J. Inst. Maths Applics (1976) **18**, 341–349

Error Analysis of the Lanczos Algorithm for Tridiagonalizing a Symmetric Matrix

C. C. PAIGE[†]

School of Computer Science, McGill University, Montreal, Quebec, Canada

[Received 5 June 1975]

The Lanczos algorithm for tridiagonalizing a symmetric matrix is the basis for several methods for solving sets of linear equations as well as for solving the eigenproblem. These methods are very useful when the matrix is large and sparse. A complete rounding error analysis of the algorithm is presented here, giving among other results an important expression for the loss of orthogonality of the computed vectors. The results here can be used to analyze the many methods which are based on the Lanczos algorithm.

1. Introduction

IN 1950 Cornelius Lanczos described the application of the three term recurrence relations for orthogonal polynomials to the reduction of a matrix to tridiagonal form. In theory, for a given $n \times n$ symmetric matrix $\mathbf{A}$ and a given vector $\mathbf{v}_1$ with unit 2-norm, his algorithm produces in n steps an orthonormal matrix $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n]$ and a tridiagonal matrix $\mathbf{T}$ such that

$$\mathbf{AV} = \mathbf{VT}.$$

The eigenvalues of $\mathbf{T}$ are clearly the eigenvalues of $\mathbf{A}$, and so the eigen decomposition of $\mathbf{A}$ can be found from that of the more easily handled symmetric matrix $\mathbf{T}$. Lanczos suggested this in his original paper on the subject (1950), but he also noted that several of the eigenvalues of $\mathbf{T}_k$, the leading $k \times k$ part of $\mathbf{T}$, and the result of the first k steps of the algorithm, were usually good approximations to some eigenvalues of $\mathbf{A}$, even for $k \ll n$. As a result of this behaviour, and because the matrix $\mathbf{A}$ is only required in one matrix-vector multiplication per step, this algorithm can be used economically to approximate eigenvalues of big matrices in far fewer than n steps. Lehmann (1966) has shown how optimal eigenvalue intervals can be obtained in such uncompleted cases.

Lanczos (1952) showed how his algorithm, which he called the method of minimized iterations, could be applied to solving sets of linear equations. At about the same time Hestenes & Stiefel (1952) published their method of Conjugate Gradients for solving systems of linear equations with a positive definite matrix; this too is based on the Lanczos algorithm. Householder (1964) describes this connection between the method of Conjugate Gradients and the Lanczos algorithm, and devotes considerable space to this important basic algorithm for tridiagonalizing matrices. Some methods for solving linear least squares problems are also based on the Lanczos algorithm (see for example Paige, 1974), as is a method for finding singular values that was suggested by Golub & Kahan (1965). More recently some

[†] Supported by National Research Council of Canada Grant A8652.

methods for solving systems of equations with an indefinite symmetric matrix have been proposed by Paige & Saunders (1975), and these use the Lanczos algorithm directly.

Unfortunately, because of the presence of rounding errors the Lanczos algorithm does not behave in the way the mathematical theory indicates, and partly for this reason the methods based on the Lanczos algorithm were largely ignored once they had been superseded by superior methods for small matrices. More recently a focusing of attention on problems involving large sparse matrices has led to a reawakening of interest in these methods, because they are so well suited to such problems. Paige (1971, 1972) and Reid (1971, pp. 231–254; 1972), among others, have demonstrated the effectiveness of some of these methods, both for the eigenproblem and for solutions of equations. These studies indicate that these algorithms are competitive for problems involving large sparse matrices, despite the effect of rounding errors. An understanding of some of the error properties of the Lanczos algorithm has already led to a more informed choice of computational algorithm (Paige, 1972), and the full rounding error analysis to be given here will lead to more understanding, and hopefully better computational methods, as well as to proofs of convergence of some of these methods in the presence of rounding errors.

Because the Lanczos algorithm is so basic to many methods and so relevant to many present day computations, a full rounding error analysis will be given here for one computational variant. The analysis is lengthy and involved, and to counteract this a little all the important results are summarized in equations (15) to (23) in a theorem in Section 2, and the meaning of the results is discussed briefly following the statement of the theorem. The results of an error analysis of another possible computational variant of the algorithm will be quoted. In fact the two computational algorithms to be considered are the successful ones described by Paige (1972).

The rounding error analysis only gives relations and bounds for the rounding errors produced in computing $\mathbf{V}$ and $\mathbf{T}$ in $\mathbf{AV} = \mathbf{VT}$, and for the difference between $\mathbf{V}^T\mathbf{V}$ and $\mathbf{I}$. The applications of these results to specific methods for finding eigenvalues, solving equations, etc., will be treated in later papers. The complete error analysis is essentially different from those given by Wilkinson (1963, 1965), although the basic tools for finding the initial error terms will come from those works, as did the initial understanding of rounding error behaviour that led to the results presented here.

2. Rounding Error Analysis

The computational variant of the Lanczos algorithm that will be analyzed here will be called A1, and in the absence of rounding errors can be described as follows. Let $\mathbf{v}_1$ be given with $\mathbf{v}_1^T\mathbf{v}_1 = 1$, then

$$\mathbf{u}_1 := \mathbf{A}\mathbf{v}_1 \tag{1}$$

and for $j = 1, 2, 3, \ldots$ do steps (2) to (6)

$$\alpha_j := \mathbf{v}_j^T\mathbf{u}_j \tag{2}$$

$$\mathbf{w}_j := \mathbf{u}_j - \alpha_j\mathbf{v}_j \tag{3}$$

$$\beta_{j+1} := +(\mathbf{w}_j^T\mathbf{w}_j)^{\frac{1}{2}}, \quad \text{if} \quad \beta_{j+1} = 0 \quad \text{then } STOP \tag{4}$$

$$\mathbf{v}_{j+1} := \mathbf{w}_j/\beta_{j+1} \tag{5}$$

$$\mathbf{u}_{j+1} := \mathbf{A}\mathbf{v}_{j+1} - \beta_{j+1}\mathbf{v}_j. \tag{6}$$

This essentially describes a way of producing the vectors $\mathbf{v}_j$ and the coefficients α_j, β_{j+1} of the symmetric tridiagonal matrix. The actual stopping criterion would depend on the use to which these were being put. This particular variant has been chosen for analysis because it is the one requiring the least storage, in fact only 2 vectors $\mathbf{u}_j$ and $\mathbf{v}_j$ are needed in (1) and (2), while $\mathbf{w}_j$ can overwrite $\mathbf{u}_j$ in (3) and $\mathbf{v}_{j+1}$ can overwrite $\mathbf{w}_j$ in (5), so finally $\mathbf{u}_{j+1}$ can overwrite $\mathbf{v}_j$ in (6) if it is computed an element at a time. The square root in (4) and the normalization in (5) are not essential, but they prevent overflow or underflow, and make the algorithm and its analysis more elegant, for a very small cost. Another satisfactory variant, A2, requires the computation $\alpha_j := \mathbf{v}_j^T\mathbf{A}\mathbf{v}_j$, however this requires a third storage vector. This second algorithm has been analyzed in part, under the name A(1, 7), by Paige (1972) and in detail by Paige (1971), and the results will be quoted here. Paige (1972) shows that not all computational variants of the Lanczos process are well behaved, and as a result attention need only be focused on algorithms A1 and A2 here.

In the rounding error analysis it will be assumed that floating point computation with a relative precision ε is used, and the derivation of the basic error terms will follow the work of Wilkinson (1963, 1965). For simplicity we will ignore terms in ε^2 and higher, as these would have a negligible effect on our results. What is more, the symbol ε will be used with some abandon to represent terms whose absolute values are bounded by the relative precision, as well as representing the relative precision itself when ε appears as a factor in a bound. Finally $\mathbf{D}(f(\varepsilon))$ will be used to represent a diagonal matrix each of whose diagonal elements is bounded by f(relative precision). The purpose of this approach is to avoid the essentially straightforward parts of the analysis being drowned in unwieldy subscripts and superscripts.

It will be assumed that

$$\|\mathbf{A}\| = \sigma, \qquad \|\,|\mathbf{A}|\,\| = \beta\sigma, \tag{7}$$

where from now on $\|\cdot\|$ represents the 2-norm, and $|\mathbf{A}|$ is the matrix with elements $|\alpha_{ij}|$, α_{ij} being the elements of $\mathbf{A}$.

Several basic computations are used repeatedly, and so their results will be summarized here (see Wilkinson, 1963), with $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{w}$ representing n-vectors, and α and β scalars.

Vector subtraction

$$fl(\mathbf{u}-\alpha\mathbf{v}) = \mathbf{u}-\alpha\mathbf{v}-\delta\mathbf{w}, \qquad \|\delta\mathbf{w}\| \leqslant (\|\mathbf{u}\|+2\|\alpha\mathbf{v}\|)\varepsilon \tag{8}$$

where fl represents floating point computation, and $\delta\mathbf{w}$ represents the error.

Vector inner-product

$$fl(\mathbf{v}^T\mathbf{u}) = (\mathbf{v}+\delta\mathbf{v})^T\mathbf{u}, \qquad \|\delta\mathbf{v}\| \leqslant n\varepsilon\|\mathbf{v}\| \tag{9}$$

Matrix-vector multiplication

If there are at most m non-zero elements per row of $\mathbf{A}$, then

$$fl(\mathbf{A}\mathbf{u}) = (\mathbf{A}+\delta\mathbf{A})\mathbf{u}, \qquad |\delta\mathbf{A}| \leqslant m\varepsilon|\mathbf{A}|, \tag{10}$$

so with (7)

$$\|\delta\mathbf{A}\| \leqslant \|\,|\delta\mathbf{A}|\,\| \leqslant m\varepsilon\|\,|\mathbf{A}|\,\| = m\beta\varepsilon\sigma. \tag{11}$$

This bound will be used in place of the usual $\|\,|\mathbf{A}|\,\| \leqslant n^{\frac{1}{2}}\|\mathbf{A}\|$, as this latter bound would give a poor indication of the accuracy of the process for large n and small m, which is just the case where the algorithm is most useful.

Normalization

Making use of (9) and assuming that taking a square root introduces a relative error no greater than ε,

$$\beta = fl((\mathbf{w}^T\mathbf{w})^{\frac{1}{2}}) = [1+\varepsilon(n+2)/2]\ \|\mathbf{w}\| \tag{12}$$

$$\mathbf{v} = fl(\mathbf{w}/\beta) = \mathbf{D}(1+\varepsilon)\mathbf{w}/\beta \tag{13}$$

which gives the theoretical result

$$\mathbf{v}^T\mathbf{v} = (1+2\varepsilon)\mathbf{w}^T\mathbf{w}/\beta^2 = 1+(n+4)\varepsilon. \tag{14}$$

We have now supplied the necessary background for the error analysis. The analysis itself is surprisingly lengthy and involved, and so we will now give a theorem summarizing the results that will be proved in the remainder of this section.

THEOREM. *Let* $\mathbf{A}$ *be an* $n\times n$ *real symmetric matrix with at most m non-zero elements in any row, and such that* $\|\mathbf{A}\| = \sigma$, $\|\,|\mathbf{A}|\,\| = \beta\sigma$. *If the variant of the Lanczos algorithm described by equations* (1) *to* (6) *is implemented on a floating point digital computer with relative precision* ε *and applied for k steps to* $\mathbf{A}$ *starting with a normalized initial vector* $\mathbf{v}_1$, *then* α_j, β_{j+1}, $\mathbf{v}_{j+1}$ *will be computed for* $j = 1, 2, \ldots, k$ *such that*

$$\mathbf{A}\mathbf{V}_k = \mathbf{V}_k\mathbf{T}_k+\beta_{k+1}\mathbf{v}_{k+1}\mathbf{e}_k^T+\delta\mathbf{V}_k \tag{15}$$

$$\mathbf{V}_k \equiv [\mathbf{v}_1, \ldots, \mathbf{v}_k], \qquad \mathbf{T}_k \equiv \begin{bmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \beta_3 & \\ & \cdot & \cdot & \cdot & \cdot \\ & & \beta_k & \alpha_k \end{bmatrix}$$

$$\delta\mathbf{V}_k \equiv [\delta\mathbf{v}_1, \ldots, \delta\mathbf{v}_k],$$

where $\mathbf{e}_k$ *is the kth column of the unit matrix, and for* $j = 1, 2, \ldots, k$,

$$|\mathbf{v}_{j+1}^T\mathbf{v}_{j+1}-1| \leqslant \varepsilon_0 \tag{16}$$

$$\|\delta\mathbf{v}_j\| \leqslant \sigma\varepsilon_1 \tag{17}$$

$$\beta_{j+1}|\mathbf{v}_j^T\mathbf{v}_{j+1}| \leqslant 2\sigma\varepsilon_0 \tag{18}$$

$$|\beta_j^2+\alpha_j^2+\beta_{j+1}^2-\|\mathbf{A}\mathbf{v}_j\|^2| \leqslant 4j(3\varepsilon_0+\varepsilon_1)\sigma^2, \tag{19}$$

and we have used the notation

$$\varepsilon_0 \equiv (n+4)\varepsilon, \qquad \varepsilon_1 \equiv (7+m\beta)\varepsilon. \tag{20}$$

What is more, if $\mathbf{R}_k$ *is the strictly upper triangular matrix such that*

$$\mathbf{V}_k^T\mathbf{V}_k = \mathbf{R}_k^T+\mathrm{diag}\,(\mathbf{v}_j^T\mathbf{v}_j)+\mathbf{R}_k \tag{21}$$

then

$$\mathbf{T}_k\mathbf{R}_k-\mathbf{R}_k\mathbf{T}_k = \beta_{k+1}\mathbf{V}_k^T\mathbf{v}_{k+1}\mathbf{e}_k^T+\mathbf{H}_k \tag{22}$$

where $\mathbf{H}_k$ *is upper triangular with elements* η_{ij} *such that*

$$\left.\begin{aligned} &|\eta_{11}| \leqslant 2\sigma\varepsilon_0 \\ &\textit{and for } j = 2, 3, \ldots, k \\ &|\eta_{jj}| \leqslant 4\sigma\varepsilon_0 \\ &|\eta_{j-1,j}| \leqslant 2\sigma(\varepsilon_0+\varepsilon_1) \\ &|\eta_{ij}| \leqslant 2\sigma\varepsilon_1, \qquad i = 1, 2, \ldots, j-2. \end{aligned}\right\} \tag{23}$$

Throughout this theorem it has been assumed that $\beta_{j+1} \neq 0$ and $4j(3\varepsilon_0+\varepsilon_1) \ll 1$, and terms in ε^2 and higher have been ignored. A full analysis has shown that with a restriction on n similar to that on j here, the results are essentially the same when no terms are ignored.

To gain some insight into these results, we note that (16) is to be expected from the normalization, while (15) and (17) express an obvious discrepancy between the ideal and computed results. (18) describes the orthogonality of a vector to its predecessor, and indicates that a cancellation that results in small β_{j+1} can cause significant loss of orthogonality. The surprising result (19) shows that the jth column of $\mathbf{T}_k$ has almost the same 2-norm as $\mathbf{A}\mathbf{v}_j$. To make use of this error analysis in methods using the Lanczos algorithm to find eigenvalues or solve equations, it turns out to be important to have an expression describing the total loss of orthogonality, and this is why $\mathbf{R}_k$ has been introduced in (21). The growth of this orthogonality loss can be described by (22), with (23) giving bounds on the elements of $\mathbf{H}_k$. Equation (22) is therefore very important for subsequent analyses.

Two more comments are relevant. First, the effect of double length accumulation of vector inner products can essentially be accounted for by replacing n in (20) by unity. Second, with the algorithm A2 mentioned earlier, for which the unnormalized version has been described and analyzed by Paige (1971), the results are essentially the same except that two of the bounds become

$$\beta_{j+1}|\mathbf{v}_j^T\mathbf{v}_{j+1}| \leqslant 2j\sigma\varepsilon_0 \tag{24}$$

$$|\eta_{jj}| \leqslant 4j\sigma\varepsilon_0, \tag{25}$$

the only important difference being the factor j.

The remainder of this section will now be devoted to proving the theorem by making use of the results (7) to (14).

To start with, the vectors $\mathbf{v}_j$ are obtained by normalization in (4) and (5), and so (14) shows that (16) holds. Next, using (10)

$$\mathbf{u}_1 = \mathbf{A}\mathbf{v}_1 - \delta\mathbf{u}_1, \qquad \|\delta\mathbf{u}_1\| = \|\delta\mathbf{A}\mathbf{v}_1\| \leqslant m\beta\varepsilon\sigma, \tag{26}$$

$$\|\mathbf{u}_1\| \leqslant [1+\varepsilon(n+2m\beta+4)/2]\sigma \tag{27}$$

where it is assumed that $\mathbf{v}_1$ was found by normalizing a given vector as in (14).

In the initial stages of the analysis it will be impossible to give *a posteriori* bounds on all variables, and so several bounds will initially be given in terms of $\|\mathbf{u}_j\|$. It will be proved later that

$$\|\mathbf{u}_j\| \leqslant \sigma\{1+2j[7+m\beta+3(n+4)]\varepsilon\}. \tag{28}$$

The analysis of (2) is carried out by using (9) to give

$$\alpha_j = \mathbf{v}_j^T\mathbf{u}_j - \delta\alpha_j, \qquad |\delta\alpha_j| \leqslant n\|\mathbf{u}_j\|\varepsilon \tag{29}$$

$$|\alpha_j| \leqslant [1+\varepsilon(3n+4)/2]\|\mathbf{u}_j\|. \tag{30}$$

Equations (3) and (8) then give

$$\mathbf{w}_j = \mathbf{u}_j - \alpha_j\mathbf{v}_j - \delta\mathbf{w}_j, \qquad \|\delta\mathbf{w}_j\| \leqslant 3\|\mathbf{u}_j\|\varepsilon, \tag{31}$$

which combine with (29) to give

$$\|\mathbf{w}_j\|^2 = \|\mathbf{u}_j\|^2 + \alpha_j^2(\|\mathbf{v}_j\|^2 - 2) - 2\alpha_j\delta\alpha_j - 2\delta\mathbf{w}_j^T(\mathbf{u}_j - \alpha_j\mathbf{v}_j) + \|\delta\mathbf{w}_j^T\|^2 \tag{32}$$

and using the bounds (16), (29), (30) and (31)

$$\|\mathbf{w}_j\|^2 + \alpha_j^2 - \|\mathbf{u}_j\|^2 \leqslant (3n+10)\varepsilon\|u_j\|^2. \tag{33}$$

The vector $\mathbf{w}_j$ must now be normalized as in (4) and (5), so by using (12) and (33)

$$\beta_{j+1} = [1+\varepsilon(n+2)/2]\|\mathbf{w}_j\| \leqslant [1+(2n+6)\varepsilon]\|\mathbf{u}_j\|, \tag{34}$$

$$\beta_{j+1}\mathbf{v}_{j+1} = \mathbf{w}_j+\delta\mathbf{w}'_j, \qquad \|\delta\mathbf{w}'_j\| \leqslant \|\mathbf{u}_j\|\varepsilon. \tag{35}$$

Finally it can be seen by using (10) and (8), with the bounds (11) and (34), that the computation of (6) gives

$$\mathbf{u}_j = \mathbf{A}\mathbf{v}_j-\beta_j\mathbf{v}_{j-1}-\delta\mathbf{u}_j, \qquad \|\delta\mathbf{u}_j\| \leqslant (1+m\beta)\sigma\varepsilon+2\|\mathbf{u}_{j-1}\|\varepsilon, \tag{36}$$

and the special case of $j = 1$ is given in (26).

All the rounding errors that can be introduced by the computation have now been described and bounded in terms of $\|\mathbf{u}_j\|$. It now remains to manipulate these error terms to obtain a bound on $\|\mathbf{u}_j\|$ and to indicate what effect these rounding errors have on the computed results.

The error in each step is found by combining (35), (31) and (36) to give

$$\beta_{j+1}\mathbf{v}_{j+1} = \mathbf{A}\mathbf{v}_j-\alpha_j\mathbf{v}_j-\beta_j\mathbf{v}_{j-1}-\delta\mathbf{v}_j, \tag{37}$$

$$\|\delta\mathbf{v}_j\| = \|\delta\mathbf{w}'_j-\delta\mathbf{w}_j-\delta\mathbf{u}_j\| \leqslant (1+m\beta)\sigma\varepsilon+(4\|\mathbf{u}_j\|+2\|\mathbf{u}_{j-1}\|)\varepsilon. \tag{38}$$

It can be seen that equation (37) is just the jth column of equation (15).

It was shown by Paige (1972) that the performance of the algorithm depends partly on the successive vectors $\mathbf{v}_j$ and $\mathbf{v}_{j+1}$ not losing orthogonality unnecessarily. This orthogonality can be displayed by combining equations (35), (31) and (29) to give

$$\begin{aligned}\beta_{j+1}\mathbf{v}_j^T\mathbf{v}_{j+1} &= \mathbf{v}_j^T\mathbf{u}_j-\alpha_j\mathbf{v}_j^T\mathbf{v}_j+\mathbf{v}_j^T(\delta\mathbf{w}'_j-\delta\mathbf{w}_j)\\ &= \delta\alpha_j-\alpha_j(\mathbf{v}_j^T\mathbf{v}_j-1)+\mathbf{v}_j^T(\delta\mathbf{w}'_j-\delta\mathbf{w}_j)\end{aligned}$$

which with (16) gives

$$\beta_{j+1}|\mathbf{v}_j^T\mathbf{v}_{j+1}| \leqslant 2(n+4)\|\mathbf{u}_j\|\varepsilon, \tag{39}$$

as a result we see that orthogonality between these two vectors can be lost only if there is significant cancellation in (3), resulting in a small β_{j+1}.

In order to examine the possible loss of orthogonality in all the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_{k+1}$ we consider the strictly upper triangular matrix $\mathbf{R}_k$, with elements ρ_{ij}, defined in (21). If (15) is multiplied on the left by $\mathbf{V}_k^T$ and the resulting right hand side is equated with its own transpose, then

$$\begin{aligned}\mathbf{T}_k(\mathbf{R}_k^T+\mathbf{R}_k)-(\mathbf{R}_k^T+\mathbf{R}_k)\mathbf{T}_k = \beta_{k+1}(\mathbf{V}_k^T\mathbf{v}_{k+1}\mathbf{e}_k^T-\mathbf{e}_k\mathbf{v}_{k+1}^T\mathbf{V}_k)+\mathbf{V}_k^T\delta\mathbf{V}_k-\delta\mathbf{V}_k^T\mathbf{V}_k+\\ \operatorname{diag}(\mathbf{v}_i^T\mathbf{v}_i)\mathbf{T}_k-\mathbf{T}_k\operatorname{diag}(\mathbf{v}_i^T\mathbf{v}_i),\end{aligned} \tag{40}$$

where the diagonal elements on each side must be zero. Now

$$\mathbf{M}_k \equiv \mathbf{T}_k\mathbf{R}_k-\mathbf{R}_k\mathbf{T}_k \tag{41}$$

is upper triangular, and since the left hand side of (40) is just $\mathbf{M}_k-\mathbf{M}_k^T$, the strictly upper triangular portion of $\mathbf{M}_k$ can be equated directly to that of the right hand side of (40). But it follows directly from (41) that the diagonal elements μ_{jj} of $\mathbf{M}_k$ are just

$$\mu_{11} = -\beta_2\rho_{12}, \qquad \mu_{kk} = \beta_k\rho_{k-1,k}$$

$$\mu_{jj} = \beta_j\rho_{j-1,j}-\beta_{j+1}\rho_{j,j+1}, \qquad j = 2, \ldots, k-1,$$

where the individual terms are bounded in (39). Thus

$$\mathbf{M}_k = \mathbf{T}_k\mathbf{R}_k-\mathbf{R}_k\mathbf{T}_k = \beta_{k+1}\mathbf{V}_k^T\mathbf{v}_{k+1}\mathbf{e}_k^T+\mathbf{H}_k \tag{42}$$

where $\mathbf{H}_k$ is upper triangular with elements η_{ij} satisfying

$$\left.\begin{aligned} \eta_{11} &= -\beta_2\rho_{12} \\ &\text{and for } j = 2, \ldots, k \\ \eta_{jj} &= \beta_j\rho_{j-1,j} - \beta_{j+1}\rho_{j,j+1} \\ \eta_{j-1,j} &= \mathbf{v}_{j-1}^T\delta\mathbf{v}_j - \delta\mathbf{v}_{j-1}^T\mathbf{v}_j + \beta_j(\mathbf{v}_{j-1}^T\mathbf{v}_{j-1} - \mathbf{v}_j^T\mathbf{v}_j) \\ \eta_{ij} &= \mathbf{v}_i^T\delta\mathbf{v}_j - \delta\mathbf{v}_i^T\mathbf{v}_j, \qquad i = 1, 2, \ldots, j-2. \end{aligned}\right\} \tag{43}$$

From now on it will simplify matters to use the notation

$$\mu_j = \max_{i=1,2,\ldots,j} \{\|\mathbf{u}_i\|\}. \tag{44}$$

Using this with (32), (38), (34) and (16) gives the following bounds on the elements of $\mathbf{H}_k$

$$\left.\begin{aligned} |\eta_{11}| &\leqslant 2(n+4)\mu_1\varepsilon, \\ &\text{and for } j = 2, 3, \ldots, k \\ |\eta_{jj}| &\leqslant 4(n+4)\mu_j\varepsilon, \\ |\eta_{ij}| &\leqslant 2[(1+m\beta)\sigma + 6\mu_j]\varepsilon, \qquad i = 1, 2, \ldots, j-2, \\ |\eta_{j-1,j}| &\leqslant 2[(1+m\beta)\sigma + (n+10)\mu_j]\varepsilon. \end{aligned}\right\} \tag{45}$$

A bound must first be found on $\rho_{j-2,j} = \mathbf{v}_j^T\mathbf{v}_{j-2}$ before a bound can be given for μ_j, and to do this, note that the (1, 2) element of (42) gives

$$\alpha_1\rho_{12} - \alpha_2\rho_{12} - \beta_3\rho_{13} = \eta_{12}$$

while for $j = 3, 4, \ldots, k$ the $(j-1, j)$ element gives

$$\beta_{j-1}\rho_{j-2,j} + (\alpha_{j-1} - \alpha_j)\rho_{j-1,j} - \beta_{j+1}\rho_{j-1,j+1} = \eta_{j-1,j}$$

so that for $j = 2, 3, \ldots, k$, defining

$$\zeta_j \equiv (\alpha_{j-1} - \alpha_j)\beta_j\rho_{j-1,j} - \beta_j\eta_{j-1,j} \tag{46}$$

it follows that

$$\beta_j\beta_{j+1}\rho_{j-1,j+1} = \beta_{j-1}\beta_j\rho_{j-2,j} + \zeta_j = \zeta_j + \zeta_{j-1} + \,.\,.\, + \zeta_2$$

which with (46), (45), (39) and (30) gives

$$\beta_j\beta_{j+1}|\rho_{j-1,j+1}| \leqslant 2(j-1)[3(n+6)\mu_j + (1+m\beta)\sigma]\mu_j\varepsilon. \tag{47}$$

We can now proceed towards proving (28). Equation (36) gives

$$\|\mathbf{u}_j + \delta\mathbf{u}_j\|^2 = \|\mathbf{A}\mathbf{v}_j\|^2 + \beta_j^2\|\mathbf{v}_{j-1}\|^2 - 2\beta_j\mathbf{v}_j^T\mathbf{A}\mathbf{v}_{j-1} \tag{48}$$

and using (37)

$$\begin{aligned} \beta_j\mathbf{v}_j^T\mathbf{A}\mathbf{v}_{j-1} &= \beta_j\mathbf{v}_j^T(\beta_j\mathbf{v}_j + \alpha_{j-1}\mathbf{v}_{j-1} + \beta_{j-1}\mathbf{v}_{j-2} + \delta\mathbf{v}_{j-1}) \\ &= \beta_j^2 + \delta\beta_j \end{aligned}$$

where from (16), (34), (30), (39), (47) and (38)

$$|\delta\beta_j| \leqslant (2j-1)[3(n+6)\mu_j + (1+m\beta)\sigma]\mu_j\varepsilon.$$

This suggests, incidentally, that a small β_j can result in β_j and $\mathbf{v}_j^T\mathbf{A}\mathbf{v}_{j-1}$ being significantly different (see Paige, 1972)

The above results combine with (36) to give

$$\|\mathbf{u}_j\|^2 = \|\mathbf{A}\mathbf{v}_j\|^2 + \beta_j^2(\|\mathbf{v}_{j-1}\|^2 - 2) + \delta\beta_j' \tag{49}$$

$$|\delta\beta_j'| \leqslant \{4j(1+m\beta)\sigma + [(2j-1)6(n+6)+4]\mu_j\}\mu_j\varepsilon. \tag{50}$$

Now let

$$\mu \equiv \max(\mu_j, \sigma);$$

if $\mu = \sigma$ then (28) holds, otherwise $\mu = \mu_j$ and (49) gives with (16)

$$\|\mathbf{u}_j\|^2 \leqslant \sigma^2+4j[7+m\beta+3(n+4)]\mu^2\varepsilon$$

Clearly this bound holds for $\|\mathbf{u}_i\|^2$, $i = 1, 2, \ldots, j$, and so it also holds for $\mu_j^2 = \mu^2$, and then

$$\mu^2 \leqslant \sigma^2\{1+4j[7+m\beta+3(n+4)]\varepsilon\}$$

so that, on taking the square root, (28) is seen to hold in this case as well. Thus (28) may be combined with (38) to give (17), with (39) to give (18), and with (45) to give (23), with the proviso of course that

$$2j[7+m\beta+3(n+4)]\varepsilon \leqslant 1. \tag{51}$$

The only equation that has not yet been proved is (19), and to do this we combine (33), (35), (49) and (28), to give

$$|\beta_{j+1}^2\|\mathbf{v}_{j+1}\|^2-2\beta_{j+1}\mathbf{v}_{j+1}^T\delta\mathbf{w}_j'+\|\delta\mathbf{w}_j'\|^2+\alpha_j^2+\beta_j^2(2-\|\mathbf{v}_{j-1}\|^2)-\|\mathbf{A}\mathbf{v}_j\|^2-\delta\beta_j'| \leqslant (3n+10)\varepsilon\sigma^2$$

and (19) follows immediately on applying the bounds in (16), (34), (35) and (50).

All the stated rounding error results in (15) to (23) have now been obtained, and so the theorem has been proved.

3. Discussion

It was pointed out by Dr J. H. Wilkinson (personal communication) that algorithm A1 here is really the modified Gram–Schmidt approach to computing the Lanczos vectors, whereas the more usual approach, as in A2, is the classical Gram–Schmidt approach. Since $\mathbf{A}\mathbf{v}_j$ is only orthogonalized against two previous vectors $\mathbf{v}_j$ and $\mathbf{v}_{j-1}$, it is not surprising that there is no great difference in the results. However it is interesting to note that the bounds for A1 are definitely superior to those for A2; in fact comparing (24), (25) with (18) and (23) we see that an extra factor of j occurs in some of the bounds for A2. This suggests that A1 might be superior to A2 on all counts, although the computations carried out by Paige (1972) did not indicate any significant numerical difference between the two when used for the eigenproblem, even with $j = 600$ in one simple case.

4. Conclusion

The purpose of this paper has been to present the results (15) to (23) for the rounding error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix. This makes a deliberate distinction between this algorithm and its many applications, some of which have been mentioned in the introduction. Now that these basic results have been set down, they can be used to analyze the performance of the many methods which have the Lanczos algorithm as a basis. The results that are given here have been used by Paige (1971) to study the behaviour of the method that was originally proposed by Lanczos (1950) for finding eigenvalues, and have also been used to study the method proposed by Lehmann (1966) for finding eigenvalue bounds. The results have also been used to study several of the methods for solving solutions of equations that are based on the Lanczos algorithm. It is intended that these applications of the present analysis will be published separately. One interesting observation on that work is that the analysis of each application of the Lanczos algorithm made great use of (22), and it appears that equation (22), with the bounds in (23), gives the key to the performance of the Lanczos algorithm and its many applications in the presence of rounding errors.

References

GOLUB, G. & KAHAN, W. 1965 *SIAM J. Num. Anal.* **2,** 205–224.

HESTENES, M. R. & STIEFEL, E. 1952 *J. Res. natn. Bur. Stand.* **49,** 409–436.

HOUSEHOLDER, A. S. 1964 *The theory of matrices in numerical analysis.* New York: Blaisdell Publishing Company.

LANCZOS, C. 1950 *J. Res. natn. Bur. Stand.* **45,** 255–282.

LANCZOS, C. 1952 *J. Res. natn. Bur. Stand.* **49,** 33–53.

LEHMANN, N. J. 1966 *Num. Math.* **8,** 42–55.

PAIGE, C. C. 1971 *The computation of eigenvalues and eigenvectors of very large sparse matrices.* Ph.D. thesis, University of London.

PAIGE, C. C. 1972 *J. Inst. Maths Applics* **10,** 373–381.

PAIGE, C. C. 1974 *SIAM J. Num. Anal.* **11,** 197–209.

PAIGE, C. C. & SAUNDERS, M. A. 1975 *SIAM J. Num. Anal.* **12,** 617–629.

REID, J. K. (Ed.) 1971 *Proc. Conference on large sparse sets of linear equations.* New York: Academic Press.

REID, J. K. 1972 *SIAM J. Num. Anal.* **9,** 325–332.

WILKINSON, J. H. 1963 *Rounding errors in algebraic processes.* London: H.M.S.O.

WILKINSON, J. H. 1965 *The algebraic eigenvalue problem.* Oxford: Clarendon Press.