Analysis of the Symmetric Lanczos Algorithm with Reorthogonalization Methods

Horst D. Simon*
Boeing Computer Services Company
565 Andover Park West
Tukwila, Washington 98188

Submitted by Richard A. Brualdi

ABSTRACT

We present an error analysis of the symmetric Lanczos algorithm in finite precision arithmetic. The loss of orthogonality among the computed Lanczos vectors is explained with the help of a recurrence formula. A backward error analysis shows that semiorthogonality among the Lanczos vectors is enough to guarantee the accuracy of the computed quantities up to machine precision. The results of this analysis are then extended to the more general case of the Lanczos algorithm with a semiorthogonalization strategy. Based on the recurrence formula, a new reorthogonalization method called partial reorthogonalization is introduced. We show that both partial reorthogonalization and selective orthogonalization as introduced by Parlett and Scott [15] are semiorthogonalization strategies. Finally we discuss the application of our results to the solution of linear systems of equations and to the eigenvalue problem.

1. INTRODUCTION

The Lanczos algorithm [6] is becoming accepted as a powerful tool for finding the eigenvalues of a matrix and for solving linear systems of equations. In recent years there has been considerable interest in the algorithm and its applications [1–4, 7–18]. Paige [8–10] and Grcar [4] have given detailed error analyses of the simple Lanczos algorithm. Here we will discuss a backward error analysis of the Lanczos algorithm with various reorthogonalization methods. In order to present a unifying treatment of methods like full reorthogonalization, selective orthogonalization [15], periodic reorthogonalization [4], and partial reorthogonalization, we introduce the new concept of a semiorthogonalization strategy. The Lanczos algorithm will be considered as a

*The author gratefully acknowledges support through ONR contract N00014-69-0200-1017.

52 Vanderbilt Ave., New York, NY 10017 0024-3795/84/$3.00

method for tridiagonalizing a symmetric $n \times n$ matrix A. Our main result will be that the Lanczos algorithm with a semiorthogonalization strategy computes after j steps a tridiagonal matrix T_j which is, up to roundoff, the orthogonal projection of A onto span(Q_j).

In order to obtain this result, we first introduce the Lanczos algorithm in exact arithmetic in Section 2. In Section 3 we set up a mathematical model of the Lanczos algorithm in the presence of roundoff errors. This model will then serve as the basis of our analysis. In Section 4 we then derive a recurrence formula, which is originally due to Paige [8]. This formula is of central importance for this analysis, because it explains the loss of orthogonality among the Lanczos vectors.

After some preliminary lemmas in Section 5, we will discuss in Section 6 the simple Lanczos algorithm and glean some insight into why the semiorthogonality of the Lanczos vectors is crucial for the accuracy of the computed tridiagonal matrix T_j. This insight leads to the definition of a semiorthogonalization strategy in Section 7. The main theorem concerning the accuracy of T_j follows then directly from the results of Section 6. Based on the recurrence from Section 4, the new method of partial reorthogonalization is introduced. We then show that the various reorthogonalization methods mentioned above are indeed semiorthogonalization strategies. In the case of selective orthogonalization (SO) this involves a new proof of the fact that SO maintains semiorthogonality.

2. THE LANCZOS ALGORITHM IN EXACT ARITHMETIC

The simple Lanczos algorithm for a symmetric $n \times n$ matrix A computes a sequence of Lanczos vectors q_j and scalars α_j, β_j as follows:

1: *choose a starting vector* r_1, $r_1 \neq 0$, set $q_0 \equiv 0$, $\beta_1 = \|r_1\|$
2: for $j = 1, 2, \ldots$ *do*

$$q_j = r_j / \beta_j$$
$$u_j = Aq_j - \beta_j q_{j-1}$$
$$\alpha_j = u_j^* q_j$$
$$r_{j+1} = u_j - \alpha_j q_j$$
$$\beta_{j+1} = \|r_{j+1}\|$$

One pass through step 2 is a Lanczos step. The equation for one Lanczos step can be written as

$$\beta_{j+1} q_{j+1} = Aq_j - \alpha_j q_j - \beta_j q_{j-1}. \tag{2.1}$$

The first j equations (2.1) can be condensed in matrix form as

$$AQ_j - Q_jT_j = \beta_{j+1}q_{j+1}e_j^*, \tag{2.2}$$

where $Q_j = (q_1,\dots,q_j)$, $e_j^* = (0,0,\dots,1)$, and

$$T_j \equiv \begin{bmatrix} \alpha_1 & \beta_2 & 0 & \cdots & 0 \\ \beta_2 & \alpha_2 & \beta_3 & & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & & \beta_{j-1} & \alpha_{j-1} & \beta_j \\ 0 & \cdots & 0 & \beta_j & \alpha_j \end{bmatrix}.$$

The vectors q_j are orthonormal, i.e.

$$Q_j^*Q_j = I_j, \tag{2.3}$$

where I_j is the $j \times j$ identity matrix. Paige [9] has shown that the above implementation is the best among several alternatives.

The algorithm terminates if $\beta_{j+1} = 0$, and this will happen for some $j \leqslant n$ in exact arithmetic. The eigenvalues of the tridiagonal matrix T_j are called the Ritz values. If s_i, $i = 1,\dots,j$, are the eigenvectors of T_j, the vectors $y_i = Q_js_i$ are called the Ritz vectors. Ritz values and vectors are the Rayleigh-Ritz approximations to the eigenvalues and vectors of A from span(Q_j), the subspace spanned by the vectors $q_1,\dots,q_j$. More details on the Lanczos algorithm for computing eigenvalues can be found in [13].

The algorithm can also be used for solving linear systems of equations $Ax = b$. Then b is chosen as starting vector, and at the jth step one approximate solution is given by $x_j \equiv Q_jT_j^{-1}\beta_1e_1$. This is explained in detail in [14] and [18]. If A is also positive definite, the so defined x_j is identical to the jth iterate produced by the conjugate gradient algorithm. This connection is explained in [12], where it is also used to derive a different, yet closely related algorithm for the solution of symmetric indefinite systems.

3. A MATHEMATICAL MODEL OF THE LANCZOS ALGORITHM IN THE PRESENCE OF ROUNDOFF

Most error analyses start out by making some assumptions on the roundoff errors which will occur when elementary operations like addition are carried out in floating-point computation with relative precision ε. Based on these assumptions, upper bounds on the errors in vector inner products, matrix-vector multiplications, etc., are derived or the reader is referred to Wilkinson [20]. After providing these tools, then, finally the object of analysis itself is approached. Lengthy and complicated derivations finally yield error bounds which are rigorous.

We try here a different approach. In this section we are going to state a set of assumptions on the behavior of the rounding errors occurring in the Lanczos algorithm in finite precision. These assumptions constitute a model for the actual computation. This model includes certain features (the essential ones in my opinion), but discards others (the irrelevant ones). On this model we build a rigorous analysis. By dealing only with the important points, we will be able to present an analysis which brings clarity and is much easier to follow than a completely rigorous analysis. The simplification of the results and their relation to the observed behavior of the Lanczos algorithm will eventually justify our choice of model. However, it is clear that we can only be *sure* of this type of analysis after going through a fully rigorous analysis. For the basic Lanczos algorithm this has been done already by Paige [10].

The presentation of the Lanczos algorithm in Section 2 assumed an ideal mathematical setting. However, Lanczos himself [6] was already aware of the strong influence which roundoff had on the algorithm. The computed quantities can differ greatly from their theoretical counterparts.

In the context of finite precision arithmetic, the basic three term recurrence between the Lanczos vectors at the jth step can be written

$$\beta_{j+1}q_{j+1} = Aq_j - \alpha_j q_j - \beta_j q_{j-1} - f_j, \tag{3.1}$$

where the n-vector f_j accounts for the local roundoff errors at the jth step, and the α_j, β_j, q_j denote (as they will from now on) the corresponding computed quantities. As in (2.2), the first j equations (3.1) can be written in matrix form

$$AQ_j - Q_jT_j = \beta_{j+1}q_{j+1}e_j^* + F_j, \tag{3.2}$$

where the $n \times j$ matrix F_j is given by $F_j = (f_1, f_2, \ldots, f_j)$. A bound on $\|F_j\|$ depends on the specific implementation of the Lanczos algorithm, and on the machine roundoff unit ε. Parlett [13, p. 268] has observed no exception to the assertion that

$$\|F_j\| \leqslant \varepsilon \|A\|, \tag{3.3}$$

where $\|\cdot\|$ denotes the 2-norm, as it will from now on. The actual bound is only a small multiple of this [10]. The formula (3.3) is also supported by a study of $\|f_j\|$, reported in [18]. In the following analysis we assume that (3.3) holds, i.e. that the local errors are at roundoff level.

Let the $j \times j$ matrix $W_j = (\omega_{ik})$ be defined by

$$W_j = Q_j^* Q_j. \tag{3.4}$$

Ideally the Lanczos vectors should be orthogonal, i.e. $W_j = I_j$. But this relation is completely destroyed by the effects of finite precision arithmetic. No implementation of the Lanczos algorithm as described in Section 2 yields a small *a priori* bound on $\|W_j - I_j\|$; in fact the elements of $W_j - I_j$ can become as big as 1. The computed Lanczos vectors not only lose orthogonality, but become linearly dependent to working precision. The growth of the elements of $W_j - I_j$ will be referred to as the *loss of orthogonality* among the Lanczos vectors. Let the first j Lanczos vectors $q_1, q_2, \ldots, q_j$ satisfy

$$|q_i^* q_k| \leqslant \omega_j \tag{3.5}$$

for $i = 1, \ldots, j$, $k = 1, \ldots, j$, $k \neq i$, and $0 \leqslant \omega_j \leqslant 1$. The smallest ω_j for which (3.5) holds will be called the *level of orthogonality* among the Lanczos vectors. If $\omega_j = \sqrt{\varepsilon}$, then the Lanczos vectors will be called *semiorthogonal.* Clearly, if $\omega_j = 0$ the vectors are orthonormal. The example in Figure 1 illustrates the typical loss of orthogonality as the Lanczos algorithm proceeds. The level of orthogonality among the Lanczos vectors is plotted on a logarithmic scale for the first 55 steps of a run of the algorithm with a matrix of order $n = 961$, resulting from an approximation to Poisson's equation on

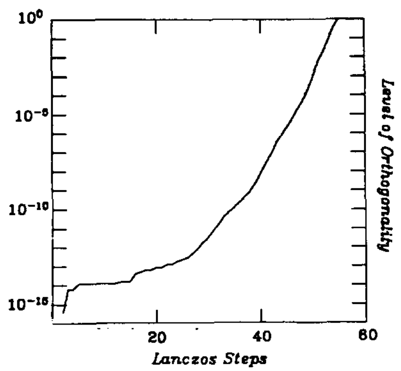

FIG. 1. The loss of orthogonality among the Lanczos vectors.

the unit square with 31×31 grid points. The starting vector is $q_1 = (1,1,\dots,1)^*/\sqrt{961}$.

Some more assumptions are necessary in order to simplify the technical details of the analysis of the loss of orthogonality. It will be assumed that the Lanczos vectors are exactly normalized, i.e. that

$$q_k^* q_k = 1 \qquad \text{for} \quad k = 1,\dots,j, \tag{3.6}$$

and that locally the level of orthogonality among the q_i's satisfies the following relation:

$$\beta_{k+1} q_{k+1}^* q_k = O(\varepsilon)\|A\| \qquad \text{for} \quad k = 1,\dots,j. \tag{3.7}$$

Paige [10, p. 344] shows that the actual bound on the local loss of orthogonality is $\beta_{k+1}|q_{k+1}^* q_k| \leqslant 2(n+4)\|A\|\varepsilon$. For our purposes (3.7) is sufficient, since as long as $2(n+4)\|A\|\varepsilon \ll \sqrt{\varepsilon}$ the actual size of $\beta_{k+1} q_{k+1}^* q_k$ is not important for the following analysis. Similarly, the later analysis will show that roundoff errors in the normalization of the q_j's are inconsequential for the loss of orthogonality.

Finally let us assume that

$$\text{no } \beta_{j+1} \text{ ever becomes negligible.} \tag{3.8}$$

This is almost always true in practice, and the rare cases where a β_{j+1} does become small are actually the lucky ones, since then the algorithm should be terminated, having found an invariant subspace.

(3.1)–(3.8) constitute the mathematical model of the Lanczos algorithm which we are going to investigate further. The goal of the rest of this paper is to explain the mechanism which causes the loss of orthogonality in the Lanczos algorithm, and then to analyze the algorithm in the light of this understanding. The results will help to clarify the role of the $\sqrt{\varepsilon}$ threshold, which appears both in Parlett and Scott's [15] and Grcar's [4] work. The insight will also lead to a new orthogonalization procedure, which will be discussed in Section 7.

4. THE LOSS OF ORTHOGONALITY

The loss of orthogonality and the associated "instability" of the Lanczos algorithm in the past has sometimes been simply credited to an accumulation of roundoff and cancellation errors. Paige [8–11] was the first to provide an understanding of what exactly is happening when orthogonality is lost. He is the first one to regard the loss of orthogonality as an amplification of the local errors which can be explained through recurrence formulas. In this section we will follow Paige's main ideas and present some of his results. Related ideas have been discussed by Grcar [4], and by Takahasi and Natori [19].

The loss of orthogonality can also be understood if one follows a simple geometrical argument. Suppose the algorithm was carried out for j steps without any error and the vectors $q_1,\ldots,q_j$ were perfectly orthogonal. Now at the $j+1$st step a small error occurs, such that q_{j+1} is no longer orthogonal to the previous Lanczos vectors. From then on the algorithm is again continued without error. Even if q_{j+2} were constructed perfectly orthogonal to q_{j+1} and q_j, it would no longer be orthogonal to the vectors $q_1,\ldots,q_{j-1}$, because q_{j+1} was not orthogonal to them. The same is true for all consecutive Lanczos vectors. *The error once introduced is propagated to future Lanczos vectors.*

Now if two consecutive Lanczos vectors q_{k-1} and q_k deviate slightly from their correct direction, then of course the vector Aq_k will be also slightly wrong. This by itself would not be so bad, but this already slightly wrong Aq_k will now additionally be orthogonalized against already deviating vectors, and thus the resulting q_{k+1} will differ even more from its true direction. Once introduced, the error is thus not only propagated, but depending on the geometry of the q_j's, it may be additionally amplified.

The loss of orthogonality therefore can be viewed as the result of an amplification of each local error after its introduction into the computation. The following theorem is the arithmetic equivalent of the geometric considerations above. It quantifies precisely how the local error is propagated in the algorithm, and how the level of orthogonality rises due to the mechanisms of the algorithm.

THEOREM 1. *The elements ω_{ik} of the $j\times j$ matrix $W_j = Q_j^*Q_j$ satisfy the following recurrence*:

$$\begin{aligned}
\omega_{kk} &= 1 \qquad \textit{for} \quad k=1,\ldots,j,\\
\omega_{k\,k-1} &= \varepsilon_k \qquad \textit{for} \quad k=2,\ldots,j,\\
\beta_{j+1}\omega_{j+1\,k} &= \beta_{k+1}\omega_{j\,k+1} + (\alpha_k-\alpha_j)\omega_{jk} + \beta_k\omega_{j\,k-1} - \beta_j\omega_{j-1\,k}\\
&\quad + q_j^*f_k - q_k^*f_j \qquad \textit{for} \quad 1\leqslant k<j,\\
\omega_{j\,k+1} &= \omega_{k+1\,j}.
\end{aligned} \tag{4.1}$$

Here $\omega_{k0}\equiv 0$ and $\varepsilon_k = q_k^*q_{k-1}$.

Proof. Write (3.1) for j and for k:

$$\beta_{j+1}q_{j+1} = Aq_j - \alpha_j q_j - \beta_j q_{j-1} - f_j, \tag{4.2}$$

$$\beta_{k+1}q_{k+1} = Aq_k - \alpha_k q_k - \beta_k q_{k-1} - f_k. \tag{4.3}$$

Forming $q_k^*(4.2) - q_j^*(4.3)$ and simplifying yields the result ■

Theorem 1 and the following formulas (4.4) and (4.5) first appeared in [8, (8.23)] (see also [10, p. 346]).

Note that (4.1) can be also obtained in vector form. First premultiply (3.1) by Q_j^*:

$$\begin{aligned}\beta_{j+1}Q_j^*q_{j+1} &= Q_j^*Aq_j - \alpha_j Q_j^*q_j - \beta_j Q_j^*q_{j-1} - Q_j^*f_j \\ &= T_jQ_j^*q_j - \alpha_j Q_j^*q_j - \beta_j Q_j^*q_{j-1} + \beta_{j+1}e_j q_{j+1}^*q_j + F_j^*q_j - Q_j^*f_j.\end{aligned} \tag{4.4}$$

Let R_j be the strictly upper triangular part of W_j (i.e., R_j is zero on and below the diagonal), and let $\overline{w}_1, \overline{w}_2, \dots, \overline{w}_j$ be the columns of R_j. Let $w_{j+1} \equiv Q_j^*q_{j+1}$. Then from (4.4) it follows that

$$\beta_{j+1}w_{j+1} = T_j\overline{w}_j - \alpha_j\overline{w}_j - \beta_j\overline{w}_{j-1} + g_j + e_j\left(\beta_{j+1}q_{j+1}^*q_j - \beta_j q_j^*q_{j-1}\right), \tag{4.5}$$

where $g_j \equiv F_j^*q_j - Q_j^*f_j$. [Equation (4.5) could have been obtained directly from (4.1), by writing (4.1) in vector form for $k = 1, \dots, j$.] From (4.5) we can obtain an estimate for the loss of orthogonality:

$$\beta_{j+1}\|w_{j+1}\| \leqslant \left(\|T_j\| + |\alpha_j|\right)\|\overline{w}_j\| + \beta_j\|\overline{w}_{j-1}\| + O(\varepsilon\|A\|) \tag{4.6}$$

$$\leqslant 2\|A\|\max\left\{\|\overline{w}_j\|, \|\overline{w}_{j-1}\|\right\} + O(\varepsilon\|A\|). \tag{4.7}$$

Therefore the level of orthogonality grows at most by a factor of $2\|A\|/\beta_{j+1}$ at each step. A small β_{j+1} will cause a great loss of orthogonality. A Lanczos run which has rapidly decreasing or greatly varying β_i's will therefore suffer from a larger loss of orthogonality than a run with nearly constant β_i's. In order to obtain (4.7) we also used $\|T_j\| \leqslant \|A\| + \|F_j\|$, which was shown by Paige [10].

The recurrence formula (4.1) shows that the loss of orthogonality is merely initiated by the local error f_j. The growth of the elements of W_j depends mainly on the α_i's and β_i's. It is therefore definitely *not* due to an accumulation of roundoff or cancellation errors. Once the ω_{jk} have grown to a certain level, the local error terms $q_i^*f_k - q_k^*f_j$, which are $O(\varepsilon)$, contribute negligibly to the growth of the loss of orthogonality.

The loss of orthogonality is hence a phenomenon which is started by the f_j, but from then on its growth is determined by the α_j's and β_j's, i.e. by the eigenvalue distribution of A and by the starting vector q_1.

The way in which orthogonality is lost can be understood better if Equation (4.5) is analyzed further. Let the exact spectral factorization of T_j be given by $T_jS_j = S_j\Theta_j$, where $\Theta_j = \mathrm{diag}(\vartheta_1^{(j)},\dots,\vartheta_j^{(j)})$, $S_j = (s_1^{(j)},\dots,s_j^{(j)})$, and $S_j^* = S^{-1}$; and define the vectors $y_i \equiv Q_js_i$ for $i = 1,\dots,j$. Note that, contrary to Section 2, we consider here the *exact* eigendecomposition of the *computed* T_j. Therefore the $\vartheta_i^{(j)}$ and $y_i^{(j)}$ should be referred to as the computed Ritz values and vectors. They may differ from their ideal counterparts as defined in Section 2. In particular, there is no reason to expect the computed Ritz vectors to be orthonormal. Nevertheless we will refer to them here simply as Ritz values and vectors, since no confusion with the ideal quantities is likely. Furthermore let $\sigma_{ji} \equiv e_j^*s_i^{(j)}$, the bottom element of the eigenvector $s_i^{(j)}$, and let the eigenvectors $s_i^{(j)}$ be normalized to make σ_{ji} positive.

With all this notation the remaining analysis becomes quite simple. Considering the first j steps of the algorithm, the corresponding instances of (4.5) can be combined in matrix form as (cf. [8, (8.26)])

$$\beta_{j+1}w_{j+1}e_j^* = T_jR_j - R_jT_j + G_j, \tag{4.8}$$

where G_j is the strictly upper triangular part of $F_j^*Q_j - Q_j^*F_j$. Forming $s_i^*(4.8)s_i$, one obtains

$$\begin{aligned} \beta_{j+1}s_i^*w_{j+1}\sigma_{ji} &= \vartheta_i s_i^*R_js_i - s_i^*R_js_i\vartheta_i + s_i^*G_js_i, \\ \beta_{j+1}y_i^*q_{j+1}\sigma_{ji} &= s_i^*G_js_i \equiv \gamma_{ii}. \end{aligned} \tag{4.9}$$

This is precisely Paige's theorem:

THEOREM 2 (Paige). *Let S_j, Θ_j, G_j, σ_{ji}, and γ_{ii} be defined as above. Then the vectors $y_i = Q_js_i$ for $i = 1,\dots j$ satisfy*

$$y_i^*q_{j+1} = \frac{\gamma_{ii}}{\beta_{j+1}\sigma_{ji}}. \tag{4.10}$$

Equation (4.10) describes the way in which the orthogonality is lost. We have assumed in (3.8) that no β_{j+1} becomes negligible. If we also assume that γ_{ii} is tiny like $\varepsilon\|A\|$, the only way that $y_i^* q_{j+1}$ can become large is by σ_{ji} becoming small. As Paige pointed out

$$\|Ay_i - y_i\vartheta_i\| = \|AQ_j s_i - Q_j s_i \vartheta_i\| = \|\beta_{j+1} q_{j+1} e_j^* s_i + F_j s_i\|$$

$$\leqslant \beta_{j+1}\sigma_{ji} + \varepsilon\|A\|$$

and so a small σ_{ji} indicates that (ϑ_i, y_i) is an approximate eigenpair of the matrix A. Paige's theorem therefore can be stated as follows: loss of orthogonality implies convergence of a Ritz pair to an eigenpair [8]. It is however not trivial to prove the convergence of the Ritz pair rigorously, since $\|y_i\|$ in (4.10) may be small. This has been done in [11, Theorem 3.1].

5. LEMMAS

In this section we will state and prove several lemmas, which will be needed in the later analysis of the Lanczos algorithm. These lemmas are mainly concerned with certain properties of the matrix $W_j = Q_j^* Q_j$ and related matrices, and are therefore completely independent of any properties of the Lanczos algorithm.

Let the $j \times j$ matrix W be given by $W = (\omega_{ik})$, with $\omega_{ii} = 1$ for $i = 1, \ldots, j$, and $-1 \leqslant \omega_{ik} \leqslant 1$ for $i \neq k$, $i, k = 1, \ldots, j$; and let $W = W^*$. Then define

$$\omega = \max_{\substack{1 \leqslant i, k \leqslant j \\ i \neq k}} |\omega_{ik}|.$$

Denote by $\lambda_1(W)$ the smallest and by $\lambda_j(W)$ the largest eigenvalue of the matrix W.

LEMMA 1.

(a) $\lambda_1(W) \geqslant 1 - (j-1)\omega$.
(b) $\lambda_j(W) \leqslant 1 + (j-1)\omega$.
(c) $\|W\| \leqslant 1 + (j-1)\omega$.
(d) *If* $\omega < 1/(j-1)$, *then* W^{-1} *exists and* $\|W^{-1}\| \leqslant 1/[1-(j-1)\omega]$.

Proof. Application of Gershgorin's theorem ■

LEMMA 2. *Let $j \geqslant 2$ and let*

$$\omega \leqslant \frac{1}{2}\frac{1}{j-1}.$$

Then $LL^* = W$, the Choleski factorization of W, exists and

$$\|L\| = \|L^*\| \leqslant \sqrt{2},$$

$$\|L^{-1}\| = \|L^{-*}\| \leqslant \sqrt{2}.$$

Proof.

$$\|L\| = \sqrt{\lambda_j(LL^*)} = \sqrt{\lambda_j(W)} \leqslant [1+(j-1)\omega]^{1/2} \leqslant \sqrt{2}$$

$$\|L^{-*}\| = \sqrt{\lambda_j(L^{-*}L^{-1})} = \sqrt{[\lambda_1(W)]^{-1}} \leqslant [1-(j-1)\omega]^{-1/2} \leqslant \sqrt{2}. \quad \blacksquare$$

LEMMA 3. *If*

$$j \geqslant 2 \quad \textit{and} \quad \omega \leqslant \frac{1}{2}\frac{1}{j-1}$$

so the Choleski factorization $W = LL^*$ exists, then the elements η_{ik} of the Choleski factor L satisfy

$$\eta_{ik} = \omega_{ik} + O(j\omega^2) \qquad \textit{for} \quad 1 \leqslant k < i \leqslant j, \tag{5.1}$$

$$\eta_{ii} = 1 + O(j\omega^2) \qquad \textit{for} \quad 1 \leqslant i \leqslant j. \tag{5.2}$$

Proof. We prove (5.1) and (5.2) by induction. For $j = 1$ (5.2) obviously holds, and for $j = 2$ we have

$$\begin{bmatrix} 1 & \omega_{21} \\ \omega_{21} & 1 \end{bmatrix} = \begin{bmatrix} \eta_{11} & 0 \\ \eta_{21} & \eta_{22} \end{bmatrix}\begin{bmatrix} \eta_{11} & \eta_{21} \\ 0 & \eta_{22} \end{bmatrix}.$$

Hence $\eta_{11} = 1$, $\eta_{21} = \omega_{21}$, $\eta_{22} = \sqrt{1-\omega_{21}^2} = 1 + O(\omega_{21}^2)$, and both (5.1) and (5.2) hold.

For general j we can partition the Choleski factor as follows:

$$W = \left[\begin{array}{c|c} \bar{L}_{j-1} & 0 \\ \hline l_j^* & \eta_{jj} \end{array}\right]\left[\begin{array}{c|c} \bar{L}_{j-1}^* & l_j \\ \hline 0^* & \eta_{jj} \end{array}\right] = \left[\begin{array}{c|c} \bar{L}_{j-1}\bar{L}_{j-1}^* & \bar{L}_{j-1}l_j \\ \hline l_j^*\bar{L}_{j-1}^* & l_j^*l_j + \eta_{jj}^2 \end{array}\right].$$

By induction the elements of the $(j-1)\times(j-1)$ lower triangular matrix $\overline{L}_{j-1}$ satisfy (5.1) and (5.2). What is left to be shown is that the elements of l_j^* satisfy (5.1), and that η_{jj} satisfies (5.2). Let $l_j^* = (\eta_{j1}, \eta_{j2}, \dots, \eta_{j\,j-1})$. Then $\omega_{jk} = \sum_{p=1}^{k} \eta_{jp}\eta_{kp}$, where η_{kp} are the corresponding elements of the matrix $\overline{L}_{j-1}$. Equation (5.1) can now be shown by induction over k. For $k = 1$ we obtain

$$\omega_{j1} = \eta_{j1}\eta_{11} = \eta_{j1}\left[1 + O(j\omega^2)\right];$$

hence $\eta_{j1} = \omega_{j1}[1 + O(j\omega^2)] = \omega_{j1} + O(j\omega^2)$, and (5.1) holds. For general k we obtain

$$\begin{aligned}\omega_{jk} &= \sum_{p=1}^{k-1} \eta_{jp}\eta_{kp} + \eta_{jk}\eta_{kk}\\ &= \sum_{p=1}^{k-1} \left[\omega_{jp} + O(j\omega^2)\right]\left[\omega_{kp} + O(j\omega^2)\right] + \eta_{jk}\left[1 + O(j\omega^2)\right].\end{aligned}$$

Hence

$$\omega_{jk} = \eta_{jk}\left[1 + O(j\omega^2)\right] + O(j\omega^2) + O(j^2\omega^3) + O(j^3\omega^4),$$

and (5.1) holds for all elements in l_j^*.

Finally

$$\begin{aligned}\eta_{jj}^2 &= 1 - \sum_{p=1}^{j-1} \eta_{jk}^2 = 1 - \sum_{p=1}^{j-1} \left[\omega_{jk} + O(j\omega^2)\right]^2\\ &= 1 + O(j\omega^2) + O(j^2\omega^3) + O(j^3\omega^4),\end{aligned}$$

and it follows that (5.2) also holds for η_{jj}. ■

6. ANALYSIS OF THE SIMPLE LANCZOS ALGORITHM

There are two quantities at hand which could be the object of an error analysis of the Lanczos algorithm: the Lanczos vectors Q_j and the matrix T_j formed by the α_i and β_i. It is important to note the following fact at the outset of any further analysis: If a matrix $\bar{A}$ is close to A, this does not imply that the sequence of Lanczos vectors computed from $\bar{A}$ is in any way close to the sequence of Lanczos vectors computed from A. This has already been remarked in [11, pp. 252–253]. The following example shows that Krylov subspaces can be very sensitive to small perturbations in the matrix.

EXAMPLE. Consider

$$A = \begin{bmatrix} 1 & \eta & 0 \\ \eta & 2 & 3 \\ 0 & 3 & 4 \end{bmatrix}, \qquad q_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},$$

where η is a real parameter. Then the first Lanczos step yields

$$Aq_1 = \begin{bmatrix} 1 \\ \eta \\ 0 \end{bmatrix}, \qquad \alpha_1 = 1, \qquad q_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

However, for the same q_1 and

$$\bar{A} = \begin{bmatrix} 1 & 0 & \eta \\ 0 & 2 & 3 \\ \eta & 3 & 4 \end{bmatrix},$$

one obtains that

$$\bar{A}q_1 = \begin{bmatrix} 1 \\ 0 \\ \eta \end{bmatrix}, \qquad \bar{\alpha}_1 = q_1^* \bar{A} q_1 = 1, \qquad \bar{q}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

Hence $\bar{q}_2^* q_2 = 0$ independently of η, and even a small perturbation in the matrix may therefore result in totally different, i.e. orthogonal, Lanczos vectors.

This example shows the limitations of a forward error analysis of the Lanczos algorithm. Since the Lanczos vectors may differ considerably even when there is only a small perturbation, any error analysis which attempts to compare the "ideal" (i.e. exact arithmetic) Lanczos vectors with the computed ones, has to run into considerable difficulties. Grcar [4] avoids this difficulty in his forward analysis, by making the strong assumption *a priori* that the error between "ideal" and computed Lanczos vectors is small.

The example above indicates that the situation for the Lanczos algorithm is comparable to other methods for tridiagonalizing a symmetric matrix, like Givens's or Householder's method. The computed intermediate quantities may differ from their ideal counterparts, but this is not the important issue if one performs a backward analysis. For these methods it can be shown that the computed tridiagonal matrix is exactly similar to a perturbation of the original matrix A, where the relative size of the perturbation is a modest multiple of the roundoff unit and $\|A\|$. We feel that a similar approach is also appropriate for the Lanczos algorithm. Our main attention will therefore be directed towards the matrix T_j.

Let us first remark that the loss of linear independence of the Lanczos vectors does not concern us here. From Lemma 1 we can conclude that as long as $\omega < 1/(j-1)$ the Lanczos vectors are linearly independent. This means in a typical situation with $\varepsilon = 10^{-15}$, $j = 100$, that ω can be as large as 10^{-2}, and the Lanczos vectors will be still linearly independent. Hence the level of orthogonality can grow by a factor of 10^{13} without affecting the linear independence of the q_k. The loss of linear independence is therefore a consequence of the loss of orthogonality, and will not concern us for the moment.

The following theorem complements [11, Theorem 4.2] and shows how the loss of orthogonality affects the matrix T_j.

THEOREM 3. *If ω, the level of orthogonality among the Lanczos vectors $q_1, q_2, q_3, \dots, q_{j+1}$ satisfies*

$$\omega \leqslant \frac{1}{2}\frac{1}{j-1},$$

then the computed tridiagonal matrix T_j is similar to a matrix $\bar{T}_j$, which is a perturbation of the orthogonal projection of A onto span(Q_j). *If A_P denotes this projection, then*

$$\|A_p - \bar{T}_j\| \leqslant \sqrt{j}\,\omega\beta_{j+1} + \sqrt{2}\,\varepsilon\|A\| + O(\varepsilon^2)\|A\| + O(j^{3/2}\omega^2)\beta_{j+1}. \quad (6.1)$$

Proof. Since $q_1, q_2, \dots, q_{j+1}$ are linearly independent, the QR factorization of Q_j has the form $Q_j = N_j L_j^*$, where N_j is a $n \times j$ matrix with orthonormal columns, and L_j^* is a $j \times j$ upper triangular matrix with positive diagonal elements. Moreover N_j and L_j^* are uniquely determined. Since $W_j = Q_j^* Q_j = L_j N_j^* N_j L_j^* = L_j L_j^*$, L_j is also the Choleski factor of W_j.

Similarly $Q_{j+1} = N_{j+1}L^*_{j+1}$, where $N_{j+1} = [N_j \mid n_{j+1}]$ and

$$L^*_{j+1} = \begin{bmatrix} L^*_j & \bar{l}_{j+1} \\ 0^* & \delta_{j+1} \end{bmatrix} = \left[\begin{array}{c|c} L^*_j & l_{j+1} \\ 0^* & \end{array}\right].$$

Now the basic Lanczos equation can be rearranged as follows:

$$\begin{aligned} AQ_j - Q_jT_j &= \beta_{j+1}q_{j+1}e^*_j + F_j, \\ Q^*_jQ_jT_j &= Q^*_jAQ_j - \beta_{j+1}Q^*_jq_{j+1}e^*_j - Q^*_jF_j, \\ L_jL^*_jT_j &= L_jN_j^*AN_jL^*_j - \beta_{j+1}L_jN_j^*N_{j+1}l_{j+1}e^*_j - L_jN_j^*F_j, \\ L^*_jT_jL_j^{-*} &= N_j^*AN_j - \beta_{j+1}N_j^*N_{j+1}l_{j+1}e^*_jL_j^{-*} - N_j^*F_jL_j^{-*}. \end{aligned} \tag{6.2}$$

Let $\overline{T}_j \equiv L^*_jT_jL_j^{-*}$; then $\overline{T}_j$ is similar to T_j. $\overline{T}_j$ can now be considered as a perturbation of $A_P \equiv N_j^*AN_j$, the orthogonal projection of A onto span(Q_j). The norm of this perturbation can be bounded by

$$\|A_P - \overline{T}_j\| \leqslant \beta_{j+1}\|N_j^*N_{j+1}l_{j+1}e^*_jL_j^{-*}\| + \|N_j^*F_jL_j^{-*}\|$$

Now $N_j^*N_{j+1} = \tilde{I}_j \equiv [I_j|0]$ and $L_j^{-1}e_j = \eta_{jj}^{-1}e_j$, where η_{jj} is the bottom diagonal element of the Choleski factor L_j. As $\|N_j^*\| = 1$ and $\|L_j^{-*}\| \leqslant \sqrt{2}$ by Lemma 2, we obtain

$$\begin{aligned} \|A_P - \overline{T}_j\| &\leqslant \beta_{j+1}\eta_{jj}^{-1}\|\bar{l}_{j+1}\| + \sqrt{2}\,\|F_j\| \\ &\leqslant \beta_{j+1}\eta_{jj}^{-1}\|\bar{l}_{j+1}\| + \sqrt{2}\,\varepsilon\|A\|. \end{aligned}$$

Now we obtain from Lemma 3 that $\eta_{jj} = 1 + O(j\omega^2)$, whence $\eta_{jj}^{-1} = 1 + O(j\omega^2)$, and that $\|\bar{l}_{j+1}\|^2 = \sum_{k=1}^{j}[\omega_{kj} + O(j\omega^2)]^2 \leqslant j\omega^2 + O(j^2\omega^3)$, whence $\|\bar{l}_{j+1}\| \leqslant \sqrt{j}\,\omega + O(j^{3/2}\omega^2)$. With these results (6.1) follows. ■

Theorem 3 says that the norm of the perturbation is proportional to the level of orthogonality among the Lanczos vectors, as long as the loss of linear independence among the Lanczos vectors is not imminent. The conclusion of Theorem 3 no longer holds when the Lanczos vectors begin to lose their linear independence. Because if ω is of the size of $1/j$, the term $O(j^{3/2}\omega^2)\beta_{j+1}$ becomes comparable to the first term in (6.1). Also $\|L_j^{-1}\|$ can no longer be bounded if ω becomes too large.

Another important quantity to consider is the matrix W_j, which ideally should be the identity matrix I_j. Much of the material in the previous section was related to the question how much we can allow W_j to deviate from I_j and yet satisfy some useful properties enjoyed by I_j. Lemma 3 is the most important result in this direction. This lemma gives a first insight into why the particular bound $\omega \leqslant \sqrt{\varepsilon / j}$ is crucial for the Lanczos algorithm. In this case W_j has still, at least up to roundoff, the property of the identity matrix, that its lower triangular part is also its Choleski factor. This can be also seen by writing $W_j = (I - K_j^*)(I - K_j)$. Now if the bound on ω holds, $K_j^* K_j$ becomes negligible like ε. It seems that this property of W_j is enough to assure that the Lanczos algorithm in finite precision behaves up to roundoff like its ideal counterpart. This will be shown in the following

THEOREM 4. *If $j\omega^2 \leqslant \varepsilon$, i.e. if*

$$\omega \leqslant \sqrt{\frac{\varepsilon}{j}}, \tag{6.3}$$

then

$$N_j^* A N_j = T_j + V_j, \tag{6.4}$$

where the elements of V_j are of order $O(\varepsilon \| A \|)$, and N_j is the matrix as defined in Theorem 3.

Proof. Since $N_j^* A N_j$ is symmetric, it is sufficient to show by induction that the last column of $N_j^* A N_j$ and T_j differ only by a vector of order $O(\varepsilon \| A \|)$.

For $j = 1$ this is trivial since $n_1 = q_1$.

For general j transpose and rearrange Equation (6.2) from Theorem 3:

$$N_j^* A N_j = L_j^{-1} T_j L_j + \beta_{j+1} L_j^{-1} e_j \bar{l}_{j+1}^* + L_j^{-1} F_j^* N_j.$$

The last column of $N_j^* A N_j$ then is given by

$$N_j^* A N_j e_j = L_j^{-1} T_j L_j e_j + \beta_{j+1} L_j^{-1} e_j \bar{l}_{j+1}^* e_j + L_j^{-1} F_j^* N_j e_j. \tag{6.5}$$

The proof now rests upon the fact that the last two terms in (6.5) are small and that $L_j^{-1}T_jL_j$ is a lower Hessenberg matrix and thus only the last two elements of $L_j^{-1}T_jL_je_j$ are nonzero. Since L_j and L_j^{-1} are almost like the identity matrix, we should obtain that $L_j^{-1}T_jL_je_j \approx \alpha_je_j + \beta_je_{j-1}$. In the remaining part of the proof we are going to show that these statements are true.

Let the elements in the bottom right corner of L_j be denoted by

$$L_j = \begin{bmatrix} \ddots & \vdots & \vdots \\ \cdots & \eta_{j-1\,j-1} & 0 \\ \cdots & \eta_{j\,j-1} & \eta_{jj} \end{bmatrix}.$$

Then the corresponding elements of L_j^{-1} are

$$L_j^{-1} = \begin{bmatrix} \ddots & \vdots & \vdots \\ \cdots & \eta_{j-1\,j-1}^{-1} & 0 \\ \cdots & \eta_{j\,j-1}^{(-1)} & \eta_{jj}^{-1} \end{bmatrix},$$

where $\eta_{j\,j-1}^{(-1)} = -\eta_{j\,j-1}/\eta_{jj}\eta_{j-1\,j-1}$. With this notation we can evaluate the terms in (6.5) further:

$$\begin{aligned} L_j^{-1}T_jL_je_j &= L_j^{-1}T_j(\eta_{jj}e_j) = \eta_{jj}L_j^{-1}(\alpha_je_j + \beta_je_{j-1}) \\ &= \eta_{jj}\alpha_jL_j^{-1}e_j + \eta_{jj}\beta_jL_j^{-1}e_{j-1} \\ &= \eta_{jj}\alpha_j\eta_{jj}^{-1}e_j + \eta_{jj}\beta_j\left(\eta_{j-1\,j-1}^{-1}e_{j-1} + \eta_{j\,j-1}^{(-1)}e_j\right). \end{aligned}$$

For the second term one obtains

$$\beta_{j+1}L_j^{-1}e_j\bar{l}_{j+1}^{*}e_j = \beta_{j+1}\eta_{jj}^{-1}\eta_{j+1\,j}e_j.$$

Here $\eta_{j\,j+1}$ is the jth element of the vector l_{j+1}.

Applying Lemma 3, we obtain now because of (6.3) that

$$\eta_{j-1\,j-1} = 1 + O(\varepsilon),$$

$$\eta_{jj} = 1 + O(\varepsilon),$$

and therefore also $\eta_{jj}^{-1} = 1 + O(\varepsilon)$, $\eta_{j-1\,j-1}^{-1} = 1 + O(\varepsilon)$. Also from Lemma 3 together with (6.3) it follows that

$$\eta_{j\,j-1} = \omega_{j\,j-1} + O(\varepsilon),$$

$$\eta_{j+1\,j} = \omega_{j+1\,j} + O(\varepsilon).$$

Then $\eta_{j\,j-1}^{(-1)} = -\eta_{j\,j-1}\eta_{jj}^{-1}\eta_{j-1\,j-1}^{-1} = -\omega_{j\,j-1} + O(\varepsilon)$. And finally, since $\|L_j^{-1}\| \leqslant \sqrt{2}$ by Lemma 2, it follows that $L_j^{-1}F_j^*n = O(\varepsilon)\|A\|$. Combining all these facts together, we have

$$N_j^*AN_je_j = \alpha_je_j + \beta_je_{j-1} + \left(\beta_{j+1}\omega_{j+1\,j} - \beta_j\omega_{j\,j-1}\right)e_j + O(\varepsilon)\|A\|,$$

where we have used that $\beta_j + \beta_{j+1} \leqslant \|T_j\| \leqslant (1+\varepsilon)\|A\|$.

Now we can use most advantageously Paige's result in the form of (3.7) and obtain

$$N_j^*AN_je_j = \alpha_je_j + \beta_je_{j-1} + O(\varepsilon)\|A\|,$$

which concludes the proof. ■

In the proof of Theorem 4 it was assumed that

$$j\omega^2 \leqslant \varepsilon \quad \text{implies} \quad \omega \leqslant \frac{1}{2}\frac{1}{j-1}.$$

This will be satisfied for all practical applications of the Lanczos algorithm. From now on we will assume that $j \ll \varepsilon^{-1}$.

Theorem 4 sets the stage for the next section. If it is possible to keep orthogonality at a level of $\sqrt{\varepsilon}$, then the Lanczos algorithm actually computes a matrix T_j which is, up to roundoff, the *orthogonal* projection of A onto $\operatorname{span}(Q_j)$, even though the Lanczos vectors themselves are no longer orthogonal to working precision.

7. SEMIORTHOGONALIZATION STRATEGIES

As soon as the level of orthogonality deteriorates so much that $|q_{j+1}^*q_k| > \omega_0 \equiv \sqrt{\varepsilon/j}$ for some $k<j$, the nice result $T_j \approx N_j^*AN_j$ does not hold any longer. The main goal in this section is to show that if by some means

semiorthogonality (i.e., $|q_i^* q_k| \leqslant \sqrt{\varepsilon}$ for $i \neq k$) can be maintained among the Lanczos vectors, the result of Theorem 4 will still hold for the modified algorithm.

Traditionally one was advised to perform the Lanczos algorithm with full reorthogonalization of the Lanczos vectors (Lanczos [6], Wilkinson [21]). This modification aims at maintaining orthogonality to working precision among the Lanczos vectors. Theorem 4 shows that not all this effort is necessary. More recently, selective orthogonalization (Parlett and Scott [15]) has been suggested as means of keeping semiorthogonality among the Lanczos vectors. Grcar [4] proposed periodic reorthogonalization in order to stabilize the Lanczos algorithm. The analysis for all these orthogonalization methods can be unified with the concept of a *semiorthogonalization strategy* for the Lanczos algorithm.

Suppose at the jth step of the Lanczos algorithm

$$\beta'_{j+1} q'_{j+1} = A q_j - \alpha_j q_j - \beta_j q_{j-1} - f_j'$$

and $|q'^*_{j+1} q_k| = |\omega'_{j+1\,k}| > \omega_0$ for some $k < j$. Then we choose $j-1$ real numbers $\xi_1, \ldots, \xi_{j-1}$, and form

$$\beta_{j+1} q_{j+1} = \beta'_{j+1} q'_{j+1} - \sum_{k=1}^{j-1} \xi_k q_k - f_j. \tag{7.1}$$

The algorithm will be continued with q_{j+1} instead of q'_{j+1}. This modification of the Lanczos algorithm will be called a *semiorthogonalization strategy* if the following conditions are satisfied:

(1) The numbers ξ_k, $k = 1, \ldots j-1$ are chosen such that

$$|q_l^* q_{j+1}| = |\omega_{j+1\,l}| \leqslant \omega_0, \tag{7.2}$$

where $\omega_0 \equiv \sqrt{\varepsilon / j}$.

(2) The computation of the ξ_k's and the formation of q_{j+1} causes at most roundoff errors of $O(\varepsilon \|A\|)$, i.e., we have

$$\beta_{j+1} q_{j+1} = A q_j - \alpha_j q_j - \beta_j q_{j-1} - \sum_{k=1}^{j-1} \xi_k q_k - f_j \tag{7.3}$$

and $\|f_j\| \leqslant \varepsilon \|A\|$.

We now can show that Paige's important result (3.7) concerning the local orthogonality of the Lanczos vectors also holds for the Lanczos algorithm with a semiorthogonalization strategy:

LEMMA 4. *Let β_{j+1} and q_{j+1} be computed by the Lanczos algorithm with a semiorthogonalization strategy* (7.2)–(7.3). *Then*

$$\beta_{j+1}q_{j+1}^*q_j = O(\varepsilon)\|A\|.$$

Proof. From (7.1) we obtain

$$\beta_{j+1}q_{j+1}^*q_j = \beta_{j+1}'q_{j+1}'^*q_j - \sum_{k=1}^{j-1} \xi_k q_k^* q_j - f_j^* q_j.$$

Using Paige's result [10, p. 344] [see also (3.7)] for the first term, and (7.2) and (7.3) for the second and third terms, it follows that

$$\beta_{j+1}|q_{j+1}^*q_j| \leqslant 2(n+4)\|A\|\varepsilon + \max_{1\leqslant k\leqslant j-1}|\xi_k|\sqrt{j\varepsilon} + \varepsilon\|A\|.$$

In order to complete the proof the lemma we therefore have to find a bound on $M \equiv \max_{1\leqslant k\leqslant j-1}|\xi_k|$ which is of $O(\sqrt{\varepsilon})$. Again from (7.1) we obtain, after some rearranging and with (3.6),

$$\xi_l = \beta_{j+1}'q_{j+1}'^*q_l - \beta_{j+1}q_{j+1}^*q_l - \sum_{\substack{k=1\\k\neq l}}^{j-1} (q_k^*q_l)\xi_k - f_j^*q_l$$

for $l = 1,2,\ldots,j-1$. Therefore,

$$|\xi_l| \leqslant 2\|A\|\sqrt{\varepsilon} + \beta_{j+1}\sqrt{\frac{\varepsilon}{j}} + j\sqrt{\frac{\varepsilon}{j}}M + O(\varepsilon)\|A\|.$$

Here we have used again the properties (7.2) and (7.3) of a semiorthogonalization strategy. The bound on $\beta_{j+1}'|q_{j+1}'^*q_l|$ is obtained in the same way as (4.7).

Now the right side does not depend on l any more, and we finally obtain by taking the maximum on the left side

$$(1-\sqrt{j\varepsilon})M \leqslant 2\|A\|\sqrt{\varepsilon} + \beta_{j+1}\sqrt{\frac{\varepsilon}{j}} + O(\varepsilon)\|A\|.$$

Since we assumed previously that $j \ll \varepsilon^{-1}$, the desired bound on M follows, and this concludes the proof of the Lemma. ■

All orthogonalization methods mentioned above can be summarized under the new concept of a semiorthogonalization strategy. The details, which are nontrivial in the case of selective orthogonalization, will be discussed later. Surprisingly, under very general assumptions we can prove the following

THEOREM 5. *Let T_j be the tridiagonal matrix computed by the Lanczos algorithm with a semiorthogonalization strategy. Then $N_j^*AN_j$, the orthogonal projection of A on* span(Q_j), *satisfies*

$$N_j^*AN_j = T_j + V_j, \tag{7.4}$$

where the elements of V_j are $O(\varepsilon\|A\|)$.

Proof. For a certain number of steps the algorithm will be just the ordinary Lanczos algorithm and Theorem 4 can be applied. Suppose now at step j for the first time the semiorthogonalization strategy comes into play:

$$\beta_{j+1}q_{j+1} = Aq_j - \alpha_j q_j - \beta_j q_{j-1} - \sum_{k=1}^{j-1} \xi_k q_k - f_j,$$

or in terms of matrices,

$$\begin{aligned} AQ_j &= Q_jT_j + \sum_{k=1}^{j-1} \xi_k q_k e_j^* + \beta_{j+1}q_{j+1}e_j^* + F_j \\ &= Q_j\left(T_j + \sum_{k=1}^{j-1} \xi_k e_k e_j^*\right) + \beta_{j+1}q_{j+1}e_j^* + F_j. \end{aligned} \tag{7.5}$$

Transposing and multiplying by Q_j, one obtains

$$Q_j^*AQ_j = \left(T_j + \sum_{k=1}^{j-1} \xi_k e_j e_k^*\right)Q_j^*Q_j + \beta_{j+1}e_j q_{j+1}^* Q_j + F_j^*Q_j.$$

As before, let $Q_j = N_jL_j^*$, $Q_j^*Q_j = L_jL_j^*$. Then

$$N_j^*AN_j = L_j^{-1}\left(T_j + \sum_{k=1}^{j-1} \xi_k e_j e_k^*\right)L_j + \beta_{j+1}L_j^{-1}e_j l_{j+1}^* + L_j^{-1}F_j^*N_j.$$

Now comes the important observation when we consider, as before, the jth column of the matrix $N_j^*AN_j$. *The perturbation term in T_j simply cancels out*, as $e_k^*L_je_j = \eta_{jj}e_k^*e_j = 0$ for $k = 1,\ldots,j-1$. Thus

$$N_j^*AN_je_j = L_j^{-1}T_j\eta_{jj}e_j + \beta_{j+1}L_j^{-1}e_j\bar{l}_{j+1}^*e_j + L_j^{-1}F_j^*n_j. \tag{7.6}$$

Now we can estimate the terms in (7.6) in the same way as in Theorem 4. We only have to use Lemma 4 instead of (3.7), and the result follows.

Suppose that for every step from step j onward an orthogonalization occurs. If not, then we can simply set the corresponding $\xi_k = 0$. Then the governing equation (7.5) at step $m > j$ can be written

$$AQ_m - Q_m\tilde{T}_m = \beta_{m+1}q_{m+1}e_{m+1}^* + F_m,$$

where

$$\tilde{T}_m = T_m + \sum_{l=j}^{m}\sum_{k=1}^{l-1}\xi_k^{(l)}e_ke_l^*.$$

However, it is clear that again $\tilde{T}_m^*e_m = T_m^*e_m = T_me_m$, and then the argument of Theorem 4 can be also used for the general case. ∎

At first glance the result of Theorem 5 is very surprising. Because any orthogonalization appears to be such a disruption of the otherwise simple structure of the Lanczos algorithm, one might expect the output of the algorithm to be changed drastically as well. But this is only true if one thinks in terms of the exact algorithm. There the matrix T_j loses its simple tridiagonal structure when it is modified to $\tilde{T}_j$. In finite precision the quantity to consider is not T_j, but $L_j^*T_jL_j^{-*}$, which is almost the exact projection of A on $\operatorname{span}(Q_j)$. Moreover, it is an upper Hessenberg matrix, but so is $L_j^*\tilde{T}_jL_j^{-*}$. Therefore the modification of T_j due to an orthogonalization actually does not change the structure of the important quantities in the algorithm. This explains the relative ease with which Theorem 5 follows from Theorem 4.

In order to prove Theorem 5 within our model, we had to assume that the semiorthogonalization strategy maintains a level of orthogonality of $\omega_0 = \sqrt{\varepsilon/j}$ among the Lanczos vectors. The dependence of j in ω_0 is a nuisance, since practical experience shows that semiorthogonality, i.e., a $\sqrt{\varepsilon}$ level, is enough for computing an accurate T_j. The reason that Theorem 5 is weaker than we would like resides in the assumption implicitly made by using Lemma 3 that *all* off diagonal elements of W_j assume the maximum value ω_0. This assumption cannot be avoided, because there could be several largish elements in the last column of W_j. In many cases only some elements in the last column of W_j will be large enough to force an orthogonalization. The majority of off diagonal elements of W_j will be well below this threshold. Therefore the use of $\sqrt{\varepsilon}$ for a practical algorithm is justified, although we did not prove it rigorously.

7.1. *Partial Reorthogonalization*

Originally the Lanczos algorithm was executed only with full reorthogonalization (FRO). This amounted to an orthogonalization of the new q'_{j+1} against all previous q_j at every step, i.e.

$$r'_j \equiv \beta'_{j+1} q'_{j+1} = Aq_j - \alpha_j q_j - \beta_j q_{j-1} - f'_j,$$

$$r_j \equiv r'_j - \sum_{k=1}^{j} \left(r_j'^* q_k\right) q_k.$$

It is clear that FRO will satisfy (7.2) and (7.3) for a general semiorthogonalization strategy. Actually we expect that $|q_k^* q_{j+1}| \leqslant \sqrt{n}\,\varepsilon$, i.e. much more than necessary for (7.2).

There is a minor point still to be considered. In (7.1) we do not consider an orthogonalization against q_j. However, since $|r_j'^* q_j| = \beta'_{j+1}|q_{j+1}'^* q_j| \leqslant \varepsilon \|A\|$, we can write the FRO as

$$\beta_{j+1} q_{j+1} = Aq_j - \alpha_j q_j - \beta_j q_{j-1} - \sum_{k=1}^{j-1} \xi_k q_k - f_j,$$

where $f_j = f'_j + \beta'_{j+1}(q_{j+1}'^* q_j) q_j$. Therefore with $\xi_k = r_j'^* q_k$, $k = 1, \ldots, j-1$, FRO is a semiorthogonalization strategy for the Lanczos algorithm and Theorem 5 holds. On the other hand Theorem 5 assures us that only a level of orthogonality of ω_0 among the Lanczos vectors is sufficient. FRO is therefore not efficient, since the extra orthogonality gained does not produce a more accurate T_j.

This insight is the basis for Grcar's [4] periodic reorthogonalization. In this method one has to update an n-vector which simulates the error in the current Lanczos vector as compared to the ideal Lanczos vector. If this estimate for the error rises above the $\sqrt{\varepsilon}$ level, a full reorthogonalization of the current Lanczos vector and the one preceding it against all the previous ones is performed. If the error estimate is correct, Grcar's analysis shows that the

Lanczos algorithm with periodic reorthogonalization computes a T_j which is accurate up to roundoff.

Periodic reorthogonalization can be improved in two ways by using the recurrence from Theorem 1. Based on this recurrence, we only update a j-vector, which contains estimates $\omega_{j+1\,k}$ for the terms $q_{j+1}'^{*}q_k$, $k=1,\dots,j$. Secondly, since the $\omega_{j+1\,k}$'s indicate against which previous Lanczos vectors orthogonality has been lost, the current Lanczos vector has to be orthogonalized only against *some* of the previous Lanczos vectors. The resulting new method is called *partial reorthogonalization* (PRO).

The success of PRO depends very much on an accurate estimate for $q_{j+1}'^{*}q_k$. This is not a trivial task, since the recurrence (4.1) involves among others terms of the type $f_j^{*}q_k - f_k^{*}q_j$, which are not directly available in the algorithm, yet crucial for the recurrence. This problem is discussed in detail in [18]. Similarly it is not obvious against which previous Lanczos vectors to orthogonalize when the recurrence signals that orthogonality beyond the threshold value of $\sqrt{\varepsilon}$ has been lost. Of course PRO forces an orthogonalization against all q_k where $|q_{j+1}'^{*}q_k|$ exceeds the threshold, but it is more economical to perform orthogonalizations against "batches" of Lanczos vectors, containing the offending ones and a certain number of neighboring vectors. These computational details of PRO are discussed in [18].

The Lanczos algorithm with PRO at an abstract level therefore can be written as follows:

(1) Perform a regular Lanczos step:

$$\beta_{j+1}'q_{j+1}' = Aq_j - \alpha_j q_j - \beta_j q_{j-1} - f_j'. \tag{7.7a}$$

(2) Update the estimates $\omega_{j+1\,k}$ for $q_{j+1}'^{*}q_k$ for $k=1,\dots,j$, using the recurrence (4.1).

(3) Based on the information from the $\omega_{j+1\,k}$, determine a set of indices $L(j)=\{k \mid 1 \leqslant k \leqslant j\}$ and compute

$$\beta_{j+1}q_{j+1} = \beta_{j+1}'q_{j+1}' - \sum_{k\in L(j)} \left(\beta_{j+1}'q_{j+1}'^{*}q_k\right)q_k - f_j. \tag{7.7b}$$

Clearly, with $\xi_k = \beta_{j+1}'q_{j+1}'^{*}q_k$, PRO is a semiorthogonalization strategy. Theorem 5 can be applied and guarantees the computation of a T_j which up to roundoff is the orthogonal projection of A onto span(Q_j).

7.2. *Selective Orthogonalization*

The previous section was a natural application of Theorem 5. In order to check whether selective orthogonalization (SO) is also a semiorthogonalization strategy for the Lanczos algorithm, let us first recall the result and the notation of Paige's theorem (Theorem 2), which forms the basis for SO.

Paige's theorem describes how the new vector q'_{j+1} behaves when orthogonality is lost: it is tilted towards the vectors y_i, which are approximate eigenvectors for the matrix A. The quantity $\gamma_{ii}/\beta'_{j+1}\sigma_{ji}$ is a measure for the loss of orthogonality in direction of a certain vector y_i. Our general assumptions on the Lanczos algorithm imply that γ_{ii} is of the order of the roundoff unit. Let us therefore assume that $|\gamma_{ii}| \leqslant \varepsilon\|A\|$. The only way that $y_i^* q'_{j+1}$ can become large is by $\beta_{ji} \equiv \beta'_{j+1}\sigma_{ji}$ becoming small. SO therefore computes and monitors some of the β_{ji}. If one β_{ji} becomes smaller than a certain threshold value κ_j, then q'_{j+1} is orthogonalized against the corresponding y_i. The jth step of the Lanczos algorithm with SO can therefore be written as follows:

(1) Perform a regular Lanczos step:

$$\beta'_{j+1}q'_{j+1} = Aq_j - \alpha_j q_j - \beta_j q_{j-1} - f'_j. \tag{7.8a}$$

(2) Determine the set

$$L(j) = \{\, i | 1 \leqslant i \leqslant j,\ \beta_{ji} < \kappa_j \,\}. \tag{7.8b}$$

(3) Compute $y_i = Q_j s_i$ for $i \in L(j)$. Then the next Lanczos vector is given by

$$\beta_{j+1}q_{j+1} = \beta'_{j+1}q'_{j+1} - \sum_{i \in L(j)} \left(\beta'_{j+1}q'^{*}_{j+1}y_i\right)y_i - f_j. \tag{7.8c}$$

The set $L(j)$ may be empty; then nothing will be done in step 3. This is only a simplified version of an actual implementation of SO; for example, the y_i are not recomputed. However, (7.8) catches the main features of SO, and it is sufficient to consider here as a model of the actual computation.

It is not obvious at all that SO as defined in (7.8) is a semiorthogonalization strategy. We want to show first that SO formally follows the pattern in (7.2). We have

$$\begin{aligned}
\beta_{j+1}q_{j+1} &= \beta'_{j+1}q'_{j+1} - \sum_{i \in L(j)} \left(\beta'_{j+1}q'^{*}_{j+1}y_i\right)Q_j s_i - f_j \\
&= \beta'_{j+1}q'_{j+1} - \sum_{k=1}^{j} \sum_{i \in L(j)} \left(\beta'_{j+1}q'^{*}_{j+1}y_i\right)\sigma_{ki}q_k - f_j.
\end{aligned} \tag{7.9}$$

Recall that the eigendecomposition of T_j is given by $T_jS_j = S_j\Theta_j$, $s_i^* = (\sigma_{1i},\dots,\sigma_{ji})$. Also note that for the unwanted jth term

$$|\xi_j| \leqslant \sum_{i\in L(j)} \beta'_{j+1}\frac{|\gamma_{ii}|}{\beta'_{j+1}\sigma_{ji}}\sigma_{ji} = |L(j)||\gamma_{ii}| \leqslant j\varepsilon\|A\|.$$

The effects of SO regarding the jth Lanczos vector are hence of the same order as the roundoff unit, and we can subsume them in the f_j-term. Hence SO is formally a semiorthogonalization strategy with

$$\xi_k = \sum_{i\in L(j)} \left(\beta'_{j+1}q'^*_{j+1}y_i\right)\sigma_{ki}, \qquad k=1,\dots,j-1.$$

The method of SO is due to Parlett and Scott [15]. They suggest the use of the threshold $\kappa_j = \sqrt{\varepsilon}\|T_j\|$ in order to maintain semiorthogonality among the Lanczos vectors. The following theorem shows, from a different perspective, why this is the right choice.

THEOREM 6. *If the first j Lanczos vectors are semiorthogonal, if $|\gamma_{ii}| \leqslant \varepsilon\|A\|$, and if κ_j is chosen such that $\kappa_j \geqslant j\|A\|\sqrt{\varepsilon}$, then one step of SO according to* (7.8) *produces a vector q_{j+1} such that*

$$\|Q_j^*q_{j+1}\| \leqslant \sqrt{\varepsilon} + O(j^{3/2}\|A\|\varepsilon). \tag{7.10}$$

Proof. Let $w_k = Q_j^*q_k$ for $k=1,\dots,j+1$ and $w'_{j+1} = Q_j^*q'_{j+1}$. Then $W_j = (w_1,\dots,w_j)$. Let $S_j = (s_1,\dots,s_j)$ be defined as before. Multiplying (7.9) by Q_j^* and using this notation, one obtains

$$\beta_{j+1}w_{j+1} = \beta'_{j+1}w'_{j+1} - \beta'_{j+1}\sum_{k=1}^{j}\sum_{i\in L(j)}\left(q'^*_{j+1}y_i\right)\sigma_{ki}w_k - Q_j^*f_j$$

$$= \beta'_{j+1}w'_{j+1} - \beta'_{j+1}\sum_{i\in L(j)}\left(w'^*_{j+1}s_i\right)\sum_{k=1}^{j}\sigma_{ki}w_k - Q_j^*f_j. \tag{7.11}$$

Because of the symmetry of W_j it follows that

$$\beta_{j+1}w_{j+1} = \beta'_{j+1}\left(w'_{j+1} - \sum_{i\in L(j)}\left(w'^*_{j+1}s_i\right)W_js_i\right) - Q_j^*f_j. \tag{7.12}$$

Since w'_{j+1} is a j-vector, it can be expanded in terms of the orthonormal vectors s_i,

$$w'_{j+1} = \sum_{i=1}^{j} \varphi_i s_i \qquad \text{with} \quad \varphi_i \equiv w'^{*}_{j+1} s_i .$$

Then (7.12) becomes

$$\begin{aligned} \beta_{j+1} w_{j+1} &= \beta'_{j+1}\left(\sum_{i=1}^{j} \varphi_i s_i - \sum_{i \in L(j)} \varphi_i W_j s_i \right) - Q_j^* f_j \\ &= \beta'_{j+1}\left(\sum_{i \notin L(j)} \varphi_i s_i + \sum_{i \in L(j)} \varphi_i (I_j - W_j) s_i \right) - Q_j^* f_j . \qquad (7.13) \end{aligned}$$

Therefore

$$\begin{aligned} \beta_{j+1} \| w_{j+1} \| &\leqslant \beta'_{j+1} \Big\{ (j - |L(j)|) \max_{i \notin L(j)} |\varphi_i| \\ &\quad + |L(j)| \| I_j - W_j \| \max_{i \in L(j)} |\varphi_i| \Big\} + \varepsilon \| A \| \\ &\leqslant \beta'_{j+1} \Big\{ j \max_{i \notin L(j)} |\varphi_i| + j \| I_j - W_j \| \max_{i \in L(j)} |\varphi_i| \Big\} + \varepsilon \| A \| . \qquad (7.14) \end{aligned}$$

We can bound the terms in (7.14) further. Consider the definition of $L(j)$ in (7.8b). It follows that $i \notin L(j)$ iff $\beta'_{j+1} \sigma_{ji} \geqslant \kappa_j$. Using Paige's theorem, we have $\beta'_{j+1} \sigma_{ji} = \gamma_{ii} / \varphi'_i$; hence for $i \notin L(j)$ we have the following bound:

$$|\varphi_i| = \frac{|\gamma_{ii}|}{\beta'_{j+1} \sigma_{ji}} \leqslant \frac{\varepsilon \| A \|}{\kappa_j} \leqslant \frac{\sqrt{\varepsilon}}{j}, \qquad (7.15)$$

with the choice of $\kappa_j \geqslant j \| A \| \sqrt{\varepsilon}$. On the other hand, if $i \in L(j)$, we simply estimate $|\varphi_i| \leqslant \| w'_{j+1} \|$, and $\| w'_{j+1} \|$ can be estimated by using (4.7) and the semiorthogonality of the first j Lanczos vectors. One obtains

$$\beta'_{j+1} |\varphi_i| \leqslant \beta'_{j+1} \| w'_{j+1} \| \leqslant 2 \| A \| \sqrt{j \varepsilon} + O(\varepsilon \| A \|). \qquad (7.16)$$

Finally, using a result analogous to Lemma 1 and again the semiorthogonality, it follows that

$$\|I_j - W_j\| \leqslant (j-1)\sqrt{\varepsilon}. \tag{7.17}$$

Assume now that $|L(j)| = k$, where k is a small integer; then substituting (7.15)–(7.17) into (7.14), it follows that

$$\begin{aligned}\beta_{j+1}\|w_{j+1}\| &\leqslant \beta'_{j+1} j \frac{\sqrt{\varepsilon}}{j} + k(j-1)\sqrt{\varepsilon}\left[2\|A\|\sqrt{j\varepsilon} + O(\varepsilon\|A\|)\right] + O(\varepsilon\|A\|) \\ &\leqslant \beta'_{j+1}\sqrt{\varepsilon} + O(j^{3/2}\|A\|\varepsilon). \end{aligned} \tag{7.18}$$

Now it can be shown that $\beta_{j+1} = [1 + O(\varepsilon)]\beta'_{j+1}$, and (7.10) follows. ■

In order to appreciate Theorem 6 several more remarks are necessary. The proof of Theorem 6 seems to indicate that from an SO point of view it would be more natural to define the level of orthogonality by using $\|Q_j^* q_{j+1}\|$ instead of using $\|Q_j^* q_{j+1}\|_\infty$ as we did. Assuming that $\|Q_k^* q_{k+1}\| < \sqrt{\varepsilon}$, it would be possible to prove (7.10) with an $O(j\varepsilon\|A\|)$ term. With this more realistic interpretation Theorem 6 indeed shows that SO maintains semiorthogonality among the Lanczos vectors in the sense that $\|Q_k^* q_{k+1}\| \leqslant \sqrt{\varepsilon}$ for $k = 1, \ldots, j$.

Equation (7.13) makes clear how SO goes about maintaining semiorthogonality. The loss of orthogonality vector w'_{j+1} is decomposed into its eigencomponents. The components which have grown too large $[i \in L(j)]$ are reduced by orthogonalization to roundoff level; the other components $(i \notin L(j))$ remain unchanged. The key to the understanding why SO maintains semiorthogonality hence lies in (7.13). The remaining part of the proof of Theorem 6 only translates the informal argument above into exact estimates.

Equation (7.13) also illustrates why SO had some problems in gaining wide acceptance as a means of maintaining semiorthogonality. The proper way to study SO is in terms of the y_i or, as in (7.13), in terms of the s_i. This is conceptually more difficult than the apparent and "natural" way to study SO in terms of the Lanczos vectors. This different point of view only involves a change of basis in span(Q_j); however, the failure to recognize this prompted wrong judgements about SO.

Finally, by making the requirements on κ_j more stringent, it is possible to show that SO will also be able to maintain a level of orthogonality of ω_0 as defined in (7.2). To be precise, we have the following

COROLLARY. *If the level of orthogonality among the Lanczos vectors is* ω_0, *if* $\gamma_{ii} \leqslant \varepsilon\|A\|$, *and if* κ_j *is chosen such that* $\kappa_j \geqslant j^{1/2}\|A\|\sqrt{\varepsilon/2}$, *then one* *step of selective orthogonalization produces a* q_{j+1} *such that*

$$\max_{1 \leqslant k \leqslant j} |q_{j+1}^* q_k| \leqslant \omega_0 + O(j\varepsilon\|A\|).$$

The proof is analogous to Theorem 6. With the help of this corollary we can apply Theorem 5, and it follows that SO also produces a matrix T_j which is, up to roundoff, the orthogonal projection of A onto span(Q_j).

8. APPLICATIONS

So far we have discussed the Lanczos algorithm in finite precision only as a way of tridiagonalizing the given matrix A. One main application we had in mind, however, was solving linear systems of equations. Recall from Section 2 that in order to compute an approximate solution vector x_j to $Ax = b$, we solved $T_j h_j = \beta_1 e_1$ and then computed $x_j = Q_j h_j$. Suppose that we have employed some semiorthogonalization strategy, computed T_j and Q_j, and determine now

$$x_j = Q_j T_j^{-1} \beta_1 e_1. \tag{8.1}$$

In this case it is easier to compare x_j with $\bar{x}_j$, rather than to estimate $b - Ax_j$. Here $\bar{x}_j$ is the best approximation from span(Q_j), i.e., using orthogonal projections,

$$\bar{x}_j = N_j\left(N_j^* A N_j\right)^{-1} N_j^* b. \tag{8.2}$$

Recall that $N_j N_j^*$ is the orthogonal projector onto span(Q_j), where $N_j = Q_j L_j^*$ is defined as in Section 6. Since $q_1 = n_1$, we have $\bar{x}_j = N_j(N_j^* A N_j)^{-1}\beta_1 e_1$. According to Theorem 5, we have that $T_j + V_j = N_j^* A N_j$, where the elements of V_j are of $O(\varepsilon\|A\|)$. Therefore we can replace $N_j^* A N_j$ by T_j, since the perturbation introduced this way in the computation of x_j is of the same order as the backward error, which we have to take into account anyway when solving linear systems by Gaussian elimination. It depends only on $\kappa(T_j)$, the condition number of T_j.

The only way that the finite precision Lanczos algorithm affects the computation of x_j versus $\bar{x}_j$ is through the formation of x_j as a linear combination of the q_k, which are not orthogonal. This effect can be estimated by comparing Q_j with N_j:

$$\|Q_j - N_j\| \leqslant \|N_j L_j^* - N_j\| = \|L_j^* - I_j\| \leqslant \sqrt{2j\varepsilon}, \tag{8.3}$$

where we have used that the level of orthogonality among the Lanczos vectors is ω_0. Hence if the Lanczos algorithm is used for solving linear systems of equations, and the required accuracy is not less than $\sqrt{\varepsilon}$, the solution x_j computed from (8.1) is as good as the best solution obtainable from span(Q_j). Only when a higher accuracy is required additional steps have to be taken (cf. Parlett [14]).

If the algorithm is used for computing eigenvalues the situation is even better. Theorem 5 assures us that the eigenvalues of T_j are, up to roundoff, the Rayleigh-Ritz approximations from span(Q_j). This is the best we could hope for.

I wish to thank Beresford Parlett for his constructive criticism regarding the presentation of the results, and Chris Paige, who suggested several valuable improvements.

REFERENCES

1 J. Cullum and R. Willoughby, Lanczos and the computation in specified intervals of the spectrum of large sparse real symmetric matrices, in *Sparse Matrix Proceedings* (I. Duff and G. W. Stewart, Eds.), SIAM, Philadelphia, 1979.
2 J. Cullum and R. Willoughby, Computing eigenvectors and eigenvalues of large sparse symmetric matrices using Lanczos tridiagonalization, in *Numerical Analysis Proceedings, Dundee 1979* (G. A. Watson, Ed.), Springer, Berlin, 1980.
3 G. Golub, R. Underwood, and J. H. Wilkinson, The Lanczos algorithm for the $Ax - \lambda Bx$ Problem, Tech. Rep. STAN-CS-72-720, Comput. Sci. Dept., Stanford Univ., 1972.
4 J. Grcar, Analyses of the Lanczos algorithm and of the approximation problem in Richardson's method, Ph.D. Thesis, Univ. of Illinois at Urbana-Champaign, 1981.
5 S. Kaniel, Estimates for some computational techniques in linear algebra, *Math. Comp.* 20:369–378 (1966).
6 C. Lanczos, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, *J. Res. Nat. Bur. Standards* 45:255–282 (1950).
7 J. G. Lewis, Algorithms for sparse matrix eigenvalue problems, Ph.D. Thesis, Report STAN-CS-77-595, Comput. Sci. Dept., Stanford Univ., 1977.
8 C. Paige, The computation of eigenvalues and eigenvectors of very large sparse matrices, Ph.D. Thesis, Univ. of London, 1971.
9 C. Paige, Computational variants of the Lanczos method for the eigenproblem, *J. Inst. Math. Appl.* 10:373–381 (1972).
10 C. Paige, Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix, *J. Inst. Math. Appl.* 18:341–349 (1976).
11 C. Paige, Accuracy and effectiveness of the Lanczos algorithm for the symmetric eigenproblem, *Linear Algebra Appl.* 34:235–258 (1980).

12 C. Paige and M. Saunders, Solution of sparse indefinite systems of linear equations, *SIAM J. Numer. Anal.* 12:617–629 (1975).
13 B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, N.J., 1980.
14 B. N. Parlett, A new look at the Lanczos algorithm for solving symmetric systems of linear equations, *Linear Algebra Appl.* 29:323–346 (1980).
15 B. N. Parlett and D. Scott, The Lanczos algorithm with selective orthogonalization, *Math. Comp.* 33:217–238 (1979).
16 Y. Saad, On the rate of convergence of the Lanczos and the block Lanczos Methods, *SIAM J. Numer. Anal.* 17:687–706 (1980).
17 D. S. Scott, How to make the Lanczos algorithm converge slowly, *Math. Comp.* 33:239–247 (1979).
18 H. Simon, The Lanczos algorithm with partial reorthogonalization, *Math. Comp.* 42 (1984).
19 H. Takahasi and M. Natori, Eigenvalue problem of large sparse matrices, *Rep. Comput. Centre Univ. Tokyo* 4:129–148 (1971–1972).
20 J. H. Wilkinson, *Rounding Errors in Algebraic Processes*, Prentice-Hall, Englewood Cliffs, N.J., 1964.
21 J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon, Oxford, 1965.

Received 28 June 1982; revised 6 June 1983