AN ANALYSIS OF LANCZOS ALGORITHMS FOR SYMMETRIC MATRICES

by

W. Kahan and B. Parlett

Memorandum No. ERL-M467

September 1974

AN ANALYSIS OF LANCZOS ALGORITHMS FOR SYMMETRIC MATRICES

by

W. Kahan and B. Parlett

Memorandum No. ERL-M467

September 1974

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

AN ANALYSIS OF LANCZOS ALGORITHMS FOR SYMMETRIC MATRICES[†]

W. Kahan[*] and B. Parlett[*]

September 1974

ABSTRACT

The Lanczos algorithm is presented as a way of generating bases for a sequence of Krylov subspaces. Explicit expressions are given for the departure of the bases from orthogonality. These relations enable one to comprehend the behavior of the algorithm in practice with a minimum of conventional error analysis.

In particular this approach sheds light on the central, and difficult problem of ascertaining the right moment to stop the algorithm.

Reorthogonalization and block versions are also examined.

Key Words: eigenvalues/eigenvectors, invariant subspaces, symmetric and Hermitian matrices,large, sparse matrices, tridiagonal form, Lanczos' algorithm, block Lanczos, reorthogonalization.

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

[*] Departments of Mathematics and Electrical Engineering and Computer Sciences, University of California, Berkeley.

CONTENTS

1. Introduction

Lanczos presented an algorithm for reducing a symmetric matrix to tridiagonal form in 1950, [5]. In the light of exact arithmetic it promised to be very effective. In practice it can compete neither in speed nor in accuracy with rival techniques which use a sequence of explicit orthogonal similarity transformations.

For a while interest in the method died.

When the attention of numerical analysts was turned to the problem of finding a few eigenvectors of matrices of huge order the algorithm was resurrected. However, the process was now seen in a new light, as a way of finding invariant subspaces, and for this purpose it can be very effective -- provided that it is implemented properly.

To be effective the computation must be stopped at the right moment. To determine this moment the algorithm must be understood in some detail. This poses a problem. Can an analysis of such a numerical process be both rigorous and readable?

The most thorough study of Lanczos' method that we know of was made by Paige in his doctoral thesis [7], and we see our analysis as a further development of his work [8,8.5]. Our approach is outlined in Section 3.

Reorthogonalization and block versions of the algorithm are also examined.

It would seem only proper to begin with a clear description of the algorithm itself. This we shall not do. In fact we postpone discussion of the details of the process as long as we can. Why? The quantities computed by the algorithm satisfy certain relationships. The forms of these relations are often independent of the specific details of the implementation. Moreover some properties of the algorithm can be, and are best understood at this level. Not to be ignored is the fact that the exposition is greatly simplified. Lemma 2 in Section 6 is an example of this approach.

Any readers who are interested in Computer Science may speculate on the influence of structured programming and top-down parsing on our thinking.

2. Notational Conventions

The exposition in the remaining sections is made smoother by bringing the standard definitions together at the start.

$\equiv$ denotes a definition.

Equation or Relation (m) refers to the one in the current section.

Equation (n.m) refers to Equation (m) in Section n.

Capital letters for matrices: A, B, Z, Λ, ...

Symmetric letters (about a vertical axis) for symmetric or Hermitian

matrices: A, H, M, ...

Lower case roman letters for (column) vectors: c, f, q, ...

Lower case greek letters for scalars: α, β, ...

S^*, x^* denote the conjugate transpose of S, x respectively.

$\mathcal{E}_n$ denotes Euclidean n-space (either real or complex).

S is orthonormal if $S^*S = 1$.

$\|v\| \equiv \sqrt{v^*v}$.

$\lambda_i[M] \equiv$ i-th eigenvalue of M, (increasing algebraically).

$\lambda_{-i}[M] \equiv$ i-th eigenvalue of M, (decreasing algebraically).

$\|S\| \equiv \max\limits_{v \neq 0} \|Sv\|/\|v\| = \sqrt{\lambda_{-1}[S^*S]}$.

$A - \sigma$ denotes $A - \sigma I$, where I is the identity matrix.

$\|S\|_E^2 \equiv \mathrm{trace}(S^*S) = \Sigma\lambda_i[S^*S]$ over all i.

All vectors have dimension n unless the contrary is stated.

$K_j(S) \equiv (S, AS, A^2S, \ldots, A^{j-1}S)$, a Krylov matrix.

Span $(B) \equiv$ the subspace generated by B's columns.

$\mathcal{K}_j(S) \equiv$ span $K_j(S)$, a Krylov subspace of $\mathcal{E}_n$ generated by S.

$$T_j \equiv \begin{bmatrix} \alpha_1 & \beta_1 & & & \bigcirc \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \cdot & \cdot & \\ & & \cdot & \cdot & \beta_{j-1} \\ \bigcirc & & & \beta_{j-1} & \alpha_j \end{bmatrix} \quad ; \ \beta_i > 0,\ i > 0,\ \beta_0 \equiv 0.$$

P^*AP is called the projection of A onto the range of P; $P^*P = 1$.

3. The Accuracy of Dwindling Eigenvectors

The goal is to compute a few eigenvectors, belonging to either end of the spectrum, of a real symmetric or complex Hermitian matrix A of huge order, perhaps 10,000. In other words, an invariant subspace of low dimension ν, say $\nu \leq 100$, is wanted. The Lanczos method begins with an initial subspace which we shall take to be one-dimensional for simplicity. The more general case is considered in the final sections. This starting vector q_1 is supposed not to be orthogonal to the invariant subspace $\mathcal{S}_\nu$ which is wanted.

Let us now say what the algorithm does without saying how it does it. After j steps we learn that, with exact arithmetic, A is orthogonally similar to a matrix

$$(1)\quad \tilde{A} \equiv \left[\begin{array}{c|c} T_j & \blacksquare \\ \hline \blacksquare & W_{n-j} \end{array}\right], \quad \blacksquare = \beta_j > 0,\ T_j \text{ defined above.}$$

where T_j, β_j have been computed but W_{n-j} has not. In fact T_j is the projection[††] of A onto the Krylov subspace $K(q_1)$. By computing the appropriate ν eigenvalues and eigenvectors of T_j one could obtain the best approximation to S_ν from $K_j(q_1)$. This is the Rayleigh-Ritz approximation.

There is a precise theory, the Kaniel-Paige[†] theory [4,7], which says exactly how bad the approximation can be. Even the worst case turns out to be very satisfactory provided only that q_1 is not too nearly orthogonal to S_ν . By their nature these bounds cannot be known in advance.

It is an essential feature of the Lanczos method that the Ritz approximation not be computed at each step. In its early versions the process was continued until $j = n$ at which point the one and only Rayleigh-Ritz "approximation" (with no errors in this case) was made. More appropriate for the task in hand is to stop immediately $\beta_j = 0$. It is also possible to stop if β_j is very small, like round off in $\|A\|$, because of the following result.

THEOREM 1 (Kahan). Let H be $j \times j$ and S be orthonormal and $n \times j$. Then there is a one-one correspondence between H's eigenvalues $\lambda_i[H]$, $i = 1, \ldots, j$, and a subset of A's spectrum, $\lambda_{i'}[A]$, such that for $i = 1, \ldots, j$

$$|\lambda_i[H] - \lambda_{i'}[A]| \leq \|AS - SH\| .$$

The proof is based on the Weyl/Wielandt monotonicity theorem and

[†] Kaniel began the study in [4]. However Paige, in [7], gives a much better exposition of the results and corrects the errors in [4]. A minor error is corrected in [0].

[††] Our definition is given in Section 2.

can be found in [3]. In our case $H = T_j$ and

$$(2) \qquad \|AS - ST_j\| = \beta_j \quad .$$

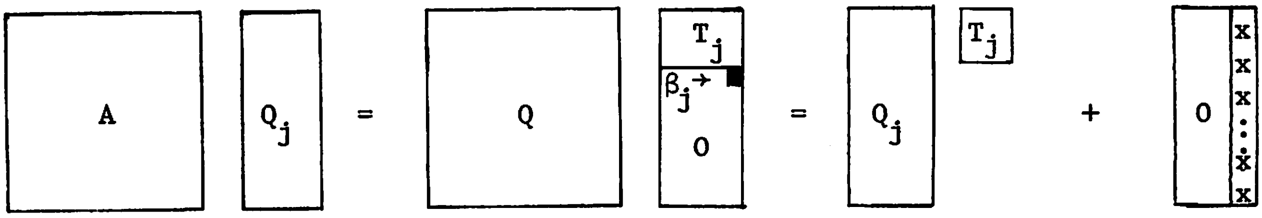
[To prove (2), let $Q = (S,s,P)$ be the unitary matrix such that $\tilde{A} = Q^*AQ$; then examine $AQ = Q\tilde{A}$]. We will not discuss the relationship of i' to i but remark that if the choice of q_1 is not too unfortunate then the set $\{i', i=1,\ldots,j\}$ will include the wanted eigenvalues.

This application of the Theorem 1 gives a sufficient condition for stopping, but one that is rarely encountered. Experience shows that some of T_j's eigenvalues furnish excellent approximations to eigenvalues of A <u>despite the absence of any small</u> β's.

In order to make a more illuminating application of Theorem 1 we must know more about the Lanczos Algorithm. After j steps in exact arithmetic it yields not only T_j but a matrix $Q_j \equiv (q_1,\ldots,q_j)$ whose columns span the subspace $K_j(q_1)$ and which satisfies

$$(3) \qquad Q_j^* Q_j = 1$$

and



$$= Q_j T_j + q_{j+1} \beta_j e_j^* \quad ,$$

where $e_j^* = (0,\ldots,0,1)$ has j elements. If $\beta_j = 0$ then q_{j+1} is not uniquely determined, yet the product

$$(4) \qquad q_{j+1}\beta_j \equiv r_j \equiv (AQ_j - Q_jT_j)e_j$$

is always fixed by the preceding quantities.

A key point here is that the residual $AQ_j - Q_jT_j$ has all its substance concentrated in its final column.

Not all of the eigenvalues and eigenvectors of T_j are of interest, only ν of them in fact. Now the normalized eigenvectors of any symmetric tridiagonal matrix enjoy rather special properties, see [7]; in particular there must be some whose last elements are tiny. The greater the order the tinier is this last element. The example in which this property is least pronounced is, we believe, a T_j with $\alpha_i = \alpha$, $\beta_i = \beta$ for all i. One eigenvector v is given by $v^{(k)} = \sqrt{\frac{2}{j+1}}\sin(\frac{kj\pi}{j+1})$, $k = 1,\ldots,j$, and

$$v^{(j)} = \sqrt{\frac{2}{j+1}}\sin(\frac{\pi}{j+1}) = O(j^{-3/2}) \quad .$$

It turns out that it is the eigenvalues belonging to these eigenvectors of T_j which are close to A's spectrum. For the moment we drop the index j.

COROLLARY. Let (i) $AQ - QT = re^*$, where $Q^*Q = 1$, T is tridiagonal, and $e^* = (0,\ldots,0,1)$,

(ii) $TP - PM = 0$, where $P^*P = 1$ and $\|e^*P\| \leq \eta << 1$.

Then

$$|\lambda_i[M] - \lambda_{i'}[A]| \leq \eta\|r\| = \eta\|AQ - QT\| \quad .$$

Proof. $$\begin{aligned} A(QP) - (QP)M &= AQP - QTP , \quad \text{by (ii)}, \\ &= (AQ - QT)P \\ &= re^*P \quad . \end{aligned}$$

Now apply Theorem 1 with $S = QP$, $H = M$. □

This result explains why a small β_j is not essential in order that <u>some</u> eigenvalues of T_j be very close to A's spectrum. There is no inference from what has been said so far that the dwindling eigenvectors of T_j correspond to the wanted eigenvectors of A. This is where the Kaniel-Paige theory comes in.

What will remain of these error bounds after an attack of roundoff errors?

4. When Should the Lanczos Process be Stopped?

In practice the situation is very different from the one described in the previous paragraph because rounding errors intervene. The approximate basis Q_j generated to span $K_j(q_1)$ will not be orthonormal. Consequently T_j is not A's projection and

$$AQ_j - Q_jT_j = r_je_j^* + \text{roundoff} \quad .$$

Although the Kaniel-Paige theory cannot be applied directly to T_j yet T_j is still tridiagonal and this coerces the elements of some of its eigenvectors just as before.

Fortunately the bounds derived from Theorem 1 can be extended to the practical situation.

THEOREM 2[†] (Kahan). Let H be $j \times j$, S be $n \times j$ and of full rank j; then there is a one-one correspondence between $\lambda_i[H]$ and $\lambda_{i'}[A]$, $i = 1,\ldots,j$ such that

$$|\lambda_i[H] - \lambda_{i'}[A]| \le \sqrt{2}\|AS - SH\| \cdot \|(S^*S)^{-1/2}\|$$

Note that $1/\|(S^*S)^{-1/2}\| = \sqrt{\lambda_1[S^*S]}$ is S's smallest singular value.

Now we apply Theorem 2 with $S = QP$, $H = M$ as in Corollary 1. The new bound is

$$|\lambda_i[M] - \lambda_{i'}[A]| \le \sqrt{2}(\beta_j + \text{roundoff})\eta_j \|(P^*Q^*QP)^{-1/2}\| \quad .$$

With the best of current techniques P will be very close to orthonormal. In the sense of quadratic forms

$$0 \le PP^* \lesssim 1$$

and hence, by the Cauchy inequalities,

$$\|(P^*Q^*QP)^{-1/2}\| \lesssim \|(Q^*Q)^{-1/2}\| \quad .$$

We conclude that the Lanczos process may be terminated as soon as $\|e_j^* P_j\|\beta_j$ becomes negligible, provided that the columns of Q_j are decidedly linearly independent. Without this proviso we can infer nothing from Theorem 2 about $|\lambda_i[M] - \lambda_{i'}[A]|$; try $S = zw^*$, $H = S^*AS$, $Az = z\alpha$, $\|z\| = \|w\| = 1$, where α is not one of the wanted eigenvalues.

When the Lanczos process is continued until $j > n$, without the appearance of a small β_j; independence is inevitably lost and it becomes necessary to identify which subset of T_j's spectrum is relevant to S_ν. At present we can see no point in breaking the linear independence barrier.

[†] This is Theorem 10 in [3]. The factor $\sqrt{2}$ is believed to be superfluous.

Our problem is reduced to monitoring loss of orthogonality among Q's columns.

5. Loss of Orthogonality and Independence

In order to study the Lanczos process we have replaced the original desired relation

$$(1) \qquad Q_n^* Q_n = 1 \ , \quad Q_n^* A Q_n = T_n \quad ,$$

specifying Q_n, by the intermediate relations

$$(2) \qquad Q_j^* Q_j = 1 \quad ,$$

$$(3) \qquad AQ_j = Q_j T_j + r_j e_j^* \ , \quad r_j = q_{j+1} \beta_j$$

which specify Q_j for $j = 1,2,\dots,n$.

Of course in deriving (2) and (3) from (1) it is inevitable that q_{j+1} is orthogonal to all the previous q's. This assumes that a matrix Q_n satisfying (1) always exists and, although not strictly necessary, it is more consistent to prove that, given A, then a matrix Q_j can be found satisfying (2) and (3). We shall not do this formally because the Lanczos algorithm itself shows how to determine α_j, β_j, and q_{j+1} to satisfy (3) when T_{j-1}, β_{j-1}, and Q_j are in hand. The key question is whether this q_{j+1} will be orthogonal to all the previous q_i, i.e. to Q_j. With an eye to later applications we answer this question. Recall that q_{j+1} is a multiple of r_j.

LEMMA 1. Let Q_j, T_j, and r_j satisfy (2) and (3) with α_j arbitrary. Then $Q_{j-1}^* r_j = 0$. If, in addition, $\alpha_j = q_j^* A q_j$ then $Q_j^* r_j = 0$.

Proof.

$$\begin{aligned} Q_j^* r_j &= Q_j^*(AQ_j - Q_j T_j)e_j \ , && \text{using (3),} \\ &= ((AQ_j)^* Q_j - T_j)e_j \ , && \text{using } A^* = A, \text{and (2),} \\ &= [(Q_j T_j + r_j e_j^*)^* Q_j - T_j]e_j \ , && \text{using (3) again,} \\ &= e_j r_j^* Q_j e_j \ , && \text{using } T_j^* = T_j, \\ &= e_j \overline{(e_j^* Q_j^* r_j)} \ , && \text{because this factor is a scalar,} \\ &= e_j \overline{[q_j^* A q_j - \alpha_j]} \ , && \text{using the first line of the proof.} \end{aligned}$$

□

Note that the algorithm does not explicitly force Q_{j+1} to satisfy (2), the property is inferred from the chain of reasoning given above.

Should one link break ... turn to Lemma 2 in Section 6.

The basic Lanczos Algorithm does force $r_j^* q_j$ to be like roundoff so that there is local orthogonality whilever β_j is not small.

At this point we make a standard but important change of notation. We forget the quantities that would be produced in exact arithmetic and let our symbols Q_j, T_j, r_j, etc. stand for the quantities stored in the computer under these names. Because of roundoff error the orthogonality relation (2) will not hold. In its place we write

$$(4) \quad \|1 - Q_j^*Q_j\| \le \kappa_j .$$

Later we shall determine some specific expressions for κ_j. Note that $\kappa_j \le \kappa_{j+1}$, and that

$$(5) \quad \|Q_j\|^2 = \|Q_j^*Q_j\| = \|1 - (1 \_ Q_j^*Q_j)\| \le 1 + \kappa_j ,$$

and, in the sense of quadratic forms,

$$(6) \quad 1 - \kappa_j \le Q_j^*Q_j \le 1 + \kappa_j ,$$

whence, if $\kappa_j < 1$,

$$(7) \quad \sqrt{1 - \kappa_j} \le \sqrt{\lambda_1(Q_j^*Q_j)} \equiv \text{smallest singular value of } Q_j .$$

Kahan's Theorem 2 (in Section 4) shows that loss of orthogonality is not catastrophic provided that Q_j retains *full rank*.

This result supplies a natural stopping criterion for the Lanczos method; namely, *stop as soon as* $\kappa_{j+1} > 7/8$, in which case $\kappa_j \le 7/8$ and, by Theorem 2

$$|\lambda_i[M_j] - \lambda_i[A]| \le \sqrt{2}\|r_j\|\|e_j^*P_j\|/\sqrt{1-\kappa_j} \le 4\beta_j\eta_j \quad ,$$

where η_j is a bound on the last row of the matrix of those eigenvectors of T_j which we chose to compute.

Our problem is thus reduced to computing a suitable bound κ_j. Consider

$$(8) \quad 1 - Q_{j+1}^* Q_{j+1} = \begin{pmatrix} 1 - Q_j^* Q_j & -Q_j^* q_{j+1} \\ -q_{j+1}^* Q_j & 1 - \|q_{j+1}\|^2 \end{pmatrix} .$$

By partitioning the quadratic form it can be seen that

$$(9) \quad \|1 - Q_{j+1}^* Q_{j+1}\| \le \left\| \begin{pmatrix} \kappa_j & \xi_j \\ \xi_j & \kappa_1 \end{pmatrix} \right\|$$

where ξ_j will be determined later and will satisfy

$$(10) \quad \|Q_j^* q_{j+1}\| \le \xi_j .$$

Note that, by definition,

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

but, in fact,

$$(11) \quad |1 - \|q_i\|^2| \le \kappa_1$$

for all i, because this bound is independent of i. In fact, if κ bounds the error in the length of any vector v which has been normalized to 1 in working precision arithmetic then

$$(12) \quad |1 - \|v\|^2| = |1 - \|v\|| \cdot |1 + \|v\|| \le \kappa(1 + 1 + \kappa) \equiv \kappa_1 .$$

If it is possible to compute a bound ξ_j as defined in (10) then the definition

$$(13) \quad \kappa_{j+1} \equiv \left\| \begin{pmatrix} \kappa_j & \xi_j \\ \xi_j & \kappa_1 \end{pmatrix} \right\| = \frac{1}{2}\left\{\kappa_j + \kappa_1 + \sqrt{(\kappa_j - \kappa_1)^2 + 4\xi_j{}^2}\right\}$$

yields a computable bound for $\|1 - Q^*_{j+1} Q_{j+1}\|$.

An alternative bound for $1 - Q^*_j Q_j$, called the scoreboard, is presented in Section 14. We now focus attention on $Q^*_j q_{j+1}$.

6. An Expression for $Q^*_j r_j$

Orthogonality among the computed q_i is lost because the fundamental relation

$$"AQ_j = Q_jT_j + r_je^*_j"$$

no longer holds. Instead

$$(1) \quad AQ_i = Q_iT_i - F_i + (r_i - s_i)e^*_i \ , \ i = 1, \ldots, j$$

for some

$$(2) \quad F_i \equiv (f_1, f_2, \ldots, f_{i-1}, 0).$$

The quantities F_i and s_i represent the cumulative effect of round-off error. It is natural to split off the error in the last column and put it with the already present 'truncation error' r_i. We choose to call it s_i rather than f_i for reasons (given in (8.5)) that have no importance at this stage. We postpcne a detailed examination of F_i as long as possible.

LEMMA 2. If (1) holds then

$$Q^*_j r_j = [(1 - Q^*_j Q_j)T_j - (1 - e_je^*_j)T_j(1 - Q^*_j Q_j)]e_j$$
$$- F^*_j q_j + (q^*_j Aq_j - \alpha_j)e_j + Q^*_j s_j.$$

Proof. Imitate the argument of Lemma 1 in Section 4.

$$\begin{aligned}
Q_j^*(r_j - s_j) &= Q_j^*(AQ_j - Q_jT_j + F_j)e_j, \text{ using (1)},\\
&= (Q_j^* AQ_j - Q_j^* Q_jT_j)e_j, \text{ since } F_je_j = 0,\\
&= [(AQ_j)^*Q_j - Q_j^* Q_jT_j]e_j, \text{ using } A^* = A,\\
&= [T_jQ_j^* Q_j - F_j^* Q_j + e_j(r_j - s_j)^*Q_j - Q_j^* Q_jT_j]e_j,\\
&\qquad \text{using (1) again,}\\
&= [-T_j(1 - Q_j^* Q_j) + (1 - Q_j^* Q_j)T_j - F_j^* Q_j]e_j\\
&\qquad + e_j(r_j - s_j)^*Q_je_j, \text{ adding and subtracting } T_j.
\end{aligned}$$

The fourth term on the right can be evaluated using the second line above,

$$\begin{aligned}
e_j^* Q_j^* (r_j - s_j) &= q_j^* Aq_j - e_j^* Q_j^* Q_jT_je_j\\
&= q_j^* Aq_j - \alpha_j + e_j^* (1 - Q_j^* Q_j)T_je_j.
\end{aligned}$$

After transposing, substituting, and rearranging terms the lemma's assertion is obtained. □

By contemplating this expression we can assess the contribution of various errors to the departure from orthogonality of the $\{q_i\}$. Recall that in exact arithmetic $q_{j+1} = r_j/\beta_j$. In practice, therefore

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

where g_j accounts for the errors in the division by β_j. In (8.10) a bound is given on $\|g_j\|$ which shows that it is always insignificant. So

$$(4) \quad Q_j^* q_{j+1} = Q_j^* r_j/\beta_j + Q_j^* g_j$$

and the first term on the right dominates the second.

Now we see that $\|Q_j^* q_{j+1}\|$, which is really $\|Q_j^* r_j\|/\beta_j$, need not be small like roundoff in 1; it may, for all we know, have inherited the amplified consequences of past rounding errors to an extent which makes $\|Q_j^* r_j\| \doteq \|Q_j\| \|r_j\|$ and $\|Q_j^* q_{j+1}\|$ comparable to 1.

Paige [6] points out that this defect must be seen in perspective. Abrupt loss of orthogonality happens only when β_j is tiny. Since $\beta_j \doteq \|r_j\|$ this loss signals that span (Q_j) is nearly invariant, which is just what we want.

Here is the dilemma: stop too soon and T_j's eigenvalues will be unnecessarily poor approximations to A's, stop too late and the columns of Q_j will be dependent and Theorem 2 will not give a useable bound on the accuracy of the eigenvalue approximations. To resolve the dilemma we must have realistic estimates for $\|Q_j^* q_{j+1}\|$.

7. Computable Bounds on $Q_j^* q_{j+1}$

In this section we obtain computable bounds on $\|Q_j^* r_j\|$ and $\|Q_j^* q_{j+1}\|$. The expansion for $Q_j^* r_j$ in Lemma 2 fell into two parts

$Q_j^* r_j = c_j + d_j$, where

$$(1) \quad \begin{aligned} c_j &\equiv [(1 - Q_j^* Q_j)T_j - (1 - e_j e_j^*)T_j(1 - Q_j^* Q_j)]e_j, \\ d_j &\equiv -F_j^* q_j + (q_j^* A q_j - \alpha_j)e_j + Q_j^* s_j. \end{aligned}$$

Note that c_j does not depend on the specific implementation of the algorithm. It turns out that c_j dominates d_j as j increases and so we spurn the crude bound

$$(2) \quad \|c_j\| \le 2\kappa_j \|T_j\|$$

which ignores the role of e_j in (1).

LEMMA 3. Let $\|Q_i^* q_{i+1}\| \le \xi_i$, $i < j$. Then

$$\|c_j\| \le \|(T_{j-1} - \alpha_j)\|\xi_{j-1} + \beta_{j-1}(\xi_{j-1} + \xi_{j-2} + 2\kappa_1) + |\alpha_j|\kappa_i.$$

Proof. Partition T_j and $1 - Q_j^* Q_j$ to find

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

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

The factor $(1 - e_j e_j^*)$ simply annihilates the bottom element. Moreover, by (5.11)

$$|1 - \|q_i\|^2| < \kappa_1$$

and

$$|q_j^* q_{j-1}| = |q_{j-1}^* q_j| \le \xi_{j-1}.$$

On collecting terms the asserted bound is obtained. □

A more convenient, but weaker, bound is

$$(3) \quad \|c_j\| \le \|T_{j-1}\|\xi_{j-1} + (|\alpha_j| + 2\beta_{j-1})(\xi_{j-1} + \kappa_1), \ \beta_0 = \xi_0 = 0,$$

$$< \sqrt{3}\|T_j\|_E(\xi_{j-1} + \kappa_1) \quad \text{by the Cauchy-Schwarz inequality,}$$

and, less crudely,

$$(4)\quad \begin{aligned} \|T_{j-1}\| &\leq \|T_{j-1}\|_\infty \text{ , using symmetry,} \\ &= \max \{\max_{i<j-1} (\beta_{i-1} + |\alpha_i| + \beta_i), \beta_{j-2} + |\alpha_{j-1}|\}. \end{aligned}$$

The quantity κ_1 will be our basic unit of roundoff. It is constrained by (5.12) and can be given the specific value

$$(4)\quad \kappa_1 = (n + 6)\varepsilon$$

where ε is the relative precision of the basic arithmetic operations. We must have

$$(5)\quad n\varepsilon + 2\sigma \leq \kappa_1$$

where σ is a bound on the relative error in the square root subroutine.

Consider now d_j, the second term in (1). In order to bound $\|d_j\|$ it is necessary to specify the Lanczos algorithm (at last!) and perform an error analysis to see how the roundoff vectors s_i and f_i arise in (6.1). Unfortunately the reward for this labor is a term which becomes unimportant as soon as orthogonality between the q_i evaporates. In order to avoid a digression at this point we quote the results from Lemma 5 (Section 8). For all i,

$$(6)\quad \left\{ \begin{aligned} \|f_i\| &< \kappa_1 \|A\|_E/(1 + \kappa_1) \\ \|s_i\| &< \kappa_1 \|A\|_E/(1 + \kappa_1) \\ |q_i^* Aq_i - \alpha_i| &< 2\kappa_1 \|A\|_E \text{ .} \end{aligned} \right.$$

These bounds are very crude and the factor $\|A\|_E$ is unpleasant. Frequently, but not always, $\|A\|_E \leq w\|A\|$ with $w << \sqrt{n}$. The factor $\|A\|_E$ comes from the only place in which A appears explicitly, namely in computing u_i which is intended to be Aq_i. The error in this operation is discussed in Section 8.

LEMMA 4. With the bounds given in (6),

$$(7) \quad \|d_j\| \le \{\sqrt{j-1} + 3 + \kappa_j\}\ \kappa_1\|A\|_E.$$

Proof. $\|F_j^* q_j\| \le \|F_j\|_E\|q_j\| \le \sqrt{j-1}\ \max_{i<j}\|f_i\|\sqrt{1+\kappa_1} \le \sqrt{j-1}\ \kappa_1\|A\|_E$,

$\|Q_j^* s_j\| \le \|Q_j\|\|s_j\| \le \sqrt{1+\kappa_j}\ \kappa_1\|A\|_E < (1+\kappa_j)\kappa_1\|A\|_E$,

$\|d_j\| \le \|F_j^* q_j\| + \|Q_j^* s_j\| + |q_j^* Aq_j - \alpha_j|$. □

Let ω_j be the sum of the bounds in Lemmas 3 and 4,

$$(8) \quad \omega_j \equiv \xi_{j-1}\|T_{j-1}\|_\infty + (|\alpha_j| + 2\beta_{j-1})(\xi_{j-1} + \kappa_1) + [\sqrt{j-1} + 3 + \kappa_j]\kappa_1\|A\|_E.$$

Then $\|Q_j^* q_{j+1}\| \le \xi_j$, where $\xi_0 = 0$ and, by (6.4),

$$(9) \quad \boxed{\xi_j \equiv \omega_j/\beta_j + \sqrt{1+\kappa_j}\ \varepsilon.}$$

For actual use it would be preferable to have a bound ω_j constructed entirely from computed quantities. All that is needed is a bound on the relative error in u_j. Such a bound comes easily if extended precision is used in evaluating Aq_j. Even in standard working precision it is possible to represent u_j as the difference of two nonnegative vectors, $u_j = x_j - y_j$, and use $m\varepsilon\|x_j + y_j\|/\|u_j\|$ as the bound. Machine language programming is needed to keep down the cost of this device. In any case, let

$$(10) \quad \begin{aligned} s_j' &\equiv u_j - Aq_j\ , \\ \|s_j'\| &< \rho\|u_j\| \quad . \end{aligned}$$

For $i < j$, $\|u_j\|$ can be bounded, using (8.3.iii), as follows

$$\|u_i\|^2 = \|\beta_{i-1}q_{i-1} + \alpha_i q_i + \beta_i q_{i+1}\|^2 + \|s_i'' + \beta_i q_i\|^2 ,$$

$$(11) \qquad \leq (\beta_{i-1}^2 + \alpha_i^2 + \beta_i^2)(1 + 2\kappa_1) + 2[\beta_i|\alpha_i|\xi_i + \beta_i\beta_{i-1}\xi_i + \alpha_i\beta_{i-1}\xi_{i-1}] ,$$

$$\leq \sigma_i^2(1 + 2\bar{\xi}_i + 2\kappa_1), \text{ using } 2\gamma\delta \leq \gamma^2 + \delta^2, \text{ where}$$

$$\sigma_i^2 \equiv \beta_{i-1}^2 + \alpha_i^2 + \beta_i^2, \qquad \sum_{i=1}^{j-1}\sigma_i^2 = \|T_j\|_E^2 - \alpha_j^2$$

$$\bar{\xi}_i = \max_{k \leq i} \xi_k .$$

When $i = j$, β_j is unknown and we use the crude bound $\|u_j\| < 3\sigma_j$. Without proof we present the new computable bound for $\|d_j\|$.

LEMMA 4'. $\|d_j\| \leq (\rho + 5\varepsilon)(1 + 2\xi_{j-1} + 2\kappa_1)\sqrt{1 + \kappa_1}\,\|T_j\|_E$

$$+ 6\sigma_j(\rho + \kappa_1)\sqrt{1 + \kappa_j}.$$

Let us recapitulate. Since the bound on $\|f_i\|$ is independent of i and is like roundoff in $\|A\|_E$ we may say that the Q_j given by the basic Lanczos algorithm does indeed satisfy $AQ_j = Q_jT_j + r_je_j^*$ to within working accuracy. However the second relation $1 = Q_j^* Q_j$ may breakdown completely. Nevertheless it is useful to continue building up Q_j while it retains full rank. There is no guarantee that span (Q_j) will be almost invariant when rank $(Q_j) < j$ but there is also no guarantee that further steps will yield any improvement in eigenvalue approximations. On the other hand convergence of span (Q_j) to an invariant subspace is inevitably accompanied by orthogonality loss between the q_i.

Please note that the details of the algorithm were needed only to give a specific value to the error bound ξ_j, not for understanding the general behavior of the process.

8. The Basic Lanczos Algorithm

The algorithm gradually builds up matrices Q_j and T_j which are intended to satisfy

$$(1) \qquad \begin{aligned} AQ_j &= Q_jT_j + r_je_j^* , \\ 1 - Q_j^*Q_j &= 0 . \end{aligned}$$

One of the attractions of the process is its simplicity. Initially $T_0 = \beta_0 = 0$. The j-th step begins with T_{j-1}, β_{j-1}, and Q_j in hand. In exact arithmetic the following quantities are then computed.

$$(2) \qquad \begin{aligned} &\text{(i)} & u_j &= Aq_j, \\ &\text{(ii)} & \alpha_j &= q_j^*u_j \text{ , (which ensures } q_j^*r_j = 0) \text{ ,} \\ &\text{(iii)} & r_j &= u_j - \alpha_jq_j - \beta_{j-1}q_{j-1}, \\ &\text{(iv)} & \beta_j &= \|r_j\|, \\ &\text{(v)} & &\text{if } \beta_j > 0 \text{ then } q_{j+1} = r_j/\beta_j, \text{ otherwise stop.} \end{aligned}$$

Note that Q_{j-2} is not needed and q_{j+1} will automatically be orthogonal to Q_{j-1}.

For completeness we mention that a comparable analysis could be made of the very similar algorithm obtained by replacing (i) by

$$\text{(i)}' \quad u_j = Aq_j - \beta_{j-1}q_{j-1},$$

and (iii) by

$$\text{(iii)}' \quad r_j = u_j - \alpha_jq_j .$$

The reader who is not interested in an error analysis of the algorithm may safely skip the rest of this section.

In finite precision arithmetic none of the steps will be executed exactly. The goal is to find a small $\beta_j \doteqdot \|r_j\|$ and when cancellation occurs in forming r_j the computed vector will have a high relative error unless extra precision is used in this step. In any case the computed quantities satisfy relations involving certain round-off terms:

$$
(3)\qquad
\begin{array}{rl}
\text{(i)} & u_j = Aq_j + s_j', \ (s_j' \text{ will be discussed below}),\\
\text{(ii)} & \alpha_j = q_j^* u_j - \delta_j \ ,\\
\text{(iii)} & r_j = u_j - \alpha_j q_j - \beta_{j-1} q_{j-1} + s_j'',\\
\text{(iv)} & \beta_j = \|r_j\|/(1+\eta_j),\\
\text{(v)} & q_{j+1} = r_j/\beta_j + g_j, \text{ provided that } \kappa_{j+1} \le 7/8,\\
 & \text{otherwise stop.}
\end{array}
$$

Before discussing bounds on the round-off terms we see how f_i arises from (3);

$$
(4)\qquad Aq_i = u_i - s_i', \quad = \begin{cases} \beta_{i-1}q_{i-1} + \alpha_i q_i + \beta_i q_{i+1} - f_i, & i < j,\\ \beta_{j-1}q_{j-1} + \alpha_j q_j + r_j - s_j & , i = j. \end{cases}
$$

where

$$
(5)\qquad f_i \equiv s_i + \beta_i g_i, \ s_i \equiv s_i' + s_i''.
$$

The term which dominates the errors is s_j' and its assessment poses a special problem . In applications of the Lanczos algorithm to large sparse matrices the user is expected to supply a procedure or subroutine which computes u_j for a given q_j. This is the only way in which A enters the process and the subroutine is presumed to be specially adapted to take advantage of A's structure. Without an assumption about the accuracy of the compuation of u_j there is little point in using the subroutine. Here are three possible assumptions.

$$
(6) \qquad
\begin{array}{ll}
(\alpha) & s_j' = 0, \text{ no error,} \\
(\beta) & \|s_j'\| < \varepsilon\|u_j\|, \\
(\gamma) & \|s_j'\| < m\varepsilon\|A\|_E\|q_j\| < \kappa_1\|A\|_E
\end{array}
$$

where m is comparable to the number of nonzero elements in any row of A. These assumptions correspond approximately to infinite, double, and single precision arithmetic respectively. To be definite we shall presume that standard working precision is used with <u>no</u> accumulation of inner products.

LEMMA 5. With $\kappa_1 = (n + 6)\varepsilon$, $n^2\varepsilon < 6$, and (6γ) governing the evaluation of Aq_j, then, for $i = 1, \ldots, j$

$$\|s_i\| < \|f_i\| < \kappa_1\|A\|_E/(1 + \kappa_1),$$

$$|q_i{*}Aq_i - \alpha_i| < 2\kappa_1\|A\|_E. \qquad .$$

Proof. We assume that the reader has some familiarity with Wilkinson's treatment of round-off error; [9, Chapter 3]. However no explicit backward analysis will be relevant here. A useful bound concerns the error in forming an inner product between two n-vectors

$$(7) \quad |fl(x{*}y) - x{*}y| < n\varepsilon\|x\|\|y\| < \kappa_1\|x\|\|y\|.$$

where fl denotes the result of the specified calculation executed in standard floating point arithmetic with relative error ε in the basic operations.

Consider the error terms in (3), (3i) being covered by hypothesis. Recall, from (5.8), that $\|q_i\|^2 < 1 + \kappa_1$

$$(8) \quad |\delta_j| < n\varepsilon\|q_j\|\|u_j\| < n\varepsilon\|q_j\|^2\|A\|_E < \kappa_1\|A\|_E.$$

To assess s_j'' it is necessary to note that $\|\alpha_j q_j + \beta_{j-1}q_{j-1}\| \leq \|u_j\|$. For each element

$$r_j^i = fl[u_j^i - fl(\alpha_j q_j^i + \beta_{j-1}q_{j-1}^i)]$$

$$= u_j^i(1+\varepsilon) - [\alpha_j q_j^i(1+\varepsilon) + \beta_{j-1}q_{j-1}^i(1+\varepsilon)](1+\varepsilon),$$

whence

$$(9) \quad \|s_j''\| < 4\varepsilon\|u_j\| < 4\varepsilon(1 + m\varepsilon)\|A\|_E\|q_j\|.$$

From (6.4) and (3iii)

$$(10) \quad \|g_j\| < \varepsilon\|r_j\|/\beta_j < \varepsilon(1+\eta_j) < \varepsilon\sqrt{1+\kappa_1}$$

and

$$(11) \quad (1-\varepsilon)(1+\eta_j) \leq \|q_{j+1}\| \leq (1+\eta_j)(1+\varepsilon).$$

whereas

$$\sqrt{1-\kappa_1} < \|q_{j+1}\| < \sqrt{1+\kappa_1}$$

by definition of κ_1. Hence $\kappa_1 = 2(\eta_j + \varepsilon)$ and η_j need not appear explicitly.

A bound for s_j comes from (5), (6), and (9),

$$(12) \quad \|s_j\| \leq \|s_j'\| + \|s_j''\| < m\varepsilon\|A\|_E\|q_j\| + 4\varepsilon(1+m\varepsilon)\|A\|_E\|q_j\|.$$

Now $f_j = s_j + \beta_j g_j$ and $\beta_j \leq \|A\|_E$, $\|g_j\| < \varepsilon\sqrt{1+\kappa_1}$. Hence both s_j and f_j are generously bounded by $\kappa_1\|A\|_E/(1+\kappa_1)$, where the divisor $1+\kappa_1$ is inserted for convenience in applications. Finally

$$(13) \quad |q_j{*}Aq_j - \alpha_j| = |-q_j{*}s_j{}' + \delta_j| < \sqrt{1 + \kappa_1 m\varepsilon}\|A\|_E$$

$$+ \kappa_1\|A\|_E < 2\kappa_1\|A\|_E. \qquad \square$$

9. Reorthogonalization

As we have seen in the previous sections it is possible that the basic algorithm will be halted before any error bound becomes negligible. One alternative is to start the Lanczos algorithm again with the best approximate vector that can be derived from the final T_j. Another remedy, suggested by Lanczos himself, is not to compute q_{j+1} by normalizing r_j but, first, to purge r_j of any remaining components in the earlier q_i obtaining

$$p_j = r_j - \sum_{i=1}^{j} q_i q_i^* r_j = (1 - Q_j Q_j^*) r_j$$

and then to normalize p_j.

The extra cost of this cure is substantial. Not only does the multiplication count per step go from 5n to (2j + 5)n (and at least half of the inner products should be accumulated in higher precision) but, of more consequence, all the q_i are needed at each step. When treating large order matrices it may not be possible to hold all these vectors in fast storage and a nasty data handling problem has to be faced.

In this section we consider whether the new algorithm will satisfy the tridiagonal relation $(AQ_j - Q_jT_j = r_je_j^*)$ and the orthogonal relation $(1 = Q_j^*Q_j)$ to working accuracy. It had been supposed that the latter would always hold and that the algorithm could be run on to $j = n$ quite safely. In [6] Paige pointed out that although this is frequently the case it cannot be guaranteed when standard arithmetic facilities are used. Again there must be a stopping criterion. The big difference from the basic algorithm is that with a suitable test *the algorithm will only stop when a subspace,* span (Q_j), *has been found which is invariant to working accuracy*.

Let us specify the modified algorithm. The j-th step delivers quantities satisfying the following relations

$$(1)\quad \begin{aligned}
&\text{(i)} && u_j = Aq_j + s_j{}',\\
&\text{(ii)} && \alpha_j = q_j^* u_j - \delta_j,\\
&\text{(iii)} && r_j = u_j - \alpha_j q_j - \beta_{j-1} q_{j-1} + s_j{}'',\\
&\text{(iv)} && p_j = r_j - Q_j Q_j^* r_j + t_j,\\
&\text{(v)} && \beta_j \doteq \|p_j\|, \text{ and if the termination test is not passed,}\\
&\text{(vi)} && q_{j+1} = p_j/\beta_j + g_j.
\end{aligned}$$

Putting all these relations together gives

$$Aq_i = \beta_{i-1} q_{i-1} + \alpha_i q_i + \beta_i q_{i+1} - s_i{}' - s_i{}'' + Q_i Q_i^* r_i - t_i - \beta_i g_i, \; i < j,$$

$$Aq_j = \beta_{j-1} q_{j-1} + \alpha_j q_j + r_j - s_j{}' - s_j{}''.$$

Thus, exactly as in the basic algorithm, with $s_i = s_i{}' + s_i{}''$,

$$(2)\quad AQ_j = Q_j T_j - F_j + (r_j - s_j) e_j^*,$$

but now, for $i < j$,

$$(3)\quad \begin{aligned} f_i &\equiv -Q_i Q_i^* r_i + h_i,\\ h_i &\equiv t_i + s_i + \beta_i g_i, \end{aligned}$$

and two of the terms in f_i are new. It looks as though the error term F_j may be bigger than before. Indeed it may.

Lemma 2 (Section 6) depends solely on (2) and so

$$(4)\quad Q_j^* r_j = -F_j^* q_j + k_j, \text{ where}$$
$$k_j \equiv [(1 - Q_j^* Q_j)T_j - (1 - e_j e_j^*)T_j(1 - Q_j^* Q_j) + (\delta_j - q_j^* s_j')]e_j + Q_j^* s_j.$$

Recall from (7.3) and (7.10) that in the basic algorithm k_j dominated $F_j^* q_j$. This is no longer always true and the analysis of $Q_j^* r_j$ is quite complicated. Fortunately $Q_j^* r_j$ is no longer the crucial part of $Q_j^* q_{j+1}$ in the new algorithm. From (1, vi) and (1, iv)

$$\begin{aligned} Q_j^* q_{j+1} &= Q_j^* p_j/\beta_j + Q_j^* q_j, \\ (5)\qquad &= Q_j^*[(1 - Q_j Q_j^*)r_j + t_j]/\beta_j + Q_j^* q_j. \\ &= \boxed{(1 - Q_j^* Q_j)}\; Q_j^* r_j/\beta_j + \boxed{Q_j^* t_j/\beta_j} + Q_j^* g_j. \end{aligned}$$

The new terms are in boxes.

The middle term in (5) shows why reorthogonalization does not unconditionally guarantee the orthogonality of the q_i to working precision. The vector t_j is the absolute error incurred in reorthogonalizing r_j. Bounds are given in the appendix for the various ways of computing p_j. They all have a component which is the roundoff in r_j; say $\|t_j\| \leq \tau_j \|r_j\|$. Now $\beta_j \doteq \|p_j\| \leq \|r_j\|$ and <u>there is no a priori upper bound on</u> $\|r_j\|/\beta_j$. So orthogonality will leak away whenever $\|p_j\|$ is appreciably less than $\|r_j\|$, an event which is much rarer than cancellation in the calculation of r_j.

10. <u>Termination Criteria</u>

The way out of this difficulty is to stop the algorithm appropriately. Criteria arise naturally from the illuminating formula (9.5). Before presenting a detailed discussion we point out a difficulty. Appropriate criteria turn out to depend quite strongly on the way the reorthogonalization step is carried out. There is, however, no canonical way of computing p_j from r_j. In the appendix we present three possible implementations of this step and their associated error bounds.

The simplest rule is to stop when convenient storage space is exhausted. The process may then be iterated as described in Section 3. There is not much that can be said about each pass separately and we will not pursue this aspect any further.

The ideas determining the specific stopping rules are quite simple. From (9.5) and the bounds on $\|t_i\|$ it is apparent that the critical terms in bounding $\|Q_k^* q_{i+1}\|$ are $\kappa_i \|Q_i^* r_i\|/\beta_i$ and $\|r_i\|/\beta_i$. Our criteria must keep them small. However there is a tradeoff. The tighter the bounds on these terms the weaker will be the bounds on the final residual

$$\|AQ_j - Q_j T_j\| \leq \|F_j\| + \|r_j - s_j\|.$$

In order to specify appropriate stopping rules it is helpful to know how the bound ξ_i on $\|Q_i^* r_i\|$ affects the growth of the bound κ_{i+1} on $\|1 - Q_{i+1}^* Q_{i+1}\|$.

LEMMA 6. If $\xi_i \leq \xi$ then $\kappa_{i+1} \leq \kappa_1 + \xi\sqrt{2i}$,

if $\xi_i \leq \xi\sqrt{i}$ then $\kappa_{i+1} \leq \kappa_1 + \xi i$,

if $\xi_i \leq \xi i$ then $\kappa_{i+1} \leq \kappa_1 + \xi i\sqrt{i+1}$.

Proof. Recall from (5.10) that

$$\kappa_{i+1} \equiv \left\| \begin{pmatrix} \kappa_i & \xi_i \\ \xi_i & \kappa_1 \end{pmatrix} \right\| = \frac{1}{2}\{\kappa_i + \kappa_1 + \sqrt{(\kappa_i - \kappa_1)^2 + 4\xi_i^2}\ \}.$$

It is more convenient to work with $\bar{\kappa}_i = \kappa_i - \kappa$. Our object is to majorize the solution to the nonlinear difference equation $2\bar{\kappa}_{i+1} = \bar{\kappa}_i + \sqrt{\bar{\kappa}_i^2 + 4\xi_i^2}$ with initial condition $\bar{\kappa}_1 = 0$.

Case 1: $\xi_i \leq \xi$. Then $\bar{\kappa}_i \leq \xi\sqrt{2(i-1)}$ yields

$$2\bar{\kappa}_{i+1} \leq \xi\{\sqrt{2(i-1)} + \sqrt{2(i-1)+4}\} \leq 2\xi\sqrt{2i}\ [1 - 1/32j^2] \leq 2\xi\sqrt{2i}.$$

The other two cases are similar. □

As an illustration of the way to choose a stopping rule we consider the most favorable, and most expensive, computation of p_i. The associated bound is given in (12.8).

From (9.5) and (6.4)

(1) $$\|Q_j^* q_{j+1}\| \leq \kappa_j \| Q_j^* r_j \|/\beta_j + \|Q_j\|(\|t_j\| + \beta_j\|q_j\|)/\beta_j,$$

$$\leq (\kappa_j + \varepsilon\sqrt{1+\kappa_j})\| Q_j^* r_j \|/\beta_j + \sqrt{1+\kappa_j}\ (\varepsilon + \varepsilon)\|r_j\|/\beta_j.$$

Now select a parameter δ, $0 < \delta < 1$, which is to serve as a tolerance on the relative diminution in $\beta_j (\doteq \|p_j\|)$ below $\|r_j\|$. Suitable values for δ will be discussed in the next section. Stop the Lanczos algorithm if

(2a) $\beta_j < \sqrt{1+\kappa_j}\ \|r_j\|/(1+\delta)$, or

(2b) $\beta_j < (\kappa_j + \varepsilon\sqrt{1+\kappa_j})\|Q_j^* r_j\|/\kappa_1(1+\delta)$

For all i before termination (1) yields

(3) $\|Q_i^* q_{i+1}\| \leq (1+\delta)(\kappa_1 + 2\varepsilon) \equiv \xi,$

and, by Lemma 6

(4) $\|1 - Q_{i+1}^* Q_{i+1}\| \leq \kappa_{i+1} \leq \kappa_1 + \xi\sqrt{2i}$.

The cost of these rules is the computation of $\|r_i\|$ and $\|c_i\|$. Note that $c_i \equiv Q_i^* r_i$ will be formed in the course of the reorthogonalization. Neither inner product ($r_i^* r_i$ and $c_i^* c_i$) need be accumulated in double precision. So this is a small extra expense compared to the j accumulated inner products of length n needed for c_i and the n accumulated inner products of length j+1 needed for $p_i = r_i - Q_i c_i$. In these circumstances it is only reasonable to compute $\beta_i \doteq \|p_i\|$ with accumulation in double precision. This brings down κ_1 from $n\varepsilon$ to 2ε. Thus

$$(5) \quad \xi = 4\varepsilon(1 + \delta) \ , \quad (\kappa_i + 2\varepsilon)/\kappa_1(1 + \delta) < 2(1 + \sqrt{2(i - 1)})$$

In words, orthogonality leaks away very slowly. There is no need to take seriously the κ_j appearing in (2a,2b).

It is worth noting that Paige realized the necessity for a stopping criterion and formulated one like (2b), namely $\beta_j < j\| Q_j^* r_j \|$. Our criterion permits the algorithm to go for more steps.

Since the computation of p_i from r_i dominates the cost of each step it is unreasonable not to accumulate <u>all</u> inner products to double precision if the smallest bound on $\|t_i\|$ is to be used. Using (7.11) we find

$$(6) \quad \|s_i\| \leq \|s_i'\| + \|s_i''\| \leq 2\varepsilon\|Aq_i\|,$$
$$\leq 2\varepsilon\sqrt{\beta_{i-1}^2 + \alpha_i^2 + \beta_i^2}\ \sqrt{1 + 2\xi + 2\varepsilon}, \text{ up to termination.}$$

This bound will be used in (11.2).

11. Bounds on the Final Residual

The algorithm stops at Step j with

$$AQ_j - Q_jT_j = -F_j + (r_j - s_j)e_j^*$$

and either (10.2a) or (10.2b) satisfied. Is $\|F_j\|$ like round-off in $\|A\|$? Is $\|r_j\|$ comparable to $\|F_j\|$? We hope that the answer is yes.

Everything depends on the reorthogonalization and we are analyzing the case when all inner products are accumulated in double precision. First we bound $\|r_j\|$ in terms of $\|Q_j^* r_j\|$.

LEMMA 7. If the algorithm halts because $\beta_j < \sqrt{1 + \kappa_j}\|r_j\|/(1 + \delta)$, (10.2a), and $\kappa_j < \delta$, then

$$\|r_j\| < (1 + \delta)\sqrt{1 + \kappa_j}\|Q_j^* r_j\|/[\delta - (\kappa_j/2) - 2\varepsilon(1 + \delta)] \doteq \delta^{-1}\|Q_j^* r_j\|.$$

If the algorithm halts because $\beta_j < (\kappa_j + \varepsilon\sqrt{1 + \kappa_j})\|Q_j^* r_j\|/(1 + \delta)\kappa_1$, (10.2b), then

$$\|r_j\| < [(\kappa_j + \sqrt{2\varepsilon})/\kappa_1(1 + \delta) + \sqrt{1 + \kappa_j}]\|Q_j^* r_j\|(1 + \varepsilon)^2,$$
$$< 3\sqrt{j}\|Q_j^* r_j\|.$$

Proof. By (9.1,iv) and (12.8)

$$\|r_j\| = \|p_j + Q_jQ_j^* r_j - t_j\|$$
$$\leq \beta_j(1 + \varepsilon) + \sqrt{1 + \kappa_j}\|Q_j^* r_j\|(1 + \varepsilon) + \varepsilon\|r_j\|.$$

Substitute the appropriate terminal bound, rearrange terms and the asserted inequalities are obtained. For the final inequality in the lemma use (10.5). □

Remark. The two bounds will be approximately equal when $\delta \doteq \kappa_1/\kappa_j \doteq \sqrt{2/j}$. This suggests that a variable tolerance $\delta_j \doteq \kappa_1/\kappa_j$ should be used in practice.

Lemma 7 does not show that $\|r_j\|$ is comparable to $\|F_j\|$. Recall from (9.3) that, for $i \leq j$,

$$(1) \quad f_i = -Q_i Q_i^* r_i + h_i,$$

$$h_i = t_i + \beta_i g_i + s_i \equiv \bar{t}_i + s_i \ .$$

Using (12.8) and (10.6)

$$(2) \quad \|f_i\| < (\sqrt{1+\kappa_i} + \varepsilon)\| Q_i^* r_i\| + (2+\delta)\varepsilon\beta_i$$

$$+ 2\varepsilon\sqrt{1+2\xi+\varepsilon}\ \sqrt{\beta_{i-1}^2 + \alpha_i^2 + \beta_i^2} \ , \text{ for } i < j.$$

In order to bound $\| Q_i^* r_i\|$ bounds are needed on $\| Q_k^* r_k\|$, $k < i$, and an inductive argument is called for.

LEMMA 8. Accumulation of all inner products in (9.1) in double precision yields

(i) $\| Q_i^* r_i\| < 25\varepsilon\|A\|_E$, $i \leq j$,

(ii) $\|F_j\|_E < 25\sqrt{j-1}\ \varepsilon(1+\delta)\|A\|_E$.

Proof. From (9.4) $Q_j^* r_j = -F_j^* q_j + k_j$.

In examining the basic Lanczos algorithm it was appropriate to bound $\| F_j^* q_j\|$ by $\|F_j\|\|q_j\|$ because each $\|f_i\|$ was bounded by a constant. A more careful analysis of the modified algorithm will show that the bound on $\| F_j^* q_j\|$ remains the same as in the basic process despite the fact that $\|F_j\|$ has increased by a factor $\sqrt{j}$.

Using (1) above we write $F_j = \sum_{i=1}^{j-1} f_i e_i^*$ and

$$(3) \quad F_j^* q_j = \sum_{i=1}^{j-1} e_i f_i^* q_j$$

$$+ \sum e_i r_i^* Q_i Q_i^* q_j + \sum e_i t_i^* q_j + \sum e_i s_i^* q_j .$$

Thus, using $\| Q_i^* q_j \| \leq \|Q_{j-1}^* q_j\| \leq \xi$,

$$(4) \quad \| F_j^* q_j \| \leq \xi\sqrt{\Sigma\| Q_i^* r_i\|^2} + (1 + \varepsilon)\{\sqrt{\Sigma\|\bar{t}_i\|^2} + \sqrt{\Sigma\|s_i\|^2}\}.$$

It will turn out that the last two terms in (4) dominate the first.

Observe that

$$(5) \quad \| Q_1^* r_1 \| = \quad q_1^* r_1 \leq (1 + \varepsilon)\|r_1\| \leq (1 + \varepsilon)\|A\|$$

so that (i) holds for $i = 1$. Now make the inductive hypothesis that (i) holds for all $i < j$. Then

$$(6) \quad \sqrt{\sum_{i=1}^{j-1} \| Q_i^* r_k\|^2} \leq 25\varepsilon\|A\|_E \sqrt{\Sigma(i - 1)} + \text{lower order terms}$$

$$\leq 13(j - 1)\varepsilon\|A\|_E + \dots ,$$

Also

$$\xi \leq 4\varepsilon(1 + \delta), \text{ by } (10.5).$$

Thus the first term on the right in (4) is $0(\varepsilon^2)$. Using (2)

$$\sum_{i=1}^{j-1} \|\bar{t}_i\|^2 \leq 2\varepsilon^2\{\Sigma\|Q_i{}^*r_k\|^2 + (2 + \delta)^2\ \Sigma\beta_i^2\},$$

$$(7) \quad \sqrt{\Sigma\|\bar{t}_i\|^2} \leq \sqrt{2}(2 + \delta)\varepsilon\|T_j\|_E + 0(\varepsilon^2),$$

$$\leq \sqrt{2}(2 + \delta)\varepsilon\|A\|_E.$$

Finally, using (8.6β) and (10.6) to allow for accumulation of inner products

$$(8) \quad \sqrt{\sum_{i=1}^{j-1} \|s_i\|^2} < 2\varepsilon\sqrt{1 + 2\xi + \varepsilon}\|T_j\|_E \leq 2\varepsilon\sqrt{1 + 2\xi + \varepsilon}\ \|A\|_E.$$

Putting (6), (7), (8) into (4) shows that

$$(9) \quad \| F_j^* q_j\| \leq 7\varepsilon\|A\|_E.$$

A bound on $\|k_j\|$ in (9.4) is needed. Recall that

$$(10)\qquad \begin{aligned} k_j &= c_j + (q_j^* A q_j - \alpha_j)e_j + Q_j^* s_j, \\ \|c_j\| &\le \sqrt{3}\xi\|T_j\|_E, \text{ from (7.3).} \end{aligned}$$

With accumulation of inner products

$$(11)\qquad \begin{aligned} |q_j^* A q_j - \alpha_j| &< \varepsilon(1+\varepsilon)\|Aq_j\| < \varepsilon(1+\varepsilon)^2\|A\|, \\ \|s_j\| &< 2\varepsilon\|Aq_j\| < 2\varepsilon(1+\varepsilon)\|A\|. \end{aligned}$$

Using (10.5) in the form $\xi < 8\varepsilon$ we find that

$$(12)\qquad \|k_j\| \le 18\varepsilon\|A\|_E,$$

$$\|Q_j^* r_j\| \le 25\varepsilon\|A\|_E.$$

By the principle of induction (i) is established.

Using this bound in (1) yields

$$\|f_i\| \le \sqrt{1+\kappa_i}\; 25\varepsilon\|A\|_E ,$$

$$\|F_j\|_E \le 25\sqrt{j-1}\;\varepsilon(1+\delta)\|A\|_E, \text{ provided } \kappa_i < \delta. \qquad □$$

Comparing Lemmas 7 and 8 we see that the bounds on $\|r_j\|$ and $\|f_{j-1}\|$ are approximately the same. Moreover $\|F_j\|_E < 25\sqrt{n}\;\varepsilon\;\|A\|_E$ which is certainly like round-off in $\|A\|_E$. Thus the fundamental relations

$$\text{"}AQ_j = A_jT_j\text{"} , \text{"}1 = Q_j^* Q_j\text{"}$$

are satisfied to working precision upon termination of the algorithm.

Similar analyses can be made for the two other ways of reorthogonalizing.

12. Error Bounds for Re-orthogonalization

Given is an $n \times j$ matrix $Q = (q_1, \ldots, q_j)$, whose columns are not necessarily orthonormal, and a vector r which is not arbitrary but is constructed so that $\|Q^*r\| << \|r\|$. Consider the computation of

$$(1) \quad p = r - \sum_{i=1}^{j} q_i \gamma_i = (1 - QQ^*)r,$$

where

$$(2) \quad c = Q^*r = (\bar{\gamma}_1, \ldots, \bar{\gamma}_j)^*.$$

The standard notations $fl(x^*y)$ and $fl_2(x^*y)$ will be used to denote the computation of x^*y in standard working precision and with accumulation of inner products, respectively. The standard bounds are

$$\begin{aligned} &|fl(x^*y) - x^*y| < n\,\varepsilon\,\|x\|\|y\|, \\ (3) \quad &|fl_2(x^*y) - x^*y| < \varepsilon\,|x^*y| + n\varepsilon^2\|x\|\|y\|. \\ &|fl(x+y) - x - y| < \varepsilon(\|x\| + \|y\|). \end{aligned}$$

The terms which are $O(\varepsilon^2)$ may be ignored by making a small relative increment in the value of ε.

There are three ways to implement (1):

$$(4) \quad \begin{aligned} \text{(i)} \quad & p^{(1)} = fl(r - Qc^{(1)}), \; c^{(1)} = fl(Q^*r), \\ \text{(ii)} \quad & p^{(2)} = fl(r - Qc^{(2)}), \; c^{(2)} = fl_2(Q^*r), \\ \text{(iii)} \quad & p^{(3)} = fl_2(r - Qc^{(2)}). \end{aligned}$$

The fact is that much of the value of reorthogonalization is discarded when (i) is used, but some current computers exact such a heavy penalty for accumulation of inner products that it is useful to consider computations which are confined to working precision. Working precision may already be long precision on some computers. Using (3) we find

$$(5)\quad\left\{\begin{aligned}
c^{(1)} &= fl(Q^*r) = c - y^{(1)}, \quad \|y^{(1)}\| < n\varepsilon\|Q\|_E\|r\|,\\
c^{(2)} &= fl_2(Q^*r) = c - y^{(2)}, \quad \|y^{(2)}\| < \varepsilon\|c\|,\\
p^{(1)} &= fl[r - fl(Qc^{(1)})] = fl[r - Qc + Qy^{(1)} + s^{(1)}],\\
&\qquad\qquad\qquad\qquad \|s^{(1)}\| < j\varepsilon\|Q\|_E\|c^{(1)}\|,\\
&= p - h + \boxed{Qy^{(1)}} + s^{(1)}, \quad \|h\| < 2\varepsilon\|r\| \quad \text{since} \quad \|Qc\| < \|r\|,\\
p^{(2)} &= p - \boxed{h'} + Qy^{(2)} + \boxed{s^{(1)\prime}}, \quad \|h'\| < 2\varepsilon\|r\|, \quad \|s^{(1)\prime}\| < j\varepsilon\|Q\|_E\|c^{(1)}\|,\\
p^{(3)} &= p - \boxed{h''} + Qy^{(2)} + s^{(2)}, \quad \|h''\| < \varepsilon\|r\|, \quad \|s^{(2)}\| < \varepsilon\|Qc^{(2)}\|.
\end{aligned}\right.$$

The dominant terms are in boxes.

We assume that rank $(Q) = j$ and, more precisely, that

$$(6)\quad 0 < 1 - \kappa \le Q^*Q \le 1 + \kappa.$$

Then

$$(7)\quad \|Q\| \le \sqrt{1+\kappa} < \sqrt{2}, \quad \|Q\|_E \le \sqrt{j}\ \sqrt{1+\kappa}.$$

Using (7) and (5) the final bounds are

$$(8)\quad\left\{\begin{aligned}
\|p^{(1)} - p\| &< [n\sqrt{j}(1+\kappa) + 2]\varepsilon\|r\| + j\sqrt{j}\ \sqrt{1+\kappa}\ \varepsilon\|Q^*r\| + O(\varepsilon^2),\\
\|p^{(2)} - p\| &< 2\varepsilon\|r\| + (j+1)\sqrt{j}\ \sqrt{1+\kappa}\ \varepsilon\|Q^*r\| + O(\varepsilon^2),\\
\|p^{(3)} - p\| &< \varepsilon\|r\| + \sqrt{1+\kappa}\ \varepsilon\|Q^*r\| + O(\varepsilon^2).
\end{aligned}\right.$$

It is quite legitimate to use $\|Q^*r\| < \sqrt{1+\kappa}\ \|r\|$ in order to give the bounds in terms of $\|r\|$ alone, but this is an unnecessary and crude waste of information.

The κ's in (8) could be omitted because their contribution is $O(\varepsilon^2)$.

13. The Block Lanczos Method

The basic algorithm can be generalized in a natural way [1,2] to produce a block tridiagonal matrix

$$(1)\quad \tilde{T}_j \equiv \begin{bmatrix} A_1 & B_1^* & & & \\ B_1 & A_2 & B_2^* & & \\ & B_2 & \cdot & \cdot & \\ & & \cdot & \cdot & B_{j-1}^* \\ & & & B_{j-1} & A_j \end{bmatrix} .$$

This method is particularly appropriate if several, say p, eigenvectors of A are wanted. The process begins not with a single vector q, but with p orthonormal vectors. We think of them as columns of an n x p starting matrix Q_1. In theory, the Lanczos method will then build up a big matrix

$$(2)\quad \tilde{Q}_j = (Q_1, Q_2, \ldots, Q_j)$$

satisfying

$$(3)\quad A\tilde{Q}_j = \tilde{Q}_j\tilde{T}_j + R_jE_j^*$$

where

$$E_j^* = (0, 0, \ldots, 0, I_{p_j}),\ p_j = \text{rank}\ (Q_j).$$

The j-th step involves the following computations

$$(4)\quad\begin{cases}\text{(i)} & A_j = Q_j^* A Q_j, \\ \text{(ii)} & R_j = -Q_{j-1}B_{j-1}^* + AQ_j - Q_jA_j,\ B_0 = 0, \\ \text{(iii)} & Q_{j+1}B_j = R_j, \text{ where } B_j^*B_j = R_j^*R_j,\ Q_{j+1}^*\ Q_{j+1} = 1\end{cases}$$

What is new here is that Q_{j+1} and B_j are not uniquely defined by (iii) and must be determined by the use of some appropriate convention. We have in mind some stable form of the Gram-Schmidt process which will produce an upper triangular B_j with positive diagonal when R_j has full rank. And when R_j's rank is not full there is still a unique echelon form for B_j, for example

$$B_j = \begin{bmatrix} o & + & x & x & x \\ o & o & o & + & x \\ o & o & o & o & + \end{bmatrix}.$$

Because of rounding errors the relations in (4) will not hold exactly and the columns of $\hat{Q}_j$ will not be orthonormal.

We do not know precisely when the algorithm should be halted but, by Theorem 2 (Section 3), there is no point in continuing after linear independence among the columns of $\tilde{Q}_j$ is lost. So the focus of our attention is on a computable bound for $1 - \tilde{Q}_j^*\ \tilde{Q}_j$.

In this connection the following facts are useful. Let $\tilde{M}$ be any block matrix with M_{ij} as its (i,j) submatrix.

LEMMA 9. Suppose that $\|M_{ij}\| \le w_{ij}$, all i, j's and $W = (w_{ij})$ then, in the sense of quadratic forms,

$$1 - W \le 1 - \tilde{M} \le 1 + W$$

Proof. Let $\tilde{x}^* = (x_1^*, \ldots, x_n^*)$, $y^* = (\|x_1\|, \ldots, \|x_n\|)$. Then it can be verified that $\|\tilde{x}\| = \|\tilde{y}\|$ and

$$\tilde{x}^*\tilde{M}\tilde{x} \le y^*Wy,$$

It follows that

$$-W \le (\pm \tilde{M}) \le + W. \qquad \square$$

COROLLARY 1. If $\|W\| < 1$ then

$$0 < \lambda_1[1 - W] \le \lambda_1[1 - \tilde{M}].$$

COROLLARY 2. If $\tilde{M}_j = 1 - \tilde{Q}_j^* \tilde{Q}_j$ and $1 - W_j$ is positive definite then

$$\sqrt{\lambda_1[1 - W_j]} \le \text{smallest singular value of } \tilde{Q}_j.$$

The proofs are omitted.

These results reduce the problem to one of majorizing $\|Q_\ell^* Q_m\|$ by some $w_{\ell m}$. Moreover there is no <u>need</u> to compute $\lambda_1[1 - W_j]$. A novel termination criterion is the following.

Attempt to compute the Choleski factorization of $1 - W_j$. If successful keep the factor L_j and continue the algorithm. If the computation breaks down, due to 0 or negative pivots, then stop.

Here

$$W_j = \begin{pmatrix} W_{j-1} & w_j \\ w_j^* & w_{jj} \end{pmatrix}, \quad L_j = \begin{pmatrix} L_{j-1} & 0 \\ \ell_j^* & \delta_j \end{pmatrix}.$$

The cost of this test is quite small. First solve $L_{j-1}\ell_j = -w_j$ for ℓ_j, then compute

$$\delta_j = \sqrt{1 - w_{jj} - \ell_j^*\ell_j} \quad .$$

If linear independence between the blocks Q_i is lost too rapidly the simple version of the block Lanczos process ceases to be useful. On the other hand reorthogonalization is a heavy insurance against such a misfortune, so heavy in fact that the Lanczos algorithm loses many of its attractions.

There is room for some technique in between these extremes and Jane Cullum has made a valuable addition to the lore of the Lanczos process in [1]. When the block method is being used iteratively the vectors in a new starting matrix Q, split into two groups, those that have "settled down" and have already been accepted as eigenvectors and, in the other group, the remainder. It is important, almost essential, to *reorthogonalize each* R_j, $j \geq 1$, *in* (4 ii) *to all accepted eigenvectors*.

This feature complicates the process enough that we have not incorporated it into our analysis, though Cullum's reorthogonalization could be treated as a deflation technique auxiliary to the block Lanczos process.

14. The Scoreboard

A computable bound on $Q_\ell^* Q_m$ is derived in this section.

As usual we now let the previous symbols denote quantities actually stored in the computer. Because of round-off error the following quantities are not zero,

$$\Omega_j \equiv Q_j^* A Q_j - A_j,$$

$$(1) \qquad S_j' \equiv R_j - (AQ_j - Q_j A_j - Q_{j-1} B_{j-1}^*),$$

$$G_j \equiv Q_{j+1} B_j - R_j.$$

$$S_j \equiv S_j' + G_j$$

Let us assume that bounds are available in the form

$$(2) \quad \|\Omega_j\| \leq \omega_j, \ \|S_j\| \leq \sigma_j \ , \ \|1 - Q_j^* Q_j\| \leq \kappa_{jj}.$$

In fact we can expect these bounds to be independent of j but this is not necessary. For example the corresponding bounds of Section 8 could be multiplied by rank (Q_j) and used here.

Suppose, further, that we have bounds

$$(3) \quad \| Q_i^* Q_k \| \leq \kappa_{ik} = \kappa_{ki}; \ i, k \leq j, \ i \neq k \ .$$

We are going to derive computable values for $\kappa_{i,j+1}$, $i \leq j$. In this way, we can gradually build up a 'scoreboard' $W_{j+1}, (w_{im} = \kappa_{im})$, which is (j+1) x (j+1) and satisfies

$$(4) \quad 1 - \tilde{Q}_{j+1} \tilde{Q}_{j+1} \le W_{j+1} \, .$$

In order to obtain computable bounds we have to compute something. In this case, at Step j, we need

$$(5) \quad \|A_j\| \le \alpha_j , \ \delta_j = \sqrt{\lambda_1(B_j B_j^*)} \ , \ \beta_j \ge \|B_j\| \equiv \sqrt{\lambda_{\max}(B_j B_j^*)} \ ,$$

and

$$(6) \quad \|Q_i^* \, S_j\| \le \sigma_{ij} .$$

We digress to make a few comments on the bounds in (5) and (6). Only modest accuracy (25%?) is needed. If A is positive definite then trace (A_j) will do for α_j, otherwise $\|A_j\|_E$. There are several ways of approximating $\lambda_{\min}$ and $\lambda_{\max}$ of $B_j B_j^*$ without forming $B_j B_j^*$ explicitly. Certainly $\|B_j\|_E$ can be used for β_j.

The tricky question concerns δ_j. Any scheme which assigns a rank to R_j adequately for practical purposes turns out to involve an estimate δ_j of B_j's smallest singular value. This is so because the construction of Q_{j+1} and B_j can be regarded as an attempt to reduce $\|Q_{j+1}B_j - R_j\|$ to negligibility without making δ_j unnecessarily small, doing so in steps each of which increases the ranks of Q_{j+1} and B_j by 1 while diminishing both $\|Q_{j+1}B_j - R_j\|$ and δ_j. The pivoting process is Gram-Schmidt orthogonalization can be regarded as a way of avoiding, in each step, a small reduction in $\|Q_{j+1}B_j - R_j\|$ at the cost of a serious reduction in δ_j. A consequence of that pivoting process is a constraint upon the elements of B_j which allows estimates like Karasalo's [4.5] to be used with confidence provided the rank of B_j is small (< 10, say). Since the intricacies of adequate estimates for δ_j and rank (R_j) could distract us far from the discussion of the scoreboard, we shall say no more about them here.

From (2) $\sigma_{ij} \leq \sqrt{1 + \kappa_{ii}}\,\sigma_j$ but that is rather crude. Note that

$$\sum_{i=1}^{j} \| Q_i^* S_j \|^2 = \|\tilde{Q}_j S_j\|^2 \leq (1 + \|W_j\|)\sigma_j^2 \leq 2\sigma_j^2$$

because the process stops before $\|W_j\| > 1$.

Ideally, then, we would have $\sum \sigma_{ij}^2 \leq 2\sigma_j^2$.

Now we present the formulas for the $(j + 1)$-th column of W. The bounds in Lemma 10 are not for insight but for numbers!

LEMMA 10. Using the notation developed above

$$\kappa_{j,j+1} \equiv (\kappa_{jj}\alpha_j + \omega_j + \kappa_{j,j-1}\beta_{j-1} + \sigma_{jj})/\delta_j,$$

$$\kappa_{j-1,j+1} \equiv [\beta_{j-2}\kappa_{j-2,j} + \alpha_{j-1}\kappa_{j-1,j} + \beta_{j-1}(\kappa_{jj} + \kappa_{j-1,j-1}) + \sigma_{j,j-1} + \alpha_j\kappa_{j-1,j} + \sigma_{j-1,j}]/\delta_j,$$

For $i > 1$,

$$\kappa_{j-i,j+1} \equiv [\beta_{j-i-1}\kappa_{j-i-1,j} + (\alpha_{j-i} + \alpha_j)\kappa_{j-i,j} + \beta_{j-i}\kappa_{j+1-i,j} + \beta_{j-1}\kappa_{j-i,j-1} + \sigma_{j,j-i} + \sigma_{j-i,j}]/\delta_j.$$

Proof. The derivation is in the spirit of Lemma 2. We shall invoke the generalized inverse B_j^+ but with no intention of computing it. Using (1)

$$Q_{j+1} = (AQ_j - Q_jA_j - Q_{j-1}B^*_{j-1} + S_j)B_j^+.$$

So, using (1,i),

$$Q_j^*Q_{j+1} = (A_j + \Omega_j - Q_j^*Q_jA_j - Q_j^*Q_{j-1}B^*_{j-1} + Q_j^*S_j)B_j^+.$$

Using $\|B_j^+\| \le \delta_j^{-1}$ and (2), (5), (6) the first formula appears.

Next

$$Q^*_{j-1}Q_{j+1} = [(AQ_{j-1})^*Q_j - Q^*_{j-1}Q_jA_j - Q^*_{j-1}Q_{j-1}B_{j-1} + Q^*_{j-1}S_j)B_j^+,$$

while, from (1),

$$AQ_{j-1} = Q_{j-2}B^*_{j-2} + Q_{j-1}A_{j-1} + Q_jB_{j-1} - S_{j-1}.$$

On substituting for AQ_{j-1} above and using (2), (5), (6) the second formula arises. The third is similar. □

This technique can also be used with the basic algorithm and in that case the cost of the bounds in (5) disappears. Recall that the goal behind this more elaborate bound is to keep the Lanczos algorithm going as long as this is warranted and hence to cut down on the number of passes required of the whole iterative Lanczos algorithm. Note also that the generalization of the bounds in Section 7 to block form requires $\|\tilde{T}_j\|_E$ as well as δ_j. It is not clear that the scoreboard will be preferable to the simpler bounds presented earlier. Some complicated tradeoffs are involved.

REFERENCES

0. G.G. Belford and E.H. Kaufman, Jr., An Application of Approximation Theory to an Error Estimate in Linear Algebra, Math. Comp. 28 (1974), 711-713.

1. Jane Cullum and 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 10598.

2. G.H. Golub, Some Uses of the Lanczos Algorithm in Numerical Linear Algebra, in Topics in Numerical Analysis, edited by J.J.H. Miller Academic Press, 1974.

3. W. Kahan, Inclusion Theorems for Clusters of Eigenvalues of Hermitian Matrices, Computer Science Report, University of Toronto, (Feb.1967).

4. S. Kaniel, Estimates for Some Computational Techniques in Linear Algebra, Math. Comp. 20 (1966), 369-378.

4.5. I. Karasalo, A Criterion for the Truncation of the QR Decomposition Algorithm for the Singular Linear Least Squares Problem, BIT 14 (1974), 156-166.

5. C. Lanczos, An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators, J. Res. Nat. Bur. Standards 45 (1950), 255-282.

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

7. __________, The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices, (Ph.D. thesis), London University, 1971.

8. __________, Computational Variants of the Lanczos Method for the Eigenproblem, Jour. IMA 10 (1972), 373-381.

8.5. _________, Eigenvalues of Perturbed Hermitian Matrices, J. Lin. Alg. and Appl. 8 (1974), 1-10.

9. J.H. Wilkinson, The Algebraic Eigenvalue Problem, O.U.P. (1965).