An Implicitly Restarted Bidiagonal Lanczos Method for Large-Scale Singular Value Problems

Xiaohui Wang and Hongyuan Zha

Computing Sciences Directorate
National Energy Research
Scientific Computing Division

August 1998

DISCLAIMER

An Implicitly Restarted Bidiagonal Lanczos Method for Large-scale Singular Value Problems

Xiaohui Wang and Hongyuan Zha
Department of Computer Science and Engineering,
The Pennsylvania State University,
State College, PA, 16802, U.S.A
E-mail: {xwang, zha}@cse.psu.edu

Low rank approximation of large and/or sparse rectangular matrices is a very important topic in many application problems and is closely related to the singular value decomposition of the matrices. In this paper, we propose an implicit restart scheme for the bidiagonal Lanczos algorithm to compute a subset of the dominating singular triplets. We also illustrate the connection of the method with inverse eigenvalue problems. In the Lanczos process, we use the so-called one-sided reorthogonalization strategy to maintain the orthogonality level of the Lanczos vectors. The efficiency and the applicability of our algorithm are illustrated by some numerical examples from information retrieval applications.

1 Introduction

Low rank approximation of a large, sparse matrix $A \in \mathcal{R}^{m \times n}$ is very important in many applications as discussed by H. Simon and H.Zha [10]. Generally A is a rectangular matrix with $m \gg n$ or $m \ll n$. Suppose the rank of A is r, then the *singular value decomposition* (SVD) of A is written as:

$$A = U \Sigma V^T, \quad \Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_{\min(m,n)}),$$

where $\sigma_1 \geq \ldots \geq \sigma_r \geq \sigma_{r+1} = \ldots = \sigma_{\min(m,n)} = 0$, and $U \in \mathcal{R}^{m \times n}$, $V \in \mathcal{R}^{n \times n}$, $U^T U = V^T V = I_n$. The singular values of A are defined as the diagonal elements of Σ. The problem we discuss here is:

Problem: Given a large and/or sparse matrix A, determine the k-largest singular triplets of A, i.e., the best rank-k approximation of A.

In this paper, we explore one approach that can be used to compute the dominating k singular triplets. The algorithm we discuss here is bidiagonal Lanczos reduction which has been use to compute a few dominating singular triplets of large sparse matrices as in [2,5]. Our purpose here is to propose a new strategy to restart the bidiagonal Lanczos reduction. Compared to the restarted bidiagonal Lanczos proposed by Björck *etc* [3], our algorithm does not depend on the shifted QR decomposition. Instead, we first truncate the Lanczos reduction and then restore the truncated equations into the standard form of the bidiagonal Lanczos reduction by a procedure based on the idea we developed in [12].

The rest of this paper is organized as follows: Section 2 gives a brief review of the bidiagonal Lanczos reduction. Our restart algorithm and its implementation are given in section 3. Reorthogonalization of the Lanczos vectors and the stopping criteria are discussed in section 4. In section 5, some numerical results are provided and the conclusion remark will be given in section 6.

2 The Bidiagonal Lanczos Process

Given a rectangular matrix $A \in \mathcal{R}^{m \times n}$, without loss of generality, we may assume $m \geq n$. Briefly speaking, the bidiagonal Lanczos can be described as follows[2,5,10]: Given $b \in \mathcal{R}^m$ as the starting vector, in matrix form, this process can be written as:

$$U_k(\beta_1 e_1) = b,$$
$$AV_k = U_k B_k + \beta_{k+1} u_{k+1} e_k^T,$$
$$A^T U_k = V_k B_k^T,$$

Here, $U_k = (u_1, u_2, \ldots, u_k), V_k = (v_1, v_2, \ldots, v_k)$, $B_k = bidiag \begin{pmatrix} \alpha_1 & \alpha_2 & \ldots & \alpha_k \\ \beta_2 & \beta_3 & \ldots & \beta_k \end{pmatrix}$ and $r_k = \beta_{k+1} u_{k+1}$ satisfy $U_k^T U_k = V_k^T V_k = I_k$ and $U_k^T r_k = 0$.

After we get the bidiagonal Lanczos reduction B_k of A, we can use the singular values of B_k to approximate those of A. This can be seen as follows: Let (t, σ, s) be a singular triplet of B_k, we have

$$B_k s = \sigma t, \quad B_k^T t = \sigma s.$$

Let $s = (s_1, s_2, \ldots, s_k)^T$ and denote by $\tilde{u} \equiv U_k t, \tilde{v} \equiv V_k s$, $(\tilde{u}, \sigma, \tilde{v})$ can be used to approximate a singular triplet of A and the accuracy is

$$tol = \|A\tilde{v} - \sigma\tilde{u}\| = \|r_k\| \cdot |s_k|,$$

since $A^T\tilde{u} - \sigma\tilde{v} = 0$. If tol is sufficiently small, then $(\tilde{u}, \sigma, \tilde{v})$ can be accepted as an a singular triplet of A.

3 A New Implicitly Restarted Algorithm

As we know in the case of computing the eigenvalues by Lanczos reduction, if we can perform the bidiagonal Lanczos reduction for sufficiently many steps, we can get a good approximation of the dominating singular triplets. But here we face the following problems: (i) The effort to maintain the orthogonality level of the Lanczos vectors; (ii) The storage for the Lanczos vectors and (iii) The expense for the computation of the SVD of B_k. To avoid these problems, restart is necessary. As we know for eigenproblem, implicit restart is prefered to explicit one, the same thing is true for singular value problem. The main point for the preference of implicitly restart scheme is that many steps of matrix-vector multiplication can be saved,and we know when the size of the matrix is large, the matrix-vector multiplication is the heavy burden of the Lanczos reduction.

Inspired by the work of D. Sorenson [11] and Björck *etc* [3], we want to extend our idea in [12] to the case of singular value problem. Our algorithm can be described as follows:

Implictly Restarted Bidiagonal Lanczos Algorithm(IRBL)

1. For a given starting vector b, perform $k+p$ steps of bidiagonal Lanczos reduction on A to get matrices $U_{k+p}, V_{k+p}, B_{k+p}$ and a vector r_{k+p} such that

$$AV_{k+p} = U_{k+p}B_{k+p} + r_{k+p}e_{k+p}^T, \quad A^TU_{k+p} = V_{k+p}B_{k+p}^T,$$

2. Until convergence

   (a) Compute the k dominating singular triplets $\{T_k, \Theta_k, S_k\}$ of B_{k+p}, here $\Theta_k = diag(\theta_1, \cdots, \theta_k)$.

   (b) Denote by $s = S_k^T e_{k+p}$, compute two orthogonal matrices Q_1, Q_2 and a low bidiagonal matrix $\hat{B}_k$ of size $k \times k$ such that

$$Q_1^T s = \|s\|e_k, \quad \Theta_k Q_1 = Q_2\hat{B}_k, \quad \Theta_k Q_2 = Q_1\hat{B}_k^T.$$

   (c) Let $\hat{U}_k = U_kT_kQ_2$, $\hat{V}_k = V_kS_kQ_1$, $\hat{r}_k = \|s\|r_{k+p}$, we get the following new length-k bidiagonal Lanczos reduction

$$A\hat{V}_k = \hat{U}_k\hat{B}_k + \hat{r}_k e_k^T, \quad A^T\hat{U}_k = \hat{V}_k\hat{B}_k^T,$$

   and extend it to length-$(k+p)$ bidiagonal Lanczos reduction.

In the above algorithm, the main problem here is how to perform step 2(b). In the early restart algorithm by Björck *etc* [3], this step is realized by bulge chasing using the shifted QR-decomposition on the matrix B_{k+p}, and after the truncation, one step of the Lanczos reduction was lost, resulting in relatively high complexity. Here we want to avoid these problems by using the idea we developed in [12] for eigenproblem. As we can see later, our algorithm is straight forward and easy to implement. Here two ways to achieve this point are proposed. The first way is a direct method and can be carried out as below, we call it the **inverse Lanczos** process. Suppose we start with vector s and a matrix B of dimension $k \times k$. Denoted by $Q_1 = [q_1, q_2, \ldots, q_k]$, $Q_2 = [\hat{q}_1, \hat{q}_2, \ldots, \hat{q}_k]$ and $\hat{B} = bidiag\begin{pmatrix} \alpha_1 & \alpha_2 & \ldots & \alpha_k \\ \beta_2 & \beta_3 & \ldots & \beta_k \end{pmatrix}$. This algorithm can be described as:

Inverse Lanczos Algorithm

Given nonzero vector s and a matrix B :

$q_k = s/||s||_2, \qquad \alpha_k \hat{q}_k = Bq_k.$

for $i = k-1$ to 1

$\beta_{i+1} q_i = B^T \hat{q}_{i+1} - \alpha_{i+1} q_{i+1}.$

$\alpha_i \hat{q}_i = Bq_i - \beta_{i+1} \hat{q}_{i+1}.$

end;

Instead of using the inverse Lanczos algorithm as above, another way is to use the one way chasing scheme introduced in [1]. Here we want to use its generalization by H.Zha in [13] to transform the matrix of the form $\begin{pmatrix} \Theta_k \\ \hat{s}^T \end{pmatrix}$ into the form of bidiagonal matrix. The detail of this process can be found in [13]. One important observation of this process is that the last row of the matrix is never touched during the chasing consisting of Givens rotations as pointed out in [13].

Now let's see how the one way chasing scheme works. After the truncation, we have:

$$A\tilde{V}_k = \tilde{U}_k \Theta_k + r_k s^T, \quad A^T \tilde{U}_k = \tilde{V}_k \Theta_k.$$

Taking $u_{k+1} = r_k/||r_k||$ and $\hat{s} = ||r_k|| s = (s_1, s_2, \ldots, s_k)^T$, the first equation can be written as:

$$A\tilde{V}_k = \tilde{U}_{k+1} \tilde{B}_{k+1},$$

where $\tilde{U}_{k+1} = (u_1, u_2, \ldots, u_{k+1})$ and $\tilde{B}_{k+1} = \begin{pmatrix} \Theta_k \\ \hat{s}^T \end{pmatrix}$. The one way chasing consists of a sequence of Givens rotations $\{S_i\}_{i=1}^{s}$ and $\{T_j\}_{j=1}^{t}$ such that $\hat{B}_{k+1} = (S_1 \cdots S_s)^T \tilde{B}_{k+1} (T_1 \cdots T_t)$ is bidiagonal.

Denoted by $Q_1 = S_1 \cdots S_s$ and $Q_2 = T_1 \cdots T_t$, the first equation can be written as:

$$A\tilde{V}_k = \tilde{U}_{k+1} Q_1 \hat{B}_{k+1} Q_2^T,$$

i.e., $A\tilde{V}_k Q_2 = \tilde{U}_{k+1} Q_1 \hat{B}_{k+1}$ and $\hat{B}_{k+1}$ has the form:

$$\hat{B}_{k+1} = bidiag \begin{pmatrix} \alpha_1 \ \alpha_2 \ \ldots \ \alpha_k \\ \beta_2 \ \beta_3 \ \ldots \ \beta_{k+1} \end{pmatrix} = \begin{pmatrix} B_k \\ \beta_{k+1} e_k^T \end{pmatrix}.$$

As we mentioned above, the Givens rotations corresponding to S_i never touch the last row of the matrix $\tilde{B}_{k+1}$, thus each S_i has the form: $S_i = \begin{pmatrix} \tilde{S}_i & \\ & 1 \end{pmatrix}$. So Q_1 can be written as $\begin{pmatrix} \tilde{Q}_1 & \\ & 1 \end{pmatrix}$. These observation results in:

$$A(\tilde{V}_k Q_2) = (\tilde{U}_k \tilde{Q}_1) B_k + \beta_{k+1} u_{k+1} e_k^T.$$

If we take the new V_k and U_k as $V_k = \tilde{V}_k Q_2$ and $U_k = \tilde{U}_k \tilde{Q}_1$, we have

$$AV_k = U_k B_k + \beta_{k+1} u_{k+1} e_k^T,$$

The first equation of the bidiagonal Lanczos reduction is restored.

For the second equation $A^T \tilde{U}_k = \tilde{V}_k \Theta_k$. Multiplying both sides by $\tilde{Q}_1$, we get

$$A^T U_k = \tilde{V}_k \Theta_k \tilde{Q}_1,$$

From $\hat{B}_{k+1} = Q_1^T \tilde{B}_{k+1} Q_2$, we see $\tilde{B}_{k+1}^T Q_1 = Q_2 \hat{B}_{k+1}^T$, i.e.,

$$(\Theta_k, \hat{s}) \begin{pmatrix} \tilde{Q}_1 & \\ & 1 \end{pmatrix} = Q_2 (B_k^T, \beta_{k+1} e_k).$$

This implies

$$\Theta_k \tilde{Q}_1 = Q_2 B_k,$$

Thus the second equation after the truncation is reduced to:

$$A^T U_k = \tilde{V}_k \Theta_k \tilde{Q}_1 = \tilde{V}_k Q_2 B_k = V_k B_k.$$

By now, we have restored both of the truncated equations to the standard form of the bidiagonal Lanczos reduction.

4 Reorthogonalization of the Lanczos Vectors and Stopping Criteria

As we know, in finite precision arithematic, the orthogonality of the Lanczos vectors get lost very quickly if no reorthogonalization is performed. This phenomenum results in the appearance of spurious singular triplet. To avoid this, reorthogonalization of the Lanczos vectors during the bidiagonal Lanczos reduction are necessary. As is pointed out by H. Simon and H.Zha [10], the orthogonality level of the left and right Lanczos vectors are closely related. This can be seen by the following proposition. Denote by

$$\eta(U_k) \equiv \|I_k - U_k^T U_k\|_2, \quad \eta(V_k) \equiv \|I_k - V_k^T V_k\|_2.$$

Then we have:

Proposition: If B_k is nonsingular, then

$$\eta(U_k) \leq \|B_k^{-1}\|_2 \cdot \|\tilde{B}_k\|_2 \cdot \eta(U_{k+1}) + O(\|B_k^{-1}\|_2 \cdot \|A\|_F \cdot \epsilon_M),$$

$$\eta(U_{k+1}) \leq \frac{\|B_k\|_2 \cdot \eta(V_k)}{2\sigma_{min}(\tilde{B}_k)} + O(\frac{\|A\|_F \cdot \epsilon_M}{\sigma_{min}(\tilde{B}_k)}).$$

Here, ϵ_M is the precision of the machine and $\tilde{B}_k = \begin{pmatrix} B_k \\ \beta_{k+1} e_k^T \end{pmatrix}$.

The proof of this proposition can be found in [10] and we omit it here. From this proposition, we see if we can maintain the orthogonality level of the columns of either U_k or V_k under some level, the orthogonality level of the other one should be tolerable and the analysis in [10] shows this point. This is the so-called one-sided reorthogonalization of the Lanczos vectors. Here we also use this approach to treat the orthogonality level of the Lanczos vectors.

As we know form tridiagonal Lanczos reduction for symmetric matrix, keeping the orthogonality level of U_k and V_k to full machine precision is not necessary, thus we can use the semi-orthogonality among U_k or V_k to control the orthogonality level. Our discussion is based on the recurrence formula in [9,10] and we sketch it here.

Since we are using one-sided reorthogonalization technique, we want to control the orthogonality level of either U_k or V_k. Without loss of generality, let's consider V_k. Define $\omega_{ij} = u_i^T u_j = u_j^T u_i$, $\delta_{ij} = v_i^T v_j = v_j^T v_i$, $\epsilon_l = \epsilon_M \sqrt{m}$, $\epsilon_r = \epsilon_M \sqrt{n}$, then we have [10],

$$\omega_{ii} = \delta_{ii} = 1,$$
$$\omega_{i+1,k} = (\alpha_k \delta_{ik} + \beta_k \delta_{i,k-1} - \alpha_i \omega_{ik} + \epsilon_l)/\beta_{i+1}, \quad \text{for} \quad k = 1, 2, \ldots, i$$
$$\delta_{i+1,k} = (\beta_{k+1} \omega_{i+1,k+1} + \alpha_k \omega_{i+1,k} - \beta_{i+1} \delta_{ik} + \epsilon_r)/\alpha_{i+1},$$

where $j = 1, 2 \ldots, i$ and $\delta_{i0} = 0$. Thus the one-sided reorthogonalization can be described as following:

Algorithm One-side Semi-Orth

Using bidiagonal Lanczos recurrence to compute
$u_{i+1}, v_{i+1}, \alpha_{i+1}, \beta_{i+1}$ and do
Update the $\omega - \delta$ recurrence as above.
Let $\xi_{i+1} = (\delta_1, \delta_2, \ldots, \delta_{i+1})$.
If $||\xi_{i+1}||_2 > \sqrt{\epsilon_M}$ then do
Orthogonalize v_{i+1} against V_i .
Reset the $\omega - \delta$ recurrence with $\delta_{i+1,j} = \epsilon_r$, $j = 1, 2, \ldots, i$.
end

In the above algorithm, we use the $||\xi_{i+1}||_2$ as the criteria to decide whether to perform reorthogonalization of the Lanczos vectors instead of the infinity-norm as is used in [10] If the reorthogonalization of v_{i+1} with respect to the previous Lanczos vectors is needed, we use Graham-Schmidt procedure to achieve it.

About the stopping criteria, once we have k approximate singular triplets (U_k, Σ_k, V_k), we can use the difference of AV_k and $U_k\Sigma_k$ to control the accuracy since it's always true that $A^T U_k = V_k \Sigma_k$ from our procedure. This value can be obtained from the process of the bidiagonal Lanczos reduction without extra computation.

Table 1: The largest 10 singular values of MEDLINE by IRBL and IRL

Number of singer value	IRBL	IRL
1	6.210792091961386	6.210792091961375
2	3.980420006133190	3.980420006133188
3	3.383805043683148	3.383805043683148
4	3.078064428388271	3.078064428388259
5	2.998105596001940	2.998105596001936
6	2.762547312516954	2.762547312516948
7	2.707227682527872	2.707227682527876
8	2.627467647986986	2.627467647986989
9	2.463307556859210	2.463307556859209
10	2.435207038023210	2.435207038023206

5 Numerical Results

In this section we report some numerical results based on the above algorithm. We test three matrices from the information retrieval, the matrices MEDLINE and CRANFIELD that can found in Cornell Smart system[4] and the last one is provided by O. Marques[7]. For the first two matrices, we compare our algorithm with the method of computing the eigenvalues of A^TA by implicitly restarted Lanczos(IRL) method. The results show that the restarted bidiagonal Lanczos reduction has almost the same convergent history as the IRL method. In these examples, we set the error tolerance to be 10^{-10} and we use the one side reorthogonalization to mauitain the orthogonal level of the Lanczos vectors. For each method, we tabulate the singular values that are computed by this two methods. All the computations are done on Sun Ultra I workstation using MATLAB 5.0 except exapmle 3.

EXAMPLE 1. We use the MEDLINE text collection from [4]. This is a term-document matrix of order 3681×1033. To this matrix, we are trying to compute 10 dominating singular triplets. The maximal length of the Lanczos reduction for both the bidiagonal Lanczos and the IRL is 20, and we always keep 10 singular triplets after the truncation. The results from these two algorithms are summarized in table 1:

For both method, the numbers of restart are the same, both of them need 6 steps of restart to achieve the accurancy 10^{-10} and the final error of these two methods are $2.8364417 \times 10^{-12}$ for IRBL and $1.7380783 \times 10^{-12}$ for IRL.

EXAMPLE 2. In this experiment, we perform the same work as above to CRANFIELD collection from[4]. This is also a term-document matrix of order 2331×1400 and this time we want to find 5 dominating singular triplets. In this case, we perform 10 steps of Lanczos reduction before we truncate it and we always keep 5 singular trilpets when we restart the IRBL. All these singular

Table 2: The largest 5 singular values of CRANKFIELD by IRBL and IRL

Number of singer value	IRBL	IRL
1	13.67540981336799	13.67540981336800
2	5.89221597986800	5.89221597986801
3	5.64398483359925	5.64398483359926
4	4.81863954716185	4.81863954716184
5	4.25876402496324	4.25876402496324

Table 3: Time distribution for the computation of 180 singular values

	Matrix Vector prod.	Reorth	Ritz
Time	$2.77E+02$	$1.06E+03$	$2.37E+2$
percentage	16.8%	64.4%	14.4%

values for these two methods can be found in table 2.

The steps of restart for both methods are 2 and the corresponging error after 2 steps of restart are $5.3803216 \times 10^{-14}$ and $2.5472307 \times 10^{-13}$, respectively.

EXAMPLE 3. (PRELIMINARY) The matrix we encounter here has the size 846968×96300 and was provided by O.Marques [7]. The number of nonzeros of this matrix is 28587210. It is also a term-document matrix. This part of the computation was performed on Cray-T3E at NERSC, Lawrence Berkeley National Lab. In this example, we are using the BLZPACK package which was implemented by O. Marques to compute as many singular values as possible. If we denote this matrix by A, what we did here was to compute the a subset of the largest eigenpairs of $A^T A$ since A is a thin matrix, then we used these data to compute the singular triplets of A. At this time, the maximal number of dominating singular triplets that can be computed is 180 and the distribution of all these 180 singular values are recorded in 1. The maximal step of Lanczos reduction we use here is 300. The number of restart steps is 3. As we can see from table 3 where the time of the running is recorded, in the case of large sparse matrix and if we have to compute a moderate large number of dominating singular values, most of the time is spent on the reorthogonalization of the Lanczos vectors. The implementation of our algorithm on Cray-T3E is still under way and we hope we can report the result from our algorithm in the near future.

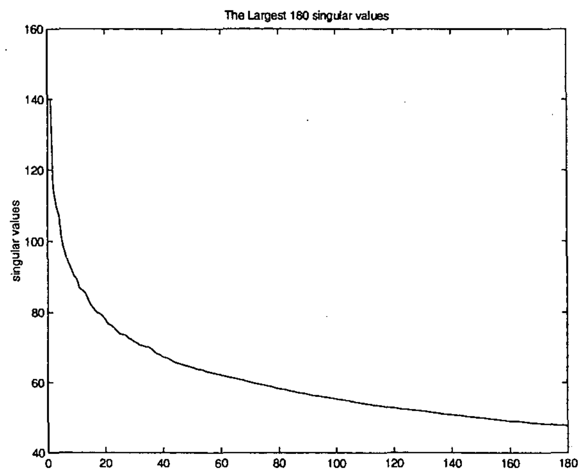

Figure 1: 180 Singular value distribution of the 846968×96300 term-document matrix

6 Concluding Remarks and Future Work

In this paper, we studied the implicitly restarted bidiagonal Lanczos reduction to compute a subset of the dominating singular triplets of a rectangle matrix. We proposed a direct and effective method that can be used to restore the standard form of bidiagonal Lanczos reduction after we truncated the bidiagonal Lanczos reduction. Compared with the traditional restarted scheme that made use of the QR decomposition, our scheme is easier to implement and the computational complexity can be maintained at least at the same level. In the future, we plan to use the restart strategy in this paper to compute a small number(about 5) of dominating singular triplets of a hypertext link structure matrix which has the size of 20 million by 20 million.

Acknowledgement: This work was partly supported by NSF grant CCR-9619452 and by the the Director, Office of Energy Research, Office of Laboratory Policy and Infrastructure Management, of the U.S.Department of Energy under Contract No. DE-AC03-76SF00098 and part of the work was finished when the authors visited the Lawrence Berkeley National Laboratory during the summer 98. The authors thank John Wu and Osni Marques of NERSC, Lawrence Berkeley National Laboratory for many helpful assiatance for the preparation of the Numerical results.

References

1. A. Abdallah and Y. Hu ,**Parallel VLSI computing array implementation for signal subspace updating algorithm** , IEEE Trans. Speech, signal Processing ASSP 37(1989), 742 - 748.
2. M. Berry,**Large Scale singular value computations,** International J. of Supercomputer Application, 6(1992), 13 - 49.
3. A. Björck, E. Grimme and P.V. Dooren,**An implicitly shift bidiagonalization algorithm for ill-posed systems,** BIT 34(1994), 510 - 534.
4. Cornell SMART system,**ftp : // ftp.cs.cornell.edu/pub/smart.**
5. J. Cullum, R.A. Willoughby and M. Lake, **A Lanczos algorithm for computing singular values and vectors of large matrices,** SIAM J. Sci. Stat. Comput. 4(1983), 195 - 215.
6. G.H, Golub and C.F. Van Loan, **Matrix Computations,** The John Hopkins University Press, Baltimore, second edition 1997.
7. O. Marques, private communication, 1998
8. B. N. Parlett,**The Symmetric Eigenvalue Problem,** Classics in Applied Mathematics, SIAM 1998
9. H. D. Simon,**Analysis of the symmetric Lanczos algorithm with reorthogonalization methods,** Lin. Alg. Appl. , 61(1984), 103 - 131.
10. H. D. Simon and H. Zha,**Low rank Matrix approximation using the Lanczos bidiagonalization Process,** Technique Report CSE - 97 - 008, Department of Computer Science and Engineering, The Penn State University, 1997.
11. D. Sorenson,**Implicitly Application of polynomial filters in a k-step Arnoldi method,** SIAM J. Matrix Anal. Appl. 13(1992), 357 - 385.
12. X. Wang and H. Zha,**A New Implementation of Implicitly Restart Arnoldi/Lanczos Method,** in prepararion.
13. H. Zha, **A Two-Way chasing scheme for reducing a symmetric matrix to tridiagonal form,** J. Numer. Lin. Alg. Appl. 1(1992), 49 - 57.

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