THE COMPUTATION OF EIGENVALUES AND EIGENVECTORS OF VERY LARGE SPARSE MATRICES

by

Christopher Conway Paige, B.Sc., B.E., Dip.N.A.
London University Institute of Computer Science

Thesis submitted for the degree of
Doctor of Philosophy
University of London

April 1971

Dedication

To Françoise

Acknowledgments

The author is very grateful to his supervisor Dr. M.J.M. Bernal for his thoughtful guidance and encouragement. He would also like to thank his friends at the Institute of Computer Science for their help and discussions, and in particular he would like to thank Christine Fair, Mary Della-Valle, and Mrs. M. McCluskey for the excellence of their typing.

2012 Addendum

Chris Paige is also very grateful to Ivo Panayotov for LaTeXing the original thesis during 2011–2012 in order to provide this much improved version. He corrected errors and improved the format. Some extra ‘newpage’ commands have now been entered so that the pages of this version roughly correspond to those of the original.

Abstract

Several methods are available for computing eigenvalues and eigenvectors of large sparse matrices, but as yet no outstandingly good algorithm is generally known. For the symmetric matrix case one of the most elegant algorithms theoretically is the method of minimized iterations developed by Lanczos in 1950. This method reduces the original matrix to tri-diagonal form from which the eigensystem can easily be found. The method can be used iteratively, and here the convergence properties and different possible eigenvalue intervals are first considered assuming infinite precision computation. Next rounding error analyses are given for the method both with and without re-orthogonalization. It is shown that the method has been unjustly neglected, in fact a particular computation algorithm for the method without re-orthogonalization is shown to have remarkably good error properties. As well as this the algorithm is very fast and can be programmed to require very little store compared with other comparable methods, and this suggests that this variant of the Lanczos process is likely to become an extremely useful algorithm for finding several extreme eigenvalues, and their eigenvectors if needed, of very large sparse symmetric matrices.

Contents

Notation

Unless otherwise stated upper case and subscripted lower case Latin letters will represent matrices and their elements, thus a_{ij} is the (i, j) element, and a_j the j-th column of the matrix A. One exception is e_j which represents the j-th column of the identity I. Unsubscripted lower case Latin letters usually represent scalars, as do small Greek letters. However a Latin letter immediately preceded by δ will represent a small quantity.

A norm without a subscript always represents the 2-norm.

There are occasional inconsistencies between main sections, for instance early in the thesis the subscript E is used to represent the Frobenius norm, and an eigenvector of the tri-diagonal matrix is denoted z_i ; y_i being used to denote an approximation to an eigenvector of A. In later sections this E is replaced by F and the roles of z_i and y_i are interchanged. This is not likely to lead to any confusion.

For convenience the scalar δ_i in Section 7 is represented by δ_i^2 in Sections 8, 9, and 10, as it is known to be non-negative in these later sections.

Whenever δ appears with two subscripts, e.g. δ_{ij}, it represents the Kronecker delta.

Section 1

Introduction

chp:1

1.1 The Problem

sec:1.1

A significant problem in computational linear algebra is finding reliable fast and accurate methods for computing some or all of the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ of a real n dimensional square matrix A, along with some of the corresponding eigenvectors. For matrices that can be wholly stored in the fast store of a digital computer this problem has been satisfactorily solved (see Wilkinson, 1965) except for a few minor points, and several excellent computational algorithms are available, particularly in the 'Handbook Series on Linear Algebra' in the journal 'Numerische Mathematik'. For larger matrices these same algorithms can be used but obvious difficulties of speed, storage, and transfer between large slow store and smaller fast store arise. However a type of large matrix that is regularly encountered in numerical work is the large sparse matrix which can be fully described in much less than n^2 words of store, in particular those matrices with only relatively few nonzero elements. With these matrices many of the methods that are satisfactory for small full matrices are uneconomic and often impossible to apply, and methods that take account of this sparsity must be sought, tested, and analysed.

1.2 Special Cases

sec1.2

For some sparse matrices satisfactory algorithms are already available, in particular matrices with narrow band form have been extensively dealt with. For instance Rutishauser (1963) and Schwarz (1968) show how the bandwidth of symmetric matrices may be reduced while preserving the band property, and Rutishauser and Schwarz (1963) give an algorithms for applying the Cholesky LR technique (see Wilkinson, 1965, pp. 544–556) directly to symmetric band matrices. As well as this, inverse iteration may be used for finding some eigenvectors and even eigenvalues of both symmetric and unsymmetric band matrices, excellent algorithms being given by Martin and Wilkinson (1967). With the above algorithms the band is considered full and any zero elements within the band cannot generally be taken into account, and so for very large matrices which are sparse within the band even these algorithms may be rather uneconomic.

Other partly satisfactory methods are available for matrices of specialized form, for example Kron's method of 'tearing' large systems (matrices) into subsystems with very few interconnections (e.g. Simpson and Tabarrok, 1968) can be quite useful in circuit analysis, while the special forms of matrices arising from the finite difference replacement of some elliptic partial differential equation problems suggest other methods of limited application. Although these methods are interesting and sometimes useful it is more important to have more general methods available, and no time will be spent on such special cases.

1.3 The General Case

sec:1.3

General methods for the matrix eigenproblem can be roughly divided for present purposes into methods which alter the matrix in a series of transformations to a more amenable form, and those methods which work with the original unchanged matrix.

1.3.1 Transformations of the Matrix

These suffer from the introduction of extra nonzero elements in the intermediate stages and so most such methods are at present limited for large sparse matrices. However much work is being done on the solution of large sparse matrix equations (e.g. IBM Symposium, 1968) (IMA Conference, 1970) and here great advantage can be taken of sparsity, for instance a matrix A having only 5% nonzero elements may have a decomposition into lower and upper triangular matrices with a total number of nonzero elements less than twice that of A, and with careful programming very little more than these nonzero elements need be stored and used. This immediately suggests the use of such techniques in inverse iteration for eigenvectors and eigenvalues of large sparse matrices. Even more exciting is the thought that some direct similarity transformation methods of reduction to tridiagonal or Hessenberg form may be able to preserve and take full advantage of sparsity throughout the computation, (Tewarson, 1970). In fact in the discussion at the I.B.M. symposium (1968, p. 163) it was suggested by Tinney that a technique for using the QR algorithm on sparse matrices was already being developed. Such an approach thus suggests a possibly very useful line of research which should definitely be examined, although until thorough rounding error analyses have been carried out the validity of these methods for either solving equations or eigenproblems will be in some doubt. Such methods are not considered here.

1.3.2 Preservation of the Matrix

Methods which leave the matrix unchanged are more immediately appealing and easily implemented. The basic use of the matrix in these methods is, on being given a vector v, to form the matrix-vector product Av. Often it is not even necessary to store the nonzero elements of A, as these can be generated as needed, and all that is needed is a simple procedure for forming this matrix-vector product. The appeal of such methods as regards storage and time per step is obvious, also they are quite general and in fact such methods are often suitable for finding approximate eigensolutions of more general linear operators than matrices. For the above reasons this class of methods was chosen for closer study, but even then the possibilities are so great as to preclude a detailed study of the whole class. These matrix-vector product methods can either be essentially iterative in concept, such as the power methods (see Wilkinson, 1965, p. 570), or designed to produce a simpler matrix having the same eigenvalues as the original matrix in a finite number of steps, such as the generalized Hessenberg methods (see Wilkinson, 1965, p. 377).

a) Iterative Methods

The iterative methods are designed to form sequences of vectors converging to eigenvectors of the matrix A, usually corresponding to extreme eigenvalues of A. Direct iteration with a single vector is the most familiar of these methods, and here convergence is hopefully to the eigenvector corresponding to the dominant eigenvalue, deflation could then possibly be used to find the next dominant pair. The trouble with this method is that it is slow, restricted to very few extreme eigenvalues, and if more than one is wanted it is cumbersome and even inaccurate for close eigenvalues. Useful generalizations of this which iterate with several vectors were developed by F. L. Bauer in 1957 and 1958, these are treppen-iteration and its orthogonal variant, and bi-iteration (see, for example, Wilkinson, 1965, pp. 602–614). A very effective method of accelerating the convergence of this type of technique was examined by Laasonen (1959) whereby a smaller eigenvalue problem is solved at intermediate stages. Other work continuing in this direction has been by Jennings (1967), Clint and Jennings (1970), Rutishauser (1969), and Stewart (1969), and it is clear that this is a very useful method for finding some extreme eigenvalues and eigenvectors of both symmetric and unsymmetric matrices with good accuracy in reasonable time. The obvious drawbacks of such methods are the uncertainty as to how many vectors will be needed and the difficulty of storing several vectors for very large matrices. The fact that only extreme eigenvalues are easily found is rarely a drawback as this is usually what is wanted. It does however appear that there have been no rounding error analyses of these later techniques.

b) Methods based on Krylov sequences

The remaining direct matrix-vector product techniques are based on the fact that if v_1 is an arbitrary vector and $v_{i+1} = Av_i$, then there will be a vector v_{m+1}, $m \leq n$, which is expressible as a linear combination of the preceding v_i. Krylov in 1931 based a method on these $v_1, \ldots, v_{m+1}$ (see, for example, Wilkinson, 1965, p. 369), while the generalized Hessenberg processes were attempts at producing more accurate methods using related sequences. The most elegant of these is the ingenious method propounded by Lanczos (1950) which is theoretically perhaps the most appealing of all possible methods for very large sparse matrices. Unfortunately it has certain practical behaviour which has led to its neglect by numerical analysts over the last ten years, although quantum chemists and others have found it very useful (e.g. Sebe and Nachamkin, 1969). In 1957 Engeli, Ginsburg, Rutishauser, and Stiefel (1959) examined different methods for solution of certain large sparse matrix problems, and one of these methods, called the cgT - method (conjugate gradient - Tchebycheff), is the Lanczos method for symmetric matrices applied in effect to the matrix $T_m(A)$ rather than to A, where $T_m(\lambda)$ is the Tchebycheff polynomial of the first kind, the purpose of this being to expand the spectrum at one end of the range. In their very thorough work Engeli et al. (1959, p. 105) reached the following conclusion –

> “For the computation of some eigenvalues at the lower end of the spectrum of a positive definite, symmetric matrix, of all methods treated here, the cgT - method is superior to such an extent that the complexity of the method is fully compensated for and it can by highly recommended.”

> They are of course referring to large sparse matrices.

Because of the beautiful simplicity of the Lanczos method, the work of Engeli et al., and initial computer runs, it was decided here to examine the Lanczos process in detail in order that it may be more fully understood and so used with expertise and confidence.

It will by now be obvious that the reader is expected to have some familiarity with the comprehensive and authoritative work of Wilkinson (1965), from which much of the work here has been developed. Many older methods are described in more detail by Fadeev and Fadeeva (1963), (in this text the Lanczos process is denoted by his own description as 'the method of minimal iterations'), while the very concise and excellent work of Householder (1964) analyzes these methods and others showing clearly their theoretical interrelations.

1.4 Outline of the Thesis

sec:1.4

The purpose of this thesis will not be to introduce new methods, but to concentrate mainly on the Lanczos method for real symmetric matrices with the intention of showing that it has been unjustly discarded and has a lot to offer in practice. Rounding error analysis will be the main tool used for this purpose and so Section 2 will present the necessary theory which will largely be a reiteration of some of Wilkinson's work (1963, 1965), but with an additional development and terminology designed to facilitate the analyses. Comments by Dr. Wilkinson rightly indicated that the error analysis of the Lanczos process would be no simple matter and so this is approached indirectly by first analyzing in Section 3 the more straightforward generalized Hessenberg processes as described by Wilkinson (1965, pp. 377–378). Sections 4 and 5 then gather together a lot of important theory on the Lanczos process and present and extend some of the relevant work of Lehmann (1963, 1966). The results of the error analysis of the symmetric Lanczos process with re-orthogonalization, and some practical results showing a definite case for the use of this process in favour of Householder's method for some eigenvalue problems of fairly large matrices are given in Section 6. Then in Sections 7, 8 and 9 an analysis of the symmetric Lanczos process without re-orthogonalization is given in an attempt to explain some remarkable computational results. Finally Section 10 sums up the more important results of the thesis.

All the rounding error analyses are believed to be original, as is the small development of rounding error analysis technique in Section 2 and the extension of Lehmann's work in Section 5. The sensitivity analysis of Hermitian matrices included at the back of the thesis is also believed to contain several new results. Although this analysis turned out not to be essential for the remainder of the work it was developed in an attempt to explain the behaviour of the Lanczos algorithm, and since it contains some useful results in its own right it is included here.

Section 2

Rounding Error Analysis Techniques

chp:2

Since this thesis contains several analyses, the techniques and terminology to be used will be summarized here along with the analyses of the more basic computations, such as forming inner products of two vectors. The characteristics of the particular computer used will then be given.

2.1 Basic Theory

sec:2.1

The analysis will be for computations carried out in normalized floating point base-B arithmetic on a computer with a double precision accumulator, and the techniques developed by Wilkinson (1963) will be used. Thus if $*$ denotes any of the four arithmetic operations $+ - \times /$, then $a = fl(b * c)$ will imply a, b and c are floating point computer numbers and a is obtained from b and c using the appropriate floating point operation. If the computer numbers have a significand represented by t base-B digits and a sign, then

$$fl(b * c) = (b * c)(1 + \epsilon_1),\ |\epsilon_1| \le u, \tag{2.1}$$

eq:2.1

$$\text{where} \begin{cases} u = \frac{1}{2}B^{1-t} & \text{for rounding by adding,} \\ u = B^{1-t} & \text{for chopping and forcing.} \end{cases}$$

Sometimes it will be assumed that a facility is provided for accumulating double length numbers g and h in double precision, giving a double length number

$$g(1+\eta_1) + h(1+\eta_2), \quad |\eta_1|, |\eta_2| \leq u' \tag{2.2}$$

eq:2.2

$$\text{where} \begin{cases} u' = \frac{1}{2}(B+1)B^{-2t} & \text{for rounding by adding,} \\ u' = (B+1)B^{-2t} & \text{for chopping and forcing,} \end{cases}$$

while the notation $fl_2(g+h+\ldots)$ will apply when several such numbers are accumulated in double precision and then rounded to single length.

2.2 A Simplified Notation

sec:2.2

The analyses will involve many products and quotients of factors $(1+\epsilon_i)$ like the one in (2.1), and to prevent the analyses and the equations from becoming too cumbersome a new notation and the rules for its use in the error analyses will be introduced, based on the following theorem.

thm:2.1 **Theorem 2.1.**

$$1-(p+q)u \leq \frac{\prod_{i=1}^{p}(1+\epsilon_i)}{\prod_{i=1}^{q}(1+\epsilon'_i)} \leq 1+1.01(p+q)u$$

where $|\epsilon_i|, |\epsilon'_i| \leq u$*,* $0 \leq u \leq 0.001$*, and* p *and* q *are non-negative integers such that* $(p+q)u \leq 0.01$*.*

Proof. Following Forsythe and Moler (1967, p. 91)

$$1-pu \leq (1-u)^p.$$

Now

$$(1-u)(1+u) \leq 1$$

so

$$1 - pu \leq (1-u)^p \leq \frac{1}{(1+u)^p}.$$

Next since $1 + u \leq e^u$, and $e^y \leq 1 + 1.006y$ for $0 \leq y \leq .01$, it follows that $(1+u)^p \leq e^{pu} \leq 1 + 1.006pu$. Then $(1-u)^p/(1+u)^q \geq (1-pu)(1-qu) \geq 1-(p+q)u$, and since $u \leq 0.001$

$$\begin{aligned}\frac{(1+u)^p}{(1-u)^q} = \frac{(1+u)^{p+q}}{(1-u^2)^q} \leq \frac{(1+u)^{p+q}}{1-qu^2} &= (1+u)^{p+q}\left(1 + \frac{qu^2}{1-qu^2}\right) \\ &\leq [1 + 1.006(p+q)u][1 + 0.0011qu] \\ &\leq 1 + 1.01(p+q)u,\end{aligned}$$

and the result follows. □

It will be assumed in all that follows that the limitations imposed for the theorem are observed, since that on the unit error u in (2.1) will ordinarily be observed, while the constraint involving p, q, and u still allows the consideration of extremely large matrices on modern computers. With this result the analyses can be considerably simplified by using the following notational convention.

Convention: in these analyses unless otherwise indicated α, ϵ, ζ and η (always without subscripts) will represent real numbers satisfying

$$\left.\begin{array}{ll} |\alpha - 1| \leq u, & |\epsilon| \leq (1.01)u, \\ |\zeta - 1| \leq u' & |\eta| \leq (1.01)u', \end{array}\right\} \qquad (2.3) \quad \boxed{\texttt{eq:2.3}}$$

where u and u' are constants for the particular computer, as given in (2.1) and (2.2). $D(\alpha)$ etc. will represent diagonal matrices whose (not necessarily equal) elements satisfy the above bounds. Thus if $|\epsilon_i| \leq u$ there exists a value α such that

$$|\alpha - 1| \leq u, \quad \prod_{i=1}^{p}(1+\epsilon_i) = \alpha^p,$$

and such product terms will conveniently be represented by this power notation. Now since any one of these Greek letters may appear several times in the one equation representing different numbers, it will be necessary to specify clearly the possible rules for their manipulation. From (2.3) and Theorem 2.1 it will be obvious that if $p, q \geq 0$ are integers such that $(p+q)u \leq 0.01$ and x, y and z are real numbers then the following hold

$$\left.\begin{aligned} &\alpha^p \cdot \alpha^q = \alpha^{p+q} \\ &x = \alpha(y+z) \Rightarrow x = \alpha y + \alpha z \\ &\left.\begin{aligned} &x = (\alpha^p/\alpha^q)y \\ &x = \alpha^p \cdot \alpha^q y \end{aligned}\right\} \Rightarrow x = [1+(p+q)\epsilon]y = y + (p+q)\epsilon y \\ &\text{where } |\epsilon y| \leq (1.01)u|y|, \end{aligned}\right\} \tag{2.4}$$

eq:2.4

and the same sort of rules hold for ζ and η.

Note that the implications are not necessarily true in reverse since for example the α's in $\alpha y + \alpha z$ may be different. Note also that since the analyses will always go from expressions involving α's to those involving ϵ's and finally to bounds involving u, it will be notationally convenient to replace α^p/α^q by α^{p+q}, the possible small error so induced disappearing when this is replaced by $[1+(p+q)\epsilon]$. Finally whenever α, ϵ, ζ, or η appear on the right hand side of an inequality they will represent their upper bounds, that is $1+u$, $(1.01)u$, $1+u'$, and $(1.01)u'$ respectively.

2.3 Analyses of some Basic Computations

sec:2.3

Vector inner-products will frequently be required so, using the notational convention, if v and w are n dimensional computer vectors with components v_i, w_i, $i = 1, 2, \ldots, n$, then

$$fl(v^Tw) = v_1w_1\alpha^n + \sum_{i=2}^{n} v_iw_i\alpha^{n+2-i}$$

$$\left.\begin{aligned} &= v^TD(\alpha^n)w \\ &= v^TD(1+n\epsilon)w \\ &= v^Tw + n\epsilon|v^T||w|, \\ \text{while } fl(v^Tv) &= \alpha^n v^Tv, \end{aligned}\right\} \qquad (2.5) \quad \boxed{\texttt{eq:2.5}}$$

and for double length accumulation of vector inner-products

$$fl_2(v^Tw) = \left(v_1w_1\zeta^{n-1} + \sum_{i=2}^{n} v_iw_i\zeta^{n+1-i}\right)\alpha$$

$$\left.\begin{aligned} &= \alpha v^Tw + \alpha(n-1)\eta|v^T||w| \\ &= \alpha\zeta^{n-1}v^Tv, \quad \text{if } w = v. \end{aligned}\right\} \qquad (2.6) \quad \boxed{\texttt{eq:2.6}}$$

The basic computation involved in the methods to be studied is the product of a large sparse n by n matrix A with a vector. Suppose A has at most m non-zero elements per row, then following (2.5)

$$fl(Av) = (A + \delta A)v \qquad (2.7) \quad \boxed{\texttt{eq:2.7}}$$

where

$$|\delta A| \leq m\epsilon|A|.$$

Now $\|\delta A\|_2 \leq \||\delta A|\|_2 \leq m\epsilon\||A|\|_2 \leq m\epsilon\|A\|_E$, but since for many large sparse matrices $\|A\|_E$ is a gross over-bound for $\||A|\|_2$, it will be assumed that

$$\||A|\|_2 = \beta\|A\|_2$$

so that

$$\|\delta A\|_2 \leq m\epsilon\beta\|A\|_2 \qquad (2.9) \quad \boxed{\texttt{eq:2.8}}$$

where a bound can be found on β; clearly $\beta \leq n^{\frac{1}{2}}$.

2.4 The Atlas Computer & Timing of Algorithms

sec:2.4

The computer used in all the computations here is an I.C.T. Atlas computer, this is a floating point 13 bit octal machine with a double precision accumulator and rounding by forcing for single precision. For double precision accumulation both forcing and chopping are used. Thus in (2.1) and (2.2)

$$\begin{aligned} &B = 8, \quad t = 13 \\ &u = 8^{-12} = 2^{-36} \doteq 10^{-10.84} \\ &u' = 9 \times 8^{-26} \end{aligned}$$

so that the computer stores at least ten decimal figures of a number accurately.

Nearly every multiplication in a matrix algorithm is accompanied by an addition, and vice versa, and so in order to compare algorithms an ‘operation count’ will be defined as the number of multiplications with additions in the algorithm. s will denote the time in microseconds for adding the single-length product of two single-length numbers to a single-length number, while d will denote the time for adding the double-length product of two single-length numbers to a double-length number. For a computer with a double-precision accumulator d will be fairly reasonable as the double-length product will be already available and the addition is simplified too.

On Atlas the operation $c := c + a \times b$, all in single-length, can be roughly described as follows (I.C.T. ABL Manual, 1965; and Fairbourn, 1965)

Accumulator Code	Approximate Time μ Sec	Function
324	2	acc:= a
362	7	acc:= acc$\times b$
320	2	acc:= acc$+c$
356	2	c:= acc

so that $s = 13\mu$ sec. If c represents a double-length number then one way of performing the double-length addition of the product of the single-length numbers a and b

may be roughly described as

Code or Extracode	Approximate Time μ Sec	Function
324	2	acc:= a
342	7	acc:= acc$\times b$
1500	29	acc:= acc$+c$
1556	10	c:= acc

so that $d = 48$ μSec. In fact if $\sum_{i=1}^{n} a_i b_i$ is to be accumulated in double-length then an even faster means, extracode 1437, is available. This last takes about $30n$ μSec for the complete inner-product, compared with $13n$ μSec for ordinary single-length accumulation, so that at the worst $d < 4s$ while $d < 2\frac{1}{2}s$ is possible.

Section 3

The Generalized Hessenberg Processes

chp:3

Although the main purpose of this work will be to examine the Lanczos process for the symmetric matrix eigenproblem, this may be classified as one of the generalized Hessenberg processes (see, for example, Wilkinson, 1965, pp. 377–395) and it will be advantageous to examine these processes as a group. Later in this section it will be shown how the Lanczos process differs significantly in its application from the other generalized Hessenberg processes, and while this difference makes the former more important for large sparse matrices it unfortunately makes the error analysis of the Lanczos process substantially more difficult. The error analyses do have a lot in common and so an error analysis for the generalized Hessenberg processes other than the Lanczos process will be developed both for its own particular interest and as a step towards the analysis of the Lanczos process.

3.1 Basic Theory

sec:3.1

Given an n by n matrix A, each of the generalized Hessenberg processes requires a set of linearly independent vectors, $w_1, w_2, \ldots, w_n$ and an arbitrary starting vector v_1, and forms a series of vectors v_j satisfying

$$h_{j+1,j}v_{j+1} = Av_j - \sum_{i=1}^{j} h_{ij}v_i, \quad j = 1, 2, \ldots, k_0, \tag{3.1}$$

eq:3.1

where the scalars h_{ij}, $i = 1, 2, \ldots, j$, have been chosen so that v_{j+1} is orthogonal to $w_1, w_2, \ldots, w_j$,

$$\left.\begin{aligned} h_{1j} &= w_1^T Av_j / w_1^T v_1, \\ h_{ij} &= \left(w_i^T Av_j - \textstyle\sum_{r=1}^{i-1} h_{rj} w_i^T v_r \right) / w_i^T v_i, \quad i = 2, 3, \ldots, j, \end{aligned}\right\} \tag{3.2}$$

eq:3.2

and $h_{j+1,j}$ is an arbitrary normalizing factor.

Only k_0 steps are indicated as the theoretical process can be considered complete for the first vector $v_{k_0+1} = 0$. Clearly $k_0 \leq n$ otherwise there would be a non-zero vector orthogonal to n linearly independent vectors w_i. If

$$w_i^T v_i = 0, \quad i \leq k_0 \tag{3.3}$$

eq:3.3

then (3.2) is of no use, the method breaks down, and a new initial vector v_1 must be chosen. It will be assumed that this does not occur.

To examine the situation after $k \leq k_0$ steps the two n by k matrices $V = [v_1, v_2, \ldots, v_k]$ and $W = [w_1, w_2, \ldots, w_k]$ can be considered, then

$$W^T V = L \tag{3.4}$$

eq:3.4

where L is a non-singular k by k lower triangular matrix so that V has linearly independent columns.

The matrix form of the set of equations (3.1) is

$$AV = VH + E \tag{3.5}$$

eq:3.5

where $E = [0, \ldots, 0, v_{k+1}h_{k+1,k}]$ and H is the k by k upper Hessenberg matrix of elements h_{ij}. In particular if $k = k_0$ then E is the null matrix and to every eigenvalue μ of H with eigenvector z there corresponds an eigenvalue μ of A with eigenvector Vz. The problem of finding the (partial) eigensolution of A has then been reduced to that of finding the eigensolution for the Hessenberg matrix H. The extension of the process to the complete eigensolution of A if $k_0 < n$ is straightforward (see, for example, Wilkinson, 1965, p. 378) and need not be repeated here.

The different members of the class of generalized Hessenberg methods depend on the choice of the linearly independent vectors w_i; for example in his own method Hessenberg chose $w_i = e_i$, the i^{th} column of the identity matrix. Another choice is $w_i = v_i$ which is the method of Arnoldi (1951), while if the w_i are derived from A^T in the same way as the v_i are derived from A then this is the Lanczos process (1950). Only the symmetric variant of the Lanczos process will be considered in this thesis, and this has $A^T = A$ so again $w_i = v_i$ making it theoretically equivalent to Arnoldi's method for a symmetric matrix. Now from (3.4), for these two methods

$$V^TV = L,$$

which must be diagonal since it is both lower triangular and symmetric. That is the v_i form an orthogonal set and from (3.5)

$$V^TAV = V^TVH + V^TE = LH, \tag{3.6}$$

eq:3.6

which is symmetric and of Hessenberg form, so that H is tri-diagonal and (3.1) becomes a three term recurrence relation. It is the tremendous saving in computation produced by this simple recurrence with only two coefficients h_{jj} and $h_{j-1,j}$ to be computed at each step that makes the Lanczos process so fast for very large sparse matrices. As well as this only the two previous vectors need be held in the fast store at each step and the resulting matrix H is an easy to handle tri-diagonal matrix.

Unfortunately when rounding errors are present the vectors v_i will not necessarily form an orthogonal set and the computation using the full expansions (3.1) and (3.2) with $w_i = v_i$ will give a significantly different result from that using

$$\left.\begin{aligned} h_{j+1,j}v_{j+1} &= Av_j - h_{jj}v_j - h_{j-1,j}v_{j-1} \\ \text{with} \quad h_{jj} &= v_j^T A v_j / v_j^T v_j, \quad h_{j-1,j} = v_{j-1}^T A v_j / v_{j-1}^T v_{j-1}. \end{aligned}\right\} \qquad (3.7)$$

eq:3.7

For convenience the former method will be referred to as Arnoldi's method, the latter being of course Lanczos' method. Because the Lanczos method assumes the omitted terms are zero, it turns out that the error analysis is in fact more difficult than that of the other generalized Hessenberg processes. Here an analysis will be given for the process described by (3.1) and (3.2) where nothing will be assumed of the w_i other than linear independence; this analysis, with slight modifications, will then hold for all these methods apart from that of Lanczos. This will be a stepping stone to the more difficult analysis and will also provide familiarity with the notational 'shorthand' introduced in Section 2.

3.2 Error Analysis

sec:3.2

The computation in the j-th step can be considered in three parts – computing Av_j, calculating the h_{ij} for $i = 1, 2, \ldots, j$, and forming and normalizing the next vector v_{j+1}. The same notation will be used for computed components as was used in Section 3.1, as at no stage will a comparison of computed and theoretical components be needed.

With the assumptions made in Section 2 it follows from (2.7) and (2.9) that

$$fl(Av_j) = (A + \delta A_j)v_j, \quad \|\delta A_j\|_2 \le m\beta\epsilon\|A\|_2. \qquad (3.8)$$

eq:3.8

Ordinary fl arithmetic will be assumed for the formation of vector inner-products, although the analysis using fl_2 is no more difficult and is given by Paige (1969a). The computation of (3.2) is not fully defined until the order of operations is given, and it will always be assumed that such right hand sides are evaluated from left to right and with the index increasing in the sum, then with the notation introduced in Section 2.2 and using (2.5),

$$h_{1j} = \frac{\alpha w_1^T D(\alpha^n)(A + \delta A_j)v_j}{w_1^T D(\alpha^n)v_1}$$

$$\text{or} \quad w_1^T D(\alpha^n)(A + \delta A_j)v_j = h_{1j} w_1^T D(\alpha^{n+1})v_1$$

remembering that the α's are not necessarily equal, that $D(\alpha^n)$ is a diagonal matrix with different elements all of which may be represented by α^n, and that $1/\alpha$ may be represented by α at this stage and vice versa. Now replacing $D(\alpha^n)$ by $D(1 + n\epsilon)$ etc. gives

$$w_1^T A v_j - h_{1j} w_1^T v_1 = f_{1j}, \tag{3.9}$$

eq:3.9

with f_{1j} satisfying

$$f_{1j} = (n+1) w_1^T \left[h_{1j} D(\epsilon) v_1 - D(\epsilon)(A + \delta A_j)v_j\right] - w_1^T \delta A_j v_j.$$

In a similar manner

$$h_{ij} = \frac{\alpha \left[\alpha^{i-1} w_i^T D(\alpha^n)(A + \delta A_j)v_j - \sum_{r=1}^{i-1} \alpha^{i-r+1} h_{rj} w_i^T D(\alpha^n) v_r\right]}{w_i^T D(\alpha^n) v_i},$$

or

$$w_i^T D(\alpha^{n+i-1})(A + \delta A_j)v_j = \sum_{r=1}^{j} h_{rj} w_i^T D(\alpha^{n+i-r+1}) v_r,$$

giving

$$w_i^T A v_j - \sum_{r=1}^{i} h_{rj} w_i^T v_r = f_{ij}, \quad i = 2, 3 \ldots, j, \tag{3.10}$$

eq:3.10

with f_{ij} satisfying

$$f_{ij} = (n+i) w_i^T \left[\sum_{r=1}^{i} h_{rj} D(\epsilon) v_r - D(\epsilon)(A + \delta A_j)v_j\right] - w_i^T \delta A_j v_j.$$

The elements f_{ij} so far defined may be regarded as the upper triangular elements of the full matrix F defined by

$$W^T AV - LH = F \tag{3.11}$$ $\boxed{\texttt{eq:3.11}}$

where L is the lower triangular matrix with non-zero elements $l_{ir} = w_i^T v_r$, $r = 1, 2, \ldots, i$. Equation (3.11) thus describes the errors introduced by the computation of the elements of H using (3.2).

In order to form v_{j+1} the right hand side of (3.1) is computed from left to right and then each element divided by the chosen normalizing factor $h_{j+1,j}$, giving

$$h_{j+1,j} D(\alpha^2) v_{j+1} = D(\alpha^{j-1})(A + \delta A_j) v_j - \sum_{i=1}^{j} h_{ij} D(\alpha^{j-i+1}) v_i \tag{3.12}$$ $\boxed{\texttt{eq:3.12}}$

which for $j = 1, 2, \ldots, k$ may be re-written in matrix notation as

$$AV = VH + E + \delta V \tag{3.13}$$ $\boxed{\texttt{eq:3.13}}$

in analogy with (3.5). Here δV is a matrix with columns satisfying

$$\delta v_j = j \left[\sum_{i=1}^{j+1} h_{ij} D(\epsilon) v_i - D(\epsilon)(A + \delta A_j) v_j \right] - \delta A_j v_j.$$

The rounding errors introduced by the computation have now been described, with their bounds being given implicitly by the notational convention used. It now remains to find a bound on the effect of these rounding errors on the eigenvalues. This can be done by considering the departure from orthogonality of v_{j+1} to $w_1, \ldots, w_j$, as a result of which (3.4) no longer holds, instead

$$W^T V = L + U, \quad W^T v_{k+1} = u_{k+1} \tag{3.14}$$ $\boxed{\texttt{eq:3.14}}$

where U is a strictly upper triangular matrix, and U and the vector u_{k+1} need to be expressed in terms of the rounding errors.

First defining the two k by k upper triangular matrices

$$\bar{U} = [u_2, u_3, \ldots, u_{k+1}], \qquad \bar{H} = \begin{bmatrix} h_{21} & h_{22} & \ldots & h_{2k} \\ & h_{32} & \ldots & h_{3k} \\ & & \ddots & \vdots \\ & & & h_{k+1,k} \end{bmatrix},$$

where $u_1, \ldots, u_k$ are the columns of U, it follows from (3.11), (3.13) and (3.14) that

$$F = W^T AV - LH = UH + W^T E + W^T \delta V = \bar{U}\bar{H} + W^T \delta V. \tag{3.15}$$

eq:3.15

Now since $\bar{U}\bar{H} = F - W^T \delta V$ is upper triangular it can be seen from the expressions for F and δV that this can be bounded in terms of H, V, W, A, and ϵ. Then while $h_{j+1,j} \neq 0$

$$\bar{U} = \left(F - W^T \delta V\right) \bar{H}^{-1}. \tag{3.16}$$

eq:3.16

This useful result shows how the departure from the required orthogonality, represented by $\bar{U}$, is related to the elements of H. This error can be considered in two parts, that in brackets in (3.16) which represents the rounding errors introduced in each step, and $\bar{H}^{-1}$ which shows how the effect on later steps of previous errors is magnified. To examine the first component it follows from the expressions for f_{ij} and δv_j that for $i \leq j$

$$f_{ij} - w_i^T \delta v_j = (n + i + j) w_i^T \left[\sum_{r=1}^{j+1} h_{rj} D(\epsilon) v_r - D(\epsilon)(A + \delta A_j) v_j \right].$$

Note in passing that the error in computing Av_j has at most a second order effect on orthogonality. However no *a priori* bound may be found on the above expression because of its dependence on the elements of H, and from (3.2) these may be large if any pair w_i and v_i are nearly orthogonal – and any method keeping $w_i^T v_i$ relatively large will prosper accordingly, for example Arnoldi's method with $w_i = v_i$.

The magnifying factor in (3.16) may have very large elements even when none of the normalizing factors (chosen for example to give $\|V_j\|_2 = 1$) are very small; this can occur if there are several instances of mild cancellation, and explains why even Arnoldi's method is likely to be numerically unstable.

An indication of the effect of the loss of orthogonality U on the eigenvalue problem can easily be shown for the case where a complete solution is obtained ($k = k_0 = n$). Here (3.13) may be re-written, if V is non-singular,

$$H = V^{-1}(A + \delta A)V, \quad \delta A \equiv -(E + \delta V)V^{-1}, \tag{3.17}$$

eq:3.17

with the non-zero column of E given by

$$h_{n+1,n}v_{n+1} = (W^T)^{-1}(f_n - Uh_n) - \delta v_n$$

from the last column of (3.15), h_n being the last column of H. Thus in the spirit of reverse error analysis H has the same eigenvalues as the perturbed matrix $A + \delta A$. Clearly δA cannot in general be bounded *a priori* and may in fact be quite large because of large U, or large H, or both. The special form of E has considerable significance, but a discussion of this will be left till later.

At this point it is worthwhile digressing and examining Hessenberg's method more closely. Since $w_i = e_i$ in (3.2) the inner-products $w_i^T v_r$ introduce no error, and by making the first j elements of v_{j+1} zero, orthogonality is automatically obtained, there is then no magnified effect of earlier errors on later steps. The only worry then is that large elements of H will lead to large rounding errors being introduced at each step. Fortunately the elements of H can be kept to a reasonable size by taking the w_i to be the columns of the identity matrix in an order which ensures that $w_i^T v_i$ is as large as possible in (3.2), this is equivalent to the use of interchanges in the direct reduction to Hessenberg form (see Wilkinson, 1965, p. 357).

3.3 Re-orthogonalization

sec:3.3

For methods other than that of Hessenberg, re-orthogonalization is usually required to maintain orthogonality, and double length accumulation of vector products is often used for increased accuracy. In step j the elements of H are computed as previously from (3.2), but an intermediate vector c_j is then formed

$$c_j = Av_j - \sum_{i=1}^{j} h_{ij} v_i \tag{3.18}$$

eq:3.18

and v_{j+1} is found from this by re-orthogonalization

$$h_{j+1,j} v_{j+1} = -\sum_{r=1}^{j} b_{rj} v_r + c_j, \tag{3.19}$$

eq:3.19

where since the b_{rj} will usually be very small this order of computation will probably reduce errors. The coefficients b_{ij} are computed from

$$\left.\begin{aligned} b_{1j} &= w_1^T c_j / w_1^T v_1, \\ b_{ij} &= \left(w_i^T c_j - \textstyle\sum_{r=1}^{i-1} b_{rj} w_i^T v_r \right) / w_i^T v_i, \quad i = 2, 3, \ldots, j. \end{aligned}\right\} \tag{3.20}$$

eq:3.20

A rounding error analysis of this general process is given by Paige (1969a). The analysis uses the same approach as that for the process without re-orthogonalization, and that is to describe the errors introduced in each computation in the j-th step, put them in matrix form representing the first k steps and manipulate these matrix equations to obtain an expression for the loss of orthogonality. The analysis will not be repeated here because of its limited interest, but it can be shown that if $W^T V = L + U$ as in (3.14) then under certain restrictions on n and L

$$\frac{\|W^T v_{k+1}\|_2}{\|W^T\|_2 \|v_{k+1}\|_2} \le 2\epsilon + \frac{\left[\|U\|_2 + 2(k+2)^2 \|W\|_2 \epsilon\right]^2 \|L^{-1}\|_2^2 \|A\|_2}{|h_{k+1,k}| \cdot \|v_{k+1}\|_2}.$$

Small values of $w_i^T v_i$ lead to large L^{-1} and so it can be seen that the same factors that led to inaccuracy in the method without re-orthogonalization (i.e. near orthogonality of a pair w_i and v_i, and cancellation leading to small $h_{k+1,k}$) could cause trouble here, except that the squared term in square brackets here leaves a much greater margin of safety.

Of these generalized Hessenberg processes with reorthogonalization only the Lanczos process is ever likely to be used, and a full analysis of this will be given later. This is more complicated than for the other methods, but the same approach as was described earlier in this section can be used. Although other of the generalized Hessenberg processes without re-orthogonalization are of interest for smaller matrices, only the Lanczos process is likely to be of continuing interest for large sparse matrices and an analysis of this will be given later, based on the insight gained in this section.

Section 4

Theoretical Convergence of the Lanczos Process

chp:4 Although the Lanczos process applied to an n by n real symmetric matrix A is usually thought of as a direct method producing an n by n tri-diagonal matrix T in n steps, it is in fact far more useful as an iterative method producing after the k-th step a k by k tri-diagonal matrix T, the eigenvalues of which approximate k eigenvalues of A. In fact Lanczos intended such a use of his method as the following comment in his paper (1950, pp. 270–271) clearly shows.

“The correct eigenvalues of the matrix A are obtained by evaluating the zeros of the last polynomial $p_m(x) = 0$. What actually happens however, is that the zeros of the polynomials $p_i(x)$ do not change much from the beginning. If the dispersion is strong, then each new polynomial basically adds one more root but corrects the higher roots by only small amounts. It is thus well possible that the series of largest roots in which we are primarily interested is practically established with sufficient accuracy after a few iterations. .... The same can be said about the vibrational modes associated with these roots.”

Here the roots of the polynomial $p_i(x)$ are just the eigenvalues of the i by i matrix T after i steps.

In order to support this iterative use, the theory will be developed here to obtain expressions for the rates of convergence of the eigenvalues and eigenvectors of T with increasing k to those of A in the manner given by Kaniel (1966), where infinite precision computation is assumed. However because of the lack of clarity of Kaniel's paper and several significant errors in the working and results his basically good ideas will be developed fairly fully in a very different, and hopefully more easily digestible form.

4.1 Background Theory

sec:4.1

Let A be a real symmetric matrix on the real n dimensional space $\mathbb{R}^n$, then it has been shown in Section 3 how k steps of the error free Lanczos process starting with an arbitrary non-zero vector v_1 produce a k by k tri-diagonal matrix T such that if $k \leq k_0$

$$AV = VT + E, \quad V^T E = 0, \tag{4.1}$$

eq:4.1

where the n by k matrix $V = [v_1, v_2, \ldots, v_k]$ has non-zero orthogonal columns lying in the linear manifold, from here on to be denoted by M_k, in $\mathbb{R}^n$ spanned by the sequence of Krylov vectors $v_1, Av_1, \ldots, A^{k-1}v_1$. The matrix $E = [0, \ldots, 0, v_{k+1}t_{k+1,k}]$ has as its last column the component of Av_k orthogonal to M_k. As k_0 is the first value of k for which $E = 0$ it follows that k_0 is the maximum number of linearly independent vectors in the Krylov sequence starting with v_1. From the above it also follows that

$$T = \left(V^T V\right)^{-1} V^T A V. \tag{4.2}$$

eq:4.2

Let the eigensystem of A be such that

$$Ax_i = x_i\lambda_i, \quad x_i^T x_j = \delta_{ij}; \quad i, j = 1, 2, \ldots n, \tag{4.3}$$

eq:4.3

where δ_{ij} is the Kronecker delta. Then v_1 can be expressed

$$v_1 = \sum_{i=1}^{n} x_i \alpha_i.$$

Suppose $\lambda_1 = \lambda_2$, then for any r

$$A^r v_1 = (x_1\alpha_1 + x_2\alpha_2)\lambda_1^r + \sum_{i=3}^{n} x_i \alpha_i \lambda_i^r,$$

so that if the vectors of this Krylov sequence alone are considered, the matrix A can be thought of as acting in that subspace of $\mathbb{R}^n$ spanned by the $n-1$ vectors $x_1\alpha_1 + x_2\alpha_2, x_3, \ldots, x_n$. It is because only one vector in the space spanned by x_1, x_2

can be obtained from these Krylov vectors that coincident eigenvalues of A appear as simple eigenvalues of T (see Section 4.1.3). Likewise if $\alpha_j = 0$ then every vector in the Krylov sequence will be orthogonal to x_j, so A here will again be acting in a subspace of dimension less than n, and no combination of these Krylov vectors can produce x_j.

Thus only the components of the eigensystem of A lying in the linear manifold M_{k_0}, which is invariant with respect to A, will be found. Realizing that the total eigensystem is unobtainable in the above cases with this one Krylov sequence, it is now only necessary to examine the case where A has simple eigenvalues and $\alpha_i \neq 0$, $i = 1, 2, \ldots, n$. Note here that $k_0 = n$. In what follows the eigenvalues of A may then be ordered

$$\lambda_n < \lambda_{n-1} < \ldots < \lambda_1. \tag{4.4}$$

eq:4.4

In order to relate the k by k matrix T in (4.2) to the n by n matrix A some basic theory will now be given.

4.1.1 The Orthogonal Projection Operator

sec:4.1a

Let $y \in M_k$, where $\{v_1, v_2, \ldots, v_k\}$ is a basis for M_k, then there exists a k dimensional vector z such that $y = Vz$, and vice versa. But M_k and the real k dimensional space $\mathbb{R}^k$ are isomorphic, and z is then the representation of y in $\mathbb{R}^k$ with this basis. Let $W = [w_{k+1}, \ldots, w_n]$ be a matrix whose $n - k$ linearly independent columns are orthogonal to those of V, then $[V, W]$ is nonsingular and any $x \in \mathbb{R}^n$ may be represented as $x = Vz + Wu$ for some k vector z and $(n-k)$ vector u.

The symmetric matrix

$$P_V = V \left(V^T V\right)^{-1} V^T$$

is the orthogonal projection operator onto M_k, for using the above representation and the fact that $V^T W = 0$,

$$P_V x = V\left(V^T V\right)^{-1} V^T V z = Vz \in M_k$$

and

$$(x - P_V x)^T P_V x = (Wu)^T Vz = 0.$$

4.1.2 The Restriction of $P_V A$ to M_k

sec:4.1b

Note that $P_V A : \mathbb{R}^n \to M_k$, but the main interest will be the operation of $P_V A$ on elements of M_k. The k by k matrix

$$T = \left(V^T V\right)^{-1} V^T A V$$

is called the restriction of $P_V A$ to M_k, since if $y = Vz \in M_k$ then

$$Tz = \left(V^T V\right)^{-1} V^T A V z$$

and

$$P_V A y = V\left(V^T V\right)^{-1} V^T A V z = VTz,$$

so if z is the representation of y in $\mathbb{R}^k$ with basis $\{v_1, v_2, \ldots, v_k\}$, then Tz is the representation of $P_V A y$ with the same basis.

If M_k is invariant with respect to A then it follows that T is the restriction of A to M_k.

4.1.3 The Eigensystem of T

sec:4.1c

The n by n symmetric matrix $P_V A P_V$ will be shown to be closely related to T. Let its n orthonormal eigenvectors be y_i, with corresponding real eigenvalues μ_i, $i = 1, 2, \ldots, n$. Let $w_{k+1}, \ldots, w_n$ be any $n - k$ orthonormal vectors which are all orthogonal to v_i, $i = 1, 2, \ldots, k$, then

$$P_V A P_V w_i = 0, \quad i = k+1, \ldots, n$$

so at least $n-k$ eigenvalues $\mu_{k+1}, \ldots, \mu_n$ are zero, and these have eigenvectors $y_i = w_i$, $i = k+1, \ldots, n$. The remaining k eigenvectors, being orthogonal to these, must then lie in M_k, and so there exist vectors z_i such that

$$y_i = V z_i, \quad y_i^T y_j = z_i^T V^T V z_j = \delta_{ij}; \quad i, j = 1, \ldots, k, \tag{4.5}$$

eq:4.5

and defining $Z = (z_1, \ldots, z_k)$ this gives $Z^T V^T V Z = I$, so these z_i are linearly independent. Then

$$V z_i \mu_i = y_i \mu_i = P_V A P_V y_i = V \left(V^T V\right)^{-1} V^T A V z_i,$$

or by multiplying on the left by $\left(V^T V\right)^{-1} V^T$

$$T z_i = z_i \mu_i, \quad i = 1, 2, \ldots, k, \tag{4.6}$$

eq:4.6

so that the k eigenvectors of T correspond to the k eigenvectors of $P_V A P_V$ lying in M_k, and the corresponding eigenvalues are the same for both matrices. The larger matrix however has an extra $n-k$ zero eigenvalues.

Now in (3.6) it was shown that in the Lanczos process T is tri-diagonal and $V^T V = D$, say, is diagonal. D clearly has positive diagonal elements so

$$T = D^{-1} V^T A V = D^{-\frac{1}{2}} \left(D^{-\frac{1}{2}} V^T A V D^{-\frac{1}{2}}\right) D^{\frac{1}{2}}$$

which is similar to a symmetric matrix, thus T is quasi-symmetric. But from (3.7)

$$v_j^T A v_{j+1} = v_{j+1}^T A v_j = t_{j+1,j} v_{j+1}^T v_{j+1}$$

and $t_{j+1,j} v_{j+1} \neq 0$, $j < k_0$, so that no next to diagonal elements of $V^T A V$ or T can be zero. From these results it follows that unity and the leading principal minors of $T - \mu I$, taken in increasing order as polynomials in μ, form a Sturm sequence; but more importantly, the roots of the leading principal minor of degree r strictly separate the roots of that of degree $r+1$ (see for example, Wilkinson, 1965, p. 300).

Thus the eigenvalues of the k by k matrix T are real and distinct and can be ordered

$$\mu_k < \mu_{k-1} < \ldots < \mu_1. \tag{4.7}$$

eq:4.7

Note that this argument does not depend on the eigenvalues of A being simple, it is in fact a rigorous way of showing that repeated eigenvalues of A will not be detected by the Lanczos process.

4.1.4 Comparison of the Eigenvalues of T with those of A

sec:4.1d

The Courant-Fischer minimum-maximum characterization of eigenvalues (see for example, Wilkinson, 1965, p. 99) can be used very effectively here to obtain inequalities involving the two sets of eigenvalues. Here the eigenvalues of A will be compared with those of $P_V A P_V$ as this seems an easier approach than is usually given in the literature (see for example, Gould, 1957, p. 39), even if the execution here is slightly longer. If the n eigenvalues of $P_V A P_V$ are ordered

$$\mu_n' \leq \mu_{n-1}' \leq \cdots \leq \mu_1'$$

then from the minimum-maximum characterization, for $i = 1, 2, \ldots, n$

$$\mu_i' = \min_{w_1,\ldots,w_{i-1}} \max_{\substack{y^T w_j = 0 \\ j=1,\ldots,i-1}} \frac{y^T P_V A P_V y}{y^T y}, \qquad \text{(of course } y \neq 0\text{)}$$

where the w_j are arbitrary vectors in $\mathbb{R}^n$. A vector $y = y_i'$ giving this μ_i' is then a corresponding eigenvector. However k of these eigenvalues and their eigenvectors correspond to those of T, these eigenvectors of $P_V A P_V$ lying in M_k, while the remaining $n-k$ eigenvalues of $P_V A P_V$ are zero and have eigenvectors orthogonal to M_k. Thus if the vectors y in the above characterization are constrained to lie in M_k, then only those eigenvalues of $P_V A P_V$ corresponding to eigenvalues of T will be given, and denoting these by $\mu_1, \ldots, \mu_k$ with the ordering given in (4.7), it follows that for

$i = 1, 2, \ldots, k$

$$\mu_i = \min_{w_1,\ldots,w_{i-1}} \max_{\substack{y^T w_j = 0 \\ j=1,\ldots,i-1;k+1,\ldots,n}} \frac{y^T P_V A P_V y}{y^T y} \qquad (4.8)$$

eq:4.8

where $w_1, \ldots, w_{i-1}$ are arbitrary vectors in $\mathbb{R}^n$ and $w_{k+1}, \ldots, w_n$ are linearly independent vectors orthogonal to M_k. But for a vector $y \in M_k$, $y = P_V y$, and so $y^T P_V A P_V y$ may be replaced by $y^T A y$ in (4.8) so that for $i = 1, 2, \ldots, k$

$$\lambda_i = \min_{w_1,\ldots,w_{i-1}} \max_{\substack{y^T w_j = 0 \\ j=1,2,\ldots,i-1}} \frac{y^T A y}{y^T y} \geq \mu_i,$$

since μ_i is subject to $n - k$ extra constraints. If now the $w_{k+1}, \ldots, w_n$ can be varied over all of $\mathbb{R}^n$ in (4.8) the value of the right hand side may be decreased, giving for $i = 1, 2, \ldots, k$

$$\mu_i \geq \min_{\substack{w_j \in \mathbb{R}^n \\ j=1,\ldots,i-1;k+1,\ldots,n}} \max_{\substack{y^T w_j = 0 \\ j=1,\ldots,i-1;k+1,\ldots,n}} \frac{y^T A y}{y^T y} \equiv \lambda_{i+n-k}$$

so that

$$\lambda_i \geq \mu_i \geq \lambda_{i+n-k}, \quad i = 1, 2, \ldots, k. \qquad (4.9)$$

eq:4.9

Thus finding the eigenvalues μ_i of T gives lower bounds on the k greatest and upper bounds on the k least, eigenvalues of A.

Suppose y_i gives μ_i in (4.8), then it can be shown that

$$P_V A P_V y_i = y_i \mu_i,$$

and from (4.5) and (4.6) $y_i = V z_i$ where z_i is the corresponding eigenvector of T, so that finding the eigenvectors of T will allow these y_i to be obtained. These y_i can then be taken as approximations to some of the eigenvectors of A. From (4.8) and (4.9) it can be seen that μ_1 is the best approximation to λ_1 that can be found by considering Rayleigh quotients with respect to A of vectors in M_k, and y_1 can be taken as a corresponding approximation to x_1. The maximum value of the Rayleigh quotient for a vector in M_k which is orthogonal to y_1 then turns out to be μ_2, and the vector y_2 which gives this can be taken as an approximation to x_2, and so on. These y_i are certainly not the best approximations to the x_i in the 2-norm, as these would be $P_V x_i$.

4.2 Accuracy of Eigenvector Approximations

sec:4.2

As a result of the remarks in the last paragraph it is necessary to examine the closeness of approximation of the vectors y_j to some eigenvectors x_j of A. Here a theoretical result will be derived that will also be useful for obtaining theoretical eigenvalue bounds later. First the y_j, normalized as in (4.5), can be expressed in terms of the x_j in (4.3), let

$$y_j = \sum_{i=1}^{n} x_i \beta_{ij}, \quad j = 1, 2, \ldots, k.$$

The 2-norm error of y_j as an approximation to x_j can then be denoted by ϵ_j where

$$\epsilon_j^2 = \|y_j - x_j \beta_{jj}\|_2^2 = \sum_{i \neq j} \beta_{ij}^2 = 1 - \beta_{jj}^2, \tag{4.10}$$

eq:4.10

with the normalization already chosen. Note that ϵ_j is a measure of the component of y_j orthogonal to x_j rather than the difference between y_j and x_j as is sometimes chosen. Now

$$\mu_j = \frac{y_j^T A y_j}{y_j^T y_j} = \sum_{i=1}^{n} \beta_{ij}^2 \lambda_i$$

so

$$\begin{aligned} \lambda_j - \mu_j + \sum_{i=1}^{j-1} \beta_{ij}^2 (\lambda_i - \lambda_j) &= \sum_{i=j+1}^{n} \beta_{ij}^2 (\lambda_j - \lambda_i) \\ &\geq (\lambda_j - \lambda_{j+1}) \sum_{i=j+1}^{n} \beta_{ij}^2, \end{aligned}$$

because of the ordering in (4.4); but from (4.10)

$$\sum_{i=j+1}^{n} \beta_{ij}^2 = \epsilon_j^2 - \sum_{i=1}^{j-1} \beta_{ij}^2$$

therefore

$$\epsilon_j^2 \leq \frac{\lambda_j - \mu_j + \sum_{i=1}^{j-1} \beta_{ij}^2 (\lambda_i - \lambda_{j+1})}{\lambda_j - \lambda_{j+1}} \tag{4.11}$$

eq:4.11

with equality if $\beta_{ij} = 0$, $i > j+1$. In order to bound β_{ij}^2, $i < j$, use will be made of $\beta_{jj}^2 = 1 - \epsilon_j^2$, $j = 1, 2, \ldots, k$. Now

$$\beta_{ij} = y_j^T x_i = -y_j^T (y_i - x_i \beta_{ii})/\beta_{ii}, \quad i \neq j,$$

since $y_j^T y_i = \delta_{ij}$, so that

$$\begin{aligned} \beta_{ij}^2 &\leq \|y_j\|_2^2 \epsilon_i^2 / \beta_{ii}^2 \quad \text{by Cauchy-Schwarz,} \\ &= \epsilon_i^2/(1 - \epsilon_i^2), \quad i \neq j. \end{aligned}$$

Combining this result with (4.11) gives

$$\epsilon_j^2 \leq \frac{\lambda_j - \mu_j + \sum_{i=1}^{j-1} (\lambda_i - \lambda_{j+1}) \epsilon_i^2 / (1 - \epsilon_i^2)}{\lambda_j - \lambda_{j+1}}. \qquad (4.12) \quad \boxed{\texttt{eq:4.12}}$$

Kaniel (1966) sought to determine a result of this kind, but the result given in his Lemma 5.3 is incorrect. Thus, with a knowledge of the eigenvalues μ_i of T, (4.12) gives theoretical bounds, depending on the eigenvalues λ_i of A, on the error in the vector y_j as an approximation to the eigenvector x_j of A, as long as the ϵ_i, $i = 1, 2, \ldots, j-1$, can be bounded so that $\epsilon_i^2 < 1$. From (4.10) $\epsilon_j^2 \leq 1$ so that (4.12) is only useful when $\lambda_j - \mu_j < \lambda_j - \lambda_{j+1}$, that is, when there are no eigenvalues of A between λ_j and μ_j with the ordering in (4.4) and (4.7). It is now clear that (4.12) is most useful for the larger eigenvalues, but a similar expression could also be found to bound ϵ_k^2, ϵ_{k-1}^2, $\ldots$, in that order.

It is of passing interest to note that most of the theory so far given in this section applies equally well for any set of linearly independent vectors $v_1, v_2, \ldots, v_k$ spanning an arbitrary k dimensional linear manifold M_k.

4.3 Rate of Convergence of the Eigenvalues and Eigenvectors

sec:4.3

In this section a comparison of the Lanczos process with the Tchebycheff iteration will be used to bound $\lambda_j - \mu_j$ after k steps with a given distribution of eigenvalues λ_i and initial vector v_1. The expression (4.12) can then be used to bound the eigenvector errors. The theory was developed by Kaniel (1966), but his proofs and results are often obscure and sometimes erroneous, and so a simplified but full and hopefully correct account will be given here.

First for all $g \in M_k$ such that $g^T g = 1$ and $g^T x_i = 0$, $i = 1, 2, \ldots, j-1$, it will be shown that for $j \leq k$ with the ordering given in (4.4) and (4.7)

$$\lambda_j \geq \mu_j \geq g^T A g - \sum_{i=1}^{j-1} \mu_i \epsilon_i^2 \geq g^T A g - \sum_{i=1}^{j-1} \lambda_i \epsilon_i^2, \tag{4.13}$$

eq:4.13

with ϵ_i as in (4.10) and (4.12). The first and last inequalities follow directly from (4.9), while if $g = g_1 + \sum_{i=1}^{j-1} \beta_i y_i$ where $g_1^T y_i = 0$, $i = 1, \ldots, j-1$, then using (4.10) and the Cauchy-Schwarz inequality

$$|\beta_i| = |g^T y_i| = |g^T (y_i - x_i \beta_{ii})| \leq \epsilon_i, \quad i = 1, \ldots, j-1.$$

Now $g_1 = P_V g_1$ and for $i \leq k$, $y_i = P_V y_i$ so that

$$g_1^T A y_i = g_1^T P_V A P_V y_i = \mu_i g_1^T y_i = 0, \quad i < j,$$

and

$$y_i^T A y_j = y_i^T P_V A P_V y_j = \mu_i \delta_{ij}, \quad i, j \leq k,$$

from Section 4.1.3, so that as a result

$$\frac{g^T A g}{g^T g} = \frac{g_1^T A g_1}{g^T g} + \frac{\sum_{i=1}^{j-1} \mu_i \beta_i^2}{g^T g} \leq \frac{g_1^T A g_1}{g_1^T g_1} + \sum_{i=1}^{j-1} \mu_i \epsilon_i^2.$$

But $g_1 \in M_k$ and is orthogonal to $y_1, \ldots, y_{j-1}$, so

$$\mu_j = \max_{\substack{y \in M_k,\ y^T y_i = 0 \\ i=1,2,\ldots,j-1}} \frac{y^T A y}{y^T y} \geq \frac{g_1^T A g_1}{g_1^T g_1}$$

and the result (4.13) follows.

Note that for g satisfying the conditions of (4.13), $g^T Ag \leq \lambda_j$, and the closer this is to equality, the closer μ_j approximates λ_j. The aim then is to find g satisfying these conditions such that $g^T Ag$ is as large as possible. If $T_k(t)$ denotes the Tchebycheff polynomial normalized so that $\max_{\{-1\leq t\leq 1\}} |T_k(t)| = 1$, then use will be made of the rapid increase of $T_k(t)$ with increasing $t > 1$ (see, for example, Todd, 1962, p. 127), to obtain such a vector g giving a good lower bound on μ_j in (4.13). For this purpose it will be necessary to consider the shifted and scaled matrix

$$B \equiv [2A - (a+b)I]\,/(a-b) = cA + dI, \quad \text{say}, \qquad (4.14) \quad \boxed{\texttt{eq:4.14}}$$

which has the same eigenvectors as A, but with eigenvalues

$$\nu_i = c\lambda_i + d, \quad i = 1, 2, \ldots, n, \qquad (4.15) \quad \boxed{\texttt{eq:4.15}}$$

and taking $a = \lambda_s$, s any integer such that $1 < s < n$, and $b = \lambda_n$ gives

$$\nu_i = [2\lambda_i - (\lambda_s + \lambda_n)]\,/(\lambda_s - \lambda_n) \qquad (4.16) \quad \boxed{\texttt{eq:4.16}}$$

so that

$$-1 = \nu_n < \ldots < \nu_s = 1 < \nu_{s-1} < \ldots < \nu_1. \qquad (4.17) \quad \boxed{\texttt{eq:4.17}}$$

In order to bound the j-th eigenvalue, $j < s$, the n indices in (4.17) will be divided into three sets as follows

$$\left.\begin{aligned} j &= \{j\}, \\ L &= \{1, 2, \ldots, j-1, j+1, \ldots, s-1\}, \\ M &= \{s, s+1, \ldots, n\}. \end{aligned}\right\} \qquad (4.18) \quad \boxed{\texttt{eq:4.18}}$$

Now the linear manifold M_k has the Krylov sequence $\{v_1, Av_1, \ldots, A^{k-1}v_1\}$ as a basis, so from (4.14) any vector $y \in M_k$ can be represented as

$$y = p_k(B)v_1 \qquad (4.19) \quad \boxed{\texttt{eq:4.19}}$$

where $p_k(t)$ is a polynomial in t of degree less than or equal to k. So writing $v_1 = \sum_{i=1}^n \alpha_i x_i$ and defining

$$v_L = \begin{cases} \prod_{l \in L} (B - \nu_l I)\, v_1 = \sum_{i=1}^n \alpha_i \prod_{l \in L} (\nu_i - \nu_l) x_i \\ v_1, \quad \text{if } L \text{ is the empty set,} \end{cases} \qquad (4.20) \quad \boxed{\texttt{eq:4.20}}$$

and

$$g = T_m(B) v_L = \sum_{i=1}^n \alpha_i T_m(\nu_i) \prod_{l \in L} (\nu_i - \nu_l) x_i, \qquad (4.21) \quad \boxed{\texttt{eq:4.21}}$$

then this is of the form $g = p_{m+s-2}(B) v_1$, so if $m = k-s+2 \geq 0$ it follows that $g \in M_k$, and from the choice of v_L, $g^T x_i = 0, i \in L$. Thus g satisfies the conditions for (4.13), apart from normalization, and all the components of g corresponding to eigenvalues of B in (4.17) with moduli greater than unity are suppressed except x_j. The choice of v_L is then just a way of singling out ν_j, and so the Tchebycheff polynomial will give a very fast increase with increasing k in the coefficient of x_j, relative to the remainder.

To be useful in (4.13) a lower bound on $g^T Bg/g^T g$ is required, and from (4.21)

$$\begin{aligned} \frac{g^T Bg}{g^T g} &= \nu_j - \frac{\sum_{i=1}^n \left[\alpha_i T_m(\nu_i) \prod_{l \in L} (\nu_i - \nu_l)\right]^2 (\nu_j - \nu_i)}{\sum_{i=1}^n \left[\alpha_i T_m(\nu_i) \prod_{l \in L} (\nu_i - \nu_l)\right]^2} \\ &\geq \nu_j - \frac{\sum_{i=s}^n \left[\alpha_i \prod_{l \in L} (\nu_i - \nu_l)\right]^2 (\nu_j - \nu_i)}{\left[\alpha_j T_m(\nu_j) \prod_{l \in L} (\nu_j - \nu_l)\right]^2}. \end{aligned} \qquad (4.22) \quad \boxed{\texttt{eq:4.22}}$$

Now using the expressions (4.14) for B, (4.15) for ν_i, ν_j and ν_l, and (4.16) for ν_j in (4.22), and substituting in (4.13) gives

$$\lambda_j \geq \mu_j \geq \lambda_j - \frac{\sum_{i=s}^{n}\left[\alpha_i \prod_{l\in L}\left(\lambda_i-\lambda_l\right)\right]^2\left(\lambda_j-\lambda_i\right)}{\left[\alpha_j T_{k-s+2}\left(\frac{\lambda_j-\lambda_n+d_j}{\lambda_j-\lambda_n-d_j}\right)\prod_{l\in L}\left(\lambda_j-\lambda_l\right)\right]^2} - \sum_{i=1}^{j-1}\lambda_i\epsilon_i^2, \qquad (4.23) \quad \boxed{\texttt{eq:4.23}}$$

where $d_j = \lambda_j - \lambda_s$, and the ϵ_i can be bounded as in (4.12). Thus by varying the integers j and s , various bounds can be given for the different eigenvalues μ_j of T after k steps of the Lanczos process. For instance putting $j = 1$, $s = 2$, gives

$$\lambda_1 \geq \mu_1 \geq \lambda_1 - \frac{\left(\|v_1\|_2^2 - \alpha_1^2\right)\left(\lambda_1-\lambda_n\right)}{\left[\alpha_1 T_k\left(1+2\frac{\lambda_1-\lambda_2}{\lambda_2-\lambda_n}\right)\right]^2}$$

so that for a well separated maximum (or minimum) eigenvalue, the rate of convergence is always very fast, in fact at least as fast as that obtained using the optimum Tchebycheff iteration (see, for example, Wilkinson, 1965, p. 617), with the Rayleigh quotient.

For a closely bunched set of large eigenvalues it will still be possible to choose s in (4.23) so that $d_1 = \lambda_1 - \lambda_s$ is a reasonable size, thus again establishing fast convergence to λ_1 even though the initial error may be large because of the $\prod(\lambda_j - \lambda_l)^2$ term in the denominator.

In order to illustrate the bounds that can be obtained by the above expressions Kaniel considers the following example: $\lambda_1 = 1.00$, $\lambda_2 = 0.99$, $\lambda_3 = 0.96$, $\lambda_i \le 0.9$ for $i > 3$, $\lambda_n = 0$, while in (4.20) $\|v_1\|_2^2 = 1$ and $|\alpha_i| = 0.01$, $i = 1, 2, 3$.

Bounds after, for example, $k = 52$ steps can be found by taking $s = 4$ in (4.18) and choosing j to be 1, 2, and 3 successively. For this example the numerator in (4.23) satisfies

$$\sum_{i=s}^{n} \left[\alpha_i \prod_{l \in L} (\lambda_i - \lambda_l) \right]^2 (\lambda_j - \lambda_i) \le \sum_{i=1}^{n} \alpha_i^2 = 1$$

so that (4.23) simplifies to

$$\lambda_j \ge \mu_j \ge \lambda_j - \left[0.01 T_{50} \left(\frac{2\lambda_j - 0.9}{0.9} \right) \prod_{l \in L} (\lambda_j - \lambda_l) \right]^{-2} - \sum_{i=1}^{j-1} \lambda_i \epsilon_i^2$$

for $j = 1, 2, 3$; L being the complement of $\{j\}$ with respect to $\{1, 2, 3\}$. The ϵ_i^2 must be bounded successively using (4.12) so that the errors in the eigenvector approximations, described by (4.10), are also bounded. Thus for the largest eigenvalue

$$\begin{aligned} 1 \ge \mu_1 \ge 1 &- \left[0.01 \times 0.01 \times 0.04 \times T_{50} \left(\frac{1.1}{0.9} \right) \right]^{-2} \\ &> 1 - \left[4 \times 10^{-6} \times 0.83 \times 10^{14} \right]^{-2} \\ &> 1 - 10^{-17} \end{aligned}$$

using $T_m(\cosh \beta) = \cosh m\beta > \frac{1}{2} e^{m\beta}$. It then follows from (4.12) that

$$\epsilon_1^2 < 10^{-17} / 10^{-2} = 10^{-15}$$

so for the next eigenvalue

$$\begin{aligned} 0.99 \ge \mu_2 \ge 0.99 &- [0.01 \times 0.01 \times 0.03 \times T_{50}(1.2)]^{-2} - 10^{-15} \\ &> 0.99 - 1.5 \times 10^{-15}, \end{aligned}$$

which with (4.12) gives

$$\begin{aligned} \epsilon_2^2 &< \left(1.5 \times 10^{-15} + 0.041 \times 10^{-15} \right) / 0.03 \\ &< 0.6 \times 10^{-13}. \end{aligned}$$

Finally

$$\begin{aligned} 0.96 \geq \mu_3 &\geq 0.96 - \left[0.01 \times 0.04 \times 0.03 \times T_{50}\left(\frac{1.02}{0.9}\right)\right]^{-2} - 10^{-15} - 0.6 \times 10^{-13} \\ &> 0.96 - 2 \times 10^{-12} \end{aligned}$$

while

$$\begin{aligned} \epsilon_3^2 &< \left(2 \times 10^{-12} + 0.11 \times 10^{-15} + 0.091 \times 0.6 \times 10^{-13}\right)/0.06 \\ &< 0.4 \times 10^{-10}. \end{aligned}$$

In summary

$$\begin{aligned} \lambda_1 - \mu_1 &< 10^{-17} \quad ; & \epsilon_1^2 &< 10^{-15} \\ \lambda_2 - \mu_2 &< 2 \times 10^{-15}; & \epsilon_2^2 &< 10^{-13} \\ \lambda_3 - \mu_3 &< 2 \times 10^{-12}; & \epsilon_3^2 &< 10^{-10} \end{aligned}$$

so that these eigenvalues and their eigenvectors can certainly be found very accurately in 52 steps of the Lanczos process. Two points must be emphasized however, first these bounds may in theory be far too large, for example if the dimension of the matrix $n \leq 52$ then the eigensolution is theoretically exact; secondly these results assume perfect computation and in any practical computation these bounds are likely to be exceeded. All that can really be said from these results is that for very large symmetric matrices the Lanczos process carried out with infinite precision is an excellent method for finding some extreme eigenvalues and their eigenvectors in much less than the full number of steps.

It is also indicated in Kaniel's paper that most of the theory in this section can be extended to Hermitian operators on separable Hilbert spaces.

.

Section 5

Eigenvalue and Eigenvector Intervals

chp:5

Since it has been shown in the previous section that the symmetric Lanczos process with exact arithmetic converges rapidly when used as an iterative method, the question now arises as to what bounds may be computed on the eigenvalues and eigenvectors of A after $k < n$ steps. The most easily obtainable useful intervals, to be called the basic intervals, will first be developed here, and these will also be viewed as a lead into Lehmann's work. Lehmann (1963, 1966) considered very thoroughly the optimum intervals of this type when exact arithmetic is assumed, and since an English translation of his work does not appear to be easily available, some of the relevant theory will be developed here. An approach different from that of Lehmann will be used in places as, as well as giving motivation, it leads to three different algorithms for computing eigenvalue intervals, one of which is more accurate than the equivalent one suggested by Lehmann when finite precision arithmetic is used; the other two are of interest, but of only minor importance in practice.

The advantages of the different possible intervals will be compared and some conclusions on their usefulness will be drawn.

5.1 The Basic Eigenvalue Intervals

sec:5.1

First the most obvious approach to finding eigenvalue intervals will be given. Later the Lehmann intervals will be seen as a development of these. Let $V = (v_1, v_2, \ldots, v_k)$ be an n by k matrix with linearly independent columns, then as in (4.2) put

$$T = \left(V^T V\right)^{-1} V^T A V.$$

Suppose $\quad Tz_j = \mu_j z_j, \quad y_j = V z_j, \quad y_i^T y_j = \delta_{ij}; \quad i, j = 1, \ldots, k,$

then

$$y_j^T A y_j = z_j^T V^T A V z_j = z_j^T V^T V T z_j = \mu_j z_j^T V^T V z_j = \mu_j,$$

so that μ_j is the Rayleigh quotient for the matrix A with the vector y_j. Now using (4.3) to give the spectral expansion of y_j

$$y_j = \sum_{i=1}^{n} \alpha_i x_i \tag{5.1}$$

eq:5.1

it follows that

$$\|Ay_j - \mu_j y_j\|_2^2 = \sum_{i=1}^{n} \alpha_i^2 (\lambda_i - \mu_j)^2$$

so that for at least one eigenvalue, λ_{r_j} say, of A

$$|\lambda_{r_j} - \mu_j| \leq \|Ay_j - \mu_j y_j\|_2 \tag{5.2}$$

eq:5.2

since $\sum_{i=1}^{n} \alpha_i^2 = 1$.

If the vectors v_i were obtained from the Lanczos process then from (4.1)

$$AVz_j = \mu_j V z_j + E z_j$$

giving

$$|\lambda_{r_j} - \mu_j| \leq \|E z_j\|_2 = |\zeta_{kj} t_{k+1,k}| \, \|v_{k+1}\|_2, \tag{5.3}$$

eq:5.3

where ζ_{kj} is the last element of z_j.

In order to obtain an idea of the over-all error, let λ_{r_j} be the closest eigenvalue of A to μ_j, $j = 1, \ldots, k$, then

$$\sum_{j=1}^{k} \left|\mu_j - \lambda_{r_j}\right|^2 \le \sum_{j=1}^{k} \|Ez_j\|_2^2 = \|EZ\|_E^2$$

where the subscript E indicates the Frobenius (Euclidean, Schur) norm, and $Z \equiv (z_1, z_2, \ldots, z_k)$. But $V^T V = D$ is diagonal, and from (4.5)

$$Z^T V^T V Z = \left(D^{\frac{1}{2}} Z\right)^T D^{\frac{1}{2}} Z = I \tag{5.4}$$

eq:5.4

so that $D^{\frac{1}{2}} Z$ is an orthogonal matrix, and

$$\begin{aligned} \sum_{j=1}^{k} \left|\mu_j - \lambda_{r_j}\right|^2 &\le \|ED^{-\frac{1}{2}} D^{\frac{1}{2}} Z\|_E^2 = \|ED^{-\frac{1}{2}}\|_E^2 \\ &= t_{k+1,k}^2 v_{k+1}^T v_{k+1} / v_k^T v_k. \end{aligned} \tag{5.5}$$

eq:5.5

The magnitudes of the last two vectors thus give an immediate indication of the overall state of the process, while with a knowledge of any eigenvalue-eigenvector pair of T an interval containing an eigenvalue of A is given by (5.3). The value μ_j giving (5.2) and (5.3) is the Rayleigh quotient corresponding to y_j, and so gives the best interval using the given vector y_j, but it will be shown in Section 5.3 how an even smaller interval may be obtained using Lehmann's work to find a more suitable vector than y_j.

5.2 Eigenvector Bounds

sec:5.2

Eigenvector bounds are not so easily available as are eigenvalue bounds, since information on the separation of the eigenvalues of A is first needed. For example if for some scalar μ and vector

$$y = \sum_{i=1}^{n} \alpha_i x_i, \quad y^T y = 1,$$

a constant a is known such that

$$|\lambda_i - \mu| \geq a, \quad i \neq j$$

then

$$\|Ay - \mu y\|_2^2 = \sum_{i=1}^{n} \alpha_i^2 (\lambda_i - \mu)^2 \geq a^2 \sum_{i \neq j} \alpha_i^2$$

so that for the component of y orthogonal to x_j

$$\|y - \alpha_j x_j\|_2^2 = \sum_{i \neq j} \alpha_i^2 \leq \|Ay - \mu y\|_2^2 / a^2. \tag{5.6}$$

eq:5.6

Thus with a knowledge of a this could be combined with (5.2) and (5.3) to obtain a useful bound by taking $y = Vz_j$ and $\mu = \mu_j$, as long as convergence of the process is such that

$$|\lambda_j - \mu_j| << a$$

(see Wilkinson, 1965, pp. 173-4). However it is also true that if such a constant a is known then very much better eigenvalue bounds than (5.2) and (5.3) may be obtained (ibid).

The aim of the next section is to decrease the right hand side of (5.2), which will do the same for (5.6), and so nothing further need be said in this section on eigenvector bounds.

5.3 The Lehmann Eigenvalue Intervals

sec:5.3

The bound (5.2) considered only one trial vector y_j and the corresponding Rayleigh quotient μ_j, and the particular pair chosen was useful because it gave an immediately obtainable bound on the distance from an eigenvalue of T to the nearest eigenvalue of A. However from its derivation it is clear that (5.2) is independent of the choice of μ or y, as long as $y^T y = 1$, and so a different choice of y could give a smaller interval.

Lehmann (1963) considered the following problem:– Given only the information

$$\mathcal{J}(A) \equiv \{v_i, Av_i\}_{i=1}^{k} \qquad (5.7) \quad \boxed{\texttt{eq:5.7}}$$

where the $v_1, \ldots, v_k$ are a linearly independent set of real vectors, find optimum intervals containing eigenvalues of the real symmetric matrix A. (In fact he considered a symmetric or self-adjoint operator A over a real separable inner-product space H with domain D_A dense in H or, if A is bounded, equal to H, and assumed that the eigenvalues of A did not have an accumulation point and there existed a corresponding orthonormal system of eigenvectors spanning H.)

By considering all combinations of the v_i in (5.7) the vectors

$$y = Vz \qquad (5.8) \quad \boxed{\texttt{eq:5.8}}$$

are obtained where there are no restrictions on the real k-vectors z, and an obvious approach is to optimize (5.2) over all such vectors y with unit norm and real values μ. (5.2) then becomes, for some i,

$$|\lambda_i - \mu| \le \|(A - \mu I)Vz\|_2 \quad \text{where } \|Vz\|_2 = 1, \qquad (5.9) \quad \boxed{\texttt{eq:5.9}}$$

and the problem of minimizing the bound can then be stated

$$\left.\begin{array}{ll} \underset{\mu,\, z}{\text{minimize}} & z^T V^T (A - \mu I)^2 V z, \\ \text{subject to} & z^T V^T V z = 1. \end{array}\right\} \qquad (5.10) \quad \boxed{\texttt{eq:5.10}}$$

This is a constrained minimization, but by introducing the Lagrange multiplier γ it can be reformulated as the unconstrained minimization

$$\left.\begin{array}{ll} & \underset{\gamma,\, \mu,\, z}{\text{minimize}} \quad \phi(\gamma, \mu, z) \\ \text{where} & \phi(\gamma, \mu, z) = z^T V^T (A - \mu I)^2 V z - \gamma \left(z^T V^T V z - 1\right). \end{array}\right\} \qquad (5.11) \quad \boxed{\texttt{eq:5.11}}$$

Since this is unconstrained with continuous derivatives, the extrema will be given by the vanishing of the partial derivatives of the function $\phi(\gamma, \mu, z)$.

Now $(\partial\phi/\partial\gamma) = 0$ gives the required normalization, while

$$(\partial\phi/\partial\mu) = 0 \;\Rightarrow\; \mu = z^T V^T A V z / z^T V^T V z, \tag{5.12}$$

eq:5.12

with an obvious minimum for this value, and

$$(\partial\phi/\partial z) = 0 \;\Rightarrow\; V^T (A - \mu I)^2 V z = \gamma V^T V z. \tag{5.13}$$

eq:5.13

This last is a k by k eigenproblem, and since $V^T V$ is positive definite, while the matrix on the left is at least positive semi-definite, the eigenvalues must be real and non-negative and may be written as $\gamma \equiv \Delta^2(\mu)$ or just Δ^2. Now it is no simple matter to find z and μ satisfying both (5.12) and (5.13), and so first (5.13) will be examined in isolation for an arbitrary choice of μ. Multiplying (5.13) on the left by z^T and combining with (5.9) then gives for some eigenvalue λ_i of A

$$|\lambda_i - \mu| \leq \Delta \tag{5.14}$$

eq:5.14

where Δ^2 is the minimum eigenvalue of

$$V^T (A - \mu I)^2 V z = \Delta^2 V^T V z. \tag{5.15}$$

eq:5.15

Different values of μ will vary this value, and values μ and z that simultaneously satisfy (5.12) and (5.15) (for the minimum eigenvalue) will give best bounds of this type. A means of finding these will be presented later, but for the moment (5.15) will be examined in more detail.

First let the eigenvalues of (5.15) be ordered according to

$$0 \leq \Delta_1 \leq \Delta_2 \leq \ldots \leq \Delta_k \tag{5.16}$$

eq:5.16

and following Lehmann, denote the eigenvalues of A by λ_i^μ, the superscript μ determining the ordering by distance from μ

$$|\lambda_1^\mu - \mu| \leq |\lambda_2^\mu - \mu| \leq \ldots. \tag{5.17}$$

eq:5.17

Then because of the similarity of (5.15) to the eigenproblem

$$V^TAVz = \mu V^TVz$$

the theory of Section 4.1.4 may be applied here which, with the ordering just given, gives the equivalent of (4.9)

$$\left(\lambda^\mu_{n-j+1} - \mu\right)^2 \geq \Delta^2_{k-j+1} \geq \left(\lambda^\mu_{k-j+1} - \mu\right)^2, \quad j = 1, \ldots, k,$$

or writing $i = k - j + 1$

$$|\lambda^\mu_i - \mu| \leq \Delta_i \leq |\lambda^\mu_{i+n-k} - \mu|, \quad i = 1, \ldots, k. \qquad (5.18) \quad \boxed{\texttt{eq:5.18}}$$

From this and (5.17) it follows that in each complete λ-interval

$$\{\lambda : \ |\lambda - \mu| \leq \Delta_i\}, \quad i = 1, \ldots, k, \qquad (5.19) \quad \boxed{\texttt{eq:5.19}}$$

lie at least i eigenvalues of A, while in each region

$$\{\lambda : \ |\lambda - \mu| \geq \Delta_i\}, \quad i = 1, \ldots, k, \qquad (5.20) \quad \boxed{\texttt{eq:5.20}}$$

lie at least $k - i + 1$ eigenvalues.

So far a fixed value μ has been taken in (5.9) and the smallest intervals of this type have been found by choosing optimum z. These intervals have centre μ and width 2Δ. However if more information is available then better bounds may be found, for instance Temple (1928) considered the case where a value t was known such that with the ordering (4.4)

$$\lambda_2 \leq t < \lambda_1, \qquad (5.21) \quad \boxed{\texttt{eq:5.21}}$$

and he used a single trial vector to find a useful upper bound τ on λ_1. This result was generalized by Lehmann (1963) as follows. Rearranging (5.15) and defining $t = \mu - \Delta$ and $\tau = \mu + \Delta$, that is, letting $[t, \tau]$ denote the previous interval,

$$V^T\left[A^2 - (t + \tau)A + t\tau I\right]Vz = 0$$

so

$$V^T(A - tI)^2 Vz = (\tau - t)V^T(A - tI)Vz$$

or

$$V^T(A - tI)Vz = \frac{1}{(\tau - t)} V^T(A - tI)^2 Vz \qquad (5.22) \quad \boxed{\texttt{eq:5.22}}$$

so that given a real value t this eigenvalue problem can be solved for the real eigenvalues $1/(\tau_i - t)$ as long as t is not an eigenvalue of A. Now if t is not an eigenvalue of A then $\tau - t \neq 0$ and the k eigenvalues can be ordered so that

$$\tau_{-r} \leq \ldots \leq \tau_{-1} < t < \tau_1 \leq \ldots \leq \tau_s, \quad r + s = k. \qquad (5.23) \quad \boxed{\texttt{eq:5.23}}$$

From the definitions of t and τ and the results for the intervals (5.19) and (5.20) it then follows that in each closed interval

$$[t, \tau_i], \quad i = 1, \ldots, s, \qquad (5.24) \quad \boxed{\texttt{eq:5.24}}$$

lie at least i, and in each remaining region

$$\mathbb{R}^1 \backslash (t, \tau_i), \quad i = 1, \ldots, s, \qquad (5.25) \quad \boxed{\texttt{eq:5.25}}$$

lie at least $k - i + 1$ eigenvalues of A, and the same holds for $[\tau_{-i}, t], \quad i = 1, \ldots, r$. Such results can be very useful if for instance a value t close to an eigenvalue λ_{j+1} is known such that with the ordering (4.4)

$$\lambda_{j+1} \leq t < \lambda_j \qquad (5.26) \quad \boxed{\texttt{eq:5.26}}$$

then τ_1 is an upper bound on λ_j, τ_2 on λ_{j-1}, and so on. Lehmann (1966) shows how such added information may often be found for a symmetric matrix, thus giving much better results than could be found by the μ, Δ approach without any added information.

In the next section a means of computing values of μ and z satisfying (5.12) and (5.15) simultaneously will be presented, but first, in order to gain insight into just how good the corresponding intervals (5.19) and (5.20) will be, consider the case where $k = 1$ and $v_1 = (x_1+x_2)/\sqrt{2}$, with the eigenvalue ordering (4.4). Here (5.15) becomes

$$\left[(\lambda_1 - \mu)^2 + (\lambda_2 - \mu)^2\right]/2 = \Delta^2$$

and the minimum value of Δ^2 is given by $\mu = (\lambda_1+\lambda_2)/2$, which gives $\Delta = (\lambda_1-\lambda_2)/2$. The results (5.19) and (5.20) then state that at least one eigenvalue of A lies in $[\lambda_2, \lambda_1]$ while at least one eigenvalue lies outside (λ_2, λ_1). Now as no eigenvalues lie inside (λ_2, λ_1), only one of the end points of the inclusion interval can be dropped without destroying the theorem. Lehmann (1963) proves in general that if M_k contains no eigenvectors of A and if only the information of (5.7) is to be used then only one of the end points of each of the intervals (5.19), (5.20), (5.24) and (5.25) can be dropped without the stated results being destroyed, that is, the results use the information $\mathcal{J}(A)$ in (5.7) optimally.

5.4 Computation of the Lehmann Intervals

sec:5.4

In his 1966 paper Lehmann considered means of computing these eigenvalue intervals, however he concluded that to obtain μ, z, and Δ such that both (5.12) and (5.15) were simultaneously satisfied was a difficult problem. Instead he chose to take the eigenvalues of T, the matrix defined by (4.2), as the centres of his intervals – these being good approximations to the required μ giving the smallest intervals. Next he gave methods for finding the values of Δ on being given μ, or τ on being given t, and to do these he considered mainly the Lanczos vectors and a limited generalization of these.

It is however clear from the different approach used in this thesis that for given values of μ or t the corresponding values of Δ or τ could always be found from the eigenvalue problem formulations (5.15) and (5.22), for any set of linearly independent vectors $v_1, \ldots, v_k$. What is more, if the v_i are the Lanczos iterates then these eigenproblems became particularly simple, as will be shown later. Now the smallest eigenvalue of (5.15) will be of particular interest, and so it is interesting to note that with this eigenproblem formulation it is possible to find values of μ and z satisfying (5.12) and (5.15) simultaneously, with not much more trouble than for just finding the smallest eigenvalue Δ^2 for a given μ, and this can be done for a general linearly independent set of vectors $\{v_1, v_2, \ldots, v_k\}$.

The required local minimum of (5.9) as a function of both μ and z can in fact be found by the following simple iterative process, and as only the smallest eigenvalue of (5.15) will be considered here the usual notation will be temporarily dropped for expediency, and in what immediately follows the subscript i will denote the iterate. Thus having chosen μ_1, for $i = 1, 2, \ldots$ define

$$M_i \equiv V^T \left(A - \mu_i I\right)^2 V \tag{5.27}$$

eq:5.27

and solve

$$M_i z_i = \Delta_i^2 V^T V z_i, \quad z_i^T V^T V z_i = 1, \tag{5.28}$$

eq:5.28

where Δ_i^2 is the minimum eigenvalue of M_i, then form

$$\mu_{i+1} = z_i^T V^T A V z_i \tag{5.29}$$

eq:5.29

and name this Algorithm (1).

Such a process commencing with any μ_1 will converge to values μ, Δ, and z so that both (5.12) and (5.15) are satisfied, for

$$M_{i+1} - M_i = (\mu_i - \mu_{i+1}) V^T \left[2A - (\mu_i + \mu_{i+1}) I\right] V, \tag{5.30}$$

eq:5.30

so

$$z_i^T M_{i+1} z_i = z_i^T M_i z_i - (\mu_{i+1} - \mu_i)^2,$$

but by the minimum property of this eigenvalue

$$\Delta_{i+1}^2 \leq z_i^T M_{i+1} z_i = \Delta_i^2 - (\mu_{i+1} - \mu_i)^2, \tag{5.31}$$

eq:5.31

so that Δ_i is strictly monotonically decreasing with i while $\mu_{i+1} \neq \mu_i$. But $\Delta_i \geq 0$, so $\Delta_i \to \Delta$, therefore from (5.31) $\mu_i \to \mu$, and from the form of the algorithm μ and Δ satisfy (5.12) and (5.15), whether z_i converges or not.

The disadvantage is that there must be two iteration processes here, the inner one using for example inverse iteration to find the smallest eigenvalue, and the outer iteration towards the required μ, Δ, and (subspace for) z. It is natural to attempt to telescope these into one iteration process, such as the one given below, the question then arising as to whether the resulting iteration will converge as required. Consider the combined inverse iteration commencing with a vector z_1

$$\mu_i = z_i^T V^T A V z_i / z_i^T V^T V z_i,$$
$$z_{i+1} = M_i^{-1} V^T V z_i, \quad M_i \text{ as in } (5.27).$$

Let this be named Algorithm (2), and define

$$\delta_i = z_i^T M_i z_i / z_i^T V^T V z_i \geq 0$$

then if δ_i decreases, convergence can be proven. Now

$$\delta_{i+1} - \delta_i = \frac{z_{i+1}^T M_{i+1} z_{i+1}}{z_{i+1}^T V^T V z_{i+1}} - \frac{z_{i+1}^T M_i \left(V^T V\right)^{-1} M_i \left(V^T V\right)^{-1} M_i z_{i+1}}{z_{i+1}^T M_i \left(V^T V\right)^{-1} M_i z_{i+1}}$$

but using (5.30)

$$\delta_{i+1} = z_{i+1}^T M_i z_{i+1} / z_{i+1}^T V^T V z_{i+1} - (\mu_{i+1} - \mu_i)^2$$

and since $V^T V$ is positive definite it is possible to define a symmetric matrix B so that

$$V^T V = B^2, \quad C \equiv B^{-1} M_i B^{-1}, \quad w \equiv B z_{i+1}$$

giving

$$\delta_{i+1} - \delta_i = w^T C w / w^T w - w^T C^3 w / w^T C^2 w - (\mu_{i+1} - \mu_i)^2.$$

Now C is clearly symmetric positive semi-definite and so has an eigensystem

$$Cw_i = \nu_i w_i, \quad w_i^T w_j = \delta_{ij}; \quad i, j = 1, 2, \ldots, k,$$

so let $w = \sum \alpha_i w_i$ and denote the Rayleigh quotients

$$\rho(C^r) = \sum \alpha_i^2 \nu_i^r / \sum \alpha_i^2,$$

then if $\rho(C) = 0$ it also follows that $\rho(C^r) = 0$, $r = 2, 3, \ldots$. Next

$$\left(\sum \alpha_i^2 \nu_i\right)^2 = \left[\sum (\alpha_i)(\alpha_i \nu_i)\right]^2 \leq \left(\sum \alpha_i^2\right)\left(\sum \alpha_i^2 \nu_i^2\right)$$

by Hölder's inequality, giving $\rho(C)^2 \leq \rho(C^2)$, while

$$\left(\sum \alpha_i^2 \nu_i^2\right)^2 = \left[\sum (\alpha_i \nu_i^{3/2})(\alpha_i \nu_i^{1/2})\right]^2 \leq \left(\sum \alpha_i^2 \nu_i^3\right)\left(\sum \alpha_i^2 \nu_i\right),$$

again by Hölder's inequality, so on dividing through

$$\begin{aligned} &\rho(C^2)^2 \leq \rho(C^3)\rho(C) \\ \therefore \quad &\rho(C)\rho(C^2) \leq \rho(C^3) \end{aligned}$$

as a result

$$\delta_{i+1} \leq \delta_i - (\mu_{i+1} - \mu_i)^2$$

so with the same argument as before $\delta_i \to \delta$, $\mu_i \to \mu$, and therefore $M_i \to M$. Now it is clear from the inverse iteration that δ will be an eigenvalue of

$$Mz = \delta V^T V z$$

but whether or not this will be the required Δ^2 will depend on z_1.

For Algorithm (1) a good choice of the initial value μ_1 is an eigenvalue of T, while for Algorithm (2) z_1 can be taken as the eigenvector corresponding to the smallest eigenvalue of M_1, again choosing μ_1 to be an eigenvalue of T. In practice however the results of these (believed new) algorithms were rarely significantly better than

Lehmann's suggested approach of just finding the smallest eigenvalue of M_1 where μ_1 is an eigenvalue of T, and in this light the reward would scarcely be worth the extra effort involved in either of these iterative procedures. The eigenproblem formulation (5.15) for fixed μ however still has an advantage over Lehmann's suggested algorithm (1966), especially when the intervals become small, because then inverse iteration converges very swiftly to the minimum eigenvalue Δ^2 and corresponding eigenvector z, and by using double-length accumulation in the formulation of the Rayleigh quotient with z as will be shown in Section 6.3, the loss of accuracy noted by Lehmann (1966) in his calculation of Δ can be avoided.

5.4.1 (a) Simplification using the Lanczos vectors

sec:5.4a

It has already been mentioned how the problem of finding optimum intervals is simplified in the case of the Lanczos process. Suppose the vectors of V are normalized to have unity 2-norm in (4.1), then

$$AV = VT + E, \quad V^TV = I, \quad V^TE = 0$$

and substituting $AV - \mu V = VT - \mu V + E$ in (5.15) gives

$$\left[(T - \mu I)^2 + E^TE\right] z = \Delta^2 z. \tag{5.32}$$

eq:5.32

Now E^TE has only its (k, k) element non-zero and so this is an easily obtained penta-diagonal symmetric matrix eigenvalue problem, and the smallest eigenvalue can be found very quickly, for example by inverse iteration. With the same substitution equation (5.22) becomes

$$(T - tI)z = \frac{1}{\tau - t}\left[(T - tI)^2 + E^TE\right] z \tag{5.33}$$

eq:5.33

which again is a reasonably simple symmetric form, and the eigenvalues can be found fairly quickly using standard procedures. However the t, τ intervals are only likely to be of use when extra information of the type given in (5.26) is available, and usually several eigenvalues will be wanted, none of which is likely to be very small. Also the eigenvectors are unlikely to be needed to give the required accuracy in the presence of rounding errors, as is the case with the μ, Δ problem, and the method described by Lehmann (1966) for finding the values τ which make the determinant

$$D_k(t,\tau) = \det\left[(T-\tau I)(T-tI)+E^TE\right] \tag{5.34}$$

eq:5.34

vanish for a given value of t will be much faster than solving the eigenvalue problem (5.33) using standard procedures. As it will be assumed that there is no added information available on the eigenvalue distribution of A, this t, τ problem will not be considered further.

5.5 Comparison of the Possible Eigenvalue Intervals

sec:5.5

After step k the accurate Lanczos process applied to the symmetric matrix A will have produced a symmetric k by k tri-diagonal matrix T and a residual vector $t_{k+1,k}v_{k+1}$, where $\|v_{k+1}\|_2 = 1$, and using these, intervals may be found containing eigenvalues of A. The four possible types of interval will be referred to by the following names:–

The Basic Intervals

If $Tz_i = \mu_i z_i$, $\|z_i\|_2 = 1$, $\Delta_i = |\delta_{k+1}e_k^T z_i|$, $i = 1, \ldots, k$, then from (5.3) each interval $[\mu_i - \Delta_i, \mu_i + \Delta_i]$ contains at least one eigenvalue of A.

The Approximate Lehmann Intervals

Given μ find the eigenvalues $\Delta_1^2 \leq \Delta_2^2 \leq \ldots$ of the matrix

$$(T-\mu I)^2 + \delta_{k+1}^2 e_k e_k^T \tag{5.32}$$

then from (5.19) each interval $[\mu - \Delta_i, \mu + \Delta_i]$ contains at least i eigenvalues of A. Here the values μ will be taken to be the eigenvalues of T as these give good approximations to the optimum intervals.

The Optimum Lehmann Intervals

If for an approximate Lehmann interval above, μ is varied to give a local minimum to Δ_1 then an optimum Lehmann interval is obtained. Again each interval $[\mu - \Delta_i, \mu + \Delta_i]$ contains at least i eigenvalues of A.

The t, τ Intervals

Given t solve the eigenvalue problem

$$(T - tI)z = [1/(\tau - t)] \left[(T - tI)^2 + \delta_{k+1}^2 e_k e_k^T\right] z \tag{5.33}$$

to find $\tau_{-r} \leq \ldots \leq \tau_{-1} < t < \tau_1 \leq \ldots \leq \tau_s$, $r + s = k$, then in each interval $[t, \tau_i]$, $i = 1, \ldots, s$ lie at least i eigenvalues of A, and the same for $[\tau_{-i}, t]$, $i = 1, \ldots, r$.

As each of these last three approaches gives a set of k intervals for each μ or t chosen, the smallest of each set will be called the first interval of that kind.

Now each basic interval and each first approximate Lehmann interval is known to contain at least one eigenvalue of A, however there is nothing in the theory to say that if several basic intervals or several first approximate Lehmann intervals overlap then each refers to a different eigenvalue. In fact if two intervals of either kind overlap then their union may contain only one eigenvalue of A, as the following example illustrates. Let

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad v_1 = a \begin{bmatrix} 1 \\ m \\ 1 \end{bmatrix}, \quad a^2 = \left(m^2 + 2\right)^{-1}; \quad a, m \geq 0,$$

then

$$t_{21}v_2 = Av_1 - t_{11}v_1 = a \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix},$$

since $t_{11} = v_1^T Av_1 = 0$, giving $t_{21} = t_{12} = \sqrt{2}a$. Finally

$$t_{32}v_3 = Av_2 - t_{22}v_2 - t_{12}v_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} - \sqrt{2}a^2 \begin{bmatrix} 1 \\ m \\ 1 \end{bmatrix} = \frac{ma^2}{\sqrt{2}} \begin{bmatrix} m \\ -2 \\ m \end{bmatrix},$$

since $t_{22} = v_2^T Av_2 = 0$, giving $\|t_{32}v_3\|_2 = ma$. Thus taking $k = 2$ gives

$$T = \begin{bmatrix} 0 & \sqrt{2}a \\ \sqrt{2}a & 0 \end{bmatrix}, \quad \text{and if} \quad Z = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad D = \begin{bmatrix} \sqrt{2}a & 0 \\ 0 & -\sqrt{2}a \end{bmatrix}$$

then $TZ = ZD$, $Z^T Z = I$, and from (5.3) the basic intervals are

$$|\sqrt{2}a - \lambda| \leq ma/\sqrt{2}, \quad |-\sqrt{2}a - \lambda| \leq ma/\sqrt{2}.$$

Now $m = 0$ gives $\lambda = 1$, $\lambda = -1$ as expected; but as m increases, $a \to 0$, $ma \to 1$, and both intervals tend towards $|\lambda| \leq 1/\sqrt{2}$, that is, if m is large enough then only the zero eigenvalue of A is contained in the union of the two intervals. In fact if $m = 4$ then the eigenvalues 1 and -1 are already at the ends of the intervals, $|\lambda - 1/3| \leq 2/3$, $|\lambda + 1/3| \leq 2/3$.

It has already been shown that the first approximate Lehmann intervals are at least as small as these basic intervals, so again for m large enough the two first intervals will refer to only one eigenvalue of A. It is however useful to compare these intervals with the basic intervals, and to do this, note that the matrix in (5.32) becomes

$$(T - \mu I)^2 + E^T E = a^2 \begin{bmatrix} 4 & \mp 4 \\ \mp 4 & 4{+}m^2 \end{bmatrix}$$

for $\mu = \pm\sqrt{2}a$. Thus the eigenvalues of this matrix are the same for both μ, and these are given by

$$\Delta^2 = \left[m^2 + 8 \pm (m^4 + 64)^{\frac{1}{2}}\right] / (2m^2 + 4).$$

For example if $m = 4$ then $\Delta_1 \doteq 0.41$, $\Delta_2 \doteq 1.1$, so that the first intervals are satisfactorily smaller than the basic intervals, while the larger intervals, in which must lie at least two eigenvalues, are again reasonable. What is more, as m increases the two eigenvalues tend to 0 and 1 respectively, and since eigenvalues of T tend to 0 the intervals become tight, this is in marked contrast to the basic intervals whose widths here are never less than $\sqrt{2}$, as long as m is finite.

In more realistic computations it was found that the basic intervals were usually much less than twice the first approximate Lehmann intervals, while the first optimum Lehmann intervals were rarely more than ten percent better than the approximate ones. Significant differences were encountered, but these usually occurred when one eigenvalue of T had nearly converged to one of A, whereas a corresponding very close eigenvalue of T was a fair way from convergence. In this case Algorithm (1) in Section 5.4 often converged to the first eigenvalue, even when it started the iteration with the second, and this somewhat negates its small advantage.

In Table 1 a comparison of these bounds is given for the matrix A resulting from the finite difference replacement of Laplace's equation with zero boundary conditions on a 5 by 4 square grid. The Lanczos process with re-orthogonalization was applied and the eigenvalues and the corresponding half-intervals are given for the 2 by 2, the 4 by 4, and the first six eigenvalues of the 18 by 18 tri-diagonal matrices. The similarity of the different intervals is clearly indicated, except in the case of the roots of the 18 by 18 matrix corresponding to the repeated root of A, and here the centre of the optimum interval shifts to the already converged root in both cases.

Thus the most easily obtained eigenvalue intervals using the Lanczos process are the basic intervals (5.3), as these require no extra computation once the eigenproblem

CLOSEST EIGEN-VALUE OF ‘A’	k	EIGENVALUE OF T_k	BASIC HALF INTERVAL	APPROXIMATE HALF-INT.	OPTIMUM HALF-INT.	OPTIMUM CENTRE
6.6180339887	2	6.8658824736	.3471431873	.3466372513	.3466372424	6.8658037263
2.3819660113		2.3943542595	.2409520870	.2402271539	.2402271413	2.3944320290
7.3500847963	4	7.3497871198	.0311372110	.0302972647	.0302972631	7.3497775042
6.6180339887		6.6177590368	.0254816508	.0245918621	.0245918613	6.6177525221
2.3819660113		2.3820172842	.0130713426	.0070683876	.0070682740	2.3819771961
1.3819660112		1.5001195650	1.3717864333	.8818294920	.0070682740	2.3819771961
7.3500847963	18	7.3500847966	.0000000000	.0000000002	.0000000000	7.3500847961
6.6180339887		6.6180339890	.0000000000	.0000000002	.0000000001	6.6180339883
6.3500847964		6.3500847964	.0000000000	.0000000001	.0000000000	6.3500847965
5.6180339887		5.6180339887	.0000000034	.0000000002	.0000000001	5.6180339888
5.6180339887		5.6180116657	.0056739163	.0000223231	.0000000001	5.6180339888
5.1140168189		5.1140168186	.0000000000	.0000000003	.0000000001	5.1140168188

and so on, with all the remaining eigenvalues being accurate to within 1 digit in the ninth decimal place.

tab:1

Table 1: COMPARISON OF DIFFERENT EIGENVALUE INTERVALS FOR THE LANCZOS PROCESS.

of T has been solved. Usually these will be close enough to the optimum obtainable intervals to make any other intervals unnecessary. A small disadvantage is that if r intervals overlap then their union may not contain r eigenvalues of A. Nevertheless these are the obvious choice of intervals when using the Lanczos process.

If smaller intervals are really necessary then at some computational expense the approximate Lehmann intervals can be computed by solving the eigenproblem (5.32). Intervals containing more than one eigenvalue of A can be found from the higher eigenvalues, as in this case (5.19) and (5.20) hold.

Optimum intervals of this kind could be found by either Algorithm (1) or with less certainty Algorithm (2) in Section 5.4, but in general the gain does not seem worth the computational effort involved.

If added information is available, such as knowing that a value t lies between two adjacent eigenvalues of A, then the very useful intervals (5.24) and (5.25) can be obtained by solving the eigenproblem (5.33), and these appear to be the most practically useful of the results given by Lehmann (1966).

Although in both the μ, Δ and t, τ cases an eigenvalue problem has to be solved to gain full use of the available information, this is a k by k penta-diagonal matrix problem, and as at most only a few of the smaller eigenvalues will be wanted, the relative time taken will not be too great if $k << n$.

Finally the usual warning must be given that these intervals assume infinite precision computation, both in the Lanczos process and in the computation of the intervals. Any rounding errors may well negate the bounds obtained. In fact it will be shown in the next section how bounds can be obtained on the errors in the Lanczos computation with re-orthogonalization, while bounds on the errors in the direct solution of the eigenproblems (5.32) and (5.33) using stable methods may be obtained from the analyses given by Wilkinson (1965), so in this case rigorous intervals are obtainable for practical computations. However in applications of the symmetric Lanczos process without re-orthogonalization no small bound can be found on the departure of the v_i from orthogonality, the Lehmann intervals obtained from (5.32) and (5.33) are then fairly meaningless, while intervals would most likely not be computed from (5.15) and (5.22) because of storage and time difficulties. For reasons to be given in Sections 7 to 9 the intervals given by (5.3) are also doubtful in this form when re-orthogonalization is not used, and the most reliable intervals will be obtained by computing approximate eigenvectors $y_j = Vz_j$ of A, where $Tz_j = \mu_j z_j$, and using (5.2) directly. The more easily obtained intervals (5.3) will however still be excellent guides.

Section 6

The Symmetric Lanczos Process with Re-orthogonalization

chp:6

The properties of the symmetric Lanczos process given in Sections 4 and 5 were derived assuming infinite precision computation. In this light the method is extremely attractive, especially when considered as an iterative process with a very large matrix, for not only is the convergence of both the eigenvalues and eigenvectors seen to be fast, but also useful eigenvalue intervals may be fairly easily obtained at any stage of the process.

Unfortunately working with a digital computer to a fixed precision introduces rounding errors, and it is well known from experience that these can greatly alter the expected course of the method, and in practice the properties derived in the last two sections need not hold. The most startling divergence from theory is the loss of orthogonality of a vector v_{j+1} to the previous vectors $v_1, \dots, v_j$ whenever cancellation occurs in the computation indicated by (3.7). Lanczos (1950, p. 271) noted the possible loss of orthogonality and advocated re-orthogonalization, that is, whenever

$v_i^T v_{j+1}$, $i = 1, \ldots, j$, is noticeably different from zero, add the correction term

$$-\frac{v_i^T v_{j+1}}{v_i^T v_i} v_i$$

to v_{j+1}. He then found the method to be extremely accurate. Now adding the correction term takes about the same number of operations as testing whether it is necessary, and since both require only the presence of v_{j+1} and v_i it is clear that not too much time will be lost by systematically re-orthogonalizing each new vector against all the previous vectors, and this also avoids the difficulty of defining ‘noticeably different from zero’ exactly. If the vectors are normalized to have unity 2-norm then the jth step of one particular algorithm with full re-orthogonalization, starting with a vector v_1, with unity 2-norm, may be as follows. An intermediate vector c_j is first formed

$$c_j = Av_j - t_{jj}v_j - t_{j-1,j}v_{j-1} \tag{6.1}$$

eq:6.1

where

$$\left.\begin{aligned} &t_{0,1} = 0, \\ &t_{ij} = v_i^T A v_j, \quad i = j-1 \text{ and } j \text{ if } i > 0, \end{aligned}\right\} \tag{6.2}$$

eq:6.2

then the re-orthogonalization is carried out

$$w_j = -b_{1j}v_1 - b_{2j}v_2 - \ldots - b_{jj}v_j + c_j \tag{6.3}$$

eq:6.3

where

$$b_{ij} = v_i^T c_j, \quad i = 1, 2, \ldots, j, \tag{6.4}$$

eq:6.4

and w_j is normalized to give the next vector

$$v_{j+1} = w_j / t_{j+1,j} \tag{6.5}$$

eq:6.5

where

$$t_{j+1,j} = \left(w_j^T w_j\right)^{\frac{1}{2}}. \tag{6.6}$$

eq:6.6

In the literature the Lanczos process with re-orthogonalization usually refers to such full re-orthogonalization, and the extensive use of such processes since Lanczos' paper supports his hypothesis that the re-orthogonalization compensates for the influence of rounding errors. This process was then seen to be the most effective method of reducing a full symmetric matrix to tri-diagonal form, until it was superseded by the faster method advocated by Givens (1954), and later by the even faster method proposed by Householder (Householder and Bauer, 1959). When the Lanczos method was the best available for such full matrices it would have been desirable to have a rounding error analysis giving rigorous bounds on the possible errors in the computation, and in fact in 1956 J. H. Wilkinson did carry out a rough analysis but did not publish it because he was not fully satisfied with it. However since the Givens and Householder methods are both more efficient and have been shown by rigorous analyses to be very accurate (Wilkinson, 1965), there is no longer such a need for an analysis of the Lanczos process, at least for the complete reduction of a full symmetric matrix to tri-diagonal form.

If less than the full number of steps are needed, then because of the relatively small number of operations required in the early steps, the Lanczos process becomes relatively more attractive, especially for large sparse matrices, and an operation count will be given later in this section showing under what circumstances the Lanczos process with re-orthogonalization is actually faster than the equivalent Householder process. It follows then that there are circumstances where a full and rigorous rounding error analysis is still needed, and this will accordingly be given here. The time and storage requirements of the method will then be given and compared with the equivalent formulation of Householder's method, and finally some computational results will be given and some conclusions drawn.

6.1 Rounding Error Analysis

sec:6.1

The rounding error analysis of the symmetric Lanczos process with re-orthogonalization has been presented by the author in two documents (Paige, 1969b, 1970b). The first, an Institute of Computer Science internal document, gives the full error analysis, obtains all the error bounds, and gives an example where re-orthogonalization actually breaks down; it then considers briefly the effect of errors on computing the approximate Lehmann intervals mentioned in Section 5. Although the second document gives the basic error analysis it does not obtain the error bounds, a lengthy and tedious process, but summarizes the results of the first paper instead. Thus although the first is the more complete work, both are included at the end of this thesis since the second is more clear, precise and readable, and corrects several minor errors occurring in the original as well as including a computational example showing the rapid convergence of the process.

Before going on to point out the essential parts of the above papers it is only fair to mention that some of the results may not be original. It was noted earlier that J. H. Wilkinson of the National Physical Laboratory in Teddington England carried out a rough analysis of the process in 1956 but did not publish it, while J. Meinguet at the University of Louvain Belgium indicated in a letter in December 1968 that he had done some work in this direction, but that the results were mainly of academic interest and had not been written up. As neither of these two works were published or available in any form the author is uncertain as to what extent his own results replicate these works.

The method of analysis used followed that outlined in Section 3, and so the possible rounding errors occurring in the practical computation of each of equations (6.1) to (6.6) were first described using the theory of Section 2. Now each of these equations involves individual vectors v_j etc. and describes the computation at one step only, whereas the results of the first k steps can be combined for each of these equations, giving equivalent matrix equations. For instance (6.1) becomes

$$C = AV - V(T - K) + \delta V \qquad (6.7) \quad \boxed{\texttt{eq:6.7}}$$

where $C \equiv (c_1, \ldots, c_k)$, $V \equiv (v_1, \ldots, v_k)$, T is the k by k tri-diagonal matrix of coefficients t_{ij}, and K is just T without its diagonal and super-diagonal elements. These elements represent the actually computed elements, while δV, which describes the rounding errors that occur in the computation of (6.1) in steps 1 to k, can be bounded in terms of k, $\|A\|$, and the characteristics of the computer used, using the theory given in Section 2.

The matrix equations of the form (6.7) were then combined to give the practical equivalent of (4.1)

$$AV = V(T + B) + E + \delta C - \delta V, \qquad (6.8) \quad \boxed{\texttt{eq:6.8}}$$

where B is the upper triangular matrix of elements b_{ij} used in the re-orthogonalization (6.3), and δC describes the errors occurring in (6.3), (6.5) and (6.6). Unfortunately δC depends on the elements of B, while these depend on the off-diagonal elements of $V^T V$, among other factors, and so in order to find bounds on the error matrices in (6.8) it was necessary to bound the departure from orthogonality of the computed vectors v_j.

The k by k matrix δU was defined to be the strictly upper triangular part of $V^T V$, while

$$\delta u_{k+1} \equiv V^T v_{k+1}.$$

The particular process analysed used double length accumulation of inner products in (6.2), (6.4), and (6.6) for increased accuracy, and under certain restrictions on the size of the problem and accuracy of the computer (Paige, 1970b, (4.2)) it was shown that

$$\|\delta u_{k+1}\|_2 < 2.02\epsilon\,(1 + \|\delta U\|_2) + 1.01\left[2\|\delta U\|_2 + \left(7 + k^{3/2}\right)\epsilon\right]\|b_k\|_2/t_{k+1,k} \qquad (6.9) \quad \boxed{\texttt{eq:6.9}}$$

where b_k is the kth column of B and ϵ is as in (2.3). As a result of this no a priori bounds can be obtained for the columns of δU or B without setting a lower bound on the allowable normalization factors $t_{j+1,j}$ in (6.6), in fact an example was given showing how orthogonality could actually be lost when several very small normalizing factors occurred in a sequence. The normalizing factors in the example were not very much greater than the machine precision, and would certainly have been taken as zero in any reasonable algorithm and the process curtailed, or a new vector chosen which was orthogonal to the previous vectors. What the analysis and the example pointed out however was that re-orthogonalization is certainly not good enough to produce new vectors orthogonal to the previous ones in such extreme cases. As well as this the tests to determine when the vector $t_{k+1,k}v_{k+1}$ can be considered negligible have always been somewhat arbitrary in the past, whereas the analysis allows rigorous error bounds to be obtained when particular stopping criteria are chosen. A test which struck a reasonable balance between stopping the process too early and allowing it to go on too long with possible resulting errors was the following:–
CRITERION: stop the process for the first value of k for which

$$t_{k+1,k} < k\|b_k\|_2, \quad \text{or} \quad k = n. \qquad (6.10) \quad \boxed{\texttt{eq:6.10}}$$

A priori error bounds were then found for the process, and it was shown that if the stopping criterion was triggered after k steps then

$$(A + \delta P)V = VT$$

where

$$\|\delta P\|_2 < \left[40 + 60k^{\frac{1}{2}} + 60k + \left(4 + 3k^{\frac{1}{2}} + 1.2k\right) m\beta\right] k^{\frac{1}{2}}\epsilon\|A\|_2$$

where A has at most m non-zero elements per row, and as in (2.9), $\beta = \||A|\|_2/\|A\|_2$. Thus if $Tz = \mu z$ then $(A + \delta P)y = \mu y$ where $y = Vz$, and μ and y are an eigenvalue-eigenvector pair belonging to the perturbed matrix $A + \delta P$. The bound on δP is of course very pessimistic as it assumed $t_{k+1,k} = k\|b_k\|_2$ at every step, whereas $t_{k+1,k}$ is almost always far greater than this.

If the process is stopped before the criterion is triggered, then since it is shown that

$$\|\delta u_j\|_2 < 10j^{\frac{1}{2}}, \quad j = 2, 3, \ldots, k,$$

it is clear that the vectors $v_1, \ldots, v_k$ are satisfactorily orthogonal, and using this and bounds on the errors in (6.8), the possible errors in the approximate Lehmann intervals could be found as indicated in (Paige, 1969b, Section 5). However it was mentioned in Section 5 here that the basic intervals are usually sufficient, and thus it can be said that the interval

$$|\mu - \lambda| < \|(A - \mu I)Vz\|_2/\|Vz\|_2$$

contains an eigenvalue of A for any μ and z. But from (6.8) if $Tz = \mu z$ then

$$\|(A - \mu I)Vz\|_2 < \|Ez\|_2 + \|VB + \delta C - \delta V\|_2\|z\|_2,$$

where it can be shown that

$$\|VB + \delta C - \delta V\|_E < \left(35 + 31k^{\frac{1}{2}} + 3.6m\beta\right) k^{\frac{1}{2}}\epsilon\|A\|_2,$$

while

$$\begin{aligned}\|z\|_2 &< \|(V^TV)^{-1}V^T\|_2\|Vz\|_2 \\ &< 1.024\|Vz\|_2\end{aligned}$$

since it can be shown that

$$\|V\|_2 < 1.008, \quad \|(V^TV)^{-1}\|_2 < 1.015$$

with the given restrictions on the size of the problem. As a result it can be said that at least one eigenvalue λ of A satisfies

$$|\mu - \lambda| < 1.03\left[|\zeta_k t_{k+1,k}|/\|z\|_2 + \left(35 + 31k^{\frac{1}{2}} + 3.6m\beta\right) k^{\frac{1}{2}}\epsilon\|A\|_2\right]$$

where ζ_k is the kth element of z. Of course here it is assumed that μ and z are an accurate eigenvalue-eigenvector pair for T, in practice the error involved in computing these (Wilkinson, 1965) should be considered as well, for completeness.

6.2 Time and Storage for the Lanczos Process

sec:6.2

To obtain an idea of the time for the jth step of the process described by (6.1) to (6.6) let s and d denote the times for the single and double-length accumulation of products as described in Section 2.4. If double-length accumulation of inner-products is used in steps (6.2), (6.4) and (6.6), and the n by n matrix A has only nm non-zero elements, then the time can be broken down as follows

Equation	To Form	Time in μ Sec
	Av_j	nms
(6.2)	$t_{j-1,j}$ and t_{jj}	$2nd$
(6.1)	c_j	$2ns$
(6.4)	$b_{1j}, b_{2j}, \ldots, b_{jj}$	jnd
(6.3)	w_j	jns
(6.6)	$t_{j+1,j}$	nd + a square root
(6.5)	v_{j+1}	ns + a division
(6.10)	$t^2_{j+1,j} < j^2 b_j^T b_j$?	$(j+2)s$ + a test.

The basic part of the computation takes $(m+3)ns+3nd$ + square root + division, per step, while the re-orthogonalization and stopping criterion takes $jn(s+d)+(j+2)s$ + test, per step, and is likely to be the dominant part of the computation. Now it was seen in Section 2.4 that for the Atlas computer $d < 4s$ certainly, so taking $d = 4s$ and summing over the first k steps the total time is about

$$nk\,(m + 5k/2 + 18)\,s \quad \mu \text{ Sec.}$$

ignoring the square roots, divisions, and tests. If single length accumulation is used throughout this time reduces to about

$$nk(m+k+8)s \quad \mu \text{ Sec.}$$

This analysis ignores any transfers of vectors that might be required, and assumes that Av_j could be formed with only nm operations (an operation being one multiplication and one addition). If only the non-zero elements of A are stored, for example in rows with their appropriate column indices, then no more operations are needed and the usage of store is more economic too. If the matrix has a very well defined structure and relatively few different elements, then negligible storage may be needed to give the full information required to form Av for any vector v, again requiring only nm operations. Thus at worst only nm ordinary storage locations plus nm index locations are required to store A, and often much less.

In the computation all the values t_{ij} and vectors v_j must be kept, while in the kth step only one n-vector is needed for all of Av_k, c_k, and w_k, while one k-vector can be used for b_k, giving a total number of storage locations of about

$$nk + n + 4k + \text{ storage of } A.$$

If the v_j are kept in a larger slow store then the computation can very simply be programmed to require three vectors of dimension n in the fast store (plus storage of A) and bring down and replace each of $v_1, \ldots, v_j$ once only in the jth step.

From this analysis it can be seen that the Lanczos process requires more store and more computation with each successive step, and even if m were negligible in comparison to n and only single length accumulation was used, n steps of the process would require over n^3 operations and over n^2 storage locations, making it a poor performer for this full reduction when compared with Householder's method.

6.3 Comparison with Householder's Method

sec:6.3

Although the symmetric Lanczos algorithm with re-orthogonalization is unsatisfactory for n steps, its simplicity and initial small computation per step makes it very effective for a limited number of steps, and as often only some fraction of n steps is required to give a few eigenvalues accurately, the Lanczos process will be compared with the equivalent Householder method in such a situation.

Householder (1964, Section 6.4) showed that if the elementary Hermitian matrix $P_0 = I - 2vv^T$, $v^T v = 1$, is chosen so that if $v_1^T v_1 = 1$ then $P_0 v_1 = \pm e_1$, e_1 the first column of the identity, then the Householder algorithm applied to $A_0 \equiv P_0 A P_0$ gives the same result as the Lanczos algorithm starting with v_1, except for possible changes of sign. With this terminology (4.1) becomes

$$A_0 V_0 = V_0 T + E_0, \quad V_0^T E_0 = 0, \quad V_0^T V_0 = I$$

where

$$V_0 \equiv P_0 V, \quad E_0 \equiv P_0 E,$$

so that V_0 has for its first column $\pm e_1$. Now for $k-1$ exact steps of the Householder tri-diagonalization on A_0

$$A_0 \Big[\underset{n\times k}{P} \;,\; \underset{n\times n-k}{Q}\Big] = \Big[P \;,\; Q\Big] \begin{bmatrix} T_k & C^T \\ C & B \end{bmatrix} \begin{matrix} \}\, k \\ \}\, n-k \end{matrix}$$

where $[P, Q] = P_1 P_2 \cdots P_{k-1}$ is the product of the elementary Hermitians of the process, T_k is tri-diagonal, and $C = [0, \ldots, 0, c]$. As a result

$$A_0 P = P T_k + QC, \quad P^T P = I, \quad P^T Q = 0$$

and P has for its first column e_1. Then (see for example Wilkinson, 1965, p. 352) as long as $t_{j+1,j} \neq 0$, $j = 1, \ldots, k-1$,

$$P = V_0 D, \quad T_k = D^T T D, \quad QC = E_0 D$$

where D is a diagonal matrix with elements of ± 1. Thus for exact arithmetic all the information that was found from k steps of the Lanczos process and used in obtaining eigenvalue bounds could also have been found from $k-1$ steps of Householder's method applied to A_0. As a result all the interesting properties of convergence and all the eigenvalue intervals discussed in Sections 4 and 5 apply equally well to such an incomplete Householder tri-diagonalization.

The accuracy of Householder's method is well known, it remains to give an operation count for the initial transformation of A followed by $k-1$ ordinary Householder steps. By examining the algorithm suggested by Wilkinson (1965, p. 292) it can be seen that without taking any advantage of sparsity the initial transformation takes about $2n(n+2)$ operations while the jth step takes about $2(n-j)^2+4(n-j)$ operations, and each step requires one square root. Summing for $k-1$ steps gives a total of

$$\frac{k}{3}(6n^2+18n+7-6nk+2k^2-9k)$$

operations plus k square roots. The Lanczos process thus takes less computation per step initially, even if $m=n$. The difference is that the amount of computation per step diminishes as the Householder algorithm progresses, whereas for the Lanczos algorithm it increases, as a result there is a break-even point at which both (uncompleted) algorithms take the same time. By putting $k=n/3$ in both counts for the Lanczos method with single-length accumulation and for the Householder method it is seen that the first is faster in this case even taking $m=n$, as long as $n \geq 42$. For the Lanczos method using a double-length accumulation which takes four times as long as single-length, the break-even point for $m=n$ is at about $k=n/5$, as long as $n \geq 100$.

Thus even for full matrices the Lanczos process with re-orthogonalization has an initial advantage in speed, though not store, over Householder's method, and could be used if only a few steps were needed, however such a situation is hard to imagine. For a very sparse matrix the speed advantage of the early steps of the Lanczos process is greatly enhanced, and the storage requirements are also much less. A means of applying Householder's algorithm that takes some advantage of sparsity can be devised but it takes about twice as much storage (apart from the storage for A), and about as much computing time, as the Lanczos method with double length accumulation given here, while its numerical stability is uncertain without an analysis.

Now in most large sparse matrix problems the complete eigensolution is not wanted, instead several of the extreme eigenvalues and sometimes their eigenvectors are usually required. For example in the matrix formulation of a vibrational problem the eigenvalues correspond to frequencies of vibration, and perhaps only 5 or 6 of the lowest frequencies will be required, possibly together with their modes of oscillation, or eigenfunctions. In many cases a vector v_1 with substantial projections on the subspaces of the wanted eigenvalues will be readily available as well, thus ensuring even faster convergence to the desired eigenvalues and eigenvectors than would be achieved with an arbitrary initial vector. In such problems the symmetric Lanczos process with re-orthogonalization has definite advantages over Householder's method in store, speed, and simplicity, and in (Paige, 1970b, Section 5) an example is given of the rapid convergence of the process for some extreme eigenvalues of a 300 by 300 large sparse symmetric matrix.

The approximate Lehmann intervals centred on the eigenvalues of T are tabulated for this example (ibid., Table 1). The algorithm for finding these suggested by Lehmann (1966) was first used, but because of rounding errors occurring in the computation of these intervals, this algorithm never gave an interval of half-width less than 10^{-6}, even when both an eigenvalue and its eigenvector had converged to machine accuracy. This is quite understandable as, with rounding errors, Δ^2 in (5.32) is liable to be in error by $\|(T-\mu I)^2+E^TE\|\cdot O(\epsilon)$, and so taking the square root will magnify this, making it impossible to find accurate small intervals. However when the corresponding eigenvector z in (5.32) is also found to machine accuracy by inverse iteration, by using for example one of the algorithms given by Martin and Wilkinson (1967), then $y = Tz - \mu z$ can be computed using double-length accumulation of inner-products, and forming

$$\frac{\left(\sum_{i=1}^{k} \eta_i^2 + t_{k+1,k}^2 \zeta_k^2\right)}{\left(\sum_{i=1}^{k} \zeta_i^2\right)}$$

gives Δ^2 very accurately, where η_i and ζ_i are the elements of y and z respectively. The error in this is $O(\epsilon^2)$, and so taking the square root gives Δ with an error $O(\epsilon)$, these then are the values B in the given table.

It was seen in the operation count for the Lanczos process that if a slow double-length accumulation procedure was used the process was considerably slowed down, but without the occasional use of double-length there is likely to be a small loss in accuracy. For instance an examination of the error analysis shows that if the number 7 in the inequality (6.9) is replaced by $2n + 2$, then this is very close to the corresponding bound using ordinary accumulation. Then it can be seen that the same stopping criterion could be used with a possibly greater loss of orthogonality and loss of over-all accuracy, or

$$t_{k+1,k} < kn\|b_k\|_2$$

could be used as a stopping criterion, ensuring about as much orthogonality as previously, but with a still possibly greater error if the process was forced to stop by this criterion.

As cancellation is the basic cause of loss of orthogonality of c_j to $v_1, \ldots, v_j$ in (6.1), a further saving of time could be made by only re-orthogonalizing when $\|c_j\|_2$ is small compared with $\|Av_j\|_2$, this however would require another error analysis to define 'small' accurately, and is moving towards the subject matter of the next section, the Lanczos method without re-orthogonalization.

Finally it should be mentioned again that a disadvantage of both the Householder and Lanczos algorithms used in this incomplete form is that repeated eigenvalues may not be found early on, even if they are among the extreme eigenvalues, this difficulty has already been mentioned in the analysis in Section 4.

Section 7

The Symmetric Lanczos Process without Re-orthogonalization

chp:7

For large sparse symmetric matrices the most obvious practical advantage of the Lanczos algorithm in its simple form is the small amount of computation and storage required per step. If only the eigenvalues are wanted then v_{j-1} and v_j are the only vectors that are needed to form v_{j+1}, and the amount of storage and computation per step remains constant. If eigenvectors are also required then $v_1, v_2, \ldots, v_{j-2}$ etc. may be put in the backing store and brought back one at a time at the end of the algorithm to build up the required eigenvectors; otherwise $v_1, v_2, \ldots$ can be re-formed, used, and discarded, in a second pass of the algorithm once their contributions to the required eigenvectors have been found by solving the eigenvector problem for the tri-diagonal matrix T obtained in the first pass. Another practical advantage of the algorithm is that it requires no estimation of parameters, as do some iterative methods, and it also finds several eigenvalues in one go, rather than just one as does the power method with one vector, the steepest descents method, and the Tchebycheff iteration. Other advantages of the algorithm in theory are the extremely rapid convergence discussed in Section 4 and the possibility of finding useful eigenvalue intervals discussed in

Section 5. These last two advantages suggest the use of the algorithm as an iterative method.

However the Lanczos algorithm in its simple form appears to have been largely discarded as a useful method, and this is almost certainly because of the severe loss of orthogonality that occurs when any significant cancellation takes place. In the work for this thesis this loss of orthogonality was first examined and it was found that it could be bounded in terms of the computed elements of the tri-diagonal matrix. However in computations comparing the accuracy of computed eigenvalues with both the loss of orthogonality and the bounds on this, it was found in many cases that several eigenvalues converged to great accuracy despite complete loss of orthogonality. Startling examples of this occurred when the number of steps far exceeded the dimension of the matrix, as in such cases it often happened that repeated eigenvalues of the tri-diagonal matrix corresponded accurately with single eigenvalues of the original matrix.

As a result of the above computations attention was switched to trying to find out under what circumstances convergence occurred and attempting to understand why this was possible despite the loss of orthogonality. There are several possible minor variants of the basic algorithm, and one interesting result shows that the particular algorithm that appears initially most satisfactory has a basic flaw which negates its usefulness, particularly when close eigenvalues are sought. Instead an even more simple algorithm is seen not to suffer from this flaw, and is an extremely useful algorithm for large sparse matrices when used iteratively.

This section will be devoted to analyzing why the most obvious algorithm fails, and, since this analysis suggests that the more simple algorithm will not suffer in the same manner, obtaining initial expressions for the errors in this second algorithm. It will be left to Section 8 to show why in fact this second algorithm is so remarkably accurate.

7.1 The Basic Method and the Different Possible Algorithms

sec:7.1

First the method will be presented in a slightly different light to that given earlier. Suppose in the jth step it is intended to find the component of Av_j which is orthogonal to $v_1, \ldots, v_j$, and that these are themselves orthogonal. That is, find $t_{1j}, \ldots, t_{jj}$ in

$$t_{j+1,j} v_{j+1} = Av_j - t_{jj} v_j - \ldots - t_{ij} v_i - \ldots - t_{1j} v_1$$

so that

$$t_{j+1,j} v_i^T v_{j+1} = 0, \quad i = 1, \ldots, j.$$

Note that this will also ensure that $\|t_{j+1,j} v_{j+1}\|_2$ is minimal in the above expression, and this is why Lanczos called it the method of minimized iterations.

In order to satisfy these conditions it is necessary to have

$$\begin{aligned} t_{ij} v_i^T v_i &= v_i^T A v_j = v_j^T A v_i \\ &= v_j^T \left(t_{i+1,i} v_{i+1} + t_{ii} v_i + \ldots + t_{1i} v_1 \right) \\ &= 0 \quad \text{if} \quad i < j - 1, \end{aligned}$$

since $v_1, \ldots, v_j$ are orthogonal. As a result there are only three coefficients required per step. Now because of the complexity involved, some of the following work will be more easily followed when written using the small notational change

$$\begin{aligned} &\delta_1 \equiv 0; \quad \delta_j \equiv t_{j-1,j}, \quad j = 2, 3, \ldots; \\ &\gamma_j \equiv t_{jj}, \quad \beta_{j+1} \equiv t_{j+1,j}, \quad j = 1, 2, \ldots. \end{aligned}$$

With this notation the coefficients become

$$\delta_j = v_{j-1}^T A v_j / v_{j-1}^T v_{j-1}, \qquad j > 1 \qquad (7.1) \quad \boxed{\texttt{eq:7.1}}$$

$$= \beta_j v_j^T v_j / v_{j-1}^T v_{j-1}, \qquad j > 1 \qquad (7.2) \quad \boxed{\texttt{eq:7.2}}$$

$$\gamma_j = v_j^T A v_j / v_j^T v_j, \qquad j \geq 1 \qquad (7.3) \quad \boxed{\texttt{eq:7.3}}$$

and the equation for forming and normalizing the next vector becomes

$$\beta_{j+1} v_{j+1} = A v_j - \gamma_j v_j - \delta_j v_{j-1}. \tag{7.4}$$

eq:7.4

Thus there are certainly two possible algorithms given by the choice between (7.1) and (7.2), and these will be denoted by A1 and A2 respectively (A for Algorithm). A1 and A2 are the same in theory but behave markedly differently in practice. Now it is interesting to note that the so-called 'conjugate gradient' algorithms for the solution of linear equations problems arose from the Lanczos process for the eigenproblem, (Lanczos, 1950, p. 256; 1952), and that in theory coefficients corresponding to those in (7.1) to (7.4) can be derived directly from the coefficients occurring in the different 'conjugate gradient' algorithms for the solution of the matrix equation $Ax = v_1$, taking as the starting vector $x_0 \equiv 0$ (see also for example Engeli et al., 1959, p. 45). Reid (1970) has summarized and compared these different possible solution of equations algorithms, and the most economic, called here A3, will be compared in Table 2 for economy per step with A1 and A2 above. A comparison will also be made with one step of the Tchebycheff iteration for finding an extreme eigenvalue and its eigenvector. The time and storage required in the matrix-vector product computation will be omitted in the comparison as these will be the same for each algorithm; however unless the matrix is extremely sparse these may well dominate the computations. In A1 and A2 it will be assumed that $\beta_{j+1} = 1$ in (7.4), this lack of normalization is unimportant on floating point arithmetic computers except that it may lead to exponent overflow in some computations.

The small storage for A2 is possible because v_{j-1} in (7.4) may be overwritten by $Av_j - \delta_j v_{j-1}$ element by element as Av_j is being formed, $v_j^T A v_j$ being accumulated at the same time. A2 is thus the most economic of the Lanczos algorithms and indeed compares well with the Tchebycheff iteration. There is also a variant of A2 that will be commented on later.

Algorithm	Number of vectors stored	Vector inner-products	Scalar by vector products
A1	3	3	2
A2	2	2	2
A3	4	2	3
Tchebycheff	2	0	1

tab:2

Table 2: Time and storage comparison of algorithms.

If A is very large and the full information required for forming Av_j occupies negligible store then the number of vectors stored per step may be very important. As well as this an algorithm written for the main purpose of finding eigenvalues is hopefully more accurate than a more complicated algorithm which was written for another purpose and which gives the eigenvalues as an afterthought. For these reasons only the two most economic algorithms A1 and A2 have been analyzed, although initial computations using the conjugate gradient algorithms suggest that these may also be viable.

Wilkinson (1965, p. 395) discards (7.2) in favour of (7.1), and this is perfectly reasonable as in the first case he is not considering sparse matrices, so that the difference in economy is negligible, while secondly he is considering re-orthogonalization, and in this case the method for computing δ_j is probably somewhat arbitrary. His choice perhaps follows from the theoretical observation that if v_{j-1} and v_j are orthogonal in (7.1) to (7.4) then v_{j+1} will be orthogonal to these if (7.1) is used, whereas the use of (7.2) requires that v_j be orthogonal to $v_1, \ldots, v_{j-1}$ to achieve the same result. It is then all the more astounding in the light of this argument that whereas A1 turns out to have a significant flaw in the presence of rounding errors A2 is in fact extremely reliable when used iteratively. Thus the most economic algorithm turns out to be the most accurate as well, a parallel result to that found by Reid (1970) for the different ‘conjugate gradient’ algorithms.

7.2 Initial Rounding Error Analysis

sec:7.2

The analysis will be for ordinary accumulation of inner products and as in (2.3) α and ϵ will represent real numbers satisfying

$$|\alpha - 1| \le u, \quad |\epsilon| \le (1.01)u, \quad u \le 0.001 \tag{7.5}$$

eq:7.5

where u is a machine constant, and the conventions given in Section 2 will be used without further comment. The matrix A will be real symmetric n by n with at most m non-zero elements per row, and such that

$$\|\,|A|\,\|_2 = \beta\|A\|_2;$$

and it will simplify the analysis to assume that

$$(2n+1)\epsilon,\ \ m\beta\epsilon < 0.01. \tag{7.6}$$

eq:7.6

Since only the 2-norm will be used the subscript 2 will be omitted in the future.

It will be assumed that there is no normalization in (7.4), that is, $\beta_{j+1} = 1$ always, as this will make the more complex parts of the analysis easier to follow; the analysis will then also hold rigorously for any normalization that does not introduce rounding errors, and in fact it is almost accurate for any normalization since the rounding errors thus introduced will be seen to be insignificant, the important errors in the evaluation of (7.4) occurring as a result of the subtractions on the right hand side.

Throughout the analysis v_j, γ_j, δ_j, etc. will represent the actually computed values and relations will be found between these. The same set of symbols will be used for both algorithms A1 and A2, the distinction being made in the text.

Now from (2.7), (2.9) and (7.6) it follows that

$$\left.\begin{array}{l} fl(Av_j) = A_j v_j \quad \text{where } A_j \equiv A + \delta A_j, \\ \|\delta A_j\| \le m\beta\epsilon\|A\|, \quad \text{so } \|A_j\| < 1.01\|A\| \end{array}\right\} \qquad (7.7) \quad \boxed{\texttt{eq:7.7}}$$

and using (2.5) it is seen that

$$fl(v_i^T fl(Av_j)) = v_i^T D(\alpha^n) A_j v_j = v_i^T A_j v_j + n\epsilon|v_i^T||A_j v_j| \qquad (7.8) \quad \boxed{\texttt{eq:7.8}}$$

so in (7.3) for A1 and A2

$$|\gamma_j| \le \alpha^{2n+1}\|A_j v_j\|/\|v_j\| < 1.03\|A\| \qquad (7.9) \quad \boxed{\texttt{eq:7.9}}$$

and in (7.1) for A1

$$|\delta_j| \le \alpha^{2n+1}\|A_j v_j\|/\|v_{j-1}\| < 1.03\|A\|\|v_j\|/\|v_{j-1}\|. \qquad (7.10) \quad \boxed{\texttt{eq:7.10}}$$

If the right hand side of (7.4) is evaluated from left to right the computational equivalent for $j > 1$ is

$$\begin{aligned} v_{j+1} &= D(\alpha)\left\{D(\alpha)\left[A_j v_j - \gamma_j D(\alpha) v_j\right] - \delta_j D(\alpha) v_{j-1}\right\} \\ &= A_j v_j - \gamma_j v_j - \delta_j v_{j-1} - \delta v_j \end{aligned} \qquad (7.11) \quad \boxed{\texttt{eq:7.11}}$$

where

$$\left.\begin{array}{ll} \delta v_j & \equiv 2\delta_j D(\epsilon) v_{j-1} + \left[3\gamma_j D(\epsilon) - 2D(\epsilon)A_j\right] v_j, \quad j > 1, \\ \delta v_1 & \equiv \left[2\gamma_1 D(\epsilon) - D(\epsilon)A_1\right] v_1, \quad (\delta_1 \equiv 0) \end{array}\right\} \qquad (7.12) \quad \boxed{\texttt{eq:7.12}}$$

so that from (7.9)

$$\|\delta v_1\| < 3.02\epsilon\|A_1 v_1\| < 3.06\epsilon\|A\|\|v_1\|. \qquad (7.13) \quad \boxed{\texttt{eq:7.13}}$$

So far (7.11) to (7.13) are true for both A1 and A2, but now for A1 using (7.9) and (7.10)

$$\|\delta v_j\| < 7.05\epsilon\|A_j v_j\| < 7.2\epsilon\|A\|\|v_j\| \qquad (7.14) \quad \boxed{\texttt{eq:7.14}}$$

while for A2

$$\|\delta v_j\| < 2\epsilon\delta_j\|v_{j-1}\| + 5.03\epsilon\|A_j v_j\|. \tag{7.15}$$

eq:7.15

Now by using these results it will be possible to bound the loss of orthogonality between v_j and v_{j+1}, and this will lead to some significant conclusions. First from (7.11) it can be seen that for both A1 and A2

$$\left.\begin{aligned} v_j^T v_{j+1} &= -\delta_j v_{j-1}^T v_j + \theta_j, \quad j > 1 \\ v_1^T v_2 &= \theta_1 \\ \text{where}\quad \theta_j &\equiv v_j^T A_j v_j - \gamma_j v_j^T v_j - v_j^T \delta v_j, \quad j \geq 1 \end{aligned}\right\} \tag{7.16}$$

eq:7.16

and making use of (7.8) and (2.5), the computation of (7.3) gives

$$\begin{aligned} \alpha^{n+1}\gamma_j v_j^T v_j &= v_j^T D(\alpha^n) A_j v_j \\ \therefore \quad \theta_j &= (n+1)\epsilon\gamma_j v_j^T v_j - n\epsilon |v_j^T||A_j v_j| - v_j^T \delta v_j \end{aligned}$$

so that using (7.9) and (7.13) for A1 and A2

$$|\theta_1| < 2.02(n+2)\epsilon\|A_1 v_1\|\|v_1\| < 2.05(n+2)\epsilon\|A\|\|v_1\|^2 \tag{7.17}$$

eq:7.17

while for $j > 1$ for A1, using (7.9) and (7.14)

$$|\theta_j| < 2.02(n+4)\epsilon\|A_j v_j\|\|v_j\| < 2.05(n+4)\epsilon\|A\|\|v_j\|^2 \tag{7.18}$$

eq:7.18

and for $j > 1$ for A2, using (7.9) and (7.15)

$$|\theta_j| \leq 2\left[\delta_j\|v_{j-1}\| + 1.01(n+3)\|A_j v_j\|\right]\epsilon\|v_j\|. \tag{7.19}$$

eq:7.19

Thus it can be seen from (7.16) that for both algorithms, if in the following the product term is taken to be unity for $r > j$,

$$v_j^T v_{j+1} = \sum_{i=1}^{j} (-1)^{j-i}\theta_i \prod_{r=i+1}^{j} \delta_r \tag{7.20}$$

eq:7.20

where the θ_j have been bounded. This expression will now be used to indicate a deficiency in A1, whereas later the same expression will be used to indicate some excellent properties of A2. The two rather different results are caused by the small differences in the δ_i in the two algorithms.

7.3 Failure of the Obvious Algorithm (A1)

sec:7.3

The way to exhibit the failure of a given algorithm is to produce a numerical example of this failure, and in fact a computational example of the failure of A1 will be given in Section 9. Here however an attempt will be made to give some understanding of why A1 fails, as it was this that led to the close examination of A2 and the eventual revelation of its remarkable properties.

As insight only, rather than proof, is being given here, there will be little attempt at rigour, and the rather clumsy notation $O(\epsilon)$ will be used, where

$$a = O(\epsilon)$$

means that $|a| = cu$, where $|\epsilon| < 1.01u$ as in (2.1), and c is some positive constant not too different from unity, say $0.1 < c < 10$. The notation $O(\epsilon)$ is used rather than the slightly more sensible notation $O(u)$ because the error bounds have been given in terms of ϵ throughout, as explained in Section 2.

Now it has been shown for A1 that (7.20) holds with

$$|\theta_j| < \theta \|v_j\|^2, \quad \theta \equiv 2.05(n+4)\epsilon\|A\|,$$

and if it were true that $\delta_i = \|v_i\|^2/\|v_{i-1}\|^2$ for $i = 2, \ldots, j$ then it would follow that

$$|v_j^T v_{j+1}| < j\theta\|v_j\|^2 \tag{7.21}$$

eq:7.21

showing that no matter what cancellation had occurred earlier, if v_j and v_{j+1} were comparable in size then the orthogonality of these two would be commendable, and could be used to establish the accuracy of the algorithm. Unfortunately the values of δ_i computed using (7.1) do not obey this simple relation, and can be greatly different for very small δ_i, in fact negative values are sometimes encountered. In practice it is found that orthogonality in the sense of (7.21) is lost when any δ_i approaches $O(\epsilon)\|A\|^2$ and as this corresponds to an off-diagonal element in the corresponding symmetric tri-diagonal matrix of $O(\epsilon^{\frac{1}{2}})\|A\|$ it may well be premature to curtail the iteration. Again practice suggests that if orthogonality is lost between v_j and v_{j+1} then it is never regained for any later consecutive pair, and the resulting eigenvalues wander about and never converge as k increases; this is quite understandable when it is seen in (7.16) that the orthogonality in any step is directly dependent on that in the previous step.

The departure of δ_j from its desired value can be understood as follows. Suppose that v_{j-2}, v_{j-1}, Av_{j-1}, γ_{j-1} and δ_{j-1} are known completely accurately, and then ordinary rounding errors come into play in the computation of the next vector, so that instead of the error free vector v_j, the vector $u_j = v_j + w_j$ is obtained, where from (7.11) and (7.14) the best that can be said of w_j is that $\|w_j\| \leq 7\epsilon\|Av_{j-1}\|$. As a result even if there are no further errors in the computation of (7.1) the following approximation to δ_j is obtained

$$\overline{\delta}_j = v_{j-1}^T Au_j / v_{j-1}^T v_{j-1} = \delta_j + v_{j-1}^T Aw_j / v_{j-1}^T v_{j-1}$$

so that

$$|\overline{\delta}_j - \delta_j| \leq 7\epsilon\|Av_{j-1}\|^2/\|v_{j-1}\|^2 \leq 7\epsilon\|A\|^2. \qquad (7.22)$$ eq:7.22

As a value of δ_j approaching $O(\epsilon^2)\|A\|^2$ is quite possible and would be considered a satisfactory value on which to curtail the algorithm, the possible relative error in $\overline{\delta}_j$ can obviously be huge, causing drastic departures from (7.21).

The fact that the absolute error in the value of δ_j computed by (7.1) may be $O(\epsilon)\|A\|^2$ also has a significant effect on the accuracy of the off-diagonal elements of the corresponding symmetric tri-diagonal matrix whose eigenvalues are meant to be those of A, even when $\delta_j \gg O(\epsilon)\|A\|^2$, for then if

$$\overline{\delta}_j = \delta_j + O(\epsilon)\|A\|^2$$

it follows that

$$\overline{\delta}_j^{\frac{1}{2}} = \delta_j^{\frac{1}{2}} \left[1 + O(\epsilon)\|A\|^2/\delta_j\right]$$

and the corresponding absolute error is $O(\epsilon)\|A\|^2/\delta_j^{\frac{1}{2}}$. This is satisfactory while δ_j is not very much smaller than $\|A\|^2$, but is progressively worse for smaller values of δ_j, and the symmetric tri-diagonal matrix can have off-diagonal elements in error by as much as $O(\epsilon^{\frac{1}{2}})\|A\|$. Any negative values of δ_j would of course have to be set to zero, and again this would give an error $O(\epsilon^{\frac{1}{2}})\|A\|$.

In practice then even when loss of orthogonality does not occur in A1 the resulting symmetric tri-diagonal matrix is likely to be in error. However in such cases if the distinct eigenvalues of A are well separated it turns out that these are given quite accurately by A1. Close eigenvalues may well be in error by up to $O(\epsilon^{\frac{1}{2}})\|A\|$ though, and so the method is unreliable unless only this reduced accuracy is required, in which case the process would be curtailed whenever $\delta_j \leq O(\epsilon)\|A\|^2$.

Unfortunately many computations and a great deal of analysis were devoted to trying to explain these particular properties of A1 before the above simple answer was obtained. In particular some work on the sensitivity of the eigenvalues of Hermitian matrices (Paige, 1970a) was developed with this one aim; this does explain why well separated roots are well conditioned and close roots poorly conditioned but does not, on its own, explain why A1 converges even as well as it does.

The analysis just given for A1 immediately suggested the superior properties of A2, since it is apparent that (7.21) will be closely approximated in A2, thus suggesting that the near-orthogonality of any two consecutive vectors of comparable size will always be maintained. As well as this the equivalent of (7.22) for A2 is given by

$$\begin{aligned} \bar{\delta}_j^{\frac{1}{2}} &= \|v_j + w_j\|/\|v_{j-1}\|, \quad \|w_j\| \leq 7\epsilon\|Av_{j-1}\| \\ \therefore \quad \left|\bar{\delta}_j^{\frac{1}{2}} - \delta_j^{\frac{1}{2}}\right| &\leq 7\epsilon\|A\| \end{aligned}$$

and since an eigenvalue accuracy of $O(\epsilon)\|A\|$ is the best that can be hoped for anyway, the main inaccuracy that upset A1 does not occur in A2.

As a result of its shortcomings A1 will not be analyzed further, the remainder of the thesis being directed solely towards understanding A2. Practical experience suggests that the eigenvalues obtained using A2 always converge, and to within an accuracy of $f(j)\epsilon\|A\|$, $f(j)$ being a function of the number of steps so far carried out. The main problem then is to prove if this is true and to find a useful expression for $f(j)$. It turns out that the accuracy of the algorithm relies heavily on the bound on $v_j^T v_{j+1}$, at which point it should be noted that the maintenance of orthogonality of consecutive vectors produced by A2 has not been rigorously proven in these previous pages, as δ_j in (7.2) and so θ_j in (7.19) have not been satisfactorily bounded; all that has been shown is that A2 does not suffer the drastic loss of accuracy in one step that A1 can. The bounding of δ_j will be given in the remainder of this section. This is not trivial as it requires a lengthy induction proof, but unfortunately I can see no way of avoiding this. The induction is apparently necessary because of the dependence of the size of v_{j+1} on all the previous vectors, as a result of (7.2) being used for computing δ_j. The remaining important properties of A2 will be studied in Sections 8 and 9.

7.4 Further Analysis of the Most Economic Algorithm (A2)

sec:7.4

It has already been indicated that A2 is not only the most economic algorithm, but also apparently the best. Here some more of the properties that will be needed to prove its accuracy will be given.

It has already been shown for A2 that

$$v_1^T v_2 = \theta_1; \quad v_j^T v_{j+1} = -\delta_j v_{j-1}^T v_j + \theta_j, \quad j > 1, \tag{7.16}$$

where

$$|\theta_1| < 2.05(n+2)\epsilon\|A\|\|v_1\|^2, \tag{7.17}$$

$$|\theta_j| < \left[2.02\delta_j^{\frac{1}{2}} + 2.05(n+3)\|A\|\right]\epsilon\|v_j\|^2, \quad j > 1 \tag{7.19}$$

but unfortunately no useful bound on δ_j is readily available. However it follows from (7.11), (7.8), (7.9), (7.13), (7.16) and (7.17) that

$$\begin{aligned}\|v_2 + \delta v_1\|^2 &= \|A_1 v_1 - \gamma_1 v_1\|^2 \\ &\leq \|A_1 v_1\|^2 - 2\gamma_1 v_1^T(v_2 + \delta v_1) \\ &< 1.03\|A\|^2\|v_1\|^2\left[1 + 4.1(n+2)\epsilon\right]\end{aligned}$$

therefore

$$\|v_2\| < 1.04\|A\|\|v_1\|$$

from (7.6), and so

$$\delta_2^{\frac{1}{2}} = \alpha^{n+\frac{1}{2}}\|v_2\|/\|v_1\| < 1.05\|A\|. \tag{7.23}$$

$\boxed{\texttt{eq:7.23}}$

Thus δ_2 is bounded, and a bound on δ_j may be found by induction, although the proof also involves bounding the terms $|v_{i-1}^T v_{i+1}|$.

Assume that

$$\delta_i^{\frac{1}{2}} < 1.6\|A\|, \quad i = 2, 3, \ldots, j \tag{7.24}$$

$\boxed{\texttt{eq:7.24}}$

then from (7.15) and (7.19)

$$\|\delta v_i\| < 9\epsilon\|A\|\|v_i\|, \quad i = 1, 2, \ldots, j, \tag{7.25}$$

$\boxed{\texttt{eq:7.25}}$

$$\left.\begin{aligned}|\theta_i| &< \theta\|v_j\|^2, \quad i = 1, 2, \ldots, j, \\ \theta &\equiv 2.05(n+4)\epsilon\|A\|.\end{aligned}\right\} \tag{7.26}$$

$\boxed{\texttt{eq:7.26}}$

But the computational equivalent of (7.2) with $\beta_i = 1$ is

$$\delta_i = \alpha fl(v_i^T v_i)/fl(v_{i-1}^T v_{i-1}) = \alpha^{2n+1} v_i^T v_i / v_{i-1}^T v_{i-1} \tag{7.27}$$

$\boxed{\texttt{eq:7.27}}$

so that (7.20) and (7.26) give

$$\begin{aligned} |v_i^T v_{i+1}| &< \theta\left(1+\alpha^{2n+1}+\alpha^{2n+2}+\dots\alpha^{2n+i-1}\right)\|v_i\|^2 \\ &< 1.03 i\theta\|v_i\|^2, \quad i=1,2,\dots,j, \end{aligned} \tag{7.28} \quad \boxed{\texttt{eq:7.28}}$$

as long as $j\epsilon \le .01$ which in view of (7.6) will certainly be the case in any practical computation.

Next from (7.11), (7.27), for $i=3,4,\dots$

$$\begin{aligned} v_{i-1}^T v_{i+1} &= v_i^T\left(Av_{i-1}-\alpha^{2n+1}v_i\right)-\gamma_i v_{i-1}^T v_i + v_{i-1}^T\left(\delta A_i v_i - \delta v_i\right) \\ &= \delta_{i-1}v_{i-2}^T v_i + (\gamma_{i-1}-\gamma_i)v_{i-1}^T v_i + \phi_{i-1} \end{aligned} \tag{7.29} \quad \boxed{\texttt{eq:7.29}}$$

$$\phi_{i-1} \equiv v_{i-1}^T\left(\delta A_i v_i - \delta v_i\right) - v_i^T\left(\delta A_{i-1}v_{i-1}-\delta v_{i-1}\right) - (2n+1)\epsilon v_i^T v_i$$

so that using (7.7), (7.25), (7.24) and (7.27) for $i=3,4,\dots,j$

$$\left.\begin{aligned} |\phi_{i-1}| &< \phi'\|v_{i-1}\|\cdot\|v_i\| \\ &< \phi\|v_{i-1}\|^2, \quad \text{where} \\ \phi &\equiv 4.04\epsilon(2n+m\beta+11)\|A\|^2 \\ \phi' &\equiv (3.3n+2m\beta+20)\epsilon\|A\| \end{aligned}\right\} \tag{7.30} \quad \boxed{\texttt{eq:7.30}}$$

while with a similar argument

$$\left.\begin{aligned} &v_1^T v_3 = (\gamma_1-\gamma_2)v_1^T v_2 + \phi_1, \\ &|\phi_1| < \phi'\|v_1\|\cdot\|v_2\| < \phi\|v_1\|^2. \end{aligned}\right\} \tag{7.31} \quad \boxed{\texttt{eq:7.31}}$$

The results (7.29), (7.30) and (7.31) could be combined immediately with (7.28) to produce the bound (7.37) on $v_{j-1}^T v_{j+1}$, and the reader can go straight to this, but as this tends to be excessive in practice a more refined bound will first be obtained by considering (7.20) in order to indicate an aspect of the stability of the process. Equation (7.29) gives for $i=3,4,\dots$, using the same convention as in (7.20),

$$v_{i-1}^T v_{i+1} = \sum_{r=1}^{i-1}\phi_r \prod_{q=r+1}^{i-1}\delta_q + \sum_{r=1}^{i-1}(\gamma_r-\gamma_{r+1})v_r^T v_{r+1}\sum_{q=r+1}^{i-1}\delta_q$$

but with (7.20) the second term on the right hand side becomes

$$\sum_{r=1}^{i-1}(\gamma_r - \gamma_{r+1})\left[\sum_{p=1}^{r}(-1)^{r-p}\theta_p \prod_{q=p+1}^{r}\delta_q\right]\prod_{q=r+1}^{i-1}\delta_q$$

which on combining the product terms and re-ordering the summation becomes

$$\sum_{p=1}^{i-1}\theta_p\left(\prod_{q=p+1}^{i-1}\delta_q\right)s_{p,i-1}$$

with

$$s_{p,i-1} \equiv \sum_{r=p}^{i-1}(-1)^{r-p}\left(\gamma_r - \gamma_{r+1}\right) \tag{7.32}$$ eq:7.32

so that

$$v_{i-1}^T v_{i+1} = \sum_{r=1}^{i-1}\left(\phi_r + s_{r,i-1}\theta_r\right)\prod_{q=r+1}^{i-1}\delta_q. \tag{7.33}$$ eq:7.33

But from (7.9) it follows that

$$|s_{r,i-1}| < 2.06(i-r)\|A\|, \quad r = 1, 2, \ldots, i-1, \tag{7.34}$$ eq:7.34

while from (7.27)

$$\left.\begin{array}{lll} \prod_{q=r+1}^{i-1}\delta_q & = \alpha^{2n+i-r-1}\|v_{i-1}\|^2/\|v_r\|^2 & , \quad r \le i-2 \\ & = 1 & , \quad r > i-2 \end{array}\right\} \tag{7.35}$$ eq:7.35

so that making use of (7.26) and (7.30)

$$\begin{aligned} \left|v_{i-1}^T v_{i+1}\right| &< \alpha^{2n}\|v_{i-1}\|^2\sum_{r=1}^{i-1}\alpha^{i-r-1}\left(\phi + \theta|s_{r,i-1}|\right) \\ &< 1.021\|v_{i-1}\|^2\left[(i-1)\phi + \theta\sum_{r=1}^{i-1}|s_{r,i-1}|\right] && (7.36) \\ &< 1.06\|v_{i-1}\|^2(i-1)\left(\phi + i\theta\|A\|\right) \\ &< (i-1)^2\phi\|v_{i-1}\|^2, \quad i = 3, 4, \ldots j, && (7.37) \end{aligned}$$ eq:7.36 eq:7.37

since $8.08\theta\|A\| < 2.05\phi$.

In practice (7.34) is found to be a large over-estimate, in fact it often happens that $\gamma_r \doteq \gamma_{r+1}$ for the greater part of a computation with a large matrix, and there is a great deal of cancellation in the $s_{r,i}$, the bound (7.36) then being proportional to $(i-1)\phi$ rather than $(i-1)^2\phi$ as suggested by (7.37). How such properties of the γ_i and δ_i can be incorporated to give very good bounds will become clearer later.

It is still necessary to prove that (7.24) holds for $i = j+1$ in order to complete the induction proof, and for this it is sufficient to use (7.37). Thus, making the assumption that

$$\left.\begin{aligned} & 2.12(j-1)\left(\phi + j\theta\|A\|\right) \le \|A\|^2 \\ \text{i.e.}\quad & 4.4j\left[(3+j)n + 4j + 2m\beta + 18\right]\epsilon \le 1 \end{aligned}\right\} \qquad (7.38) \quad \boxed{\texttt{eq:7.38}}$$

which is a considerably stronger restriction on the size of the problem than previously, then for $i = 3, 4, \ldots, j$,

$$\left.\begin{aligned} & |v_{i-1}^T v_{i+1}| < 0.5\|A\|^2\|v_{i-1}\|^2 \\ \text{and}\quad & |v_i^T v_{i+1}| < 0.1\|A\|\|v_i\|^2. \end{aligned}\right\} \qquad (7.39) \quad \boxed{\texttt{eq:7.39}}$$

Now if u, v and w are real vectors with $w = u + v$, then

$$w^T w = u^T u + 2v^T w - v^T v \le u^T u + 2v^T w,$$

so from (7.11), (7.39), (7.9), (7.24), (7.25), and (7.5), (7.6),

$$\begin{aligned} \|v_{j+1} + \delta v_j\|^2 &= \|A_j v_j - \gamma_j v_j - \delta_j v_{j-1}\|^2 \\ &\le \|A_j v_j\|^2 - 2\left(\gamma_j v_j + \delta_j v_{j-1}\right)^T \left(v_{j+1} + \delta v_j\right) \\ &< \|A\|^2\|v_j\|^2(1.03 + 0.21 + 1.04 + 0.07) \\ \therefore \quad \|v_{j+1}\| &< 1.55\|A\|\|v_j\| \end{aligned}$$

i.e.

$$\begin{aligned} \delta_{j+1}^{\frac{1}{2}} &= \alpha^{n+\frac{1}{2}}\|v_{j+1}\|/\|v_j\| \\ &< 1.6\|A\| \end{aligned} \qquad (7.40) \quad \boxed{\texttt{eq:7.40}}$$

and the induction is complete, (7.24) then certainly being true for all values of j such that (7.38) holds. The other bounds which were derived on the way are thus also rigorous under the same condition. Naturally a more simple proof and a more elegant result would be desirable, however the size of the bound (7.24) is not critical, and in fact it is obvious from the argument that $\delta_j \leq \|A\|^2$ is usually a satisfactory bound, even for values of j far in excess of (7.38); the important point is that such a bound can be found 'a priori', as this is needed to give validity to the important bounds (7.25), (7.28), and (7.37).

Thus we see from (7.28) that unlike A1, orthogonality between successive vectors in A2 is never lost as a result of an earlier cancellation. It can be seen that (7.28) is a really remarkable result, for if we consider the measure of orthogonality

$$\frac{|v_j^T v_{j+1}|}{\|v_j\|\|v_{j+1}\|} \leq 1.02 j\theta \frac{\|v_j\|}{\|v_{j+1}\|} \leq 1.03 j\theta \delta_{j+1}^{-\frac{1}{2}}, \qquad (7.41)$$

eq:7.41

then the bound is purely dependent on the step number and the cancellation in that step. This property will be used in Section 8 to try to show why the algorithm is so good. In fact although the bounds on $v_j^T v_{j+1}$ and $v_{j-1}^T v_{j+1}$ depended directly on the methods used for calculating the elements of the final tridiagonal matrices, it will follow from the nature of the algorithm that all the remaining errors $v_i^T v_{j+1}$, $i < j-1$, are determined by the errors that have already been considered. This, along with other properties of A2, will be explained after some new theory has been developed at the start of Section 8.

Section 8

Error Behaviour & Convergence of the Symmetric Lanczos Process

(Using Algorithm A2)

chp:8

As the most obvious algorithm, A1, was shown to be unstable in the previous section, only A2 will now be considered. First some theory necessary for understanding the error behaviour and convergence properties of the algorithm will be developed. Next a beautiful result will be derived for the loss of orthogonality of the process, this result clearly showing the relation between this loss and the convergence of the eigenvalues of the successive tri-diagonal matrices. Then after some important results on the approximate eigenvectors obtained from the algorithm have been proven, a proof of convergence and accuracy of the algorithm will tentatively be given. This is a very unsatisfactory proof which does little to show the true value of the algorithm, and almost certainly much stronger results on convergence can be proven. However, because the aim of Section 8.7 has not been satisfactorily achieved, several results are developed throughout Section 8 that are not essential to the rest of this thesis but may help to furnish a stronger proof of convergence at some later date. For this I apologize to the reader and hope that the plethora of results presented here does not confuse of irritate him too much.

Because the main interest is the error analysis of the reduction of A to some tri-diagonal form, say T_j, after j steps of the algorithm, no errors will be considered in the computation of the eigensystem of T_j or any other computation which is not part of the central reduction. That is, apart from the reduction everything will be thought of as fully accurate, the effect of any other rounding errors can easily be dealt with using the already well known theory.

Now in A2 $\delta_i = fl(v_i^T v_i / v_{i-1}^T v_{i-1})$, and since this will always be non-negative, it will be convenient to replace δ_i by δ_i^2 throughout the remainder of the thesis.

Again norms will be assumed to be 2-norms unless otherwise indicated, and the subscript 2 will be omitted throughout. The subscript F when it appears indicates the Frobenius norm.

8.1 The Symmetric Matrix of the Process and Related Polynomials

sec:8.1

The $\beta_{j+1} = 1$, γ_j, and δ_j^2 in the computed versions of (7.2) and (7.3) are the elements of a tri-diagonal matrix whose eigenvalues hopefully approximate those of A. However symmetric matrices are more easily handled, and submatrices of the main matrix will also be considered, so the two more general $j-r$ by $j-r$ matrices

$$T_{r,j} \equiv \begin{bmatrix} \gamma_{r+1} & \delta_{r+2}^2 & & \\ 1 & \gamma_{r+2} & \ddots & \\ & \ddots & \ddots & \delta_j^2 \\ & & 1 & \gamma_j \end{bmatrix}, \quad C_{r,j} \equiv \begin{bmatrix} \gamma_{r+1} & \delta_{r+2} & & \\ \delta_{r+2} & \gamma_{r+2} & \ddots & \\ & \ddots & \ddots & \delta_j \\ & & \delta_j & \gamma_j \end{bmatrix} \qquad (8.1) \quad \boxed{\texttt{eq:8.1}}$$

will be defined for $r = 0, 1, \ldots, j-1$ and $j = 1, 2, \ldots, k$. It will be assumed that $\delta_i \neq 0$, $i = 2, 3, \ldots, k+1$. Next defining

$$D_{r,j} \equiv \mathrm{Diag}(1,\ \delta_{r+2},\ \delta_{r+2}\delta_{r+3},\ \ldots,\ \delta_{r+2}\cdots\delta_j) \qquad (8.2) \quad \boxed{\texttt{eq:8.2}}$$

it follows that

$$C_{r,j} = D_{r,j} T_{r,j} D_{r,j}^{-1} \tag{8.3}$$

eq:8.3

so that C and T have the same eigenvalues.

The leading principal minors of $\mu I - T_{r,j}$ and $\mu I - C_{r,j}$ will be denoted by

$$\left.\begin{aligned} &p_{r,r}(\mu) \equiv 1, \quad p_{r,r+1}(\mu) \equiv \mu - \gamma_{r+1}, \\ &p_{r,s}(\mu) \equiv (\mu - \gamma_s) p_{r,s-1}(\mu) - \delta_s^2 p_{r,s-2}(\mu), \quad s = r+2, \ldots, j. \end{aligned}\right\} \tag{8.4}$$

eq:8.4

Thus $p_{r,s}(\mu)$ is a monic polynomial of degree $s - r$ such that

$$p_{r,s}(\mu) = \det(\mu I - T_{r,s}) = \det(\mu I - C_{r,s}), \quad s = r+1, \ldots, j, \tag{8.5}$$

eq:8.5

so $p_{r,j}(\mu)$ is the monic polynomial whose zeros are the eigenvalues of $C_{r,j}$.

8.2 A Useful Theorem on Cofactors

sec:8.2

A fascinating theorem in (Thompson and McEnteggert, 1968) that relates the elements of the eigenvectors of a symmetric matrix to its eigenvalues and the eigenvalues of its principal submatrices will be applied and extended here, and as a proof is not difficult one will now be given.

Consider the k by k matrix $C \equiv C_{0,k}$ in (8.1), this has distinct eigenvalues $\mu_1 > \mu_2 > \ldots > \mu_k$ and an orthogonal matrix of eigenvectors $Y \equiv (y_1, \ldots, y_k)$ such that $CY = YD$, where $D = \mathrm{diag}(\mu_1, \ldots, \mu_k)$, thus

$$p_{0,k}(\mu) = (\mu - \mu_1) \cdots (\mu - \mu_k).$$

Now for any square matrix B, $B \,\mathrm{adj}\,(B) = \det(B) I$, where 'adj' stands for adjugate, so for any scalar μ

$$\begin{aligned} \det(\mu I - C) Y &= Y \det(\mu I - D) \\ = (\mu I - C)\,\mathrm{adj}\,(\mu I - C) Y &= Y (\mu I - D)\,\mathrm{adj}\,(\mu I - D) \\ &= (\mu I - C) Y \,\mathrm{adj}\,(\mu I - D) \end{aligned}$$

$$\therefore \quad \operatorname{adj}(\mu I - C) = Y \operatorname{adj}(\mu I - D) Y^T \qquad (8.6) \quad \boxed{\texttt{eq:8.6}}$$

as long as $\mu I - C$ is nonsingular. But the elements of the matrices on each side of (8.6) are polynomials in μ and there is equality for all but k values of μ, thus (8.6) holds for all values of μ. Now

$$\operatorname{adj}(\mu I - D) = \operatorname{diag}[p_{0,k}(\mu)/(\mu - \mu_1), \ldots, p_{0,k}(\mu)/(\mu - \mu_k)]$$

so that

$$\operatorname{adj}(\mu_i I - C) = f(i) y_i y_i^T \qquad (8.7) \quad \boxed{\texttt{eq:8.7}}$$

where

$$f(i) \equiv \prod_{\substack{r=1 \\ r \neq i}}^{k} (\mu_i - \mu_r).$$

Thompson considered general Hermitian matrices, but because of the tri-diagonal form of C here it is easy to obtain some further interesting results. By equating the (r,s) elements, $s \geq r$, on each side of (8.7) it can be seen from the form of $C \equiv C_{0,k}$ in (8.1) that

$$f(i) y_{ri} y_{si} = \begin{cases} \delta_{r+1} \cdots \delta_s p_{0,r-1}(\mu_i) p_{s,k}(\mu_i), & s > r \\ p_{0,r-1}(\mu_i) p_{r,k}(\mu_i), & s = r \end{cases} \qquad (8.8) \quad \boxed{\texttt{eq:8.8}}$$

where $(-1)^{s-r}$ in the (r,s) cofactor cancels with the sign in the product of the $-\delta_i$.

These relations between the elements of the eigenvector of C corresponding to μ_i and the above principal minors of $\mu_i I - C$ are so simple and elegant that they could not be new, and this derivation may not even be original, nevertheless they are included here for their interest and possible value. In particular for $s > r$

$$\begin{aligned} y_{ri}^2 &= p_{0,r-1}(\mu_i) p_{r,k}(\mu_i) / f(i) \qquad (8.9) \quad \boxed{\texttt{eq:8.9}} \\ &= y_{ri} y_{si} p_{r,k}(\mu_i) / \left[\delta_{r+1} \cdots \delta_s p_{s,k}(\mu_i)\right] \end{aligned}$$

so that

$$y_{si}p_{r,k}(\mu_i) = \delta_{r+1}\cdots\delta_s p_{s,k}(\mu_i)y_{ri}. \qquad (8.10)$$ eq:8.10

(8.9) is the important result for this analysis, but it is also useful to note that an alternate result to (8.10) is available, for suppose $t \geq s > r$ then from (8.8)

$$\begin{aligned} f(i)y_{ri}y_{ti} &= \delta_{r+1}\cdots\delta_t p_{0,r-1}(\mu_i)p_{t,k}(\mu_i), & \\ f(i)y_{si}y_{ti} &= \delta_{s+1}\cdots\delta_t p_{0,s-1}(\mu_i)p_{t,k}(\mu_i), & t > s \\ &= p_{0,s-1}(\mu_i)p_{t,k}(\mu_i), & t = s \\ \therefore \quad y_{ri}p_{0,s-1}(\mu_i) &= \delta_{r+1}\cdots\delta_s p_{0,r-1}(\mu_i)y_{si}, & \end{aligned}$$

and in fact this result is just (8.10) for C 'transposed' about its secondary diagonal.

The result (8.9) was given by Thompson et al. (1968) in the following instructive form. Let $\nu_1 \geq \nu_2 \geq \ldots \geq \nu_{k-1}$ be the totality of eigenvalues of $C_{0,r-1}$ and $C_{r,k}$, then from the Cauchy inequalities

$$\mu_1 \geq \nu_1 \geq \mu_2 \geq \ldots \geq \nu_{k-1} \geq \mu_k \qquad (8.11)$$ eq:8.11

and (8.9) becomes

$$y_{ri}^2 = \left\{\frac{\mu_i - \nu_1}{\mu_i - \mu_1}\right\}\cdots\left\{\frac{\mu_i - \nu_{i-1}}{\mu_i - \mu_{i-1}}\right\}\left\{\frac{\mu_i - \nu_i}{\mu_i - \mu_{i+1}}\right\}\cdots\left\{\frac{\mu_i - \nu_{k-1}}{\mu_i - \mu_k}\right\} \qquad (8.12)$$ eq:8.12

where each of the factors in brackets lies between 0 and 1. These bounds on these factors turn out to be very important in part of the following analysis.

8.3 Some Properties of the Eigensystems of the $C_{0,j}$

sec:8.3

Let $C_j \equiv C_{0,j}$, $j = 1, 2, \ldots, k$, then some relations between the eigensystem of C_j and that of C_k, $k > j$, will be essential for future results. The following terminology will be used to distinguish the eigensystems for different values of $j = 1, 2, \ldots, k$

$$\left.\begin{array}{ll} C_j Y^{(j)} = Y^{(j)} \mathrm{diag}(\mu_t^{(j)}), & Y^{(j)} = (y_{st}^{(j)}) = (y_1^{(j)}, \ldots, y_j^{(j)}) \\ \left(Y^{(j)}\right)^T Y^{(j)} = I, & \mu_1^{(j)} > \mu_2^{(j)} > \ldots > \mu_j^{(j)}. \end{array}\right\} \qquad (8.13) \quad \boxed{\texttt{eq:8.13}}$$

Then for $j < k$

$$C_k \left[\begin{array}{c} y_r^{(j)} \\ \hline 0 \\ \vdots \\ 0 \end{array}\right] = \left[\begin{array}{cc|cc} C_j & & & \\ & & \delta_{j+1} & \\ \hline & \delta_{j+1} & & \\ & & & C_{j,k} \end{array}\right] \left[\begin{array}{c} y_r^{(j)} \\ \hline 0 \\ \vdots \\ 0 \end{array}\right] = \left[\begin{array}{c} \mu_r^{(j)} y_r^{(j)} \\ \hline \delta_{j+1} y_{jr}^{(j)} \\ 0 \\ . \\ 0 \end{array}\right] \qquad (8.14) \quad \boxed{\texttt{eq:8.14}}$$

and so from the usual theory (Wilkinson, 1965, p. 171)

$$\min_i |\mu_i^{(k)} - \mu_r^{(j)}| \le \delta_{j+1} |y_{jr}^{(j)}| = a_r, \text{ say,} \qquad (8.15) \quad \boxed{\texttt{eq:8.15}}$$

that is, for every $k \ge j$ there is an eigenvalue of C_k within a distance a_r from $\mu_r^{(j)}$, and it will be said that $\mu_r^{(j)}$ has converged to an accuracy a_r.

Now denoting $M \equiv (y_t^{(j)}, \ldots, y_{t+s}^{(j)})$ it follows from (8.14) that

$$C_k \begin{bmatrix} M \\ 0 \end{bmatrix} = \begin{bmatrix} M \\ 0 \end{bmatrix} \mathrm{diag}(\mu_t^{(j)}, \ldots, \mu_{t+s}^{(j)}) + \delta_{j+1} e_{j+1} e_j^T \begin{bmatrix} M \\ 0 \end{bmatrix}$$

so that from the generalisation of the Wielandt-Hoffman theorem (Wilkinson, 1970) there exist integers $1 \le i_0 < i_1 < \ldots < i_s \le k$, for $k > j$, such that

$$\sum_{r=0}^{s} (\mu_{i_r}^{(k)} - \mu_{t+r}^{(j)})^2 \le \delta_{j+1}^2 \sum_{r=0}^{s} (y_{j,t+r}^{(j)})^2. \qquad (8.16) \quad \boxed{\texttt{eq:8.16}}$$

Thus if a group of $s+1$ eigenvalues of C_j are well converged in the sense of small a_r in (8.15) then (8.16) indicates that $s+1$ eigenvalues have converged to a certain accuracy; that is there are $s+1$ different eigenvalues of C_k, $k \ge j$, close to these. This result is useful when these eigenvalues of C_j are close together, resulting in the intervals given by (8.15) overlapping.

An important relation between the eigenvector of C_j and C_k, $k > j$, can also be found, for later use, by multiplying (8.14) by $(y_i^{(k)})^T$, giving

$$(\mu_i^{(k)} - \mu_r^{(j)}) y_i^{(k)T} \begin{bmatrix} y_r^{(j)} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \delta_{j+1} y_{jr}^{(j)} y_{j+1,i}^{(k)}. \qquad (8.17)$$

eq:8.17

Now if $y_{jr}^{(j)} = 0$ then from (8.14) $\mu_r^{(j)}$ is an eigenvalue of C_j and C_{j+1}, which is impossible, therefore $y_{jr}^{(j)} \neq 0$. Next if $\mu_i^{(k)} = \mu_{r_0}^{(j)}$ for some r_0 then $y_{j+1,i}^{(k)} = 0$, necessarily, and so (8.17) will be zero for $r = 1, \ldots, j$, but the eigenvalues of C_j are distinct

$$\therefore \quad (y_r^{(j)T}, 0, \ldots, 0) y_i^{(k)} \begin{cases} = 0 & , \quad r \neq r_0, \\ = \pm \left[\sum_{t=1}^{j} (y_{ti}^{(k)})^2 \right]^{\frac{1}{2}} & , \quad r = r_0, \end{cases}$$

the second result being a consequence of the first.

Next assuming $\mu_i^{(k)} \neq \mu_r^{(j)}$, $r = 1, \ldots, j$ and $i = 1, \ldots, k$ in (8.17), two extra relations are

$$\left\| Y^{(k)T} \begin{bmatrix} y_r^{(j)} \\ 0 \end{bmatrix} \right\|^2 = 1 = (\delta_{j+1} y_{jr}^{(j)})^2 \sum_{i=1}^{k} \left[y_{j+1,i}^{(k)} / (\mu_i^{(k)} - \mu_r^{(j)}) \right]^2$$

and

$$\sum_{t=1}^{j} (y_{ti}^{(k)})^2 = (\delta_{j+1} y_{j+1,i}^{(k)})^2 \sum_{r=1}^{j} \left[y_{jr}^{(j)} / (\mu_i^{(k)} - \mu_r^{(j)}) \right]^2.$$

8.4 Loss of Orthogonality in A2

sec:8.4

The effect of rounding errors on orthogonality after j steps of Algorithm A2 can now be described in terms of the error expressions in Section 7 and $C_j \equiv C_{0,j}$ in (8.1). The loss of orthogonality can then be related to the eigenvectors of C_j and finally through equation (8.15) to the convergence of the eigenvalues.

First j steps of (7.11) may be described by

$$AV_j = V_jT_j + v_{j+1}e_j^T + G'_j$$

where from (7.7) G'_j has columns $\delta v_r - \delta A_r v_r$, so that multiplying on the right by $D_{0,j}^{-1}$ and defining

$$\left.\begin{aligned} & G_j \equiv (g_1, \dots, g_j) \equiv G'_j D_{0,j}^{-1}, \\ & W_j \equiv (w_1, \dots, w_j) \equiv V_j D_{0,j}^{-1} \\ \text{and} \quad & w_{j+1} \equiv v_{j+1}/(\delta_2\delta_3 \cdots \delta_{j+1}) \end{aligned}\right\} \qquad (8.18) \quad \boxed{\texttt{eq:8.18}}$$

gives

$$AW_j = W_jC_j + \delta_{j+1}w_{j+1}e_j^T + G_j. \qquad (8.19) \quad \boxed{\texttt{eq:8.19}}$$

For simplicity it will be assumed that $\|v_1\| = 1$, so from (7.27), (remembering the change in notation)

$$\left.\begin{aligned} & \delta_2\delta_3 \cdots \delta_r = \alpha^{n+(r-1)/2}\|v_r\| \\ \therefore \quad & \|w_1\| = 1, \quad \|w_r\| = \alpha^{n+(r-1)/2}, \quad r > 1, \end{aligned}\right\} \qquad (8.20) \quad \boxed{\texttt{eq:8.20}}$$

and from (7.7), (7.13), and (7.25)

$$\|g_r\| < 1.01(9 + m\beta)\epsilon\|A\|. \qquad (8.21) \quad \boxed{\texttt{eq:8.21}}$$

Now defining the strictly upper triangular matrix $U_j \equiv (0, W_1^T w_2, \dots, W_{j-1}^T w_j)$, (with obvious licence),

$$W_j^T W_j = U_j^T + D(\alpha^{2n+j-1}) + U_j \qquad (8.22) \quad \boxed{\texttt{eq:8.22}}$$

so that multiplying (8.19) on the left by W_j^T, and equating the right hand side with its own transpose

$$\begin{aligned} C_j(U_j^T + U_j) - (U_j^T + U_j)C_j = {} & \delta_{j+1}(W_j^T w_{j+1}e_j^T - e_j w_{j+1}^T W_j) \qquad (8.23) \quad \boxed{\texttt{eq:8.23}} \\ & + W_j^T G_j - G_j^T W_j + \mathrm{diag}(w_i^T w_i)C_j - C_j\mathrm{diag}(w_i^T w_i) \end{aligned}$$

where the diagonal on both sides is zero. Next note that

$$\text{diagonal of}(C_jU_j^T - U_j^TC_j) = \text{diag}(\delta_2 u_{12}, \delta_3 u_{23} - \delta_2 u_{12}, \ldots, -\delta_j u_{j-1,j}) \qquad (8.24) \quad \boxed{\texttt{eq:8.24}}$$

where $u_{ir} = w_i^T w_r$ so that

$$\delta_{i+1}u_{i,i+1} = v_i^T v_{i+1}/(\delta_2\delta_3\cdots\delta_i)^2 = \alpha^{2n+i-1}v_i^T v_{i+1}/v_i^T v_i$$

and therefore from (7.26) and (7.28)

$$|\delta_{i+1}u_{i,i+1}| < 2.2(n+4)i\epsilon\|A\|. \qquad (8.25) \quad \boxed{\texttt{eq:8.25}}$$

Thus it is possible to equate the upper triangular parts of the matrix equation (8.23) to give the extremely important result on the loss of orthogonality of the algorithm

$$C_jU_j - U_jC_j = \delta_{j+1}W_j^T w_{j+1}e_j^T - H_j \qquad (8.26) \quad \boxed{\texttt{eq:8.26}}$$

where H_j is upper triangular having elements h_{ir}

$$\left.\begin{aligned} h_{11} &= \delta_2 u_{12}, \quad h_{ii} = \delta_{i+1}u_{i,i+1} - \delta_i u_{i-1,i}, \quad i = 2, \ldots, j \\ h_{i-1,i} &= -w_{i-1}^T g_i + g_{i-1}^T w_i + \delta_i(w_i^T w_i - w_{i-1}^T w_{i-1}) \\ h_{r,i} &= -w_r^T g_i + g_r^T w_i, \quad r = 1, 2, \ldots, i-2. \end{aligned}\right\} \qquad (8.27) \quad \boxed{\texttt{eq:8.27}}$$

Now from (7.27) and (8.20)

$$\begin{aligned} w_i^T w_i - w_{i-1}^T w_{i-1} &= (v_i^T v_i - \delta_i^2 v_{i-1}^T v_{i-1})/(\delta_2\cdots\delta_i)^2 \\ &= w_i^T w_i(1 - \alpha^{2n+1}) = \alpha^{2n+i-1}(2n+1)\epsilon \end{aligned}$$

so with (7.24), (8.20), (8.21), and (8.25)

$$\left.\begin{aligned} |h_{ii}| &< 4.4(n+4)i\epsilon\|A\| \\ |h_{i-1,i}| &< 1.01(3.2n + 2m\beta + 20)\epsilon\|A\| \\ |h_{r,i}| &< 2.02(m\beta + 9)\epsilon\|A\|, \quad 1 \le i < r-1. \end{aligned}\right\} \qquad (8.28) \quad \boxed{\texttt{eq:8.28}}$$

Using these results it is possible to show that the Frobenius norm of H_j in (8.26) satisfies

$$\|H_j\|_F < \left[2.6(n+4)(j+1)^{3/2} + 1.5(m\beta+10)j\right] \epsilon\|A\|, \qquad (8.29) \quad \boxed{\texttt{eq:8.29}}$$

although more specific bounds will be needed later.

For the moment only the eigensystem of C_j will be considered, and the following simple notation will be used

$$\left.\begin{array}{l} C_jY_j = Y_j\text{diag}(\mu_i), \quad Y_j \equiv (y_1,\ldots,y_j), \quad Y_j^TY_j = I \\ \text{and} \quad Z_j \equiv (z_1,\ldots,z_j) \equiv W_jY_j, \end{array}\right\} \qquad (8.30) \quad \boxed{\texttt{eq:8.30}}$$

so that the z_i are the approximations to the eigenvectors of A after step j. Substituting in (8.26) gives

$$\text{diag}(\mu_i)Y_j^TU_jY_j - Y_j^TU_jY_j\text{diag}(\mu_i) = \delta_{j+1}Z_j^Tw_{j+1}e_j^TY_j - Y_j^TH_jY_j \qquad (8.31) \quad \boxed{\texttt{eq:8.31}}$$

and the elements $\epsilon_{ir}^{(j)} \equiv y_i^TH_jy_r$, or ϵ_{ir} here for simplicity, can be bounded 'a priori'.

A result of great significance for the understanding of the Lanczos process is obtained by equating the (i,i) elements of both sides of (8.31), giving,

$$z_i^Tw_{j+1} = \epsilon_{ii}/(\delta_{j+1}y_{ji}), \qquad (8.32) \quad \boxed{\texttt{eq:8.32}}$$

where y_{ji} is the last element of y_i and so cannot be zero. The significance of this result follows from (8.15), and it means in effect that an approximate eigenvector z_i of A at step j is largely orthogonal to w_{j+1} unless μ_i has converged to an accuracy approaching $|\epsilon_{ii}|$.

This result will be examined further later, but for the present considering (8.32) for $i = 1,\ldots,j$ gives

$$\delta_{j+1}W_j^Tw_{j+1} = Y_jb_j \qquad (8.33) \quad \boxed{\texttt{eq:8.33}}$$

where b_j is a vector with elements $b_{ij} \equiv \epsilon_{ii}/y_{ji}$. In particular

$$\begin{aligned}\delta_{j+1} w_j^T w_{j+1} &= \sum_{i=1}^{j} \epsilon_{ii} = \operatorname{trace}(Y_j^T H_j Y_j) \\ &= \sum_{i=1}^{j} h_{ii} = \delta_{j+1} u_{j,j+1} \qquad (8.34)\end{aligned}$$

eq:8.34

from (8.27), as expected, while

$$\delta_{j+1} w_{j-1}^T w_{j+1} = \sum_{i=1}^{j} \epsilon_{ii} y_{j-1,i}/y_{ji} = \left[\sum_{i=1}^{j} (\mu_i - \gamma_j)\epsilon_{ii}\right]/\delta_j \qquad (8.35)$$

eq:8.35

since from

$$(C_j - \mu_i I) y_j = 0, \quad \delta_j y_{j-1,j} + (\gamma_j - \mu_i) y_{ji} = 0.$$

If as well as this the fact that

$$\sum_{i=1}^{j} \mu_i \epsilon_{ii} = \operatorname{trace}\left[\operatorname{diag}(\mu_i) Y_j^T H_j Y_j\right] = \operatorname{trace}(C_j H_j) \qquad (8.36)$$

eq:8.36

was used, the result corresponding to (7.33) would appear. Although this indicates a possibly faster way of obtaining an old result, and suggests a re-organization of the whole presentation so that it is centred around (8.26) from near the start, the main reason for this repetition was to show how (8.26) gives the key to the error behaviour of the algorithm. This is because it expresses the loss of orthogonality not in terms of the individual elements but as a function of the resulting matrix C_j, and so of the eigensystem of C_j, and this eigensystem is the main interest in the algorithm. Note that expressions for $v_{i-2}^T v_{i+1}$, etc. could have been found by the approach used in Section 7 to express $v_{i-1}^T v_{i+1}$, but the results would have been cumbersome and almost useless, while (8.33), which has been derived very easily from (8.26), is remarkably simple and clearly brings out the important factors in the process. It is in fact this relation between the loss of orthogonality and the eigensystem of C_j that makes the process so accurate, roughly speaking orthogonality is not fully lost in certain directions until eigenvalues corresponding to eigenvectors in these directions have converged. The theory leading up to (8.32) has been carefully checked several times, but it would also be interesting to test this result computationally.

Many other results follow directly from those just found, for instance equating the (i, r) elements in (8.31) and using (8.32) gives

$$\begin{aligned}(\mu_i - \mu_r) y_i^T U_j y_r &= \delta_{j+1} z_i^T w_{j+1} y_{jr} - \epsilon_{ir} \\ &= \epsilon_{ii} y_{jr} / y_{ji} - \epsilon_{ir} \qquad (8.37)\end{aligned}$$

eq:8.37

and this can be combined with the equivalent (r, i) expression to give

$$(\mu_i - \mu_r) y_i^T (U_j^T + U_j) y_r = \epsilon_{ii} y_{jr} / y_{ji} - \epsilon_{rr} y_{ji} / y_{jr} + \epsilon_{ri} - \epsilon_{ir} \qquad (8.38)$$

eq:8.38

which could then give $z_i^T z_r$. Instead of this, recourse will be made to (8.19) with both sides multiplied by z_i^T and y_r on the left and right respectively to give

$$z_i^T A z_r = \mu_r z_i^T z_r + \delta_{j+1} y_{jr} z_i^T w_{j+1} + y_i^T W_j^T G_j y_r \qquad (8.39)$$

eq:8.39

and since this must equal $z_r^T A z_i$, if $i \neq r$,

$$z_i^T z_r = \left[\delta_{j+1} (y_{jr} z_i - y_{ji} z_r)^T w_{j+1} + f_{ir} \right] / (\mu_i - \mu_r) \qquad (8.40)$$

eq:8.40

where here $f_{ir} \equiv f_{ir}^{(j)} \equiv y_i^T (W_j^T G_j - G_j^T W_j) y_r$. The result (8.32) can now be substituted to give

$$z_i^T z_r = (\epsilon_{ii} y_{jr} / y_{ji} - \epsilon_{rr} y_{ji} / y_{jr} + f_{ir}) / (\mu_i - \mu_r). \qquad (8.41)$$

eq:8.41

Of these two expressions for orthogonality of the approximate eigenvectors z_i of A, (8.40) is important when μ_i and μ_r have converged in the sense of small $\delta_{j+1} y_{ji}$ and $\delta_{j+1} y_{jr}$, while (8.41) is more important before convergence.

Taking $r = i$ in (8.39) gives with (8.32)

$$z_i^T A z_i = \mu_i z_i^T z_i + y_i^T (H_j + W_j^T G_j) y_i \qquad (8.42)$$

eq:8.42

so that μ_i is a very good approximation to the Rayleigh quotient if $\|z_i\|$ is not small.

Another approach of possible interest is to compare the eigensystem of C_j with the eigensystem of $W_j^T A W_j y = \mu W_j^T W_j y$ by using (8.19) and (8.33) to give

$$W_j^T A W_j y_i = \mu_i W_j^T W_j y_i + y_{ji} Y_j b_j + W_j^T G_j y_i, \tag{8.43} \qquad \boxed{\texttt{eq:8.43}}$$

this would then relate the convergence back to the convergence of the error-free process given in Section 4, but as work in this direction has so far yielded no significant results it will not be continued here.

Specific bounds on the basic error terms can now be given. If $y^T \equiv (\eta_1, \ldots, \eta_j)$ and $z^T \equiv (\zeta_1, \ldots, \zeta_j)$ are real vectors such that $y^T y = z^T z = 1$ then in (8.26)

$$\begin{aligned} y^T H_j z &= \sum_{r=1}^{j} \zeta_r \sum_{s=1}^{r} \eta_s h_{sr} \\ &= \sum_{r=1}^{j} \eta_r \zeta_r h_{rr} + \sum_{r=2}^{j} \eta_{r-1} \zeta_r h_{r-1,r} + \sum_{r=3}^{j} \zeta_r \sum_{s=1}^{r-2} \eta_s h_{sr}. \end{aligned}$$

By putting $a_s = 1$, $b_s = \eta_s$ in Hölder's inequality

$$\left(\sum |a_s b_s|\right)^2 \leq \sum |a_s|^2 \sum |b_s|^2,$$

it follows that

$$\sum_{s=1}^{r-2} |\eta_s| \leq (r-2)^{\frac{1}{2}} \quad \text{etc.},$$

so from (8.28)

$$\left.\begin{aligned} |y^T H_j z| &\leq \max_{1\leq r\leq j} |h_{rr}| + \max_{2\leq r\leq j} |h_{r-1,r}| + \sqrt{\frac{(j-1)(j-2)}{2}} \max_{\substack{3\leq r\leq j \\ 1\leq s\leq r-2}} |h_{sr}| \\ &< j(4.4n + 1.5m\beta + 32)\epsilon\|A\| \\ &= j\chi, \ \text{ say}, \end{aligned}\right\} \tag{8.44} \qquad \boxed{\texttt{eq:8.44}}$$

giving

$$|\epsilon_{ir}^{(j)}| < j\chi, \quad i, r = 1, \ldots, j. \tag{8.45} \qquad \boxed{\texttt{eq:8.45}}$$

Finally in (8.40) using (8.21)

$$|f_{ir}^{(j)}| < \left(\|z_i^{(j)}\| + \|z_r^{(j)}\|\right) 1.01 j^{\frac{1}{2}} (9 + m\beta)\epsilon\|A\|. \tag{8.46}$$

eq:8.46

Now that all the basic error expressions and bounds have been obtained it will be possible to examine the effectiveness of the algorithm.

8.5 The Concept of Convergence of the Algorithm A2

sec:8.5

In sub-section 8.3 several results were given on convergence of eigenvalues of the consecutive matrices C_j. Three main questions then arise

1) Need any of the a_r in (8.15) necessarily be small, i.e. is convergence assured?

2) If so, are the resulting eigenvalues good approximations to some eigenvalues of A?

3) What is the accuracy of the corresponding approximate eigenvectors $z_i = W_j y_i$?

First noting that if $C_j y_i = \mu_i y_i$, $y_i^T y_i = 1$, then (8.19) gives

$$A z_i = \mu_i z_i + \delta_{j+1} y_{ji} w_{j+1} + G_j y_i \tag{8.47}$$

eq:8.47

so that with (8.20) and (8.21) there exists an eigenvalue λ_s of A such that

$$\left.\begin{aligned} &|\mu_i - \lambda_s| < 1.01[\delta_{j+1}|y_{ji}| + g(j)]/\|z_i\| \\ \text{with}\quad &g(j) \equiv j^{\frac{1}{2}}(9 + m\beta)\epsilon\|A\|. \end{aligned}\right\} \tag{8.48}$$

eq:8.48

Thus if an eigenvalue μ_i of C_j has converged in the sense that $\delta_{j+1}|y_{ji}|$ is very small, then μ_i is also a good approximation to an eigenvalue λ_s of A, as long as $\|z_i\|$ is not small. In the same circumstances z_i is a good approximation to the corresponding eigenvector of A if λ_s is well separated from the other eigenvalues of A (Wilkinson, 1965, p. 173).

From (8.40) and (8.20)

$$\frac{|z_i^T z_r|}{\|z_i\|\|z_r\|} < 1.01 \left[\frac{\delta_{j+1}|y_{ji}| + g(j)}{\|z_i\|} + \frac{\delta_{j+1}|y_{jr}| + g(j)}{\|z_r\|} \right] / |\mu_i - \mu_r| \tag{8.49}$$

eq:8.49

which is just the sum of the two bounds (8.48) for μ_i and μ_r, divided by $|\mu_i - \mu_r|$; the orthogonality of any two approximate eigenvectors can then also be easily bounded if the three denominators on the right of (8.49) can be bounded below. Note that there is a lower bound on $\delta_{j+1}|y_{ji}|$ for a given error term ϵ_{ii}, for from (8.32)

$$\begin{aligned} \epsilon_{ii} &= \delta_{j+1} y_{ji} z_i^T w_{j+1} \\ \therefore \quad \delta_{j+1}|y_{ji}| &> |\epsilon_{ii}| / (\|z_i\|\|w_{j+1}\|). \end{aligned} \tag{8.50}$$

eq:8.50

Thus the smallest attainable value of $\delta_{j+1}|y_{ji}|$ cannot be smaller than $|\epsilon_{ii}|/(1.01\|z_i\|)$, and it can be seen by using the bounds (8.45) and (8.48) that an *a priori* bound on eigenvalue accuracy will almost certainly not be better than

$$j\chi/\|z_i\|^2 + g(j)/\|z_i\|, \tag{8.51}$$

eq:8.51

where it is hoped that $\|z_i\| \doteq 1$.

Unfortunately (8.48), (8.49), and (8.51) depend inversely on the size of z_i, and as this also comes into convergence proofs it will now be considered in detail.

8.6 Lower Bounds on $\|W_j y_i\|_2$

sec:8.6

Here an expression will be found for $z_i^T z_i$, where $z_i = W_j y_i$, this expression depending on the eigenvalues of C_j and the errors in the process. Unfortunately $\|z_i\| \doteq 1$ need not necessarily hold, as the following example illustrates. Let

$$C_2 Y = \begin{bmatrix} \gamma & \delta \\ \delta & \gamma \end{bmatrix} (1/\sqrt{2}) \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = (1/\sqrt{2}) \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \gamma + \delta & 0 \\ 0 & \gamma - \delta \end{bmatrix}$$

with $w_1^T w_1 = w_2^T w_2 = 1$, then from (8.27)

$$\delta \cdot w_1^T w_2 = h_{11}$$

so

$$W_2^T W_2 = \begin{bmatrix} 1 & h_{11}/\delta \\ h_{11}/\delta & 1 \end{bmatrix}$$

giving

$$\begin{aligned} z_i^T z_i &= 1 + 2y_{1i}y_{2i}h_{11}/\delta, \quad z_1^T z_2 = 0, \\ &= \begin{cases} 1 + h_{11}/\delta & \text{for } i = 1 \\ 1 - h_{11}/\delta & \text{for } i = 2 \end{cases} \end{aligned}$$

where it is possible from (8.25) and (8.27) that $h_{11} \doteq n\epsilon\|A\|$. Thus for very small δ, corresponding to very close eigenvalues of C_2, it is possible to have $\|z_i\|$ significantly different from unity, and even zero.

The unfortunate fact about this situation is that, as will be shown later, it is possible to have several very close eigenvalues of C_j corresponding to only one, even well separated, eigenvalue of A. Luckily the above example also suggests how to deal with this difficulty, since here $z_1^T z_1 + z_2^T z_2 = 2$.

Suppose $C_j y_i = \mu_i y_i$, $y_i^T y_i = 1$, $z_i = W_j y_i$, $i = 1, \ldots, j$, then it will be shown that $\|z_i\| \doteq 1$ for any well separated eigenvalue μ_i of C_j. If however, a group of very close eigenvalues $\mu_t, \ldots, \mu_{t+s}$ are well separated from the rest, then it will be shown that

$$\sum_{i=t}^{t+s} z_i^T z_i \doteq s + 1.$$

As an initial step, note from (8.20) that

$$\begin{aligned} \sum_{i=1}^{j} z_i^T z_i &= \operatorname{trace}(Y_j^T W_j^T W_j Y_j) = \operatorname{trace}(W_j^T W_j) \\ &= \sum_{i=1}^{j} w_i^T w_i = \sum_{i=1}^{j} \alpha^{2n+i-1} = j + \frac{j(4n+j)\epsilon}{2}. \end{aligned} \tag{8.52}$$

$\boxed{\texttt{eq:8.52}}$

The following analysis involves eigensystems corresponding to different values of j above, and so the terminology used in (8.13) will be used here, together with $z_i^{(j)} \equiv W_j y_i^{(j)}$. Thus after the kth step, from (8.20) and (8.22)

$$\begin{aligned} {z_i^{(k)}}^T z_t^{(k)} &= {y_i^{(k)}}^T \{I + \mathrm{diag}[(2n+i-1)\epsilon] + U_k^T + U_k\} y_t^{(k)} \\ &= \begin{cases} 1 + (2n+k)\epsilon + 2{y_i^{(k)}}^T U_k y_i^{(k)}, & t = i, \\ (2n+k)\epsilon + {y_i^{(k)}}^T (U_k^T + U_k) y_t^{(k)}, & t \neq i. \end{cases} \end{aligned} \qquad (8.53) \quad \boxed{\texttt{eq:8.53}}$$

So now using (8.33) and (8.22)

$$\begin{aligned} {y_i^{(k)}}^T U_k y_t^{(k)} &= \sum_{j=1}^{k-1} y_{j+1,t}^{(k)} {y_i^{(k)}}^T \begin{bmatrix} Y^{(j)} \\ 0 \\ \cdot \\ 0 \end{bmatrix} b_j/\delta_{j+1} \\ &= \sum_{j=1}^{k-1} y_{j+1,t}^{(k)} \sum_{r=1}^{j} \frac{\epsilon_{rr}^{(j)}}{\delta_{j+1} y_{jr}^{(j)}} {y_i^{(k)}}^T \begin{bmatrix} y_r^{(j)} \\ 0 \\ \cdot \\ 0 \end{bmatrix} \end{aligned} \qquad (8.54) \quad \boxed{\texttt{eq:8.54}}$$

which, with (8.17), leads to

$${y_i^{(k)}}^T U_k y_t^{(k)} = \sum_{j=1}^{k-1} y_{j+1,t}^{(k)} \sum_{r=1}^{j} \frac{\epsilon_{rr}^{(j)} y_{j+1,i}^{(k)}}{\mu_i^{(k)} - \mu_r^{(j)}} \qquad (8.55) \quad \boxed{\texttt{eq:8.55}}$$

with the r-sum replaced by

$$\frac{\pm \epsilon_{r_0 r_0}^{(j)} \left[\sum_{s=1}^{j} (y_{si}^{(k)})^2\right]^{1/2}}{\delta_{j+1} y_{jr_0}^{(j)}}$$

if $\mu_{r_0}^{(j)} = \mu_i^{(k)}$ for some $1 \leq r_0 \leq j$. If however $t = i$ then such equality of eigenvalues implies that $y_{j+1,i}^{(k)} = 0$ and so this difficult term disappears from the sum giving

$${y_i^{(k)}}^T U_k y_i^{(k)} = \sum_{j=1}^{k-1} (y_{j+1,i}^{(k)})^2 \sum_{r=1}^{j} \frac{\epsilon_{rr}^{(j)}}{\mu_i^{(k)} - \mu_r^{(j)}} \qquad (8.56) \quad \boxed{\texttt{eq:8.56}}$$

where the r-sum is ignored if $y_{j+1,i}^{(k)} = 0$.

Use can now be made of the fascinating result of Thompson et al. (1968) given in (8.12), and for this the eigenvalues of C_j here can be thought of as just j of the eigenvalues of the $k-1$ by $k-1$ matrix obtained by omitting row and column $j+1$ in C_k. The totality of these eigenvalues can now be named

$$\nu_1^{(j)} \geq \nu_2^{(j)} \geq \ldots \geq \nu_{k-1}^{(j)} \qquad (8.57) \quad \boxed{\texttt{eq:8.57}}$$

with $\mu_r^{(j)} = \nu_{m_r}^{(j)}$, $r = 1, \ldots, j$, defining a strictly increasing set of integers $m_1, \ldots, m_j$. As a result of using (8.12), (8.56) becomes

$$y_i^{(k)^T} U_k y_i^{(k)} = \left\{ \sum_{j=1}^{k-1} \sum_{r=1}^{j} \epsilon_{rr}^{(j)} \prod_{\substack{m=1 \\ m \neq m_r}}^{k-1} (\mu_i^{(k)} - \nu_m^{(j)}) \right\} / \prod_{\substack{m=1 \\ m \neq i}}^{k} (\mu_i^{(k)} - \mu_m^{(k)}) \qquad (8.58) \quad \boxed{\texttt{eq:8.58}}$$

where here there is no proviso at all.

It will now be easy to bound this for a well separated eigenvalue $\mu_i^{(k)}$. Suppose

$$b \equiv \min_{s \neq i} |\mu_i^{(k)} - \mu_s^{(k)}| \qquad (8.59) \quad \boxed{\texttt{eq:8.59}}$$

then using the inequalities mentioned for the factors in (8.12)

$$|y_i^{(k)^T} U_k y_i^{(k)}| \leq \sum_{j=1}^{k-1} \sum_{r=1}^{j} |\epsilon_{rr}^{(j)}| / |\mu_i^{(k)} - \mu_{\overline{m}_r}^{(k)}| \qquad (8.60) \quad \boxed{\texttt{eq:8.60}}$$

where

$$\overline{m}_r \equiv \begin{cases} m_r & \text{if } m_r < i, \\ m_r + 1 & \text{if } m_r \geq i, \end{cases}$$

thus $1 \leq \overline{m}_1 < \overline{m}_2 < \ldots < \overline{m}_j \leq k$, and $\overline{m}_r \neq i$.

Now from (8.45) and (8.59) this becomes

$$|y_i^{(k)^T} U_k y_i^{(k)}| < (\chi/b) \sum_{j=1}^{k-1} j^2 < k^3 \chi/(3b) \qquad (8.61) \quad \boxed{\texttt{eq:8.61}}$$

so that if

$$b > 2k^3\chi = k^3(8.8n + 3m\beta + 64)\epsilon\|A\| \tag{8.62}$$

eq:8.62

then from (8.53) with $|(2n+k)\epsilon| < 0.01$

$$1.2 > \|z_i^{(k)}\| > 0.8$$

and (8.48) becomes a useful bound. If b is very small compared with $\|A\|$ then it is most unlikely that a large proportion of the eigenvalues of C_k will be very close to $\mu_i^{(k)}$, and (8.61) will usually be a large over-bound, a factor k being more reasonable than the k^3 that appears. Thus in general much less separation than b in (8.62) will still usually ensure a reasonable value of $\|z_i\|$.

For a group of very close eigenvalues the analysis is far more complicated and deserves a sub-section of its own.

8.6.1 The Effect of Close Eigenvalues of C_k

subsec:8.6.1

Suppose that $\mu_t^{(k)}, \ldots, \mu_{t+s}^{(k)}$ are separated from the remaining eigenvalues by b, that is, with the usual ordering (8.13)

$$\mu_{t-1}^{(k)} - \mu_t^{(k)} > b, \quad \mu_{t+s}^{(k)} - \mu_{t+s+1}^{(k)} \geq b \tag{8.63}$$

eq:8.63

then this group of $s+1$ eigenvalues can be considered together. If $t = 1$ then the first inequality is meaningless while if $t + s = k$ the second is, but since the following results will be seen to hold for these two cases by a simple restriction of the argument for the case $1 < t < t + s < k$, only this last case need be considered.

Taking equation (8.58) and summing gives

$$\sum_{i=t}^{t+s} y_i^T U_k y_i = \sum_{j=1}^{k-1} \sum_{r=1}^{j} \epsilon_{rj} S_{rj}(t, t+s) \tag{8.64}$$

eq:8.64

where

$$S_{rj}(t,t+s) = \sum_{i=t}^{t+s} \left\{ \left[\prod_{\substack{m=1\\m\neq l}}^{k-1} (\mu_i - \nu_m) \right] \Big/ \left[\prod_{\substack{m=1\\m\neq i}}^{k} (\mu_i - \mu_m) \right] \right\}$$

and $l \equiv m_r$ is dependent only on r, j, and k, and the cumbersome superscripts have been dropped as S_{rj} need only be considered for fixed r, j, and k. The aim is to bound S_{rj} in terms of b, so defining $p(\mu)$ to be the monic polynomial of degree $k-s-1$ with roots

$$\nu_1, \ldots, \nu_{t-1}, \nu_{t+s}, \ldots, \nu_{k-1}$$

and defining $q(\mu)$ to be the monic polynomial of degree $k-s-1$ with roots

$$\mu_1, \ldots, \mu_{t-1}, \mu_{t+s+1}, \ldots, \mu_k$$

so that $q(\mu)$ has no zeros in $\mu_{t+s+1} < \mu < \mu_{t-1}$, and defining

$$r(\mu) \equiv p(\mu)/q(\mu), \quad r_i \equiv r(\mu_i), \quad i = t, \ldots, t+s,$$

the expression for S_{rj} may be re-written

$$S_{rj}(t,t+s) = \sum_{i=t}^{t+s} \left\{ r_i \left[\prod_{\substack{m=t\\m\neq l}}^{t+s-1} (\mu_i - \nu_m) \right] \Big/ \left[\prod_{\substack{m=t\\m\neq i}}^{t+s} (\mu_i - \mu_m) \right] \right\}.$$

Then if $t \leq l \leq t+s$ the coefficient of r_l in this sum is

$$\left[\prod_{\substack{m=t\\m\neq l}}^{t+s-1} (\mu_l - \nu_m) \right] \Big/ \left[\prod_{\substack{m=t\\m\neq l}}^{t+s} (\mu_l - \mu_m) \right]$$

and for the moment considering μ_l as a variable this coefficient can be decomposed into partial fractions to give

$$\sum_{\substack{i=t\\i\neq l}}^{t+s} \left\{ \frac{1}{\mu_l - \mu_i} \left[\prod_{\substack{m=t\\m\neq l}}^{t+s-1} (\mu_i - \nu_m) \right] \Big/ \left[\prod_{\substack{m=t\\m\neq i,l}}^{t+s} (\mu_i - \mu_m) \right] \right\}.$$

Now since the sum of the terms other than that involving r_l in S_{rj} may be written

$$\sum_{\substack{i=t\\i\neq l}}^{t+s}\left\{\frac{r_i}{\mu_i-\mu_l}\left[\prod_{\substack{m=t\\m\neq l}}^{t+s-1}(\mu_i-\nu_m)\right]\Bigg/\left[\prod_{\substack{m=t\\m\neq i,l}}^{t+s}(\mu_i-\mu_m)\right]\right\}$$

the total sum $S_{rj}(t,t+s)$ becomes

$$\sum_{\substack{i=t\\i\neq l}}^{t+s}\left\{\frac{r_l-r_i}{\mu_l-\mu_i}\left[\prod_{\substack{m=t\\m\neq l}}^{t+s-1}(\mu_i-\nu_m)\right]\Bigg/\left[\prod_{\substack{m=t\\m\neq i,l}}^{t+s}(\mu_i-\mu_m)\right]\right\}$$

$$=\sum_{\substack{i=t\\i\neq l}}^{t+s}\left\{\frac{r_l-r_i}{\mu_l-\mu_i}\left[\prod_{\substack{m=t\\m\neq l}}^{i-1}\frac{\mu_i-\nu_m}{\mu_i-\mu_m}\right]\left[\prod_{\substack{m=i\\m\neq l}}^{t+s-1}\frac{\mu_i-\nu_m}{\mu_i-\mu_{m+1}}\right]\right\}$$

but

$$\mu_1\geq\nu_1\geq\ldots\geq\mu_{i-1}\geq\nu_{i-1}\geq\mu_i\geq\nu_i\geq\ldots\geq\nu_{k-1}\geq\mu_k$$

so

$$0\leq(\mu_i-\nu_m)/(\mu_i-\mu_m)\leq 1\quad\text{for } m<i$$

and

$$0\leq(\mu_i-\nu_m)/(\mu_i-\mu_{m+1})\leq 1\quad\text{for } m\geq i$$

$$\therefore\quad |S_{rj}(t,t+s)|\leq\sum_{\substack{i=t\\i\neq l}}^{t+s}\left|\frac{r_l-r_i}{\mu_l-\mu_i}\right|.$$

However from the definition of $r(\mu)$, $r'(\mu)$ exists for $\mu_{t-1}>\mu>\mu_{t+s+1}$ so by the Mean Value Theorem

$$(r_l-r_i)/(\mu_l-\mu_i)=r'(\xi)$$

for some value of ξ lying between μ_l and μ_i, and the above sum can be bounded if a bound can be found on $|r'(\mu)|$ for $\mu_t>\mu>\mu_{t+s}$. Now

$$r(\mu)=\left\{\frac{\mu-\nu_1}{\mu-\mu_1}\right\}\cdots\left\{\frac{\mu-\nu_{t-1}}{\mu-\mu_{t-1}}\right\}\left\{\frac{\mu-\nu_{t+s}}{\mu-\mu_{t+s+1}}\right\}\cdots\left\{\frac{\mu-\nu_{k-1}}{\mu-\mu_k}\right\}$$

so that if $\nu_{t-1} \geq \mu \geq \nu_{t+s}$ then each factor in brackets lies between 0 and 1, and noting that

$$r'(\mu) = r(\mu)(d/d\mu) \ln r(\mu)$$

gives

$$r'(\mu) = r(\mu) \left[\sum_{\substack{m=1 \\ m \neq t,\dots,t+s-1}}^{k-1} (\mu - \nu_m)^{-1} - \sum_{\substack{m=1 \\ m \neq t,\dots,t+s}}^{k} (\mu - \mu_m)^{-1} \right].$$

Next examine the sum

$$\begin{aligned} S_1 &\equiv \sum_{m=1}^{t-1} [(\nu_m - \mu)^{-1} - (\mu_m - \mu)^{-1}] \\ &= (\nu_{t-1} - \mu)^{-1} - \left\{ (\mu_1 - \mu)^{-1} + \sum_{m=1}^{t-2} [(\mu_{m+1} - \mu)^{-1} - (\nu_m - \mu)^{-1}] \right\} \end{aligned}$$

for values of μ satisfying $\nu_{t-1} > \mu > \nu_{t+s}$. It can be seen that every term in the sum on the first line is non-negative, while the term in curly brackets on the second line is positive, so that

$$0 \leq S_1 < (\nu_{t-1} - \mu)^{-1}.$$

Similarly if

$$S_2 \equiv \sum_{m=t+s}^{k-1} [(\mu - \nu_m)^{-1} - (\mu - \mu_{m+1})^{-1}]$$

then

$$0 \leq S_2 < (\mu - \nu_{t+s})^{-1}.$$

But

$$r'(\mu) = r(\mu)[S_2 - S_1]$$

so for $\mu_t \geq \mu \geq \mu_{t+s}$ since $r(\mu) \geq 0$

$$|r'(\mu)| < r(\mu) \max_{\mu_t \geq \mu \geq \mu_{t+s}} [(\nu_{t-1} - \mu)^{-1}, (\mu - \nu_{t+s})^{-1}],$$

thus taking the product of $r(\mu)$ with each term inside the square brackets and using the boundedness of the factors of $r(\mu)$

$$\begin{aligned} |r'(\mu)| &< \max[(\mu_{t-1} - \mu_t)^{-1}, (\mu_{t+s} - \mu_{t+s+1})^{-1}] \\ &\leq 1/b, \quad \text{for } \mu_t \geq \mu \geq \mu_{t+s}. \end{aligned}$$

As a result, if $t \leq l \leq t+s$ then

$$|S_{rj}(t, t+s)| < s/b.$$

On the other hand if l does not lie within this region then from (8.63), (8.64), and the bounds on the factors in (8.12)

$$|S_{rj}(t, t+s)| \leq \sum_{i=t}^{t+s} (1/b) = (s+1)/b$$

giving from (8.45), no matter where l lies,

$$\begin{aligned} \left| \sum_{i=t}^{t+s} y_i^T U_k y_i \right| &< \frac{(s+1)\chi}{b} \sum_{j=1}^{k-1} j^2 \\ &< (s+1)k^3\chi/(3b) \qquad (8.65) \end{aligned}$$

eq:8.65

so that from (8.53)

$$\sum_{i=t}^{t+s} z_i^T z_i > (s+1)[0.99 - 2k^3\chi/(3b)]. \qquad (8.66)$$

eq:8.66

As a result it does not matter how close $\mu_t, \ldots, \mu_{t+s}$ are to each other, for as long as their separation b from the rest satisfies

$$b > 2k^3\chi = k^3(8.8n + 3m\beta + 64)\epsilon\|A\| \qquad (8.67)$$

eq:8.67

then

$$\left. \begin{array}{r} 1.35(s+1) > \sum_{i=t}^{t+s} z_i^T z_i > 0.65(s+1) \\ \text{so for at least one } i, \quad \|z_i\| > 0.8, \quad t \leq i \leq t+s. \end{array} \right\} \qquad (8.68)$$

eq:8.68

From (8.62) it can be seen that this is also true for $s = 0$, and the same sort of remarks that followed (8.62) apply here too.

8.7 A Proof of Convergence of the Algorithm

sec:8.7

First it will be shown that at least one of the eigenvalues of the consecutive C_j must converge in a manner to be indicated. Secondly when an eigenvalue does converge in this sense it will be shown that it must be a good approximation to an eigenvalue of A. The theory can easily be extended to several eigenvalues but the result obtained does not nearly indicate the rate of convergence found in practice; for this see Section 9.

Now from (8.20) $w_i^T w_i = \alpha^{2n+i-1}$, so defining $D_k \equiv \operatorname{diag}(\|w_i\|^{-1})$, the matrix $D_k W_k^T W_k D_k$ is non-negative definite with eigenvalues π_i such that

$$0 \le \pi_1 \le \pi_2 \le \ldots \le \pi_k$$

with

$$\|W_k D_k\|_F^2 = \operatorname{trace}(D_k W_k^T W_k D_k) = \sum_{i=1}^{k} \pi_i = k. \qquad (8.69) \quad \boxed{\texttt{eq:8.69}}$$

Next

$$\begin{aligned} D_k W_k^T W_k D_k &= I + D_k(U_k^T + U_k)D_k \\ &= Q_k^T \operatorname{diag}(\pi_i) Q_k, \text{ say, with } Q_k^T Q_k = I \end{aligned}$$

so that if $Q_k = (q_{ij})$ then

$$\begin{aligned} \|\operatorname{diag}(\pi_i)Q_k - Q_k\|_F^2 = \sum_{i=1}^{k}\sum_{j=1}^{k}(\pi_i - 1)^2 q_{ij}^2 = \sum_{i=1}^{k}(\pi_i - 1)^2 \\ = 2\|D_k U_k D_k\|_F^2 \le 2\alpha^{4n+2k}\|U_k\|_F^2 \le 2.041\|U_k\|_F^2 \end{aligned} \qquad (8.70) \quad \boxed{\texttt{eq:8.70}}$$

assuming as usual that $(2n+k)\epsilon < 0.01$.

The convergence proof depends on the fact that $D_k W_k^T W_k D_k$ must have at least $k - n$ zero roots for $k > n$, so suppose that $\pi_1 = \ldots = \pi_r = 0$, then

$$\sum_{i=1}^{k}(\pi_i - 1)^2 = r + \sum_{i=r+1}^{k}(\pi_i - 1)^2. \qquad (8.71) \quad \boxed{\texttt{eq:8.71}}$$

But by Hölder's inequality

$$\left[\sum_{i=r+1}^{k}(\pi_i - 1)\right]^2 \leq \sum_{i=r+1}^{k}(\pi_i - 1)^2 \sum_{i=r+1}^{k} 1^2 \qquad (8.72) \quad \boxed{\texttt{eq:8.72}}$$

with equality only if $\pi_i =$ constant, $i = r+1, \ldots, k$, i.e. if $\pi_i = k/(k-r)$ from (8.69). Thus combining (8.69) to (8.72)

$$2.041\|U_k\|_F^2 \geq r + (k-k+r)^2/(k-r) = kr/(k-r) \qquad (8.73) \quad \boxed{\texttt{eq:8.73}}$$

so that if

$$a = \max_{i \leq j < k} |(z_i^{(j)})^T w_{j+1}| \qquad (8.74) \quad \boxed{\texttt{eq:8.74}}$$

then since $W_k^T W_k$ must be singular, and so $r = 1$, for some value of $k \leq n+1$,

$$\begin{aligned} 0.49 < \|U_k\|_F^2 &= \sum_{j=1}^{k-1} \|W_j^T w_{j+1}\|^2 = \sum_{j=1}^{k-1} \|Z^{(j)^T} w_{j+1}\|^2 \\ &= \sum_{j=1}^{k-1}\sum_{i=1}^{j} (z_i^{(j)^T} w_{j+1})^2 \leq a^2 k(k-1)/2 \end{aligned} \qquad (8.75) \quad \boxed{\texttt{eq:8.75}}$$

so in (8.74)

$$a > 0.98/k. \qquad (8.76) \quad \boxed{\texttt{eq:8.76}}$$

As a result of this and (8.32), for some value of $k \leq n+1$, and some $i \leq k$

$$|z_i^{(k)^T} w_{k+1}| = |\epsilon_{ii}^{(k)}/(\delta_{k+1} y_{ki}^{(k)})| > 0.98/k \qquad (8.77) \quad \boxed{\texttt{eq:8.77}}$$

which with (8.15) and (8.45) shows that there will always be an eigenvalue μ of C_m, $m \geq k$ such that

$$\begin{aligned} |\mu - \mu_i^{(k)}| \leq \delta_{k+1}|y_{ki}^{(k)}| &< 1.03k^2\chi \\ &= 1.03k^2(4.4n + 1.5m\beta + 32)\epsilon\|A\|. \end{aligned} \qquad (8.78) \quad \boxed{\texttt{eq:8.78}}$$

Unfortunately this has only proven the necessary convergence to a practical tolerance of one eigenvalue of C_k, and it has not yet been shown that it is close to an eigenvalue of A. However (8.48) and (8.62) show that if for this eigenvalue $\mu_i^{(k)}$

$$|\mu_i^{(k)} - \mu_j^{(k)}| > k^3(8.8n + 3m\beta + 64)\epsilon\|A\|, \quad j \neq i, \tag{8.79}$$ $\boxed{\texttt{eq:8.79}}$

then there exists an eigenvalue λ of A such that

$$|\lambda - \mu_i^{(k)}| < 1.3\left[k^2(4.4n + 1.5m\beta + 32) + k^{\frac{1}{2}}(9 + m\beta)\right]\epsilon\|A\|. \tag{8.80}$$ $\boxed{\texttt{eq:8.80}}$

In practice well separated eigenvalues of A (this includes multiple eigenvalues too) have been found to have an error proportional to k, and since if the maximum possible error is proportional to k^2 the expected error would be proportional to k for stochastic errors, the above bound is probably a very good one.

If $\mu_i^{(k)}$ is not well separated then from (8.20), (8.45) and (8.77)

$$\|z_i^{(k)}\| > 0.96/k,$$
$$\delta_{k+1}|y_{ki}^{(k)}| < k^2\chi/0.98,$$

thus using (8.48) and (8.78) there exists an eigenvalue λ of A such that

$$|\lambda - \mu_i^{(k)}| < 1.1\left[k^3(4.4n + 1.5m\beta + 32) + k^{3/2}(9 + m\beta)\right]\epsilon\|A\| \tag{8.81}$$ $\boxed{\texttt{eq:8.81}}$

this being a weaker bound than (8.80).

The above proof of convergence of at least one eigenvalue can be extended to prove that other eigenvalues of C_k must converge in the sense of (8.15) as k increases. This can be done by noting that the right hand side of (8.73) must increase more than linearly with k, and so since the elements of U_k are bounded above it is possible to show that more and more elements of U_k must be large, and this fact can then be used to prove convergence of more and more eigenvalues of C_k. This only ensures that about $n/2$ eigenvalues of C_k must have converged by $k = n^2$, and as this is such a poor result there is no point in including the proof.

It is not clear if this approach can be significantly improved or if a completely new approach is needed, but whatever the case a proof of the convergence of at least

r eigenvalues to within the accuracy given in (8.80) in $k = n + r$ steps would seem a reasonable one to hope for. What appears intuitively likely is that greater use should be made of the equation (8.32)

$$z_i^{(j)^T} w_{j+1} = \epsilon_{ii}^{(j)} / (\delta_{j+1} y_{ji}^{(j)}),$$

meaning that orthogonality can only be lost as a result of convergence. In particular since $W_j = Z^{(j)} Y^{(j)^T}$, it follows that

$$w_j = \sum_{i=1}^{j} y_{ji}^{(j)} z_i^{(j)} \qquad (8.82) \quad \boxed{\texttt{eq:8.82}}$$

so that in step $j + 1$

$$\delta_{j+2} w_{j+2} = A w_{j+1} - \gamma_{j+1} w_{j+1} - \sum_{i=1}^{j} \delta_{j+1} y_{ji}^{(j)} z_i^{(j)} - g_{j+1} \qquad (8.83) \quad \boxed{\texttt{eq:8.83}}$$

and if $\delta_{j+1} |y_{ji}^{(j)}|$ is very small, meaning $\mu_i^{(j)}$ has converged to this accuracy, then an equally small amount of $z_i^{(j)}$ is subtracted at this step. That is, once an eigenvalue has converged, its eigenvector is ignored in the orthogonalization process (8.83). This indicates why a given eigenvector, and so eigenvalue, can appear again and again.

Such insights as these have not yet led anywhere, and so will not be pursued here. However even a proof of convergence in n steps would not truly indicate the value of the process in practice, and perhaps it is best to say that convergence is remarkably swift in practice (see Section 4), to illustrate this with examples, and to give simple 'a posteriori' bounds, as these can certainly be derived from the previous work.

Section 9

Computational Use of the Algorithm A2

chp:9

Some methods will be given for obtaining useful 'a posteriori' bounds for those eigenvalues of A that appear in any practical use of the Lanczos process using algorithm A2 described in Section 7 and 8. As is well known the conditioning of the eigenvectors depends on the eigenvalue separations, and so 'a posteriori' eigenvector bounds are not so easily obtained; nevertheless some results will be given and some suggestions made.

In the second part of this section computational results will be presented showing the deficiencies of algorithm A1 and the excellent properties of algorithm A2.

Again the subscript 2 will be dropped for the 2-norm, and the only other norm used, the Frobenius norm, will be indicated by the subscript F.

9.1 A Posteriori Eigenvalue Bounds

sec:9.1

A computation using the A2 variant of the symmetric matrix Lanczos process will produce the equivalent of the symmetric tridiagonal matrix C_k and the matrix $W_k \equiv$

$(w_1, w_2, \ldots, w_k)$ in (8.19). In Section 8 the true eigensolution of C_k was

$$C_k Y_k = Y_k \mathrm{diag}(\mu_1, \ldots, \mu_k), \quad Y_k \equiv (y_1, \ldots, y_k), \quad Y_k^T Y_k = I \qquad (9.1) \quad \boxed{\texttt{eq:9.1}}$$

with the eigenvalue ordering $\mu_1 > \ldots > \mu_k$, and intervals containing eigenvalues of A could be found as in (8.48). Unfortunately μ_i and y_i will not be accurately known because of rounding errors in solving the eigenproblem for C_k, so let $\nu_1, \ldots, \nu_k$ and $u_1, \ldots, u_k$ be the corresponding eigenvalue and eigenvector approximations obtained from some reliable algorithm (e.g. Bowdler, Martin, Reinsch, Wilkinson, 1968) where the u_i have been normalized so that $u_i^T u_i = 1$. In practice not all the eigenvalues and eigenvectors need be computed. It now remains to be shown that reliable eigenvalue intervals can be found using these computed values. As in Section 8 the difficulties will occur when there are several close eigenvalues of C_k.

Suppose it is known that the eigenvalues of C_k can be found to within a possible error d

$$|\mu_i - \nu_i| \le d, \quad i = 1, \ldots, k \qquad (9.2) \quad \boxed{\texttt{eq:9.2}}$$

then in order to treat $\nu_t, \ldots, \nu_{t+s}$ as a separate group, if

$$\left.\begin{aligned} &b = \min(\nu_{t-1} - \nu_t, \nu_{t+s} - \nu_{t+s+1}) - 2d \\ \text{with} \quad &b > k^3(8.8n + 3m\beta + 64)\epsilon\|A\| \end{aligned}\right\} \qquad (9.3) \quad \boxed{\texttt{eq:9.3}}$$

it follows for $j = t, \ldots, t+s$ and $i = 1, \ldots, t-1, t+s+1, \ldots, k$, that

$$|\mu_i - \mu_j| \ge b, \quad |\mu_i - \nu_j| \ge b + d = c, \quad \text{say}, \qquad (9.4) \quad \boxed{\texttt{eq:9.4}}$$

so with $z_i \equiv W_k y_i$, from (8.68)

$$1.35(s+1) > \sum_{i=t}^{t+s} z_i^T z_i > 0.65(s+1). \qquad (9.5) \quad \boxed{\texttt{eq:9.5}}$$

Next it will be important to know if such an inequality holds for the corresponding vectors obtained from $u_t, \ldots, u_{t+s}$.

If the approximate eigenvectors of A

$$\overline{v}_i = fl(W_k u_i)$$

are computed then there is no problem, ν_i and $\overline{v}_i$ are the approximate eigenvalue-vector pair and bounds may easily be obtained (see, for example, Wilkinson, 1965, pp. 172–3). However on large problems the vectors $w_1, \ldots, w_{k-2}$ will usually not be kept and the $\overline{v}_i$ will not be readily available.

In order to examine this last possibility define

$$\left.\begin{array}{ll} Y \equiv (y_t, \ldots, y_{t+s}), & \overline{Y} \equiv (y_1, \ldots, y_{t-1}, y_{t+s+1}, \ldots, y_k), \\ Z \equiv W_k Y, & \overline{Z} \equiv W_k \overline{Y}, \\ U \equiv (u_t, \ldots, u_{t+s}), & V \equiv (v_t, \ldots, v_{t+s}) \equiv W_k U, \\ H \equiv Y^T U, & \overline{H} \equiv \overline{Y}^T U, \\ \multicolumn{2}{l}{F \equiv (f_t, \ldots, f_{t+s}) \equiv C_k U - U \mathrm{diag}(\nu_t, \ldots, \nu_{t+s}).} \end{array}\right\} \qquad (9.6) \quad \boxed{\texttt{eq:9.6}}$$

Note that F can easily be computed if needed. Now multiplying this last equation by $\overline{Y}^T$ gives

$$y_i^T u_j = y_i^T f_j / (\mu_i - \nu_j), \quad j = t, \ldots, t+s; \; i \neq t, \ldots, t+s, \qquad (9.7) \quad \boxed{\texttt{eq:9.7}}$$

so with (9.4)

$$\|\overline{H}\|_F^2 \leq \sum_{j=t}^{t+s} \|f_j\|^2 / c^2 = \|F\|_F^2 / c^2 \qquad (9.8) \quad \boxed{\texttt{eq:9.8}}$$

while

$$\|H\|_F^2 \leq \|Y_k^T U\|_F^2 = s + 1. \qquad (9.9) \quad \boxed{\texttt{eq:9.9}}$$

Note that if there were no errors in U then $H = I$ and $\overline{H} = 0$. Define the symmetric matrices

$$E \equiv H^T H - I, \quad E' \equiv U^T U - I \qquad (9.10) \quad \boxed{\texttt{eq:9.10}}$$

then since

$$U^T U = U^T Y_k Y_k^T U = H^T H + \overline{H}^T \overline{H}, \qquad (9.11) \quad \boxed{\texttt{eq:9.11}}$$

$$\sigma \equiv \|E\| = \|E' - \overline{H}^T\overline{H}\| \leq \|E'\| + \|F\|_F^2/c^2 \tag{9.12}$$ eq:9.12

and so σ can be bounded once $u_t, \ldots, u_{t+s}$ are known. σ will be small for accurately computed eigenvalues and vectors of C_k, so from now on it will be assumed that $\sigma < 1$.

Now in order to obtain useful computational bounds it will be necessary to bound trace(V^TV), v defined in (9.6). Before doing this two simple bounds will be derived for traces of matrix products. If P and L are any m by n matrices then with obvious notation

$$\begin{aligned} |\text{trace}(P^HL)| &\leq \sum |p_i^H l_i| \leq \sum \|p_i\| \cdot \|l_i\| \\ &\leq \left(\sum \|p_i\|^2 \sum \|l_i\|^2\right)^{1/2} = \|P\|_F \|L\|_F \end{aligned} \tag{9.13}$$ eq:9.13

with equality if $P = L$. Next if M and N are Hermitian, $Q^HMQ = D = \text{diag}(d_i)$, Q unitary, and N is non-negative definite, then

$$\begin{aligned} &\text{trace}(MN) = \text{trace}(Q^HMNQ) = \text{trace}(DQ^HNQ) = \sum d_i q_i^H N q_i \\ \therefore\quad &|\text{trace}(MN)| \leq \|M\| \sum q_i^H N q_i = \|M\|\text{trace}(N). \end{aligned} \tag{9.14}$$ eq:9.14

Since $\sigma < 1$ in (9.10) and (9.12) the $s+1$ by $s+1$ matrix H is nonsingular giving

$$H^T = (I+E)H^{-1} = H^{-1} + E(I+E)^{-1}H^T \tag{9.15}$$ eq:9.15

where $E(I+E)^{-1}$ is symmetric. Next from (9.6)

$$V = W_kU = W_kY_kY_k^TU = ZH + \overline{Z}\,\overline{H}$$

so that using (9.15)

$$V^TV = H^{-1}Z^TZH + E(I+E)^{-1}H^TZ^TZH + \overline{H}^T\overline{Z}^T\overline{Z}\,\overline{H} + H^TZ^T\overline{Z}\,\overline{H} + \overline{H}^T\overline{Z}^TZH$$

and from (9.13), (9.14), and (9.12)

$$\mathrm{trace}(V^TV) \geq \mathrm{trace}(Z^TZ) + \|\overline{Z}\ \overline{H}\|_F^2 - 2\|ZH\|_F\|\overline{Z}\ \overline{H}\|_F - \sigma\|ZH\|_F^2/(1-\sigma).$$

But from (8.52) and (9.5), trace$(\overline{Z}^T\overline{Z}) < k$, for values of k of interest, and if for the computed eigensystem of C_k

$$\|E'\| + \|F\|_F^2/c^2 < 1/(25k) \qquad (9.16) \quad \boxed{\texttt{eq:9.16}}$$

then from (9.8)

$$\|\overline{Z}\ \overline{H}\|_F < 0.2,$$

and

$$\sigma\|H\|_F^2/(1-\sigma) < (s+1)/(25k-1) \leq 1/24,$$

giving with (9.5)

$$\begin{aligned}\mathrm{trace}(V^TV) &> 23\mathrm{trace}(Z^TZ)/24 - 0.4(s+1)^{1/2}\|Z\|_F,\\ &> (0.65 \times 23/24 - 0.4 \times 0.8)(s+1) > 0.3(s+1).\end{aligned} \qquad (9.17) \quad \boxed{\texttt{eq:9.17}}$$

From this it follows for at least one value of i

$$\|v_i\| > 0.54, \quad t \leq i \leq t+s \qquad (9.18) \quad \boxed{\texttt{eq:9.18}}$$

but from (8.19) and (9.6) for this value of i

$$(A - \nu_i I)v_i = W_k f_i + \delta_{k+1}u_{ki}w_{k+1} + G_k u_i, \qquad (9.19) \quad \boxed{\texttt{eq:9.19}}$$

u_{ki} being the last element of u_i. Thus there is an eigenvalues λ of A such that

$$|\lambda - \nu_i| \leq 2\left[k^{1/2}(m\beta+9)\epsilon\|A\| + k^{1/2}\|f_i\| + \delta_{k+1}|u_{ki}|\right] \qquad (9.20) \quad \boxed{\texttt{eq:9.20}}$$

and the desired computable bound can be obtained by taking i in $t \leq i \leq t+s$, which gives the maximum right hand side in (9.20), to be denoted by $m(t, t+s)$.

So in practice if $\nu_t, \ldots, \nu_{t+s}$ and $u_t, \ldots, u_{t+s}$ are computed using some reliable algorithm then a knowledge of the error analysis of that algorithm can often be used to bound d in (9.2), $\|f_i\|$ in (9.6), and $\|E'\|$ in (9.10). As a result c can be found in (9.4) and a check can be made to ensure that (9.16) is satisfied, if so the computed u_{ki} can be used in (9.20). If (9.16) is not obeyed then this partial eigensystem of C_k can be refined or more eigenvalues can be included in the group under investigation if the separation of eigenvalues is the limiting factor.

With an algorithm such as that given by Bowdler et al. (1968) one QR reduction step consists of $k-1$ rotations in the planes i, $i+1$; $i = 1, \ldots, k-1$, and if s steps are needed altogether the error in any one root is bounded by a constant of order unity times $s \cdot \epsilon \cdot \max|\mu_i|$. The authors found that the average number of steps required per eigenvalue was about 1.6, with no eigenvalue requiring more than 6 steps. For such an algorithm $d \ll b$ in (9.2) and (9.3). What is more every vector u_i is an exact eigenvector of some matrix very close to C_k, so the f_i in (9.6) will be very small. Finally a characteristic of this particular algorithm is that the u_i are always very accurately orthogonal and so E' in (9.16) will be negligible. The error analysis for this QR algorithm is covered by the general analyses given by Wilkinson (1965, pp. 131-143), and these indicate that using standard floating point arithmetic the errors will always satisfy (9.16) for $\nu_t, \ldots, \nu_{t+s}$ satisfying (9.3).

On the other hand if an error analysis of the algorithm for finding the eigensystem of C_k is not available, then the $f_i = C_k u_i - \nu_i u_i$ in (9.6) may be computed, as may E' in (9.10). Then since an eigenvalue μ of C_k satisfies

$$|\mu - \nu_i| \leq \|f_i\|$$

b and c in (9.3) and (9.4) can be bounded and (9.16) checked. If this is satisfied the computed f_i may be used in (9.20). Note that computing the u_i by inverse iteration may not give satisfactorily small E' in (9.10) for a very close bunch of eigenvalues, and some orthogonalization technique may be necessary.

A more simple analysis would be possible for one well separated μ_t, but because of the simplicity of the condition (9.16) i.e.

$$c > 5k^{1/2}\|f_t\|$$

which for any reasonable algorithm will be obeyed if (9.3) is satisfied there is no need to do the analysis to get an easier condition.

Unfortunately (9.20) suggests the possibility of one or more large values of $\delta_{k+1}|u_{ki}|$ for the μ_i in the close bunch. In the few computations that have been performed so far this has not occurred. Nevertheless it would be satisfying to prove that this could not happen, or that if it did the converged roots in the group were excellent approximations to a root of A. It is possible using (8.32) and (8.37) to prove that if $z_r^T w_{k+1}$ is not small then $\delta_{k+1}|y_{ki}|$ is small for all i such that $\mu_i \doteq \mu_r$, but this is not quite the same thing.

9.2 A Posteriori Eigenvector Bounds

sec:9.2

Eigenvector bounds are not readily obtainable unless the separation of the eigenvalues of A is known, in which case (9.18) and (9.19) could be used with the usual analysis (Wilkinson, 1965, pp. 172 - 3). Note that if $\mu_t, \ldots, \mu_{t+s}$ are all approximations to one eigenvalue λ_j of A the analysis is still useful if this group is separated by b from the other eigenvalues of A, all that is necessary is to take the maximum as was done in (9.20). So if v_r gives the maximum value $\|v_r\|$ in (9.18) and

$$v_r = \sum_{i=1}^{n} \alpha_i x_i, \quad Ax_i = \lambda_i x_i, \quad x_i^T x_j = \delta_{ij},$$

then

$$\|v_r - \alpha_j x_j\|^2 = \sum_{i \neq j} \alpha_i^2$$

but

$$\begin{aligned}\|(A-\nu_r I)v_r\|^2 &= \sum_{i=1}^{n} \alpha_i^2(\lambda_i-\nu_r)^2 \\ &\geq b^2 \sum_{i\neq j} \alpha_i^2\end{aligned}$$

which certainly gives with (9.19) and (9.20)

$$\|v_r-\alpha_j x_j\|/\|v_r\| < m(t,t+s)/b, \tag{9.21}$$

eq:9.21

and here $\alpha_j x_j$ is the projection of v_r on x_j.

Since here $\nu_t,\ldots,\nu_{t+s}$ are all approximations to λ_j of A, $v_t,\ldots,v_{t+s}$ are all approximations to x_j of A, and so s of these v_i will be redundant. It would be wasteful to compute the $s+1$ vectors, especially if the $w_1,\ldots,w_{k-2}$ are not stored, but there is a small chance that some of the v_i will be very small, and as a result probably very poor approximations to x_j. This difficulty is less daunting when it is realized that the earlier w_i will tend to be orthogonal, so that the computed u_r among the $u_t,\ldots,u_{t+s}$ having most weight in its early elements will probably give a v_r of reasonable size. This is of course just a heuristic approach and no analysis has been done to verify it. Practical experience of deliberately computing many redundant vectors u_i using a QL algorithm has suggested that such a u_r can easily be chosen, but so far no v_i have been formed. Another approach would be to find the values k where only one μ_t was a good approximation to λ_j.

Usually the separation of the eigenvalues of A will not be known 'a priori' and so a possible difficulty is that a value μ_t may approximate the eigenvalue λ_j of A and yet no other eigenvalue of C_k may be close to a nearby eigenvalue λ_{j+1} of A and it will be hard to judge the separation of these eigenvalues from the Lanczos process. No work has been done to resolve this difficulty, though it might be hoped to show that since no such eigenvalue of C_k had appeared, the projection of x_{j+1} on W_k, and so on v_t, would be small, thus allowing a good eigenvector bound.

For close eigenvalues of A well separated from the rest, the subspace of the corresponding vectors v_i should be considered as an approximation to a subspace of vectors of A.

At this point an advantage can be seen in the strange property that A2 has of producing redundant eigenvalues and vectors. The error free Lanczos process will not pick up multiple eigenvalues of A, and has to be restarted with a new w_1 to do so, and if $\lambda_1 = \ldots = \lambda_s$ the process would have to be run at least s times to find all these. However when errors are present the process can just be continued on, supplying several eigenvectors corresponding to the extreme eigenvalues of A. If r orthogonal vectors of reasonable size can be found by orthogonalization, but not normalization, of the $v_1, \ldots, v_s$ all corresponding to a well separated eigenvalue λ_1, then it could be assumed that λ_1 is at least an r-fold eigenvalue of A. If w_1 is arbitrary and $r < s$ then it is likely that λ_1 is no more than r-fold. Of course here it is impossible to tell the difference between a repeated eigenvalue and several very close eigenvalues of A – all that should be said is that r eigenvalues of A lie in such an such an interval.

This approach would not be such a good one if $s \times n$ steps were needed above, but in practice for very large matrices the extreme eigenvalues tend to be repeated several times before the less extreme ones are even partially converged, and in much less than n steps. This can be seen in the computational examples.

Thus computable bounds for some of the eigenvalues of A are easily obtainable, although slightly better results may be hoped for with a fuller understanding of the algorithm. For eigenvectors the well known difficulty of close eigenvalues of A arises, nevertheless it is hoped that some insight has been given along with several suggestions for further research into the effects of close or repeated eigenvalues.

9.3 Results of some Computations with A1 and A2

sec:9.3

Initially two examples will be given showing how the algorithm A1 can break down in practice; these support the arguments presented in Section 7.3. Later examples will show how A2 is not affected by such difficulties and produces remarkably accurate eigenvalues. Although some of the computations gave the eigenvectors of C_k, in no case was a corresponding set of approximate eigenvectors of A computed, although such a computation and comparison with the true values would have been instructive. The algorithms used to compute the eigensystems of the resulting tri-diagonal matrices C_k were the tql1 and tql2 algorithms given by Bowdler et al. (1968), and standard floating point arithmetic was used throughout, as described in Section 2.

Only two classes of matrices were used in the computation, the first of these being a single 8 by 8 matrix will be called A, while the matrices of the second class, which depend on two positive integers m and n and have dimensions mn will be denoted by $A_{m,n}$. The first, the Rosser matrix (Westlake, 1968, p. 150) has the following useful form and properties:–

$$A = \begin{bmatrix} 611 & 196 & -192 & 407 & -8 & -52 & -49 & 29 \\ & 899 & 113 & -192 & -71 & -43 & -8 & -44 \\ & & 899 & 196 & 61 & 49 & 8 & 52 \\ & & & 611 & 8 & 44 & 59 & -23 \\ & & & & 411 & -599 & 208 & 208 \\ & & & & & 411 & 208 & 208 \\ \text{symmetric} & & & & & & 99 & -911 \\ & & & & & & & 99 \end{bmatrix} \tag{9.22}$$

eq:9.22

Eigenvalues (let $a = \sqrt{10405}$, $b = \sqrt{26}$)

$$\left.\begin{array}{rcll}
\lambda_1 & = & 1020.04901843 & = 10a \\
\lambda_2 & = & 1020.0000 & \\
\lambda_3 & = & 1019.90195136 & = 510 + 100b \\
\lambda_4 & = & 1000. & \\
\lambda_5 & = & 1000. & \\
\lambda_6 & = & \quad 0.09804864072 & = 510 - 100b \\
\lambda_7 & = & \quad 0.0 & \\
\lambda_8 & = & -1020.04901843 & = -10a
\end{array}\right\} \tag{9.23} \boxed{\texttt{eq:9.23}}$$

This is a very good test matrix with its three large close roots, its large negative well separated one, its two equal roots and its two small close roots.

The other matrices, the Laplace matrices, have the form

$$\underset{mn\times mn}{A_{m,n}} \equiv \begin{bmatrix} B & -I & & & \\ -I & B & -I & & \\ & \cdots & \cdots & \cdots & \\ & & -I & B & -I \\ & & & -I & B \end{bmatrix}, \quad \underset{n\times n}{B} \equiv \begin{bmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \cdots & \cdots & \cdots & \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{bmatrix} \tag{9.24} \boxed{\texttt{eq:9.24}}$$

where I is the identity matrix of order n. $A_{m,n}$ has eigenvalues

$$\lambda_{p,q} = 4 - 2\cos p\pi/(m+1) - 2\cos q\pi/(n+1) \tag{9.25} \boxed{\texttt{eq:9.25}}$$

$$p = 1, \ldots, m; \quad q = 1, \ldots, n,$$

with each corresponding eigenvector having

$$x(p,q)_{r,s} = \sin pr\pi/(m+1)\sin qs\pi/(n+1) \tag{9.26} \boxed{\texttt{eq:9.26}}$$

as its $(r-1)n + s$ element, $r = 1, \ldots, m$; $s = 1, \ldots, n$.

The initial vector is given by

$$v = \sum_{p=1}^{m} \sum_{q=1}^{n} \alpha_{p,q} x(p, q) \tag{9.27}$$

eq:9.27

which is then normalized.

These matrices were chosen as useful examples of large sparse matrices, being both flexible with known properties, and easy to handle. Clearly with these it takes negligible storage to give all the information needed to form $A_{m,n}v$, v a given mn vector.

Computations Using A1

(1) The algorithm A1 described in Section 7.1 was iterated for $k = 60$ steps with the Rosser matrix A and an initial vector having all elements equal. The smallest next to diagonal element of C_{60} was $\delta_{47} \doteq 0.012$. For the eigenvalue ordering in (9.23) the following numbers of converged roots of C_k were obtained.

λ_1	λ_2	λ_3	λ_4 and λ_5	λ_6	λ_7	λ_8
7	5	8	9	9	9	11

The well separated root and the repeated root were always accurate in the 4th decimal place and often in the 5th and 6th, e.g. -1020.04901896. The close roots however were sometimes only accurate in the 2nd decimal place, e.g. 1020.0527.

This example brought out the possible weakness of A1 for close eigenvalues, the results were supported by other initial vectors with this matrix. It is also possible for the algorithm to break down for well separated eigenvalues at some stage, as the following result indicates.

(2) $k = 30$ steps of A1 were carried out with $A_{4,5}$ and an initial vector made up of equal amounts of $x(p, q)$ in (9.26), with $(p, q) = (4, 5)$, $(3, 5)$, $(4, 4)$, $(3, 4)$, $(2, 5)$, none of the remaining eigenvectors contributing. The process should ideally curtail after the 5th step. In fact $\delta_6 \doteq 3.14 \times 10^{-5}$ and the eigenvalues at this stage, being well separated, were accurate to 9 decimal places (i.e. 10 figures). Other eigenvalues that appeared later were often only accurate in the 5th decimal place, the remaining figures wandering around somewhat haphazardly as k increased, e.g. one calculated value stayed constant for 6 steps at 7.350082036 (true value 7.350084796) before changing to 7.350082921.

Computations Using A2

(3) The same computation as in (1) above was carried out using Algorithm A2. The smallest next to diagonal element of C_{60} was $\delta_{20} \doteq 0.00025$. Only 38 of the eigenvalues of C_{60} were computed, but to compare with (1) there were 9 of λ_6, 8 of λ_7, and 13 of λ_8 that had clearly converged. As expected, the different rounding errors lead to different numbers of repeated eigenvalues. Of those eigenvalues that were computed and had converged all were accurate in the 5th decimal place and usually in the 6th too. This is in marked contrast to A1, here even the well separated roots being found more accurately.

All the eigenvalues and eigenvectors were computed for $k = 10, 20, 30, 45, 46$, and when eigenvalues of C_k had converged they always represented the eigenvalues of A accurately in at least the 5th decimal place. In Table 3 the computed eigenvalues ν_i and the rough bounds $\delta_{k+1}u_{ki}$ are given to 7 decimal places for $k = 20$, this is about the limit of accuracy for the eigenvalues of C_k using the QL algorithm.

It can be seen that the algorithm A2 is remarkably accurate in this example, the larger eigenvalues often being given accurately to 11 decimal figures whereas the machine precision $\epsilon \doteq 10^{-10.8}$. The rough bound $\delta_{k+1}|u_{ki}|$ is seen to be exceeded occasionally, but is usually a fair indicator of convergence. The correct bound (9.20) is never exceeded but is seen to be if anything slightly pessimistic. Other computations with this matrix supported these conclusions perfectly.

(4) In the equivalent of (2) above for A2, $\delta_6 \doteq 3.5 \times 10^{-7}$, and all the eigenvalues that had converged at $k = 30$ were accurate in the 8th decimal place and often in the 9th, thus avoiding the inaccuracy that A1 suffered. Other similar computations support this conclusion.

(5) $k = 60$ steps of algorithm A2 were carried out on the 182 by 182 matrix $A_{13,14}$ with an initial vector in (9.27) having $\alpha_{13,14} = 300$, $\alpha_{13,13} = \alpha_{12,14} = 200$, $\alpha_{12,13} = 60$, $\alpha_{13,12} = \alpha_{11,14} = 32$, and $\alpha_{p,q} = 1$, otherwise. The smallest next to diagonal

Eigenvalues of A	Computed eigenvalues of C_{20}	$\delta_{21}\lvert u_{21,i}\rvert$ $\begin{pmatrix}\text{Leading}\\ \text{figure}\end{pmatrix}, \begin{pmatrix}\text{Decimal}\\ \text{exponent}\end{pmatrix}$
1020.0490184	1020.0490184	3 , -8
	1020.0490184	1 , -7
1020.0000	1020.0000000	5 , -8
	1019.9999998	6 , -7
1019.9019514	1019.9019514	3 , -8
	1019.9019513	6 , -8
1000.	1000.0000000	3 , -9
	999.9999999	4 , -9
	999.9999998	3 , -9
0.0980486	0.0980488	6 , -5
	0.0980485	6 , -6
	0.0980484	7 , -7
0.0	0.0000001	6 , -6
	0.0000001	6 , -6
	0.0000002	9 , -6
-1020.0490184	-987.2137098	3 , 2
	-1020.0490184	1 , -6
	-1020.0490186	0 ,
	-1020.0490187	0 ,
	-1020.0490183	0 ,

tab:3

Table 3: Eigenvalues of A using Algorithm A2, case(3).

element was $\delta_2 \doteq 0.16$, even so after 60 steps 7 of the extreme eigenvalues were given accurately in the 8th decimal place, none of these eigenvalues being redundant. The most accurate eigenvalues are given in Table 4 to 11 figures, as any more would certainly be meaningless.

Note that only the extreme eigenvalues are given accurately at this stage, as might be expected from Section 4. The large eigenvalues are given slightly more accurately than the small ones, this is probably because of their larger weighting in the initial vector. An accuracy of 11 figures is again achieved on two occasions, and as 60 steps of this algorithm takes very little time on the computer (compared with say 60 steps of the equivalent Householder algorithm which would require considerably more store as well) this is certainly a remarkable algorithm. The Lanczos reduction took 1100 instruction interrupts on the I.C.T. Atlas, each interrupt being 2048 machine instructions. The eigenvalue routine tql1 took 680 interrupts to find the 60 roots.

(6) As a final test of A2 it was applied for $k = 600$ steps to the 1000 by 1000 sparse matrix $A_{50,20}$, with the initial vector containing equal components of all the eigenvectors. All the next to diagonal elements lay between 1.5 and 2.5, so no convergence criteria could possibly be based on these alone. The Lanczos reduction took about 54000 instruction interrupts. The total real time was then about 6 minutes, which was just the time needed to find all the eigenvalues of C_{600}.

The results are too numerous to present in a table and a verbal description will suffice. 98 of the lowest eigenvalues of C_{600} agreed with corresponding eigenvalues of $A_{50,20}$ in at least the 8th decimal place, these true eigenvalues lay between 0.0261316900 and 0.9176084016 and included 64 different eigenvalues. From the smallest upwards the numbers of each of these were

$$5, 5, 4, 3, 3, 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,$$

with the remaining 44 appearing once only. The majority of the eigenvalues in the middle range were only accurate in the 2nd or 3rd decimal place. The roots of

p, q	$\lambda_{p,q}$ of $A_{13,14}$	ν_i of C_{60}
13,14	7.9061510257	7.9061510257
13,13	7.7769467396	7.7769467396
12,14	7.7582329373	7.7582329372
12,13	7.6290286511	7.6290286506
13,12	7.5678898130	7.5678897541
11,14	7.5199581664	7.5199579231
3, 1	0.4800418336	0.4800605972
1, 3	0.4321101869	0.4321131322
2, 2	0.3709713489	0.3709713857
2, 1	0.2417670627	0.2417670587
1, 2	0.2230532604	0.2230532563
1, 1	0.0938489742	0.0938489702

tab:4

Table 4: Eigenvalues of $A_{13,14}$ using Algorithm A2, case(5).

$A_{50,20}$ from 7.04474544 to the largest one 7.97386831 were given accurately in the 7th decimal place by the roots of C_{600} except for the adjacent pair 7.21238221 and 7.21292580 which were only accurate in the 6th decimal place. None of this group of 109 eigenvalues of C_{600} was more accurate than this. From the largest down, the numbers of each eigenvalue of $A_{50,20}$ appearing were

$$5, 5, 4, 3, 3, 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,$$

just at for the lower group, but this time with 55 roots appearing once only.

Note that the extreme eigenvalues were given several times in less than 1000 steps, when there had been negligible cancellation in any given step, and where the majority of the middle range of eigenvalues had not converged significantly at all. This is a good indication of the effect of powering with very large matrices, the components of the rounding errors in the directions of the eigenvectors of the matrix corresponding to the extreme eigenvalues are easily magnified to become huge in comparison with the magnified components of the initial vector in the other directions. Of course here the Lanczos algorithm treats the matrix as if it were shifted so that zero is in the middle of the eigenvalue range (Lanczos, 1950), and so both groups of extreme eigenvalues are given equal predominance.

It is interesting to note that the smaller group of eigenvalues were given with a greater absolute accuracy than the larger group, so that here in fact the relative accuracies were fairly comparable: a strange result in view of the above-mentioned equivalent shift.

In this example the largest eigenvalues were computed for k = 100, 200, 300, 400, 500 as well as 600. It was found that an eigenvalue that had converged would still change linearly with k, sometimes at a rate of as much as 10^{-10} per step, but usually less. It is not clear whether this is caused by the Lanczos A2 algorithm or the QL eigenvalue algorithm, as it is within the error bound of the latter. However it is assumed to be caused by the former because of the remarkable linearity over the k taken.

Thus although only a limited number of computational experiments have been performed they can be seen to support the findings of the error analysis very well indeed, as well as this the rates of convergence in these cases are seen to be very satisfying, and altogether suggest that the A2 variant of the symmetric Lanczos algorithm is a remarkably swift, accurate, and practical algorithm. The times for carrying out the Lanczos algorithm with the very sparse Laplace matrices were examined, with the result that these were directly comparable with the times required to find all the eigenvalues of the resulting tri-diagonal matrices C_k using the very fast QL algorithm. Naturally only a few of the eigenvalues of C_k would be needed in practice, but the comparison serves to emphasize the speed of the Lanczos process.

Finally it can be seen that no stopping criterion is necessary with A2 other than if $\delta_{k+1} = 0$, or perhaps if the limit on k given in (7.38) is exceeded by too much. Nevertheless since from (8.51) it seems that no more than a certain accuracy can be relied on, the criterion

$$\text{stop if} \quad \delta_{k+1} < 5k(n + 0.3m\beta + 7)\epsilon\|A\|$$

would be a possible one for preventing unnecessary extra iterations.

Section 10

Summary, Suggestions and Conclusion

chp:10

In the introduction the various possible well known methods for computing eigensolutions of large sparse matrices were considered and it was decided that the Lanczos algorithms might be particularly applicable in this case, especially with symmetric matrices. Most of the thesis was then devoted to analyses of the symmetric Lanczos process, both with and without re-orthogonalization, and so a more appropriate title would be

"A Detailed Analysis of Lanczos' Method for Finding Eigensolutions of Large Symmetric Matrices".

The results given in Section 4 suggested the excellence of the symmetric Lanczos process as an iterative method, while Section 5 examined the different possible ways of obtaining bounds in the case where less than the full number of steps had been carried out. In this section Lehmann's work in its full generality was seen to be an excellent theoretical completion of earlier work by Rayleigh, Temple, Kato and others, and as such is important; however as a means of computing eigenvalue bounds for large matrices when using the Lanczos process, it is concluded that the improvements over the more simple bounds are not worth the extra labour involved unless some extra information is available, in which case the excellent t, τ intervals can be very useful. In a more general linear operator problem where the formation of Av is far more costly the other Lehmann intervals could also be useful, but because of its poor computational performance, the algorithm suggested by Lehmann should not be used in either the matrix or the more general case, instead a more accurate algorithm has been given here.

Section 6 presents an analysis of the symmetric Lanczos process with re-orthogonalization, and it is shown how the orthogonality of the vectors can actually break down if there is too great a cancellation in successive steps. However, if a stopping criterion depending on the amount of cancellation is used then excellent ‘a priori’ bounds are given for the eigenvalues obtained up to that point, as well as for the orthogonality of the vectors. From the theoretical equivalence of the symmetric Lanczos process and a slight variant of Householder’s method, together with the excellent convergence properties discussed in Section 4, it is concluded that, because of its earlier speed and much simpler use of store, the Lanczos algorithm has a significant advantage over the Householder algorithm for some fairly large (especially sparse) matrices where only a few extreme eigenvalues and their eigenvectors are required.

Section 7 proceeds to the most important contribution of this thesis, the analysis and assessment of the symmetric Lanczos process without re-orthogonalization. It indicates why the standard algorithm, denoted A1, can be inaccurate and then goes on to suggest why a second algorithm A2 will not suffer from this particular inaccuracy. The algorithm A2 is considered further in Section 8 and its remarkably good rounding error properties are clearly brought out together with some important results concerning the approximate eigenvectors. A proof of necessary convergence and accuracy of one eigenvalue of the algorithm is then given, but unfortunately the proof of necessary convergence of more than one eigenvalue is not satisfactory and is not included as it does not begin to demonstrate the true speed of convergence that is encountered in practice. Section 9 then derives useful 'a posteriori' bounds for the algorithm A2; these are sufficient for the eigenvalue problem, but some more work could be done on deriving eigenvector bounds for close eigenvalues of A. Finally at the end of Section 9 some computational results are given that clearly show the remarkable properties of this A2 variant of Lanczos' beautiful method.

Thus because of its extremely small storage requirements, its ease of implementation and relatively small amount of computation per step, its rapid convergence for extreme eigenvalues and its remarkable accuracy coupled with the availability of simple 'a posteriori' bounds, it is reasonable to hope that the variant of the Lanczos process for symmetric matrices described in Section 7 as A2 will be recognised as one of the most useful possible methods for computing some extreme eigenvalues and their eigenvectors for large sparse symmetric matrices.

At this point it is worth commenting that the error analysis in Sections 7, 8 and 9 was for standard floating point arithmetic. Now as the dimension n of the matrix only comes into the error analysis in the formation of vector inner-products, the use of double length accumulation in these would effectively mean replacing n by 1 in all the error bounds, and for very large problems this might well be worth-while if the extra time was not too great.

Finally a slight variant of A2 that should perhaps be examined depends on the fact that δ_j^2 (δ_j in Section 7) can be computed before the rest of step j. Thus, instead of (7.3) to (7.4), as Av_j is being formed do not store it but form and store

$$v_j' = Av_j - \delta_j^2 v_{j-1}$$

in the old v_{j-1} vector, then theoretically

$$\gamma_j = v_j^T v_j' / v_j^T v_j$$

so that finally

$$\beta_{j+1}v_{j+1} = v'_j - \gamma_j v_j$$

can overwrite v'_j.

This variant is suggested because of the difficulty of using a fast double length accumulation of inner products routine in A2 if storage space is only available for two n-vectors, as indicated after Table 2 in Section 7.1.

10.1 Other Computational Problems

sec:10.1

Only the computation of some extreme eigenvalues and their eigenvectors of large sparse symmetric matrices has been dealt with here. There remain the problems of finding eigenvalues near the middle of the range and dealing with unsymmetric matrices.

For symmetric matrices the successful variant A2 of the Lanczos process can be applied to some polynomial of A, say $p(A)$, where of course only $p(A)v$ is ever formed, not $p(A)$. This polynomial must be designed so that its eigenvalues $p(\lambda_i)$ corresponding to the λ_i of interest are extreme eigenvalues (see, for example, Stiefel, 1958). Rounding errors introduce extra complications here and a best strategy would have to be evolved. For example if an eigenvalue λ_i of A was wanted and $p(A) = (A - \lambda)^2$ then using the Lanczos algorithm might give

$$\nu_1 = (\lambda_i - \lambda)^2 + \epsilon\|A\|^2$$

so that

$$\lambda_i = \lambda + \sqrt{\nu_1 - \epsilon\|A\|^2}$$

and if ν_1 is very small there is a large uncertainty in λ_i. Note that this is similar to the difficulty encountered with Lehmann's algorithm for finding Δ in Section 5, the trouble is here the Rayleigh quotient is not so easily available for resolving this difficulty.

The behaviour of the Lanczos process for unsymmetric matrices (Lanczos, 1950) should perhaps be re-examined in the light of the new results given in this thesis, but a successful outcome seems at first glance unlikely (Wilkinson, 1965, pp. 388–394). Another possible approach to the large sparse unsymmetric matrix problem (Lanczos?, reference mislaid) is to consider the Hermitian matrices

$$G \equiv K^H K \quad \text{and} \quad H \equiv \overline{K} K^T$$

where $K \equiv A - \lambda I$ for an approximate value λ. Then $Gx = o$ if $Ax = \lambda x$ and $Hy = o$ if $Ay = \lambda y$, so that approximations x_1 and y_1 to the eigenvectors corresponding to the minimum eigenvalues of G and H could be computed by a few steps of the Lanczos process for Hermitian matrices. The new value of λ could then be taken as

$$y_1^T A x_1 / y_1^T x_1$$

(e.g. Wilkinson, 1965, p. 179), and an iterative procedure carried out. Unfortunately this appears to be a rather clumsy and slow process, and the methods mentioned in the introduction that iterate with several vectors and use a convergence accelerating technique are probably better for unsymmetric matrices.

10.2 Conclusion

sec:10.2

The analysis given in this thesis supports the conclusion suggested by the computational results in Section 9 that Lanczos' method for finding eigenvalues has been far too readily abandoned. In fact the initial error analysis suggested the importance of the less favoured algorithm A2 compared with A1, and the later analysis and computations indicated just how remarkably accurate A2 is likely to be. As a result this thesis has presented a very firm case for resurrecting Professor Lanczos' beautiful method of minimized iterations for large sparse symmetric matrices using the computational algorithm A2 described in Section 7.

It is clear that the full power of the method has not yet been fully explored, and the analysis given here can perhaps also be used to examine the Lanczos process for large sparse unsymmetric matrices as well.

References

Arn51 ARNOLDI, W. E. (1951) *The Principle of Minimized Iterations in the Solution of the Matrix Eigenvalue Problem,* Quart. Appl. Math. 9, 17–29.

BMRW68 BOWDLER, H., MARTIN, R. S., REINSCH, C., AND WILKINSON, J. H. (1968) *The QR and QL Algorithms for Symmetric Matrices,* Numer. Math. 11, 293–306. (Handbook Series Linear Algebra)

CJ70 CLINT, M. AND JENNINGS, A.(1970) *The evaluation of eigenvalues and eigenvectors of real symmetric matrices by simultaneous iteration,* Comput. J. 13, 76–80.

EGRS59 ENGELI, M., GINSBURG, T., RUTISHAUSER, H., AND STIEFEL, E. (1959) *Refined Iterative Methods for Computations of the Solution and the Eigenvalues of Self-Adjoint Boundary Value Problems.* Birkhäuser Verlag, Basel/Stuttgart, 107 pp.

FF63 FADDEEV, D. K., AND FADDEEVA, V. N. (1963) *Computational Methods of Linear Algebra,* Freeman & Co. San Francisco and London. Original Moscow (1960) 656 pp.

Fai65 FAIRBOURN, A. (1965) *Atlas Basic Instruction Statistics,* I.C.S. Study Note 237.

FM67 FORSYTHE, G. E., AND MOLER, C. B. (1967) *Computer Solution of Linear Algebraic Systems.* Prentice-Hall, Inc. Englewood Cliffs, N. J. , 148 pp.

Giv54 GIVENS, W. (1954) *Numerical computation of the characteristic values of a real symmetric matrix.* Oak Ridge National Laboratory, ORNL–1574.

Gou57 GOULD, S. H. (1957) *Variational Methods for Eigenvalue Problems.* University of Toronto Press.

Hou64 HOUSEHOLDER, A. S. (1964) *The Theory of Matrices in Numerical Analysis.* Blaisdell, New York, 257 pp.

HB59 HOUSEHOLDER, A. S. AND BAUER, F. L.(1959) *On certain methods for expanding the characteristic polynomial,* Numer. Math. 1, 29–37.

IBM68 I.B.M. (1968) *Symposium on Sparse Matrices and Their Applications,* Willoughby, R.A. (Ed), I.B.M. Watson Research Center, Sept. 9–10 Yorktown Heights, New York.

ICT65 I.C.T. (1965) *ABL Manual.*

IMA70 I.M.A. (1970) *Conference on Large Sparse Sets of Linear Equations,* Reid, J. K. (Ed), April 5–8, Oxford.

Jen67 JENNINGS, A. (1967) *A Direct Iteration Method for Obtaining the Latent Roots and Vectors of a Symmetric Matrix,* Proc. Camb. Phil. Soc., 63, 755–765.

Kan66 KANIEL, S. (1966) *Estimates for Some Computational Techniques in Linear Algebra,* Math. Comp. 20, 369–378.

Kat49 KATO, T. (1949) *On the Upper and Lower Bounds of Eigenvalues,* J. Phys. Soc. Japan 4, 334–339.

Laa59 LAASONEN, P. (1959) *A Ritz method for simultaneous determination of several eigenvalues of a big matrix,* Ann. Acad. Sci. Fenn. Ser. AI, 265–280.

Lan50 Lanczos, C. (1950) *An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators,* J. Res. Nat. Bur. Standards 45, 255–282.

Lan52 Lanczos, C. (1952) *Solutions of Systems of Linear Equations by Minimized Iterations,* J. Res. Nat. Bur. Standards 49, 33–53.

Leh63 Lehmann, N. J. (1963) *Optimale Eigenwerteinschließungen*, Numer. Math. 5, 246–272.

Leh66 Lehmann, N. J. (1966) *Zur Verwendung optimaler Eigenwerteingrenzungen bei der Lösung symmerischer Matrizenaufgaben*, Numer. Math. 8, 42–55.

MW67 Martin, R. S. and Wilkinson, J. H.(1967) *Solution of Symmetric and Unsymmetric Band Equations and the Calculation of Eigenvectors of Band Matrices,* (Handbook Series, Linear Algebra) Numer. Math. 9, 279–301.

Pai69a Paige, C. C. (1969a) *Error Analysis of the Generalized Hessenberg Processes for the Eigenproblem,* London Univ. Inst. of Computer Science, Tech. Note ICSI 179.

Pai69b Paige, C. C. (1969b) *Error Analysis of the Symmetric Lanczos Process for the Eigenproblem,* London Univ. Inst. of Computer Science, Tech. Note ICSI 209.

Pai70a Paige, C. C. (1970a) *Eigenvalues of Perturbed Hermitian Matrices,* London Univ. Inst. of Computer Science, Tech. Note ICSI 248.

Pai70b Paige, C. C. (1970b) *Practical Use of the Symmetric Lanczos Process with Reorthogonalization,* BIT 10, 183–195.

Ray77 Rayleigh, Lord (1877) *Theory of Sound,* Macmillan and Co. London.

Rei70 REID, J. K. (1970) *On the Method of Conjugate Gradients for the Solution of Large Sparse Systems of Linear Equations,* Proc. IMA Conf. on Large Sparse Sets of Linear Equations, Oxford, April 1970, Academic Press.

Rut63 RUTISHAUSER, H.(1963) *On Jacobi Rotation Patterns,* Proc. A.M.S. Symposium in Applied Mathematics. 15, 219–239.

Rut69 RUTISHAUSER, H.(1969) *Computational Aspects of F. L. Bauer's Simultaneous Iteration Method,* Numer. Math. 13, 4–13.

RS63 RUTISHAUSER, H. AND SCHWARZ, H. R. (1963) *The LR Transformation Method for Symmetric Matrices,* Numer. Math. 5, 273–289. (Handbook Series Linear Algebra)

Sch68 SCHWARZ, H. R. (1968) *Tridiagonalisation of a Symmetric Band Matrix,* Handbook Series Linear Algebra, Numer. Math. 12, 231–241.

SN69 SEBE, T. AND NACHAMKIN, J. (1969) *Variational Buildup of Nuclear Shell Model Bases,* Annals of Physics 51, 100–123.

ST68 SIMPSON, A. AND TABARROK, B. (1968) *On Kron's Eigenvalue Procedure and Related Methods of Frequency Analysis,* Quart. J. Mech. Appl. Math. 21, 1–41.

Ste69 STEWART, G. W. (1969) *Accelerating the Orthogonal Iteration for the Eigenvectors of a Hermitian Matrix,* Numer. Math. 13, 362–376.

Sti58 STIEFEL, E. L. (1958) *Kernel Polynomials in Linear Algebra and Their Numerical Applications,* Appl. Math. Ser. Nat. Bur. Stand. 49, 1–22.

Sto68 STONE, H. L. (1968) *Iterative Solution of Implicit Approximations of Multidimensional Partial Differential Equations,* SIAM J. Numer. Anal. 5, 530–558.

Sze39 SZEGÖ, G. (1939) *Orthogonal Polynomials,* A.M.S. Colloquium Publications 23, New York City.

Tem28 TEMPLE, G. (1928) *The Theory of Rayleigh's Principle as Applied to Continuous Systems,* Proc. Roy. Soc. A119, 276–291.

Tew70 TEWARSON, R. P. (1970) *On the Transformation of Symmetric Sparse Matrices to the Triple Diagonal Form,,* Internat. J. Comput. Math. 2, (to appear)

TME68 THOMPSON, R. C. AND MCENTEGGERT, P. (1968) *Principal Submatrices II: The Upper and Lower Quadratic Inequalities,* Linear Algebra and Its Applications 1, 211–243.

Tod62 TODD, J. (ED) (1962) *Survey of Numerical Analysis*, McGraw Hill.

Wes68 WESTLAKE, JOAN R. (1968) *A Handbook of Numerical Matrix Inversion and Solution of Linear Equations,* John Wiley, New York.

Wil63 WILKINSON, J. H.(1963) *Rounding Errors in Algebraic Processes*, Notes on Applied Science No. 32, H.M.S.O., London, 161 pp.

Wil65 WILKINSON, J. H.(1965) *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 662 pp.

Wil70 WILKINSON, J. H.(1970) *Elementary Proof of the Wielandt-Hoffman Theorem and of Its Generalization*, Stanford University Computer Science Dept. Tech. Report No. CS 150.

Of these references, the ones by Givens and Tewarson have not been seen by the author. There are of course many papers that have been written on Lanczos' method of Minimized Iterations, several such references are given in the bibliographies in the books by Wilkinson (1965) and Householder (1964).