

THE LANCZOS ALGORITHM FOR SYMMETRIC EIGENVALUE PROBLEMS*

D. S. Scott
Mathematics and Statistics Research Department
Computer Sciences Division
Union Carbide Corporation, Nuclear Division
Oak Ridge, Tennessee

*Research sponsored by the Applied Mathematical Sciences Research Program, Office of Energy Research, U.S. Department of Energy under contract W-7405-eng-26 with the Union Carbide Corporation.

By acceptance of this article, the publisher or recipient acknowledges the U.S. Government's right to retain a non-exclusive, royalty-free license in and to any copyright covering the article.

DISCLAIMER

This book was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

fy

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

ABSTRACT

The Lanczos algorithm was originally used to tridiagonalize symmetric matrices but it was soon replaced by more effective methods based on explicit orthogonal similarity transformations. The algorithm was revived as an effective scheme for solving sparse symmetric eigenvalue problems. This paper surveys the historical development and the current state of the Lanczos algorithm.

$$T_j = Q_j^* A Q_j,$$

which means that T_j is the projection of A onto the subspace $K_j(r_0)$.

Using T_j it is possible (via the Rayleigh-Ritz procedure) to compute the optimal approximations to eigenpairs of A from the subspace $K_j(r_0)$ as follows. Let $\theta_j = \text{diag}(\theta_1^{(j)}, \theta_2^{(j)}, \dots, \theta_j^{(j)})$ be the eigenvalues of T_j and let $S_j = (s_1^{(j)}, s_2^{(j)}, \dots, s_j^{(j)})$ be the corresponding matrix of eigenvectors, so that $T_j = S_j \theta_j S_j^*$. Then the θ 's are the optimal eigenvalue approximations (Ritz values) and the columns of $Y_j = (y_1^{(j)}, y_2^{(j)}, \dots, y_j^{(j)}) = Q_j S_j$ are the corresponding approximate eigenvectors (Ritz vectors).

Since the Lanczos vectors are orthogonal the process must terminate ($\beta_j = 0$) for some $j \leq n$. Originally, the algorithm was used as a way of tridiagonalizing A and early termination ($\beta_j = 0$ for $j < n$) was a nuisance requiring the selection of a vector orthogonal to all the current Lanczos vectors to continue the process. Now the algorithm is used primarily to find a few eigenpairs of a large matrix and early termination is a delight since the computed eigenpairs will be exact.

Much is known about the accuracy of Rayleigh-Ritz approximations obtained from Krylov subspaces. Explicit bounds were obtained by [Kaniel 1966], corrected in [Paige 1971], and improved in [Saad 1980] (see [Parlett 1980] for a unified presentation) based on the characterization of $K_j(r_0)$ in terms of polynomials

$$K_j(r_0) = \{p(A)r_0 \mid \deg p < j\}.$$

Three factors influence the accuracy of an approximation to an eigenpair

(λ, z) :

1. $z^* r_0$, the component of r_0 in the direction of z . The larger the component the better the approximation.
2. The location of λ in the spectrum of A . The closer λ is to the edge of the spectrum the better the approximation.
3. The relative separation of λ from the other eigenvalues of A . The bigger the separation the better the approximation.

1. measures the situation at the beginning while 2 and 3 indicate the convergence rate as j increases. It is common for the Ritz approximations to well separated extreme eigenvalues to be correct to working accuracy for values of $j \ll n$ even when r_0 is chosen randomly. On the other hand if $z^* r_0 = 0$ then the whole Krylov subspace $K_j(r_0)$ will be orthogonal to z and z will never be detected. For simple eigenvectors this situation has probability zero but for multiple eigenvectors it is inevitable. The Lanczos algorithm without modification is incapable of finding more than one eigenvector for any eigenvalue and so is unable to determine the multiplicity of any eigenvalue it finds.

It is of theoretical interest to note that for most matrices it is possible to choose a starting vector for which no eigenvalue converges until $j = n$ [Scott 1979a]. Its importance is that it gives insight into why such behavior is never seen in practice.

The Lanczos algorithm is also an attractive method for solving generalized eigenvalue problems $A - \lambda M$ where M is positive definite. If $M = LL^*$ is the Choleski factorization of M then the pencil $A - \lambda M$ can be reduced to an equivalent pencil $L^{-1}AL^{-*} - \lambda I$. The Lanczos algorithm

can then be applied to the operator $L^{-1}AL^{-*}$ without forming the matrix explicitly. The Lanczos algorithm can even be applied directly to the pencil $A-\lambda M$ without factoring M provided that some method exists (for instance conjugate gradients) for solving equations $Mx = b$. The details are in section 15-11 of [Parlett 1980].

2. Early Practice

When he introduced the algorithm [Lanczos 1950], Lanczos was aware that the quantities computed by the algorithm in finite precision arithmetic would eventually diverge entirely from their theoretical counterparts.

Equations (1.1) and (1.2) had to be modified to

$$(2.1) \quad Aq_j - Q_j T_j = r_j e_j^* + F_j$$

$$(2.2) \quad I - Q_j^* Q_j = H_j$$

where F_j and H_j account for the rounding errors. It was discovered that while (1.1) is always nearly satisfied ($\|F_j\|$ is tiny), the orthogonality among the Lanczos vectors was completely destroyed ($\|H_j\| \geq 1$).

So Lanczos suggested that the simple algorithm be modified so that the computed r_j is explicitly orthogonalized against all the earlier q 's. This process is known as reorthogonalization and with careful implementation it is sufficient to assure that the Lanczos vectors are orthogonal to working accuracy. This modification greatly increases the cost of the algorithm but allowed it to be successfully used to reduce symmetric matrices to tridiagonal form. With the advent of explicit orthogonal similarity transforms, more efficient methods of tridiagonalizing symmetric matrices were developed and interest in the Lanczos algorithm died.

The increased use of sparse matrices sparked new interest in the Lanczos algorithm. Of central importance is that the only way in which the matrix A enters the calculation is in the formation of the matrix-vector product Aq_j . Thus A may be stored in any compact way whatever which allows such products to be computed efficiently. Also important is that only 3 n -vectors are needed in fast memory since the earlier q 's can be put into secondary storage until they are needed to form the Ritz

vectors. In fact in [Paige 1972] it is shown that with clever programming two n -vectors are sufficient. On the other hand full reorthogonalization requires all the Lanczos vectors at each step and the use of this device nullifies the second advantage.

In practice the simple Lanczos algorithm (without reorthogonalization) was found to behave quite oddly. After computing an accurate approximation to an eigenpair of the matrix A the algorithm "forgets" that it has done so and starts computing another. For larger values of j there will be two Ritz pairs which accurately approximate the same eigenpair of A . The Ritz vectors will be parallel which manifests the loss of linear independence among the Lanczos vectors. Still later a third copy will be found. The simple algorithm will grind away forever computing more and more copies of extreme eigenvalues for each new interior one which converges.

The loss of orthogonality in the algorithm left the users in a quandary. The use of full reorthogonalization is quite expensive and becomes more so as j increases. On the other hand allowing the orthogonality to dissipate required successful identification of the spurious copies as well as raising theoretical questions concerning the accuracy of the computed results. The most effective solution combined two ideas.

Iterative Lanczos periodically restarts the algorithm with a special starting vector chosen as some linear combination of Ritz vectors from the current subspace. If full reorthogonalization is used then the start over point is determined by cost and storage considerations while

some kind of monitoring of the loss orthogonality is required if no reorthogonalization is performed (see [Parlett and Scott 1979] for a discussion of monitoring the loss of orthogonality).

The block Lanczos algorithm replaces each Lanczos vector q_i by an $n \times p$ orthonormal matrix P_i . The resulting matrix T is block tridiagonal with $p \times p$ blocks. The off diagonal blocks can be chosen to be triangular so that T is a band matrix with $2p + 1$ diagonals. Block Lanczos has a theoretical advantage in that multiple eigenvalues can be found (up to the block size p). It has three further advantages. The loss of orthogonality tends to be directly dependent on the number of Lanczos steps taken. Thus with block Lanczos, when loss of orthogonality requires an iteration, the size of the current subspace will be larger so that more and better eigenvalue approximations will be available. More importantly, p vectors can be chosen for the new starting block which allows more information to be retained from one step to the next. Finally in many applications the matrix A will be in secondary storage and will have to be fetched (in pieces) to form the matrix-vector product Aq . In such a situation the cost of multiplying several vectors will be little more than the cost of multiplying one.

Iterative block Lanczos programs were successfully implemented by [Cullum and Donath 1974] and by [Underwood 1975]. It should be noted that while block Lanczos has strong theoretical justification, iteration of the algorithm is employed only because of problems associated with loss of orthogonality and information is always lost when the current subspace is discarded.

3. The Results of Paige

A deep understanding of the Lanczos algorithm was first obtained by Chris Paige in his PhD thesis [Paige 1971]. Unfortunately his thesis was never published and some of the most valuable results have become generally known only recently. We present here some of his most important discoveries.

He found that it is possible to monitor the convergence of the algorithm at step j by computing the residual norm $\|Ay_i - y_i\theta_i\|$ of a Ritz pair (θ_i, y_i) without computing the Ritz vector y_i at all! By definition $y = Q_j s_i$, where (θ_i, s_i) is an eigenpair of T_j , and so multiplying equation (1.1) on the right by s_i we find

$$AQ_j s_i - Q_j T_j s_i = r_j e_j^* s_i$$

$$Ay_i - Q_j s_i \theta_i = r_j s_{ji}$$

$$Ay_i - y_i \theta_i = r_j s_{ji}$$

where s_{ji} is the bottom element of s_i . Taking norms we have

$$(3.1) \quad \|Ay_i - y_i \theta_i\| = \beta_j |s_{ji}| \equiv \beta_{ji}.$$

In finite precision arithmetic it is equation (2.1) which governs the algorithm. Fortunately $\|F_j\|$ is always small in practice and specific bounds are given in [Paige 1976]. Thus β_{ji} , which can be computed by solving a small tridiagonal eigenvalue problem, is an inexpensive and accurate estimate of the residual norm of a Ritz vector. There remained the fear that the loss of linear independence among the Lanczos vectors might lead to a Ritz vector y with $\|y\| \ll 1$, which would invalidate the use of β_{ji} as an estimate of the accuracy of the eigenvalue. A complicated analysis in [Paige 1971] which also appears in [Scott 1978] shows that

This result became known soon after its discovery but the proof and some important intermediate results have only just appeared in the open literature [Parlett 1980].

The quantity of interest in monitoring loss of orthogonality is the j -vector $Q_j^* q_{j+1}$ which would be zero in exact-arithmetic. Computational experience indicates that the elements of the vector grow in magnitude with increasing j but show no discernable pattern. The pattern can be seen only by making a change of basis by premultiplying $Q_j^* q_{j+1}$ by S_j^* the eigenvector matrix of T_j .

$$S_j^* Q_j^* q_{j+1} = Y_j^* q_{j+1},$$

the inner products of the Ritz vectors with q_{j+1} . Paige proved that

$$(3.2) \quad |y_i^* r_j| |s_{ji}| = \gamma_{ji}$$

where γ_{ji} is a messy expression involving n , j , A , and details of the arithmetic unit. Bounds for γ_{ji} exist and in practice

$$\gamma_{ji} \approx \epsilon \|A\|$$

is always satisfied. Since $r_j = q_{j+1} \beta_j$ equation (3.2) can be rewritten as

$$(3.3) \quad y_i^* q_{j+1} = \gamma_{ji} / \beta_{ji}.$$

Where β_{ji} defined in equation (3.1) is the residual norm of y_i to working accuracy.

Thus orthogonality is lost only in the directions of converging Ritz vectors. Loss of linear independence occurs when a Ritz vector converges to working accuracy at which point a second Ritz vector appears essentially parallel to the first. The second Ritz vector perturbs the

first away from the eigenvector (the β_{ji} 's are about $\sqrt{\epsilon||A||}$) and then the two vector converge together until the $\beta_{ji} \approx \epsilon||A||$ at which point a third copy appears.

Paige recommended that the simple Lanczos algorithm be run past the loss of linear independence until all the eigenvalues of interest had converged. This method was quite successful although it required the identification of the spurious copies. The identification is easy if the Ritz vectors are computed (since copies will be parallel instead of orthogonal) but it is not easy to make the identification without the Ritz vectors. The spurious copies also make it more difficult to determine when all the desired eigenvalues have converged.

4. Selective Orthogonalization

Selective orthogonalization, the idea of orthogonalizing q_{j+1} against converged Ritz vectors only, was suggested to me in 1976 by Parlett, without knowledge of equation (3.3), as a way of suppressing repeated copies of Ritz vectors. The process seemed to be effective and the discovery of equation (3.3) in Paige's thesis gave a theoretical justification to the idea as well as the analytical tool needed for its analysis. The important ideas are in [Parlett and Scott 1979] and a more detailed analysis in [Scott 1978].

The basic idea is to maintain a level of independence among the Lanczos vectors measured by $\kappa_j = \|1 - Q_j^* Q_j\|$. $\kappa_1 = \|1 - q_1^* q_1\|$ will always be tiny ($\sim \epsilon$) since it just measures how well a vector can be normalized. κ_j is an increasing function of j and in the simple Lanczos algorithm κ_j will surpass 1 as the second copy of a Ritz vector appears and the Lanczos vectors become linearly dependent.

If $\kappa_j \sim \epsilon$ is to be maintained then full reorthogonalization is required but the important observation is that quite modest values of κ (say $\kappa = .01$) are quite sufficient to prevent repeated copies from appearing. Thus selective orthogonalization allows the simple Lanczos algorithm to proceed as long as $\kappa_j < \kappa$ for some predetermined κ . It then steps in and performs the needed orthogonalizations when κ_j threatens to grow bigger than κ .

Using equation (3.3) it can be shown that as long as

$$(4.1) \quad \beta_{ji} > \epsilon \|A\| / \kappa,$$

then $|y_i^{(j)*} q_{j+1}| < \kappa$ and

$\|I - Q_{j+1}^* Q_{j+1}\| < (j + 1) \kappa$. Except for the factor $j + 1$ this is the desired result. The factor $j + 1$ is never seen in practice and it will be omitted from the further discussion.

Selective orthogonalization monitors the β_{ji} until some β_{ji} crosses the threshold, that is satisfies

$$(4.2) \quad \beta_{ji} < \epsilon \|A\| / \kappa.$$

The algorithm then pauses and computes the corresponding Ritz vector $y_i^{(j)}$ by recalling the Lanczos vectors. r_j is then explicitly orthogonalized against ("purged of") $y_i^{(j)}$. It is also necessary to purge r_{j+1} of $y_i^{(j)}$. Then later r 's will be essentially orthogonal to $y_i^{(j)}$ due to the three term recurrence. It is only necessary to monitor (bound) $|y_i^{(j)*} r_k|$ and to purge again (twice) when $|y_i^{(j)*} r_k| > \kappa$. The details are in [Parlett and Scott 1979].

If suppressing repeated copies were the sole criterion for the choice of κ then a value like $\kappa = .01$ would be sufficient. However there are two compelling reasons to insist that equation (3.3) be satisfied by the unconverged Ritz vectors even after some Ritz vector has crossed the threshold. Firstly equation (3.3) provides the only inexpensive way of measuring the loss of orthogonality in the direction of a given Ritz vector and secondly it is only equation (3.3) which guarantees that $|y_i^{(j)*} r_{j+1}| > \kappa$ will be satisfied by only a few Ritz vectors at most. This means that the orthogonalization of r_j against one Ritz vector must not significantly perturb the angles between r_j and the other Ritz vectors. This leads to a more stringent constraint on the value of κ as shown by the following analysis.

We need names for r_j and β_j before and after the orthogonalization and we will use r_j' and β_j' for the quantities before the orthogonalization. For convenience assume that only one Ritz y_m has crossed the threshold and needs to be purged from r_j' . Then

$$r_j \equiv r_j' - y_m \xi_m,$$

where $\xi_m = y_m^* r_j'$, and so for $k \neq m$,

$$y_k^* r_j = y_k^* r_j' - (y_k^* y_m) \xi_m.$$

In exact arithmetic both ξ_m and $y_k^* y_m$ would be zero but for our purposes only their product must be tiny. The factor $y_k^* y_m$ can be bounded directly by K . The analysis of ξ_m is more complicated but the gist of it is that

$$|\xi_m| = |y_m^* r_j'| = |\cos \langle y_m, r_j' \rangle| \beta_j'$$

and large values of $|\cos \langle y_m, r_j' \rangle|$ can only occur in combination with small values of β_j' so that ξ_m can be bounded by $K||A||$. See [Scott 1978] for the details. Thus the product of the two terms is bounded by $K^2||A||$ and this leads to the choice of $K = \sqrt{\epsilon}$.

Although $K = \sqrt{\epsilon}$ is much smaller than $K = .01$ a threshold vector will satisfy

$$\beta_{ji} \leq \sqrt{\epsilon} ||A||$$

which means that in most cases the Ritz vector will have converged to half the working accuracy while the Ritz value will have converged to working accuracy. Hence very few Ritz vectors will ever cross the threshold which explains the computational efficiency of the scheme.

There is another reason for choosing $K = \sqrt{\epsilon}$. The Ritz vectors computed by the algorithm will not be orthonormal to working accuracy.

Instead they will satisfy

$$|y_m^* y_k| \leq K.$$

Since the true eigenvectors are orthogonal it is reasonable to explicitly orthogonalize the Ritz vectors (in order of decreasing accuracy). The Rayleigh quotients of the new vectors will be perturbed by $O(K^2 \|A\|)$ and choosing $K = \sqrt{\epsilon}$ insures that this perturbation can be ignored.

Further insight into the behavior of selective orthogonalization can be obtained by taking the terms representing the orthogonalizations and consider them as perturbations to the matrix T_j . The new matrix T_j' will be tridiagonal except for the addition of nonzero spikes in certain columns which are multiples of the eigenvectors of T which generated the Ritz vector used in the orthogonalization. For example if $y_m^{(j)} = Q_j s_m^{(j)}$ is the first threshold vector then

$$AQ_j - Q_j T_j = r_j e_j^* + y_m^{(j)} \xi_m^{(j)}$$

can be rewritten as

$$AQ_j - Q_j [T_j + s_m^{(j)} e_j^* \beta_m^{(j)}] = r_j e_j^*.$$

The choice of $K = \sqrt{\epsilon}$ insures that the perturbation $s_m^{(j)} \beta_m^{(j)}$ can be ignored in computing the eigensystem of later T 's except for the required orthonormalization of the computed Ritz vectors. See [Farlett 1980] for more details.

The current implementation (see [Scott 1979b]) employs selective orthogonalization only for those eigenpairs desired by the user. Orthogonality is allowed to dissipate in the direction of undesired eigenvectors. In theory this invalidates the a priori error estimates but the desired Ritz vectors still converge and their residual norms provide a posteriori error estimates. For problems where eigenpairs are desired

as

$$(4.4) \quad [y_i^* P_{j+1}] [B_j s_{ji}] = \gamma_{ji}.$$

The vector $B_j s_{ji}$ still contains all the needed information. It is

$\|B_j s_{ji}\|$ which measures convergence but it is a small element of

$B_j s_{ji}$ which signals loss of orthogonality. Thus

convergence \Rightarrow loss of orthogonality

but the arrow no longer goes both ways as the following example shows.

Let $R_0 = (q, Aq)$ for an arbitrary vector q . The columns of P_1 form an orthonormal basis of $\text{span}(R_0)$ so that $\text{span}(AP_1) = \text{span}(Aq, A^2q)$.

Thus R_1 will have rank 1 and one column of P_2 would be zero.

Numerically the second column of P_2 would be normalized noise and serious loss of orthogonality would occur even though no convergence had occurred.

The Lanczos algorithm with selective orthogonalization provides a black box method for computing a few eigenvalues and eigenvectors of a symmetric operator which is ideally suited for large sparse problems.

The details of the current implementation are in [Scott 1979b].

5. Current Research

Current research in the Lanczos algorithm has been focused on using it to solve other related problems. This section will survey current research areas.

A problem which arises in some applications is to compute all the distinct eigenvalues of a large sparse symmetric matrix. The Lanczos algorithm (with no reorthogonalization) was used by Cullum and Willoughby with a clever method of identifying and discarding repeated copies [Cullum & Willoughby 1978, 1980]. The main difficulty is to determine when to stop the process. $j \geq n$ is clearly needed but $j \geq 10n$ is sometimes needed to resolve clusters of eigenvalues and the cost of computing eigenvalues of such large tridiagonal matrices is non-trivial. More recently Parlett and Reid employed efficient method of monitoring the convergence of the algorithm which allows termination to occur as soon as possible [Parlett and Reid 1980].

In some applications (especially generalized eigenvalue problems) it is possible to factor the matrix A (or the shifted matrix $A - rI$) and run the Lanczos algorithm on A^{-1} (or $(A - \sigma I)^{-1}$). The inversion of the operator causes a transformation of the spectrum. The eigenvalues far from the shift σ are compressed near zero so that the eigenvalues near σ become the extreme ones with large relative separations. Thus by the Kaniel-Paige-Saad theory discussed in section 2, convergence of the algorithm to the eigenvalues near σ is quite rapid. If only one factorization is possible then the shift σ should be chosen near the center of the desired eigenvalues and Lanczos with selective orthogonalization should be used. (One of the codes describe in [Scott 1979] is designed for this context.)

If several factorizations can be computed then the questions are how long should the Lanczos algorithm be run with one factorization before computing a new factorization, what should the new shift parameter be, and how should the new starting vector (or vectors) be chosen? One convenient result is that the factorization of $A - \sigma I$ gives a count of the number of eigenvalues of A less than σ (equal to the number of negative pivots encountered). This means that eigenvalues cannot be missed by the algorithm without a warning message.

A preliminary code for this problem is described in [Eriksson and Ruhe 1979]. The code uses simple Lanczos with full reorthogonalization. The length of a Lanczos run is determined by storage and cost considerations. The new shift is chosen so that the last converged eigenvalue lies halfway between the new and old shift. The starting vectors are chosen randomly and are then orthogonalized against the converged vectors. The code employs a clever modification which improves the accuracy of the eigenvectors.

The author has a preliminary code which uses block Lanczos with no reorthogonalization. The end of a Lanczos run is signalled by storage considerations or significant loss of orthogonality ($\beta_{ji} < 10n\epsilon \|A\|$ say). The new shift bisects the p and $(p + 1)$ st unconverged eigenvalues (p is the blocksize) and chooses the p starting vectors as linear combinations of the unconverged eigenvectors. It is recommended that the blocksize be chosen as $p \approx \sqrt{m_{\text{band}}}$ where m_{band} is the bandwidth of the matrix A . This choice is made to try to equalize to some extent the cost of a Lanczos step and the cost of a factorization.

Both codes are capable of solving generalized problems $A-\lambda M$ including the important special case when M is diagonal and only positive semidefinite.

No direct comparisons of the codes have been made but I would expect that as the bandwidth of A increased (and with it the cost of a factorization) the better and better the block version would perform. Eriksson and Ruhe make no use of the unconverged Ritz vectors due to difficulties with multiple eigenvalues. Since block Lanczos essentially solves the multiple eigenvalue difficulty my code is able to retain more information at each iteration. However a multiplicity larger than the blocksize can cause my code to return erroneous results so more work is needed.

There is a close relationship between the conjugate gradients algorithm for solving $Ax = b$ (with initial guess $x_0 = 0$) and the Lanczos algorithm with $q_1 = b/\beta_0$. In theory the conjugate gradient solution at step j is

$$x_j = A_j^t b$$

where A_j is the projection of A onto the subspace $K_j(q_1) = \text{span}(b, Ab, A^2b, \dots, A^{j-1}b)$. Using quantities computed by the Lanczos algorithm,

$$x_j = Q_j T_j^{-1} Q_j^* b,$$

where $Q_j^* b = e_1 \beta_0$ by the special choice of q_1 . Due to rounding errors both conjugate gradients and the Lanczos algorithm diverge completely from their theoretical paths but the relation

$$(5.1) \quad x_j = Q_j T_j^{-1} e_1 \beta_0$$

continues to hold.

Equation (5.1) gives an easy explanation of the potential instability of conjugate gradients for indefinite matrices. If A is indefinite T_j can be singular and the formula breaks down. Numerically this corresponds to the solution x_j growing very large at one step and then shrinking at the next causing serious loss of accuracy due to cancellation. (T_j and T_{j+1} cannot both be ill conditioned unless A itself is).

This relationship between conjugate gradients and Lanczos was exploited in [Paige and Saunders 1975] to produce a stable algorithm, SYMMLQ, for indefinite matrices. However since SYMMLQ is less efficient than simple conjugate gradients it was not recommended for positive definite A .

At the cost of storing the Lanczos vectors (q 's) in secondary store, the Lanczos formulation offers several important advantages.

1. Indefinite A causes no appreciable extra expense. It is only necessary to test the condition of the small tridiagonal matrix T_j and take one more step Lanczos if T_j is ill conditioned.
2. Selective orthogonalization can be used to maintain robust linear independence among the Lanczos vectors. This will decrease the number of steps needed by the algorithm.
3. If the solution of more than one right hand side is needed then the subspace generated for one right hand side can be used to find a good starting guess for b_2 as

$$x_0 = Q_j T_j^{-1} Q_j^* b_2.$$

It is possible (even likely) that this x_0 will already be sufficiently accurate and no further modification will be needed. (If A has multiple

or near multiple eigenvalues it will be necessary to use block Lanczos in working on b_1 to retain the high likelihood that the initial guess x_0 for b_2 is sufficiently accurate).

It is important to point out that preconditioning schemes can be incorporated into the Lanczos formulation without difficulty. See [Parlett 1979] for a more detailed discussion of using (simple) Lanczos for solving linear equations.

Another area of interest for the Lanczos algorithm is in solving λ -matrix problems of the form

$$(5.2) \quad (\lambda^k A_k + \lambda^{k-1} A_{k-1} + \dots + \lambda_0) x = 0.$$

Such problems can always be rewritten as linear problems $A - \lambda M$ where A and M are $Kn \times Kn$ matrices but it would be nice to have some method of finding eigenvalues of (5.2) working with only $n \times n$ matrices.

A method given in [Scott 1979c] uses the Lanczos algorithm to solve linear problems $A - \lambda M$ for one eigenvalue at a time without factoring any matrix. Extensions of this idea are being investigated for λ -matrices of degree bigger than 1.

