ of them are used to form the approximate singular vectors and one of them is not used. In this paper, we make two improvements on the classical Lanczos bidiagonalization method. Firstly, following Jia and Elsner's idea for eigenproblems [9], we form the new approximate singular vectors by minimizing the corresponding residual norms in subspaces generated by $2m+1$ base vectors to replace the old approximate singular vectors. Secondly, in the process of implicit restarting, we replace the classical exact shifts by new shifts based on the information of the new approximate singular vectors. The total extra cost of the new method can be neglected. Numerical experiments show that, after two improvements, the new method proposed in this paper performs much better than the classical Lanczos bidiagonalization method. It uses less restarts and CPU time to reach the desired convergence.

1. Introduction

The singular value decomposition (SVD) of a matrix $A \in \mathcal{R}^{M\times N}$, $M \geq N$ (Otherwise, we deal with A^{T}, the transpose of A) is

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

where $U = (u_1, u_2, \ldots, u_M)$ and $V = (v_1, v_2, \ldots, v_N)$ are orthogonal matrices of order M and N respectively, Σ is a diagonal matrix with nonnegative diagonal elements σ_i, $i = 1, 2, \ldots, N$. σ_i is called a singular value of A, u_i and v_i are the associate left and right singular vectors, and (σ_i, u_i, v_i) is called a singular triplet. For convenience, the singular values are labeled as $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_N$.

Received April 24, 2015; Accepted December 10, 2015.
Communicated by Raymond Chan.
2010 *Mathematics Subject Classification.* 65F15, 15A18.
Key words and phrases. Lanczos bidiagonalization method, Singular triplet, Minimizing residual norms, Implicit restart, Shift.

*Corresponding author.

Computing some largest or smallest singular triplets of large matrices arises in many scientific and engineering applications, such as least squares problems, total least squares problems, regression analysis, image processing, signal processing, information retrieval, and so on.

Since M and N are assumed to be large, we can resort only to the projection methods. The SVD of A is equivalent to the eigendecomposition of $A^{\mathrm{T}}A$, AA^{T} and

$$\widetilde{A} = \begin{pmatrix} 0 & A \\ A^{\mathrm{T}} & 0 \end{pmatrix}.$$

$A^{\mathrm{T}}A$, AA^{T} and $\widetilde{A}$ are symmetric matrices. The typical method for solving symmetric eigenproblems is the symmetric Lanczos method [16]. However, if we apply the symmetric Lanczos method directly to $A^{\mathrm{T}}A$, AA^{T} and $\widetilde{A}$, there are some drawbacks. Firstly, the condition numbers of $A^{\mathrm{T}}A$ and AA^{T} are squared of A, which may cause some numerical instability. Secondly, the eigenvalues of $A^{\mathrm{T}}A$ and AA^{T} are σ_i^2, $i = 1, 2, \ldots, N$, so it is generally difficult to compute the smallest singular triplets due to the clustering of them. Finally, the eigenvalues of $\widetilde{A}$ are $\pm\sigma_i$, and the associate eigenvectors are $(u_i^{\mathrm{T}}, v_i^{\mathrm{T}})^{\mathrm{T}}$ and $(u_i^{\mathrm{T}}, -v_i^{\mathrm{T}})^{\mathrm{T}}$, respectively. The symmetric Lanczos method tends to use twice cost to obtain the wanted singular triplets. Meanwhile, the symmetric Lanczos method can not preserve the special structure of $\widetilde{A}$.

Since the drawbacks mentioned before, the symmetric Lanczos method is impractical for computing singular triplets. Until now, the Lanczos bidiagonalization method and its variants are the popular used methods for computing the singular triplets of large matrices. Golub et al. [6] firstly proposed a block Lanczos bidiagonalization method to compute the largest singular triplets. Larsen [13], Simon and Zha [18] discussed the reorthogonalization of the Lanczos bidiagonalization process. Jia and Niu [10] proposed a refined Lanczos bidiagonalization method to compute the largest and smallest singular triplets. Kokiopoulou et al. [12] used the harmonic project technique in the Lanczos bidiagonalization method to compute the smallest singular triplets, Niu and Yuan [15] improved their method. Baglama and Reichel [1, 2] prosed an augmented Lanczos bidiagonalization method and its block version to compute the largest and smallest singular triplets. Hernandez et al. [8] provided a parallel implementation of the Lanczos bidiagonalization method. Stoll [20] applied Krylov-Schur decomposition into the Lanczos bidiagonalization method. Jia and Niu [11] proposed a refined harmonic Lanczos bidiagonalization method to compute some smallest singular triplets. The above methods form two Krylov subspaces by Lanczos bidiagonalization process and extract approximate singular triplets from them in different ways.

Due to the storage requirements and computational cost, the Lanczos bidiagonalization method must be restarted. The most commonly used restarting technique is the implicit restarting technique proposed by Sorensen [19]. The implicit restarting technique is originally designed for eigenproblems and has been applied to the Lanczos bidiagonalization method. It heavily depends on the selection of the shifts. There are some shift selection strategies for the Lanczos bidiagonalization method, such as exact shifts [14], harmonic shifts [12], refined shifts [10], refined harmonic shifts [11], and Leja shifts [3].

In this paper, we analyze the Lanczos bidiagonalization method and find that, in each restart, the Lanczos bidiagonalization method generates $2m+1$ base vectors but extracts approximate singular triplets from the subspace generated by $2m$ base vectors. So there is one vector unused in the Lanczos bidiagonalization method. Following Jia and Elsner's idea [9] for eigenproblems, we extract new approximate singular triplets, where they are the linear combination of the old approximate singular vectors and the unused base vector, and minimize their residual norms from the subspace generated by all of the $2m+1$ base vectors. Then, using the information of the new singular triplets, we design a new shift selection strategy. Numerical experiments show that the above two improvements greatly improve the performance of the method.

In this paper, denote by $\|\cdot\|$ the spectral norm of a matrix and the 2-norm of a vector, by e_m the m-th coordinate vector of dimension m, by I_K the K-dimensional identity matrix.

2. Review of Lanczos bidiagonalization method

For a given matrix A, an initial vector q_1 and a positive integer m ($\ll \min(M,N)$), the m-step Lanczos bidiagonalization process is given by the following matrix form in the absence of break-down:

$$AQ_m = P_m B_m, \tag{2.1}$$

$$A^{\mathrm{T}} P_m = Q_m B_m^{\mathrm{T}} + \beta_{m+1} q_{m+1} e_m^{\mathrm{T}}, \tag{2.2}$$

where $P_m = (p_1, p_2, \ldots, p_m)$, $Q_m = (q_1, q_2, \ldots, q_m)$, P_m and (Q_m, q_{m+1}) are column orthonormal matrices respectively, and

$$B_m = \begin{pmatrix} \alpha_1 & \beta_2 & & \\ & \alpha_2 & \ddots & \\ & & \ddots & \beta_m \\ & & & \alpha_m \end{pmatrix}$$

with the positive α_i and β_i. This process is just the truncated version of the Golub-Kahan standard SVD process [7]. It is also equivalent to the symmetric Lanczos process on $\widetilde{A}$ with the initial vector $(0^{\mathrm{T}}, q_1^{\mathrm{T}})^{\mathrm{T}}$ [16].

In finite precision arithmetic, P_m and Q_m may loss orthogonality rapidly and must be reorthogonalized. There are several reorthogonalization strategies, such as full reorthogonalization, partial reorthogonalization, and one-side reorthogonalization. The one-side reorthogonalization reorthogonalizes Q_m only when B_m is not very ill-conditioned, which can reduce the computational cost considerably when $M \ll N$. See [18] for details. Baglama and Reichel [1] provide a MATLAB code for Lanczos bidiagonalization process with one-side reorthogonalization. We adopt their code, which save our work on programming.

Let the singular triplets of B_m be $(\widetilde{\sigma}_i, \widetilde{x}_i, \widetilde{y}_i)$, $i = 1, 2, \ldots, m$, where $\widetilde{\sigma}_1 \geq \widetilde{\sigma}_2 \geq \cdots \geq \widetilde{\sigma}_m$. The Lanczos bidiagonalization method takes $(\widetilde{\sigma}_i, \widetilde{u}_i, \widetilde{v}_i)$, $i = 1, 2, \ldots, k$, to be the k wanted approximate singular triplets, where $\widetilde{u}_i = P_m\widetilde{x}_i$, $\widetilde{v}_i = Q_m\widetilde{y}_i$. We also have

$$A\widetilde{v}_i - \widetilde{\sigma}_i\widetilde{u}_i = AQ_m\widetilde{y}_i - \widetilde{\sigma}_iP_m\widetilde{x}_i = P_m(B_m\widetilde{y}_i - \widetilde{\sigma}_i\widetilde{x}_i) = 0, \tag{2.3}$$

$$\begin{aligned} A^{\mathrm{T}}\widetilde{u}_i - \widetilde{\sigma}_i\widetilde{v}_i &= A^{\mathrm{T}}P_m\widetilde{x}_i - \widetilde{\sigma}Q_m\widetilde{y}_i \\ &= Q_m(B_m^{\mathrm{T}}\widetilde{x}_i - \widetilde{\sigma}_i\widetilde{y}_i) + \beta_{m+1}e_m^{\mathrm{T}}\widetilde{x}_iq_{m+1} \\ &= \beta_{m+1}e_m^{\mathrm{T}}\widetilde{x}_iq_{m+1}. \end{aligned} \tag{2.4}$$

Therefore, the residual norm of $(\widetilde{\sigma}_i, \widetilde{u}_i, \widetilde{v}_i)$ is

$$\widetilde{r}_i = \sqrt{\|A\widetilde{v}_i - \widetilde{\sigma}_i\widetilde{u}_i\|^2 + \|A^{\mathrm{T}}\widetilde{u}_i - \widetilde{\sigma}_i\widetilde{v}_i\|^2} = \beta_{m+1}\left|e_m^{\mathrm{T}}\widetilde{x}_i\right|.$$

When $\widetilde{r}_i$ is less than tol, a prescribed tolerance, we stop the algorithm. This indicate that we need not form $\widetilde{u}_i$ and $\widetilde{v}_i$ before the algorithm converged, which can greatly decrease the computational cost.

We see that q_{m+1} is already in hand, but it is not used to form the approximate singular triplets. A natural question is "can we use q_{m+1} to improve the performance of the approximate singular triplets?"

3. Using q_{m+1} to improve approximate singular triplets

Jia and Elsner [9] proposed a modified Arnoldi method for computing some largest eigenpairs of large matrices. They used the $(m+1)$-th base vector generated by the m-step Arnoldi procedure to improve the Ritz vectors. In this section, we apply this idea to SVD problems.

According to the relation between the Lanzos bidiagonalization process of A and the symmetric Lanczos process of $\widetilde{A}$, we have

$$\widetilde{r}_i = \left\|\left[\begin{pmatrix} 0 & A \\ A^{\mathrm{T}} & 0 \end{pmatrix} - \widetilde{\sigma}_i\begin{pmatrix} I_M & 0 \\ 0 & I_N \end{pmatrix}\right]\begin{pmatrix} \widetilde{u}_i \\ \widetilde{v}_i \end{pmatrix}\right\|. \tag{3.1}$$

Consider

$$\left\| \left[\begin{pmatrix} 0 & A \\ A^{\mathrm{T}} & 0 \end{pmatrix} - \widetilde{\sigma}_i \begin{pmatrix} I_M & 0 \\ 0 & I_N \end{pmatrix} \right] \left[a \begin{pmatrix} \widetilde{u}_i \\ \widetilde{v}_i \end{pmatrix} + b \begin{pmatrix} 0 \\ q_{m+1} \end{pmatrix} \right] \right\|, \tag{3.2}$$

such that

$$a^2 + b^2 = 1. \tag{3.3}$$

Obviously, (3.1) is the special case of (3.2) when $a = 1$ and $b = 0$. If a_i and b_i minimize (3.2) and (3.3), and $\widehat{r}_i$ is the corresponding value of (3.2), it can be easily seen that $\widehat{r}_i$ is at least as small as $\widetilde{r}_i$ and may be much smaller than $\widetilde{r}_i$. Therefore, if we take $(\widetilde{\sigma}_i, \widehat{u}_i, \widehat{v}_i)$ be the approximate singular triplets, where $\widehat{u}_i = \widetilde{u}_i$, $\widehat{v}_i = \frac{a_i\widetilde{v}_i + b_i q_{m+1}}{\|a_i\widetilde{v}_i + b_i q_{m+1}\|} = a_i\widetilde{v}_i + b_i q_{m+1}$, it is better and may be much better than $(\widetilde{\sigma}_i, \widetilde{u}_i, \widetilde{v}_i)$.

It seems that minimizing (3.2) and (3.3) is impractical since A is large. Fortunately, according to (2.3) and (2.4), we have

$$\begin{aligned} &\left[\begin{pmatrix} 0 & A \\ A^{\mathrm{T}} & 0 \end{pmatrix} - \widetilde{\sigma}_i \begin{pmatrix} I_M & 0 \\ 0 & I_N \end{pmatrix} \right] \left[a \begin{pmatrix} \widetilde{u}_i \\ \widetilde{v}_i \end{pmatrix} + b \begin{pmatrix} 0 \\ q_{m+1} \end{pmatrix} \right] \\ &= a \begin{pmatrix} A\widetilde{v}_i - \widetilde{\sigma}_i\widetilde{u}_i \\ A^{\mathrm{T}}\widetilde{u}_i - \sigma_i\widetilde{v}_i \end{pmatrix} - b \begin{pmatrix} Aq_{m+1} \\ \widetilde{\sigma}_i q_{m+1} \end{pmatrix} \\ &= \begin{pmatrix} 0 & Aq_{m+1} \\ \beta_{m+1} e_m^{\mathrm{T}} \widetilde{x}_i q_{m+1} & -\widetilde{\sigma}_i q_{m+1} \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix}. \end{aligned}$$

If s_i is the smallest singular value of

$$\widetilde{B}_i = \begin{pmatrix} 0 & Aq_{m+1} \\ \beta_{m+1} e_m^{\mathrm{T}} \widetilde{x}_i q_{m+1} & -\widetilde{\sigma}_i q_{m+1} \end{pmatrix}$$

and $(a_i, b_i)^{\mathrm{T}}$ are the associate right singular vector, a_i and b_i minimize (3.2) and (3.3) and $\widehat{r}_i = s_i$. In fact, we only need to compute the SVD of a 2×2 matrix

$$\widehat{B}_i = \begin{pmatrix} 0 & \|Aq_{m+1}\| \\ \beta_{m+1} e_m^{\mathrm{T}} \widetilde{x}_i & -\widetilde{\sigma}_i \end{pmatrix}$$

since

$$\widetilde{B}_i = \begin{pmatrix} Aq_{m+1}/\|Aq_{m+1}\| & 0 \\ 0 & q_{m+1} \end{pmatrix} \begin{pmatrix} 0 & \|Aq_{m+1}\| \\ \beta_{m+1} e_m^{\mathrm{T}} \widetilde{x}_i & -\widetilde{\sigma}_i \end{pmatrix},$$

and the corresponding cost is negligible[1].

[1] We thank one reviewer for suggesting us to compute s_i, a_i, b_i from the SVD of $\widehat{B}_i$. In the first version, we took s_i and $(a_i^{\mathrm{T}}, b_i^{\mathrm{T}})^{\mathrm{T}}$ to be the smallest eigenpair of $\widetilde{B}_i^{\mathrm{T}}\widetilde{B}_i$. The reviewer's way is more accurate.

4. Implicit restarting and shift selection

Due to the storage requirements and the computational cost, m, the number of Lanczos bidiagonalization steps, can not be large and is limited to be relatively small. However, for small m, there may be not enough information of the wanted singular triplets. Practically, we must restart the algorithm and compute the wanted singular triplets iteratively. There are two restarting techniques: explicit restarting [17] and implicit restarting [19]. The implicit restarting technique was proposed by Sorensen for eigenproblems and has been applied to SVD problems. It can save the matrix-vector products, which are the main cost of the Lanczos bidiagonalization method for large matrices. So it becomes the commonly used restarting technique.

The implicit restarting technique for the Lanczos bidiagonalization process proceeds as follows: choose p ($p \leq m-k$) shifts $\mu_1, \mu_2, \ldots, \mu_p$, run the implicit QR iteration with the shifts, whose matrix form is

$$\begin{cases} (B_m^{\mathrm{T}} B_m - \mu_1^2 I)(B_m^{\mathrm{T}} B_m - \mu_2^2 I) \cdots (B_m^{\mathrm{T}} B_m - \mu_p^2 I) = \widetilde{P} R, \\ \widetilde{P}^{\mathrm{T}} B_m \widetilde{Q} \text{ is still bidiagonal,} \end{cases}$$

where $\widetilde{P}$ and $\widetilde{Q}$ are the accumulations of the left and right Givens rotation matrices applied on B_m. It is just the Golub-Kahan standard SVD process, see [7] for details.

Define $P_m^+ = P_m \widetilde{P}$, $Q_m^+ = Q_m \widetilde{Q}_m^+$, $B_m^+ = \widetilde{P}^{\mathrm{T}} B_m \widetilde{Q}$. Let $l = m - p$, P_l^+ and Q_l^+ be the first l columns of P_m^+ and Q_m^+, respectively, q_{l+1}^+ be the $(l+1)$-th column of Q_m^+, $\widetilde{P}_{m,l}$ be the (m, l)-element of $\widetilde{P}$, B_l^+ be the leading $l \times l$ submatrix of B_m^+. Then

$$AQ_l^+ = P_l^+ B_l^+, \tag{4.1}$$

$$A^{\mathrm{T}} P_m^+ = Q_l^+ {B_l^+}^{\mathrm{T}} + (\beta_l \widetilde{P}_{m,l} q_{m+1} + \beta_l^+ q_{l+1}^+) e_l^{\mathrm{T}}. \tag{4.2}$$

Since $\beta_l \widetilde{P}_{m,l} q_{m+1} + \beta_l^+ q_{l+1}^+$ is orthogonal to Q_l^+, (4.1) and (4.2) are the l-step Lanczos bidiagonalization process with the initial vector q_1^+, which can extend to the m-step Lanczos bidiagonalization process. Meanwhile,

$$\gamma q_1^+ = \prod_{j=1}^{p} (A^{\mathrm{T}} A - \mu_j^2 I) q_1, \tag{4.3}$$

where γ is a factor such that $\|q_1^+\| = 1$.

Once the shifts $\mu_1, \mu_2, \ldots, \mu_p$ are chosen, we can run the algorithm described above iteratively until the corresponding residuals are less than tol. Mathematically, we can choose any scalars to be the shifts. However, we can see from (4.3) that if the shifts near the unwanted singular values, the components of the unwanted singular vectors are damped significantly, and q_1^+ has the more components of the wanted singular vectors. As a result, the algorithm may converge faster. This phenomenon suggests us to select the shifts as close as possible to the unwanted singular values.

For the classical implicit restarting technique, the exact shift strategy are used [14,19], and the shifts are taken to be $\widetilde{\sigma}_{l+1}, \widetilde{\sigma}_{l+2}, \ldots, \widetilde{\sigma}_m$, the p smallest approximate singular values. In this paper, we obtain $\widehat{v}_i$, $i = 1, 2, \ldots, k$, which are better than $\widetilde{v}_i$, $i = 1, 2, \ldots, k$. Similar to the analysis of [10, 11], we can find the better shifts by using the information of $\widehat{v}_i$, $i = 1, 2, \ldots, k$.

Make the following orthogonal direct sum decompositions

$$\begin{aligned} \operatorname{span}\{Q_m, q_{m+1}\} &= \operatorname{span}\left\{\widetilde{V}_k\right\} \oplus \operatorname{span}\left\{\widetilde{V}_{m-k}\right\} \oplus \operatorname{span}\{q_{m+1}\}, \\ \operatorname{span}\{Q_m, q_{m+1}\} &= \operatorname{span}\left\{\widehat{V}_k\right\} \oplus \operatorname{span}\left\{\widehat{V}_k\right\}^{\perp}, \end{aligned}$$

where $\widetilde{V}_k = (\widetilde{v}_1, \widetilde{v}_2, \ldots, \widetilde{v}_k)$, $\widetilde{V}_{m-k} = (\widetilde{v}_{k+1}, \widetilde{v}_{k+2}, \ldots, \widetilde{v}_m)$, $\widehat{V}_k = (\widehat{v}_1, \widehat{v}_2, \ldots, \widehat{v}_k)$.

It can be easily verified that the wanted approximate singular values $\widetilde{\sigma}_1, \widetilde{\sigma}_2, \ldots, \widetilde{\sigma}_k$ are the singular values of

$$P_m^{\mathrm{T}} A \widetilde{V}_k$$

and the unwanted singular values $\widetilde{\sigma}_{k+1}, \widetilde{\sigma}_{k+2}, \ldots, \widetilde{\sigma}_m$ are the singular values of

$$P_m^{\mathrm{T}} A \widetilde{V}_{m-k}.$$

Since $\operatorname{span}\left\{\widehat{V}_k\right\}$ includes better information of the wanted singular vectors than $\operatorname{span}\left\{\widetilde{V}_k\right\}$, $\operatorname{span}\left\{\widehat{V}_k\right\}^{\perp}$ includes better information of the unwanted singular vectors than $\operatorname{span}\left\{\widetilde{V}_{m-k}\right\}$ $\oplus \operatorname{span}\{q_{m+1}\}$. Therefore, if we compute the singular values of $P_m^{\mathrm{T}} A \widehat{V}_k^{\perp}$, where $\widehat{V}_k^{\perp}$ is the orthonormal bases of $\operatorname{span}\left\{\widehat{V}_k\right\}^{\perp}$, and choose p smallest of them to be shifts, the obtained shifts may approximate the unwanted singular values better than the classical shifts. As mentioned above, the algorithm may converge faster. Next, we propose a practical approach that can compute the new shifts cheaply and reliably.

It has been proved in [7, Theorem 5.2.2] that if $C = QR$ is a full QR factorization of $C \in \mathcal{R}^{m \times n}$, $m < n$, then the first m columns of Q are the orthonormal bases of $\operatorname{span}\{C\}$ and the last $n - m$ columns of Q are the orthonormal bases of $\operatorname{span}\{C\}^{\perp}$. From the definitions of $\widehat{v}_i$ and $\widetilde{v}_i$, we see that

$$\operatorname{span}\left\{\widehat{V}_k\right\} = \operatorname{span}\left\{\begin{pmatrix} Q_m & q_{m+1} \end{pmatrix} \begin{pmatrix} a_1\widetilde{y}_1 & a_2\widetilde{y}_2 & \cdots & a_k\widetilde{y}_k \\ b_1 & b_2 & \cdots & b_k \end{pmatrix}\right\}.$$

If we compute the following full QR factorization

$$\begin{pmatrix} a_1\widetilde{y}_1 & a_2\widetilde{y}_2 & \cdots & a_k\widetilde{y}_k \\ b_1 & b_2 & \cdots & b_k \end{pmatrix} = \widehat{Q}\widehat{R}$$

and take $\widehat{Q}_{m+1-k}$ to be the last $m+1-k$ columns of $\widehat{Q}$, obviously $(Q_m, q_{m+1})\widehat{Q}_{m+1-k}$ forms the orthonormal bases of span $\left\{\widehat{V}_k\right\}^{\perp}$. Meanwhile,

$$P_m^{\mathrm{T}} A \widehat{V}_k^{\perp} = P_m^{\mathrm{T}} A (Q_m, q_{m+1})\widehat{Q}_{m+1-k} = (B_m, \beta_{m+1} e_m)\widehat{Q}_{m+1-k}. \tag{4.4}$$

As mentioned above, we can compute the singular values of $(B_m, \beta_{m+1} e_m)\widehat{Q}_{m+1-k}$ and take p smallest of them to be the new shifts. It is a small problem since $(B_m, \beta_{m+1} e_m) \cdot \widehat{Q}_{m+1-k}$ is a $m \times (m+1-k)$ matrix and $m \ll N$.

Now, we present the classical Lanczos bidiagonalization method and the modified version of the paper, respectively.

irlb(old): the classical Lanczos bidiagonalization method

1. Given an initial vector q_1 of order N, the Lanczos steps m, the number of the wanted singular triplets k, the number of the shifts p ($\leq m-k$), the convergence tolerance tol.
2. Run the m-step Lanczos bidiagonalization process (2.1) and (2.2).
3. Compute the singular triplets $(\widetilde{\sigma}_i, \widetilde{x}_i, \widetilde{y}_i)$, $i = 1, 2, \ldots, m$ of B_m.
4. Test if $\widetilde{r}_i$, $i = 1, 2, \ldots, k$ are less than $\mathrm{tol} \times \|A\|$. If yes, take $(\widetilde{\sigma}_i, \widetilde{u}_i = P_m \widetilde{x}_i, \widetilde{v}_i = Q_m \widetilde{y}_i)$, $i = 1, 2, \ldots, k$ to be the approximate singular triplets and stop.
5. Take $l = m - p$. Use $\widetilde{\sigma}_{l+1}, \widetilde{\sigma}_{l+2}, \ldots, \widetilde{\sigma}_m$ as shifts and implicitly restart the Lanczos bidiagonalization process.

irlb(new): the modified Lanczos bidiagonalization method

1. Given an initial vector q_1 of order N, the Lanczos steps m, the number of the wanted singular triplets k, the number of the shifts p ($\leq m-k$), the convergence tolerance tol.
2. Run the m-step Lanczos bidiagonalization process (2.1) and (2.2).
3. Compute the singular triplets $(\widetilde{\sigma}_i, \widetilde{x}_i, \widetilde{y}_i)$, $i = 1, 2, \ldots, m$ of B_m.
4. Test if $\widetilde{r}_i$, $i = 1, 2, \ldots, k$ are less than $\mathrm{tol} \times \|A\|$. If yes, take $(\widetilde{\sigma}_i, \widetilde{u}_i = P_m \widetilde{x}_i, \widetilde{v}_i = Q_m \widetilde{y}_i)$, $i = 1, 2, \ldots, k$ to be the approximate singular triplets and stop.
5. Form $\widehat{B}_i$, take s_i and $(a_i^{\mathrm{T}}, b_i^{\mathrm{T}})^{\mathrm{T}}$ to be the smallest singular value and the associate right singular vector of $\widehat{B}_i$, respectively.
6. Test if s_i, $i = 1, 2, \ldots, k$ are less than $\mathrm{tol} \times \|A\|$. If yes, take $(\widetilde{\sigma}_i, \widehat{u}_i = \widetilde{u}_i, \widehat{v}_i = a_i \widetilde{v}_i + b_i q_{m+1})$, $i = 1, 2, \ldots, k$ to be the approximate singular triplets and stop.
7. Compute the singular values of (4.4). Use p smallest of them to be the shifts and implicitly restart the Lanczos bidiagonalization process.

5. Numerical experiments

We run **irlb(old)** and **irlb(new)** on the computational environment given by Table 5.1. The convergence tolerance is taken to be tol $= 10^{-6}$. Since $\|A\|$ is hard to compute, we replace it by the maximum the largest approximate singular value obtained in the current restart and the last restart, which is a good approximation to $\|A\|$. We also take $p = m-k$. All the test matrices are from [5].

CPU	Intel Xeon E3 1230V2 3.30Ghz
RAM	32 GB (Kinston DDR3 1600 Mhz)
Operating system	Windows 7 Professional (64 bit)
Software	MATLAB 2012B
Machine epsilon	$\approx 2.22 \times 10^{-16}$
Stop criteria	residual norms $< \|A\| \times 10^{-6}$

Table 5.1: Computational environment.

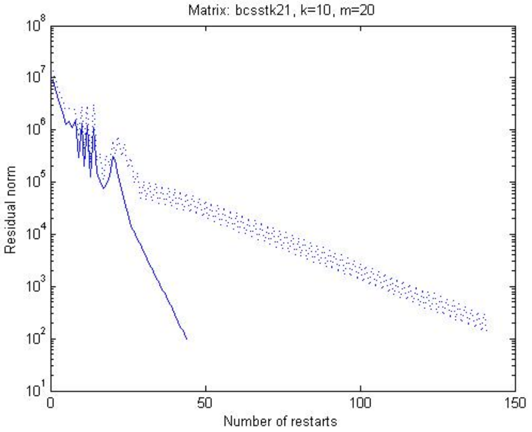

Figure 5.1: Computing ten largest singular triplets of bcsstk21.

Example 5.1. Compute the ten largest singular triplets of bcsstk21, a 3600×3600 matrix, with $k = 10$, $m = 20$.

Figure 5.1 plots the maximum of the corresponding ten residual norms at each restart. We see that, **irlb(new)** takes fewer restarts to reach the desired convergence than those of **irlb(old)**. Before the first restart, the residual norm of **irlb(new)** is smaller than that of **irlb(old)**, as mentioned in Section 3. With the increasing number of restarts, the former descends faster than the latter.

Example 5.2. Compute ten largest singular triplets of saylr4, a square matrix of order 3564, with $k = 10$, $m = 20$.

Figure 5.2 shows that, after 2000 restarts, **irlb(old)** does not converge. The curve of the residual norm descends firstly and then is stagnant. This phenomenon has been analyzed by Jia and Niu in [10, 11]. They point out that, as the cases of eigenproblems, even the approximate singular values converge, the associate approximate singular vectors may converge slowly and irregularly as the separation between the wanted and unwanted approximate singular values is very small. Conversely, **irlb(new)** only takes 103 restarts to reach the desired convergence. It shows that **irlb(new)** may be much better than **irlb(old)**.

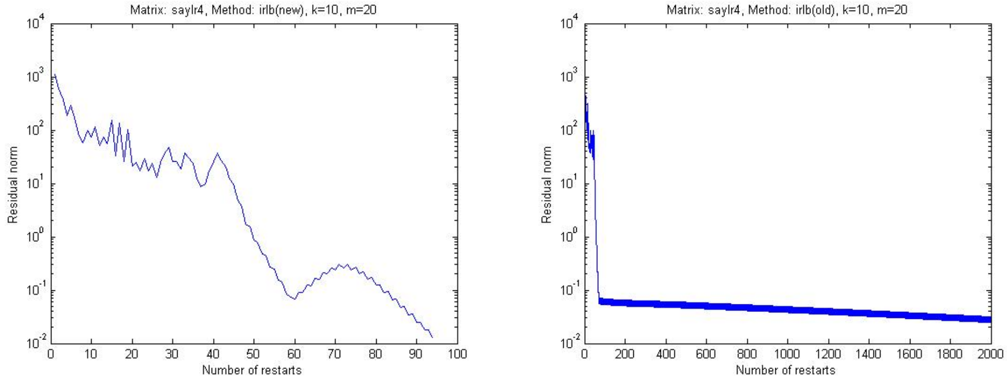

Figure 5.2: Computing ten largest singular triplets of saylr4.

Example 5.3. Compute k largest singular triplets of olm5000 for different k and m. The size of olm5000 is 5000×5000.

Since in the new algorithm we need compute Aq_{m+1}, we run **irlb(new)** with m-step Lanczos bidiagonalization process and **irlb(old)** with m and $m+1$ step Lanczos bidiagonalization process, respectively. See Table 5.2 for the numerical results.

In Table 5.2, denote by "iter" the number of restarts, by "time" the CPU time in second. When the number of restarts reaches 2000, we stop the algorithm.

From Table 5.2, it can be easily seen that, **irlb(new)** is more efficient than **irlb(old)**. It can save about 50% restarts and CPU time. Dividing "time" by "iter", we find that the CPU time of each restarts of **irlb(new)** are only a little more than those of **irlb(old)**. The reason is that the total extra cost of **irlb(new)** includes: (1) one extra matrix-vector product (needs at most nonzero number of the elements of olm5000 multiplications), (2) computing the SVD of $\widehat{B}$, a 2×2 matrix, (3) computing the new shifts (needs $O((m+1)^2(m-k)$ multiplications). It is relatively small since olm5000 is a sparse matrix and $m \ll N$. Even for dense matrices, the total CPU time of **irlb(new)** may be still smaller since **irlb(new)** may converges faster than **irlb(old)**. We also see that **irlb(new)** with parameter m is much better than **irlb(old)** with parameter $m+1$.

irlb(new)						
	$k=3$		$k=5$		$k=10$	
m	iter	time	iter	time	iter	time
20	627	64	522	46	1117	67
30	261	53	441	90	308	53
40	147	56	235	84	152	49
50	104	56	150	84	92	47
irlb(old)						
	$k=3$		$k=5$		$k=10$	
m	iter	time	iter	time	iter	time
20	1399	137	>2000	-	>2000	-
30	585	118	873	176	578	99
40	327	124	461	164	276	87
50	206	119	291	162	164	83
irlb(old)						
	$k=3$		$k=5$		$k=10$	
m	iter	time	iter	time	iter	time
21	1255	137	>2000	-	1792	127
31	549	126	817	166	535	100
41	313	126	439	167	261	87
51	198	120	278	165	150	79

Table 5.2: Computing the largest singular triplets of olm5000 with different k and m.

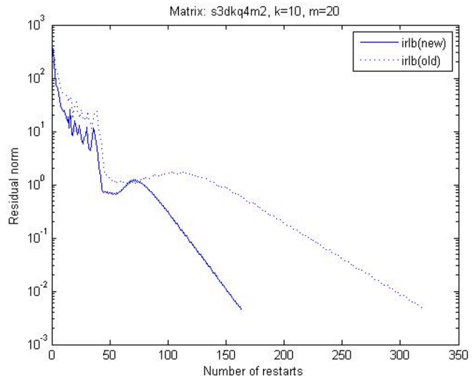

Figure 5.3: Computing ten largest singular triplets of s3dkq4m2.

Example 5.4. The test matrix is s3dkq4m2, a square matrix of order 90449. We take $k = 10$ and $m = 20$. Figure 5.3 reports the numerical results. The scale of s3dkq4m2 is very large, but **irlb(new)** and **irlb(old)** still work well, and the former is much better. In fact, the keypoint of computing singular triplets is the separation of the singular triplets rather than the scale of the matrices.

6. Conclusions

In this paper, we analyze the classical Lanczos bidiagonalization method and find that it generates $2m + 1$ base vectors but only uses $2m$ of them. We make two improvements on the classical Lanczos bidiagonalization method. One is replacing the approximate singular triplets by new ones, which are the linear combination of the old approximate singular vectors and the unused base vector. The new approximate singular vectors minimize the corresponding residual norms and can be computed from 2×2 SVD problems. The other is replacing the unwanted approximate singular values by new scalars as shifts. The new shifts are superior and can be computed cheaply and reliably. Numerical experiments show that, after two improvements, the modified method is much better than the classical one.

The corresponding MATLAB code can be obtained from the authors upon request.

Acknowledgments

We thank two referees very much for their valuable suggestions and comments, which made us improve the presentation and the algorithm implementation of the paper considerably. We thank Professor Zhongxiao Jia for his helpful discussions. We also thank Professor Lothar Reichel and Professor James Bagalama since we used their MATLAB code ablanzbd.m to build the m-step Lanczos bidiagonalization process.

This paper was supported by the National Science Foundation of China (Nos. 11001039, 61202254), the Program for Liaoning Excellent Talents in University (No. LJQ2012114), the Liaoning Provincial National Science Foundation of China (No. 2014020137), the Fundamental Research Funds for the Central Universities (No. DC201502050302).

References

[1] J. Baglama and L. Reichel, *Augmented implicitly restarted Lanczos bidiagonalization methods*, SIAM J. Sci. Comput. **27** (2005), no. 1, 19–42.
http://dx.doi.org/10.1137/04060593x

[2] ______, *Restarted block Lanczos bidiagonalization methods*, Numer. Algorithms **43** (2006), no. 3, 251–272. `http://dx.doi.org/10.1007/s11075-006-9057-z`

[3] ______, *An implicitly restarted block Lanczos bidiagonalization method using Leja shifts*, BIT **53** (2013), no. 2, 285–310. `http://dx.doi.org/10.1007/s10543-012-0409-x`

[4] Å. Björck, E. Grimme and P. Van Dooren, *An implicit shift bidiagonalization algorithm for ill-posed systems*, BIT **34** (1994), no. 4, 510–534. `http://dx.doi.org/10.1007/bf01934265`

[5] B. Boisvert, R. Pozo, K. Remington, B. Miller and R. Lipman, *Matrix Market*, available online at `http://math.nist.gov/MatrixMarket/`, 2004.

[6] G. H. Golub, F. T. Luk and M. L. Overton, *A block Lanczos method for computing the singular values and corresponding singular vectors of a matrix*, ACM Trans. Math. Software **7** (1981), no. 2, 149–169. `http://dx.doi.org/10.1145/355945.355946`

[7] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Forth edition, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, 2013.

[8] V. Hernandez, J. E. Roman and A. Tomas, *A robust and efficient parallel SVD solver based on restarted Lanczos bidiagonalization*, Electron. Trans. Numer. Anal. **31** (2008), 68–85.

[9] Z. Jia and L. Elsner, *Improving eigenvectors in Arnoldi's method*, J. Comput. Math. **18** (2000), no. 3, 265–276.

[10] Z. Jia and D. Niu, *An implicitly restarted refined bidiagonalization Lanczos method for computing a partial singular value decomposition*, SIAM J. Matrix Anal. Appl. **25** (2003), no. 1, 246–265. `http://dx.doi.org/10.1137/s0895479802404192`

[11] ______, *A refined harmonic Lanczos bidiagonalization method and an implicitly restarted algorithm for computing the smallest singular triplets of large matrices*, SIAM J. Sci. Comput. **32** (2010), no. 2, 714–744. `http://dx.doi.org/10.1137/080733383`

[12] E. Kokiopoulou, C. Bekas and E. Gallopoulos, *Computing smallest singular triplets with implicitly restarted Lanczos bidiagonalization*, Appl. Numer. Math. **49** (2004), no. 1, 39–61. `http://dx.doi.org/10.1016/j.apnum.2003.11.011`

[13] R. M. Larsen, *Lanczos Bidiagonalization with Partial Reorthogonalization*, DAIMI Report Series **27** (1998), no. 537, 1–101. `http://dx.doi.org/10.7146/dpb.v27i537.7070`

[14] ______, *Combining implicit restarts and partial reorthogonalization in Lanczos bidiagonalization*. http://soi.stanford.edu/~rmunk/PROPACK

[15] D. Niu and X. Yuan, *An implicitly restarted Lanczos bidiagonalization method with refined harmonic shifts for computing smallest singular triplets*, J. Comput. Appl. Math. **260** (2014), 208–217. http://dx.doi.org/10.1016/j.cam.2013.09.066

[16] B. N. Parlett, *The Symmetric Eigenvalue Problem*, SIAM, Philadelphia, PA, 1998. http://dx.doi.org/10.1137/1.9781611971163

[17] Y. Saad, *Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices*, Linear Algebra Appl. **34** (1980), 269–295. http://dx.doi.org/10.1016/0024-3795(80)90169-x

[18] H. D. Simon and H. Zha, *Low-rank matrix approximation using the Lanczos bidiagonalization process with applications*, SIAM J. Sci. Comput. **21** (2000), no. 6, 2257–2274. http://dx.doi.org/10.1137/s1064827597327309

[19] D. C. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal. Appl. **13** (1992), no. 1, 357–385. http://dx.doi.org/10.1137/0613025

[20] M. Stoll, *A Krylov-Schur approach to the truncated SVD*, Linear Algebra Appl. **436** (2012), no. 8, 2795–2806. http://dx.doi.org/10.1016/j.laa.2011.07.022

Datian Niu
School of Information and Computation Science, Beifang University of Nationalities,
Yinchuan 750021, China
and
School of Science, Dalian Minzu University, Dalian 116600, China
E-mail address: niudt@dlnu.edu.cn

Jiana Meng
School of Computer Science and Engineering, Dalian Minzu University, Dalian 116600,
China
E-mail address: mengjn@dlnu.edu.cn