

CAM 1146

An efficient nonsymmetric Lanczos method on parallel vector computers *

S.K. Kim and A.T. Chronopoulos

Department of Computer Science, University of Minnesota, Minneapolis, MN 55455, United States

Received 18 January 1991

Revised 27 July 1991

Abstract

Kim, S.K. and A.T. Chronopoulos, An efficient nonsymmetric Lanczos method on parallel vector computers, *Journal of Computational and Applied Mathematics* 42 (1992) 357–374.

In this paper, we introduce the s -step biorthogonal Lanczos method for finding a few eigenvalues of a large sparse nonsymmetric matrix, and we prove that the s -step method generates reduction matrices which are similar to reduction matrices generated by the standard method. We prove that the breakdown conditions of the s -step method are less stringent than the standard one. One iteration of the s -step biorthogonal Lanczos algorithm corresponds to s iterations of the standard biorthogonal Lanczos algorithm, and the s -step method has improved data locality and minimized global communication and superior parallel properties to the standard one on parallel machines (Chronopoulos and Gear (1989) and Kim and Chronopoulos (1991)). We implement the s -step biorthogonal Lanczos method on the CRAY-2 super computer and discuss the breakdown conditions and demonstrate the superior performance of the s -step method to the standard one.

Keywords: Biorthogonal; Lanczos; s -step; breakdown conditions reduction; parallel.

1. Introduction

In many cases, scientific and engineering problems require the computation of a small number of eigenvalues of large sparse nonsymmetric matrices. Earlier work on Krylov methods for nonsymmetric problems focused on variants of the Arnoldi method. The Arnoldi algorithm generates a single set of vectors and forces their orthogonality by explicitly orthogonalizing each new vector generated to all preceding Arnoldi vectors [10,12]. One difficulty with the Arnoldi procedure is that it allows only the computation of the right eigenvectors. Ruhe considered the extension of the one-sided Arnoldi procedure to the two-sided Arnoldi procedure [9]. That extension generates two orthogonal sets of vectors, essentially independently.

Correspondence to: Prof. A.T. Chronopoulos, Department of Computer Science, University of Minnesota, 4-192 EE/CSci Building, 200 Union Street S.E., Minneapolis, MN 55455, United States.

* This research was supported by NSF grant CCR-8722260. The Minnesota Supercomputing Institute provided the CRAY-2 time.

The biorthogonal Lanczos method generates two sets of vectors that are biorthogonal, and a sequence of nonsymmetric but tridiagonal Lanczos matrices. We will refer to these tridiagonal matrices as *Lanczos reduction matrices*. The Lanczos procedure has modest storage requirements and therefore can be used on very large matrices. In such procedures the eigenvalue and eigenvector computations are performed separately. The biorthogonal Lanczos method was neglected for a long time because it faces serious breakdown problems. The problem of building the Lanczos vectors in the nonsymmetric case was addressed in [8], where a lookahead Lanczos algorithm which handles possible breakdown was suggested.

The biorthogonal Lanczos method has two synchronization points due to inner product computations at each iteration step. This introduces main memory contention on a shared-memory machine, and a large amount of interprocessor communication overhead in a distributed-memory parallel machine. In this paper we introduce the s -step biorthogonal Lanczos method. In one iteration of an s -step method, $2s$ new directions are formed simultaneously based on the vectors $\{q_i, Aq_i, \dots, A^{s-1}q_i\}$, $\{p_i, (A^T)p_i, \dots, (A^T)^{s-1}p_i\}$ and the preceding direction vectors. This means, for example, that the inner products (needed for s steps of the standard method) can be performed simultaneously and the vector updates are replaced by linear combinations. We prove that the s -step biorthogonal Lanczos method like the lookahead Lanczos method introduced in [8] has fewer breakdown conditions than the standard one. It should be noted that the s -step Lanczos method is different from the block Lanczos method [5], which is used to find multiple eigenvalues. The block Lanczos method uses a block of vectors instead of a single vector in the standard Lanczos method; then the modification consists of replacing of all the operations with single vector by operations with blocks of vectors. The s -step Lanczos method is a version of the standard Lanczos method that has as few as synchronization points as possible.

In Section 2 we review the biorthogonal Lanczos procedure and its properties. In Sections 3 and 4 we derive the s -step biorthogonal Lanczos method and prove that the s -step method has fewer breakdown conditions than the standard one. In Section 5 we compare the numerical implementation of the standard and s -step methods on the CRAY-2 super computer.

2. The biorthogonal Lanczos method

Let A be an $n \times n$ nonsymmetric matrix. There are many tridiagonal matrices similar to A and T_n is one of them. Then for some matrix $Q_n = (q_1, \dots, q_n)$ we have

$$Q_n^{-1}AQ_n = T_n. \quad (2.1)$$

Let $P_n = (p_1, \dots, p_n)$ and replace (2.1) by two separate relations

$$P_n^T Q_n = I, \quad (2.2)$$

$$P_n^T A Q_n = T_n. \quad (2.3)$$

By equating columns on each side of $AQ_n = Q_n T_n$ and $P_n^T A = T_n P_n^T$ in the natural increasing order, we obtain the following equations. For each $j < n$,

$$AQ_j = Q_j T_j + r_j e_j^T, \quad (2.4)$$

$$P_j^T A = T_j P_j^T + e_j s_j^T, \quad (2.5)$$

where r_j, s_j are residual vectors after the j th iteration. A and two initial vectors p_1, q_1 essentially determine all the other elements of P_j, Q_j and T_j . In this method, the right space Q_j is a Krylov subspace $Q_j = \text{span}[q_1, Aq_1, \dots, A^{j-1}q_1]$ and the left space P_j is a Krylov subspace $P_j = \text{span}[p_1, A^T p_1, \dots, (A^T)^{j-1}p_1]$ and $P_j^T Q_j = I_j$.

The eigenvalues of the biorthogonal Lanczos matrices T_j are called *Petrov values* (or *Ritz values*) of A in Q_j . For many matrices and for relatively small j , compared to n , several of the extreme eigenvalues of A are well approximated by the corresponding Ritz values. The right Ritz vector $Q_j y (= z)$ obtained from a right eigenvector y of a given T_j is an approximation to a corresponding right eigenvector of A , and the left Ritz vector $P_j \hat{y} (= \hat{z})$ obtained from a left eigenvector \hat{y} of a given T_j is an approximation to a corresponding left eigenvector of A . A simple version of the biorthogonal Lanczos algorithm can be formulated as follows.

Algorithm 2.1. The biorthogonal Lanczos algorithm.

$q_0 = 0, p_0 = 0$ and $\beta_0 = 0, \gamma_0 = 0$

Choose q_1 and p_1 with $(p_1, q_1) = 1$

For $j = 1$ **until** Convergence **Do**

(1) Compute and store $Aq_j, p_j^T A$

(2) $\alpha_j = (Aq_j, p_j)$

(3) $r_j = Aq_j - \gamma_{j-1}q_{j-1} - \alpha_j q_j$
 $s_j^T = p_j^T A - \beta_{j-1}p_{j-1}^T - \alpha_j p_j^T$

(4) $\beta_j \gamma_j = (r_j, s_j)$

(5) $q_{j+1} = r_j / \beta_j$

$p_{j+1} = s_j / \gamma_j$

EndFor

Note that $p_j^T A = (A^T p_j)^T$ requires multiplication by the transpose of A .

Let T_j be the tridiagonal matrix at step j :

$$T_j = \begin{bmatrix} \alpha_1 & \gamma_1 & & & & & & \\ \beta_1 & \alpha_2 & \gamma_2 & & & & & \\ & \ddots & \ddots & \ddots & & & & \\ & & & & & & \gamma_{j-1} & \\ & & & & & & & \alpha_j \end{bmatrix}.$$

For each j , the nonsymmetric Lanczos matrix T_j is the biorthogonal projection of A onto the Krylov subspaces spanned by the Q_j and T_j^T is the biorthogonal projection of A^T onto P_j . The coefficients β_j and γ_j in T_j are not uniquely defined by step (4) of Algorithm 2.1. Q_j and P_j are biorthogonal with any choices of β_j, γ_j which satisfy equation $\beta_j \gamma_j = (r_j, s_j)$. One possible choice is $\beta_j = \sqrt{|(r_j, s_j)|}$ and $\gamma_j = \text{sign}(r_j, s_j) \beta_j$ [8]. Observe that the continuation of this recursion requires that $(r_j, s_j) \neq 0$ for any j . $(r_j, s_j) = 0$ causes the algorithm to break down. That is why the biorthogonal Lanczos method has not been widely used. This problem does not occur in the symmetric Lanczos method.

The simplest a posteriori bound on the accuracy of a Ritz value λ_j is obtained from the residual norm of the associated Ritz vectors. The residual norms of the Ritz value λ_j and the

right Ritz vector z_j can be computed by using the formula $\|(A - \lambda_i I)z_i\| = \|r_j\| |\bar{s}_{ji}|$, for $i = 1, \dots, j$, where \bar{s}_{ji} is the last element of the i th right eigenvector of T_j , and the residual norms of the Ritz value λ and the left Ritz vector \hat{z} can be computed by using the formula $\|(A^T - \lambda_i I)\hat{z}_i\| = \|s_j\| |\hat{s}_{ji}|$, for $i = 1, \dots, j$, where \hat{s}_{ji} is the last element of the i th left eigenvector of T_j . These error estimates are used as stopping criteria.

Remark 2.2. The vector operations for each iteration are $12N + 2Mv$, and storage requirements are $4N + 1Ms$, where Mv and Ms stand for matrix-vector product and matrix storage of A , respectively. The vectors q_1, \dots, q_j and p_1, \dots, p_j can be kept in secondary storage, because these vectors are only required when the approximate eigenvectors have to be computed.

Next we restructure the biorthogonal Lanczos algorithm. The biorthogonal Lanczos algorithm has three basic types of operations: matrix-vector products, inner products and the vector updates. In Algorithm 2.1 the inner products cannot be performed in parallel, and step (2) (or step (4)) must be completed before the rest of the computations in the same step start. This forces double accesses of vectors $p, q, r, s, Aq, p^T A$ from the main memory at each iteration. Algorithm 2.3 is a variant of Algorithm 2.1 and Algorithm 2.3 is more suitable for parallel processing.

Algorithm 2.3. The restructured biorthogonal Lanczos algorithm.

$q_0 = 0, p_0 = 0$

Choose r_0, s_0 with $(r_0, s_0) \neq 0$

For $j = 0$ **until** Convergence **Do**

(1) Compute and store $Ar_j, s_j^T A$

(2) $\beta_j \gamma_j = (r_j, s_j)$

$\alpha_{j+1} = (Ar_j, s_j) / (r_j, s_j)$

(3) $q_{j+1} = r_j / \beta_j$

$p_{j+1} = s_j / \gamma_j$

(4) $r_{j+1} = Ar_j / \beta_j - \gamma_j q_j - \alpha_{j+1} q_{j+1}$
 $s_{j+1}^T = s_j^T A / \gamma_j - \beta_j p_j^T - \alpha_{j+1} p_{j+1}^T$

EndFor

The biorthogonal vectors q_j, p_j in Algorithm 2.3 are generated in the same way as the standard biorthogonal Lanczos method. Computationally the difference between Algorithms 2.1 and 2.3 is in the computation of α_j, r_j, s_j . In step (4), Ar_j / β_j and $s_j^T A / \gamma_j$ cannot be replaced by Aq_{j+1} and $p_{j+1}^T A$, because steps (3) and (4) are computed at the same time after step (2). Therefore we need two extra vector operations (i.e., scalar times a vector operation), to compute r_j, s_j in step (4) of Algorithm 2.3. However, Algorithm 2.3 seems more promising than Algorithm 2.1 for parallel processing because the two inner products required to advance each iteration can be executed simultaneously. Also, one memory sweep through the data is required to complete each iteration allowing better management of slower memories in a memory hierarchy computer, i.e., the data locality of Algorithm 2.3 is better than that of the standard algorithm.

In the next section we propose an s -step biorthogonal Lanczos algorithm, which executes simultaneously in a certain sense s consecutive steps of Algorithm 2.1.

3. The s -step biorthogonal Lanczos method

Let us denote by k the iteration number in the s -step biorthogonal Lanczos method. We will denote by v and w the s -step biorthogonal Lanczos vectors instead of p and q of the standard biorthogonal Lanczos method. Given the vectors $v_k^1, v_k^2, \dots, v_k^s$ we will use \bar{V}_k (each of dimension N) to denote the matrix of $\{v_k^1, v_k^2, \dots, v_k^s\}$. Given the vectors $w_k^1, w_k^2, \dots, w_k^s$ we will use \bar{W}_k (each of dimension N) to denote the matrix of $\{w_k^1, w_k^2, \dots, w_k^s\}$. The subspace \bar{V}_k is spanned by $\{v_k^1, Av_k^1, \dots, A^{s-1}v_k^1\}$ so that \bar{V}_k is made orthogonal to the subspaces $\bar{W}_{k-1}, \bar{W}_{k-2}, \dots, \bar{W}_1$. Also the subspace \bar{W}_k is spanned by $\{w_k^1, A^T w_k^1, \dots, (A^T)^{s-1}w_k^1\}$, so that \bar{W}_k is orthogonal to vector sets $\bar{V}_{k-1}, \bar{V}_{k-2}, \dots, \bar{V}_1$.

Let the $s \times s$ matrix $\bar{W}_k^T \bar{V}_k$ be nonsingular. Then LU decomposition with row exchanging can be applied to the matrix $\bar{W}_k^T \bar{V}_k$ as follows:

$$\bar{\Pi}_k (\bar{W}_k^T \bar{V}_k) = \bar{L}_k * \bar{U}_k,$$

where $\bar{\Pi}_k$ is a permutation matrix, \bar{L}_k is an $s \times s$ lower triangular matrix and \bar{U}_k is an $s \times s$ upper triangular matrix.

Remark 3.1. Assume that \bar{W}_{i_1} be orthogonal to \bar{V}_{i_2} for $i_1 \neq i_2$. Then $W_k^T V_k$ can be decomposed into $L_k * U_k$ as follows:

$$\Pi_k (W_k^T V_k) = L_k * U_k, \tag{3.1.0}$$

where $L_k = \text{diag}\{\bar{L}_1, \bar{L}_2, \dots, \bar{L}_k\}$, $U_k = \text{diag}\{\bar{U}_1, \bar{U}_2, \dots, \bar{U}_k\}$ and $\Pi_k = \text{diag}\{\bar{\Pi}_1, \bar{\Pi}_2, \dots, \bar{\Pi}_k\}$.

The assumption of this remark will be proved to be true for the vector blocks $\{\bar{V}_1, \dots, \bar{V}_k\}$, $\{\bar{W}_1, \dots, \bar{W}_k\}$ generated in the s -step biorthogonal Lanczos algorithm.

Lemma 3.2. Let \bar{T}_j be a tridiagonal matrix and $\bar{T}_k = U_k^{-1} \bar{T}_j U_k$ for $j = sk$. Then \bar{T}_k is similar to the matrix \bar{T}_j and \bar{T}_k is the following block tridiagonal matrix:

$$\bar{T}_k = \begin{bmatrix} G_1 & E_1 & & & & \\ F_1 & G_2 & E_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & & & E_{k-1} & \\ & & & F_{k-1} & G_k & \end{bmatrix},$$

where G_i and E_i are $s \times s$ matrices. The matrix F_i is an $s \times s$ matrix whose only nonzero element is at location $(1, s)$.

Proof. \bar{U}_k is nonsingular if $\bar{V}_k^H \bar{W}_k$ is nonsingular. By Remark 3.1, U_k is nonsingular if all \bar{U}_i , for $i = 1, \dots, k$, are nonsingular. So \bar{T}_k is similar to the matrix \bar{T}_j . Since $U_k = \text{diag}\{\bar{U}_1, \bar{U}_2, \dots, \bar{U}_k\}$ and \bar{U}_i , for $i = 1, \dots, k$, is an $s \times s$ upper triangular matrix, U_k^{-1} has the same structure as U_k . Thus the product $U_k^{-1} \bar{T}_j U_k$ for $j = sk$ is a block tridiagonal matrix with lower diagonal blocks in a special form. We will demonstrate this for the special case $s = 3, k = 3$. The general case is shown similarly but the description is more complicated:

Lemma 3.3. Let \bar{T}_j be a tridiagonal matrix and $(\hat{T}_k)^\top = L_k^{-1} \bar{T}_j L_k$ for $j = sk$. Then $(\hat{T}_k)^\top$ is similar to the matrix \bar{T}_k and \hat{T}_k is the following block tridiagonal matrix:

$$\hat{T}_k = \begin{bmatrix} \hat{G}_1 & \hat{E}_1 & & & & \\ \hat{F}_1 & \hat{G}_2 & \hat{E}_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & & & \hat{E}_{k-1} & \\ & & & \hat{F}_{k-1} & \hat{G}_k & \end{bmatrix},$$

where \hat{G}_i and \hat{E}_i are $s \times s$ matrices. The matrix \hat{F}_i is an $s \times s$ matrix whose only nonzero element is at location $(1, s)$.

Proof. Same as the proof of Lemma 3.2. \square

We will use the following s -dimensional column notations for the matrices G_i , E_i in the matrix \bar{T}_k and \hat{G}_i , \hat{E}_i in the matrix \hat{T}_k :

$$G_i = [\alpha_i^j], \quad E_i = [\gamma_i^j], \quad \text{for } j = 1, \dots, s,$$

where $\alpha_i^j = [\alpha_i^{1,j}, \dots, \alpha_i^{s,j}]^\top$ and $\gamma_i^j = [\gamma_i^{1,j}, \dots, \gamma_i^{s,j}]^\top$,

$$\hat{G}_i = [\hat{\alpha}_i^j], \quad \hat{E}_i = [\hat{\gamma}_i^j], \quad \text{for } j = 1, \dots, s,$$

where $\hat{\alpha}_i^j = [\hat{\alpha}_i^{1,j}, \dots, \hat{\alpha}_i^{s,j}]^\top$ and $\hat{\gamma}_i^j = [\hat{\gamma}_i^{1,j}, \dots, \hat{\gamma}_i^{s,j}]^\top$.

Theorem 3.4. Let A be an $n \times n$ nonsymmetric matrix and $V_m = (\bar{V}_1, \bar{V}_2, \dots, \bar{V}_m)$, $W_m = (\bar{W}_1, \bar{W}_2, \dots, \bar{W}_m)$, for $n = sm$, are block biorthogonal. Then

$$AV_m = V_m \bar{T}_m, \quad (3.1.1)$$

where \bar{T}_m is the block tridiagonal of Lemma 3.2, and \bar{T}_m is similar to the tridiagonal matrix T_n generated by the biorthogonal Lanczos method.

Proof. By premultiplying by L_m^{-1} and postmultiplying by U_m^{-1} equation (3.1.0) we obtain

$$L_m^{-1} \Pi_m W_m^\top V_m U_m^{-1} = I.$$

Let $P^\top = L_m^{-1} \Pi_m W_m^\top$ and $Q = V_m U_m^{-1}$, i.e., $V_m = QU_m$ and $W_m^\top = \Pi_m L_m P^\top$. Then we have

$$AQ U_m = QU_m \bar{T}_m.$$

By multiplying both sides of the equations by U_m^{-1} , we obtain

$$AQ = QU_m \bar{T}_m U_m^{-1}.$$

By multiplying both sides of the equations by Q^{-1} , we obtain

$$Q^{-1}AQ = U_m \bar{T}_m U_m^{-1}.$$

From Lemma 3.2, $U_m \bar{T}_m U_m^{-1}$ is a tridiagonal matrix. We can replace Q^{-1} by P^\top from $P^\top Q = I$; then we obtain

$$P^\top AQ = U_m \bar{T}_m U_m^{-1}.$$

Then $U_m \bar{T}_m U_m^{-1}$ is the tridiagonal matrix formed by the standard biorthogonal Lanczos method. \square

Theorem 3.5. Let A be an $n \times n$ nonsymmetric matrix and $V_m = \{\bar{V}_1, \bar{V}_2, \dots, \bar{V}_m\}$, $W_m = \{\bar{W}_1, \bar{W}_2, \dots, \bar{W}_m\}$, for $n = sm$, are block biorthogonal. Then

$$A^T W_m = W_m \hat{T}_m, \quad (3.1.2)$$

where \hat{T}_m is the block tridiagonal of Lemma 3.3, and $(\hat{T}_m)^T$ is similar to the biorthogonal tridiagonal matrix T_n .

Proof. Same as the proof of Theorem 3.4. \square

Corollary 3.6. The block tridiagonal matrices \bar{T}_k , $(\hat{T}_k)^T$, for $k = 1, \dots, m$, have the same eigenvalues as the biorthogonal Lanczos reduction matrix T_j , for $j = sk$.

Proof. By Lemma 3.2, the matrices \bar{T}_k , $(\hat{T}_k)^T$ and T_j for $j = sk$ are similar. \square

In the end of this section we prove that the matrix $\bar{W}_k^T \bar{V}_k$ is symmetric and $\bar{T}_k \equiv \hat{T}_k$. If $k < m$, then by equating column blocks in (3.1.1), (3.1.2) we obtain the following equations:

$$A V_k = V_k \bar{T}_k + x_k e_{sk}^T, \quad \text{for } k = 1, \dots, m, \quad (3.2.1)$$

$$A^T W_k = W_k \hat{T}_k + y_k e_{sk}^T, \quad \text{for } k = 1, \dots, m, \quad (3.2.2)$$

where the vectors x_k , y_k are the residual vectors. From (3.2.1), (3.2.2) we derive the following block equations:

$$A \bar{V}_k = \bar{V}_{k-1} E_{k-1} + \bar{V}_k G_k + x_k e_{sk}^T, \quad (3.3.1)$$

$$A^T \bar{W}_k = \bar{W}_{k-1} \hat{E}_{k-1} + \bar{W}_k \hat{G}_k + y_k e_{sk}^T, \quad (3.3.2)$$

where $\bar{V}_0 = 0$, $E_0 = 0$ and $\bar{W}_0 = 0$, $\hat{E}_0 = 0$.

Equations (3.1.1)–(3.3.2) motivate the derivation of an s -step biorthogonal Lanczos algorithm. Let $t_k^j = [t_k^{1j}, \dots, t_k^{sj}]^T$, $\hat{t}_k^j = [\hat{t}_k^{1j}, \dots, \hat{t}_k^{sj}]^T$ denote the parameters in defining v_k^j , w_k^j . We now give the defining equations of the s -step biorthogonal Lanczos method in the form of an algorithm.

Algorithm 3.7. s -step biorthogonal Lanczos algorithm.

$$\bar{V}_0 = 0, \quad \bar{W}_0 = 0, \quad [\gamma_0^i] = 0, \quad [\hat{\gamma}_0^i] = 0, \quad 1 \leq i \leq s$$

$$\bar{V}_1 = [v_1^1, A v_1^1, \dots, A^{s-1} v_1^1], \quad \bar{W}_1 = [w_1^1, A w_1^1, \dots, (A^T)^{s-1} w_1^1]$$

For $k = 1$ **until** Convergence **Do**

Select $[\alpha_k^i]$, $[\gamma_{k-1}^i]$, $1 \leq i \leq s$, to orthogonalize \bar{V}_k against \bar{W}_{k-1} in (3.3.1). Also select $[\hat{\alpha}_k^i]$, $[\hat{\gamma}_{k-1}^i]$, $1 \leq i \leq s$, to orthogonalize \bar{W}_k against \bar{V}_{k-1} in (3.3.2). These give

$$v_{k+1}^1 = A v_k^s - \bar{V}_{k-1} \gamma_{k-1}^s - \bar{V}_k \alpha_k^s, \quad (3.4.1)$$

$$w_{k+1}^1 = A^T w_k^s - \bar{W}_{k-1} \hat{\gamma}_{k-1}^s - \bar{W}_k \hat{\alpha}_k^s. \quad (3.4.2)$$

Select $[t_k^j]$, $2 \leq j \leq s$, to orthogonalize $\{Av_{k+1}^1, \dots, A^{s-1}v_{k+1}^1\}$ against \bar{W}_k , which gives

$$v_{k+1}^j = A^{j-1}v_{k+1}^1 - \bar{V}_k t_k^j, \quad \text{for } j = 2, \dots, s. \quad (3.5.1)$$

Also select $[\hat{t}_k^j]$, $2 \leq j \leq s$, to orthogonalize $\{A^T w_{k+1}^1, \dots, (A^T)^{s-1} w_{k+1}^1\}$ against \bar{V}_k , which gives

$$w_{k+1}^j = (A^T)^{j-1} w_{k+1}^1 - \bar{W}_k \hat{t}_k^j, \quad \text{for } j = 2, \dots, s. \quad (3.5.2)$$

EndFor

The following lemma proves that the assumption in Remark 3.1 holds true.

Lemma 3.8. *In Algorithm 3.7 vector sets V_{k+1} , W_{k+1} are block biorthogonal, i.e., \bar{W}_{i_1} is orthogonal to \bar{V}_{i_2} for $i_1 \neq i_2$.*

Proof. We will prove Lemma 3.8 by induction. It is obvious that V_2, W_2 are block biorthogonal. Suppose that V_k, W_k are block biorthogonal. Then $(\bar{W}_l, A\bar{V}_k) = 0$, for $l = 1, \dots, k-2$. So v_{k+1}^1 in (3.4.1) is orthogonal to \bar{W}_l , for $l = 1, \dots, k-2$. By using this fact, we derive $(\bar{W}_l, A^{j-1}v_{k+1}^1) = 0$, for $l = 1, \dots, k-1$. So v_{k+1}^j , for $j = 1, \dots, s$, in (3.5.1) is orthogonal to \bar{W}_l , for $l = 1, \dots, k-1$. Therefore \bar{V}_{k+1} is orthogonal to $\bar{W}_k, \dots, \bar{W}_1$ in Algorithm 3.7. Similarly \bar{W}_{k+1} is orthogonal to $\bar{V}_k, \dots, \bar{V}_1$. \square

Next, we demonstrate how to determine the parameters $\alpha_k^i, \gamma_{k-1}^i, t_k^i$ and $\hat{\alpha}_k^i, \hat{\gamma}_{k-1}^i, \hat{t}_k^i$ in Algorithm 3.7. Equation (3.3.1) multiplied by \bar{W}_k^T from the left and (3.3.2) multiplied by \bar{V}_k^T from the left yield

$$\bar{W}_k^T A \bar{V}_k = \bar{W}_k^T \bar{V}_k G_k, \quad (3.6.1)$$

$$\bar{V}_k^T A^T \bar{W}_k = \bar{V}_k^T \bar{W}_k \hat{G}_k. \quad (3.6.2)$$

Equation (3.3.1) multiplied by \bar{W}_{k-1}^T from the left and (3.3.2) multiplied by \bar{V}_{k-1}^T from the left yield

$$\bar{W}_{k-1}^T A \bar{V}_k = \bar{W}_{k-1}^T \bar{V}_{k-1} E_{k-1}, \quad (3.7.1)$$

$$\bar{V}_{k-1}^T A^T \bar{W}_k = \bar{V}_{k-1}^T \bar{W}_{k-1} \hat{E}_{k-1}. \quad (3.7.2)$$

Equation (3.5.1) multiplied by \bar{W}_k^T from the left and (3.5.2) multiplied by \bar{V}_k^T from the left yield

$$0 = \bar{W}_k^T A^{j-1} v_{k+1}^1 - \bar{W}_k^T \bar{V}_k t_k^j, \quad \text{for } j = 2, \dots, s, \quad (3.8.1)$$

$$0 = \bar{V}_k^T (A^T)^{j-1} w_{k+1}^1 - \bar{V}_k^T \bar{W}_k \hat{t}_k^j, \quad \text{for } j = 2, \dots, s. \quad (3.8.2)$$

Lemma 3.9. *In equations (3.6.1)–(3.8.2), $G_l \equiv \hat{G}_l$, $E_{l-1} \equiv \hat{E}_{l-1}$ and $t_l^j \equiv \hat{t}_l^j$, $1 \leq l \leq k$, i.e., $\bar{T}_k \equiv \hat{T}_k$ in (3.1.1), (3.1.2).*

Proof. From Algorithm 3.7, we obtain $v_l^j = \bar{v}_l^j(A)v_1^1$, $w_l^j = \bar{w}_l^j(A^T)w_1^1$, for $j = 1, \dots, s$, where $\bar{v}_l^j(A)$, $\bar{w}_l^j(A^T)$ are the polynomials in A, A^T .

By induction we will prove that coefficients of $\tilde{v}_l^j(A)$, $\tilde{w}_l^j(A^T)$ are the same. It is obvious that coefficients of $\tilde{v}_l^j(A)$, $\tilde{w}_l^j(A^T)$, for $j = 1, \dots, s$, are the same in Algorithm 3.7. Suppose that coefficients of $\tilde{v}_l^j(A)$, $\tilde{w}_l^j(A^T)$, for $l \leq k-1$, are the same. Then

$$(w_l^j, v_l^j) = (\tilde{w}_l^j(A^T)w_1^1, \tilde{v}_l^j(A)v_1^1) = (\tilde{v}_l^j(A)v_1^1, \tilde{w}_l^j(A^T)w_1^1) = (v_l^j, w_l^j).$$

This implies that $\bar{W}_l^T \bar{V}_l \equiv \bar{V}_l^T \bar{W}_l$, for $l \leq k-1$. Similarly we can prove that $\bar{W}_l^T A \bar{V}_l \equiv \bar{V}_l^T A \bar{W}_l$, $\bar{W}_{k-1}^T A \bar{V}_k \equiv \bar{V}_{k-1}^T A^T \bar{W}_{k-1}$ and $\bar{W}_k^T A^{j-1} v_{k+1}^1 \equiv \bar{V}_k^T (A^T)^{j-1} w_{k+1}^1$; therefore $G_l \equiv \hat{G}_l$, $E_{l-1} \equiv \hat{E}_{l-1}$ and $t_l^j \equiv \hat{t}_l^j$ for $l \leq k-1$ and $j = 2, \dots, s$. Now we prove the induction step $l = k$. From (3.4.1)–(3.5.2), coefficients of $\tilde{v}_k^j(A)$, $\tilde{w}_k^j(A^T)$ are the same.

This then is used to prove that $\bar{W}_k^T \bar{V}_k \equiv \bar{V}_k^T \bar{W}_k$, also $G_k \equiv \hat{G}_k$, $E_{k-1} \equiv \hat{E}_{k-1}$ and $t_k^j \equiv \hat{t}_k^j$. \square

Equations (3.6.1), (3.7.1) and (3.8.1) determine $[\alpha_k^i]$, $[\gamma_{k-1}^i]$, $1 \leq i \leq s$, and $[t_k^j]$, $2 \leq j \leq s$, as solutions of $3s-1$ linear systems of size s . Let us denote by $\bar{M}_k = \bar{W}_k^T \bar{V}_k$.

Remark 3.10. Let \bar{M}_k be nonsingular. Then from (3.6.1), (3.7.1) and (3.8.1) it follows that the following linear systems must be solved to determine $[\alpha_k^i]$, $[\gamma_{k-1}^i]$, $1 \leq i \leq s$, and $[t_k^j]$, $2 \leq j \leq s$:

$$\bar{M}_k \alpha_k^i = d_k^i, \quad \text{where } d_k^i = [(w_k^1, Av_k^i), \dots, (w_k^s, Av_k^i)]^T, \quad (3.9)$$

$$\bar{M}_{k-1} \gamma_{k-1}^i = c_{k-1}^i, \quad \text{where } c_{k-1}^i = [(w_{k-1}^1, Av_k^i), \dots, (w_{k-1}^s, Av_k^i)]^T, \quad (3.10)$$

$$\bar{M}_k t_k^j = b_k^j, \quad \text{where } b_k^j = [(w_k^1, A^{j-1} v_{k+1}^1), \dots, (w_k^s, A^{j-1} v_{k+1}^1)]^T. \quad (3.11)$$

Using the block biorthogonality of the two vector sets V_k , W_k , the linear systems (3.9)–(3.11) can be determined from previously computed inner products and the following $2s$ inner products:

$$(w_k^1, v_k^1), (w_k^1, Av_k^1), \dots, (w_k^1, A^{2s-1} v_k^1), \quad (3.12)$$

i.e., the only $2s$ inner products (on vectors of size N) which are required in forming \hat{T}_k , \bar{V}_{k+1} and \bar{W}_{k+1} in Algorithm 3.7. Reducing inner products in the s -step biorthogonal Lanczos method is similar to that of the s -step Lanczos method [7].

We now reformulate the s -step biorthogonal Lanczos algorithm taking into account the theory developed above.

Algorithm 3.11. s -step biorthogonal Lanczos algorithm.

Select $v_1^1, w_1^1, [\gamma_0^i] = 0, 1 \leq i \leq s$

Compute $\bar{V}_1 = [v_1^1, Av_1^1, \dots, A^{s-1} v_1^1]$, $\bar{W}_1 = [w_1^1, Aw_1^1, \dots, (A^T)^{s-1} w_1^1]$

Compute $2s$ inner products

For $k = 1$ **until** Convergence **Do**

(1) Decompose \bar{M}_k and solve $\bar{M}_k \alpha_k^i = d_k^i$ and $M_{k-1} \gamma_{k-1}^i = c_{k-1}^i$, for $i = 1, \dots, s$

(2) Compute $v_{k+1}^1 = Av_k^s - \bar{V}_{k-1} \gamma_{k-1}^s - \bar{V}_k \alpha_k^s$, $w_{k+1}^1 = A^T w_k^s - \bar{W}_{k-1} \gamma_{k-1}^s - \bar{W}_k \alpha_k^s$

(3) Compute $Av_{k+1}^1, A^2 v_{k+1}^1, \dots, A^s v_{k+1}^1, A^T w_{k+1}^1, (A^T)^2 w_{k+1}^1, \dots, (A^T)^s w_{k+1}^1$

(4) Compute the $2s$ inner products in (3.12)

(5) Solve $M_k t_k^j = b_k^j$, for $j = 2, \dots, s$

(6) Compute $v_{k+1}^j = A^{j-1} v_{k+1}^1 - \bar{V}_k [t_k^j]$, for $j = 2, \dots, s$, $w_{k+1}^j = (A^T)^{j-1} w_{k+1}^1 - \bar{W}_k [t_k^j]$, for $j = 2, \dots, s$

EndFor

Table 3.1

Vector operations for s iterations of standard method and 1 iteration of s -step methods

Operation	Standard biorthogonal Lanczos algorithm	s -step biorthogonal Lanczos algorithm
Dotproducts	$2s$	$2s$
Vector updates	$10s$	$4s(s+1)$
Matrix*vector	$2s$	$2s+1$

In the s -step biorthogonal Lanczos method, the Ritz values of A in V_k are the eigenvalues λ_k of \bar{T}_k . The right Ritz vectors are vectors $V_k x_k (= z_k)$, where the right eigenvectors x_k of \bar{T}_k associated with the λ_k and the left Ritz vectors are vectors $W_k \hat{x}_k (= \hat{z}_k)$, where the left eigenvectors \hat{x}_k of \bar{T}_k are associated with the λ_k . The residual norms of the Ritz value λ and the right Ritz vector z can be computed by using the formula $\|(A - \lambda_i I)z_i\| = \|x_k^1\| |\bar{s}_{ki}|$, for $i = 1, \dots, sk$, where \bar{s}_{ki} is the last element of the i th right eigenvector of \bar{T}_k , and the residual norms of the Ritz value λ and the left Ritz vector \hat{z} can be computed by using the formula $\|(A^T - \lambda_i I)\hat{z}_i\| = \|y_k^1\| |\hat{s}_{ki}|$, for $i = 1, \dots, sk$, where \hat{s}_{ki} is the last element of the i th left eigenvector of \bar{T}_j .

Remark 3.12. The vector operations for each iteration in the s -step method are $2s(2s+3)N + (2s+1)Mv$, and storage requirements are $2(s+1)N + 1Ms$. We must store v_1^1, \dots, v_k^s and w_1^1, \dots, w_k^s to find eigenvectors after we find eigenvalues in the k th iteration. These vectors can be kept in secondary storage until they are used.

We compare the computational work of the s -step biorthogonal Lanczos method to the standard one. We only present the vector operations on vectors of dimensions N and neglect the operations on vectors of dimension s (see Table 3.1).

4. Decreasing the breakdown conditions

It can be easily shown that when Algorithm 2.1 does not break down for a null inner product (r_j, s_j) , then the vectors q_{j+1} and p_{j+1} satisfy the biorthonormality property. Although there are various ways of choosing β_j, γ_j satisfying $\beta_j \gamma_j = (r_j, s_j)$, it is of interest to notice that the product $\|q_{j+1}\| \|p_{j+1}\|$ will not depend upon which choice is taken, because

$$\|q_{j+1}\| \|p_{j+1}\| = \frac{\|r_j\| \|s_j\|}{\beta_j \gamma_j} = \frac{\|r_j\| \|s_j\|}{|(r_j, s_j)|}, \quad \|q_{j+1}\| \|p_{j+1}\| = \frac{1}{\cos \theta(r_j, s_j)},$$

where $\theta(r_j, s_j)$ denotes the angle between the vectors r_j and s_j [11]. The angle $\theta(r_j, s_j)$ is a function of A, q_1, p_1 apart from a normalizing factor. This angle can be equal to $\frac{1}{2}\pi$, causing the algorithm to stop. It is interesting to note that r_j and s_j can be written as $r_j = \phi_j(A)q_1$, $s_j = \phi_j(A^T)p_1$, where ϕ_j denotes a polynomial of degree j . Different choices of the β_j and γ_j correspond to different scaling of the Lanczos vectors. Hence, any resulting tridiagonal matrices T_j have the same eigenvalues [11].

The biorthogonal Lanczos algorithm can be regarded as the two-sided Gram–Schmidt process applied to the columns of the special matrices

$$R = R_j := [q_1, Aq_1, A^2q_1, \dots, A^{j-1}q_1],$$

$$L = L_j := [p_1, A^T p_1, (A^T)^2 p_1, \dots, (A^T)^{j-1} p_1].$$

The R , L matrices are called Krylov matrices. Note that the (i, j) element of $L^T R$ is $(p_1^T A^{i-1})(A^{j-1} q_1)$, so

$$L^T R = M = M(p_1, q_1), \quad \text{where } m_{i+1, j+1} = p_1^T A^{i+j} q_1.$$

The matrix M is called the *moment matrix* of (p_1, q_1) with respect to the matrix A . The following proposition gives breakdown conditions of the biorthogonal Lanczos method in terms of the nonsingularity of the moment matrices M_j . This proposition is proven in [8].

Proposition 4.1. *The biorthogonal Lanczos algorithm does not break down in the j th iteration if and only if*

$$\det(M_i) \neq 0, \quad i = 1, 2, \dots, j.$$

In the s -step method the subspace spanned by $V_k = \{\bar{V}_1, \bar{V}_2, \dots, \bar{V}_k\}$ is the same as the Krylov subspace spanned by the vectors $\{v_1^1, Av_1^1, A^2v_1^1, \dots, A^{sk-1}v_1^1\}$, and the subspace spanned by $W_k = \{\bar{W}_1, \bar{W}_2, \dots, \bar{W}_k\}$ is the same as the Krylov subspace spanned by the vectors $\{w_k^1, A^T w_k^1, \dots, (A^T)^{sk-1} w_k^1\}$.

Proposition 4.2. *The s -step biorthogonal Lanczos algorithm does not break down in the k th iteration if and only if*

$$\det(M_i) \neq 0, \quad \text{for } i = s, 2s, \dots, ks.$$

Proof. Let v_k^l and w_k^l be the s -step biorthogonal Lanczos vectors. For each k , v_k^l , $1 \leq l \leq s$, is a linear combination of the first $(k-1)s + l$ columns of R while w_k^l is the same linear combination of the columns of L , up to a scaling. This can be expressed compactly in matrix notation as

$$V_k = R\bar{K}^{-T}, \quad W_k = L\bar{K}^{-T}, \quad (4.1)$$

where \bar{K} is the lower triangular whose diagonal elements are 1.

Using (4.1) we can rewrite M as

$$M = L^T R = (\bar{K}W^T)(V\bar{K}^T),$$

that is,

$$M = \bar{K}(W^T V)\bar{K}^T.$$

By Lemma 3.8 the matrix $W^T V$ equals to $\text{diag}\{W_1^T V_1, \dots, W_k^T V_k\}$. By a sequence of row operations including exchanging rows, we can reduce $\bar{W}_k^T \bar{V}_k$ to an upper triangular matrix U_k with nonzero diagonal elements and

$$\det(\bar{W}_k^T \bar{V}_k) = \pm \det(\bar{U}_k).$$

Let $U = \text{diag}\{\bar{U}_1, \dots, \bar{U}_k\}$; then

$$\det(W^T V) = \pm \det(U),$$

that is,

$$\det(M) = \det(W^T V) = \det(\bar{W}_1^T \bar{V}_1) \det(\bar{W}_2^T \bar{V}_2) \cdots \det(\bar{W}_k^T \bar{V}_k).$$

This proves the proposition. \square

The standard biorthogonal Lanczos algorithm produces a tridiagonal matrix T_j by the end of step $j (= ks)$ and the s -step method produces a block tridiagonal matrix T_k .

5. Numerical experiments

We implemented the standard biorthogonal Lanczos algorithm and s -step methods. Firstly, we show the reduced breakdown effect of the s -step biorthogonal method compared to the standard one. We borrow an example from [11]:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

The eigenvalues of A are the roots of unity, i.e., $\lambda_j = e^{2n\pi i/j}$ for $j = 0, \dots, n - 1$ where n is the dimension of A . The biorthogonal Lanczos method starts with initial vectors $r_1^T = s_1^T = [123456]^T$ on the CRAY-2 machine. At step 4, the standard biorthogonal Lanczos method normalizes s_4 and r_4 by factors of $\sqrt{10^{-13}}$, producing elements in the reduced tridiagonal matrix T of size 10^{13} on a CRAY-2. The 3-step biorthogonal Lanczos method avoids the large element growth in reduced matrix and can generate a 3×3 block tridiagonal matrix which has the sixth root of unity. At the second iteration of the 2-step method we have to solve a linear system with singular 2×2 matrix. This can be explained by Proposition 4.2. The eigenvalues of the above matrix A by the QR algorithm and 3-step biorthogonal Lanczos algorithm are exactly the same.

Secondly we show the s -step method has better performance on vector and parallel computers. The test problem was derived from the five-point discretization of the following partial differential equation:

$$-(bu_x)_x - (cu_x)_x + (du)_x + (eu)_y + fu = g,$$

on the unit square, where

$$b(x, y) = e^{-xy}, \quad c(x, y) = e^{xy}, \quad d(x, y) = \beta(x + y),$$

$$e(x, y) = \gamma(x + y) \quad \text{and} \quad f(x, y) = \frac{1}{1 + x + y},$$

subject to the Dirichlet boundary conditions $u = 0$ on the boundary. The right-hand side g was

chosen so that the solution was known to be $e^{xy} \sin(\pi x) \sin(\pi y)$. The parameters β and γ are useful for changing the degree of nonsymmetry of the resulting coefficient matrix of the linear systems. Note that the matrix A resulting from the discretization remains positive real independent of these parameters. We denote by n_1 the number of interior nodes on each side of the square and by $h = 1/(n_1 + 1)$ the mesh size.

In the test we took $\gamma = 50.0$, $\beta = 1.0$, which yielded a nonsymmetric matrix of dimension $n = n_1^2$. We used the CRAY-2 multiprocessor at the Minnesota Supercomputer Institute. The CRAY-2 is a four-processor machine, each of whose processors can execute independent tasks concurrently. All processors have equal access to a central memory of 512 Megawords. Each CRAY-2 processor has 8 vector registers (each 64 words long) and has data access through a single path between its vector registers and main memory. Each processor has 16 Kwords of local memory with no direct path to central memory but with a separate data path between local memory and its vector registers.

The maximum performance of the CRAY-2 for specific applications comes from data movement minimization, good vectorization and division into multiprocessing tasks. Because of single paths between vector register and central or local memory on the CRAY-2 system, memory transfers constitute a severe bottleneck for achieving maximum performance. Therefore, minimization of data movement results in faster execution times. Algorithms must provide good data locality, that is, the organization of the algorithm should be such that the data can be kept as long as possible in fast registers or local memories and have many operations performed on them.

Macrotasking (often called multitasking) on a data parallel algorithm [6] is most often applied to parallel work found in the independent iterations of DO loops. If the loop has N iterations, we map the N iterations onto P processors or tasks so that each task has the same amount of work to do. The extremely large memory of the CRAY-2 means that many jobs can usually be resident in the main storage at the same time leading to very efficient multiprogramming. To achieve load balancing, we must consider static and dynamic partitioning. We use static partitioning when the times for each of the loop iterations are approximately equal.

Example of contiguous static partitioning with $P = 4$:

Processor	Assigned iterations
PE0	$I = 1, \frac{1}{4}n$
PE1	$I = \frac{1}{4}n + 1, 2 * \frac{1}{4}n$
PE2	$I = 2 * \frac{1}{4}n + 1, 3 * \frac{1}{4}n$
PE3	$I = 3 * \frac{1}{4}n + 1, n$

We must minimize calls to the multitasking library because multitasking introduces an overhead that increases CPU-time. The distribution of matrix and vectors with 4 processors is as shown in Fig. 5.1.

In the accuracy test, Table 5.1 shows that matrices generated by the standard and restructured and s -step biorthogonal Lanczos methods have the same largest eigenvalues, but for $s > 5$ in the s -step method loss of accuracy for eigenvalues has been observed. We tested the methods on the problem of size $n = 4096$. We also compared to the largest eigenvalues computed by the standard, the modified and the s -step methods with those by Saad's program

Table 5.1
Largest eigenvalues using the biorthogonal Lanczos methods on a CRAY-2

T_j	code *	Standard	Modified	2-step	3-step	4-step	5-step	6-step
10×10	0.102 040 00 · 10 ²	0.986 526 73 · 10 ²	0.986 526 73 · 10 ²	0.986 526 73 · 10 ²	-	-	0.986 526 73 · 10 ²	-
20×20	0.102 040 00 · 10 ²	0.102 024 84 · 10 ²	0.102 024 84 · 10 ²	0.102 024 84 · 10 ²	-	0.102 024 84 · 10 ²	0.102 024 91 · 10 ²	-
30×30	0.102 039 99 · 10 ²	0.102 040 00 · 10 ²	0.102 040 00 · 10 ²	0.102 040 00 · 10 ²	0.102 040 00 · 10 ²	-	0.102 020 16 · 10 ²	0.101 972 83 · 10 ²
40×40	0.102 038 72 · 10 ²	0.102 040 00 · 10 ²	0.102 040 00 · 10 ²	0.102 040 00 · 10 ²	-	0.102 040 01 · 10 ²	0.102 020 15 · 10 ²	-

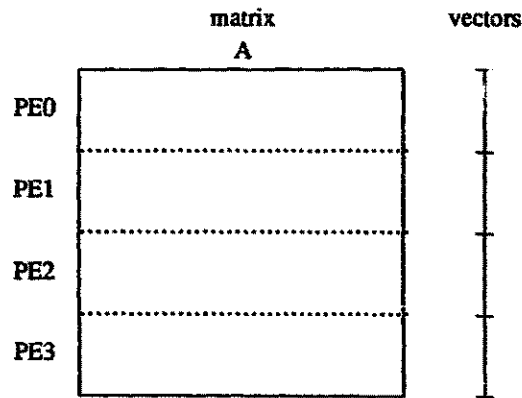


Fig. 5.1. Distribution of matrix and vectors to each processor.

code* using the Arnoldi method. Saad's code uses reorthogonalization and deflated iteration [12] to compute eigenvalues. The stopping criterion is $\epsilon = 10^{-6}$ in Saad's program. In the standard, restructured and s -step biorthogonal Lanczos methods, we find the largest eigenvalues after a reduced matrix of a certain size is generated, so these methods require minimal storage and time.

We reduced the different size matrices A of the model problem to 20×20 tridiagonal or block tridiagonal matrices using the standard, and 5-step biorthogonal Lanczos algorithms. In the s -step method the synchronization points are less as the size s is bigger, but we cannot choose $s > 5$ because the errors in the Ritz values are increased. So we choose the 5-step method in the performance test to compare with the standard biorthogonal Lanczos method (see Table 5.2). Figure 5.2 shows the time these methods took to make 20×20 reduced matrices for different size test problems in the CRAY-2 with 4 processors. Figure 5.3 shows the speedup (P1/P4) when 1 processor and 4 processors are used in CRAY-2 for the standard and

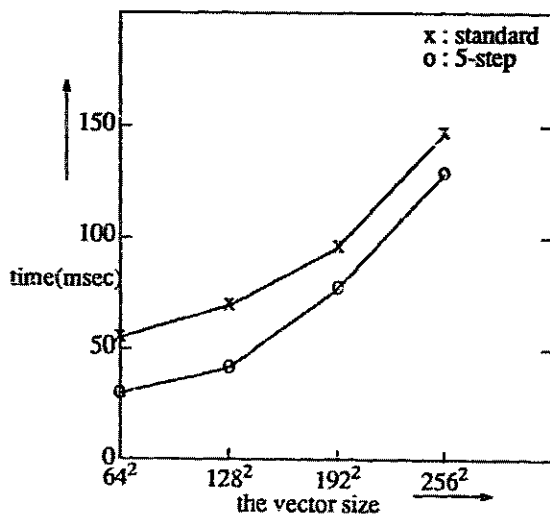


Fig. 5.2. CRAY-2 performance (msec) using 4 processors for the biorthogonal Lanczos methods.

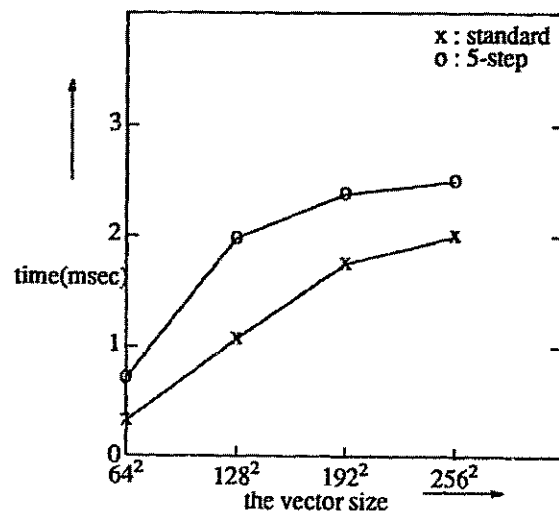


Fig. 5.3. Speedup performance (P1/P4) for the biorthogonal Lanczos methods on a CRAY-2.

Table 5.2
 CRAY-2 computation time (msec) of the 5-step and standard biorthogonal Lanczos methods

Vector size	Standard biorthogonal Lanczos algorithm		5-step biorthogonal Lanczos algorithm	
	$p = 1$	$p = 4$	$p = 1$	$p = 4$
64^2	18.89	55.03	22.13	30.82
128^2	74.74	69.68	83.72	42.28
192^2	167.98	96.04	184.19	77.52
256^2	294.95	147.30	321.85	129.05
320^2	467.72	192.33	501.93	172.94

5-step methods. In the s -step method, memory reference time and calls to the macrotasking library are decreased by making possible grain size large and by decreasing synchronization points. This accounts for the superior performance of the 5-step method over the standard biorthogonal Lanczos method on the CRAY-2.

6. Conclusions

We have introduced an s -step biorthogonal method and proved that s -step methods generate reduction matrices which are similar to reduction matrices generated by the standard biorthogonal Lanczos method. Serious breakdown conditions are decreased in the resulting algorithm. Also, the s -step method has better data locality and parallel properties than standard ones. In the s -step methods, the inner products needed for s steps of the standard methods can be performed simultaneously and the vector updates are replaced by linear combinations. The s -step Lanczos method has additional operations (compared to that of the standard Lanczos method). The design of a stable s -step Lanczos method with no additional vector operations compared to the standard method remains an open question. For large $s > 5$ loss of accuracy for eigenvalues has been observed.

Acknowledgements

We thank the anonymous referees whose comments helped enhance significantly the quality of presentation of this article.

References

- [1] A.T. Chronopoulos and C.W. Gear, s -step iterative methods for symmetric linear systems, *J. Comput. Appl. Math.* **25** (2) (1989) 153–168.
- [2] A.T. Chronopoulos and C.W. Gear, On the efficient implementation of preconditioned s -step conjugate gradient methods on multiprocessors with memory hierarchy, *Parallel Comput.* **11** (1) (1989) 37–53.
- [3] Z. Cvetanovic, The effect of problem partitioning, allocation, and granularity on the performance of multiple-processor systems, *IEEE Trans. Comput.* **C-36** (4) (1987) 421–432.
- [4] A.K. Dave and I.S. Duff, Sparse matrix calculations on the CRAY-2, *Parallel Comput.* **5** (1&2) (1987) 55–64.

- [5] G.H. Golub and C.F. Van Loan, *Matrix Computations* (Johns Hopkins Univ. Press, Baltimore, MD, 2nd ed., 1989).
- [6] W.D. Hillis and G.L. Steele, Data parallel algorithms, *Comm. ACM* **29** (12) (1986) 1170–1183.
- [7] S.K. Kim and A.T. Chronopoulos, A class of Lanczos-like algorithms implemented on parallel computers, *Parallel Comput.* **17** (6 & 7) (1991) 763–778.
- [8] B.N. Parlett, D.R. Taylor and Z.A. Liu, A lookahead algorithm for unsymmetric matrices, *Math. Comp.* **44** (169) (1985) 105–124.
- [9] A. Ruhe, The two sided Arnoldi algorithm for nonsymmetric eigenvalue problems, in: *Proc. Conf. on Matrix Pencils*, Pitea, 1982, Lecture Notes in Math. **973** (Springer, Berlin, 1982) 104–120.
- [10] Y. Saad, Variation on the Arnoldi's method for computing eigenelements of large unsymmetric matrices, *Linear Algebra Appl.* **34** (1980) 269–295.
- [11] Y. Saad, The Lanczos biorthogonalization algorithm and other oblique projection methods for solving large unsymmetric systems, *SIAM J. Numer. Anal.* **19** (3) (1982) 485–506.
- [12] Y. Saad, Partial eigensolutions of large nonsymmetric matrices, Research Report YALUE/DCS/RR-397, Yale Univ., 1985.
- [13] Y. Saad, Krylov subspace methods on supercomputers, *SIAM J. Sci. Statist. Comput.* **10** (6) (1989) 1200–1232.
- [14] P.E. Saylor, Leapfrog variants of iterative methods for linear algebraic equations, *J. Comput. Appl. Math.* **24** (1 & 2) (1988) 169–193.
- [15] J.H. Wilkinson, *The Algebraic Eigenvalue Problem* (Oxford Univ. Press, Oxford, 1965).