HOW FAR SHOULD YOU GO WITH THE LANCZOS PROCESS?

by

W. Kahan and B.N. Parlett

Memorandum No. UCB/ERL M78/48

18 July 1978

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

HOW FAR SHOULD YOU GO WITH THE LANCZOS PROCESS?†

W. KAHAN and B.N. PARLETT
Department of Mathematics
and
Computer Science Division
Department of Electrical Engineering
and Computer Sciences
University of California at Berkeley
Berkeley, California 94720

Abstract

The Lanczos algorithm can be used to approximate both the largest and smallest eigenvalues of a symmetric matrix whose order is so large that similarity transformations are not feasible. The algorithm builds up a tridiagonal matrix row by row and the key question is when to stop. An analysis leads to a stopping criterion which is inspired by a useful error bound on the computed eigenvalues.

1. INTRODUCTION

The Lanczos algorithm came back into prominence about five years ago [7,8] as the most promising way to compute a few of the eigenvectors of very large symmetric matrices. To be specific we think of computing the p smallest (or largest) eigenvalues of the n×n symmetric matrix A together with the associated eigenvectors. Typical values are $p \approx 3$, $n = 1000$.

The algorithm must be provided with a starting vector q_1 and then it builds up, one column per step, two auxiliary matrices. After j steps it will have produced j×j symmetric tridiagonal matrix T_j and an $n \times (j+1)$ matrix $Q_{j+1} = (q_1, q_2, \ldots, q_{j+1})$. Let $\theta_1, \ldots, \theta_p$ denote the p extreme eigenvalues of T_j.

What we want is that the $\{\theta_i\}$ should be good approximations to the wanted eigenvalues of A ($\alpha_1, \ldots, \alpha_p$, say) and we ask the following questions. Will the θ_i inevitably improve as j increases? How much or rather how little work is needed to compute an a posteriori bound on the errors in the θ_i?

We still do not know how best to use the Lanczos process. The surprising fact is that the θ_i are sometimes correct to 3 or 4 decimal figures even when j is as small as $\sqrt{n}$. What makes the method interesting is that such good fortune cannot be guaranteed. It depends on q_1, on the spread of the α_i, and on the precision of the arithmetic operations.

†The authors are pleased to acknowledge partial support from Office of Naval Research Contract N00014-69-A-0200-1017.

It is a pleasure to acknowledge the excellent pioneering work of Paige in his doctoral thesis [8] and the studies by Golub [2] and others [1] on the block version of the method.

Sparsity of A plays a simple but crucial role here. It permits the computation of Av, for any n-vector v, in only wn basic operations, where w is the average number of nonzero elements per row. Of more importance sparsity discourages the use of explicit similarity transformations which invariably destroy sparsity even when they preserve bandwidth. The only knowledge of A which the Lanczos algorithm demands is how A acts on selected vectors, an attractive property for sparse problems.

To get round some of the difficulties posed by the gaps in our understanding it has been proposed that the Lanczos algorithm be used iteratively as follows. Start with $q_1^{(1)}$ and run the algorithm for $j = j^{(1)}$ steps, compute the p best eigenvector approximations available from $T_j^{(1)}$ and $Q_j^{(1)}$ and take a weighted combination of them as a new starting vector $q_1^{(2)}$, run Lanczos for $j^{(2)}$ steps, compute a new $q_1^{(3)}$, and so on. The process can be continued until the $\{\theta_i\}_1$ computed at the end of each run converge to the desired number of figures.

Is this a good idea? How should the $j^{(i)}$ be chosen? We do not have definitive answers to the questions raised here but we do present a computable a posteriori error bound, an estimate for loss of orthogonality among the columns of Q_j, a tentative stopping criterion, and a useful way of analyzing the algorithm.

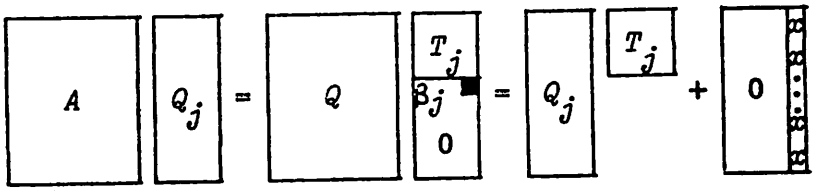
Standard Householder matrix conventions will be followed except that M* denotes the conjugate transpose of M and 1 denotes the identity matrix.

2. ERROR BOUNDS FOR THE EXACT LANCZOS ALGORITHM

When executed in exact arithmetic the matrix $Q_j \equiv (q_1,\ldots,q_j)$, which is generated column by column, is orthonormal:

$$Q_j^* Q_j = 1 \,, \tag{1}$$

and is part of an orthogonal matrix Q which reduces A to tridiagonal form. The algorithm, or rather its j-th step, is completely specified by the single matrix relation

(2) 

$$= Q_j T_j + r_j e_j^* \ ,$$

where $e_j^* = (0,\ldots,0,1)$ has j elements and $\|r_j\|^2 \equiv r_j^* r_j = \beta_j^2$. The residual vector

$$r_j \equiv \beta_j q_{j+1} = (AQ_j - Q_j T_j) e_j$$

is always uniquely determined by A, Q_j, and T_j. An essential characteristic of the Lanczos process is that the residual matrix $AQ_j - Q_j T_j$ has all its substance concentrated in its final column. An atractive feature of the process, not obvious from (2), is that only the two most recent columns of Q_k need be kept in fast memory at the k-th step, so q_{k-2} can be put out to secondary storage. Sometimes the early q vectors are discarded but this makes it very difficult to estimate the accuracy of our approximations. Equation (2) is a nice compact way of remembering the algorithm.

Suppose now that the Lanczos process is halted at the end of the j-th step. The approximations to the desired eigenvectors and values of A are made as follows. The p extreme eigenvalues of T_j are computed along with their normalized eigenvectors. We suppress the dependence on j and write

$$T_j c_i = c_i \theta_i \ , \quad i = 1,\ldots,p \ , \text{ i.e. } T_j C = C\Theta$$

where $$C = (c_1,\ldots,c_p) \ , \quad \Theta = diag(\theta_1,\ldots,\theta_p) \ .$$

Then we compute the Ritz vectors $v_i = Q_j c_i$, $i = 1,\ldots,p$. When Q_j is orthonormal these Ritz vectors are the best approximate eigenvectors, in the sense of residuals, that can be made from linear combinations of the columns of Q_j and the θ_i are the best approximations to the corresponding eigenvalues. There is a considerable body of knowledge concerning the accuracy of the exact Ritz approximations. The Kaniel-Paige theory [3] gives a priori error bounds on θ_i and v_i which tell how accuracy increases with j. Moreover there are refined a posteriori error bounds which can be computed when the algorithm is terminated. However even with exact calculations there is no inexpensive, feasible criterion which can tell us the best j at which to stop.

Let us look at some simple a posteriori bounds. They employ the spectral matrix norm $\|B\| \equiv \max\|Bv\|/\|v\|$, $v \neq 0$.

THEOREM 1 (Kahan, 1967). Let H be any p×p symmetric matrix with eigenvalues θ_i and let S be n×p and orthonormal. Then there are p eigenvalues $\alpha_{i'}$ of A such that, for $i = 1,\ldots,p$,

$$|\theta_i - \alpha_{i'}| \leq \|AS - SH\| .$$

We do not know which of A's eigenvalues are the $\alpha_{i'}$.

The proof is based on the Weyl/Wielandt monotonicity theorem and is given in [3].

We are not interested in all of T_j's eigenvalues, only p of them. In fact we have

$$T_j C = C\Theta ,$$

Now apply Theorem 1 with $S = Q_j C$ to obtain the following result.

COROLLARY. Let $AQ_j - Q_j T_j = r_j e_j^*$ and $T_j C = C\Theta$ where Q_j and C are orthonormal and $\|r_j\| = \beta_j$. Then there are p eigenvalues $\alpha_{i'}$ of A such that for $i = 1,\ldots,p$,

$$|\theta_i - \alpha_{i'}| \leq \beta_j \|e_j^* C\| .$$

Thus when the last elements of some normalized eigenvectors of T_j are small good accuracy is obtained for their eigenvalues even when β_j is not small. Unfortunately we have no easy way of guaranteeing which eigenvalues of A are being approximated. This is an intrinsic limitation of the Lanczos method and will not be discussed in detail here.

The best way we know of testing which of A's eigenvalues $\{\alpha_k\}$ are being approximated by the θ_i ($i = 1,\ldots,p$) is to perform a triangular factorization of $A - \theta_i$ into LDL^T and count the number ν_i of negative elements of D. By Sylvester's Inertia theorem there are ν_i of the α's less than θ_i.

3. PRACTICAL ERROR BOUNDS AND TERMINATION CRITERIA

From now on we let Q_j, T_j, etc. stand for the quantities stored in the computer. Because of roundoff error Q_j will not be orthonormal and the residual matrix will not have all its substance in the last column. Fortunately Theorem 1 can

be generalized.

THEOREM 2 (Kahan, 1967). Let H be any p×p matrix with eigenvalues θ_i, let S be any n×p matrix of full rank p; then there are p eigenvalues $\alpha_{i'}$ of A such that, for $i = 1,\ldots,p$,

$$|\theta_i - \alpha_{i'}| \leq \sqrt{2}\,\|AS-SH\| \cdot \|(S^*S)^{-1/2}\| \ .$$

Note that $\|(S^*S)^{-1/2}\|$ is the reciprocal of S's smallest singular value $\sigma_1(S) \equiv \sqrt{\lambda_1(S^*S)}$. A proof may be found in [3]. The factor $\sqrt{2}$ is believed to be superfluous.

In order to use Theorem 2 we observe that Q_j and T_j actually satisfy

$$AQ_j - Q_jT_j = F_j + r_je_j^* \tag{3}$$

where F_j accounts for roundoff error.

COROLLARY. Let $T_jC = C\Theta$; then there are p eigenvalues $\alpha_{i'}$ of A such that, for $i = 1,\ldots,p$,

$$|\theta_i - \alpha_{i'}| \leq \sqrt{2}(\|F_j\| \cdot \|C\| + \beta_j\|e_j^*C\|)/\sigma_1(Q_jC) \ .$$

PROOF. Apply Theorem 2 with $S = Q_jC$. □

In order to use the corollary an upper bound is needed on $\|F_j\|$ and a lower bound on the singular values of Q_jC. With the best of current techniques C will be very close to orthonormal. In the sense of quadratic forms,

$$0 \leq C^*C \leq 1 \tag{4}$$

and hence, by the Cauchy interlace inequalities

$$\sigma_1(Q_j) \leq \sigma_1(Q_jC) \ .$$

Error analyses in [5] and [8] show that $\|F_j\|$ is small, like roundoff in $\|A\|$, and so the term $\|F_j\| \cdot \|C\|$ is completely dominated by $\beta_j\|e_j^*C\|$ for all realistic values of j. Thus in practice the bound in Corollary 1 is degraded by the factor $1/\sigma_1$.

Corollary 2 assures us that it is worthwhile to continue the Lanczos algorithm even after orthogonality among the q_i has been lost provided that they are still linearly independent. Experience suggests that good approximation to

internal eigenvalues α_i can be obtained by continuing the algorithm indefinitely. However after $\sigma_1(Q_j) = 0$, i.e. when linear independence is lost, the user is faced with the identification problem, namely to say which of the θ_i do not approximate any of the α's and so are spurious. This problem can often be solved in particular applications but it is very troublesome to devise a procedure which will make this identification for the general case.

In order to escape this difficulty we would like to stop immediately $\sigma_1(Q_j) < 1/2$. However it is out of the question to compute this number at each step, or even update it. Instead we develop a computable lower bound on $\sigma_1(Q_j)$ and stop a Lanczos run as soon as it vanishes. If the approximations are not satisfactory we can form a new starting vector from the Ritz vectors and start another run.

Since C is not computed at each step there is no simple way of bounding $\sigma_1(Q_jC)$ rather than $\sigma_1(Q_j)$.

4. MONITORING LOSS OF ORTHOGONALITY

Because of finite precision arithmetic Q_j will not be orthonormal. Let us write

$$\|1 - Q_j^*Q_j\| \leq \kappa_j$$

and find some specific expressions for κ_j. Note that

$$\|Q_j\|^2 = \|Q_j^*Q_j\| = \|1 - (1-Q_j^*Q_j)\| \leq 1+\kappa_j ,$$

and, in the sense of quadratic forms

$$1 - \kappa_j \leq Q_j^*Q_j \leq 1+\kappa_j .$$

Hence, while $\kappa_j < 1$,

$$\sqrt{1-\kappa_j} \leq \sqrt{\lambda_1(Q_j^*Q_j)} \equiv \sigma_1(Q_j) \leq \sigma_1(Q_jC) .$$

Our problem is thus reduced to finding a computable bound κ_j. This can be accomplished in a variety of ways. Here is one of the simplest. Observe that

$$1 - Q_{j+1}^*Q_{j+1} = \begin{pmatrix} 1-Q_j^*Q_j & -Q_j^*q_{j+1} \\ -q_{j+1}^*Q_j & 1-\|q_{j+1}\|^2 \end{pmatrix} .$$

Let

$$\|Q_j^*q_{j+1}\| \leq \zeta_j .$$

Then, by considering the quadratic forms we see that

$$\|1-Q_{j+1}^*Q_{j+1}\| \le \left\| \begin{pmatrix} \kappa_j & \zeta_j \\ \zeta_j & \kappa_1 \end{pmatrix} \right\| .$$

Here we have used the fact that, by definition,

$$\|1-Q_1^*Q_1\| = |1 - \|q_1\|^2| \le \kappa_1$$

but, actually, κ_1 is the bound on the error in normalizing <u>any</u> vector to the given precision, including q_{j+1}, and is taken as known.

Any computable bound ζ_j will yield a corresponding definition of κ_j, namely

$$\kappa_{j+1} \equiv \left\| \begin{pmatrix} \kappa_j & \zeta_j \\ \zeta_j & \kappa_1 \end{pmatrix} \right\|$$

$$= \frac{1}{2}\{\kappa_j + \kappa_1 + \sqrt{(\kappa_j-\kappa_1)^2+4\zeta_j^2}\} .$$

Since κ_1 is known we focus our attention on ζ_j and $Q_j^*q_{j+1}$.

5. AN EXPRESSION FOR $Q_j^*r_j$

We write

$$T_j = \begin{bmatrix} \alpha_1 & \beta_1 & & \bigcirc \\ \beta_1 & \alpha_2 & \beta_2 & \\ & \bigcirc & \ddots & \ddots \\ & & & \beta_{j-1} \ \alpha_j \end{bmatrix}$$

and let $F_j = (f_1,\ldots,f_j)$. Our bound ζ_j comes from a useful expression for $Q_j^*r_j$.

LEMMA 1. Let Q_j and T_j satisfy (H): $AQ_j-Q_jT_j = F_j + r_je_j^*$. Then

$$Q_j^*r_j = [(1-Q_j^*Q_j)T_j - (1-e_je_j^*)T_j(1-Q_j^*C_j)]e_j + F_j^*q_j + (q_j^*Aq_j-\alpha_j)e_j - Q_j^*f_j .$$

PROOF.

$$Q_j^*r_j = Q_j^*(AQ_j-Q_jT_j-F_j)e_j \quad , \text{ using (H)},$$

$$= [(AQ_j^*)Q_j - Q_j^*Q_jT_j]e_j - Q_j^*f_j \quad , \text{ using } A^* = A,$$

$$= [T_j Q_j^* Q_j + F_j^* Q_j + e_j r_j^* Q_j - Q_j^* Q_j T_j] e_j - Q_j^* f_1 \ , \quad \text{(H) again,}$$
$$= [-T_j(1-Q_j^* Q_j) + (1-Q_j^* Q_j) T_j] e_j + F_j^* q_j + e_j r_j^* Q_j e_j$$
$$- Q_j^* f_j \ .$$

From the first line above

$$e_j^* Q_j^* r_j = q_j^* A q_j - e_j^* Q_j^* Q_j T_j e_j$$
$$= q_j^* A q_j - \alpha_j + e_j^* (1-Q_j^* Q_j) T_j e_j \ .$$

On transposing and rearranging the desired expression is obtained. □

In practice

$$q_{j+1} = r_j/\beta_j + g_j$$

where g_j accounts for roundoff error in the division by β_j and is always insignificant. In any case

$$Q_j^* q_{j+1} = Q_j^* r_j/\beta_j + Q_j^* g_j \ .$$

We observe that $\|Q_j^* q_{j+1}\|$ will not be small, like roundoff in 1 whenever, because of past errors, $\|Q_j^* r_j\| \doteq \|Q_j^*\| \|r_j\|$. In exact arithmetic $Q_j^* r_j = 0$ and this property fails in practice to the extent that there is cancellation in forming r_j; the resulting large relative error in the small vector r_j becomes significant when r_j is divided by β_j to produce q_{j+1}. As Paige points out in [7] this happening must be seen in perspective. We <u>want</u> cancellation to occur in the formation of r_j because cancellation is the harbinger of convergence of the θ_1 to an α.

6. A COMPUTABLE BOUND ON $Q_j^* q_{j+1}$

The expression for $Q_j^* r_j$ given in Lemma 1 is best split into two parts

$$b_j \equiv [(1-Q_j^* Q_j) T_j - (1-e_j e_j^*) T_j (1-Q_j^* Q_j)] e_j \ ,$$
$$d_j \equiv F_j^* q_j + (q_j^* A q_j - \alpha_j) e_j - Q_j^* f_j \ .$$

It turns out that our bounds on $\|b_j\|$ overwhelm those on $\|d_j\|$ as soon as $j > 4$. To bound $\|d_j\|$ a detailed error analysis of the Lanczos algorithm is needed. There is little incentive to present this because the resulting bound makes an insignificant contribution to ζ_j. We shall simply quote the results in [5] which are quite similar to those in [7].

The results are stated in terms of

$$\|A\|_E \equiv (\sum_{\mu}\sum_{\nu}|a_{\mu\nu}|^2)^{1/2} = [trace(A^*A)]^{1/2}$$

which arises in the general bound on the error in computing Av. However $\|A\|_E$ is just a convenient bound on the more realistic but less accessible quantity $\|\tilde{A}\|$ where $\tilde{A} = (|a_{ik}|)$. For large sparse matrices $\|\tilde{A}\|$ is usually much smaller than $\|A\|_E$. The following estimates are crude but adequate for our purposes: for $i \le j$,

$$\|f_i\| + \beta_i\|g_i\| < \kappa_1\|A\|_E ,$$
$$|q_i^* A q_i - \alpha_i| < \kappa_1\|A\|_E ,$$
$$\|Q_i\|^2 < 1+\kappa_i ,$$
$$\|F_j\|_E < \sqrt{j}\ max\ \|f_i\| , \quad i \le j ,$$
$$\kappa_1 < 2(n+6)\varepsilon ,$$

where ε is the precision of the arithmetic facilities. Finally we quote

LEMMA 2. $\|d_j\| \le (\sqrt{j}+3+\kappa_j)\|A\|_E$.

Turning to $\|b_j\|$ we find

LEMMA 3. If $\|Q_i^* q_{i+1}\| \le \zeta_i$, $i < j$, then

$$\|b_j\|^2 \le \{\|(T_{j-1}-\alpha_j)\|\zeta_{j-1} + \beta_{j-1}(\zeta_{j-2}+2\kappa_1)\}^2 + \{(3j+1)\kappa_1\|A\|_E\}^2$$

PROOF. Partition the terms in b_j and observe

$$(1-Q_j^*Q_j)T_j e_j = \begin{bmatrix}\{-Q_{j-1}^* q_j\}\\ 1-\|q_j\|^2\end{bmatrix}\alpha_j + \begin{bmatrix}-Q_{j-2}^* q_{j-1}\\ 1-\|q_{j-1}\|^2\\ -q_j^* q_{j-1}\end{bmatrix}\beta_{j-1} ,$$

$$T_j(1-Q_j^*Q_j)e_j = \begin{bmatrix}-T_{j-1}Q_{j-1}^* q_j + e_{j-1}\beta_{j-1}(1-\|q_j\|^2)\\ \alpha_j(1-\|q_j\|^2) - \beta_{j-1}q_{j-1}^* q_j\end{bmatrix} .$$

The factor $(1-e_j e_j^*)$ simply annihilates the bottom element. Recall that $|1-\|q_i\|^2| < \kappa_1$, by definition of κ_1. Hence the top part of b_j is

$$(T_{j-1}-\alpha_j)Q_{j-1}^* q_j - \beta_{j-1}\begin{pmatrix} Q_{j-2}^* q_{j-1} \\ 0 \end{pmatrix} + \beta_{j-1} e_{j-1}(\|q_j\|^2 - \|q_{j-1}\|^2)$$

and the bottom element is

$$(1-\|q_j\|^2)\alpha_j - \beta_{j-1} q_j^* q_{j-1} .$$

By definition of the basic Lanczos steps

$$q_{j+1}\beta_j = Aq_j - \alpha_j q_j - \beta_{j-1} q_{j-1} - f_j + \beta_j g_j$$

then, subtracting and adding α_j,

$$\begin{aligned} |q_j^* q_{j+1}\beta_j| \le\ & |q_j^* A q_j - \alpha_j| + |1 - \|q_j\|^2| \cdot |\alpha_j| + |\beta_{j-1} q_j^* q_{j-1}| \\ & + |q_j^*(f_j + \beta_j g_j)| . \end{aligned}$$

Now suppose that $|q_i^* q_{i+1}\beta_i| \le \psi_i$, $i \le j$, then the error bounds quoted above Lemma 2 yield

$$\begin{aligned} \psi_j &\le 3\kappa_1 \|A\| + \psi_{j-1} , \\ &\le 3j\kappa_1 \|A\| , \text{ since } \psi_1 \le 3\kappa_1 \|A\| . \end{aligned}$$

Straightforward application of the triangle inequality gives the desired bound. □

Adding these results together we define

$$\begin{aligned} \omega_j \equiv\ & \{[\|(T_{j-1}-\alpha_j)\|\zeta_{j-1} + \beta_{j-1}(\zeta_{j-2}+2\kappa_1)]^2 \\ & + (3j+1)^2 \kappa_1^2 \|A\|_E^2\}^{1/2} + (\sqrt{j}+3+\kappa_j)\|A\|_E , \\ \zeta_j \equiv\ & \omega_j/\beta_j + \sqrt{1+\kappa_j}\,\varepsilon . \end{aligned}$$

Then, by induction,

LEMMA 4. If $\zeta_{-1} = \zeta_0 = 0$, $\|Q_i^* q_{i+1}\| \le \zeta_i$, $i < j$ then

$$\|Q_j^* q_{j+1}\| \le \zeta_j .$$

PROOF. Apply Lemmas 2 and 3 and the error bounds to

$$Q_j^* q_{j+1} = Q_j^* r_j/\beta_j + Q_j^* g_j . \quad \square$$

Let us summarize the procedure. During the j-th step of a Lanczos run we compute

$$\alpha_j,\ r_j,\ \beta_j,\ q_{j+1}\ ,$$

then

$$\omega_j,\ \zeta_j,\ \text{and}\ \kappa_{j+1}\ .$$

While $\kappa_j < 1$ the algorithm proceeds to step (j+1).

The extra work in updating κ_j is very slight and is dominated by the computation of $\|(T_{j-1}-\alpha_j)\|$. This may be bounded as follows.

$$\begin{aligned}\|(T_{j-1}-\alpha_j)\| &\le \max_{i<j}(\beta_{i-1} + |\alpha_i-\alpha_j| + \beta_i)\ ,\\ &\le \bar{\beta} + \max_{k<j}|\alpha_k-\alpha_j|\ ,\\ &\le \bar{\beta} + \max(|\underline{\alpha}-\alpha_j|, |\bar{\alpha}-\alpha_j|)\ ,\end{aligned}$$

where

$$\underline{\alpha} = \min \alpha_i,\quad \bar{\alpha} = \max \alpha_i,\quad \bar{\beta} = \max_{i\le k}(\beta_{i-1}+\beta_i)\ .$$

Note that the third inequality involves no searching over j elements provided that $\underline{\alpha}$, $\bar{\alpha}$, and $\bar{\beta}$ are updated at each step; $\underline{\alpha} = \min(\underline{\alpha},\alpha_j)$, $\bar{\alpha} = \max(\bar{\alpha},\alpha_j)$, $\bar{\beta} = \max(\bar{\beta},\beta_{j-2}+\beta_{j-1})$. The second inequality costs (j-1) comparisons and is tighter. The first inequality is $\|(T_{j-1}-\alpha_j)\|_\infty$, except for the presence of β_{j-1}, and this can never exceed $\|(T_{j-1}-\alpha_j)\|$ by more than a factor of $\sqrt{2}$.

A more complicated procedure for monitoring loss of orthogonality is described in [5]. $1-Q_j^*Q_j$ is majorized by a j×j matrix W_j, called the scoreboard. The triangular factorization of $1-W_j$ is updated at each step and the Lanczos run continues until a nonpositive diagonal element appears. The cost is approximately $j^2/2$ arithmetic operations and, more seriously, j^2 storage locations.

The quantity κ_j grows exponentially and its use will certainly terminate Lanczos runs prematurely. This is not necessarily inefficient for the following reason. The Lanczos algorithm yields monotonically improving approximations. It thus can be used for calculations to low accuracy as well as high. It is important not to give the user unwanted figures when the extra cost is significant. If Lanczos is run until $|q_1^*q_{j+1}| > 0.1$ (say) then θ_1 may already have converged to too many figures.

In practice $\|A\|_E$ may not be readily available and the

estimate $\|(T_{j-1}-\alpha_j)\| + |\alpha_j|$ is used in its place. Sometimes the bound on $\|d_j\|$ is omitted for simplicity.

7. BEHAVIOR OF THE SPECTRUM OF T_j

In Figure 1 we show the lowest five eigenvalues of T_j, for $j = 10,20,\ldots,60$, in a Lanczos run on a matrix A with eigenvalues $\alpha_i = i$, $i = 1,2,\ldots,253$. Also shown is $\sigma_i(Q_j)$ and $\sqrt{1-\kappa_j}$. The starting vector q_1 was chosen to be rich in the first four eigenvectors.

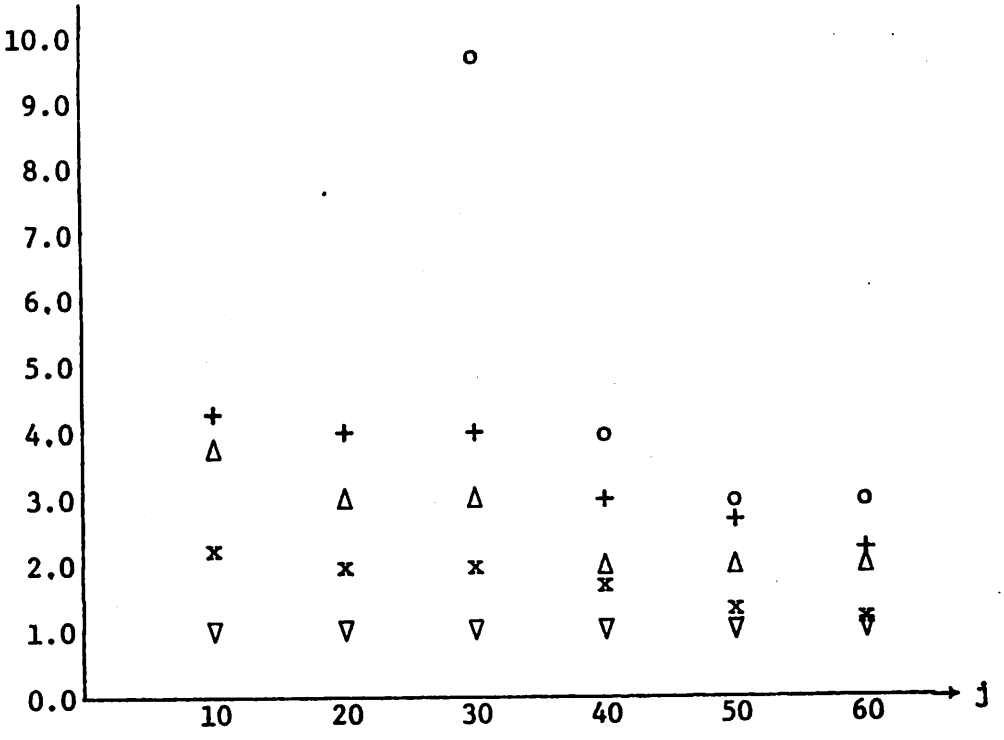


Fig. 1a. 5 smallest eigenvalues of T_j

Three phases may be distinguished in the run. Note that $p = 4$.

Early Phase: θ_i is not an accurate approximation to α_i for $i = 1,\ldots,p$.

Middle Phase: One or more of the θ_i have converged to working accuracy and there is still a one-one correspondence between θ_i and α_i for $i = 1,\ldots,p$.

Late Phase: The one-one correspondence has been lost. Among the θ_i occur multiple approximations to various α's as well as spurious values close to no α.

If q_1 had been chosen to be rich in eigenvectors 1, 3, and 4 the picture would have been more complicated. If p is

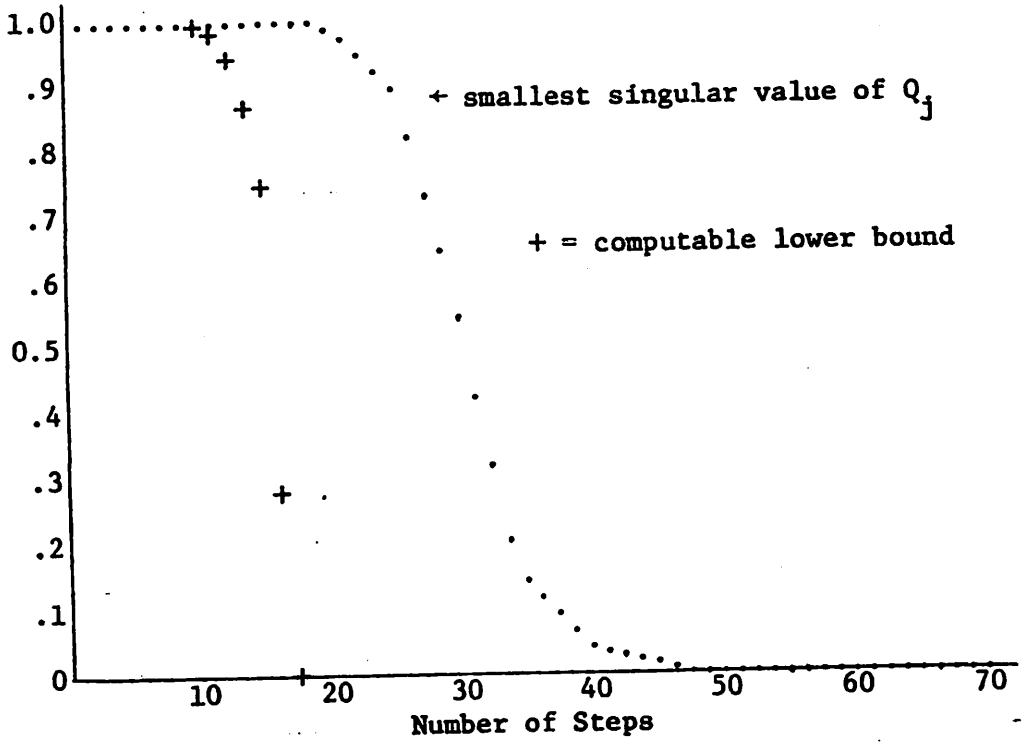


Fig. 1b.

too large then there may be multiple approximations to α_1 before any θ has converged to α_p.

Corollary 2 in Section 2 implies that the late phase cannot begin until linear independence is lost to working accuracy. Paige's error analyses [7] show that Phase 1 ends before orthogonality is lost to working accuracy.

It is usual for an outer eigenvalue to converge before an inner one but this cannot be taken for granted. The outcome depends on q_1.

We recommend that Lanczos algorithms be tailored to the problem in hand. The urge to write a universal Lanczos program should be resisted, at least until the process is better understood.

REFERENCES

[1] Jane Cullum, W.E. Donath, A Block Lanczos Generalization of the Symmetric S-STEP Lanczos Algorithm, Technical Report RC 4845 (X 21570), Mathematics Department, IBM Research Center, Yorktown Heights, New York.

[2] G.H. Golub, R. Underwood and J.H. Wilkinson, The Lanczos Algorithm for the Symmetric Ax = λBx Problem, Report CS-72-270, Stanford University, 1972.

[3] W. Kahan, Inclusion Theorems for Clusters of Eigenvalues of Hermitian Matrices, Computer Science Report, University of Toronto, 1967.

[4] W. Kahan and B. Parlett, An Analysis of Lanczos Algorithms for Symmetric Matrices, Memorandum ERL-M467, Electronics Research Laboratory, University of California, Berkeley, 1974.

[5] S. Kaniel, Estimates for Some Computational Techniques in Linear Algebra, Mathematics of Computation 20 (1966), 369-378.

[6] C. Lanczos, An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators, Journal of Research of the National Bureau of Standards 45 (1950), 255-282.

[7] C.C. Paige, Practical Use of the Symmetric Lanczos Process with Re-Orthogonalization, BIT 10 (1970), 183-195.

[8] C.C. Paige, The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices, Ph.D. Thesis, London University, Institute for Computer Science, Technical Note.

[9] C.C. Paige, Computational Variants of the Lanczos Method for the Eigenproblem, Journal of IMA 10 (1972), 373-381.

[10] J.H. Wilkinson, The Algebraic Eigenvalue Problem, O.U.P., 1965.

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*
Computer Science Division
Electronics Research Laboratory
University of California at Berkeley 94720

2a. REPORT SECURITY CLASSIFICATION

2b. GROUP

3. REPORT TITLE
HOW FAR SHOULD YOU GO WITH THE LANCZOS PROCESS?

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
Scientific Final

5. AUTHOR(S) *(First name, middle initial, last name)*
W. Kahan and B. Parlett

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
September 1975	14	10

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
ONR-N00014-69-A-0200-1017	X-748
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
d.	

10. DISTRIBUTION STATEMENT
Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Mathematics Branch Office of Naval Research Washington, D.C. 20360

13. ABSTRACT

The Lanczos algorithm can be used to approximate both the largest and smallest eigenvalues of a symmetric matrix whose order is so large that similarity transformations are not feasible. The algorithm builds up a tridiagonal matrix row by row and the key question is when to stop. An analysis leads to a stopping criterion which is inspired by a useful error bound on the computed eigenvalues.

DD FORM 1 NOV 65 1473 (PAGE 1)

Unclassified

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
eigenvalues						
eigenvectors						
Lanczos' algorithm						
sparse symmetric matrices						
stopping criteria						