LBNL-40767
UC-405
Preprint

ERNEST ORLANDO LAWRENCE
BERKELEY NATIONAL LABORATORY

Low Rank Matrix Approximation Using the Lanczos Bidiagonalization Process

Horst D. Simon and Hongyuan Zha
Computing Sciences Directorate

July 1997

Submitted to
SIAM Journal of Scientific Computing

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not neccessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

LBNL-40767
UC-405

Low Rank Matrix Approximation Using the Lanczos Bidiagonalization Process

Horst D. Simon and Hongyuan Zha

Computing Sciences Directorate
Ernest Orlando Lawrence Berkeley National Laboratory
University of California
Berkeley, California 94720

July 1997

This work was supported by the Office of Energy Research, Office of Laboratory Policy and Infrastructure Management, and the Office of Computational and Technology Research, Mathematical, Information and Computational Sciences Division, of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098.

LOW RANK MATRIX APPROXIMATION USING THE LANCZOS BIDIAGONALIZATION PROCESS

HORST D. SIMON [†] AND HONGYUAN ZHA [‡]

Abstract. Low rank approximation of large and/or sparse matrices is important in many applications. We show that good low rank matrix approximations can be directly obtained from the Lanczos bidiagonalization process without computing singular value decomposition. We also demonstrate that a so-called one-sided reorthogonalization process can be used to maintain 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.

1. Introduction.

In many applications such as compression of single images and multiple-spectral image cubes, regularization methods for ill-posed problems, latent semantic indexing, to name a few, it is necessary to find a low rank approximation of a given matrix $A \in \mathcal{R}^{m\times n}$. Often A is a general rectangular matrix and sometimes either $m \gg n$ or $m \ll n$. The theory of singular value decomposition (SVD) provides the following characterization of the best rank-j approximation of A in terms of the Frobenius norm $\|\cdot\|_F$ [6].

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

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

And the minimum is achieved with $A_j \equiv P_j\,\mathrm{diag}(\sigma_1,\ldots,\sigma_j)Q_j^T$, 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 only interested in some A_j with $j \ll \min(m,n)$, the computation of the SVD of A is rather 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 one possible avenue of applying the 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 [3, 2]. We will show that in many cases of interest good approximations can be directly obtained from the Lanczos bidiagonalization process without computing any singular value decomposition. We will also explore relations between the levels of orthogonality of the left Lanczos vectors and the right Lanczos vectors and propose some more efficient reorthogonalization schemes 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 briefly review the Lanczos bidiagonalization process and its several variations in finite precision arithmetic. In Section 3 we discuss both *a priori* and *a posteriori* estimation and stopping criteria. Section 4 is devoted to orthogonalization issues in the Lanczos bidiagonalization process and several reorthogonalization schemes are discussed in detail. In Section 5 we perform

[†] NERSC, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720.

[‡] 307 Pond Laboratory, Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16802-6103.

numerical experiments on test matrices from a variety of applications areas. Section 6 concludes the paper and points out some directions for future investigation.

2. The Lanczos bidiagonalization process. Bidiagonalization of a general 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 [13] and to finding a few dominant singular triplets of large sparse matrices [3, 2]. 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 proposed algorithm LSQR [13].[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 those spurious singular values thus generated from the 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 certain level of orthogonality among the Lanczos vectors [14, 17]. 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 [18], 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 *total* reorthogonalization is implemented. In Section 4 we will discuss two other more efficient reorthogonalization schemes using the coupled two term recurrence.

Now we briefly describe the Lanczos bidiagonalization process presented in [5, 13, 3]. Let b be a starting vector, for $i = 1, 2, \ldots$, compute

$$\begin{aligned} \beta_1 u_1 &= b, \quad \alpha_1 v_1 = A^T u_1, \\ \beta_{i+1} u_{i+1} &= A v_i - \alpha_i u_i, \\ \alpha_{i+1} v_{i+1} &= A^T u_{i+1} - \beta_{i+1} v_i. \end{aligned} \tag{2.1}$$

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

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

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 $\tilde{B}_k$ is B_{k+1} with the last column removed.

[1] Maintaining certain level of orthogonality among the Lanczos vectors will accelerate the convergence at the expense of more computational cost and storage requirement [17, Section 4].

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

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

For A with more columns than rows, this version is usually better than (2.2) because the chances of introducing spurious 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. 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.

Following the error analysis in [12], it is straightforward to show that in finite precision arithmetic, Equations (2.1) and (2.2) become

$$\begin{aligned}\beta_1 u_1 &= b, \quad \alpha_1 v_1 = A^T u_1 + g_1,\\ \beta_{i+1} u_{i+1} &= Av_i - \alpha_i u_i - f_i, \qquad i = 1, 2, \ldots\\ \alpha_{i+1} v_{i+1} &= A^T u_{i+1} - \beta_{i+1} v_i - g_{i+1},\end{aligned} \tag{2.3}$$

and in compact matrix form,

$$\begin{aligned}U_{k+1}(\beta_1 e_1) &= b,\\ AV_k &= U_{k+1}\tilde{B}_k + F_k,\\ A^T U_{k+1} &= V_{k+1} B_{k+1}^T + G_{k+1},\end{aligned} \tag{2.4}$$

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, 2]. We will show that if one is only interested in finding a good low rank approximation of A a more direct approach is possible without computing the SVD of B_k: For a given k, we will use $J_k \equiv U_k B_k V_k^T$ as an approximation of A. In the next section we will consider both *a priori* and *a posteriori* 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 [14]. 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 concerning the approximation of an eigenvector of a symmetric matrix from a Krylov subspace [14, Section 12.4].

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

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

Let $C = Z\,\mathrm{diag}(\alpha_i)Z^T$ be the eigendecomposition of C with $\alpha_1 \geq \ldots \geq \alpha_n$ its eigenvalues. Write $Z = [z_1, \ldots, z_n]$ and define $\mathcal{Z}_j = \mathrm{span}\{z_1, \ldots, z_j\}$. Then

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

where $\gamma = (\alpha_j - \alpha_{j+1})/(\alpha_{j+1} - \alpha_n)$.

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

THEOREM 3.2. *Let $A = P\,\mathrm{diag}(\sigma_i)Q^T$ be the SVD of A, and $\mathcal{P}_i \equiv \mathrm{span}\{p_1, \ldots, p_i\}$. Assume the Lanczos bidiagonalization process starts with b as in (2.2). Then for any j with $1 < j < n$ and $k > j$,*

$$\omega_k^2 \leq \sum_{i=j+1}^{n} \sigma_i^2 + \sum_{i=1}^{j} \sigma_i^2 \left(\frac{\sin \angle(b, \mathcal{P}_i)\Pi_{v=1}^{i-1}(\sigma_v^2 - \sigma_n^2)/(\sigma_v^2 - \sigma_i^2)}{\cos \angle(b, p_i)T_{k-i}(1+2\gamma_i)}\right)^2, \tag{3.5}$$

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

Proof. Using the SVD of A 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 can bound

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

It is readily verified that $\mathrm{span}\{U_k\} = \mathrm{span}[b, AA^Tb, \ldots, (AA^T)^{k-1}b] \equiv \mathcal{K}_k$. 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 (3.5), $(\sum_{i=j+1}^{n} \sigma_i^2)^{1/2}$ is the error of the best rank-j approximation of A in Frobenius norm. For $k > j$ usually $\mathrm{rank}(J_k) > j$. 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 very close to $\|A - A_j\|_F^2$.

REMARK. If we assume that the spectrum of AA^T has one cluster $[\sigma_n^2, \sigma_{j+1}^2]$ which is well separated from the rest of the spectrum, a better estimate of $\tan \angle(p_j, \mathcal{K}_m)$ can be obtained using the techniques in [21].

Now we examine the effect of loss of orthogonality among the columns of U_k and V_k has on the accuracy of J_k and give *a posterior* estimate of $\|A - J_k\|_F$. Intuitively, if both U_k and V_k contains spurious approximate singular vectors J_k will not be a good approximation. Unlike in the case of computing the singular values/vectors of A, keeping certain level of orthogonality of $\{U_k\}$ and $\{V_k\}$ is essential to obtain a good J_k. In the following we will make the above statements more precise. To this end we introduce some notation. Let the SVD of B_k be $B_k = T_k\Theta_kS_k^T$ with T_k and S_k orthogonal and $\Theta_k = \mathrm{diag}(\theta_1, \ldots, \theta_k)$. Let $X_k = U_kT_k$ and $Y_k = V_kS_k$. Define $\eta(V_k) = \|I - V_k^TV_k\|$, a measure of the level of orthogonality among $\{V_k\}$. We have for $1 < j < n$,

$$\|A - J_k\|_F = \|A - X_k\Theta_kY_k^T\|_F = \|(A - X_k\Theta_kY_k^T)[Y_j, Y_j^{\perp}]\|_F, \tag{3.6}$$

where $Y_j^{\perp}$ is the orthogonal complement of Y_j. It can be verified that

$$(A - X_k\Theta_k Y_k^T)Y_j = AY_j - X_j \operatorname{diag}(\theta_1, \dots, \theta_j) + O(\|A\|_F \eta(V_k)).$$

It follows from (2.4) that

$$AY_k = X_k\Theta_k + \beta_{k+1}u_{k+1}e_k^T S_k + F_k S_k.$$

Now write $\beta_{k+1}e_k^T S_k \equiv [\beta_{1k}, \dots, \beta_{kk}]$ and take the first j columns of the above equation, we have

$$AY_j = X_j \operatorname{diag}(\theta_1, \dots, \theta_j) + u_{k+1}[\beta_{1k}, \dots, \beta_{jk}] + F_k S_k(:, 1:j).$$

Hence

$$\|(A - X_k\Theta_k Y_k^T)Y_j\|_F = \left(\sum_{i=1}^{j} \beta_{ik}^2\right)^{1/2} + O(\|A\|_F \eta(V_k)) + O(\|A\|_F \epsilon_M).$$

Combining the above and Equation (3.6) and noticing that

$$A - U_k B_k V_k^T = (I - U_k U_k^T)A + U_k F_k^T$$

we obtain the estimate

$$\|A - U_k B_k V_k^T\|_F \leq \left(\sum_{i=1}^{j} \beta_{ik}^2\right)^{1/2} + \|(I - U_k U_k^T)AY_j^{\perp}\|_F + O(\|A\|_F \eta(V_k)) + O(\|A\|_F \epsilon_M).$$

A few words about the above estimate are in order here. For those dominant singular values that have converged, the corresponding β_{ik} will be small [14, Section 13.6]. The quantity $\|(I - U_k U_k^T)AY_j^{\perp}\|_F^2$ will be close to $\|A - A_j\|_f^2 = \sigma_{j+1}^2 + \dots + \sigma_n^2$, and the accuracy of $\|A - U_k B_k V_k^T\|_F$ as compared to $\|A - A_j\|_F$ is limited by the level of orthogonality in V_k.

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 process will be stopped when $\omega_k \leq$ `tol` with `tol` a user supplied tolerance. Therefore it will be very helpful to find an inexpensive way to compute ω_k for $k = 1, 2, \dots$. 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* $\omega_{k+1}^2 = \omega_k^2 - \alpha_{k+1}^2 - \beta_{k+1}^2$.

Proof. From Equation (2.2) we have $U_{k+1}^T A = B_{k+1}V_{k+1}^T$. Hence

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

Now write rite $I - U_k U_k^T = (I - U_{k+1}U_{k+1}^T) + u_{k+1}u_{k+1}^T$. Notice that $(I - U_{k+1}U_{k+1}^T)A$ and $u_{k+1}u_{k+1}^T A$ are orthogonal in the Frobenius norm, we obtain

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

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 arithemtic. 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. Equation 2.4)

$$A^T U_k = V_k B_k^T + G_k,$$

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

$$\omega_k^2 = \|A - U_kB_kV_k^T\|_F^2 = \|(I - U_kU_k^T)A + U_kF_k^T\|_F^2.$$

In finite precision computation, due to the loss of orthogonality the columns of U_k and V_k are not necessarily orthogonal to each other. Define

$$\eta_U = \|I - U_{k+1}^TU_{k+1}\|, \quad \eta_V = \|I - V_{k+1}^TV_{k+1}\|.$$

These two quantities measure the level of orthogonality among the columns of U_{k+1} and V_{k+1}, respectively. We will also need the fact that

$$\|I - U_k^TU_k\| \leq \eta_U, \quad \|I - V_k^TV_k\| \leq \eta_V.$$

THEOREM 3.4. *In finite precision arithmetic the computed ω_k satisfies*

$$\begin{aligned}\omega_k^2 = \;& \omega_{k+1}^2 + \alpha_{k+1}^2 + \beta_{k+1}^2 + O(\|A\|_F^2(\eta_U(1+\eta_U)(1+m\epsilon_M))) + \\ & O(\|A\|_F^2(1+\eta_U)^2\epsilon_M) + O(\|A\|_F^2\eta_V).\end{aligned}$$

Proof. We write $\omega_k^2 = \|(I - U_kU_k^T)A\|_F^2 + \|U_kF_k^T\|_F^2 + \text{term}_1$, where

$$|\text{term}_1| = 2|\text{trace}(A^T(I - U_kU_k^T)U_kF_k^T)| = O(\|A\|_F^2(1+\eta_U)\eta_U\epsilon_M).$$

Now split $(I - U_kU_k^T)A$ as $(I - U_{k+1}U_{k+1}^T)A + u_{k+1}u_{k+1}^TA$ and write

$$\|(I - U_kU_k^T)A\|_F^2 = \|(I - U_{k+1}U_{k+1}^T)A\|_F^2 + \|u_{k+1}u_{k+1}^TA\|_F^2 + \text{term}_2. \tag{3.7}$$

We have the bound

$$|\text{term}_2| = 2|\text{trace}(A^T(I - U_{k+1}U_{k+1}^T)u_{k+1}u_{k+1}^TA)| = O(\|A\|_F^2(1+\eta_U)\eta_U).$$

Since $u_{k+1}u_{k+1}^TA$ is rank-one, we have $\|u_{k+1}u_{k+1}^TA\|_F^2 = \|u_{k+1}\|\|A^Tu_{k+1}\|$. Now it follows from $A^Tu_{k+1} = \alpha_{k+1}v_{k+1} + \beta_{k+1}v_k - g_{k+1}$ (cf. Equation (2.1)) that

$$\begin{aligned}\|A^Tu_{k+1}\| = \;& (\alpha_{k+1}^2 + \beta_{k+1}^2)(1+\epsilon_M) + \|g_{k+1}\|^2 + 2\alpha_{k+1}\beta_{k+1}v_k^Tv_{k+1} + \\ & 2(\alpha_{k+1} + \beta_{k+1})O(\epsilon_M) \\ = \;& \alpha_{k+1}^2 + \beta_{k+1}^2 + O(\|A\|_F^2\eta_V) + O(\|A\|_F\epsilon_M).\end{aligned}$$

Substituting the above estimates into Equation (3.7) completes the proof. □

Therefore within the level of orthogonality of U_{k+1} and V_{k+1}, the formula $\omega_{k+1}^2 = \omega_k^2 - \alpha_{k+1}^2 - \beta_{k+1}^2$ can be used to compute ω_{k+1}.

REMARK. With some extra effort, we can improve the above result which roughly says that $\omega_{k+1}^2 = \omega_k^2 - \alpha_{k+1}^2 - \beta_{k+1}^2 + O(\eta_U) + O(\eta_V)$. We want to improve the error term from $O(\eta_U) + O(\eta_V)$ to $O(\eta_U^2) + O(\eta_U\eta_V)$. To this end, we need to make the following assumptions,[2]

[2] The assumptions $|u_{i+1}^Tu_i| = O(\eta_U^2)$ and $|v_{i+1}^Tv_i| = O(\eta_V^2)$ need an explanation: in the coupled two-term recurrence (2.2) no explicit orthogonalization is performed and therefore it is generally not true that the two consecutive Lanczos vectors u_{i+1} and u_i, v_{i+1} and v_i are orthogonal to each other to working precision. This is in sharp contrast to the case in tridiagonalizing a symmetric matrix [14, Equation (13.4.4)]. However, the assumption is usually not a severe restriction especially when semi-reorthogonality is maintained (cf. Section 4).

1. $1 - u_i^T u_i = O(\epsilon_M)$, $\quad 1 - v_i^T v_i = O(\epsilon_M)$.
2. $|u_{i+1}^T u_i| = O(\eta_U^2)$, $\quad |v_{i+1}^T v_i| = O(\eta_V^2)$.

It is then easy to see that

$$\begin{aligned} e_{k+1}^T(I - U_{k+1}^T U_{k+1}) &= [O(\eta_U), \dots, O(\eta_U), O(\eta_U^2), O(\epsilon_M)], \\ e_{k+1}^T(I - V_{k+1}^T V_{k+1}) &= [O(\eta_V), \dots, O(\eta_V), O(\eta_V^2), O(\epsilon_M)]. \end{aligned} \tag{3.8}$$

Now write ω_k as

$$\begin{aligned} \omega_k^2 &= \|A - U_k B_k V_k^T\|_F^2 \\ &= \|A - U_{k+1} B_{k+1} V_{k+1}^T\|_F^2 + \|u_{k+1}(\beta_{k+1} v_k^T + \alpha_{k+1} v_{k+1}^T)\|_F^2 + \text{term}, \end{aligned}$$

where

$$\begin{aligned} |\text{term}| &= |\text{trace}((A - U_{k+1} B_{k+1} V_{k+1}^T)^T u_{k+1}(\beta_{k+1} v_k^T + \alpha_{k+1} v_{k+1}^T))| \\ &= |\text{trace}((\beta_{k+1} v_k^T + \alpha_{k+1} v_{k+1}^T) A^T (I - U_{k+1} U_{k+1}^T) U_{k+1} e_{k+1})| \\ &= |\text{trace}((\beta_{k+1} v_k^T + \alpha_{k+1} v_{k+1}^T) A^T U_{k+1} (I - U_{k+1}^T U_{k+1}) e_{k+1}| \\ &= |\text{trace}((\beta_{k+1} v_k^T + \alpha_{k+1} v_{k+1}^T) V_{k+1} B_{k+1}^T (I - U_{k+1}^T U_{k+1}) e_{k+1}|. \end{aligned}$$

Now notice that $v_k^T V_{k+1}$ and $v_{k+1}^T v_{k+1}$ have the form

$$\begin{aligned} v_k^T V_{k+1} &= [O(\eta_V), \dots, O(\eta_V), O(\eta_V^2), 1 + O(\epsilon_M), O(\eta_V^2)], \\ v_{k+1}^T v_{k+1} &= [O(\eta_V), \dots, O(\eta_V), O(\eta_V^2), 1 + O(\epsilon_M)], \end{aligned}$$

and furthermore

$$(I - U_{k+1}^T U_{k+1}) e_{k+1} = [O(\eta_U), \dots, O(\eta_U), O(\eta_U^2), O(\epsilon_M)]^T.$$

Also notice that B_{k+1}^T is an *upper bidiagonal* matrix. We have

$$|\text{term}| = O(\eta_U \eta_V) + O(\eta_U^2).$$

4. Level of orthogonality and reorthogonalization. It should not come as a surprise that the level of orthogonality among the left Lanczos vectors $\{U_k\}$ and the level of orthogonality among the right Lanczos vectors $\{V_k\}$ are closely related to each other since the columns of U_k and V_k are generated by the coupled two term recurrence (2.2). In this section we want to explore this relation from several different viewpoints: first we will deal with the whole matrices $I - U_k^T U_k$ and $I - V_k^T V_k$, then columns of these matrices, and finally each individual elements of these matrices. A better understanding of this relation is the key to developing more efficient reorthogonalization schemes for the Lanczos bidiagonalization process, and reorthogonalization is essential not only for computing low rank matrix approximations but also for computing a few dominant singular value triplets. Therefore what we will be discussing with respect to levels of orthogonality is also relevant to singular values/vectors computation.

To proceed, we introduce the following definitions which slightly differ from those used in the previous section because we need to relate levels of orthogonality at different iterative steps of the Lanczos bidiagonalization process:

$$\eta(U_k) = \|I - U_k^T U_k\|_2, \quad \eta(V_k) = \|I - V_k^T V_k\|_2.$$

PROPOSITION 4.1. *Assume that B_k generated by the two term recurrence (2.2) is nonsingular. Then*

$$\eta(V_k) \leq \|B_k^{-1}\|\|\tilde{B}_k\|\eta(U_{k+1}) + O(\|B_k^{-1}\|A\|_F\|\epsilon_M),$$

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

$$\eta(U_{k+1}) \leq \frac{\|B_k\|\eta(V_k)}{2\sigma_{\min}(\tilde{B}_k)} + O\left(\frac{\|A\|_F\epsilon_M}{\sigma_{\min}(\tilde{B}_k)}\right).$$

Proof. We can write (2.4) as

$$AV_k = U_kB_k + \beta_{k+1}u_{k+1}e_k^T + F_k, \quad A^TU_k = V_kB_k^T + G_k,$$

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

$$U_k^TAV_k = U_k^TU_kB_k + \beta_{k+1}U_k^Tu_{k+1}e_k^T + U_k^TF_k,$$
$$U_k^TAV_k = B_kV_k^TV_k + G_k^TV_k.$$

This leads to

$$\begin{aligned} B_k(I - V_k^TV_k) &= (I - U_k^TU_k)B_k - \beta_{k+1}U_k^Tu_{k+1}e_k^T - U_k^TF_k + G_k^TV_k^T, \\ &= [I - U_k^TU_k, -U_k^Tu_{k+1}]\tilde{B}_k - U_k^TF_k + G_k^TV_k^T. \end{aligned} \tag{4.9}$$

Since B_k is nonsingular and $\|[I - U_k^TU_k, -U_k^Tu_{k+1}]\| \leq \eta(U_{k+1})$, we have

$$\eta(V_k) \leq \|B_k^{-1}\|\|\tilde{B}_k\|\eta(U_{k+1}) + O(\|B_k^{-1}\|A\|_F\|\epsilon_M).$$

On the other hand, it follows from Equation (4.9) that

$$\begin{aligned} \sigma_{\min}(\tilde{B}_k)\|[I - U_k^TU_k, -U_k^Tu_{k+1}]\| &\leq \|[I - U_k^TU_k, -U_k^Tu_{k+1}]\tilde{B}_k\| \\ &\leq \|B_k\|\|I - V_k^TV_k\| + O(\|A\|_F\epsilon_M). \end{aligned}$$

It is easy to see that $\|I - U_{k+1}^TU_{k+1}\| \leq 2\|[I - U_k^TU_k, -U_k^Tu_{k+1}]\|$, and B_k nonsingular implies $\sigma_{\min}(\tilde{B}_k) > 0$. Combining the above two inequalities completes the proof. □

The above result says that as long as B_k and $\tilde{B}_k$ are not very ill-conditioned, the level of orthogonality among the columns of U_{k+1} and the level of orthogonality among the columns of V_k should be roughly comparable to each other. We will illustrate this using some numerical examples in a moment. Now we look at the relation of levels of orthogonality from the viewpoint of individual vectors of $I - U_{k+1}^TU_{k+1}$ and $I - V_k^TV_k$.

PROPOSITION 4.2. *Assume that B_{k-1} is nonsingular. Then*

$$\|V_{k-1}^Tv_k\| \leq \|B_{k-1}^{-1}\|(\alpha_k\|U_{k-1}^Tu_k\| + \beta_{k+1}\|U_{k-1}^Tu_{k+1}\| + O(\|A\|_F\epsilon_M)).$$

Proof. Take the last column of both sides of the first equation in (4.9), we have

$$B_k\begin{bmatrix} -V_{k-1}^Tv_k \\ O(\epsilon_M) \end{bmatrix} = \alpha_k\begin{bmatrix} -U_{k-1}^Tu_k \\ O(\epsilon_M) \end{bmatrix} - \beta_{k+1}U_k^Tu_{k+1} + \text{term},$$

where $|\text{term}| = O(\|A\|_F\epsilon_M)$. Notice that B_k is lower bidiagonal, we have

$$-V_{k-1}^Tv_k = B_{k-1}^{-1}(\alpha_k(-U_{k-1}^Tu_k) - \beta_{k+1}U_{k-1}^Tu_{k+1}) + O(\|A\|_F\epsilon_M),$$

which leads to the bound we want to prove. □

EXAMPLE 1. In this example we apply the recurrence (2.2) to several test matrices. No reorthogonalization is carried out. The initial vector b is always chosen to be a vector of all ones. We first use a test matrix from SVDPACK to illustrate the relation between the levels of orthogonality among columns of U_{k+1} and V_k. The matrix is a term-document matrix from an information retrieval application by Apple Computer Inc. [18]. It is sparse and of dimension 3206 × 44. Its singular values are plotted in Figure 9 in Section 5. We first apply the Lanczos bidiagonalization process to A^T. For $k = 2, 3, \ldots, 11$, we tabulate the four quantities $\eta(U_k), \eta(V_k), \mathrm{cond}_2(B_k), \mathrm{cond}_2(\tilde{B}_k)$ as follows.

k	$\eta(U_k)$	$\eta(V_k)$	$\mathrm{cond}_2(B_k)$	$\mathrm{cond}_2(\tilde{B}_k)$
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 $\{U_k\}$ and $\{V_k\}$ are completely lost. We notice that both B_k and $\tilde{B}_k$ are well-conditioned, and therefore $\eta(U_{k+1})$ and $\eta(V_k)$ 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 B_k, the matrix B_k becomes more and more ill-conditioned as k increases. However, $\tilde{B}_k$ does not become ill-conditioned and therefore $\eta(U_{k+1})$ and $\eta(V_k)$ are still comparable to each other.

k	$\eta(U_k)$	$\eta(V_k)$	$\mathrm{cond}_2(B_k)$	$\mathrm{cond}_2(\tilde{B}_k)$
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

Now we look at another two test matrices which are taken from P.C. Hansen's Regularization toolbox [16]. The first matrix is `phillips(100)`, a square matrix of dimension 100. Its singular values are plotted in Figure 5 in Section 5. In what follows we tabulate the same set of four quantities $\eta(U_k), \eta(V_k), \mathrm{cond}_2(B_k), \mathrm{cond}_2(\tilde{B}_k)$ for $k = 2, \dots, 9$.

k	$\eta(U_k)$	$\eta(V_k)$	$\mathrm{cond}_2(B_k)$	$\mathrm{cond}_2(\tilde{B}_k)$
2	1.0838e-15	1.1567e-15	1.8355e+00	1.4510e+00
3	2.3688e-15	3.3906e-15	3.2510e+00	2.4812e+00
4	8.4092e-15	3.7248e-14	7.7514e+00	6.7785e+00
5	4.6956e-13	2.5136e-11	6.1892e+01	6.1104e+01
6	9.6335e-09	1.3857e-06	1.4612e+02	6.1700e+01
7	8.3108e-05	7.7110e-03	2.8009e+02	1.2250e+02
8	8.8663e-01	9.9535e-01	2.8010e+02	2.0295e+02
9	9.9621e-01	9.9998e-01	2.8143e+02	2.0303e+02

For this test matrix the loss of orthogonality progresses faster because some of the β_k and α_k are small. We notice that the relation $\eta(U_{k+1}) \approx \mathrm{cond}(\tilde{B}_{k+1})\eta(V_k)$ holds.

The last matrix in this example is wing(100), a square matrix of dimension 100. Its singular values are plotted in Figure 6 in Section 5. We tabulate $\eta(U_k), \eta(V_k), \mathrm{cond}_2(B_k), \mathrm{cond}_2(\tilde{B}_k)$ in the following.

k	$\eta(U_k)$	$\eta(V_k)$	$\mathrm{cond}_2(B_k)$	$\mathrm{cond}_2(\tilde{B}_k)$
2	7.8559e-16	5.3386e-15	2.2673e+01	1.3273e+01
3	8.7147e-14	3.8423e-11	7.6085e+02	4.0375e+02
4	1.8297e-08	3.9253e-04	4.0433e+04	2.5253e+03
5	9.9306e-01	1.0000e+00	4.0433e+04	1.8992e+04
6	1.0000e+00	1.0000e+00	2.8263e+06	1.8992e+04

We notice again that the relation $\eta(U_{k+1}) \approx \mathrm{cond}(\tilde{B}_{k+1})\eta(V_k)$ holds. In summary, if no reorthogonalization is performed in the Lanczos bidiagonalization process, then either $\eta(U_{k+1}) \approx \mathrm{cond}(\tilde{B}_{k+1})\eta(V_k)$ or $\eta(V_k) \approx \mathrm{cond}(B_k)\eta(U_{k+1})$ tends to hold.

Now we look at each of the individual elements of the matrices $I - U_{k+1}^T U_{k+1}$ and $I - V_k^T V_k$ and derive recurrence relations that can be used to monitor the loss of orthogonality among the Lanczos vectors $\{U_{k+1}\}$ and $\{V_k\}$. The result can be considered as extension of similar result for monitoring the loss of orthogonality in the Lanczos tridiagonalization process for symmetric matrices [17].

PROPOSITION 4.3. *Define* $\omega_{ik} = u_i^T u_k = u_k^T u_i$, $\delta_{ik} = v_i^T v_k = v_k^T v_i$. *Then those quantities satisfy the following coupled recurrences.*

$$\omega_{ii} = \delta_{ii} = 1$$

$$\beta_{i+1}\omega_{i+1,k} = \alpha_k\delta_{i,k} + \beta_k\delta_{i,k-1} - \alpha_i\omega_{ik} + (u_k^T f_i - v_i^T g_k) \tag{4.10}$$

$$\alpha_{i+1}\delta_{i+1,k} = \beta_{k+1}\omega_{i+1,k+1} + \alpha_k\omega_{i+1,k} - \beta_{i+1}\delta_{ik} + (v_k^T g_{i+1} - u_{i+1}^T f_k),$$

where $k = 1, \dots, i$ *and* $\delta_{i0} = 0$.

Proof. It follows from Equation (2.3) that

$$\beta_{i+1} u_k^T u_{i+1} = u_k^T A v_i - \alpha_i u_k^T A u_i + u_k^T f_i, \tag{4.11}$$

$$\alpha_{i+1} v_k^T v_{i+1} = v_k^T A^T u_{i+1} - \beta_{i+1} v_k^T v_i + v_k^T g_{i+1}. \tag{4.12}$$

In Equation (4.11) set i to k and k to $i+1$ we obtain

$$\beta_{k+1} u_{i+1}^T u_{k+1} = u_{i+1}^T A v_k - \alpha_k u_{i+1}^T A u_k + u_{i+1}^T f_k, \tag{4.13}$$

and in Equation (4.12) set $i+1$ to k and k to i we obtain

$$\alpha_k v_i^T v_k = v_i^T A^T u_k - \beta_k v_i^T v_{k-1} + v_i^T g_k \tag{4.14}$$

Subtracting (4.14) from (4.11) and (4.13) from (4.12) and simplifying yields the results. □

REMARK. Although the derivation of the above recurrence is straightforward, it has not been discussed in the literature before. One of the reasons might be that there has been no implementation of the Lanczos bidiagonalization process using the coupled two-term recurrence (2.2) with *partial* reorthogonalization to maintain *semi-orthogonality* (See the remarks at the beginning of Section 2).

As mentioned before we need to keep certain level of orthogonality among the Lanczos vectors $\{U_k\}$ and $\{V_k\}$ in order to obtain a good approximation J_k. Certain level of orthogonalization will also ensure that no spurious singular values will appear. As is in the Lanczos tridiagonalization process, maintaining orthogonality of both $\{U_k\}$ and $\{V_k\}$ to full machine precision is not necessary. What is needed is the so-called *semiorthogonality*among the left and right Lanczos vectors [14, 17], i.e., carrying out reorthogonalization so that $\eta(U_k) = O(\sqrt{\epsilon_M})$ and $\eta(V_k) = O(\sqrt{\epsilon_M})$ are maintained throughout the bidiagonalization process. Developing an implementation of the Lanczos bidiagonalization process using the coupled two-term recurrence (2.2) and incorporating partial reorthogonalization based on (4.10) is very similar to the symmetric tridiagonalization case. Therefore in the following we only give the outline of the approach and the list of the pseudo-code. The recurrence (4.10) is used to monitor the level of orthogonality of the left and right Lanczos vectors. Since the f_i and g_k represent local rounding errors and are therefore not known, they are replaced by terms that simulate the rounding error process. Following [14, 17], we replace (4.10) by the following,

$$\begin{aligned}
&\epsilon_l = \epsilon_M\sqrt{m}, \quad \epsilon_r = \epsilon_M\sqrt{n}\\
&\omega_{ii} = \delta_{ii} = 1\\
&\omega_{i+1,k} = (\alpha_k\delta_{i,k} + \beta_k\delta_{i,k-1} - \alpha_i\omega_{ik} + \epsilon_l)/\beta_{i+1}\\
&\delta_{i+1,k} = (\beta_{k+1}\omega_{i+1,k+1} + \alpha_k\omega_{i+1,k} - \beta_{i+1}\delta_{ik} + \epsilon_r)/\alpha_{i+1},
\end{aligned} \tag{4.15}$$

where $k = 1,\dots,i$ and $\delta_{i0} = 0$. The following pseudo-code summarizes the algorithm.

```
Algorithm. Semi-Orth
Using recurrence (2.2) to compute $u_{i+1}, v_{i+1}, \alpha_{i+1}, \beta_{i+1}$ and do
Update the $\omega$-$\delta$ recurrence (4.15)
Set $\omega_{\max} = \max_{1\le j\le i} \omega_{i+1,j}$, $\delta_{\max} = \max_{1\le j\le i} \delta_{i+1,j}$
if $\omega_{\max} \ge \sqrt{\epsilon_M}$ then
    orthogonalize $u_i$ against $U_{i-1}$
    orthogonalize $u_{i+1}$ against $U_i$
    reset the $\omega$-$\delta$ recurrence (4.15) with
    $\omega_{i+1,j} = \epsilon_l, j = 1,\dots,i$
else if $\delta_{\max} \ge \sqrt{\epsilon_M}$ then
    orthogonalize $v_i$ against $V_{i-1}$
    orthogonalize $v_{i+1}$ against $V_i$
    reset the $\omega$-$\delta$ recurrence (4.15) with
    $\delta_{i+1,j} = \epsilon_r, j = 1,\dots,i$
end if
```

Before we discuss another efficient reorthogonalization scheme we first provide some motivations. In some applications such as compression of multiple-spectral and hyper-spectral image cubes [4, 8], principal component analysis for face recognition and image databases [20, 11, 19, 15], 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 2-D image array is stacked into a single long vector.[3] For a 512 × 512 2-D image the row dimension of the resulting matrix A is 262144,[4] 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 while now channels are numbered in the several hundreds upto 1024. The number of face images used in an image database ranges from several hundred to several thousand [20]. Therefore in those applications the matrix A is very skinny, i.e., $m \gg n$. To facilitate the discussion, we will say that those Lanczos vectors with smaller dimension belong to the *short* space while those with larger dimension 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 on 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 on 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 towards 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. That we can get by with this kind of *one-sided* reorthogonalization is partially justified by the fact that the levels of orthogonality among the left and right Lanczos vectors are closely related to each other (cf. Proposition 4.1),

$$\frac{\eta(V_k)}{\|B_k^{-1}\|\|\tilde{B}_k\|} + O(\|A\|_F\epsilon_M) \le \eta(U_{k+1}) \le \frac{\|B_k\|\eta(V_k)}{2\sigma_{\min}(\tilde{B}_k)} + O\left(\frac{\|A\|_F\epsilon_M}{\sigma_{\min}(\tilde{B}_k)}\right).$$

Therefore enforcing certain level of orthogonality on $\{V_k\}$ will affect the level of orthogonality of $\{U_{k+1}\}$. However, the level of orthogonality of $\{U_{k+1}\}$ is not unconditionally controlled by the level of orthogonality of $\{V_k\}$. One also needs to take into account the effect of $\|B_k\|/\sigma_{\min}(\tilde{B}_k)$ and $\|B_k^{-1}\|\|B_k\|$, i.e., if $\tilde{B}_k$ and/or B_k is too ill-conditioned, $\eta(V_k)$ may not be comparable to $\eta(U_{k+1})$.

Now we proceed to describe the algorithm with one-sided reorthogonalization. We will come back to the issue of orthogonality later on. For $A \in \mathcal{R}^{m\times n}$ with $m \gg n$. At each step of the Lanczos bidiagonalization process, we orthogonalize v_{i+1} against all the previous Lanczos vectors and leave u_{i+1} unchanged. In the following we list the pseudo code of the one-sided reorthogonalization.

```
Algorithm. One-sided
For $i = 0, 1, \dots,$
Using recurrence (2.2) to compute $u_{i+1}, v_{i+1}, \alpha_{i+1}, \beta_{i+1}$ and do
    orthogonalize $v_{i+1}$ against $V_i$
```

EXAMPLE 2. We look at levels of orthogonality of $\{U_k\}$ and $\{V_k\}$ computed by ALGORITHM ONE-SIDED. For the following test matrices we always perform reorthogonalization in the *short* space.

[3] 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.

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

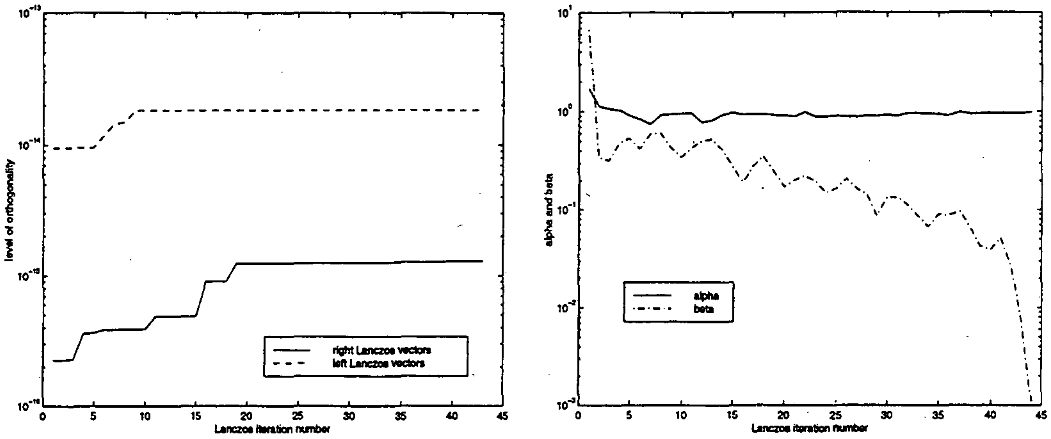

FIG. 1. *(left) level of orthogonality, (right)* α_k *and* β_k

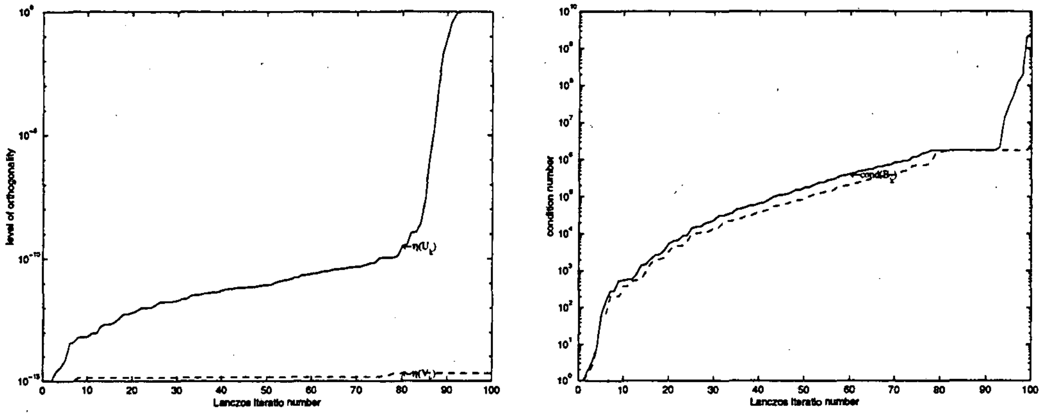

FIG. 2. *(left) level of orthogonality, (right) Condition numbers* $\mathrm{cond}(B_k)$ *and* $\mathrm{cond}(\tilde{B}_k)$

The first matrix is the 3206 × 44 matrix from SVDPACK which is also used in Example 1. We first apply ALGORITHM ONE-SIDED to A^T. In figure 1 on the left we plot $\eta(U_k)$ and $\eta(V_k)$ for $k = 2, \dots, 44$, and on the left we plot the two sequences $\{\alpha_k\}$ and $\{\beta_k\}$. Notice that β_k drops sharply towards the end of the Lanczos run, but this does not affect the level of orthogonality of either $\{U_k\}$ or $\{V_k\}$. The condition numbers for both B_k and $\tilde{B}_k$ 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 in the short space. Now we have $\eta(U_k) \approx 10^{-14}$ and $\eta(V_k) \approx 10^{-15}$. We notice that one singular value of B_k tracks a spurious zero singular value resulting in increasingly larger $\mathrm{cond}(B_k)$ but $\mathrm{cond}(\tilde{B}_K)$ stays $O(1)$. Again the level of orthogonality of the long space is well controlled by that of the short space.

Next we consider the matrix `phillips(100)`, a square matrix of dimension 100 used in Example 1. We apply ALGORITHM ONE-SIDED to A and perform reorthogonalization on $\{V_k\}$. In Figure 2 on the left we plot $\eta(U_k)$ and $\eta(V_k)$ for $k = 2, \dots, 100$, and on the left we plot the two sequences $\{\mathrm{cond}(B_k)\}$ and $\{\mathrm{cond}(\tilde{B}_k)\}$. The vectors $\{v_i\}$ are explicitly orthogonalized and hence $\eta(V_k)$ stay at the level of ϵ_M. Now both B_k and $\tilde{B}_k$ become more and more ill-conditioned as k increases, enforcing orthogonality of $\{V_k\}$ will not completely control the orthogonality of $\{U_k\}$, one has to take into account the growth of the condition numbers of B_k and $\tilde{B}_k$. Compare with the results obtained using Lanczos bidiagonalization without any reorthogonalization (cf. Example 1), one-sided reorthogonalization does delay the process of orthogonality loss in $\{U_k\}$, but it does not completely eliminate it. Does this mean one-sided reorthogonalization can not be used for matrices like phillips(100) for which the level of orthogonality of the long space can not be controlled by the level of orthogonality of the short space? The answer is it still can and we will explain why in a moment.

REMARK. If we reorthogonalize in the long space, the control on the orthogonality in the short space is not as effective. There are also many interesting issues as to whether accurate singular values and vectors (both left and right) of A can still be easily computed even if either U_k or V_k (not both) completely lose orthogonality. These issues will be dealt with in a separate paper.

Now we explain why we still can obtain good low rank approximation J_k even if the level of orthogonality in the long space is completely lost. Assume that in the recurrence (2.2) we perform reorthogonalization for each vector u_{i+1} as follows, first we compute

$$\hat{\beta}_{i+1}\hat{u}_{i+1} = Av_i - \alpha_i u_i - f_i,$$

then we orthogonalize $\hat{\beta}_{i+1}\hat{u}_{i+1}$ against all the previous vectors $u_1, \ldots, u_i$ to obtain

$$\beta_{i+1}u_{i+1} = \hat{\beta}_{i+1}\hat{u}_{i+1} - \sum_{j=1}^{i} \hat{\beta}_{i+1}(\hat{u}_{i+1}^T u_j)u_j - \tilde{f}_i,$$

where $\tilde{f}_i$ accounts for the local rounding error and β_{i+1} is chosen such that $\|u_{i+1}\| = 1$. Combining the above two equations we obtain

$$\beta_{i+1}u_{i+1} = Av_i - \alpha_i u_i - \hat{f}_i$$

with $\hat{f}_i = f_i + \tilde{f}_i + \sum_{j=1}^{i} \hat{\beta}_{i+1}(\hat{u}_{i+1}^T u_j)u_j$. In compact matrix form we have

$$AV_k = U_{k+1}\tilde{B}_k + \hat{F}_k, \quad \hat{F}_k = [\hat{f}_1, \ldots, \hat{f}_k].$$

In general there is no guarantee that $\|\hat{F}_k\| = O(\|A\|_F \epsilon_M)$ as would be the case if no reorthogonalization is performed. However, notice that the other half of the recurrence still has the form

$$A^T U_{k+1} = V_{k+1}B_{k+1}^T + G_{k+1}$$

with $\|G_{k+1}\| = O(\|A\|_F \epsilon_M)$. It follows from the above equation that

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

Notice that U_{k+1} is orthonormal within working precision since u_{i+1} is explicitly orthogonalized against U_i for $i = 1, \ldots, k$.[5] Therefore $J_{k+1} = U_{k+1}B_{k+1}V_{k+1}^T$ will be a good approximation of A_{k+1} as long as U_{k+1} is a good approximation of the first $k+1$ left singular vectors of A (cf. Section 3). The above statement is true regardless of the level of orthogonality of V_{k+1}. We will have more to say about this in the next section.

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 $J_k = U_k B_k V_k^T$ as a low rank approximation of a given matrix A without computing any SVD. Compared with the approach where SVD of B_k is computed and its left and right singular vectors are combined with the left and right

[5] Sometimes a second orthogonalization is needed to achieve orthonormality within working precision [14, Section 6.9].

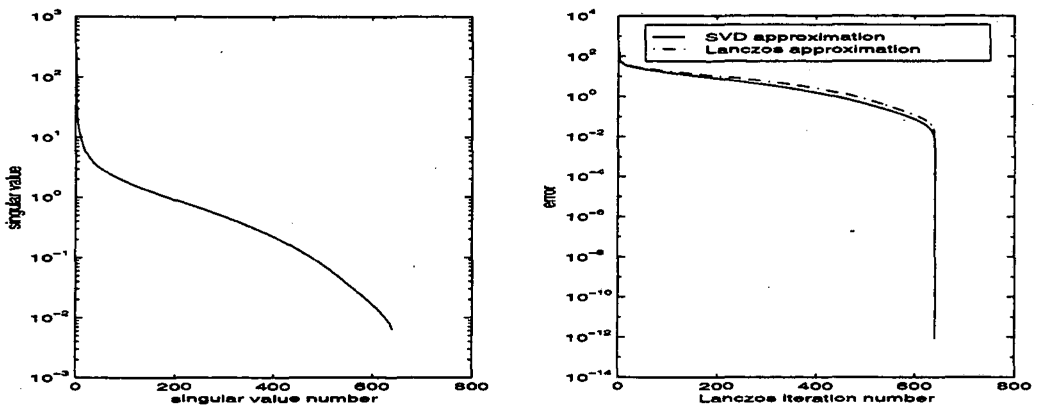

FIG. 3. *Plots for* cyclotron

Lanczos vectors, 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 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 B_k and the combination of the singular vectors and Lanczos vectors have to be done after the Lanczos bidiagonalization process. In [2] the computation of the SVD of 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. However, 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 five classes of matrices and compared the low rank approximations computed by ALGORITHM ONE-SIDED with those computed by the SVD.

- 2-D images.
- Ill-conditioned test matrices from Regularization Tools [16].
- Large sparse test matrices from SVDPACK [18].
- Several general rectangular matrix from Matrix Market [9].
- 3-D image cubes from remote sensing application.

All the computation is done using MATLAB Version 5 on a Sun sever 2000. For each test matrix we first plot the singular values of the matrix and then the two sequences $\{\|A - U_kB_kV_k^T\|_F\}$ 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 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.

EXAMPLE 1. A detailed description of using singular value decomposition for single 2-D image compression/coding is given in [1]. We have tested ALGORITHM ONE-SIDED on many 2-D image arrays which are digitized images of everyday-life photos. The results are very similar and therefore we only present one test result. This image is a digitized photo in jpeg format of a night scene of the 184-Inch

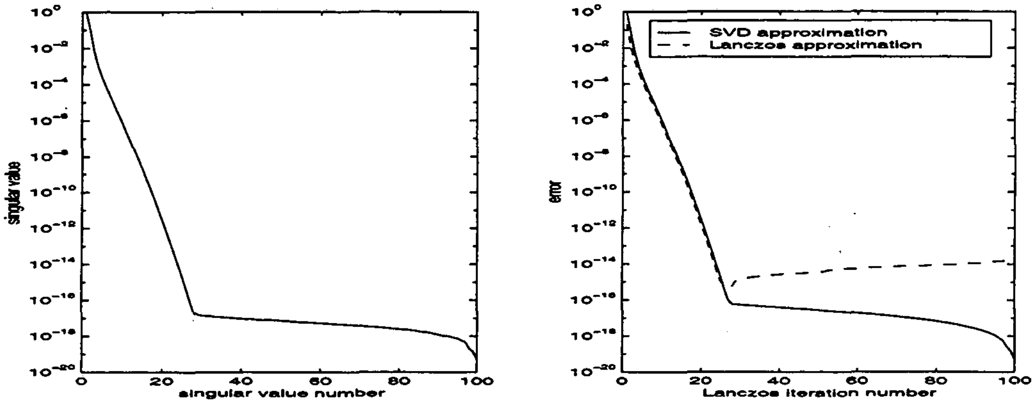

FIG. 4. *Plots for* foxgood(100)

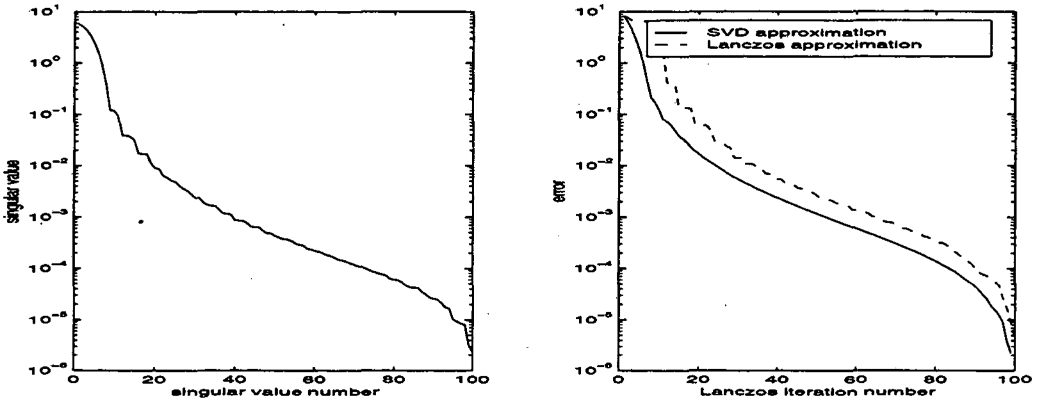

FIG. 5. *Plots for* phillips(100)

Cyclotron building at Lawrence Berkeley National Laboratory. We only use the red color component of the original image array which is of dimension $837 \times 640 \times 3$. The resulting matrix A is of dimension 837×640. The class of A is 8-bit integer and is converted to double precision real numbers before ALGORITHM ONE-SIDED is applied. The sharp dip in the curves plotted on the right of Figure 3 is due to the fact we used $\|A - A_{\min(m,n)}\|_F$ and $\|A - J_{\min(m,n)+1}\|_F$ for the last element of the two sequences plotted. In exact arithmetic they should be zero. We also noticed that the singular values of most single 2-D images have several order of magnitude spread between $\sigma_{\min}$ and $\sigma_{\max}$, but usually there is no sharp gap in the singular value spectrum.

EXAMPLE 2. Truncated SVD is a very useful tool for solving ill-posed problems. The first step involved is to find a low rank approximation A_k of the original matrix A computed from the SVD of A [7]. This step can be replaced by using $J_k = U_k B_k V_k^T$ instead, and this will be especially efficient when the matrix A is large and sparse. The MATLAB regularization tools developed by P.C. Hansen contain eleven m-files for generating test matrices which are very ill-conditioned for testing regularization algorithms [16]. We have tested all the eleven classes of matrices with dimension 100×100.[6] The singular value spectrum of the test matrices of some of the classes are rather similar

[6] In [16] the dimension of the test matrices is an input argument and therefore can be set at will. All the test matrices we used are of dimension 100×100 except parallax(100) which is 100×23 since only 23 observations are available.

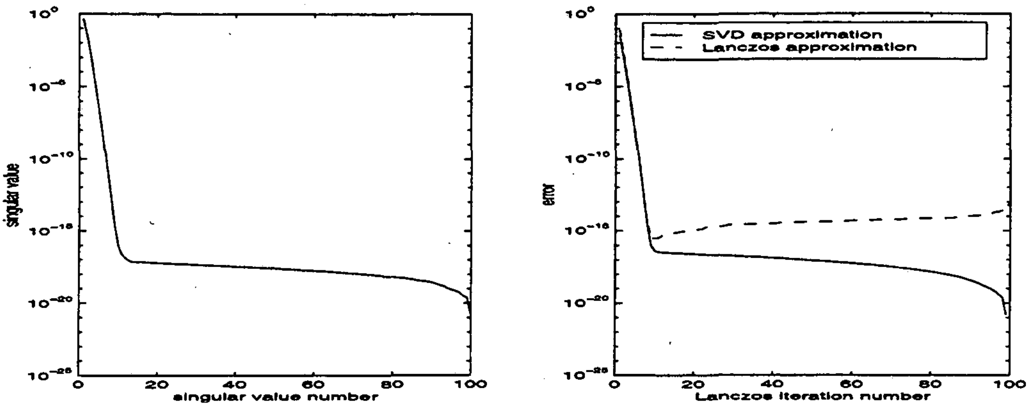

FIG. 6. *Plots for* wing(100)

and therefore we did not repeat their results here. The five classes selected are

- foxgood
- phillips
- wing
- parallax
- ilaplace

Except for phillips(100), all the other four matrices have singular values that are much smaller than eps = 2.2204e-16. In the tests we have done the J_k matrices computed by ALGORITHM ONE-SIDED will not give arise to $\omega_k = \|A - J_k\|_F$ that is below eps. Now we analyze the computed results for phillips(100) a bit further. We also computed

$$\text{ratio}_k = \frac{\left(\sum_{j=k+1}^{\min(m,n)} \sigma_j^2\right)^{1/2}}{\|A - U_k B_k V_k^T\|_F}, \tag{5.16}$$

and find out that for phillips(100),

$$\max(\text{ratio}_k) = 9.8416\text{e}-01 \quad \min(\text{ratio}_k) = 3.3009\text{e}-02.$$

Most of ratio$_k$ are above 0.1 and only 5 are under 0.1 which occur at $k = 7, 8, 9, 10, 11$. Figure 2 plots the level of orthogonality for phillips(100). We notice that even though level of orthogonality for $\{V_k\}$ is about eps, towards to the end of the Lanczos run the orthogonality of $\{U_k\}$ is completely lost. However, $\|A - J_k\|_F$ is not dominated by the level of orthogonality of $\{U_k\}$. We have, for example, $\|A - J_{100}\|_F = 4.4589\text{e}-06$ and $\|A - J_{101}\|_F = 1.1065\text{e}-14$. This confirms our analysis at the end of Section 4. Another thing we noticed is that when $\omega_k = \|A - J_k\|_F$ falls around the level of eps, the monotonicity of $\{\omega_k\}$ no longer holds, see Figure 4 and Figure 6.

EXAMPLE 3. Three test matrices are included in SVDPACK [18]. 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.

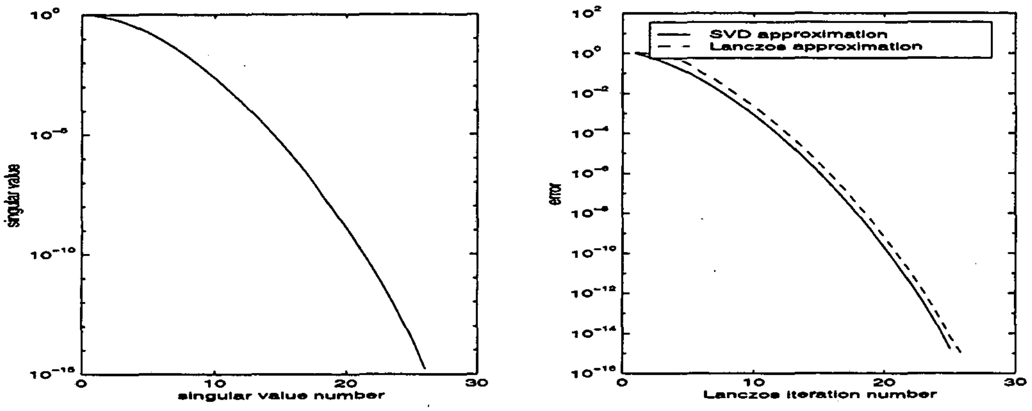

FIG. 7. *Plots for* parallax(100)

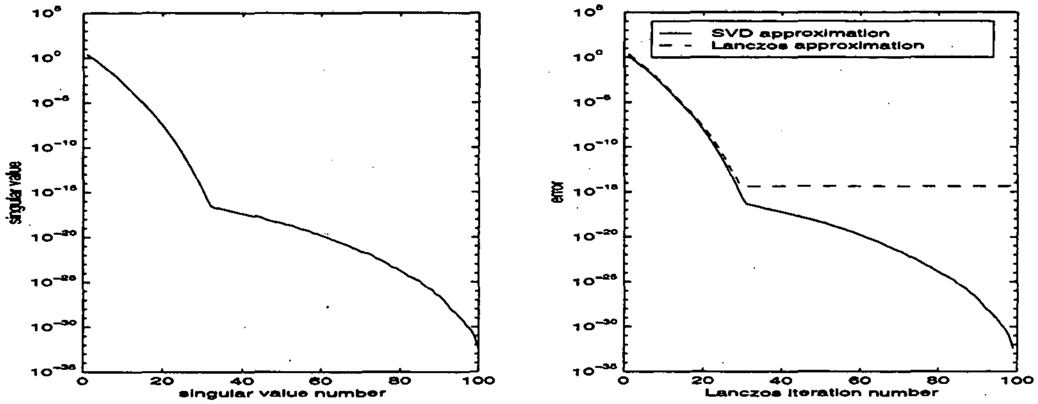

FIG. 8. *Plots for* ilaplace(100)

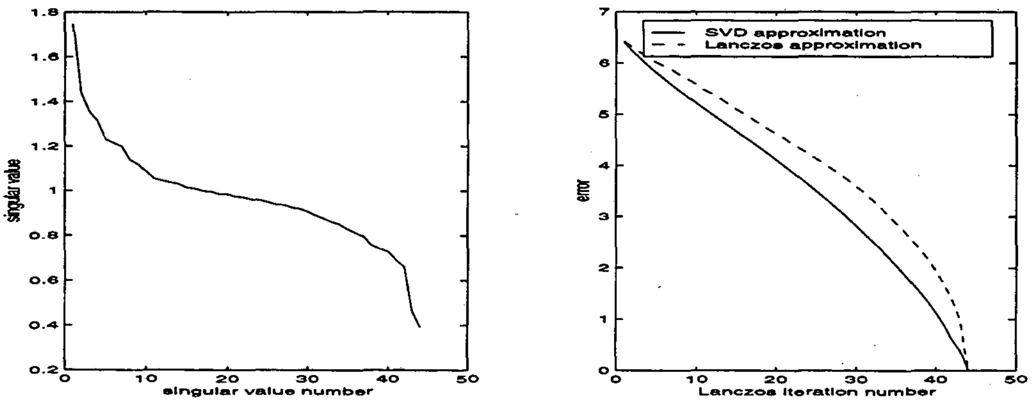

FIG. 9. *Plots for* apple1.mat

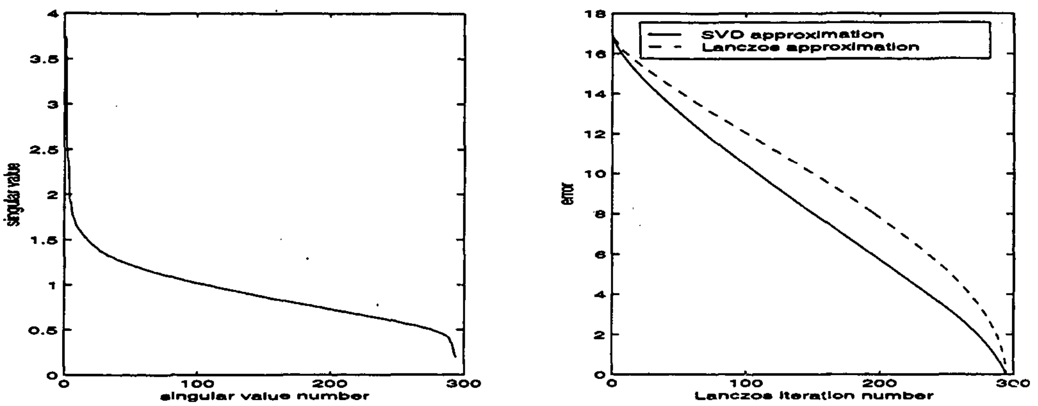

FIG. 10. *Plots for* apple2.mat

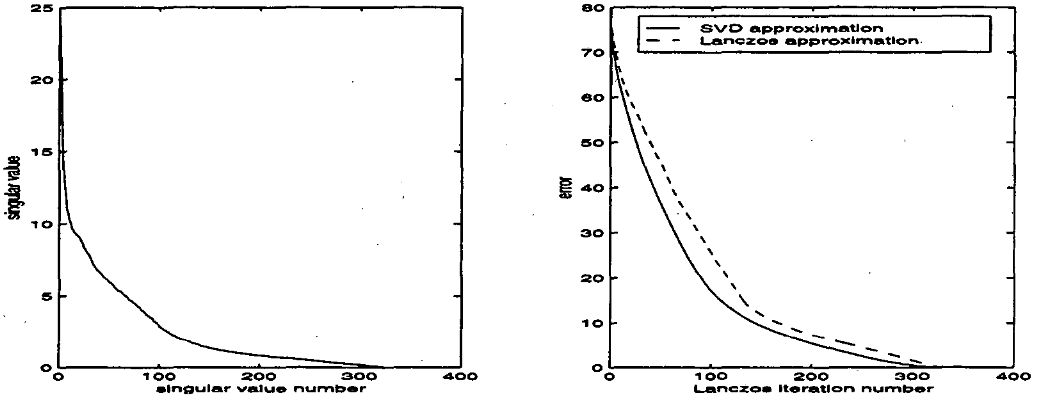

FIG. 11. *Plots for* amoco.mat

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 $\{U_k\}$ and $\{V_k\}$ is around 10^{-14}. For amoco.mat, the level of orthogonality for the long space deteriorates from 10^{-14} to 10^{-12} at the end of 330 steps. In the following table we list both the maximum and minimum of the ratio $\{\mathtt{ratio}_k\}$ defined in Equation (5.16) for the three matrices. It is also interesting to notice that even though there is difference between $\|A - A_k\|_F$ and $\|A - J_k\|_F$ for a fixed k, it is always possible to move forward a few steps s to get a J_{k+s} such that $\|A - J_{k+s}\|_F \approx \|A - A_k\|_F$. For this 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.

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

EXAMPLE 4. 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 [9]. This set contains four matrices 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

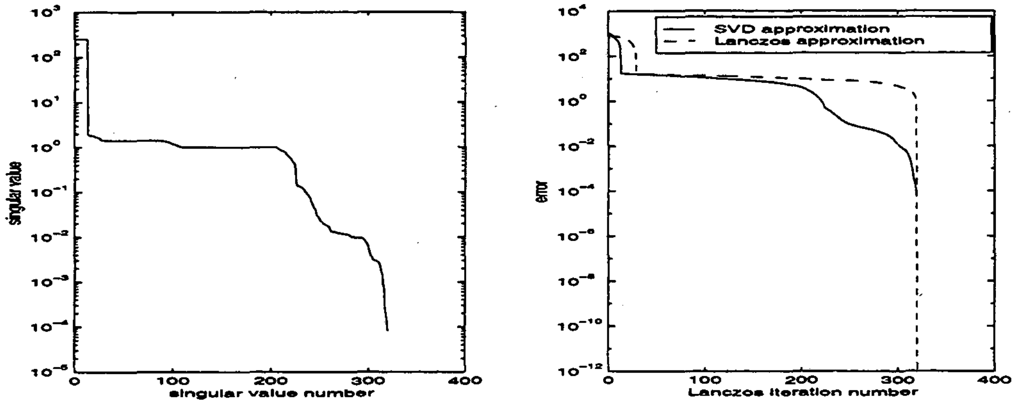

FIG. 12. *Plots for* illc1033.mat

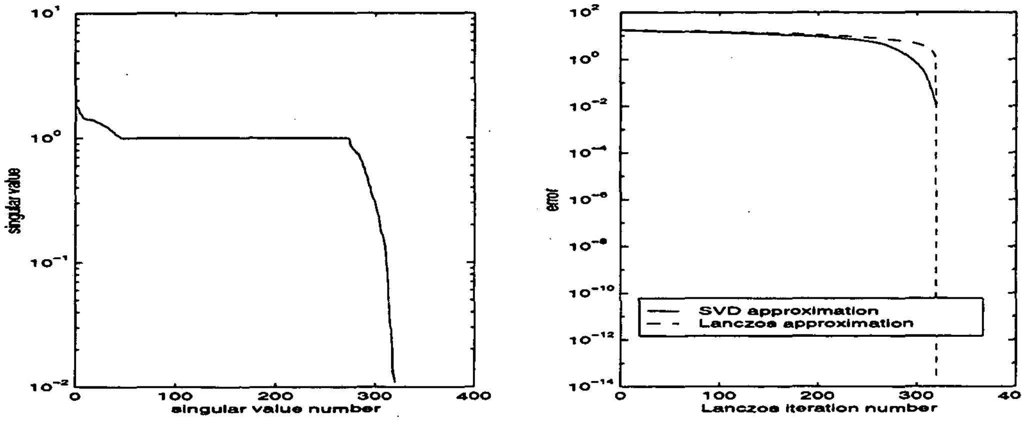

FIG. 13. *Plots for* well1033.mat

clusters of singular values which are very close to each other. For example, the 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 $J_k = U_k B_k V_k^T$ 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 $\sigma_{\max}(A)$, $J_k = U_k B_k V_k^T$ 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. This is why in the right plot of Figure 12, 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}.

EXAMPLE 5. 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 [10]. The data values are proportional to radiance units. The

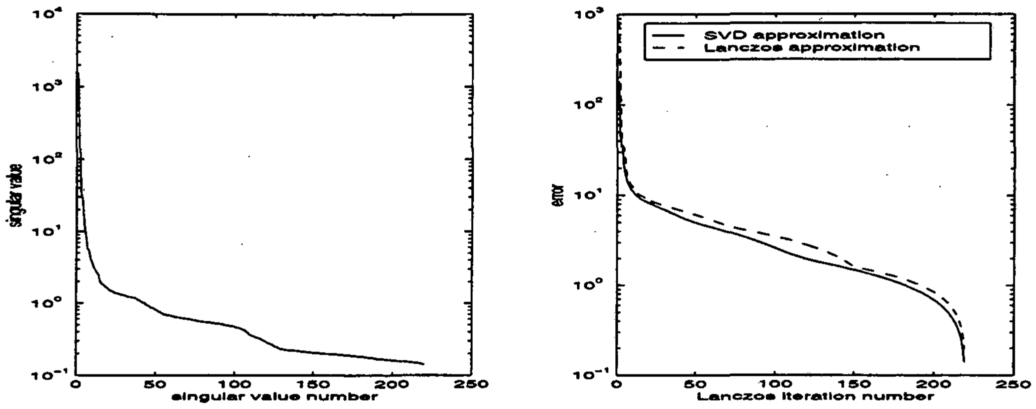

FIG. 14. *Plots for* `92AV3C.mat`

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, Illinois with the AVIRIS system which is operated by NASA JPL and AMES.[7] Each of the 2-D image is of dimension 145 x 145 and therefore the resulting matrix A is of dimension 21025 x 220. We applied ALGORITHM ONE-SIDED to A^T with the starting b a vector of all ones. The left of Figure 14 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 the 2-D image matrix of dimension 837 $\times$ 640 in Example 1, 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 $\mathtt{max(ration}_k\mathtt{)} = \mathtt{9.3841e-01}$ and $\mathtt{min(ration}_k\mathtt{)} = \mathtt{2.2091e-01}$ 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 matrix approximations can be obtained directly from the Lanczos bidiagonalization process without computing and singular value decomposition. We discussed several theoretical and practical issues such as *a priori* and *a posteriori* error estimation, recursive computation of stopping criterion, and relations between levels of orthogonality of the left and right Lanczos vectors. We also discussed two efficient reorthogonalization schemes: semi-reorthogonalization and 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. We are currently working on implementations of the algorithms proposed on distributed memory machines such as Cray T3E.

ACKNOWLEDGEMENT. The authors thank Sherry Li and John Wu of NERSC, Lawrence Berkeley National Laboratory for many helpful discussions and assistance.

REFERENCES

[7] 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.

[1] H.C. Andrews and C.L. Patterson. The singular value decomposition (SVD) image coding. *IEEE Transactions on Communications*, COM-24:425-432, 1976.

[2] M. Berry. Large Scale Singular Value Computations. *International Journal of Supercomputer Applications*, 6:13-49, 1992.

[3] J. Cullum, R. A. Willoughby and M. Lake. A Lanczos Algorithm for Computing Singular Values and Vectors of Large Matrices. *SIAM Journal on Scientific and Statistical Computing*, 4:197–215, 1983.

[4] P. Geladi and H. Grahn. *Multivariate image analysis*. Wiley, New York, 1996.

[5] G. H. Golub and W. Kahan. Calculating the Singular Values and Pseudo-Inverse of a Matrix. *SIAM Journal on Numerical Analysis*, 2:205-224, 1965.

[6] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, Maryland, 2nd edition, 1989.

[7] P. C. Hansen. Truncated SVD Solutions to Discrete Ill-Posed Problems with Ill-Determined Numerical Rank. *SIAM Journal on Scientific and Statistical Computing*, 11:503–518, 1990.

[8] C. Lee and D.A. Landgrebe. Analyzing high dimensional multispectral data. IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, 31:792-800, 1993.

[9] Matrix Market. http://math.nist.gov/MatrixMarket/.

[10] MultiSpec. http://dynamo.ecn.purdue.edu/ biehl/MultiSpec/documentation.html.

[11] A. O'Toole, H. Abdi, K.A. Deffenbacher and D. Valentin. Low-dimensional representation of faces in higher dimensions of the face space. *Journal of American Optical Society*, 10:405-411, 1993.

[12] C. C. Paige. Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix. *Journal of Institute of Mathematics and Applications*, 18: 341–349, 1976.

[13] C. C. Paige and M. A. Saunders. LSQR: an Algorithm for Sparse Linear Equations and Sparse Least Squares. *ACM Transaction on Mathematical Software*, 8:43-71, 1982.

[14] B.N. Parlett. *The Symmetric Eigenvalue Problem*. Prentice Hall Inc., Englewood Cliffs, New Jersey, 1980.

[15] A. Petland, R.W. Picard and S. Sclaroff. Photobook: content-based manipulation of image databases. *International Journal of computer vision*, 18:233-254, 1996.

[16] Regularization Tools. ftp://ftp.mathworks.com/pub/contrib/v4/linalg/regtools/.

[17] H. D. Simon. The Lanczos algorithm for solving symmetric linear systems. Ph.D. Dissertation, Dept. of Mathematics, University of California, Berkeley, 1982.

[18] SVDPACK, http://www.netlib.org/svdpack/index.html.

[19] N.E. Troje and T. Vetter. Representation of human faces. Technical Report No. 41, Max-Planck-Institute für biologische Kybernetik, Tübingen, Germany, 1996.

[20] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3:71-86, 1991.

[21] G. Xu and T. Kailath. Fast Estimation of Principal Eigenspace Using Lanczos Algorithm. *SIAM Journal on Matrix Analysis and Applications*, 15:974–994, 1994.

ERNEST ORLANDO LAWRENCE BERKELEY NATIONAL LABORATORY
ONE CYCLOTRON ROAD | BERKELEY, CALIFORNIA 94720

Prepared for the U.S. Department of Energy under Contract No. DE-AC03-76SF00098