ISE

ISE-TR-92-100

REORTHOGONALIZATION IN THE BLOCK LANCZOS ALGORITHM

by

Hiroko Iguchi, Makoto Natori and Hitoshi Imai

November 30. 1992

INSTITUTE
OF
INFORMATION SCIENCES AND ELECTRONICS
UNIVERSITY OF TSUKUBA

REORTHOGONALIZATION IN THE BLOCK LANCZOS ALGORITHM

Hiroko Iguchi, Makoto Natori and Hitoshi Imai
University of Tsukuba
Institute of Information Sciences and Electronics
Ibaraki 305, Japan

Abstract

The reorthogonalization is a significant problem in the Lanczos algorithm. Some satisfactory reorthogonalization methods have already been developed in the simple Lanczos algorithm. However, there are few reorthogonalizations in the block Lanczos algorithm. In this paper, two reorthogonalization methods PRO and RIC for the simple Lanczos algorithm are extended for the block Lanczos algorithm. Numerical results show that RIC gives satisfactory results, while PRO is superior to RIC in the computation time but inferior in the reliability.

1 Introduction

When we solve the following eigenvalue problem of a symmetric $n \times n$ matrix A

$$Ax = \lambda x , \tag{1}$$

we transform A to a tridiagonal matrix. For this purpose, the Householder transformation and the Givens method are usually applied. When the matrix is large and sparse, the Lanczos algorithm is effective. However, the orthogonality among Lanczos vectors is lost as the Lanczos steps proceed, since the Lanczos algorithm is sensitive to roundoff error. Therefore the Lanczos algorithm needs some reorthogonalizations.

For the simple Lanczos algorithm, SO (Selective Orthogonalization) [3], PRO (Partial Reorthogonalization) [4] and RIC (Reorthogonalization with Improved Convergence-Check) [1] are proposed. When A has multiple eigenvalues, however, the simple Lanczos algorithm will fail even with these reorthogonalizations. The block Lanczos algorithm has been developed to compute multiple eigenvalues [2]. Of course, the block Lanczos algorithm also needs some reorthogonalizations. SO in the simple Lanczos algorithm has been extended and applied to the block Lanczos algorithm [2]. However, it needs rather long computational time and it does not give satisfactory results in the reliability. Therefore, we will extend PRO and RIC in the simple Lanczos algorithm to the block Lanczos algorithm. In the section 6, we show the superiority of our methods compared with SO.

2 The simple Lanczos algorithm and the reorthogonalizations

2.1 The simple Lanczos algorithm

The simple Lanczos algorithm is given as follows.
(i) Choose a starting vector v_1 ($\| v_1 \| = 1$), and set $u_1 = Av_1$.
(ii) Compute α_k and β_k iteratively as follows.

$$
\begin{aligned}
(\; j \; &= 1,2,\ldots \;) \\
\alpha_j &= u_j^T v_j \; , \\
r_j &= u_j - \alpha_j v_j \; , \\
\beta_j &= \| r_j \| \; , \\
v_{j+1} &= \frac{r_j}{\beta_j} \; , \\
u_{j+1} &= A v_{j+1} - \beta_j v_j \; ,
\end{aligned}
$$

where $\{v_j\}_{j=1,2,\ldots,n}$ are Lanczos vectors which are mutually orthogonal.

The procedure (ii) is called the j-th Lanczos step. After the j-th Lanczos step, A is transformed to a tridiagonal matrix T_j :

$$
T_j = \begin{bmatrix}
\alpha_1 & \beta_1 & & & 0 \\
\beta_1 & \alpha_2 & \beta_2 & & \\
 & \cdots & \cdots & & \\
 & & \cdots & \beta_{j-1} \\
0 & & \beta_{j-1} & \alpha_j
\end{bmatrix} .
$$

The Ritz values $\{\theta_i\}_{i=1,\ldots,j}$ and Ritz vectors $\{y_i\}_{i=1,\ldots,j}$ are defined as follows.

$$
T_j s_i^j = \theta_i^j s_i^j \; , \qquad y_i^j = [v_1 \ldots v_j] s_i^j \; .
$$

If $j = n$ then Ritz values and Ritz vectors coincide with the eigenvalues and eigenvectors of the original problem eq.(1) respectively.

2.2 Reorthogonalizations

The Lanczos algorithm is so sensitive to roundoff errors that the orthogonality among Lanczos vectors is lost as the Lanczos steps proceed. This loss of orthogonality causes the redundant copies of Ritz pairs, therefore some computed eigenvalues are not true. To relax the sensitivity to roundoff errors the reorthogonalization of Lanczos vectors is carried out. Matrices used in the Lanczos algorithm are usually large, full reorthogonalization(FRO) is not practical. Hence some partial reorthogonalizations have been introduced.

2.2.1 Selective Reorthogonalization(SO)

The loss of the orthogonality at the j-th step is monitored by

$$\begin{aligned} \beta_{j,i} &= \beta_j \mid \sigma_{j,i} \mid \\ &= \| A y_i^j - y_i^j \theta_i^j \| , \qquad 1 \leq i \leq j , \end{aligned}$$

where $\sigma_{j,i}$ is the j-th component of s_i^j. If $\beta_{j,i} < \sqrt{\varepsilon}$ (ε is the machine epsilon), then the i-th Ritz vector is considered to be converged. This indicates that the Lanczos vectors have lost the orthogonality in the direction of the i-th Ritz vector [2]. Therefore r_j is reorthogonalized to the i-th Ritz vector. Thus SO is carried out as follows [3].

$$r_j \leftarrow r_j - \sum_{i \in L(j)} (r_j^T y_i^j) y_i^j ,$$

where

$$L(j) = \{ i \mid 1 \leq i \leq j , \quad \beta_{j,i} < \sqrt{\varepsilon} \} .$$

2.2.2 Partial Reorthogonalization(PRO)

The orthogonality of the Lanczos vectors

$$w_{k,i} = v_k^T v_i ,$$

satisfies the following recurrence formula. It is derived by taking into account the roundoff error in the Lanczos step.

$$
\begin{aligned}
w_{k,0} &= 0\ , w_{k,k} = 1\ , \qquad (1 \le k \le j+1)\ , \\
w_{k,k-1} &= \psi_k\ , \qquad (2 \le k \le j)\ , \\
&\psi_k = \varepsilon n\left(\tfrac{\beta_1}{\beta_k}\right)\Psi\ , \qquad \Psi \in N(0, 0.6)\ , \\
w_{k+1,i} &= \frac{1}{\beta_k}[\beta_i w_{k,i+1} + (\alpha_i - \alpha_k) w_{k,i} \\
&\quad + \beta_{i-1} w_{k,i-1} - \beta_{k-1} w_{k-1,i}] + \phi_{k,i}\ , \\
&\qquad (2 \le k \le j\ , 1 \le i \le k-1)\ , \\
&\phi_{k,i} = \varepsilon(\beta_k + \beta_i)\Phi\ , \qquad \Phi \in N(0,\ 0.3)\ .
\end{aligned}
$$

Here $N(\cdot,\cdot)$ is a set of normal random numbers. The loss of the orthogonality at the j-th step is monitored by $w_{j+1,i}$. If $\mid w_{j+1,i} \mid > \sqrt{\varepsilon}$, then the i-th Lanczos vector is chosen for the reorthogonalization. Thus PRO is carried out as follows [4].

$$
r_j \leftarrow r_j - \sum_{i \in L(j)} (r_j^T v_i^j) v_i^j\ ,
$$

where

$$
L(j) = \{\, i \mid 1 \le i \le j\ , \quad \mid w_{j+1,i} \mid > \sqrt{\varepsilon}\ \}\ .
$$

After the reorthogonalization has been carried out, we set

$$
w_{j+1,i} \in N(0,\ 1.5)\varepsilon\ .
$$

2.2.3 Reorthogonalization with Improved Convergence-Check(RIC)

The loss of the orthogonality is monitored by $w_{j+1,i}$ as in PRO. If $\mid w_{j+1,i} \mid > \sqrt{\varepsilon}$, then the convergence check of the Ritz values is carried out by comparing the Ritz values $\{\lambda_i^j\}_{i=1}^j$ at the j-th step with those $\{\lambda_i^{j-1}\}_{i=1}^{j-1}$ at the $(j-1)$-th step (see Figure 1). The Ritz vectors which correspond to the converged Ritz values are chosen for the reorthogonalization. Thus RIC is carried out as follows [1].

$$
r_j \leftarrow r_j - \sum_{i \in L(j)} (r_j^T y_i^j) y_i^j\ ,
$$

where

$$
\begin{aligned}
L(j) = \{ i \mid\ & 1 \le i \le j-1,\ \mid \lambda_i^{j-1} - \lambda_i^j \mid < \tau \ \ or \\
& 2 \le i \le j,\ \mid \lambda_{i-1}^{j-1} - \lambda_i^j \mid < \tau \}\ .
\end{aligned}
$$

Here τ is a tolerance.

$$\begin{array}{lccccc} (j-1)-th\ step & \lambda_1^{j-1} & \lambda_2^{j-1} & \ldots & \lambda_{j-2}^{j-1} & \lambda_{j-1}^{j-1} \\ & \downarrow & \downarrow & & \downarrow & \downarrow \\ j-th\ step & \lambda_1^{j} & \lambda_2^{j} & \ldots & \lambda_{j-1}^{j} & \lambda_j^{j} \end{array}$$

Figure 1: Correspondance for convergence check

3 The block Lanczos algorithm and the reorthogonalization SO

3.1 The block Lanczos algorithm

The block Lanczos algorithm is carried out as follows by replacing each Lanczos vector v_k in the simple algorithm by an $n \times p$ orthogonal matrix V_k. Here p denotes the block size [2].
(i) Choose a starting matrix V_1 ($V_1^T V_1 = I_p$) , and set

$$U_1 = AV_1 .$$

(ii) Compute M_j ($p \times p$) and B_j ($p \times p$, *upper triangle*) iteratively as follows.

$$\begin{aligned} (j &= 1,2,\ldots) \\ M_j &= U_j^T V_j , \\ R_j &= U_j - V_j M_j , \\ V_{j+1} B_j &= R_j \ (the\ QR\ factorization\ of\ R_j) , \\ U_{j+1} &= AV_{j+1} - V_j^T B_j . \end{aligned}$$

After the j-th block Lanczos step, A is transformed to a block tridiagonal matrix T_j :

$$T_j = \begin{bmatrix} M_1 & B_1^T & & 0 \\ B_1 & M_2 & B_2^T & \\ & \cdots & \cdots & \\ & & \cdots & B_{j-1}^T \\ 0 & & B_{j-1} & M_j \end{bmatrix} .$$

3.2 The reorthogonalization SO

The block Lanczos algorithm is subject to the same loss of orthogonality as the simple Lanczos algorithm. The reorthogonalization SO described earlier are available in the block Lanczos [2]. The loss of orthogonality at the j-th step is monitored by

$$\beta_{j,i} = \| B_{j+1}\sigma_{j,i} \| , \qquad 1 \le i \le j ,$$

where $\sigma_{j,i}$ is the vector whose elements are the last p elements of s_i^j . If $\beta_{j,i} < \sqrt{\varepsilon}$, then the i-th Ritz vector are chosen for the reorthogonalization.

$$r_{j,m} \leftarrow r_{j,m} - \sum_{i \in L(j)} (r_{j,m}^T y_i^j) y_i^j , \qquad (m = 1, 2, \ldots, p) ,$$

where

$$R_j = (r_{j,1} \ldots r_{j,p}) ,$$
$$L(j) = \{ i \mid 1 \le i \le j , \quad \beta_{j,i} < \sqrt{\varepsilon} \} .$$

4 The extended recurrence formula

In this section reorthogonalizations PRO and RIC are extended for the block Lanczos algorithm. First, the recurrence formula for the simple Lanczos algorithm is extended as follows. The $p \times p$ matrix

$$W_{k,i} = V_k^T V_i ,$$

is defined by the following recurrence formula.

$$\left.\begin{array}{rcl}
W_{k,0} &=& 0_p \ , \quad W_{k,k} \ = \ I_p \ , \quad (1 \leq k \leq j+1) \ , \\
W_{k,k-1} &=& \Gamma_k \ , \quad (2 \leq k \leq j) \ , \\
W_{k+1,i} &=& (B_k^T)^{-1}(W_{k,i+1}B_i + W_{k,i}M_i \\
&& +W_{k,i-1}B_{i-1} - M_k W_{k,i} - B_{k-1}^T W_{k-1,i}) \\
&& +\Theta_{k,i} \ , \\
&& (2 \leq k \leq j \ , \quad 1 \leq i \leq k-1) \ , \\
W_{i,k+1} &=& W_{k+1,i}^T \ , \\
\Gamma_k &=& \varepsilon n (B_k^T)^{-1} B_1 \Psi \\
&& (\ \textit{the elements of } \Psi \in N(0, \ 0.6)\ ) \ , \\
\Theta_{k,i} &=& \varepsilon (B_k + B_i) \Psi \\
&& (\ \textit{the elements of } \Psi \in N(0, \ 0.3)\ ) \ ,
\end{array}\right\} \quad (2)$$

where $W_{\cdot,\cdot}$,$\Theta_{\cdot,\cdot}$ and $\Gamma_{\cdot}$ are $p \times p$ matrices, I_p is a unit matrix and, 0_p is a zero matrix. Ψ and Φ are $p \times p$ matrices whose elements are adequate random numbers.

This recurrence formula is derived as follows. Taking into account the roundoff errors, the k-th Lanczos step is written as

$$V_{k+1}B_k \ = \ AV_k - V_k M_k - V_{k-1}B_{k-1}^T - F_k \ , \quad (3)$$

where the $n \times p$ matrix F_k stands for the roundoff errors in the k-th step. From eq.(3), we have

$$B_k^T V_{k+1}^T V_i \ = \ V_k^T A V_i - M_k V_k^T V_i - B_{k-1} V_{k-1}^T V_i - F_k^T V_i \ . \quad (4)$$

For $k = i$, eq.(3) becomes

$$V_{i+1}B_i \ = \ AV_i - V_i M_i - V_{i-1}B_{i-1}^T - F_i \ .$$

Then

$$AV_i \ = \ V_{i+1}B_i + V_i M_i + V_{i-1}B_{i-1}^T + F_i \ . \quad (5)$$

Substituting eq.(5) into eq.(4), we have

$$\left.\begin{aligned} B_k^T V_{k+1}^T V_i \;=\;& V_k^T V_{i+1} B_i + V_k^T V_i M_i \\ & + V_k^T V_{i-1} B_{i-1}^T + V_k^T F_i - M_k V_k^T V_i \\ & - B_{k-1} V_{k-1}^T V_i - F_k^T V_i \ . \end{aligned}\right\} \qquad (6)$$

Eq.(6) can be rewritten as

$$\begin{aligned} W_{k+1,i} \;=\;& (B_k^T)^{-1}(W_{k,i+1}B_i + W_{k,i}M_i + W_{k,i-1}B_{i-1} \\ & - M_k W_{k,i} - B_{k-1}^T W_{k-1,i} + \Theta_{k,i} \ , \\ & \qquad (2 \le k \le j \ , \quad 1 \le i \le k-1) \ , \end{aligned}$$

where

$$\Theta_{j,k} \;=\; (B_j^T)^{-1}(V_j^T F_k - f_j^T V_k) \ .$$

5 Reorthogonalization in the block Lanczos algorithm

Making use of the recurrence formula introduced in the section 4, we propose two reorthogonalization methods for the block Lanczos algorithm.

5.1 The algorithm of block PRO

(i) Carry out a block Lanczos step .
(ii) Compute $W_{j+1,i}$($i = 1, \ldots, j$) using the recurrence formula (2) and let $\tilde{\omega}_{j+1,i}$ be the largest absolute value of elements of $W_{j+1,i}$.
(iii) Reorthogonalize $R_j = (r_{j,1} \ldots r_{j,p})$ using the Lanczos vectors $V_j = (v_{i,1} \ldots v_{i,p})$ as follows.

$$r_{j,n} \leftarrow r_{j,n} - \sum_{i \in L(j)} \sum_{m=1}^{p} (r_{j,n}^T v_{i,m}) v_{i,m}, \qquad (n = 1, \ldots, p)$$

where

$$L(j) \;=\; \{\, i \mid 1 \le i \le j \ , \quad \tilde{\omega}_{j+1,i} > \sqrt{\varepsilon} \,\}$$

(iv) After the reorthogonalization has been carried out, elements of $W_{j+1,i}$ are set the elements of $\varepsilon N(0, 1.5)$.

5.2 The algorithm of block RIC

(i) Carry out a block Lanczos step.
(ii) Compute $W_{j+1,i}$ ($i = 1,\ldots,j$) using the recurrence formula eq.(2) and let $\tilde{\omega}_{j+1,i}$ be the largest absolute value of the elements of $W_{j+1,i}$.
(iii) Check the convergence of the Ritz values only when $\tilde{\omega}_{j+1,i}$ becomes greater than $\sqrt{\varepsilon}$ for some i.
(iv) Carry out the convergence check of the Ritz values by comparing the Ritz values at the j-th step with those at the $(j-1)$-th step(see Figure 1).
(v) Reorthogonalize $R_j = (r_{j,1}\ldots r_{j,p})$ using the Ritz vectors which correspond to the converged Ritz values as follows.

$$r_{j,n} \leftarrow r_{j,n} - \sum_{i\in L(j)} (r_{j,n}y_i^j)y_i^j , \qquad (n = 1,\ldots,p)$$

where

$$\begin{aligned} L(j) = \{ i \mid\ & 1 \le i \le j \times (p-1),\ \mid \lambda_i^{j-1} - \lambda_i^j \mid < \tau \ or \\ & p \le i \le j \times p,\ \mid \lambda_{i-1}^{j-1} - \lambda_i^j \mid < \tau \} . \end{aligned}$$

Here τ is a tolerance. We have used throughout the value $\tau = 10^{-8}$.

6 Numerical result

6.1 The sparse matrices used in the numerical computations

Matrices used here are obtained by discretization of the two dimensional Laplace operator which is defined in the rectangular domain and is subject to the homogeneous Dirichlet boundary condition. As for these matrices exact eigenvalues are given as follows.

$$\lambda_{i,j} = 4\left\{\sin^2\left(\frac{\pi i}{2(I+1)}\right) + \sin^2\left(\frac{\pi j}{2(J+1)}\right)\right\} ,$$
$$i = 1,\ldots,I, \quad j = 1,\ldots,J ,$$

where $(I+1)$ and $(J+1)$ are division numbers of the rectangular domain along the x-axis and the y-axis respectively. The degree of the matrix is $n(= I \times J)$.

6.2 The loss of the orthogonality

The loss of the orthogonality among the Lanczos vectors in the blockPRO and the block RIC are examined (see Figure 2. Figure 3). Here, circles denote

$$H_j = \max_{\substack{1\leq k\leq p\\ 1\leq i\leq j\\ 1\leq m\leq p}} v_{j+1,k}^T v_{i,m} ,$$

and squares denote

$$H_j^r = \max_{1\leq i\leq j} \tilde{\omega}_{j+1,i} .$$

H_j is the true orthogonality among Lanczos vectors and H_j^r is the orthogonality computed by the recurrence formula. These figures show that the loss of orthogonality can be monitored by the recurrence formula.

6.3 Numerical results in the case of multiple eigenvalues

Numerical computations by the Lanczos algorithm with the reorthogonalizations SO, PRO and RIC are carried out for the matrices which have multiple eigenvalues.

The simple Lanczos algorithms with SO, PRO and RIC are extremely inaccurate(see Table 1). While the block Lanczos algorithms give good results (see Table 2), provided that the block size is larger than or equal to the largest multiplicity.

Otherwise, in the case that the block size is smaller than the largest mul-

Table 1: Simple Lanczos,N=225(I=3,J=75), the largest multiplicity =3.

	error(average)	cpu time(sec)
SO	0.175E-3	234.392
PRO	0.159	6.664
RIC	0.126E-3	187.671

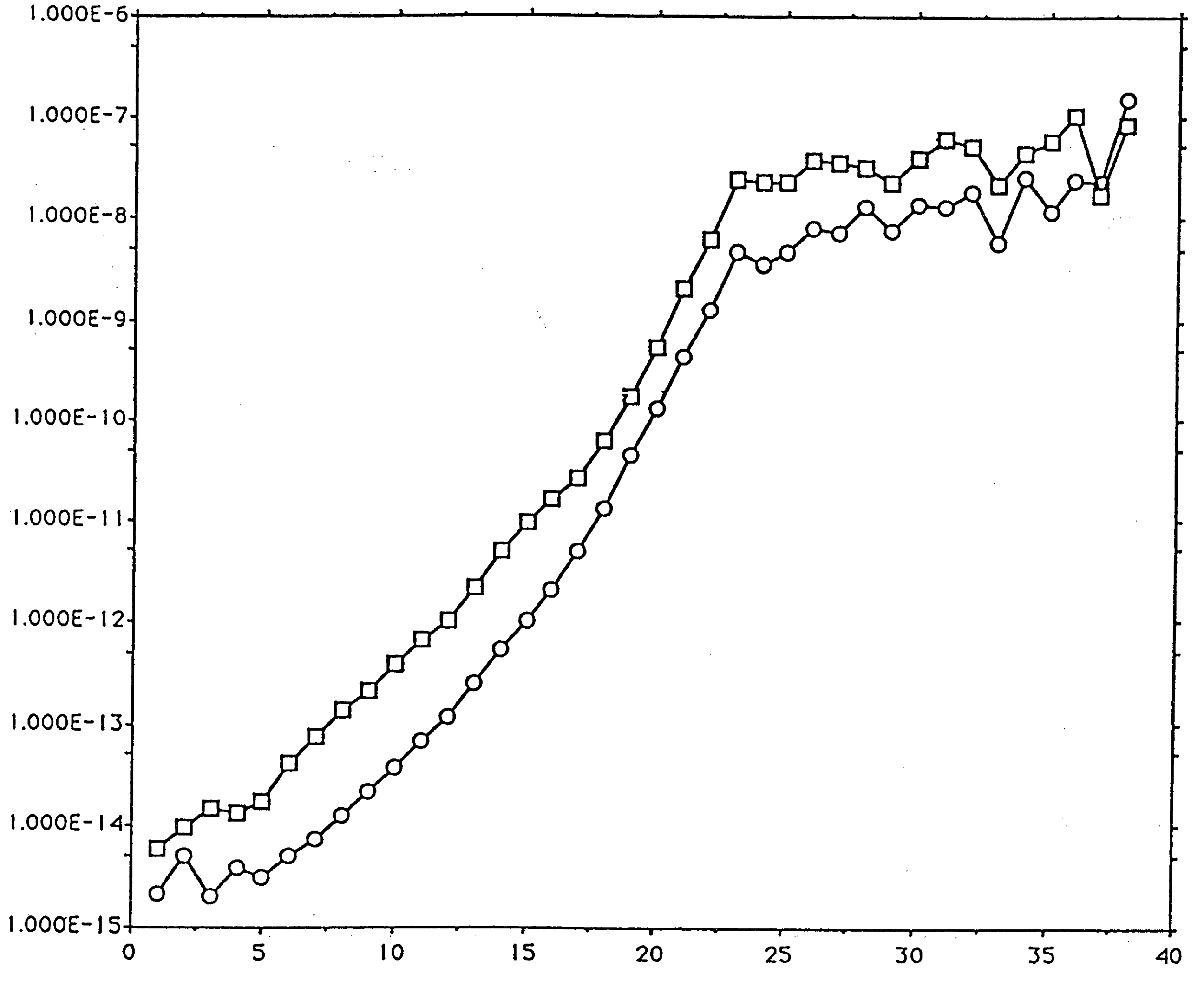

Figure 2: The orthogonality (PRO)

N=120, block size=3, the largest multiplicity=1

□ : H_j^r , ○ : H_j

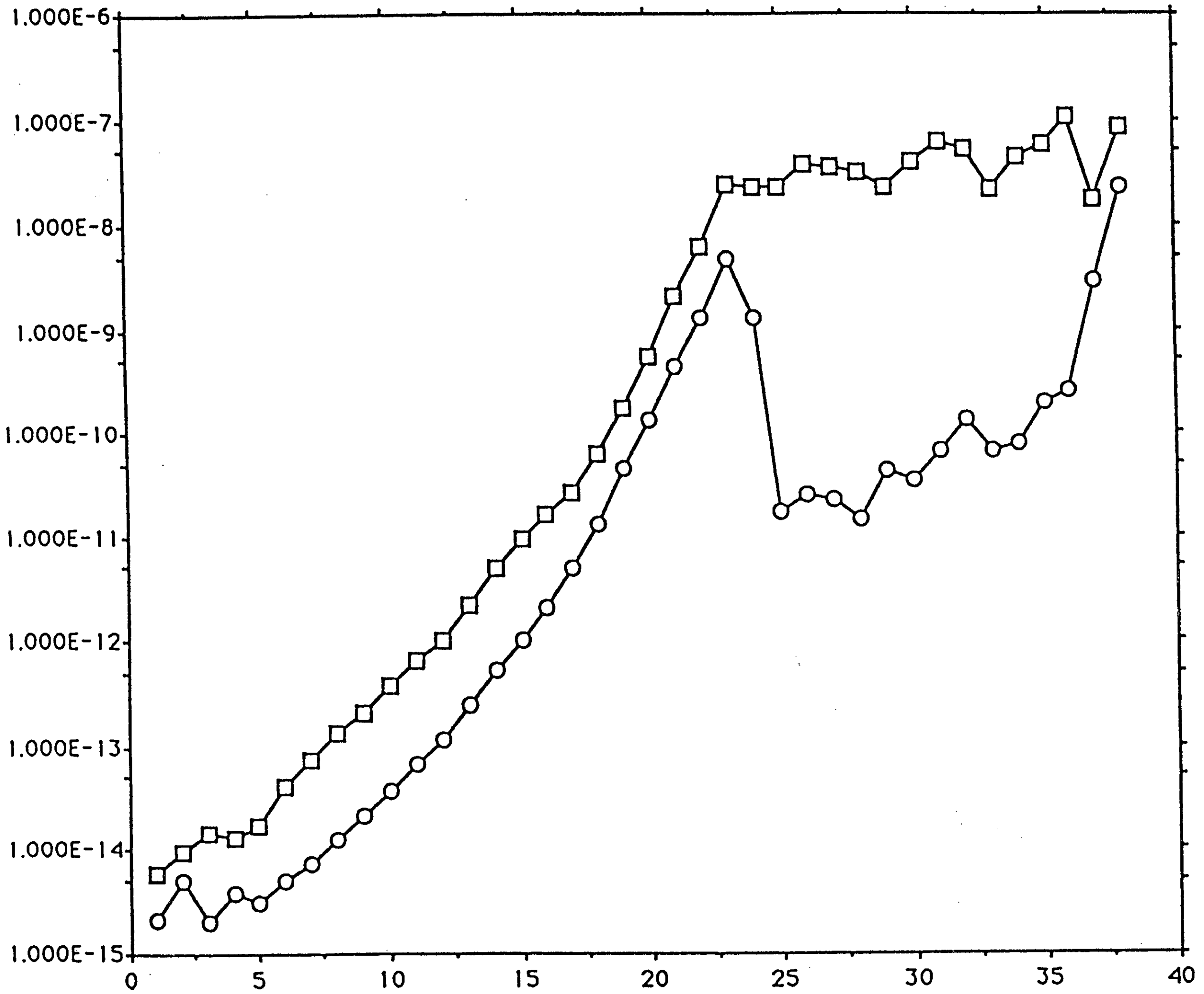

Figure 3: The orthogonality (RIC)

N=120, blocksize=3, the largest multiplicity=1

□ : H_j^r , ○ : H_j

Table 2: Block Lanczos, N=225(I=3,J=75),
block size=3, the largest multiplicity=3.

	error(average)	cpu time(sec)
SO	0.219E-13	97.652
PRO	0.220E-13	5.202
RIC	0.223E-13	79.611

tiplicity, the accuracy becomes considerably bad (see Table 3). In the practical problems, the information about multiplicity is unknown. This means that the block Lanczos algorithms with PRO and SO are not useful. While the block Lanczos algorithm with RIC indicates reliable eigenvalues among the computed eigenvalues, even if the block size is smaller than the largest multiplycity (see Figure 4). Of course, in the pratical problems exact eigenvalues and relative errors are not outputted.

The computations time of PRO is remarkable throughout numerical results.

Table 3: Block Lanczos, N=225(I=15,J=15),
block size=3,the largest multiplycity=15.

	error(average)	cpu time(sec)
SO	0.923E-3	126.301
PRO	0.314E-1	4.865

RIC
N = 225, p = 5
CPU TIME 88.9707269999999956

Largest 61 and smallest 61 eigenvalues are reliable.

**** LARGEST 61 *********

Computed	Exact	Relative Error
7.92314112161271263	7.92314112161292200	0.263993805775857942E-13
7.80932962582888007	7.80932962582903419	0.197326740195849637E-13
7.80932962582882636	7.80932962582903419	0.266135200033595445E-13
7.69551813004495961	7.69551813004514637	0.242660090699904322E-13
7.62450978541139886	7.62450978541155093	0.199197737353973948E-13
7.62450978541135578	7.62450978541155093	0.255986565985589323E-13
7.51069828962752566	7.51069828962766312	0.182999775984089147E-13
7.51069828962747632	7.51069828962766312	0.248335721852398841E-13
7.37578412317942035	7.37578412317955491	0.182433526168012690E-13
7.37578412317937976	7.37578412317955491	0.237825883948399381E-13

(The rest is omitted.)

**** SMALLEST 61 **********

Computed	Exact	Relative Error
0.768588783870802903E-01	0.768588783870781536E-01	0.278065379986437350E-13
0.190670374170970935	0.190670374170965514	0.283130480060802185E-13
0.190670374170979948	0.190670374170965526	0.756227683247232366E-13
0.304481869954862357	0.304481869954852911	0.310388710451724452E-13
0.375490214588450393	0.375490214588448475	0.517427676570348322E-14
0.375490214588465310	0.375490214588448475	0.448314122628451832E-13
0.489301710372341299	0.489301710372335835	0.111747992708182397E-13
0.489301710372353123	0.489301710372335835	0.353395936330952460E-13
0.624215876820449545	0.624215876820443830	0.911526478657739988E-14
0.624215876820457945	0.624215876820443830	0.226103031413395523E-14

(The rest is omitted.)

Figure 4: Output of the block RIC,

N=225, block size=5, the largest multiplicity=15 .

7 Conclusions

PRO and RIC for the simple Lanczos algorithm are extended for the block Lanczos algorithm. The block SO and the block PRO are not reliable when the block size is smaller than the multiplicity. Even in this case, however, the block RIC indicates reliable eigenvalues among computed ones. Therefore, the block RIC is found useful for practical problems.

References

[1] H. Imai, M. Natori and E. Kawamura : A New Reorthogonalization in the Lanczos Algorithm, J. Info. Proc., 14 (1991), pp.56-59.

[2] B.N. Parlett : *The Symmetric Eigenvalue Problem*, Prentice-Hall, 1980.

[3] B.N. Parlett and D.S. Scott : The Lanczos Algorithm with Selective Orthogonalization, Math. Comp., 33(1979), pp.217-238.

[4] H.D. Simon : The Lanczos Algorithm with Partial Reorthogonalization, Math. Comp., 42(1984), pp.115-142.

Hiroko Iguchi, Makoto Natori and Hitoshi Imai

University of Tsukuba
Institute of Information Sciences and Electronics
Ibaraki 305, Japan

INSTITUTE OF INFORMATION SCIENCES AND ELECTRONICS
UNIVERSITY OF TSUKUBA
TSUKUBA-SHI, IBARAKI 305 JAPAN

REPORT DOCUMENTATION PAGE	REPORT NUMBER ISE-TR-92-100
TITLE REORTHOGONALIZATION IN THE BLOCK LANCZOS ALGORITHM	
AUTHOR(S) Hiroko Iguchi, Makoto Natori and Hitoshi Imai University of Tsukuba Institute of Information Sciences and Electronics Ibaraki 305, Japan	
REPORT DATE November 30, 1992	NUMBER OF PAGES 15
MAIN CATEGORY Numerical Analysis	CR CATEGORIES
KEY WORDS reorthogonalization, block Lanczos, eigenvalue problem	

ABSTRACT

The reorthogonalization is a significant problem in the Lanczos algorithm. Some satisfactory reorthogonalization methods have already been developed in the simple Lanczos algorithm. However, there are few reorthogonalizations in the block Lanczos algorithm. In this paper, two reorthogonalization methods PRO and RIC for the simple Lanczos algorithm are extended for the block Lanczos algorithm. Numerical results show that RIC gives satisfactory results, while PRO is superior to RIC in the computation time but inferior in the reliability.

SUPPLEMENTARY NOTES