MATHEMATICS OF COMPUTATION
VOLUME 44, NUMBER 169
JANUARY 1985, PAGES 105–124

A Look-Ahead Lanczos Algorithm for Unsymmetric Matrices

By Beresford N. Parlett, Derek R. Taylor and Zhishun A. Liu*

Abstract. The two-sided Lanczos algorithm sometimes suffers from serious breakdowns. These occur when the associated moment matrix does not permit triangular factorization. We modify the algorithm slightly so that it corresponds to using a 2×2 pivot in triangular factorization whenever a 1×1 pivot would be dangerous. The likelihood of breakdown is greatly reduced. The price paid is that the tridiagonal matrix produced by the algorithm now has bumps whenever a 2×2 pivot is used. Experiments with several versions of the algorithm on a variety of matrices are described, including some large problems arising in the study of plasma instability.

1. The Lanczos Algorithm and Its Breakdown. The most popular way to obtain all the eigenvalues of a nonsymmetric $n \times n$ matrix B is to use the QR algorithm which is readily available in most computing centers. As the order n increases above 100 the QR algorithm becomes less and less attractive, especially if only a few of the eigenvalues are wanted. This is where the Lanczos algorithm comes into the picture. It does not alter B at all but constructs a tridiagonal matrix J gradually by adding a row and column at each step. After several steps some of the eigenvalues θ_i of J will be close to some eigenvalues λ_k of B and by the nth step, if nothing goes wrong, $\theta_i = \lambda_i$, $i = 1,\ldots,n$. This description is correct in the context of exact arithmetic. Unfortunately things can go wrong, even in the absence of rounding errors. The relations between these troubles and orthogonal polynomials are developed in [2].

In order to discuss these troubles we must say more about the algorithm. Let J_k be the $k \times k$ tridiagonal produced at step k of the algorithm. There are infinitely many tridiagonal matrices similar to B and J_n is one of them. Thus for some matrix $Q_n := (q_1,\ldots,q_n)$ we have

$$Q_n^{-1} B Q_n = J_n. \tag{1}$$

It simplifies the exposition considerably to introduce a redundant symbol and write P_n^* instead of Q_n^{-1}. The superscript * indicates conjugate transpose.

Let $P_n := (p_1,\ldots,p_n)$ and replace (1) by two separate relations

$$P_n^* Q_n = I, \tag{2}$$

Received July 6, 1981; revised March 6, 1984.
1980 *Mathematics Subject Classification*. Primary 65F15.
*The first and third authors gratefully acknowledge support by the Office of Naval Research under Contract N00014-76-C-0013. The second author thanks the Remote Sensing program at the University of California, Berkeley, for use of its computer facility.

©1985 American Mathematical Society
0025-5718/85 $1.00 + $.25 per page

$$P_n^* B_n Q_n = J_n. \tag{3}$$

We mention in passing that when $B^* = B = A$, then we can arrange that $P_n = Q_n$. The difficulty we are going to describe cannot occur when $P_n = Q_n$.

By equating elements on each side of $BQ_n = Q_n J_n$ and $P_n^* B = J_n P_n^*$ in the natural increasing order, we shall see that B, p_1 and q_1 essentially determine all the other elements of P_n, Q_n and J_n. On writing

$$J_n := \begin{bmatrix} \alpha_1 & \gamma_2 & & & \\ \beta_2 & \alpha_2 & \gamma_3 & & \\ & \beta_3 & \ddots & \ddots & \\ & & \ddots & & \gamma_n \\ & & & \beta_n & \alpha_n \end{bmatrix}$$

we find

$$p_1^* B q_1 = \alpha_1,$$

and

$$Bq_1 = q_1\alpha_1 + q_2\beta_2, \qquad p_1^* B = \alpha_1 p_1^* + \gamma_2 p_2^*.$$

Hence q_2 and p_2^* are, respectively, multiples of "residual" vectors

$$r_2 := Bq_1 - q_1\alpha_1, \qquad s_2^* := p_1^* B - \alpha_1 p_1^*.$$

Furthermore, since $p_2^* q_2 = 1$ by (2), we have

$$s_2^* r_2 = \gamma_2 p_2^* q_2 \beta_2 = \gamma_2\beta_2 := \omega_2.$$

If $\omega_2 \neq 0$ and β_2 is given any nonzero value, then γ_2, q_2 and p_2^* are all determined uniquely. One good choice is $\beta_2 = \sqrt{|\omega_2|}$.

The general pattern emerges at the next step, on equating the second columns on each side of $BQ_n = Q_n J_n$ and $P_n^* B = J_n P_n^*$,

$$p_2^* B q_2 = \alpha_2,$$

$$Bq_2 = q_1\gamma_2 + q_2\alpha_2 + q_3\beta_3, \qquad p_2^* B = \beta_2 p_1^* + \alpha_2 p_2^* + \gamma_3 p_3^*.$$

At this point we can compute the "residual" vectors

$$r_3 := Bq_2 - q_1\gamma_2 - q_2\alpha_2, \qquad s_3^* := p_2^* B - \beta_2 p_1^* - \alpha_2 p_2^*$$

and

$$\omega_3 := s_3^* r_3 = \gamma_3\beta_3.$$

If $\omega_3 \neq 0$ and β_3 is given any nonzero value then γ_3, q_3 and p_3^* are all determined uniquely. And so it goes on until some ω_j vanishes.

Example 1 (No breakdown).

$$B = \operatorname{diag}(2,3,4), \quad q_1^* = \tfrac{1}{2}[1,1,1], \quad p_1^* = \tfrac{1}{2}[1,2,1].$$

Step 1:

$$\alpha_1 = 3, \quad \omega_2 = \tfrac{1}{2}, \quad \beta_2 = 1, \quad \gamma_2 = \tfrac{1}{2}.$$

$$q_2^* = \tfrac{1}{2}[-1,0,1], \qquad p_2^* = [-1,0,1].$$

Step 2:

$$\alpha_2 = 3, \quad \omega_3 = \tfrac{1}{2}, \quad \beta_3 = 1, \quad \gamma_3 = \tfrac{1}{2}.$$

$$q_3^* = \tfrac{1}{2}[1,-1,1], \qquad p_3^* = \tfrac{1}{2}[1,-2,1].$$

Step 3:

$$\alpha_3 = 3, \qquad \omega_4 = 0.$$

$$J_3 = \begin{bmatrix} 3 & \frac{1}{2} & 0 \\ 1 & 3 & \frac{1}{2} \\ 0 & 1 & 3 \end{bmatrix}.$$

This is the Lanczos algorithm. It must terminate at the nth step with $\omega_{n+1} = 0$ but it may stop sooner.

Premature termination at say step j ($< n$) can occur in two ways:

(I) either $r_{j+1} = 0$ or $s_{j+1}^* = 0^*$ or both,

(II) $r_{j+1} \neq 0$, $s_{j+1}^* \neq 0$, but $\omega_{j+1} = 0$.

In the 1950's when the Lanczos algorithm was regarded as a way to compute J_n Case I was regarded as a mild nuisance. If $r_{j+1} = 0$, then *any* nonzero vector orthogonal to $p_1,\ldots,p_j$ can be chosen as q_{j+1}. Similarly $s_{j+1} = 0$ gives ample choice for p_{j+1}.

Today, regarding Lanczos as a way to find a few eigenvalues of large B it seems better to stop at Case I in the knowledge that every eigenvalue of J_j is an eigenvalue of B. If more eigenvalues are wanted then it is best to start the Lanczos algorithm afresh with new, carefully chosen starting vectors q_1 and p_1.

The real trouble, which cannot occur when $B = B^* = A$, is Case II. Wilkinson calls this a serious breakdown. There seems to be no choice but to start again but no one has been able to suggest a practical way to choose the new q_1 and p_1 so as to avoid another wasted run of the algorithm. That is why the Lanczos method has not been used much when $B^* \neq B$. In this article we propose a modification of the algorithm which greatly reduces the occurrence of Case II. The price paid for this convenience is that J is not quite tridiagonal. There is a small bump (or bulge) in the tridiagonal form to mark each occurrence (or near occurrence) of Case II.

2. The Two-Sided Gram-Schmidt Process. The serious breakdown described above is not limited to the Lanczos algorithm. It can occur in any attempt to use the familiar Gram-Schmidt process to produce a biorthogonal (or biorthonormal) pair of sequences. Our modification of Lanczos seems more natural in such a context.

Let $F = (f_1,\ldots,f_n)$ and $G = (g_1,\ldots,g_n)$ be given real nonsingular $n \times n$ matrices. In other words $\{f_1,\ldots,f_n\}$ and $\{g_1,\ldots,g_n\}$ are each a basis for the vector space $\mathbf{R}^n$ of column n-vectors. We want to produce a new pair of bases $\{q_1,\ldots,q_n\}$ and $\{p_1,\ldots,p_n\}$ such that

$$P_n^* Q_n = \Psi_n = \operatorname{diag}(\psi_1,\ldots,\psi_n)$$

and, for each $j = 1,\ldots,n$,

$$\operatorname{span}\{q_1,\ldots,q_j\} = \operatorname{span}\{f_1,\ldots,f_j\}, \qquad \operatorname{span}\{p_1,\ldots,p_j\} = \operatorname{span}\{g_1,\ldots,g_j\}.$$

The Gram-Schmidt process dictates that at step j

$$q_j = f_j - \sum_{i=1}^{j-1} q_i(p_i^* f_j/\psi_i), \quad p_j = g_j - \sum_{i=1}^{j-1} p_i(q_i^* g_j/\psi_i), \quad \psi_j = p_j^* q_j.$$

All goes well until $\psi_j = 0$ for some j. This can happen despite the fact that F and G are nonsingular.

Example 2.

$$F = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}, \qquad G = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

Step 1: $q_1 = f_1, p_1 = g_1, \psi_1 = 1$.

Step 2: $q_2^* = [0,1,0], p_2^* = [-1,0,1], \psi_2 = 0$.

Our remedy for the case $\psi_j = 0$ is quite natural. If $p_j \neq 0$ then recompute q_j using f_{j+1} in place of f_j. If this fails too, then try f_{j+2}, and so on. If F is nonsingular, there must be some $i > 0$ such that f_{j+i} will yield a nonzero value for ψ_j.

Here is a formal proof of the last remarks. Let $q_j^{(k)}$ denote the vector obtained by using f_k instead of f_j, i.e.,

$$q_j^{(k)} = f_k - \sum_{i=1}^{j-1} q_i (p_i^* f_k / \psi_i).$$

LEMMA. *If* $p_j \neq 0$ *and* $p_j^* q_j^{(k)} = 0$ *for* $k = j, j+1, \ldots, n$, *then* F *is singular.*

Proof. By construction $p_j^* q_i = 0$ for $i < j$.
Hence $p_j \perp \operatorname{span}(f_1, \ldots, f_{j-1})$.
Thus $0 = p_j^* q_j^{(k)} = p_j^* f_k - 0$, for all $k \geqslant j$.
Thus $p_j \perp \operatorname{span}(f_1, \ldots, f_n)$.
Thus $p_j^* F = 0$ and F is singular. □

Now consider again the use of f_{j+1} instead of f_j. If the modification succeeds the first time then only one property of Gram-Schmidt has been sacrificed, namely

$$\operatorname{span}(q_1, \ldots, q_j) \neq \operatorname{span}(f_1, \ldots, f_j).$$

Moreover, if no further breakdown occurs then

$$\operatorname{span}(q_1, \ldots, q_i) = \operatorname{span}(f_1, \ldots, f_i) \quad \text{for } i > j.$$

In many applications this price is well worth paying. Before describing our remedy we must relate Gram-Schmidt to Lanczos.

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

$$K = K_n := \left[q_1, Bq_1, B^2 q_1, \ldots, B^{n-1} q_1 \right], \quad \text{and} \quad \tilde{K}^* = \tilde{K}_n^* := \begin{bmatrix} p_1^* \\ p_1^* B \\ p_1^* B^2 \\ \vdots \\ p_1^* B^{n-1} \end{bmatrix}.$$

We will not establish this result but content ourselves with stating the key observation, namely

$$\operatorname{span}(q_1, q_2, \ldots, q_j, Bq_j) = \operatorname{span}(q_1, \ldots, q_j, B^j q_1).$$

The K matrices are called Krylov matrices and the pleasant fact is that most of the work required for general Gram-Schmidt disappears in this case because

$$p_i^* B q_j = 0 \quad \text{for } i < j - 1.$$

Thus, the general formula

$$\beta_{j+1}q_{j+1} = B^j q_1 - \sum_{i=1}^{j} q_i\left(p_i^* B^j q_1/\psi_i\right)$$

collapses to the famous three-term recurrence

$$\beta_{j+1}q_{j+1} = Bq_j - q_j\alpha_j - q_{j-1}\gamma_{j-1}.$$

3. Triangular Factorization of Moment Matrices. Consider again the matrices K and $\tilde{K}^*$ defined in the previous section. Note that the (i, j) element of $\tilde{K}^*K$ is $(p_1^* B^{i-1})(B^{j-1}q_1)$, so

$$\tilde{K}^*K = M = M(p_1, q_1); \quad \text{where } m_{i+1,j+1} = p_1^* B^{i+j} q_1.$$

In order to use this observation we need some basic facts about the Lanczos algorithm (see [3], [5] or [12]). If it does not break down it produces matrices P and Q such that

$$q_{j+1} = \chi_j(B)q_1 \Big/ \left(\prod_{i=2}^{j+1}\beta_i\right), \qquad p_{j+1} = \chi_j(B^*)p_1 \Big/ \left(\prod_{i=2}^{j+1}\gamma_i\right),$$

where

$$\chi_{j+1}(t) = (t - \alpha_j)\chi_j(t) - \omega_j\chi_{j-1}(t),$$

and χ_j is the characteristic polynomial of the tridiagonal matrix J_j. In other words, for each j, q_j is a linear combination of the first j columns of K while p_j is the *same* linear combination of the columns of $\tilde{K}$, up to scaling. This can be expressed compactly in matrix notation as

$$Q = KL^{-*}\Pi^{-1}, \qquad P = \tilde{K}L^{-*}\tilde{\Pi}^{-1}. \tag{4}$$

Here

$$\Pi = \operatorname{diag}(1, \beta_2, \beta_2\beta_3, \ldots), \qquad \tilde{\Pi} = \operatorname{diag}(1, \gamma_2, \gamma_2\gamma_3, \ldots),$$

and L is the unit lower triangular matrix such that $L^{-*} := (L^{-1})^*$ has the coefficients of χ_j above the diagonal in the jth column.

Using (4) we can rewrite $I = P^*Q$ as

$$I = P^*Q = (\tilde{\Pi}^{-1}L^{-1}\tilde{K}^*)(KL^{-*}\Pi^{-1}),$$

i.e.,

$$M = L\Omega L^*, \tag{5}$$

where

$$\Omega = \tilde{\Pi}\Pi = \operatorname{diag}(1, \omega_2, \omega_2\omega_3, \ldots, \omega_2 \cdots \omega_n).$$

This result is not new (see Householder [3]) but it is worth emphasizing that the product $\omega_2 \cdots \omega_i$ is the ith pivot used in performing Gaussian elimination on the moment matrix M associated with B, q_1 and p_1.

When $B \neq B^*$ the moment matrix M need not be positive definite and so triangular factorization need not be stable, even when M is nonsingular. Even when $B^* = B = A$ it is the condition $q_1 = p_1$ which forces M to be positive definite.

The best known remedy when $\|L\| \gg \|M\|$ is to use some form of row or column interchange whenever an ω_i is too small. The standard "partial pivoting" strategy is

not available because a whole column of M is not available in the middle of the Lanczos process. An alternative remedy is to enlarge the notion of a "pivot" to include 2×2, or even larger submatrices. This idea is discussed in Parlett and Bunch, 1971, [8]. It is the basis of our method. Whenever a 2×2 pivot is used the tridiagonal J bulges outwards temporarily.

In the context of the Lanczos process, our remedy for a tiny ω_j requires us to compute q_{j+1} at the same time as q_j, and p_{j+1} at the same time as p_j. The formulas for these "Lanczos" vectors are somewhat different from the standard ones. We call our modification the "look-ahead Lanczos algorithm" because it computes at the current step some quantities which are not usually needed until the next step in the standard Lanczos process. However, no matrix-vector products Bq or p^*B are ever wasted.

4. The Next Pivot. The decomposition $M = L\Omega L^*$ is never found explicitly. In order to make use of the idea of using a 2×2 pivot it is necessary to determine the top left 2×2 submatrix of what would be the reduced matrix in the triangular factorization of M. These three numbers can be determined from the information available in the Lanczos process.

After $i - 1$ steps of the standard algorithm we have

$$r_i := Bq_{i-1} - q_{i-1}\alpha_{i-1} - q_{i-2}\gamma_{i-1} = \chi_{i-1}(B)q_1/(\beta_2 \cdots \beta_{i-1}),$$
$$s_i^* := p_{i-1}^*B - \alpha_{i-1}p_{i-1}^* - \beta_{i-1}p_{i-2}^* = p_1^*\chi_{i-1}(B)/(\gamma_2 \cdots \gamma_{i-1}),$$
$$\omega_i := s_i^*r_i.$$

Instead of normalizing r_i and s_i^* to get q_i and p_i^* we can look ahead *not* to the next Lanczos vectors q_{i+1}, p_{i+1}^* but to any vectors r_{i+1}, s_{i+1}^* such that

$$\text{the plane } (r_i, r_{i+1}) = \text{the plane } (q_i, q_{i+1}),$$
$$\text{the plane } (s_i^*, s_{i+1}^*) = \text{the plane } (p_i^*, p_{i+1}^*).$$

The simplest choice for r_{i+1} and s_{i+1} is

$$r_{i+1} := Br_i - q_{i-1}\omega_i, \qquad s_{i+1}^* := s_i^*B - \omega_i p_{i-1}^*.$$

The coefficient ω_i ensures that r_{i+1} is orthogonal to all the known p's, namely $p_1, \ldots, p_{i-1}$, and also that s_{i+1} is orthogonal to $q_1, \ldots, q_{i-1}$.

Note that if we choose as q_i any vector in the plane (r_i, r_{i+1}) other than a multiple of r_i then q_i will be of the form

$$q_i = \psi(B)q_1/(\beta_2 \cdots \beta_i)$$

with ψ a monic polynomial of degree i instead of the usual χ_{i-1} of degree $i - 1$. Moreover, it will be possible to choose q_{i+1} to be of the same form, using a different ψ but of the same degree i. This is a mild generalization of the basic Lanczos algorithm. The degrees of the new Lanczos polynomials are still nondecreasing but they do not always go up by one at each step. Before making a choice for $q_i, q_{i+1}, p_i, p_{i+1}$ we compute

$$\|r_i\|, \|s_i\|, \text{ and } \cos\angle(r_i, s_i) = \omega_i/\|r_i\| \cdot \|s_i^*\|.$$

Also,

$$\omega_{i+1} = s_{i+1}^*r_{i+1}, \qquad \theta_i = s_i^*r_{i+1} = s_{i+1}^*r_i,$$
$$\|r_{i+1}\|, \|s_{i+1}^*\|, \cos\angle(r_{i+1}, s_{i+1}) = |\omega_{i+1}|/\|r_{i+1}\| \cdot \|s_{i+1}^*\|.$$

Our choice of q_i, etc., must be based on the matrix

$$W_i = \begin{bmatrix} \omega_i & \theta_i \\ \theta_i & \omega_{i+1} \end{bmatrix}.$$

It turns out that the top left 2×2 submatrix of the reduced matrix that occurs in the associated triangular factorization of M is $(\omega_2 \cdots \omega_{i-1})W_i$. The reduced matrix is defined in [7, p. 198].

If both $\omega_i = 0$ and W_i is singular then more drastic measures are needed to salvage the algorithm. We will pursue this case in Section 7. When $\omega_i = 0$ then the standard Lanczos algorithm breaks down. Yet, if W_i is invertible, it is easy to choose suitable q's and p's so that the algorithm can proceed.

The equations above can be condensed into block form.

$$(r_i, r_{i+1}) = B(q_{i-1}, r_i) - (q_{i-1}, r_i)\begin{bmatrix} \alpha_{i-1} & \omega_i \\ 0 & 0 \end{bmatrix} - q_{i-2}[\gamma_{i-1}, 0],$$

$$(s_i, s_{i+1})^* = (p_{i-1}, s_i)^*B - \begin{bmatrix} \alpha_{i-1} & 0 \\ \omega_i & 0 \end{bmatrix}(p_{i-1}, s_i)^* - \begin{bmatrix} \beta_{i-1} \\ 0 \end{bmatrix} p_{i-2}^*,$$

$$W_i = (s_i, s_{i+1})^*(r_i, r_{i+1}) =: \begin{bmatrix} \omega_i & \theta_i \\ \theta_i & \omega_{i+1} \end{bmatrix}.$$

Various factorizations of W_i yield various selections for new q's and p's. These are discussed below. We write $\hat{\omega}$ for ω_{i+1} and θ for θ_i. Let us drop the subscript i and write

$$S^*R = W = VU.$$

Rewrite the above equation as

$$I = V^{-1}S^*RU^{-1} = (SV^{-*})^*(RU^{-1})$$

and recall that $P^*Q = I$. We thus have

$$P = SV^{-*}, \qquad Q = RU^{-1}.$$

1. *LU factorization*

$$V = \begin{bmatrix} 1 & 0 \\ \theta/\omega & 1 \end{bmatrix}, \qquad U = \begin{bmatrix} \omega & \theta \\ 0 & \hat{\omega} - \theta^2/\omega \end{bmatrix}.$$

2. *UL factorization*

$$V = \begin{bmatrix} \omega - \theta^2/\hat{\omega} & \theta \\ 0 & \hat{\omega} \end{bmatrix}, \qquad U = \begin{bmatrix} 1 & 0 \\ \theta/\hat{\omega} & 1 \end{bmatrix}.$$

3. *QR factorization*

$$V = \begin{bmatrix} \omega & -\theta \\ \theta & \hat{\omega} \end{bmatrix}\tau^{-1}, \qquad U = \begin{bmatrix} \tau^2 & \theta(\omega + \hat{\omega}) \\ 0 & (\omega\hat{\omega} - \theta^2) \end{bmatrix}\tau^{-1}, \qquad \tau = \sqrt{\omega^2 + \theta^2}.$$

4. *LU with interchange*

$$V = \begin{bmatrix} \omega/\theta & 1 \\ 1 & 0 \end{bmatrix}, \qquad U = \begin{bmatrix} \theta & \hat{\omega} \\ 0 & \theta - \omega\hat{\omega}/\theta \end{bmatrix}.$$

5. *The smallest angle.* In order to determine the smallest angle between the planes (r_i, r_{i+1}) and (s_i, s_{i+1}) it is best to orthonormalize the bases. Let the result be

$$(\bar{r}_i, \bar{f}_{i+1}) \quad \text{and} \quad (\bar{s}_i, \bar{g}_{i+1}),$$

when the Gram-Schmidt process is used. The (matrix) angle between the two planes, call it Φ, is defined, see [1], by

$$\cos \Phi = (\bar{s}_i, \bar{g}_{i+1})^*(\bar{r}_i, \bar{f}_{i+1}).$$

Let the Singular Value Decomposition of $\cos \Phi$ be

$$\cos \Phi = U \Sigma V^*$$

where $\Sigma = \operatorname{diag}(\sigma_1, \sigma_2)$ and $\sigma_1 \geqslant \sigma_2$. The appropriate definition of new Lanczos vectors to ensure the smallest $\angle(q_i, p_i)$ is

$$(q_i, q_{i+1}) = (\bar{r}_i, \bar{f}_{i+1}) V \Sigma^{-1/2}, \qquad (p_i, p_{i+1}) = (\bar{s}_i, \bar{g}_{i+1}) U \Sigma^{-1/2}.$$

Comments on the Factorizations. No. 1 corresponds to the standard Lanczos algorithm. No. 2 corresponds to simply swapping s_i^* with $s_i^* B$ and r_i with Br_i in the Lanczos algorithm. One consequence is that the J matrix will bulge out of tridiagonal form on both sides of the diagonal. No. 3 is always stable and keeps the bulge on one side. The same advantage accrues from No. 4 and, as might be expected, is somewhat simpler than No. 3.

We want to use No. 1 whenever this is a reasonable strategy but when ω is zero or very small it is vital to choose one of the other procedures. We have tentatively chosen No. 4 on the grounds of simplicity. The interesting question which we now address is when to use No. 4.

Initial success with No. 4 did not encourage us to implement No. 5. Then, in his thesis [11], D. R. Taylor struck a fatal blow at the claims of No. 5. We summarize the results in Section 3.5 here.

If p_i and q_i are chosen to minimize $\angle(p_i, q_i)$ then biorthonormality fixes p_{i+1} and q_{i+1} and

$$\cos \angle(p_i, q_i) = \sigma_1 = \text{largest singular value of } \cos \Phi,$$
$$\cos \angle(p_{i+1}, q_{i+1}) = \sigma_2 = \text{smallest singular value of } \cos \Phi.$$

Now the best practical measure of the linear independence of the bases $\{p_1, \ldots, p_j\}$ and $\{q_1, \ldots, q_j\}$ is

$$\max_i \angle(p_i, q_i)$$

or, more practically,

$$\min_i \cos \angle(p_i, q_i).$$

From this point of view the best choice at a double step is to maximize

$$\min\{\cos \angle(p_i, q_i), \cos \angle(p_{i+1}, q_{i+1})\}.$$

Clearly, No. 5 is a poor choice. Taylor proves (Theorem 3.1) that this maximum is $2\sigma_1\sigma_2/(\sigma_1 + \sigma_2)$, the harmonic mean of σ_1 and σ_2.

These results make precise the following intuitive picture. If $\angle(r_i, s_i)$ is large but the planes spanned by (r_i, r_{i+1}), (s_i, s_{i+1}) are quite close to each other then our modifications to Lanczos will pay off handsomely. If, however, one of the angles between the planes is nearly a right angle then our device will not help.

Criterion for Choosing a 2×2 *Pivot.* When Factorization No. 4 is chosen then

$$(q_i, q_{i+1}) = (r_i, r_{i+1}) \begin{bmatrix} 1 & -\omega_{i+1}/\theta_i \\ 0 & 1 \end{bmatrix}, \qquad (p_i, p_{i+1}) = (s_{i+1}, s_i) \begin{bmatrix} 1 & -\omega_i/\theta_i \\ 0 & 1 \end{bmatrix},$$

to within scalar multiples. Please note the interchange. Hence

$$\cos\angle(q_i, p_i) = \cos\angle(r_i, s_{i+1}) = \frac{\theta_i}{\|r_i\| \cdot \|s_{i+1}\|} =: \psi_i,$$

$$\cos\angle(q_{i+1}, p_{i+1}) = \frac{(\theta_i - \omega_i\omega_{i+1}/\theta_i)}{\left\|r_{i+1} - \left(\frac{\omega_{i+1}}{\theta_i}\right) r_i\right\| \cdot \left\|s_i - \left(\frac{\omega_i}{\theta_i}\right) s_{i+1}\right\|} =: \psi_{i+1}.$$

Both these numbers are readily computable, *without* forming the new vectors, provided that $r_{i+1}^* r_i$ and $s_{i+1}^* s_i$ are known. The choice between No. 1 and No. 4 reduces to a comparison of

$$\phi_1 = |\cos\angle(r_i, s_i)| \quad \text{and} \quad \phi_2 = \min\{|\psi_i|, |\psi_{i+1}|\}.$$

If $\phi_1 < 100\varepsilon$ and $\phi_2 < 100\varepsilon$ then our algorithm stops and reports failure. Otherwise, for a given *bias factor* we proceed as follows: if $\phi_1 \geqslant$ (bias factor)ϕ_2 then use standard Lanczos else use Factorization No. 4. When bias = 0 we recover the standard algorithm. Currently bias = 2.

Sometimes the test declares that a standard Lanczos step is safe. In such cases, ψ_1 and ψ_2 are not used and their computation may be regarded as an overhead of four inner products. Fortunately no matrix-vector products are "wasted". The four inner products arise as follows. We need

$$\|\tilde{s}_i^*\|^2 = \left\|s_i^* - \left(\frac{\omega_i}{\theta_i}\right) s_{i+1}^*\right\|^2 = \|s_i\|^2 - 2\left(\frac{\omega_i}{\theta_i}\right) s_i^* s_{i+1} + \left(\frac{\omega_i}{\theta_i}\|s_{i+1}\|\right)^2,$$

$$\|\tilde{r}_{i+1}\|^2 = \left\|r_{i+1} - \left(\frac{\omega_{i+1}}{\theta_i}\right) r_i\right\|^2 = \|r_{i+1}\|^2 - 2\left(\frac{\omega_{i+1}}{\theta_i}\right) r_i^* r_{i+1} + \left(\frac{\omega_{i+1}}{\theta_i}\|r_i\|\right)^2.$$

We may regard the second and third terms on the right-hand sides as the price to be paid for knowing that a standard Lanczos step is safe. Observe that $\tilde{r}_{i+1}$ and $\tilde{s}_i^*$ are not computed.

To summarize, let us write $R_i = (r_i, r_{i+1})$, $S_i = (s_i, s_{i+1})$.
The look-ahead part of the algorithm comprises the computation of r_{i+1}, s_{i+1} and the unknown elements of $S_i^* R_i$, $R_i^* R_i$ and $S_i^* S_i$. Before specifying the algorithm we describe the bumpy tridiagonal matrix J.

5. J, The Projection of B. The standard (biorthogonal) Lanczos algorithm produces a tridiagonal matrix J_j by the end of step j. The look-ahead algorithm produces a block tridiagonal matrix, also called J_j, and written as

$$J_j = \begin{bmatrix} A_1 & \Gamma_2 & & & \\ B_2 & A_2 & \Gamma_3 & \ddots & \\ & B_3 & A_3 & \ddots & \ddots \\ & & \ddots & \ddots & \Gamma_j \\ & & & B_j & A_j \end{bmatrix}.$$

The diagonal blocks are capital α's and the subdiagonal blocks are capital β's. The A_i are either 1×1 or 2×2 and the B_i and Γ_i are shaped conformably. For

convenience in finding eigenvalues of J we have forced each B_i to have one of the following four forms

$$[+], \quad [0 \quad +], \quad \begin{bmatrix} + \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & + \\ 0 & 0 \end{bmatrix},$$

where $+$ stands for a positive quantity. It turns out that each Γ_i also has rank one.

The left and right Lanczos vectors will be grouped by step and written as $P_1^*, P_2^*, \ldots, P_j^*$ and $Q_1, \ldots, Q_j$, where sometimes Q_i is $n \times 1$ and sometimes $n \times 2$. We collect the vectors together in $\hat{Q}_j = (Q_1, \ldots, Q_j)$ and $\hat{P}_j = (P_1, \ldots, P_j)$. The matrix $\hat{Q}_j \hat{P}_j^*$ is the projector matrix onto the left and right Krylov subspaces and B's (oblique) projection into them is defined by

$$(\hat{Q}_j \hat{P}_j^*) B (\hat{Q}_j \hat{P}_j^*) = \hat{Q}_j J_j \hat{P}_j^*.$$

Thus J_j is the representation of this projection with respect to the bases given by the rows of $\hat{P}_j^*$ and the columns of $\hat{Q}_j$.

Please note that j is not the order of J_j.

6. The Look-Ahead Lanczos Algorithm (called LAL hereafter). We have chosen our notation to camouflage the complications caused by the fact that each step may be either a single one or a double one. It turns out that quantities are computed in a somewhat different order and way from the standard two-sided Lanczos algorithm (called LAN) and the reader may find the differences loom larger than the similarities. We have found it helpful to think that step i takes certain residual matrices R_i and S_i, decides how to modify them, then turns them into the new Q_i and P_i^*, and finally computes part of the next set of residual matrices.

It is convenient to define the index l by

$$l = 1 + \text{order}(\hat{Q}_{i-1}).$$

Thus by the end of step i we shall have

$$Q_i = \begin{cases} q_l, & \text{if step } i \text{ is single,} \\ (q_l, q_{l+1}) & \text{if step } i \text{ is double.} \end{cases}$$

Similarly for P_i^*. However, in all cases, $R_i = (r_l, r_{l+1})$ and $S_i = (s_l, s_{l+1})$.

In describing LAL we call any computation involving n scalar multiplications or divisions a *vector* operation and abbreviate it as 1 v.op. The algorithm requires that the user supply a subroutine (or subroutines) for computing Bx and y^*B from x and y^*. The cost of these operations will be problem dependent. We stress that this is the only way in which B enters the process.

Step i of LAL. On hand are P_{i-1}^*, Q_{i-1} (both are null when $i = 1$), and z_i which is a multiple of column 1 of Γ_i ($z_1 = 1$), together with nonnull residual vectors r_l, s_l^* and scalars $\omega = \omega_l = s_l^* r_l$, $\|r_l\|$, $\|s_l^*\|$.

1. Look-ahead:

(a) Compute $R_i = (r_l, r_{l+1})$ and $S_i^* = (s_l, s_{l+1})^*$.

$$r_{l+1} = Br_l - Q_{i-1} z_i \omega,$$
$$s_{l+1}^* = s_l^* B - \omega p_{l-1}^*,$$

(2 matrix-vector products + 2 or 3 vector ops.)

(b) Compute needed inner products.

$$\theta = s_l^* r_{l+1} = s_{l+1}^* r_l, \qquad \hat{\omega} = s_{l+1}^* r_{l+1},$$
$$r_l^* r_{l+1},\ s_l^* s_{l+1},\ \|r_{l+1}\|,\ \|s_{l+1}\|.$$

(6 v.ops.)

(c) Compute cosines of important angles.

$$\phi_1 = \cos\angle(r_l, s_l) = \omega/\|r_l\| \cdot \|s_l\|, \qquad \phi_2 = 0;$$

if $\theta = 0$ then go to step 2, otherwise

$$\tau_1 = \omega/\theta, \qquad \tau_2 = \hat{\omega}/\theta,$$
$$\|\tilde{r}_{l+1}\| = \sqrt{\|r_{l+1}\|^2 - 2\tau_2(r_l^* r_{l+1}) + \tau_2^2\|r_l\|^2},$$
$$\|\tilde{s}_l\| = \sqrt{\|s_l\|^2 - 2\tau_1(s_l^* s_{l+1}) + \tau_1^2\|s_{l+1}\|^2},$$
$$\psi_1 = \cos\angle(r_l, s_{l+1}^*) = \theta/\|r_l\| \cdot \|s_{l+1}^*\|,$$
$$\psi_2 = \cos\angle(\tilde{r}_{l+1}, \tilde{s}_l^*) = (\omega\tau_2 - \theta)/\|\tilde{r}_{l+1}\| \cdot \|\tilde{s}_l^*\|,$$
$$\phi_2 = \min\{|\psi_1|, |\psi_2|\}.$$

(0 v.ops.)

2. Test for failure: if $|\phi_1| < \text{tol}$ and $\phi_2 < \text{tol}$ then exit with error message.

3. Select: if $|\phi_1| \geqslant$ (bias factor) $\cdot\ \phi_2$ then take a single step, otherwise take a double step.

4. Side-by-side details of a single/double step.

Single		Double
	(a) factor $W = \begin{bmatrix} \omega & \theta \\ \theta & \hat{\omega} \end{bmatrix} = VU.$	
$\beta_l = \|r_l\|\sqrt{\phi_1}$,		$\Delta = \operatorname{diag}(\beta_l, \beta_l\beta_{l+1})$
$\gamma_l = \omega/\beta_l$		$= \operatorname{diag}\big(\|r_l\|\sqrt{\psi_1}, \|\tilde{r}_{l+1}\|\sqrt{\psi_2}\big)$
(0 v.ops.)		(0 v.ops.)
		$V = \begin{bmatrix} \tau_1 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} \theta & 0 \\ 0 & \theta - \omega\tau_2 \end{bmatrix}\Delta^{-1},$
		$U = \Delta\begin{bmatrix} 1 & \tau_2 \\ 0 & 1 \end{bmatrix}.$
	(b) form Q_l and P_l^*	
$q_l = r_l/\beta_l$,		$Q_l = R_l U^{-1}$,
$p_l^* = s_l^*/\gamma_l$		$P_l^* = V^{-1}S_l^*$
(2 v.ops.)		(6 v.ops.)
	(c) complete Γ_l, B_l	
$\Gamma_l = z_l\gamma_l$,		$\Gamma_l = z_l[\omega, \psi_2]\Delta^{-1}$,
$B_l = \beta_l$ or $[0 \quad \beta_l]$		$B_l = \begin{bmatrix} \beta_l \\ 0 \end{bmatrix}$ or $\begin{bmatrix} 0 & \beta_l \\ 0 & 0 \end{bmatrix}$
(0 v.ops.)		(0 v. ops.)

Single		Double
	(d) form new residuals	
$r_{l+1} = r_{l+1}/\beta_l,$		$r_{l+2} = (BQ_l - Q_{l-1}\Gamma_l)\begin{bmatrix}0\\1\end{bmatrix},$
$s^*_{l+1} = s_{l+1}/\gamma_l$		$s^*_{l+2} = p_l^*B - \beta_l p^*_{l-1}$ $= [0 \quad 1](P_l^*B - B_l P^*_{l-1})$
(2 v.ops.)		(2 matrix-vector products + 2 or 3 v.ops.)
	(e) form A_l	
$\alpha_l = 1/\tau_1$		$A_l = \begin{bmatrix} \tau_2 & p_l^*Bq_{l+1} \\ \beta_{l+1} & -\dfrac{\tau_1\beta_{l+1}\psi_1}{\psi_2} p_l^*Bq_{l+1} \end{bmatrix}$
(0 v.ops.)		(1 v.ops.)
	(f) biorthogonalize	
$r_{l+1} \leftarrow r_{l+1} - q_l\alpha_l,$		$r_{l+2} \leftarrow r_{l+2} - Q_l A_l \begin{bmatrix}0\\1\end{bmatrix},$
$s^*_{l+1} \leftarrow s^*_{l+1} - \alpha_l p_l^*$		$s^*_{l+2} \leftarrow s^*_{l+2} - [1 \quad 0]A_l P_l^*$
(2 v.ops.)		(4 v.ops.)
	(g) inner products for next step	
$\|r_{l+1}\| \leftarrow \sqrt{\|r_{l+1}\|^2 - 2\alpha_l r_l^* r_{l+1} + \alpha_l^2\|r_l\|^2}/\beta_l,$		compute $\|r_{l+2}\|,$
$\|s_{l+1}\| \leftarrow \sqrt{\|s_{l+1}\|^2 - 2\alpha_l s_l^* s_{l+1} + \alpha_l^2\|s_l\|^2}/\gamma_l,$		$\|s_{l+2}\|,$
$\omega_{l+1} \leftarrow \hat{\omega}/\omega - \alpha_l^2$		$\omega_{l+2} = s^*_{l+2} r_{l+2}$
(0 v.ops.)		(3 v.ops.)
	(h) reset z	
$z_{i+1} = [1]$		$z_{i+1} = \begin{bmatrix} 1 \\ -\omega \cdot \psi_2/(\beta_l^2\beta_{l+1}) \end{bmatrix}$

end of step i of LAL.

Total cost of step i:

Look-ahead: 2 matrix-vector prods + 9 v.ops.

Single step: 6 v.ops.

Double step: 2 matrix-vector prods + 16 v.ops.

For comparison we note that the standard two-sided Lanczos algorithm which keeps $\|p_l^*\| = \|q_l\|$ requires 2 matrix-vector prods + 10 v.ops.

There are 3 different ways of advancing two steps

LAN: 4 matrix-vector prods + 20 v.ops.,

LAL, 1 double step: 4 matrix-vector prods + 25 v.ops.,

LAL, 2 single steps: 4 matrix-vector prods + 30 v.ops.

The *bias factor* in Step 3 is a programming device which permits LAL to implement standard Lanczos (bias = 0) or a sequence of double steps (bias = ∞). We do not claim that our setting (bias = 2) is optimal, but we doubt that it is far off.

7. Incurable Breakdown and the Mismatch Theorem. In Section 4 we mentioned that it is possible for both the 1×1 pivot ω_j and the 2×2 pivot W_j to be singular. In principle we could then consider the leading 3×3 submatrix of the reduced moment matrix as a pivot. If it too were singular we could consider the 4×4 candidate and so on. If *all* such principal submatrices are singular then we say that

the breakdown at step j is *incurable*. In his thesis [11] Taylor proves the following surprising consequence (Theorem 4.2) of this ultimate disaster.

MISMATCH THEOREM (TAYLOR). *Let B have distinct eigenvalues and let J_j be the block tridiagonal matrix produced by the Look-ahead Lanczos algorithm at step j. If incurable breakdown occurs at step $j+1$ then each eigenvalue of J_j is an eigenvalue of B.*

We add that neither span$(\hat{P}_j)$ nor span$(\hat{Q}_j)$ is invariant under B. The reason for the name of the theorem becomes clear in the proof. Each starting vector must contain components of eigenvectors that are not matched in the other starting vector.

8. How to Monitor the Linear Independence of the Lanczos Vectors. In practice we have found it convenient to normalize each Lanczos vector to have unit length, $\|p_i\| = \|q_i\| = 1$. Consequently,

$$P_j^* Q_j =: \Psi_j = \operatorname{diag}(\psi_1, \ldots, \psi_j), \qquad \psi_i > 0,$$

instead of the identity matrix. The formulas in both LAL and LAN must be changed accordingly. Moreover, the "Ritz" values θ at step j come from the generalized eigenvalue problem

$$\det(J_j - \theta \Psi_j) = 0.$$

One incentive for making this change is that our technique for updating "Ritz" values at each step (using the Hyman-Laguerre method [6]) extends without change to the generalized eigenproblem.

Another advantage is that the elements of Ψ indicate the quality of the Lanczos bases $\{p_1, \ldots, p_j\}$, $\{q_1, \ldots, q_j\}$. The standard measure of the linear independence of the columns of a matrix is its condition number for inversion. Recall that $P_j := (p_1, \ldots, p_j)$ and

$$\operatorname{cond}(P_j) := \left(\|P_j^* P_j\| \cdot \left\| (P_j^* P_j)^{-1} \right\| \right)^{1/2} = \|P_j\| \cdot \|P_j^+\|,$$

where P_j^+ is the pseudo inverse of P_j. The normalization $\|p_i\| = 1$, $i = 1, \ldots, j$, ensures that

$$\|P_j\|^2 := \|P_j^* P_j\| \leqslant \|\text{matrix of 1's}\| = j.$$

Similarly $\|Q_j\|^2 \leqslant j$. In exact arithmetic $P_j^* Q_j = Q_j^* P_j = \Psi_j =$ symmetric, and Ψ_j is invertible. By checking the four Moore-Penrose conditions it can be verified that

$$(P_j^*)^+ = Q_j \Psi_j^{-1}, \qquad Q_j^+ = \Psi_j^{-1} P_j^*.$$

So

$$\|P_j^+\| = \|(P_j^*)^+\| \leqslant \|Q_j\| \cdot \|\Psi_j^{-1}\|.$$

For the regular Lanczos algorithm $\|\Psi_j^{-1}\| = 1/\min_{1 \leqslant i \leqslant j} \psi_i$, and, in this case,

$$\operatorname{cond}(P_j) \leqslant \|P_j\| \cdot \|Q_j\| / \min_{1 \leqslant i \leqslant j} \psi_i \leqslant j / \min_{1 \leqslant i \leqslant j} \psi_i.$$

In practice, both $\|P_j\|$ and $\|Q_j\|$ grow much more slowly than does $\sqrt{j}$. Even $j^{1/4}$ is an overestimate in the cases we have tried. However, for Lanczos runs limited to 200 steps the factor j is not serious. With biorthogonalization maintained $1/\min_{1 \leqslant i \leqslant j} \psi_i$ is a very good—and cheap—measure of the quality of the Lanczos bases.

However, without explicit biorthogonalization of the p_i and q_k the relation $P_j^* Q_j = \Psi_j$ fails completely as soon as an eigenvalue of the pair (J_j, Ψ_j) stabilizes and after that point linear independence is lost for all practical purposes. There is no point in monitoring the ψ_i after this point.

9. Numerical Examples. We give a few examples which contrast the performance of the standard two-sided Lanczos algorithm and our Look-ahead variant. They illustrate the stabilizing effect of the new variant.

All our computations were done on one of the VAX 11/780's of the Computer Systems Research Group at the University of California, Berkeley.

The Look-ahead algorithm reduces to the regular algorithm when the bias factor = 0. We kept the bias = 2 for all our examples.

Example I:

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad r_1 = s_1 = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}, \quad \text{number of steps} = 6$$

The eigenvalues of B are the sixth roots of unity. The size of the matrix allows for the complete history.

The following table gives snapshots of a Lanczos run, exhibiting what we feel is the essential information.

l—the order of the J-matrix.

ϕ_1, ϕ_2—cosines of angles between various candidates for p_l^* and q_l. See Section 4 for precise definition.

$\rho(J)$—the spectral radius of J.

The goal of our algorithm is to keep ϕ_1 from sudden plunges towards 0. We could have used $\|J\|$ instead of $\rho(J)$, as indication of "instability". We fear the appearance of arbitrarily large "spurious" eigenvalues in J. We expect some of J's eigenvalues to stabilize, as l increases, at certain points in the complex plane. These points should be part of B's spectrum.

If a double step occurs in the Look-ahead algorithm for a particular value of l then ϕ_1 and ϕ_2 are not defined at $l + 1$ and such places are indicated by dashes.

	Regular Lanczos			Look-ahead Lanczos		
l_1	ϕ_1	ϕ_2	$\rho(J)$	ϕ_1	ϕ_2	$\rho(J)$
1	1.000e + 0	1.277e − 1	8.351e − 1	1.000e + 0	1.277e − 1	8.351e − 1
2	1.281e − 1	7.661e − 3	1.503e + 0	1.281e − 1	7.661e − 3	1.503e + 0
3	7.204e − 3	3.177e − 8	1.004e + 0	7.204e − 3	3.177e − 8	1.004e + 0
4	3.025e − 8	4.880e − 2	5.509e + 6	3.025e − 8	4.880e − 2	—
5	3.025e − 8	0.000e + 0	—	—	—	4.781e + 0
6	—	—	—	6.757e − 3	0.000e + 0	1.000e + 0

Comment: Step 1–3 of both Regular and Look-ahead Lanczos are identical. At step 4, Regular Lanczos normalizes s_4 and r_4 by factors of 10^{-3}, producing elements in J of size 10^6. Step 5 Regular Lanczos is then aborted because the vectors are now orthogonal to working precision.

The Look-ahead algorithm avoids the large element growth in J, by doing a double step. The resulting J-matrix had eigenvalues identical to B to working accuracy.

Example II: B is 12×12 block upper triangular with diagonal blocks shown below. The other upper triangular elements b_{ij} were randomly distributed in $[-5, 5]$.

$$B_1 = \begin{bmatrix} 95 & 2 \\ -2 & 95 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 2 & -95 \\ 95 & 2 \end{bmatrix}, \quad B_3 = [99], \quad B_4 = B_5 = [98],$$

$$B_6 = \begin{bmatrix} 25 & -50 \\ 50 & 25 \end{bmatrix}, \quad B_7 = [10^{-2}], \quad B_8 = \begin{bmatrix} 50 & -50 \\ 50 & 50 \end{bmatrix},$$

$r_1 = s_1 = [1, \ldots, 1]^*$.

Number of steps taken = 24.

Attributes. B is 12×12 and fairly far from normal. In particular, the five eigenvalues near 97 form an ill-defined cluster because the off-diagonal elements in the Schur form are all approximately the same magnitude as the separation between

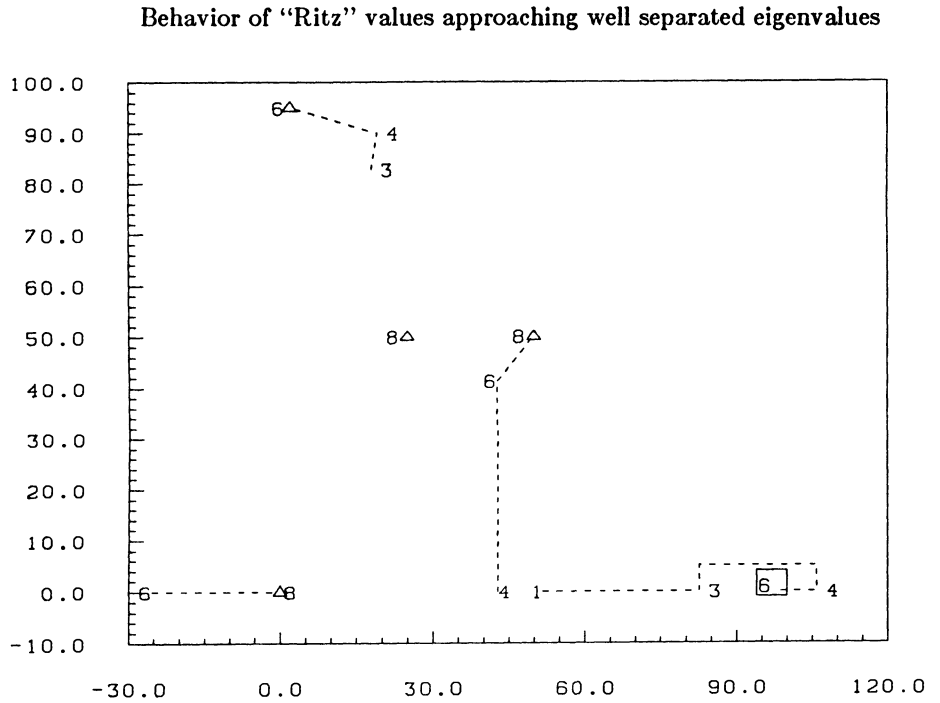

FIGURE 1

LAL with/without full rebiorthogonalization

△ *7 well-isolated eigenvalues of B*

□ *cluster of the remaining 5*

these five eigenvalues. Only half the figures in the double eigenvalue 98 will be significant and the other three eigenvalues are just a little better defined.

Results. 1. When full biorthogonalization was forced, the results from LAN and LAL were essentially the same. The process halted at step 12 with the isolated eigenvalues correct to at least 6 of the 7 decimals carried. The cluster was given as

$$95.001 \pm 1.9992i,\ 97.960 \text{ and } 98.046,\ 98.991.$$

Most of the $\psi_i\ (= p_i^* q_i)$ exceeded 0.1 (whereas $\psi_i = 1$ for symmetric matrices) but at steps 9 and 10, $\psi_9 = 0.014$ (for LAL), $\psi_9 = 0.007$ (for LAN).

2. When LAN and LAL were run with no biorthogonalization, they each lost biorthogonality by step 10 but neither algorithm came close to breakdown. The eigenvalue approximations were indistinguishable. By step 10 the 7 isolated eigenvalues were good to 6 decimals, in the neighborhood of the cluster of 5 each algorithm produced $99.11 \pm 3.26i$ and 96.72. At step 12 the cluster of 5 is

$$95.14 \pm 2.292i,\ 98.85 \pm 0.604i,\ 91.75 \quad \text{according to LAL}$$

but

$$94.96 \pm 2.325i,\ 98.87 \pm 0.577i,\ 45.6!! \quad \text{according to LAN.}$$

However the residual error bounds do show that 45.6 is indeed spurious.

Neither algorithm halts at step 12, but step 24 exhibits a peculiar feature that we cannot explain. Each algorithm has 12 Ritz values which are satisfactory approximations to the eigenvalues of B. Whereas the 12 extra Ritz values in LAN are very close

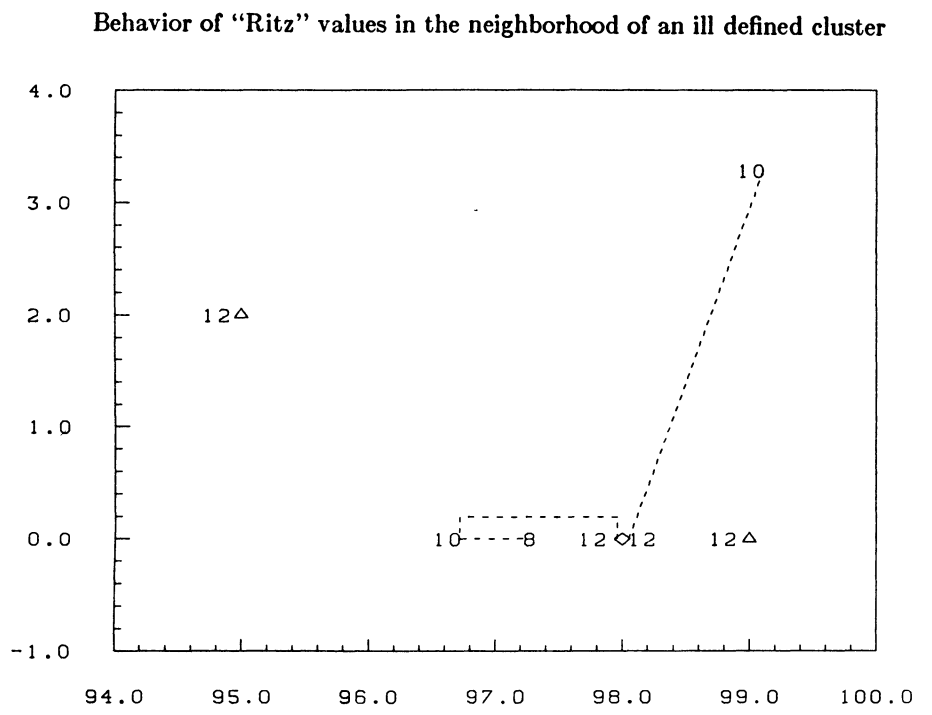

FIGURE 2

LAL with full rebiorthogonalization

△ *the 3 simple eigenvalues of B in the cluster*

◇ *the double eigenvalue* 98.0

copies of the true eigenvalues, some extra Ritz values in LAL appear to be almost random.

Figures 1 through 3 give some indication on how our LAL performs with or without full rebiorthogonalization. The numerals show the first step at which the Ritz value arrived at the indicated position. The dotted lines are an aid to imagination but have no claim to be trajectories. An isolated △ shows that the Ritz value never wandered far from its first appearance. For well-isolated eigenvalues, there was no essential difference with or without explicit rebiorthogonalization as depicted in Figure 1.

Example III: *Attributes.* B is 15×15 upper triangular matrix with evenly-spaced real eigenvalues $(-65, -55, \ldots, -5, 0, 5, \ldots, 65)$. The superdiagonal elements were randomly chosen numbers in the range $(-10, 10)$. The diagonal elements were

$$b_{jj} = \begin{cases} 10j - 75, & j = 1, \ldots, 7, \\ 0, & j = 8, \\ 10j - 85, & j = 9, \ldots, 15. \end{cases}$$

$$r_1 = s_1 = [1, 1, \ldots, 1]^*.$$

So this is an easy case from the point of view of the spectrum but it is, at the same time, far from normal. The Lanczos vectors p_i and q_i are not close at all.

Behavior of "Ritz" values in the neighborhood of an ill defined cluster

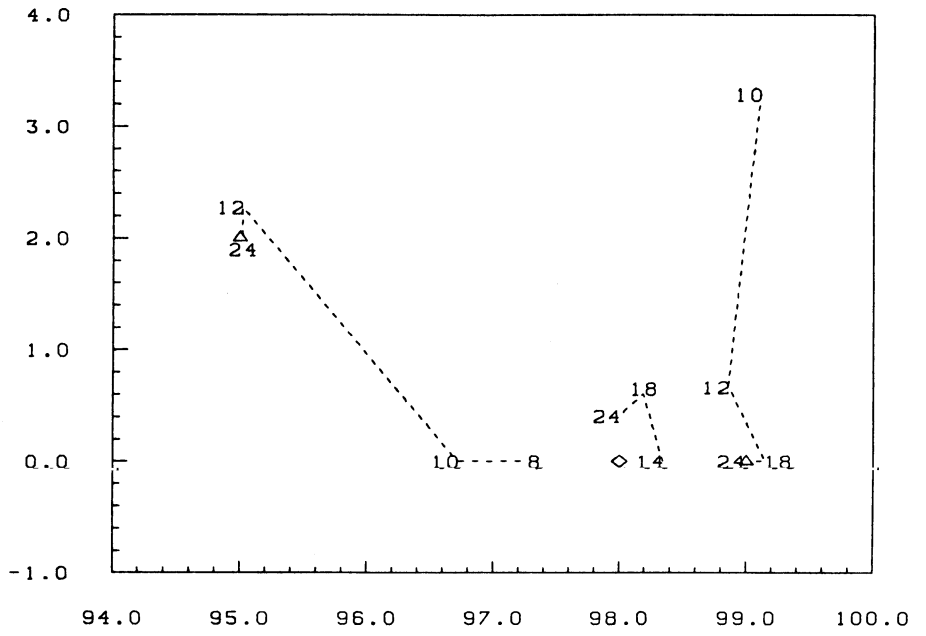

FIGURE 3

LAL without full rebiorthogonalization

△ *the 3 simple eigenvalues of B in the cluster*

◇ *the double eigenvalue* 98.0

Results. Excellent. No instability in LAN and consequently LAL used single steps throughout and produced the same results. After 15 steps all eigenvalues were good to 6 decimals (out of 7 carried) except for 0 which was represented by 3.4×10^{-5}. However, our stopping criterion was too strict to cause the algorithm to stop (the off-diagonal elements only dropped by a factor of 1000). After step 15, the two algorithms began to differ but by step 30 both had essentially obtained duplicates of the 15 eigenvalues.

With any form of biorthogonalization both algorithms terminated at step 15 with eigenvalues correct to at least 6 figures. The zero eigenvalue was given as $6 \times 10^{-8} + 1.1 \times 10^{-10} i$.

Example IV: *Attributes.* B is 100×100, diagonal, with distinct eigenvalues.

$$B = \operatorname{diag}(1, 2, \ldots, 20, 41, 62, \ldots, 440, 481, 522, \ldots, 1260, 1331, 1382, \ldots, 2480, 2561, 2642, \ldots, 4019, 4100).$$

The interest here was to start with p_1 and q_1 different and random. In fact $\psi_1 = p_1^* q_1 = 0.06$.

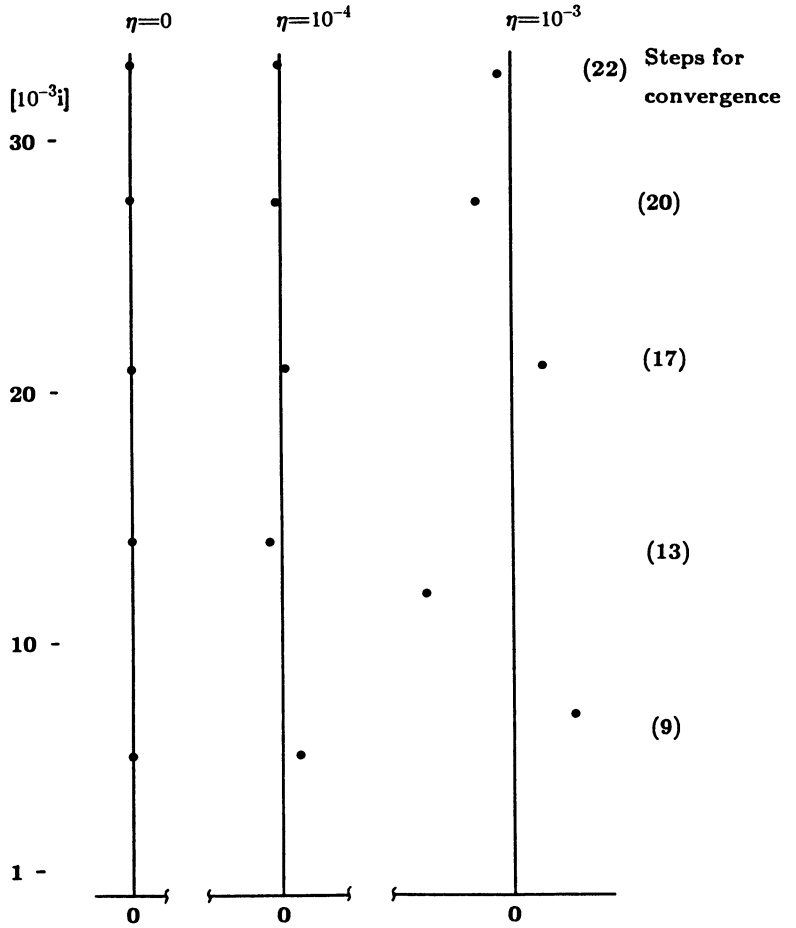

FIGURE 4
Spectrum of (F, M), $n = 94$

Results. Disappointing at first. After 19 steps the largest eigenvalue was good to 4 figures (4099.8) but no others had converged recognizably. However, the gap ratio $(\lambda_{100} - \lambda_{99})/(\lambda_{100} - \lambda_1)$, which governs the convergence rate, is only 0.02 and consequently even with $p_1 = q_1$ convergence will not be much faster. See [12, Chapter 12] for a fuller explanation. One interesting feature, not yet explained, is an unbalanced loss of biorthogonality. For example, $p_3^* q_{13} \approx \varepsilon$, $p_{13}^* q_3 \approx 10^5 \varepsilon$. At step 13 ψ_{13} dropped to 5×10^{-4} and most ψ_i were less than 10^{-2}.

We conclude that it is a mistake to begin with nearly orthogonal p_1 and q_1 unless they are known to be approximate eigenvectors of the same eigenvalue.

Example V: *Attributes.* The Max Planck Institute in Munich has supplied us with programs to generate matrices of the form $B = F^{-1}M$ arising in the study of plasma instability; of course, F is not inverted explicitly. There is a parameter η (resistivity) which can be varied. When $\eta = 0$ all the eigenvalues are pure imaginary. Only the largest few eigenvalues are of interest.

We have tried our codes on matrices of order $n = 34$, 94 and 598.

Results. Excellent. With a random starting vector ($p_1 = q_1$) the largest 8 eigenvalues converged to working accuracy in fewer than 30 steps. This phenomenon is independent of n, the size of the matrix. The dominant eigenvalue usually converged after 10 steps.

Figure 4 shows how the (reciprocals of) the eigenvalues leave the imaginary axis as η increases from 0.

10. Conclusion. The Look-ahead algorithm (LAL) is more complicated than the standard two-sided Lanczos process (LAN). We have so far found only one case (Example I) in which LAN fails while LAL succeeds. Often they both perform very satisfactorily. Each can be used with rebiorthogonalization of the left and right Lanczos vectors against each other. This safeguard increases the cost but makes the results very close to those produced by exact arithmetic. For short runs ($j < 30$) on vector computers this extra cost of explicitly forcing the Lanczos vectors to be biorthogonal may be a small fraction of the cost of the other parts of the algorithm.

The principal reason for consenting to use two sequences of vectors (P_j and Q_j) instead of one (as in Arnoldi's method, see [9]) is the expectation of convergence of the dominant eigenvalues after a small number of steps. When both the column and row subspaces contain, respectively, $\sqrt{\varepsilon}$ approximations to the eigenvectors of λ then one of the Ritz values will be an ε-approximation to λ. This cannot happen with one-sided approximations unless the matrix is normal.

LAL is an effective tool for extracting a few eigenvalues of large nonnormal matrices. Whether it is better than its rivals remains to be seen. At such a time it will be appropriate to discuss computable error estimates, efficient ways to monitor convergence of the eigenvalues of J_j, and other practical details.

Center for Pure and Applied Mathematics
University of California
Berkeley, California 94720

1. C. DAVIS & W. KAHAN, "The rotation of eigenvectors by a perturbation—III," *SIAM J. Numer. Anal.*, v. 7, 1970, pp. 1–46.

2. W. GRAGG, Notes from a "Kentucky Workshop" on the moment problem and indefinite metrics.

3. A. S. HOUSEHOLDER, *The Theory of Matrices in Numerical Analysis*, Blaisdell, New York, 1964.

4. W. KAHAN, B. N. PARLETT & E. JIANG, "Residual bounds on approximate eigensystems of nonnormal matrices," *SIAM J. Numer. Anal.*, v. 19, 1982, pp. 470–484.

5. C. LANCZOS, "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators," *J. Res. Nat. Bur. Standards*, v. 45, 1950, pp. 255–282. See pp. 266–270.

6. B. N. PARLETT, "Laguerre's method applied to the matrix eigenvalue problem," *Math. Comp.*, v. 18, 1964, pp. 464–487.

7. B. N. PARLETT, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, N. J., 1980.

8. B. N. PARLETT & J. R. BUNCH, "Direct methods for solving symmetric indefinite systems of linear equations," *SIAM J. Numer. Anal.*, v. 8, 1971, pp. 639–655.

9. Y. SAAD, "Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices," *Linear Algebra Appl.*, v. 34, 1980, pp. 269–295.

10. Y. SAAD, "The Lanczos biorthogonalization algorithm and other oblique projection methods for solving large unsymmetric eigenproblems," *SIAM J. Numer. Anal.*, v. 19, 1982, pp. 484–500.

11. D. R. TAYLOR, *Analysis of the Look Ahead Lanczos Algorithm*, Ph.D. thesis, University of California, Berkeley, 1982.

12. J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Oxford Univ. Press, Oxford, 1965.