SIAM J. SCI. COMPUT.
Vol. 21, No. 6, pp. 2257–2274

LOW-RANK MATRIX APPROXIMATION USING THE LANCZOS BIDIAGONALIZATION PROCESS WITH APPLICATIONS*

HORST D. SIMON† AND HONGYUAN ZHA‡

Abstract. Low-rank approximation of large and/or sparse matrices is important in many applications, and the singular value decomposition (SVD) gives the best low-rank approximations with respect to unitarily-invariant norms. In this paper we show that good low-rank approximations can be directly obtained from the Lanczos bidiagonalization process applied to the given matrix without computing any SVD. We also demonstrate that a so-called one-sided reorthogonalization process can be used to maintain an adequate level of orthogonality among the Lanczos vectors and produce accurate low-rank approximations. This technique reduces the computational cost of the Lanczos bidiagonalization process. We illustrate the efficiency and applicability of our algorithm using numerical examples from several applications areas.

Key words. singular value decomposition, space matrices, Lanczos algorithms, low-rank approximation

AMS subject classifications. 14A18, 65F15, 65F50

PII. S1064827597327309

1. Introduction.

In many applications such as compression of multiple-spectral image cubes, regularization methods for ill-posed problems, and latent semantic indexing in information retrieval for large document collections, to name a few, it is necessary to find a low-rank approximation of a given matrix $A \in \mathcal{R}^{m\times n}$ [4, 9, 17]. Often A is a sparse or structured rectangular matrix, and sometimes either $m \gg n$ or $m \ll n$. The theory of SVD provides the following characterization of the best rank-j approximation of A in terms of Frobenius norm $\|\cdot\|_F$ [7].

THEOREM 1.1. *Let the SVD of $A \in \mathcal{R}^{m\times n}$ be $A = P\Sigma Q^T$ with $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_{\min(m,n)})$, $\sigma_1 \geq \cdots \geq \sigma_{\min(m,n)}$, and P and Q orthogonal. Then for $1 \leq j \leq n$,*

$$\sum_{i=j+1}^{\min(m,n)} \sigma_i^2 = \min\{\ \|A - B\|_F^2 \mid \operatorname{rank}(B) \leq j\}.$$

And the minimum is achieved with $A_j \equiv P_j \operatorname{diag}(\sigma_1, \ldots, \sigma_j) Q_j^T$, where P_j and Q_j are the matrices formed by the first j columns of P and Q, respectively.

It follows from Theorem 1.1 that once the SVD of A is available, the best rank-j approximation of A is readily computed. When A is large and/or sparse, however, the computation of the SVD of A can be costly, and if we are only interested in some A_j with $j \ll \min(m, n)$, the computation of the complete SVD of A is rather

*Received by the editors September 16, 1997; accepted for publication (in revised form) September 17, 1999; published electronically May 19, 2000. This work was supported by the Director, Office of Energy Research, Office of Laboratory Policy and Infrastructure Management of the U.S. Department of Energy contract DE-AC03-76SF00098, and by Office of Computational and Technology Research, Division of Mathematical, Information, and Computational Sciences of the U.S. Department of Energy contract 76SF00098.

http://www.siam.org/journals/sisc/21-6/32730.html

†NERSC, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720 (zha@cse.psu.edu).

‡Department of Computer Science and Engineering, The Pennsylvania State University, 307 Pond Laboratory, University Park, PA 16802-6103 (hdsimon@lbl.gov). The work of this author was supported by NSF grant CCR-9619452.

wasteful. Also, in many applications it is not necessary to compute A_j to very high accuracy since A itself may contain certain errors. It is therefore desirable to develop less expensive alternatives for computing good approximations of A_j. In this paper, we explore applying Lanczos bidiagonalization process for finding approximations of A_j. Lanczos bidiagonalization process has been used for computing a few dominant singular triplets (singular values and the corresponding left and right singular vectors) of large sparse matrices [1, 3, 6, 8]. We will show that in many cases of interest, good low-rank approximations of A can be directly obtained from the Lanczos bidiagonalization process of A without computing any SVD. We will also explore relations between the levels of orthogonality of the left Lanczos vectors and the right Lanczos vectors and propose an efficient reorthogonalization scheme that can be used to reduce the computational cost of the Lanczos bidiagonalization process. The rest of the paper is organized as follows. In section 2, we review the Lanczos bidiagonalization process and its several variations in finite precision arithmetic. In section 3, we discuss a priori error estimations for using the low-rank approximations obtained from Lanczos bidiagonalization process. We also derive stopping criteria for Lanczos bidiagonalization process in the context of computing low-rank approximations. Section 4 is devoted to orthogonalization issues in Lanczos bidiagonalization process and a reorthogonalization scheme is proposed. In section 5, we perform numerical experiments on test matrices from a variety of applications areas. Section 6 concludes the paper.

2. The Lanczos bidiagonalization process. Bidiagonalization of a rectangular matrix using orthogonal transformations such as Householder transformations and Givens rotations was first proposed in [5]. It was later adapted to solving large sparse least squares problems [15] and to finding a few dominant singular triplets of large sparse matrices [1, 3, 6]. For solving least squares problems the orthogonality of the left and right Lanczos vectors is usually not a concern and therefore no reorthogonalization is incorporated in the algorithm LSQR in [15].[1] For computing a few dominant singular triplets, one approach is to completely ignore the issue of loss of orthogonality during the Lanczos bidiagonalization process and later on to identify and eliminate those spurious singular values that are copies of true ones [3]. We will not pursue this approach since spurious singular values will cause considerable complication in forming approximations of A_j discussed in the previous section. We opt to use the approach that will maintain a certain level of orthogonality among the Lanczos vectors [8, 16, 18, 19, 20]. Even within this approach there exist several variations depending on how reorthogonalization is implemented. For example in *SVDPACK*, a state-of-the-art software package for computing dominant singular triplets of large sparse matrices [22], implementations of Lanczos *tridiagonalization* process applied to either A^TA or the 2-cyclic matrix `[0 A; A' 0]` with *partial* reorthogonalization are provided. Interesting enough, for the coupled two-term recurrence that will be detailed in a moment, only a block version with *full* reorthogonalization is implemented in *SVDPACK*.

Now we describe the Lanczos bidiagonalization process presented in [5, 15, 3]. Let b be a starting vector, for $i = 1, 2, \ldots$, compute

$$\beta_1 u_1 = b, \quad \alpha_1 v_1 = A^T u_1,$$

[1] Maintaining a certain level of orthogonality among the Lanczos vectors will accelerate the convergence at the expense of more computational cost and storage requirement [18, Section 4].

$$\begin{aligned} \beta_{i+1}u_{i+1} &= Av_i - \alpha_i u_i, \\ \alpha_{i+1}v_{i+1} &= A^T u_{i+1} - \beta_{i+1}v_i. \end{aligned} \tag{2.1}$$

Here nonnegative α_i and β_i are chosen such that $\|u_i\| = \|v_i\| = 1$. Throughout the rest of the paper $\|\cdot\|$ always denotes either the vector or matrix two-norm. In compact matrix form the above equations can be written as

$$\begin{aligned} U_{k+1}(\beta_1 e_1) &= b, \\ AV_k &= U_{k+1}B_{k+1}(:, 1:k), \\ A^T U_{k+1} &= V_{k+1}B_{k+1}^T, \end{aligned} \tag{2.2}$$

where $B_{k+1} \in \mathcal{R}^{(k+1)\times(k+1)}$ is lower bidiagonal,

$$B_{k+1} = \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_{k+1} & \alpha_{k+1} \end{bmatrix}, \qquad \begin{aligned} U_{k+1} &= [u_1, \ldots, u_{k+1}], \\ V_{k+1} &= [v_1, \ldots, v_{k+1}], \end{aligned}$$

and $B_{k+1}(:, 1:k)$ is B_{k+1} with the last column removed.

Remark. There is another version of the Lanczos bidiagonalization recurrence [5, 3],

$$\begin{aligned} &\alpha_1 v_1 = b, \;\; \beta_1 u_1 = Av_1, \\ &\alpha_{i+1}v_{i+1} = A^T u_i - \beta_i v_i, \\ &\beta_{i+1}u_{i+1} = Av_{i+1} - \alpha_{i+1}v_i. \end{aligned}$$

For A with more columns than rows, this version is usually better than (2.2) because the chances of introducing extra zero singular values arising from $m \neq n$ is reduced [5, 3]. However, it is easy to see that the two versions of bidiagonalization recurrences are equivalent in the sense that if we interchange the roles of A and A^T, u_i and v_i, and α_i and β_i in (2.2), we obtain the above recurrence. In another word, we may simply apply (2.2) to A^T to obtain the above recurrence. Therefore in what follows we will deal exclusively with recurrence (2.2). When the need arises we will simply apply recurrence (2.2) to A^T.

Now we take into account the effects of rounding errors, and we denote the computed version of a quantity by adding "$\hat{}$". Following the error analysis in [14], it is straightforward to show that in finite precision arithmetic, (2.2) and (2.2) become

$$\begin{aligned} &\hat{\beta}_1\hat{u}_1 = b, \;\; \hat{\alpha}_1\hat{v}_1 = A^T\hat{u}_1 + g_1, \\ &\hat{\beta}_{i+1}\hat{u}_{i+1} = A\hat{v}_i - \hat{\alpha}_i\hat{u}_i - f_i, \qquad i = 1, 2, \ldots, \\ &\hat{\alpha}_{i+1}\hat{v}_{i+1} = A^T\hat{u}_{i+1} - \hat{\beta}_{i+1}\hat{v}_i - g_{i+1}, \end{aligned} \tag{2.3}$$

and in compact matrix form,

$$\begin{aligned} \hat{U}_{k+1}(\hat{\beta}_1 e_1) &= b, \\ A\hat{V}_k &= \hat{U}_{k+1}\hat{B}_{k+1}(:, 1:k) + F_k, \\ A^T\hat{U}_{k+1} &= \hat{V}_{k+1}\hat{B}_{k+1}^T + G_{k+1}, \end{aligned} \tag{2.4}$$

where $\|F_k\| = O(\|A\|_F \epsilon_M)$ and $\|G_{k+1}\| = O(\|A\|_F \epsilon_M)$ with ϵ_M the machine epsilon, and

$$F_i = [f_1, \ldots, f_i], \quad G_i = [g_1, \ldots, g_i].$$

To find a few dominant singular value triplets of A, one computes the SVD of B_k. The singular values of B_k are then used as approximations of the singular values of A and the singular vectors of B_k are combined with the left and right Lanczos vectors $\{U_k\}$ and $\{V_k\}$ to form approximations of the singular vectors of A [3, 1].

Now if one is only interested in finding a low-rank approximation of A, it is desirable that a more direct approach be used without computing the SVD of B_k. From the convergence theory of Lanczos process, we know that U_k and V_k contains good approximations of the dominant singular vectors of A. So it is quite natural to use $J_k \equiv U_k B_k V_k^T$ as an approximation of A. In the next section we consider a priori estimation of $\omega_k \equiv \|A - J_k\|_F$.

3. Error estimation and stopping criterion. In this section we will assess the error of using J_k as an approximation of A. We will also discuss ways to compute ω_k recursively in finite precision arithmetic. Many a priori error bounds have been derived for the Ritz values/vectors computed by the Lanczos tridiagonalization process [16]. It turns out that our problem of estimating ω_k a priori is rather straightforward. It all boils down to how well a singular vector can be approximated from a Krylov subspace. To proceed we need a result on the approximation of an eigenvector of a symmetric matrix from a Krylov subspace [16, Section 12.4].

LEMMA 3.1. *Let $C \in \mathcal{R}^{n\times n}$ be symmetric and f an arbitrary vector. Define*

$$\mathcal{K}_m \equiv \operatorname{span}\{f, Cf, \ldots, C^{m-1}f\}.$$

Let $C = Z \operatorname{diag}(\lambda_i) Z^T$ be the eigendecomposition of C with $\lambda_1 \geq \cdots \geq \lambda_n$ its eigenvalues. Write $Z = [z_1, \ldots, z_n]$ and define $\mathcal{Z}_j = \operatorname{span}\{z_1, \ldots, z_j\}$. Then

$$\tan \angle(z_j, \mathcal{K}_m) \leq \frac{\sin \angle(f, \mathcal{Z}_j) \Pi_{v=1}^{j-1} (\lambda_v - \lambda_n)/(\lambda_v - \lambda_j)}{\cos \angle(f, z_j) T_{m-j}(1 + 2\gamma)},$$

where $\gamma = (\lambda_j - \lambda_{j+1})/(\lambda_{j+1} - \lambda_n)$.

Now let $A = P \operatorname{diag}(\sigma_i) Q^T$ be the SVD of A, and write $P = [p_1, p_2, \ldots, p_m]$. Furthermore, let $P_{U_k}^{\perp} \equiv (I - U_k U_k^T)$, the orthogonal projector onto the subspace $\operatorname{span}\{U_k\}^{\perp}$, the orthogonal complement of $\operatorname{span}\{U_k\}$. We have the following error estimation.

THEOREM 3.2. *Let $\mathcal{P}_i \equiv \operatorname{span}\{p_1, \ldots, p_i\}$. Assume Lanczos bidiagonalization process starts with b as in* (2.2). *Then for any j with $1 < j < n$ and $k > j$,*

$$\omega_k^2 \leq \sum_{i=j+1}^{n} \sigma_i^2 + \sum_{i=1}^{j} \sigma_i^2 \left(\frac{\sin \angle(b, \mathcal{P}_i) \Pi_{v=1}^{i-1} (\sigma_v^2 - \sigma_n^2)/(\sigma_v^2 - \sigma_i^2)}{\cos \angle(b, p_i) T_{k-i}(1 + 2\gamma_i)} \right)^2, \tag{3.1}$$

where $\gamma_i = (\sigma_i^2 - \sigma_{i+1}^2)/(\sigma_{i+1}^2 - \sigma_n^2)$.

Proof. From (2.2) we have $U_k^T A = B_k V_k^T$. Hence

$$\omega_k^2 = \|A - U_k B_k V_k^T\|_F^2 = \|(I - U_k U_k^T) A\|_F^2.$$

Using the SVD of $A = P \operatorname{diag}(\sigma_i) Q^T$ one can verify that

$$\omega_k^2 = \|[\sigma_1 P_{U_k}^{\perp} p_1, \ldots, \sigma_n P_{U_k}^{\perp} p_n]\|_F^2,$$

where we have assumed that $m \geq n$. Now arrange the singular values of A in the following order:

$$\sigma_n \leq \cdots \leq \sigma_{j+1} \leq \sigma_j \leq \cdots \leq \sigma_1.$$

We have the bound

$$\|A - J_k\|_F^2 \leq \sum_{i=j+1}^{n} \sigma_i^2 + \sum_{i=1}^{j} \sigma_i^2 \|P_{U_k}^{\perp} p_i\|^2.$$

It is readily verified that $\text{span}\{U_k\} = \text{span}[b, AA^T b, \ldots, (AA^T)^{k-1} b] \equiv \mathcal{K}_k$. Therefore

$$\|P_{U_k}^{\perp} p_i\| = |\sin \angle(p_i, \mathcal{K}_k)| \leq |\tan \angle(p_i, \mathcal{K}_k)|.$$

Applying Lemma 3.1 with $C = AA^T$ and $b = f$ completes the proof. ∎

REMARK. Notice that the square root of the first term on the right of (3.1), $(\sum_{i=j+1}^{n} \sigma_i^2)^{1/2}$, is the error of the best rank-j approximation of A in Frobenius norm. For $k > j$, usually $\text{rank}(J_k) > j$. The a priori estimation of the above theorem states that ω_k^2 will approach $\|A - A_j\|_F^2$ when k gets large. In many examples we will discuss later in section 5, even for a k that is only slightly larger than j, ω_k^2 is already very close to $\|A - A_j\|_F^2$.

Now we examine the issue of stopping criteria. The accuracy of using $J_k = U_k B_k V_k^T$ as a low-rank approximation of A is measured by ω_k which can be used as a stopping criterion in the iterative Lanczos bidiagonalization process, i.e., the iterative process will be terminated when $\omega_k \leq \texttt{tol}$ with `tol` a user supplied tolerance. Therefore it will be very helpful to find an inexpensive way to compute ω_k for $k = 1, 2, \ldots$. We first show that ω_k is a monotonically decreasing function of k and it can be computed recursively.

PROPOSITION 3.3. *Let* $\omega_k = \|A - J_k\|_F$. *Then* $\omega_{k+1}^2 = \omega_k^2 - \alpha_{k+1}^2 - \beta_{k+1}^2$, *where* α_{k+1} *and* β_{k+1} *are from* (2.2).

Proof. Notice that

$$\omega_{k+1}^2 = \|A - U_{k+1} B_{k+1} V_{k+1}^T\|_F^2 = \|(I - U_{k+1} U_{k+1}^T) A\|_F^2.$$

Now write $I - U_k U_k^T = (I - U_{k+1} U_{k+1}^T) + u_{k+1} u_{k+1}^T$. Notice that $(I - U_{k+1} U_{k+1}^T) A$ and $u_{k+1} u_{k+1}^T A$ are orthogonal to each other; we obtain

$$\omega_k^2 = \|(I - U_{k+1} U_{k+1}^T) A\|_F^2 + \|u_{k+1} u_{k+1}^T A\|_F^2 = \omega_{k+1}^2 + \|A^T u_{k+1}\|^2.$$

The proof is completed by noticing that $\|A^T u_{k+1}\|^2 = \alpha_{k+1}^2 + \beta_{k+1}^2$. ∎

Proposition 3.3 shows that $\omega_k^2 = \omega_{k+1}^2 + \alpha_{k+1}^2 + \beta_{k+1}^2$ in exact arithmetic. Now we want to examine to what extent the above relation still holds when the effects of rounding errors need to be taken into consideration. In finite precision computation we have (cf. (2.4))

$$A^T \hat{U}_k = \hat{V}_k \hat{B}_k^T + G_k,$$

where G_k represents the effects of rounding errors and $\|G_k\|_F = O(\|A\|_F \epsilon_M)$ with ϵ_M the machine epsilon. It follows that

$$\hat{\omega}_k^2 \equiv \|A - \hat{U}_k \hat{B}_k \hat{V}_k^T\|_F^2 = \|(I - \hat{U}_k \hat{U}_k^T) A + \hat{U}_k G_k^T\|_F^2.$$

In finite precision computation, due to the loss of orthogonality the matrices $\hat{U}_k$ and $\hat{V}_k$ are not necessarily orthonormal anymore. Define

$$\eta(\hat{U}_k) = \|I - \hat{U}_k^T \hat{U}_k\|, \quad \eta(\hat{V}_k) = \|I - \hat{V}_k^T \hat{V}_k\|.$$

These two quantities measure the level of orthogonality among the columns of $\hat{U}_k$ and $\hat{V}_k$, respectively. In the following we will also need these readily-verified inequalities:

$$\eta(\hat{U}_k) \le \eta(\hat{U}_{k+1}), \quad \eta(\hat{V}_k) \le \eta(\hat{V}_{k+1}).$$

THEOREM 3.4. *In finite precision arithmetic, $\hat{\omega}_k$ satisfies*

$$\begin{aligned}\hat{\omega}_k^2 &= \hat{\omega}_{k+1}^2 + \hat{\alpha}_{k+1}^2 + \hat{\beta}_{k+1}^2 + O(\|A\|_F^2 \eta(\hat{U}_{k+1})(1+\eta(\hat{U}_{k+1}))(1+m\epsilon_M)) \\ &\quad + O(\|A\|_F^2(1+\eta(\hat{U}_{k+1}))^2\epsilon_M) + O(\|A\|_F^2\eta(\hat{V}_{k+1})).\end{aligned}$$

Proof. We write $\hat{\omega}_k^2 = \|(I - \hat{U}_k\hat{U}_k^T)A\|_F^2 + \|\hat{U}_k G_k^T\|_F^2 + \text{term}_1$, where

$$|\text{term}_1| = 2|\text{trace}(A^T(I - \hat{U}_k\hat{U}_k^T)\hat{U}_k G_k^T)| = O(\|A\|_F^2\eta(\hat{U}_{k+1})(1+\eta(\hat{U}_{k+1}))\epsilon_M).$$

Now split $(I - \hat{U}_k\hat{U}_k^T)A$ as $(I - \hat{U}_{k+1}\hat{U}_{k+1}^T)A + \hat{u}_{k+1}\hat{u}_{k+1}^T A$ and write

$$\|(I - \hat{U}_k\hat{U}_k^T)A\|_F^2 = \|(I - \hat{U}_{k+1}\hat{U}_{k+1}^T)A\|_F^2 + \|\hat{u}_{k+1}\hat{u}_{k+1}^T A\|_F^2 + \text{term}_2. \tag{3.2}$$

We have the bound,

$$|\text{term}_2| = 2|\text{trace}(A^T(I - \hat{U}_{k+1}\hat{U}_{k+1}^T)\hat{u}_{k+1}\hat{u}_{k+1}^T A)| = O(\|A\|_F^2\eta(\hat{U}_{k+1})(1+\eta(\hat{U}_{k+1}))).$$

Since $\hat{u}_{k+1}\hat{u}_{k+1}^T A$ is rank-one, we have $\|\hat{u}_{k+1}\hat{u}_{k+1}^T A\|_F^2 = \|\hat{u}_{k+1}\|\|A^T\hat{u}_{k+1}\|$. Now it follows from $A^T\hat{u}_{k+1} = \hat{\alpha}_{k+1}\hat{v}_{k+1} + \hat{\beta}_{k+1}\hat{v}_k - g_{k+1}$ (cf. (2.2)) that

$$\begin{aligned}\|A^T\hat{u}_{k+1}\| &= (\hat{\alpha}_{k+1}^2 + \hat{\beta}_{k+1}^2)(1+\epsilon_M) + \|g_{k+1}\|^2 + 2\hat{\alpha}_{k+1}\hat{\beta}_{k+1}\hat{v}_k^T\hat{v}_{k+1} \\ &\quad + 2(\hat{\alpha}_{k+1} + \hat{\beta}_{k+1})O(\epsilon_M) \\ &= \hat{\alpha}_{k+1}^2 + \hat{\beta}_{k+1}^2 + O(\|A\|_F^2\eta(\hat{V}_{k+1}) + O(\|A\|_F\epsilon_M).\end{aligned}$$

Substituting the above estimates into (3.2) completes the proof. ∎

Therefore, modulus the level of orthogonality of $\hat{U}_{k+1}$ and $\hat{V}_{k+1}$, the recursive formula $\hat{\omega}_{k+1}^2 = \hat{\omega}_k^2 - \hat{\alpha}_{k+1}^2 - \hat{\beta}_{k+1}^2$ can still be used to compute $\hat{\omega}_{k+1}$.

REMARK. With some extra technical assumptions and further assuming $\|A\|_F = O(1)$, we can improve the above result which roughly says that

$$\hat{\omega}_{k+1}^2 = \hat{\omega}_k^2 - \hat{\alpha}_{k+1}^2 - \hat{\beta}_{k+1}^2 + O(\eta(\hat{U}_{k+1})) + O(\eta(\hat{V}_{k+1}))$$

to

$$\begin{aligned}\hat{\omega}_{k+1}^2 &= \hat{\omega}_k^2 - \hat{\alpha}_{k+1}^2 - \hat{\beta}_{k+1}^2 \\ &\quad + O(\eta^2(\hat{U}_{k+1})) + O(\eta^2(\hat{V}_{k+1})) + O(\eta(\hat{U}_{k+1})\eta(\hat{V}_{k+1})).\end{aligned}$$

The proof is rather technical and is therefore omitted here. Interested readers are referred to [21] for the complete proof.

4. Level of orthogonality and reorthogonalization. It should not come as a surprise that the level of orthogonality among the computed left Lanczos vectors $\{\hat{U}_k\}$ and the level of orthogonality among the computed right Lanczos vectors $\{\hat{V}_k\}$ are closely related to each other since the columns of $\hat{U}_k$ and $\hat{V}_k$ are generated by the coupled two-term recurrence (2.3). The following result quantifies this claim.

PROPOSITION 4.1. *Assume that $\hat{B}_k$ generated by the two-term recurrence* (2.3) *is nonsingular. Then*

$$\eta(\hat{V}_k) \le \|\hat{B}_k^{-1}\|\|\hat{B}_{k+1}(:, 1:k)\|\eta(\hat{U}_{k+1}) + O(\|\hat{B}_k^{-1}\|\|A\|_F\epsilon_M),$$

and with $\sigma_{\min}(\cdot)$ denoting the smallest singular value of a matrix,

$$\eta(\hat{U}_{k+1}) \le \frac{\|\hat{B}_k\|\eta(\hat{V}_k)}{2\sigma_{\min}(\hat{B}_{k+1}(:, 1:k))} + O\left(\frac{\|A\|_F\epsilon_M}{\sigma_{\min}(\hat{B}_{k+1}(:, 1:k))}\right).$$

Proof. We can write (2.4) as

$$A\hat{V}_k = \hat{U}_k\hat{B}_k + \hat{\beta}_{k+1}\hat{u}_{k+1}e_k^T + F_k, \quad A^T\hat{U}_k = \hat{V}_k\hat{B}_k^T + G_k,$$

where $\|F_k\|_F$ and $\|G_k\|_F$ are of the order of machine epsilon. Therefore we have the following relations:

$$\begin{aligned}\hat{U}_k^T A\hat{V}_k &= \hat{U}_k^T\hat{U}_k\hat{B}_k + \hat{\beta}_{k+1}\hat{U}_k^T\hat{u}_{k+1}e_k^T + \hat{U}_k^T F_k,\\ \hat{U}_k^T A\hat{V}_k &= \hat{B}_k\hat{V}_k^T\hat{V}_k + G_k^T\hat{V}_k.\end{aligned}$$

This leads to

$$\begin{aligned}\hat{B}_k(I - \hat{V}_k^T\hat{V}_k) &= (I - \hat{U}_k^T\hat{U}_k)\hat{B}_k - \hat{\beta}_{k+1}\hat{U}_k^T\hat{u}_{k+1}e_k^T - \hat{U}_k^T F_k + G_k^T\hat{V}_k^T\\ &= [I - \hat{U}_k^T\hat{U}_k, -\hat{U}_k^T\hat{u}_{k+1}]\hat{B}_{k+1}(:, 1:k) - \hat{U}_k^T F_k + G_k^T\hat{V}_k^T.\end{aligned} \tag{4.1}$$

Since $\hat{B}_k$ is nonsingular and $\|[I - \hat{U}_k^T\hat{U}_k, -\hat{U}_k^T\hat{u}_{k+1}]\| \le \eta(\hat{U}_{k+1})$, we have

$$\eta(\hat{V}_k) \le \|\hat{B}_k^{-1}\|\|\hat{B}_{k+1}(:, 1:k)\|\eta(\hat{U}_{k+1}) + O(\|\hat{B}_k^{-1}\|A\|_F\|\epsilon_M).$$

On the other hand, it follows from (4.1) that

$$\begin{aligned}\sigma_{\min}(\hat{B}_{k+1}(:, 1:k))\|[I - \hat{U}_k^T\hat{U}_k, -\hat{U}_k^T\hat{u}_{k+1}]\| &\le \|[I - \hat{U}_k^T\hat{U}_k, -\hat{U}_k^T\hat{u}_{k+1}]\hat{B}(:, 1:k)\|\\ &\le \|\hat{B}_k\|\|I - \hat{V}_k^T\hat{V}_k\| + O(\|A\|_F\epsilon_M).\end{aligned}$$

It is easy to see that $\|I - \hat{U}_{k+1}^T\hat{U}_{k+1}\| \le 2\|[I - \hat{U}_k^T\hat{U}_k, -\hat{U}_k^T\hat{u}_{k+1}]\|$, and B_k nonsingular implies $\sigma_{\min}(\hat{B}_{k+1}(:, 1:k)) > 0$. Combining the above two inequalities completes the proof. ∎

The above result says that as long as $\hat{B}_k$ and $\hat{B}_{k+1}(:, 1:k)$ are not very ill-conditioned, the level of orthogonality among the columns of $\hat{U}_{k+1}$ and the level of orthogonality among the columns of $\hat{V}_k$ should be roughly comparable to each other. We illustrate this using a numerical example.

EXAMPLE 4.1. In this example we apply the recurrence (2.2) to a test matrices from SVDPACK to illustrate the relation between the levels of orthogonality among columns of $\hat{U}_{k+1}$ and $\hat{V}_k$. No reorthogonalization is carried out. The initial vector b is

always chosen to be a vector of all ones. The matrix we used is a term-document matrix from an information retrieval application by Apple Computer Inc. [22]. It is sparse and of dimension 3206×44. Its singular values are plotted in Figure 5.1 in section 5. We first apply the Lanczos bidiagonalization process to A^T. For $k = 2, 3, \ldots, 11$, we tabulate the four quantities $\eta(\hat{U}_k), \eta(\hat{V}_k), \mathrm{cond}_2(\hat{B}_k), \mathrm{cond}_2(\hat{B}_{k+1}(:, 1:\mathtt{k}))$ as follows.

k	$\eta(\hat{U}_k)$	$\eta(\hat{V}_k)$	$\mathrm{cond}_2(\hat{B}_k)$	$\mathrm{cond}_2(\hat{B}_{k+1}(:, 1:\mathtt{k}))$
2	2.3052e-14	4.0422e-14	1.5941e+00	1.5349e+00
3	1.0141e-13	1.4936e-13	1.8614e+00	1.7463e+00
4	3.3635e-13	4.8692e-13	2.1773e+00	2.0022e+00
5	9.6149e-13	1.5292e-12	2.6264e+00	2.4151e+00
6	4.2373e-12	8.0257e-12	2.9814e+00	2.6638e+00
7	1.7977e-11	3.5758e-11	3.6939e+00	2.9626e+00
8	8.1124e-11	1.3235e-10	4.2295e+00	3.7537e+00
9	3.5596e-10	5.9628e-10	4.3911e+00	4.2686e+00
10	2.0151e-09	3.4583e-09	4.4231e+00	4.3872e+00
11	9.6713e-09	1.5937e-08	4.4329e+00	4.4189e+00

For this example after about 20 iterations of the Lanczos bidiagonalization process, the orthogonality among $\{\hat{U}_k\}$ and $\{\hat{V}_k\}$ are completely lost. We notice that both $\hat{B}_k$ and $\hat{B}_{k+1}(:, 1:\mathtt{k})$ are well-conditioned, and therefore $\eta(\hat{U}_{k+1})$ and $\eta(\hat{V}_k)$ are comparable to each other. Next we apply the Lanczos bidiagonalization process to A itself and again b is a vector of all ones. This time since an extra zero arising from $m \neq n$ is being approximated by a singular value of $\hat{B}_k$, the matrix $\hat{B}_k$ becomes more and more ill-conditioned as k increases. However, $\hat{B}_{k+1}(:, 1:\mathtt{k})$ does not become ill-conditioned, and therefore $\eta(\hat{U}_{k+1})$ and $\eta(\hat{V}_k)$ are still comparable to each other.

k	$\eta(\hat{U}_k)$	$\eta(\hat{V}_k)$	$\mathrm{cond}_2(\hat{B}_k)$	$\mathrm{cond}_2(\hat{B}_{k+1}(:, 1:\mathtt{k}))$
2	5.3798e-14	2.0851e-15	5.9274e+00	1.6027e+00
3	5.4055e-14	1.9953e-14	2.3701e+01	1.7965e+00
4	6.1741e-14	6.4649e-14	5.9076e+01	2.0469e+00
5	1.0555e-13	2.0562e-13	1.5571e+02	2.3917e+00
6	3.5843e-13	9.7851e-13	3.3009e+02	2.7807e+00
7	1.7802e-12	3.7335e-12	5.2861e+02	3.6361e+00
8	7.3623e-12	1.8075e-11	7.5720e+02	4.2095e+00
9	3.1936e-11	8.6667e-11	1.2715e+03	4.4092e+00
10	1.4617e-10	5.3847e-10	3.2588e+03	4.4438e+00
11	9.4875e-10	2.6974e-09	7.3327e+03	4.4517e+00
12	3.9498e-09	1.0662e-08	2.0959e+04	4.4539e+00
13	1.9679e-08	6.9862e-08	5.1566e+04	4.4549e+00
14	1.4828e-07	4.2244e-07	9.0752e+04	4.4558e+00
15	8.3146e-07	2.3219e-06	1.5435e+05	4.4565e+00
16	4.1817e-06	1.9093e-05	4.3127e+05	4.4567e+00

We have also tested several other matrices. In summary, if no reorthogonalization is performed in the Lanczos bidiagonalization process, then either $\eta(\hat{U}_{k+1}) \approx \mathrm{cond}(\hat{B}_{k+1}(:, 1:\mathtt{k}))\eta(\hat{V}_k)$ or $\eta(\hat{V}_k) \approx \mathrm{cond}(\hat{B}_k)\eta(\hat{U}_{k+1})$ tends to hold.

We mentioned in section 3 that we need to keep certain level of orthogonality among the computed Lanczos vectors $\{\hat{U}_k\}$ and $\{\hat{V}_k\}$ in order to obtain a good low-rank approximation $\hat{J}_k \equiv \hat{U}_k \hat{\Sigma}_k \hat{V}_k^T$. As is in Lanczos tridiagonalization process, maintaining orthogonality of both $\{\hat{U}_k\}$ and $\{\hat{V}_k\}$ to full machine precision is not necessary. More efficient reorthogonalization schemes such as selective reorthogonalization and partial reorthogonalization have been proposed in the past for Lanczos tridiagonalization process [8, 16, 18]. Proposition 4.1 quantifies the relation between levels of orthogonality of the left and right Lanczos vectors generated by a Lanczos

process without reorthogonalization, and the reorthogonalization scheme we will propose is motivated by this relation. Another motivation for our reorthogonalization scheme comes from the observation that in some applications such as compression of multiple-spectral and hyper-spectral image cubes [4], principal component analysis for face recognition and image databases [24, 13, 23, 17], each column of the matrix A represents a single image acquired at a specific wavelength (channel) or a facial image of a particular individual: the columns of the two-dimensional (2-D) image array is stacked into a single long vector.[2] For a 512×512 2-D image, the row dimension of the resulting matrix A is 262144,[3] while the column dimension is the number of available wavelengths (channels) or the number of face images used in the image databases. In early remote sensing satellite facilities, such as Landsat thematic mapper, the number of channels is 7; now channels are numbered in the several hundreds up to 1024. The number of face images used in an image database ranges from several hundred to several thousand [24]. Therefore in those applications the matrix A is very skinny, i.e., $m \gg n$. To facilitate the discussion, we make the following definition.

> For a rectangular matrix $A \in \mathcal{R}^{m \times n}$ that is very skinny, i.e., $m \gg n$, those Lanczos vectors with dimension n are said to belong to the *short* space while those with dimension m are said to belong to the *long* space.

Recall that during the Lanczos bidiagonalization process, the left and right Lanczos vectors need to be saved so that later they can be used in the reorthogonalization process. If the dimensions of the matrix A are large, then those Lanczos vectors may have to be stored out of core in secondary storage and later be brought into main memory when reorthogonalization is carried out. The most popular secondary storage is hard disk, and disk access is always slow. With an eye toward parallel implementation of the Lanczos bidiagonalization process on distributed memory machines, sophisticated parallel I/O techniques are needed to handle the storage of the Lanczos vectors. This issue is especially relevant in the applications we just mentioned, since the row dimension of A is very large. Great efficiency can be gained if we exclusively perform reorthogonalization in the short space since those vectors have much smaller dimension and can therefore be stored in the main memory during the entire Lanczos bidiagonalization process. Disk access is now limited to saving the currently generated long Lanczos vector to secondary storage, and there is no need to retrieve those previous long Lanczos vectors to perform the reorthogonalization process.

Now we proceed to describe the algorithm with one-sided reorthogonalization for $A \in \mathcal{R}^{m \times n}$ with $m \gg n$. At each step of the Lanczos bidiagonalization process, we orthogonalize $\hat{v}_{i+1}$ against all the previous Lanczos vectors and leave $\hat{u}_{i+1}$ unchanged. In the following we list the pseudocode of the one-sided reorthogonalization.

> ALGORITHM ONE-SIDED.
> $b \neq 0$, a given vector.
> $\hat{\beta}_1 = \|b\|$, $\hat{u}_1 = b/\hat{\beta}_1$, $\hat{\alpha}_1 = \|A^T \hat{u}_1\|$, $\hat{v}_1 = A^T \hat{u}_1/\hat{\alpha}_1$.
> For $i = 1, 2, 3, \ldots$
> Compute
>
> $$\hat{r}_{i+1} = A\hat{v}_i - \hat{\alpha}_i \hat{u}_i, \quad \hat{\beta}_{i+1} = \|\hat{r}_{i+1}\|, \quad \hat{u}_{i+1} = \hat{r}_{i+1}/\hat{\beta}_{i+1},$$
>
> $$\hat{p}_{k+1} = A^T \hat{u}_{k+1} - \hat{\beta}_{k+1} \hat{v}_k.$$

[2] In latent semantic indexing approach to information retrieval, the term-document matrices can also either be very skinny or very fat, i.e., with many more terms than documents or vice versa.

[3] High resolution remote sensing facility can produce 2-D images of dimension 3000×3000.

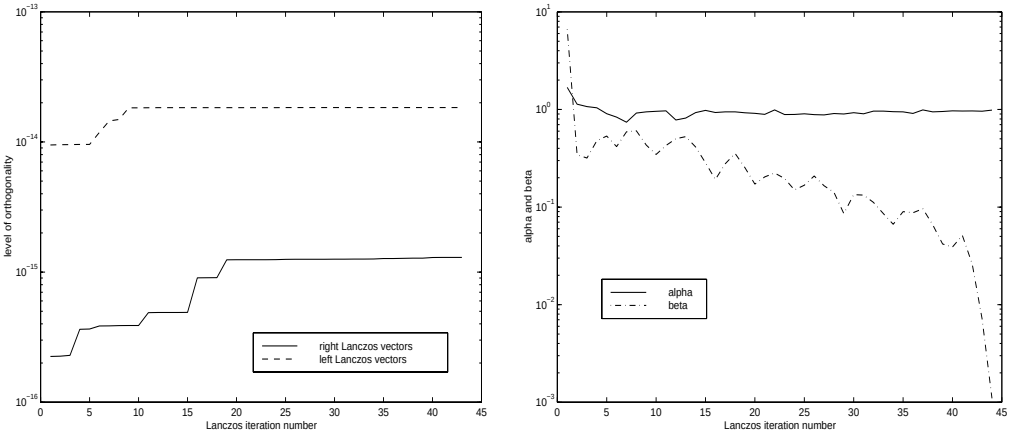

FIG. 4.1. (*Left*) *level of orthogonality,* (*right*) α_k *and* β_k.

Orthogonalize $\hat{p}_{i+1}$ against $\hat{V}_i$ to obtain $\tilde{p}_{k+1}$, and compute

$$\hat{\alpha}_{k+1} = \|\tilde{p}_{k+1}\|, \quad \hat{v}_{k+1} = \tilde{p}_{k+1}/\hat{\alpha}_{k+1}.$$

EXAMPLE 4.2. We look at levels of orthogonality of $\{\hat{U}_k\}$ and $\{\hat{V}_k\}$ computed by Algorithm One-sided. We always perform reorthogonalization in the *short* space, i.e., those Lanczos vectors that have smaller dimension. The matrix is the 3206×44 matrix from SVDPACK which is also used in Example 4.1. We first apply Algorithm One-sided to A^T. In Figure 4.1 on the left we plot $\eta(\hat{U}_k)$ and $\eta(\hat{V}_k)$ for $k = 2, \ldots, 44$, and on the right we plot the two sequences $\{\hat{\alpha}_k\}$ and $\{\hat{\beta}_k\}$. Notice that $\hat{\beta}_k$ drops sharply toward the end of the Lanczos run, but this does not affect the level of orthogonality of either $\{\hat{U}_k\}$ or $\{\hat{V}_k\}$. The condition numbers for both $\hat{B}_k$ and $\hat{B}_{k+1}(:, 1:k)$ are of order $O(1)$. The orthogonality in the long space is very well controlled by enforcing the orthogonality in the short space. We also apply Algorithm One-sided to A itself and reorthogonalize again in the short space. Now we have $\eta(\hat{U}_k) \approx 10^{-14}$ and $\eta(\hat{V}_k) \approx 10^{-15}$. We noticed that one singular value of $\hat{B}_k$ tracks a spurious zero singular value resulting in increasingly larger $\mathrm{cond}(\hat{B}_k)$ but $\mathrm{cond}(\hat{B}_{k+1}(:, 1:k))$ stays $O(1)$. Again the level of orthogonality of the long space is well controlled by that of the short space.

The observation that full reorthogonalization in the short space can, to certain extent, control the level of orthogonality in the long space can not be directly explained by Proposition 4.1 since we need to take into account of the effects of reorthogonalization. We did some preliminary analysis, but the results seem to be dependent on certain intermediate quantities arising from the reorthogonalization process. Now instead of quantifying the relation of levels of orthogonality of the left and right Lanczos vectors in the presence of reorthogonalization, we explain why we still can obtain good low-rank approximation even if the level of orthogonality in the long space is poor. Assuming that $A \in \mathcal{R}^{m \times n}$ with $m \geq n$, and the fact that in the recurrence (2.3) we perform reorthogonalization in the short space as follows, first we compute

$$\hat{p}_{i+1} = A^T \hat{u}_{i+1} - \hat{\beta}_{i+1} \hat{v}_i - g_{i+1},$$

and then we orthogonalize $\hat{p}_{i+1}$ against all the previous vectors $\hat{v}_1, \ldots, \hat{v}_i$ to obtain

$$\tilde{p}_{i+1} = \hat{p}_{i+1} - \sum_{j=1}^{i} (\hat{p}_{i+1}^T \hat{v}_j)\hat{v}_j - \tilde{g}_{i+1},$$

and

$$\hat{\alpha}_{i+1} = \|\tilde{p}_{i+1}\|, \quad \hat{v}_{i+1} = \tilde{p}_{i+1}/\hat{\alpha}_{i+1}.$$

Combining the above equations we obtain

$$\hat{\alpha}_{i+1}\hat{v}_{i+1} = A^T\hat{u}_{i+1} - \hat{\beta}_{i+1}\hat{v}_i - \hat{g}_{i+1},$$

with $\hat{g}_{i+1} = g_{i+1} + \tilde{g}_{i+1} + \sum_{j=1}^{i}(\hat{p}_{i+1}^T\hat{v}_j)\hat{v}_j$. In compact matrix form we have

$$A^T\hat{U}_{k+1} = \hat{V}_{k+1}\hat{B}_{k+1} + \hat{G}_{k+1}, \quad \hat{G}_{k+1} = [\hat{g}_1, \ldots, \hat{g}_{k+1}].$$

$\hat{G}_{k+1}$ involves terms such as $\hat{p}_{i+1}^T\hat{v}_j$ and is difficult to bound, and in general there is no guarantee that $\|\hat{G}_{k+1}\| = O(\|A\|_F\epsilon_M)$ as would be the case if no reorthogonalization is performed. However, notice that the other half of the recurrence in (2.4) still has the form

$$A\hat{V}_k = \hat{U}_{k+1}\hat{B}_{k+1}(:, 1:k) + F_k,$$

with $\|F_k\| = O(\|A\|_F\epsilon_M)$. It follows from the above equation that

$$\|A - \hat{U}_k\hat{B}_{k+1}(:, 1:k)\hat{V}_k^T\|_F = \|A(I - \hat{V}_k\hat{V}_k^T)\|_F + O(\|A\|_F\epsilon_M).$$

If columns of $\hat{V}_k$ are orthonormal to each other with high precision (notice that $\hat{v}_{i+1}$ is explicitly orthogonalized against $\hat{V}_i$ for $i = 1, \ldots, k$.), then $\hat{U}_k\hat{B}_{k+1}(:, 1:k)\hat{V}_k^T$ will be a good approximation of A_k as long as $\hat{V}_k$ is a good approximation of the first k right singular vectors of A (cf. section 3). The above statement is true regardless of the level of orthogonality of $\hat{U}_{k+1}$.

5. Numerical experiments. In this section we will use test matrices from several applications fields to demonstrate the accuracy of the low-rank approximation computed by Algorithm One-sided. Before we present the results, we want to say a few words about the efficiency of the algorithm. One contribution of this paper is the introduction of the idea of using $\hat{J}_k = \hat{U}_k\hat{B}_k\hat{V}_k^T$ as a low-rank approximation of a given matrix A. Compared with the approach where SVD of $\hat{B}_k$ is computed and its left and right singular vectors are combined with the left and right Lanczos vectors to give the left and right singular vectors of A, the savings in flop counts is approximately $24k^3 + 4mk^2 + 4nk^2$, where we have assume that $A \in \mathcal{R}^{m\times n}$ and the SVD of $\hat{B}_k$ is computed. How much of the above savings accounts for the total CPU time depends on the number of Lanczos steps k, the matrix A (e.g., its sparsity or structure and its singular value distribution) and the underlying computer architectures used (both for sequential and parallel computers). Notice that the part of computation for the SVD of $\hat{B}_k$ and the combination of the singular vectors and Lanczos vectors have to be done after the Lanczos bidiagonalization process. In [1] the computation of the SVD of $\hat{B}_k$ along on a Cray-2S accounts for 12 to 34% of the total CPU time for a 5831×1033 matrix with $k = 100$, depending on whether A^TA or the 2-cyclic matrix `[0 A; A' 0]` is used. As a concrete example, we show various timings for a term-document matrix of size 4322×11429 with 224918 nonzeros generated from the document collection

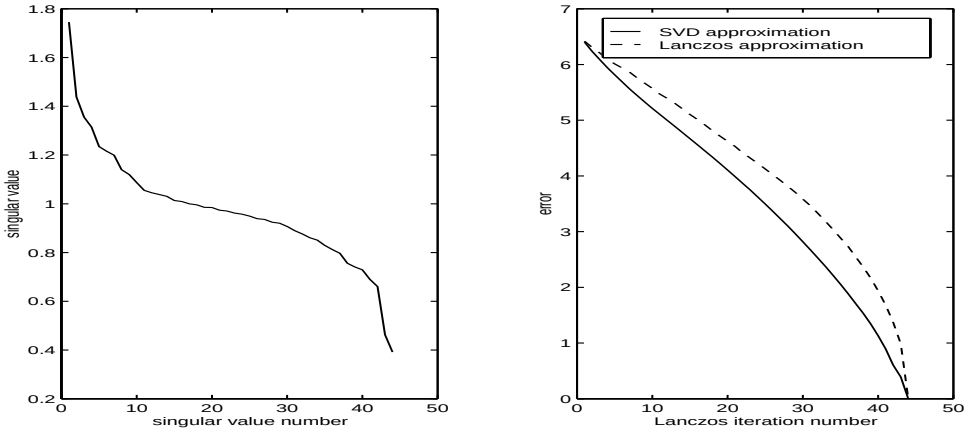

FIG. 5.1. *Plots for* `apple1.mat`.

NPL [2]. Let the SVD of $\hat{B}_k$ be $\hat{B}_k = U_k^B \Sigma_k^B (V_k^B)^T$. Then $\hat{U}_k U_k^B$ and $\hat{V}_k V_k^B$ give the left and right singular vectors of A, respectively. Here $k = 350$, and the CPU time is in seconds.

	Au	$A^T v$	$\hat{U}_k U_k^B$	$\hat{V}_k V_k^B$
Flops	449836	449836	1.06e9	2.80e9
CPU time	0.1650	0.0780	41.4800	110.2600

We should also mention that the potential savings in computational time should be weighed against the possible deterioration in the quality of the low-rank approximation. Fortunately, for most of the applications this is not a problem. Another contribution of the paper is the use of one-sided reorthogonalization technique. The major gain in efficiency from this technique is the reduction in disk access time when the Lanczos vectors have to be stored out of core and later on be brought back in for reorthogonalization. This part of the saving depends heavily on the underlying computer architectures used and is not easy to quantify.

We have tested three classes of matrices and compared the low-rank approximations computed by Algorithm One-sided with those computed by the SVD:

- Large sparse test matrices from SVDPACK [22] and document collections [2].
- Several general rectangular matrix from Matrix Market [11].
- Three-dimensional (3-D) image cubes from remote sensing applications.

All the computation is done using MATLAB Version 5 on a Sun server 2000. For each test matrix we first plot the singular values of the matrix and then the two sequences $\{\|A - \hat{U}_k \hat{B}_k \hat{V}_k^T\|_F\}$ and $\{(\sum_{j=k+1}^{\min(m,n)} \sigma_j^2)^{1/2}\}$. We run Algorithm One-sided for $\min(m,n)$ iterations just to test the algorithm, since in practice the algorithm will be stopped when a user supplied tolerance is satisfied or the maximum number of iterations has been reached, and usually the number of iterative steps will be much less than $\min(m,n)$. If the range of the quantities to be plotted is too large, we will plot them in log-scale. We also compute

$$\texttt{ratio}_k = \frac{\left(\sum_{j=k+1}^{\min(m,n)} \sigma_j^2\right)^{1/2}}{\|A - \hat{U}_k \hat{B}_k \hat{V}_k^T\|_F}. \tag{5.1}$$

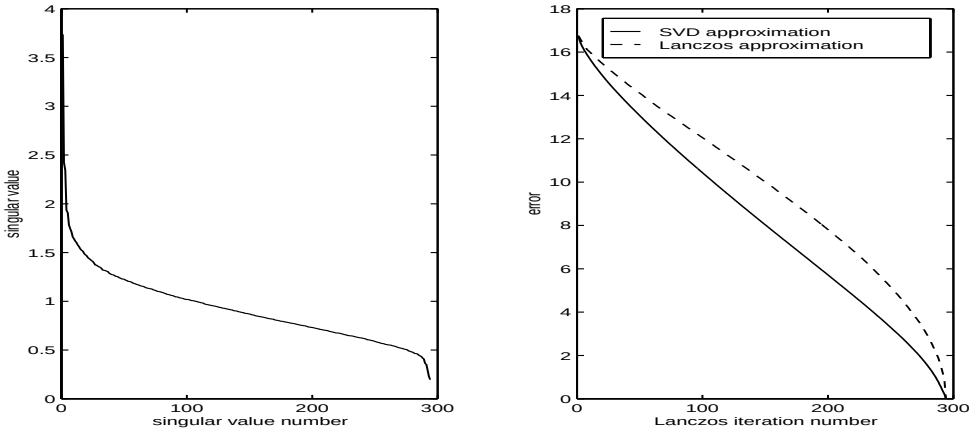

FIG. 5.2. *Plots for* `apple2.mat`.

EXAMPLE 5.1. Three test matrices are included in SVDPACK [22]. All of them are in Harwell–Boeing format. We used a utility routine that converts a Harwell–Boeing format to MATLAB's `.mat` format. A brief description of the three matrices is given in the following:

- `apple1.mat`, a 3206×44 term-document matrix from an information retrieval application by Apple Computer Inc.
- `apple2.mat`, a 1472×294 term-document matrix from an information retrieval application by Apple Computer Inc.
- `amoco.mat`, a 1436×330 Jacobian matrix from a seismic tomography application by Amoco Research Inc.

For the three test matrices in this example, Algorithm One-sided is applied with reorthogonalization in the short space. For both `apple1.mat` and `apple2.mat`, one-sided reorthogonalization controls the level of orthogonality very well, and the level of orthogonality for both $\{\hat{U}_k\}$ and $\{\hat{V}_k\}$ is around 10^{-14}. (See Figures 5.1 and 5.2.) For `amoco.mat`, the level of orthogonality for the long space deteriorates from 10^{-14} to 10^{-12} at the end of 330 steps. (See Figure 5.3.) In the following table we list both the maximum and minimum of the ratio $\{\texttt{ratio}_k\}$ defined in (5.1) for the three matrices. It is also interesting to notice that even though there is difference between $\|A - A_k\|_F$ and $\|A - \hat{J}_k\|_F$ for a fixed k, it is always possible to move forward a few steps s to get a $\hat{J}_{k+s}$ such that $\|A - \hat{J}_{k+s}\|_F \approx \|A - A_k\|_F$. For these three test matrices we can chose s to be rather small, say $s \leq 3$, especially in the initial several iterations of the Lanczos run. The following table lists $\max(\texttt{ratio}_k)$ and $\min(\texttt{ratio}_k)$ for these three matrices.

	`apple1.mat`	`apple2.mat`	`amoco.mat`
$\max(\texttt{ratio}_k)$	9.9741e-01	9.9820e-01	9.8656e-01
$\min(\texttt{ratio}_k)$	3.9749e-01	2.5214e-01	9.1414e-02

EXAMPLE 5.2. To illustrate the effectiveness of using $\hat{J}_k$ as a low-rank approximation of A for LSI information retrieval applications, we compare it with A_k obtained from the partial SVD of A, i.e., $A_k \equiv P_k \operatorname{diag}(\sigma_1, \ldots, \sigma_k) Q_k^T$, where P_k and Q_k are the matrices formed by the first k columns of P and Q, respectively. In this example we tested two data collections: a 3681×1033 term-document matrix from data

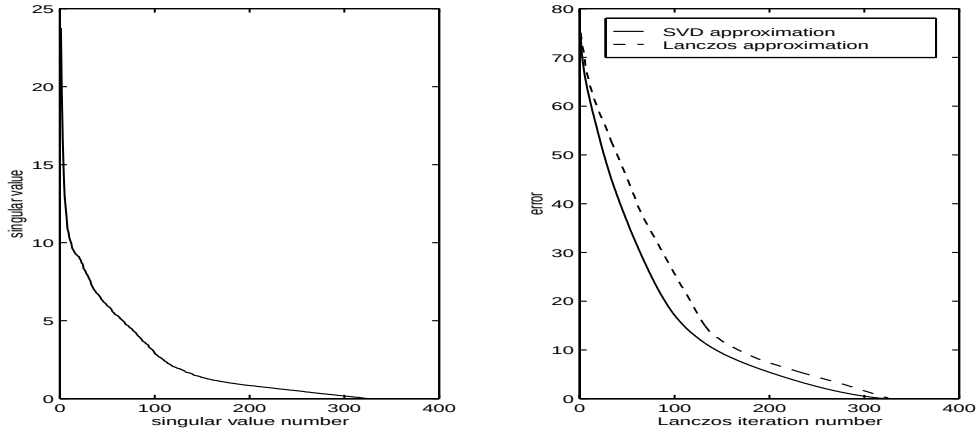

FIG. 5.3. *Plots for* `amoco.mat`.

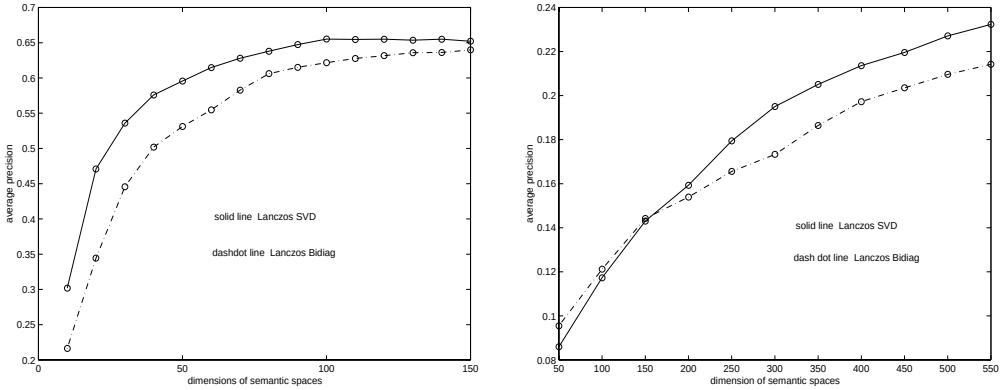

FIG. 5.4. *Comparison of average precisions.*

collection consisting of abstracts in biomedicine with 30 queries, and a 4322×11529 term-document matrix from the NPL collection with 93 queries [2]. We used 11-point average precision, a standard measure for comparing information retrieval systems [10]. For $k = 10, 20, \ldots, 150$, two sequences are plotted in the left of Figure 5.4 for the first matrix, one for J_k and one for A_k. $k = 110$ is chosen for the reduced dimension, and the precision for A_{110} and J_{110} are 65.50% and 63.21%, respectively. The right plot is for the second matrix with $k = 50, 100, \ldots, 550$. Judging from the curve, for this particular matrix we probably should have used larger k, but we are limited by computation time. The precision for A_{550} and J_{550} are 23.23% and 21.42%, respectively.

EXAMPLE 5.3. Matrix Market contains several general rectangular matrices. Of special interests to us is the set `LSQ` which comes from linear least squares problems in surveying [11]. This set contains four matrices, and all of them are in Harwell–Boeing format. We first convert them into MATLAB's `.mat` format. The matrix `illc1033.mat` is of dimension 1033×320; it is an interesting matrix because it has several clusters of singular values which are very close to each other. For example, the

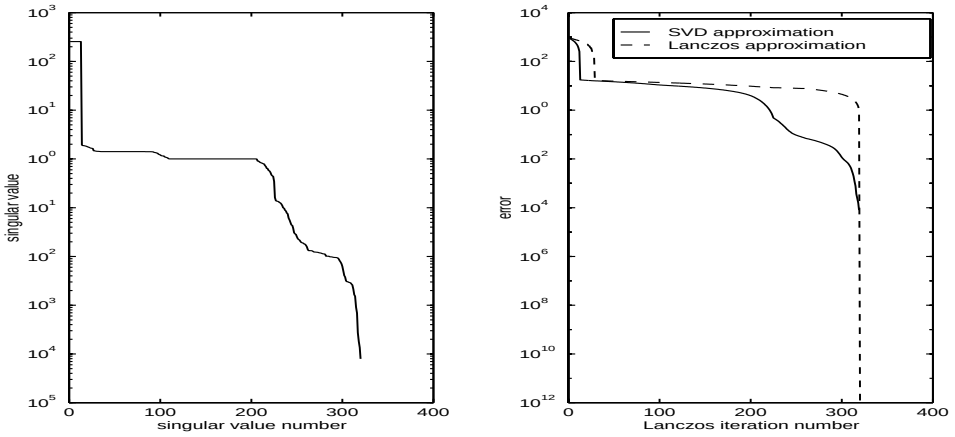

FIG. 5.5. *Plots for* `illc1033.mat`.

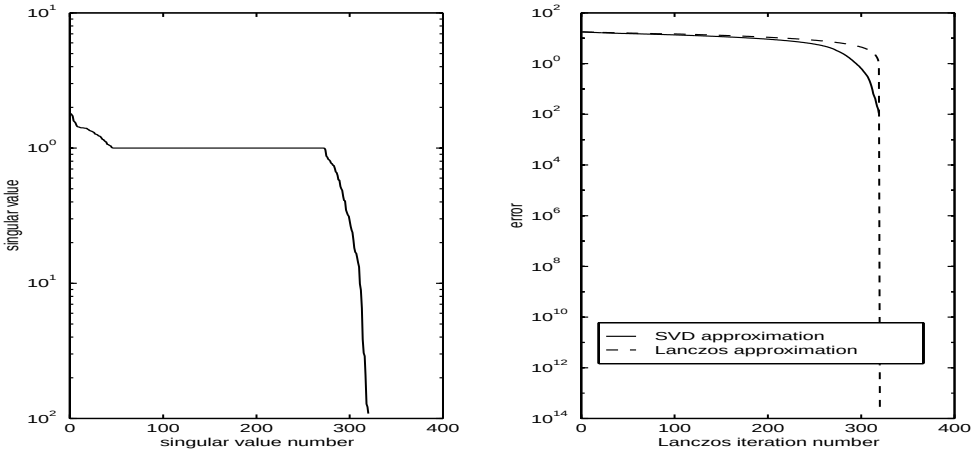

FIG. 5.6. *Plots for* `well1033.mat`.

first cluster contains the first 13 singular values ranging from `2.550026352702592e+02` to `2.550020307989260e+02` and another cluster contains σ_{113} to σ_{205} ranging from `1.000021776237986e+00` to `9.999997517165140e-01`. This clustering actually exposes one weakness of using $\hat{J}_k = \hat{U}_k \hat{B}_k \hat{V}_k^T$ as approximations of A. It is well known that single-vector Lanczos algorithm can compute multiple eigenvalues of a symmetric matrix, but the multiple eigenvalues do not necessarily converge *consecutively* one after the other. To be precise, say $\lambda_{\max}(H)$ is a multiple eigenvalue of a symmetric matrix H. Then usually a copy of $\lambda_{\max}(H)$ will converge first, followed by several other smaller eigenvalues of H, then another copy of $\lambda_{\max}(H)$ will converge, followed by still several other smaller eigenvalues, and so on. The consequence of this convergence pattern to our task of computing low-rank approximation of a rectangular matrix A is that in the first few steps with $k < l$, l the multiplicity of $\sigma_{\max}(A)$, $\hat{J}_k$ will contain fewer than k copies of $\sigma_{\max}$. Therefore J_k will not be a good approximation of A as compared with A_k if $\sigma_{max}(A)$ is much larger than the next singular value.

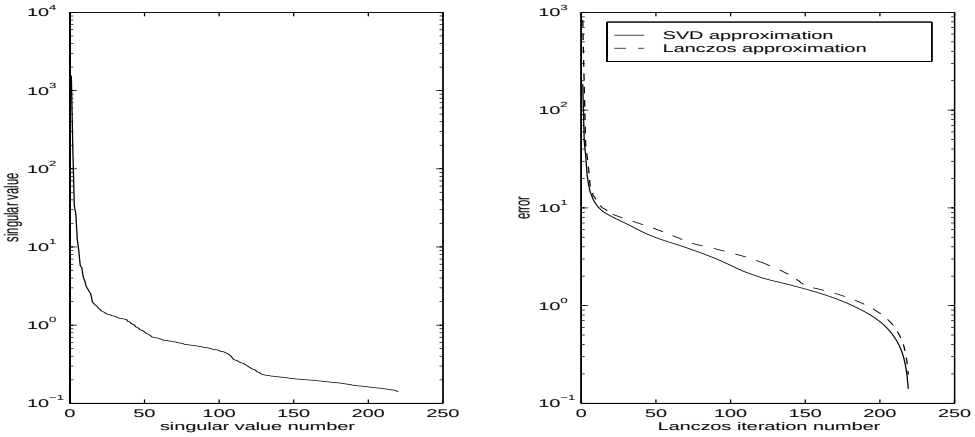

FIG. 5.7. *Plots for* `92AV3C.mat`.

This is why in the right plot of Figure 5.5, the curve for the Lanczos approximation lags behind that of the SVD approximation in the initial several iterations. We also noticed that for `illc1033.mat` the level of orthogonality changes from 10^{-14} to 10^{-10} while for `well1033.mat` it changes from 10^{-14} to 10^{-13}. (See Figure 5.6.)

EXAMPLE 5.4. This test matrix is obtained by converting a 220-band image cube taken from the homepage of MULTISPEC, a software package for analyzing multispectral and hyperspectral image data developed at Purdue University [12]. The data values are proportional to radiance units. The number 1000 was added to the data so that there were no negative data values. (Negative data values could occur in the water absorption bands where the signal was very low and noisy.) The data was recorded as 12-bit data and was collected near West Lafayette, IL with the AVIRIS system, which is operated by NASA JPL and AMES.[4] Each of the 2-D images is of dimension 145×145, and therefore the resulting matrix A is of dimension 21025×220. We applied Algorithm One-sided to A^T with the starting b a vector of all ones. The left of Figure 5.7 plots the singular values of A, and we can see there are only very few dominant singular values and all the others are relative small. The reason for this is that the 2-D images in the image cube are for the same scene acquired at different wavelengths and therefore there is very high correlation among them. In fact the largest singular value of A accounts for about 88% of $\|A\|_F$, the first three largest singular values account for about 98%, and the first five largest singular values account for more than 99%. As a comparison, for a 2-D image matrix of dimension 837×640 it takes the first 23 largest singular values to account for 88% of $\|A\|_F$, the first 261 largest singular values to account for 98%, and the first 347 largest singular values to account for 99%. We also notice that J_k gives very good approximation of A_k, and $\max(\texttt{ratio}_k) = \texttt{9.3841e}-\texttt{01}$ and $\min(\texttt{ratio}_k) = \texttt{2.2091e}-\texttt{01}$ in the first 50 iterations.

6. Concluding remarks. Low-rank matrix approximation of large and/or sparse matrices plays an important role in many applications. We showed that good low-rank

[4]Larry Biehl of Purdue University provided several MATLAB m-files for reading multiple-spectral images in BIL format with an ERDAS74 header into MATLAB. He also provided the description of the data set.

matrix approximations can be obtained directly from the Lanczos bidiagonalization process. We discussed several theoretical and practical issues such as a priori error estimation, recursive computation of stopping criterion, and relations between levels of orthogonality of the left and right Lanczos vectors. We also proposed an efficient reorthogonalization scheme: one-sided reorthogonalization. A collection of test matrices from several applications areas were used to illustrate the accuracy and efficiency of Lanczos bidiagonalization process with one-sided reorthogonalization. There are several issues that we think deserves further investigation, specifically it is of great interest to develop a theory that can quantify the relation between the levels of orthogonality of the left and right Lanczos vectors *in the presence* of reorthogonalization.

Acknowledgments. The authors thank one of the referees for many insightful suggestions and comments which greatly clarify many subtle points and improve the presentation of this paper. The authors also thank Sherry Li and John Wu of NERSC, Lawrence Berkeley National Laboratory, for many helpful discussions and assistance.

REFERENCES

[1] M. Berry, *Large scale singular value computations*, Internat. J. Supercomputer Applications, 6 (1992), pp. 13–49.
[2] *Cornell SMART System*, ftp://ftp.cs.cornell.edu/pub/smart.
[3] J. Cullum, R. A. Willoughby, and M. Lake, *A Lanczos algorithm for computing singular values and vectors of large matrices*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 197–215.
[4] P. Geladi and H. Grahn, *Multivariate Image Analysis*, John Wiley, New York, 1996.
[5] G. Golub and W. Kahan, *Calculating the singular values and pseudo-inverse of a matrix*, SIAM J. Numer. Anal., 2 (1965), pp. 205–224.
[6] G. H. Golub, F. Luk, and M. Overton, *A block Lanczos method for computing the singular values and corresponding singular vectors of a matrix*, ACM Trans. Math. Software, 7 (1981), pp. 149–169.
[7] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.
[8] J. Grcar, *Analysis of the Lanczos Algorithm and of Approximation Problem in Richardson's Method*, Ph.D. thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana-Champaign, IL, 1981.
[9] P. C. Hansen, *Truncated singular value decomposition solutions to discrete ill-posed problems with ill-determined numerical rank*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 503–518.
[10] D. Harman, *TREC-3 Conference Report*, NIST Special Publication 500-225, 1995.
[11] *Matrix Market*, http://math.nist.gov/MatrixMarket/.
[12] *MultiSpec*, http://dynamo.ecn.purdue.edu/~biehl/MultiSpec/documentation.html.
[13] A. O'Toole, H. Abdi, K. A. Deffenbacher, and D. Valentin, *Low-dimensional representation of faces in higher dimensions of the face space*, J. Amer. Optical Soc., 10 (1993), pp. 405–411.
[14] C. C. Paige, *Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix*, J. Inst. Math. Appl., 18 (1976), pp. 341–349.
[15] 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.
[16] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.
[17] A. Petland, R. W. Picard, and S. Sclaroff, *Photobook: Content-based manipulation of image databases*, Internat. J. Comput. Vision, 18 (1996), pp. 233–254.
[18] H. D. Simon, *The Lanczos Algorithm for Solving Symmetric Linear Systems*, Ph.D. dissertation, Department of Mathematics, University of California, Berkeley, CA, 1982.
[19] H. D. Simon, *Analysis for the symmetric Lanczos algorithm with reorthogonalization*, Linear Algebra Appl., 61 (1984), pp. 101–131.
[20] H. D. Simon, *The Lanczos algorithm with partial reorthogonalization*, Math. Comp., 42 (1984), pp. 115–142.

[21] H. D. Simon and H. Zha, *Low Rank Matrix Approximation Using the Lanczos Bidiagonalization Process with Applications*, Technical report CSE-97-008, Department of Computer Science and Engineering, The Pennsylvania State University, State College, PA, 1997.
[22] SVDPACK, http://www.netlib.org/svdpack/index.html.
[23] N. E. Troje and T. Vetter, *Representation of human faces*, Technical report 41, Max-Planck-Institute für biologische Kybernetik, Tübingen, Germany, 1996.
[24] M. Turk and A. Pentland, *Eigenfaces for recognition*, J. Cognitive Neuroscience, 3 (1991), pp. 71–86.