1968

Norm reduction algorithms for eigenvalues and eigenvectors of a matrix

Richard Frank Sincovec
Iowa State University

69-9894

SINCOVEC, Richard Frank, 1942-

NORM REDUCTION ALGORITHMS FOR EIGENVALUES AND EIGENVECTORS OF A MATRIX.

Iowa State University, Ph.D., 1968
Mathematics

University Microfilms, Inc., Ann Arbor, Michigan

NORM REDUCTION ALGORITHMS FOR EIGENVALUES AND EIGENVECTORS OF A MATRIX

by

Richard Frank Sincovec

A Dissertation Submitted to the
Graduate Faculty in Partial Fulfillment of
The Requirements for the Degree of
DOCTOR OF PHILOSOPHY

Major Subject: Applied Mathematics

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

Head of Major Department

Signature was redacted for privacy.

Dean of Graduate College

Iowa State University
Ames, Iowa

1968

TABLE OF CONTENTS

I. PRESENT METHODS AND PROBLEMS ASSOCIATED WITH THEM

In this chapter several known methods for finding the eigenvalues and eigenvectors of matrices will be discussed. Some of the difficulties of these methods will be indicated since they lead to the development of the method to be presented in this thesis.

No attempt will be made to discuss all known methods for the eigenvalue problem. Wilkinson (1965) has an excellent discussion of known methods for the eigenvalue problem which includes the relationship between these algorithms and a critical assessment of them based on rigorous error analysis. White (1958) has an excellent summary for solving the eigenvalue problem which includes recommendations of the most practical method or methods for real symmetric, Hermitian, real non-symmetric, and complex matrices.

If A is an arbitrary square matrix, then the scalar λ is called an eigenvalue, characteristic number, proper value, or latent root of the matrix A if the determinant of $(\lambda I-A)$ is equal to zero. The determinant of $(\lambda I-A)$ is denoted by $\det(\lambda I-A)$ and $\phi(\lambda) = \det(\lambda I-A) = 0$ is a polynomial equation of degree n, the order of A, and is called the characteristic equation of A. Associated with any root λ of this equation there is at least one non-null vector $\vec{x}$ called an eigenvector, characteristic vector, proper vector or latent vector be-

longing to λ and satisfying $A\vec{x} = \lambda\vec{x}$. The number of independent eigenvectors belonging to an eigenvalue may at most equal the multiplicity of the root λ of the characteristic equation.

If one wanted only the eigenvalues of a symmetric matrix A then the most direct theoretical approach to the problem would be to solve the characteristic polynomial for the eigenvalues. That is, one would seek a procedure for finding the coefficients $a_1, a_2, \ldots, a_n$ of the equation

$$\phi(\lambda) = \lambda^n + a_1\lambda^{n-1} + \ldots + a_n = 0 \qquad (1.1)$$

and a procedure for finding the n roots of (1.1). Goldstine, Murray, and von Neumann (1959) discuss the known methods for doing the preceding and conclude that this approach is completely unsatisfactory. Their results can be summed up as follows. Calculating the eigenvalues of a symmetric matrix by means of its characteristic equation requires carrying very large numbers of digits throughout the calculations if reasonable precisions are desired for the final results. In most computing instruments this need for many digits in excess of the normal number can be met but only at the price of considerably slowing down the computation. This objection raises a strong presumption that the entire procedure is unstable. Finally, these methods seem to require excessive matrix multiplications plus the multiplication and error involved in finding the roots of (1.1).

Several well qualified researchers including White, Householder, Todd, and Wilkinson seem to agree that Wilkinson's variation of the power method, an iteration method, is one of the better methods for real or complex nonsymmetric matrices of high order. Wilkinson's method combines the power method, displacement of the origin, Aitken's delta process and deflation.

The basic idea of this method, the power method with subsequent deflation, will now be illustrated and its failures indicated.

If $\vec{u}_1,\vec{u}_2,\ldots,\vec{u}_n$ are a linearly independent set of eigenvectors for the n x n matrix A and if λ_1, corresponding to $\vec{u}_1$, is a real simple eigenvalue exceeding all others in modulus, then the sequence of vectors $\vec{v}_{k+1} = A\vec{v}_k (k=0,1,\ldots)$ approaches the eigenvector $\vec{u}_1$ belonging to λ_1 where $\vec{v}_0$ is an arbitrary vector not orthogonal to $\vec{u}_1$. This can be seen easily by assuming that the eigenvectors from a basis for the n-dimensional vector space and hence the vector $\vec{v}_0$ can be written in the form:

$$\vec{v}_0 = c_1\vec{u}_1 + c_2\vec{u}_2 + \ldots + c_n\vec{u}_n. \qquad (1.2)$$

If $c_i \neq 0$ $(i=1,2,\ldots,n)$ in Equation (1.2), then since $c_i\vec{u}_i$ $(i=1,2,\ldots,n)$ is also an eigenvector it can be assumed that the original eigenvectors $\vec{u}_1,\vec{u}_2,\ldots,\vec{u}_n$ are such that $\vec{v}_0 = \vec{u}_1+\vec{u}_2+ \ldots + \vec{u}_n$. Then it follows that

$$\vec{v}_1 = A\vec{v}_0 = A\vec{u}_1 + A\vec{u}_2 + \ldots + A\vec{u}_n \qquad (1.3)$$

$$= \lambda_1\vec{u}_1 + \lambda_2\vec{u}_2 + \ldots + \lambda_n\vec{u}_n,$$

and in general

$$\vec{v}_k = A\vec{v}_{k-1} = A^k\vec{v}_0 = \lambda_1^k\vec{u}_1 + \lambda_2^k\vec{u}_2 + \ldots + \lambda_n^k\vec{u}_n. \qquad (1.4)$$

Thus if $|\lambda_1| > |\lambda_2| \geq \ldots \geq |\lambda_n|$, then ultimately the term $\lambda_1^k\vec{u}_1$ will dominate and the sequence $\{\vec{v}_k\}$ will tend to a vector in the direction $\vec{u}_1$. So for large k, $\vec{v}_{k+1} \doteq \lambda_1\vec{v}_k$. Hence with any vector $\vec{w} \neq \vec{0}$, an approximate value of λ_1 is obtained from $\vec{w}^*A\vec{v}_k = \lambda_1\vec{w}^*\vec{v}_k$. Convergence, when it occurs can be accelerated by application of Aitken's delta-square process componentwise to the vectors in the sequence.

In the case of a pair of complex roots, convergence does not occur, but in the limit $\vec{v}_k$ becomes parallel to the plane of the two eigenvectors. Then if $\mu_k = \vec{w}^*\vec{v}_k$ the roots of the equation

$$\det\begin{bmatrix} 1 & \mu_k & \mu_{k+1} \\ \lambda & \mu_{k+1} & \mu_{k+2} \\ \lambda^2 & \mu_{k+2} & \mu_{k+3} \end{bmatrix} = 0 \qquad (1.5)$$

approach the roots λ_1 and λ_2 as k increases. In fact, this is true whether or not λ_1 and λ_2 are equal in modulus, but provided only both exceed all others in modulus. For a complete discussion of this case see Householder (1964).

When any eigenvalue and eigenvector are known, it is possible to apply deflation as follows. It is well known,

Hotelling (1933), that the matrix $A - \lambda_1 \vec{x}_1 \vec{y}_1^*$, where $\vec{x}_1$ and $\vec{y}_1^*$ are the eigenvectors corresponding to λ_1 and satisfying $\vec{y}_1^* \vec{x}_1 = 1$, has the same set of eigenvectors of A and also the same set of eigenvalues as A except that λ_1 is replaced by zero. This can easily be seen for the special case when a dominant eigenvalue λ_1 and vector $\vec{u}_1$ of a symmetric matrix A are known and the remaining eigenvalues satisfy $|\lambda_2| > |\lambda_3| > |\lambda_4| > \ldots > |\lambda_n|$. Then normalizing $\vec{u}_1$ such that $\vec{u}_1' \vec{u}_1 = 1$, one can define the deflated matrix as $A - \lambda_1 \vec{u}_1 \vec{u}_1'$. From the orthogonality of the $\vec{u}_i (i=1,2,\ldots,n)$,

$$(A - \lambda_1 \vec{u}_1 \vec{u}_1') \vec{u}_i = A\vec{u}_i - \lambda_1 \vec{u}_1 \vec{u}_1' \vec{u}_i = 0 \quad (i = 1) \qquad (1.6)$$
$$= \lambda_i \vec{u}_i \quad (i \neq 1).$$

Hence the eigenvalues of $A - \lambda_1 \vec{u}_1 \vec{u}_1'$ are $0, \lambda_2, \ldots, \lambda_n$ corresponding to eigenvectors $\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_n$, and the dominant eigenvalue λ_1 has been reduced to zero. Now the matrix $A - \lambda_1 \vec{u}_1 \vec{u}_1'$ also has a dominant eigenvalue, say λ_2. Therefore, the preceding power method can be applied to this new matrix to determine λ_2 and its corresponding eigenvector $\vec{u}_2$. Deflation can be used again to obtain another matrix with eigenvalues $0, 0, \lambda_3, \ldots, \lambda_n$. The entire process can be repeated until all n eigenvalues and eigenvectors have been obtained.

There are also other deflation techniques but regardless of which technique is used, the drawback is essentially the same. That is, since λ_1 and $\vec{u}_1$ are only approximations to

their exact values, they contain error and it is evident that the deflated matrix inherits this error. Thus when calculating λ_2 and $\vec{u}_2$ there is a natural error inherent in the process from the previous calculation of λ_1 and $\vec{u}_1$. Further deflations can only inherit all the error from preceding calculations. So as more eigenvalues are found, the error associated with the latter values increases. If the matrix is of high order, say 20 or higher, it would not be unreasonable to expect the middle range eigenvalues to be completely dominated by error and thus worthless. This observation indicates the need for a method capable of determining all eigenvalues and eigenvectors to equal, but arbitrary precision.

By considering the transformation methods and their difficulties, one can again see the need for the development of new and possibly better methods or the improvement of present methods for solving the eigenvalue problem. The various transformation methods, for both symmetric and non-symmetric matrices, are based on the fact that one can find a matrix Y, such that

$$Y^{-1}AY = B, \qquad (1.7)$$

where A and B have the same eigenvalues and the eigenvectors of A will be those of B multiplied by Y. This relationship between the eigenvectors of A and B follows since if $B\vec{x} = \lambda\vec{x}$, then $Y^{-1}AY\vec{x} = \lambda\vec{x}$ and $A(Y\vec{x}) = \lambda(Y\vec{x})$.

The methods of Jacobi, Givens, Householder, and Lanczos essentially require the formation of a sequence of transformations which transform the given matrix into an equivalent triple diagonal matrix. A triple diagonal matrix, also called tridiagonal or codiagonal, is a square matrix $T = (t_{ij})$ such that $t_{ij} = 0$ for all integers i and j satisfying $|i-j| > 1$. After obtaining the tridiagonal matrix, the eigenvalues are usually obtained by use of a Sturm sequence in the symmetric case. A full theory of the Sturm sequence in this connection is given by Givens (1953) and Ortega (1960).

In all of the preceding similarity transform methods except for the Jacobi method the difficulty is not so much in determining the eigenvalues, but in determining the corresponding eigenvectors. Further difficulties are encountered if two eigenvalues are close together since then the problem is ill-conditioned with respect to the determination of the eigenvectors as discussed by Wilkinson (1958a).

Also for transformation methods involving the sequence of matrices

$$A_k = Y_k' A_{k-1} Y_k \qquad (1.8)$$

with $A_0 = A$, one can make the following observation. If all the arithmetic were exact all matrices A_k would have the same eigenvalues. There are, however, various sources of error. For example, the non-zero elements of Y_k are computed from

some of those of A_{k-1}, so that Y_k may not be exactly orthogonal. Also the matrix multiplications in (1.8) cannot be performed exactly in finite-length arithmetic so that (1.8) is not satisfied exactly. Wilkinson (1962) shows that some instabilities can be circumvented by the use of multiple-length arithmetic in certain parts of the computation.

Another approach to the algebraic eigenvalue problem, the gradient method, was developed by Hestenes and Karush (1951) for the calculation of the eigenvalues and eigenvectors of a real symmetric matrix A. The method is based on the principle that the Rayleigh quotient

$$\mu(\vec{x}) = \frac{\vec{x}'A\vec{x}}{\vec{x}'\vec{x}} \qquad (1.9)$$

equals an eigenvalue if and only if $\vec{x}$ is an eigenvector of A. The maximum and minimum values of $\mu(\vec{x})$ correspond to the greatest and least eigenvalues which values occur when $\vec{x}$ is replaced by the corresponding eigenvector, $\vec{x}_M$ and $\vec{x}_m$. If $\vec{x}$ is neither $\vec{x}_M$ nor $\vec{x}_m$ then $\mu(\vec{x})$ does not have its extreme value.

The gradient method has many of the same difficulties that the power method has since in order to find further eigenvalues and eigenvectors it is necessary to use a deflation technique to reduce the problem to one in which the maximum eigenvalue and vector are no longer present. Also, this method is not very good if the two greatest (or

two least) eigenvalues are close together or for a near zero eigenvalue.

The difficulties encountered in the algorithms that were discussed above show the need for a method that will solve the eigenvalue problem accurately for matrices of high order. In the remainder of this thesis a class of algorithms based on simple product theory will be developed which shows considerable promise for calculating all eigenvectors with precision. These methods have the advantage of very low round-off error. The computation is done by modifying approximations to the eigenvectors and the original matrix is left unchanged. This procedure prevents propagation of round-off error from one iteration step to the next and from one eigenvector to the next as was discussed in this chapter. Eigenvectors and hence eigenvalues can be calculated quite rapidly to nearly the precision of the computer.

II. DEVELOPMENT OF THE NORM REDUCTION METHOD

A. Introduction

The essential features of a new iterative method for finding the eigenvectors of a Hermitian matrix are developed in this chapter. Many of the details, including the proofs to some of the theorems, will be omitted and can be found in Sincovec (1967) or Lambert and Sincovec (1968). The method can be adapted to the non-symmetric complex matrices as developed in Erisman (1967) with the usual complications but only the Hermitian case is discussed in detail in this thesis.

The following theorem which was originally proved in Sincovec (1967) is basic to the description of this algorithm. This theorem involves rank one matrices (called simple products in Bodewig (1959)) of the form $\vec{x}\vec{y}^*$ where $\vec{x}$ is a non-zero n-dimensional complex column vector and $\vec{y}^*$ is the conjugate transpose of such a vector.

Theorem 2.1: The simple product matrix $\vec{x}\vec{y}^*$ commutes with the complex n x n matrix A if and only if $\vec{x}$ is a column eigenvector of A and $\vec{y}^*$ is a row eigenvector of A corresponding to the same eigenvalue λ.

It is common to deal with an Hermitian matrix P + iQ by working with the 2n x 2n real symmetric matrix A, given by

$$A = \begin{bmatrix} P & -Q \\ Q & P \end{bmatrix}. \qquad (2.1)$$

This matrix has all the eigenvalues of P + iQ repeated twice and if $(\vec{u} + i\vec{v})$ is an eigenvector of (P + iQ), then the vectors $\vec{x}$ and $\vec{y}$ given by

$$\vec{x}' = (\vec{u}', \vec{v}'), \qquad \vec{y}' = (-\vec{v}', \vec{u}') \qquad (2.2)$$

are independent eigenvectors of A. Because of this observation only real symmetric matrices will be considered in the remainder of this thesis.

The iterative procedure for a real symmetric matrix is described as follows: An arbitrary non-zero starting vector, $\vec{x}_o$, determines a residual matrix $R_o = A\vec{x}_o\vec{x}_o' - \vec{x}_o\vec{x}_o'A$. A change vector, $\vec{g}_o$, and a scalar, t_o, are sought such that for $\vec{x}_1 = \vec{x}_o + t_o\vec{g}_o$, the residual matrix $R_1 = A\vec{x}_1\vec{x}_1' - \vec{x}_1\vec{x}_1'A$ has a smaller norm than the matrix R_o. The Euclidean norm for vectors, namely $||\vec{x}|| = (\vec{x}'\vec{x})^{\frac{1}{2}}$, is used as well as the Euclidean norm for the residual matrix, $||R|| = [tr(R'R)]^{\frac{1}{2}} = \left[\sum_{i,j=1}^{n} r_{ij}^2\right]^{\frac{1}{2}}$. The change procedure is then repeated iteratively until at some stage a vector $\vec{x}_{m+1} = \vec{x}_m + t_m\vec{g}_m$ is obtained which makes the norm of R_{m+1} sufficiently close to zero that the vector $\vec{x}_{m+1}$ can be accepted as an eigenvector in view of Theorem 2.1 above.

With this brief description of the iterative procedure

two questions come to mind: How is the set $\{\vec{g}_m\}$ chosen and how is the set of scalars $\{t_m\}$ chosen? Obviously $\vec{g}_o$ can be chosen so that the procedure converges in one step if $\vec{g}_o$ is taken in the direction of the vector $\vec{u} - \vec{x}_o$ where $\vec{u}$ is an eigenvector. Since a knowledge of the eigenvectors cannot be supposed when one is trying to find the eigenvectors, this choice of $\vec{g}_o$ is not practical. The means of choosing the set $\{\vec{g}_m\}$ is crucial to the rate of convergence. The set of scalars $\{t_m\}$ is chosen to optimize the norm reduction once the set $\{\vec{g}_m\}$ is decided upon.

B. Determination of $||R_{m+1}||^2$

In this section the theory underlying the calculations of $||R_{m+1}||^2$ is given with the assumption that $\vec{x}_i$ and $\vec{g}_i$ have already been determined for $i=0,1,\ldots,m$ and the scalars t_i have been determined for $i=0,1,\ldots,m-1$.

The scalar t_m is to be found so that the vector $\vec{x}_{m+1} = \vec{x}_m + t_m\vec{g}_m$ minimizes $||R_{m+1}||^2$. Now

$$R_{m+1} = A\vec{x}_{m+1}\vec{x}'_{m+1} - \vec{x}_{m+1}\vec{x}'_{m+1}A$$

$$= [A\vec{x}_m\vec{x}'_m - \vec{x}_m\vec{x}'_mA] + t[A\vec{x}_m\vec{g}'_m - \vec{x}_m\vec{g}'_mA] \qquad (2.3)$$

$$+ t[A\vec{g}_m\vec{x}'_m - \vec{g}_m\vec{x}'_mA] + t^2[A\vec{g}_m\vec{g}'_m - \vec{g}_m\vec{g}'_mA].$$

In order to simplify the notation, the subscripts on $\vec{x}_m$, t_m, and $\vec{g}_m$ will be omitted whenever the discussion is concerned

with the single iteration step of reducing $||R_m||^2$ to $||R_{m+1}||^2$ or whenever the subscript is obvious from the context. Thus (2.3) becomes

$$R_{m+1} = R_m + t[A\vec{x}\vec{g}' - \vec{x}\vec{g}'A] + t[A\vec{g}\vec{x}' - \vec{g}\vec{x}'A]$$

$$+ t^2[A\vec{g}\vec{g}' - \vec{g}\vec{g}'A].$$

The next step is to obtain an expression for $R'_{m+1}R_{m+1}$. Because of the skew-symmetry of R_{m+1}, it suffices to calculate the negative of R^2_{m+1}. The details which are lengthy are presented in Sincovec (1967), and will not be repeated here except to comment that the results give a polynomial in t with matrix coefficients involving sums of scalar inner products multiplied by simple products such as $\vec{a}'\vec{b}(\vec{c}\vec{d}')$. Finally, the trace of $R'_{m+1}R_{m+1}$ is easily obtained by observing that $tr[\vec{a}'\vec{b}(\vec{c}\vec{d}')] = \vec{a}'\vec{b} \cdot \vec{d}'\vec{c}$ and by using the fact that the trace of a sum of matrices is the sum of the traces. The final result is

$$||R_{m+1}||^2 = c_o + c_1 t + c_2 t^2 + c_3 t^3 + c_4 t^4 \qquad (2.4)$$

where

$$c_o = ||R_m||^2 = 2[(\vec{x}'\vec{x})(\vec{v}'\vec{v}) - (\vec{x}'\vec{v})^2], \qquad (2.5)$$

$$c_1 = 4[(\vec{v}'\vec{v})(\vec{x}'\vec{g}) - 2(\vec{v}'\vec{x})(\vec{x}'\vec{h}) + (\vec{h}'\vec{v})(\vec{x}'\vec{x})], \qquad (2.6)$$

$$c_2 = 2[\vec{h}'\vec{h})(\vec{x}'\vec{x}) - 2(\vec{v}'\vec{x})(\vec{g}'\vec{h}) + (\vec{v}'\vec{v})(\vec{g}'\vec{g})$$

$$+ 4(\vec{h}'\vec{v})(\vec{x}'\vec{g}) - 4(\vec{h}'\vec{x})^2], \qquad (2.7)$$

$$c_3 = 4[(\vec{h}'\vec{h})(\vec{x}'\vec{g}) + (\vec{h}'\vec{v})(\vec{g}'\vec{g}) - 2(\vec{g}'\vec{h})(\vec{h}'\vec{x})], \quad (2.8)$$

$$c_4 = 2[(\vec{h}'\vec{h})(\vec{g}'\vec{g}) - (\vec{g}'\vec{h})^2], \quad (2.9)$$

and the vectors $\vec{v}$ and $\vec{h}$ are defined by

$$\vec{v} = A\vec{x}, \quad (2.10)$$

$$\vec{h} = A\vec{g}. \quad (2.11)$$

Even though the details of obtaining the formula for $||R_{m+1}||^2$ are lengthy, the final result is quite simple and is easy to calculate on a computer. One needs only to calculate $\vec{v}$ and $\vec{h}$ from (2.10) and (2.11), respectively, and then form the inner products to assemble the values for the c's.

Equation (2.4) is now rewritten as

$$N_{m+1} = N_m + F(t; \vec{x}_m, \vec{g}_m) \quad (2.12)$$

where

$$N_{m+1} = ||R_{m+1}||^2, \quad (2.13)$$

$$N_m = ||R_m||^2, \text{ and} \quad (2.14)$$

$$F(t; \vec{x}_m, \vec{g}_m) = c_1 t + c_2 t^2 + c_3 t^3 + c_4 t^4 \quad (2.15)$$

The c's in the polynomial (2.15) are evaluated from (2.5) to (2.9) with $\vec{x}$ replaced by $\vec{x}_m$ and $\vec{g}$ replaced by $\vec{g}_m$. When $\vec{x}$ and $\vec{g}$ are to be regarded as parameters in (2.5) to (2.9), this polynomial will be written $F(t;\vec{x},\vec{g})$. The polynomial $F(t;\vec{x}_m,\vec{g}_m)$ will be denoted simply by $F_m(t)$.

It is clear from Equation (2.12) that to make $N_{m+1} < N_m$ a

value of t must be chosen to make $F_m(t)$ negative.

C. The Nature of the Polynomial $F_m(t)$

In the discussion of the polynomial $F_m(t)$, a very useful quantity is the gradient of N_m. This gradient is given by

$$\vec{\nabla}N_m = 4[(\vec{v}'\vec{v})\vec{x} - 2(\vec{x}'\vec{v})\vec{v} + (\vec{x}'\vec{x})A\vec{v}] \qquad (2.16)$$

and is obtained by differentiating (2.14), with $||R_m||^2$ expressed in the form (2.5), with respect to each component of $\vec{x}$ and assembling these derivatives, in order, as a vector.

The following theorem gives some useful information about the polynomial $F_m(t)$. The proof of this theorem can be found in Lambert and Sincovec (1968).

Theorem 2.2: If $\vec{x} \equiv \vec{x}_m \neq \vec{0}$, $F_m(t) \not\equiv 0$, and if the matrix A satisfies a minimum function of degree greater than two, then the following are equivalent:

(i) $||R_m|| = 0$,

(ii) $\vec{\nabla}N_m = \vec{0}$,

(iii) $F_m(t) = 0$ has at least a double root at t=0 for all choices of $\vec{g} \equiv \vec{g}_m \neq 0$,

(iv) $\vec{x}$ is an eigenvector of A.

Theorem 2.2 gives a set of equivalent conditions which, if fulfilled at some stage, m, of the iterative process, imply that $\vec{x}_m$ is an eigenvector of the given matrix A. It can be shown that $F_m(t) \equiv 0$ for non-zero $\vec{x}_m$ and $\vec{g}_m$ if and

only if $\vec{x}_m$ and $\vec{g}_m$ are both eigenvectors of A corresponding to the same eigenvalue. Several exceptional cases can occur for which $F_m(t) = 0$ has a double root at $t = 0$ but $\vec{x}_m$ is not an eigenvector of A. One case occurs when $\vec{g}_m$ is an eigenvector of A such that $\vec{g}_m' \vec{x}_m = 0$. In this case it can be shown that $F_m(t) \equiv c_2 t^2$ with $c_2 \neq 0$. Another possibility is that $\vec{g}_m$ is orthogonal to $\vec{\nabla} N_m$, so that $c_1 = 0$ again implying that $F_m(t)$ has a double root at $t = 0$. It is always apparent when these exceptional cases occur because $\vec{\nabla} N_m$ and $||R_m||$ will not be zero. These cases are mentioned to emphasize that $F_m(t) = 0$ must have a double root at $t = 0$ for <u>any</u> choice of $\vec{g}_m$ in order that $\vec{x}_m$ be an eigenvector of A.

Because Theorem 2.2 is used many times in the following discussion, it will be assumed that the matrix A satisfies a minimum function of degree greater than two and that the vector $\vec{x}_o \neq \vec{0}$.

The next theorem gives a condition which guarantees that the norm N_{m+1} can be made smaller than N_m for the proper choice of t.

<u>Theorem 2.3</u>: The polynomial $F_m(t)$ has an absolute minimum less than zero if $\vec{g}' \vec{\nabla} N_m = c_1 \neq 0$.

This theorem is a direct consequence of the form of $F_m(t)$. The details are given in Lambert and Sincovec (1968).

If $\vec{g}_m$ is chosen to satisfy the requirement of Theorem

2.3, then the nature of the polynomial $F_m(t)$ suggests that the value of t yielding the absolute minimum of $F_m(t)$ can be found from the roots of $\frac{d}{dt}F_m(t) = F_m'(t) = 0$. A real root of $F_m'(t) = 0$ yielding the minimum value of $F_m(t)$ is chosen for the optimum value of t. If $F_m'(t) = 0$ has only one real root, this root is the optimum one. If there are three real roots, the optimum one is either the largest or the smallest and these can be easily determined by testing. Note that there is an interval containing this optimum value of t for which $F_m(t)<0$. Thus the optimum value need not be calculated with extreme precision.

D. Conditions on the Change Vectors $\vec{g}$

It will be shown in this section that it is possible to select change vectors, $\vec{g}$, in such a way that the iterative procedure is convergent and that the sequence of eigenvector approximations, $\{\vec{x}_m\}$, does not converge to the zero vector. The following definition is needed.

Definition 2.1: Let $G_{\vec{x}_m}$ be the set of all real vectors $\vec{g}$ which satisfy the following: $\vec{g}'\vec{\nabla}N_m \neq 0$, $\vec{g}'\vec{x}_m = 0$.

Note that the set $G_{\vec{x}_m}$ is non-empty for any $\vec{x}_m$ which is not an eigenvector of A. This follows because $\vec{\nabla}N_m \neq \vec{0}$ for such a vector $\vec{x}_m$ by Theorem 2.2 and it is easily shown that $\vec{\nabla}N_m$ is not a scalar multiple of $\vec{x}_m$. When $\vec{x}_m$ is an eigenvector of A,

again by Theorem 2.2, $\vec{\nabla}N_m = \vec{0}$ and thus the set $G_{\vec{x}_m}$ is void.

Theorem 2.4: Let the sequence $\vec{x}_m$ be defined by $\vec{x}_{m+1} = \vec{x}_m + t_m\vec{g}_m$; m=0,1,2,..., where, for each m, $\vec{g}_m$ is chosen from the set $G_{\vec{x}_m}$ when $\vec{x}_m$ is not an eigenvector. Let the scalars t_m be chosen as the optimum root of $F_m'(t) = 0$ as given in Chapter II. B. when $\vec{x}_m$ is not an eigenvector and t_m is chosen zero otherwise. Then the sequences $\{\vec{x}_m\}$ and $\{N_m\}$ have the following properties:

(i) $||\vec{x}_{m+1}|| \geq ||\vec{x}_m||$; m=0,1,2,...,

(ii) $N_m \geq N_{m+1} \geq 0$; m=0,1,2,...,

(iii) the sequence $\{N_m\}$ converges.

Furthermore, equality holds in (i) and (ii) if and only if $\vec{x}_m$ is an eigenvector of A.

The following theorem gives an additional restriction on the choice of the change vector $\vec{g}_m$ to assure that the sequence $\{N_m\}$ converges to zero.

Theorem 2.5: Let $\vec{x}_{m+1} = \vec{x}_m + t_m\vec{g}_m$; m=0,1,2,..., where $\vec{g}_m$ is chosen as a continuous vector function of $\vec{x}_m$ from the set $G_{\vec{x}_m}$. Denote this function by $\vec{g}(\vec{x}_m)$. Then if the sequence of vectors $\{\vec{x}_m\}$ is bounded, i.e., $||\vec{x}_m|| \leq K$ for some positive K, then the sequence $\{N_m\}$ converges to zero.

Proof: From Theorem 2.4, the sequence $\{N_m\}$ converges. Assume that $N_m \to C>0$ as $m\to\infty$. The sequence $\{\vec{x}_m\}$ is a bounded infinite sequence so that it must contain a subsequence $\{\vec{x}_{m'}\}$ of vectors converging to a vector $\vec{y}$ such that

$$N(\vec{y}) = ||R(\vec{y})||^2 = ||A\vec{y}\vec{y}' - \vec{y}\vec{y}'A||^2 = C.$$

The vector $\vec{y}$ is not an eigenvector because $N(\vec{y}) = C>0$. The set $G_{\vec{y}}$ is non-void so that for $\vec{g}(\vec{y})$ chosen from $G_{\vec{y}}$ and for an optimum $t = t_{\vec{y}}$ chosen from the roots of $F'(t;\vec{y},\vec{g}(\vec{y})) = 0$, the number $F(t_{\vec{y}};\vec{y},\vec{g}(\vec{y})) = -\beta<0$. Now

$$N(\vec{y}+t_{\vec{y}}\,\vec{g}(\vec{y})) = N(\vec{y})+F(t_{\vec{y}};\vec{y},\vec{g}(\vec{y})) = C-\beta.$$

Since the norm function N is continuous and $F(t_{\vec{y}};\vec{y},\vec{g}(\vec{y}))$ is a continuous function of $\vec{y}$, for m' sufficiently large, elements from the sequence $\{\vec{x}_{m'}\}$ can be chosen such that $N(\vec{x}_{m'}+t_{m'}\vec{g}_{m'})$ is arbitrarily close to $C-\beta<C$ because

$$\lim_{m'\to\infty} N(\vec{x}_{m'}+t_{m'}\vec{g}_{m'}) = N[\lim_{m'\to\infty} (\vec{x}_{m'}+t_{m'}\vec{g}_{m'})]$$

$$= N(\vec{y}+t_{\vec{y}}\vec{g}(\vec{y}))=C-\beta.$$

This is a contradiction of the assumption that $N_m \to C>0$ as $m\to\infty$.

Theorems 2.4 and 2.5 give some desirable properties of the change vectors $\vec{g}_m$. These properties are that $\vec{g}_m$ be chosen as a continuous function of $\vec{x}_m$ and further that $\vec{g}_m$ be orthogonal to $\vec{x}_m$ and not orthogonal to $\vec{\nabla}N_m$. Vectors, $\vec{g}_m$, having these properties can be easily calculated from $\vec{x}_m$ and A as will be shown in Chapter III. The property of boundedness on the norm of the vector $\vec{x}_m$ as required in

Theorem 2.5 is no serious restriction from the computational standpoint because these vectors can be normalized occasionally, say at every tenth iteration. This step also reduces N_m so that convergence of the sequence $\{N_m\}$ is not hindered.

III. CHOICES OF THE CHANGE VECTORS $\vec{g}$

A. The Modified Gradient Method

One suggested choice of $\vec{g}_m$ is as follows: Choose

$$\vec{g}_m = \vec{\nabla}N_m - c_m\vec{x}_m \tag{3.1}$$

where the constant c_m is chosen to make $\vec{g}_m$ orthogonal to $\vec{x}_m$. This value of $c_m = \dfrac{\vec{x}_m'\vec{\nabla}N_m}{\vec{x}_m'\vec{x}_m}$. Note that

$$\vec{g}_m'\vec{\nabla}N_m = \vec{\nabla}N_m'\vec{\nabla}N_m - \frac{\vec{x}_m'\vec{\nabla}N_m}{\vec{x}_m'\vec{x}_m}\,\vec{x}_m'\vec{\nabla}N_m$$

$$= \frac{1}{\vec{x}_m'\vec{x}_m}[(\vec{x}_m'\vec{x}_m)(\vec{\nabla}N_m'\vec{\nabla}N_m) - (\vec{x}_m'\vec{\nabla}N_m)^2] \geq 0.$$

Equality holds in this last expression if and only if $\vec{x}_m$ is parallel to $\vec{\nabla}N_m$ or if $\vec{\nabla}N_m = \vec{0}$. By Theorem 2.2, $\vec{\nabla}N_m = \vec{0}$ if and only if $\vec{x}_m$ is an eigenvector in which case the iterations would cease anyway. It can be shown that $\vec{x}_m$ is never parallel to $\vec{\nabla}N_m$ for if $\vec{x}_m$ is **not** an eigenvector and it is assumed that

$$\vec{\nabla}N_m = 4[(\vec{x}'\vec{x})A^2\vec{x} - 2(\vec{x}'A\vec{x})A\vec{x} + (\vec{x}'A^2\vec{x})\vec{x}] = \mu\vec{x}$$

for some μ and $\vec{x} = \vec{x}_m$ then it follows that

$$(\vec{x}'\vec{x})A^2\vec{x} - 2(\vec{x}'A\vec{x})A\vec{x} + (\vec{x}'A^2\vec{x} - \tfrac{\mu}{4})\vec{x} = \vec{0}.$$

This last equation cannot hold because $\vec{x}$, $A\vec{x}$, and $A^2\vec{x}$ are independent by the assumption that A satisfies a minimal function of degree greater than two.

This first suggested choice of $\vec{g}_m$ as shown above has the desirable properties required by Theorem 2.5. The following algorithm can now be defined.

Algorithm 1: Modified gradient method.

Step 1. Choose $\vec{x}_o \neq \vec{0}$, and normalize.

Step 2. Set m = 0.

Step 3. Calculate $\lambda_m = \dfrac{\vec{x}_m' A \vec{x}_m}{\vec{x}_m' \vec{x}_m}$.

Step 4. Calculate $||R_m||^2$ from (2.5).

Step 5. Test $||R_m|| \leq \varepsilon$ where ε is prescribed initially. If $||R_m|| > \varepsilon$ continue with Step 6. If not, select another starting vector for Step 1.

Step 6. Calculate $\vec{\nabla} N_m$ from (2.16).

Step 7. Calculate $\vec{g}_m$ from (3.1).

Step 8. Calculate t_m from the optimum root of the cubic polynomial equation $F_m'(t) = 0$ where $F_m(t) = F(t; \vec{x}_m, \vec{g}_m)$ as given by (2.15) with coefficients given by (2.5) to (2.9).

Step 9. Set $\vec{x}_{m+1} = \vec{x}_m + t_m \vec{g}_m$.

Step 10. Change m to m+1 and repeat from Step 3 iteratively until $||R_m||$ satisfies the test in Step 5.

B. Generalized Methods

In the modified gradient method the change vector $\vec{g}$ at the mth iteration is composed of a particular linear combination of the direction vectors $\vec{z}_1$ and $\vec{z}_2$ given by

$$\vec{z}_1 = A\vec{x} - \frac{\vec{x}'A\vec{x}}{\vec{x}'\vec{x}}\vec{x} \quad \text{and} \quad \vec{z}_2 = A^2\vec{x} - \frac{\vec{x}'A^2\vec{x}}{\vec{x}'\vec{x}}\vec{x}$$

where $\vec{z}_1$ and $\vec{z}_2$ are independent by the minimality condition on A and $\vec{x} = \vec{x}_m$. This follows upon combining (3.1) and (2.16) to obtain $\vec{g}_m = -8(\vec{x}'A\vec{x})\vec{z}_1 + 4(\vec{x}'\vec{x})\vec{z}_2$. This suggests generalizing the norm reduction method by choosing possibly more direction vectors and then seeking some sort of optimum linear combination of the direction vectors for the change vector $\vec{g}_m$.

Let $W_m = \{\vec{w}_1, \vec{w}_2, \ldots, \vec{w}_r \mid r \leq n\}$ be a set of r independent vectors at the mth step of the iteration. For ν equal to either 0 or 1 define the set of non-zero direction vectors $Z_m(\nu) = \{\vec{z}_1, \vec{z}_2, \ldots, \vec{z}_q \mid q \leq r\}$ by

$$\vec{z}_1 = \vec{w}_{p_1} - c_1\vec{x}_m \tag{3.2}$$

$$\vec{z}_k = \vec{w}_{p_k} - c_k\vec{x}_m - \nu \sum_{i=1}^{k-1} b_{i_k}\vec{z}_i \qquad (k=2,3,\ldots,q)$$

where

$$c_k = \frac{\vec{w}_{p_k}'\vec{x}_m}{\vec{x}_m'\vec{x}_m}, \quad b_{i_k} = \frac{\vec{w}_{p_k}'\vec{z}_i}{\vec{z}_i'\vec{z}_i}, \quad \text{and} \quad \vec{w}_{p_k} \in W_m \tag{3.3}$$

with $p_k \in \{1,2,\ldots,r\}$ such that $p_j \neq p_k$ for $j \neq k$. Clearly the

set of direction vectors $Z_m(1)$ contains at most r non-zero orthogonal vectors unless r = n and in this case $Z_m(1)$ contains at most n-1 independent vectors since there are at most n-1 vectors orthogonal to $\vec{x}_m$. The set $Z_m(0)$ does not require the $\vec{z}$'s to be orthogonal to each other and hence it is possible for $Z_m(0)$ to be a dependent set of non-zero vectors. Define q to be the number of independent non-zero vectors in $Z_m(\nu)$. In the generalized version one seeks a change vector $\vec{g}$ at the mth step which is a linear combination of all or some of the independent vectors of $Z_m(\nu)$. That is,

$$\vec{g} = \sum_{i=1}^{p} \alpha_i \vec{z}_i \tag{3.4}$$

where $\vec{z}_i \epsilon Z_m(\nu)$ (i=1,2,...,p) and $p \leq q$ is the number of vectors from $Z_m(\nu)$ used to form $\vec{g}$ at the mth step. The set of α_i (i=1,2,...,p) are parameters to be determined in some optimum way so as to reduce $||R_{m+1}||^2$. In general, p = p(m), that is, p will be considered as a function of the mth step of the iteration.

In view of the preceding discussion an entire class of norm reduction methods can be defined in terms of W_m, ν, and p(m). An algorithm is defined by selecting a set W_m, generating the corresponding set $Z_m(\nu)$ for ν equal 0 or 1, and choosing an appropriate p at each iteration. In the remainder of this chapter a technique will be developed for optimally choosing the α's and for updating $\vec{x}_m$ so as to

reduce $||R_{m+1}||^2$.

Using this notation, Algorithm 1 is defined with $W_m = \{A\vec{x}_m, A^2\vec{x}_m\}$ and

$$Z_m(0) = \{A\vec{x}_m - \frac{\vec{x}_m'A\vec{x}_m}{\vec{x}_m'\vec{x}_m}\vec{x}_m,\ A^2\vec{x}_m - \frac{\vec{x}_m'A^2\vec{x}_m}{\vec{x}_m'\vec{x}_m}\vec{x}_m\}.$$

In this method α_1 and α_2 are not parameters but are the constants $\alpha_1 = -8\vec{x}_m'A\vec{x}_m$ and $\alpha_2 = 4\vec{x}_m'\vec{x}_m$. Note that the set $W_m = \{\vec{x}_m, A\vec{x}_m, A^2\vec{x}_m\}$ yields the same $Z_m(0)$ and so W_m is not unique.

One method for determining the α's is to substitute $\vec{g}$ as given by (3.4) into (2.6) through (2.9) to obtain $F(t;\vec{x}_m,\vec{g})$. Since the α's are unknown parameters, $F(t;\vec{x}_m,\vec{g})$ is modified by replacing $t\vec{g}$ by $\vec{g}$ and $t\vec{h}$ by $\vec{h}$ which causes the unknown factor t to be absorbed into the unknown parameters $\alpha_1,\alpha_2,\ldots,\alpha_p$. Denoting the resulting expression by $F(\alpha_1,\alpha_2,\ldots,\alpha_p)$, Equation (2.12) becomes

$$N_{m+1} = N_m + F(\alpha_1,\alpha_2,\ldots,\alpha_p). \qquad (3.5)$$

Now $F(\alpha_1,\alpha_2,\ldots,\alpha_p)$ is a quartic polynomial in the α's and in order to find its minimum it is necessary to solve the following system of p simultaneous cubic equations:

$$\frac{\partial F(\alpha_1,\alpha_2,\ldots,\alpha_p)}{\partial \alpha_j} = 0 \qquad (j=1,2,\ldots,p). \qquad (3.6)$$

To solve the system given by (3.6) analytically would be a formidable undertaking. To solve (3.6) by an iterative technique such as Newton's method, a starting value for the

α's would have to be found such that convergence to a relative minimum with $F(\alpha_1,\alpha_2,\ldots,\alpha_p)<0$ could be guaranteed.

Because of these difficulties a more fruitful approach for determining the α's might be to linearize the system by considering the residual matrix $R_{m+\frac{1}{2}}$ defined by

$$R_{m+\frac{1}{2}} = A(\vec{x}_m + \vec{g})\vec{x}_m' - (\vec{x}_m' + \vec{g})\vec{x}_m'A \qquad (3.7)$$

where $\vec{g}$ is given by (3.4) as a function of the α's. The norm $||R_{m+\frac{1}{2}}||^2$ is minimized by choosing the α's as the simultaneous solution of the system

$$\frac{\partial ||R_{m+\frac{1}{2}}||^2}{\partial \alpha_j} = 0 \qquad (j=1,2,\ldots,p). \qquad (3.8)$$

When the α's are determined by (3.8) they will be said to have been chosen in an optimum manner.

It can easily be shown that

$$||R_{m+\frac{1}{2}}||^2 = ||R_m||^2 + [2(\vec{v}_m'\vec{v}_m)(\vec{x}_m'\vec{g}) - 4(\vec{v}_m'\vec{x}_m)(\vec{v}_m'\vec{g}) + 2(\vec{x}_m'\vec{x}_m)(\vec{v}_m'\vec{h}) + (\vec{v}_m'\vec{v}_m)(\vec{g}'\vec{g}) - 2(\vec{v}_m'\vec{x}_m)(\vec{g}'\vec{h}) + (\vec{x}_m'\vec{x}_m)(\vec{h}'\vec{h})] \qquad (3.9)$$

where, as before $\vec{v}_m = A\vec{x}_m$ and $\vec{h} = A\vec{g}$. The quantity in the brackets is quadratic in the α's so that the equations given by (3.8) are linear equations in the α's.

To simplify the notation, let

$$Q_m = d_o A^2 - 2d_1 A + d_2 I \qquad (3.10)$$

where $d_j = \vec{x}_m' A_j \vec{x}_m$ $(j=0,1,2)$. Then Equation (3.9) can be written as follows:

$$||R_{m+\frac{1}{2}}||^2 = ||R_m||^2 + \vec{g}' Q_m \vec{g} + 2\vec{g}' Q_m \vec{x}_m \,. \qquad (3.11)$$

The system of equations (3.8) is then given by

$$\frac{\partial ||R_{m+\frac{1}{2}}||^2}{\partial \alpha_j} = 2\vec{z}_j' Q_m \vec{g} + 2\vec{z}_j Q_m \vec{x}_m = 0 \quad (j=1,2,\ldots,p). \qquad (3.12)$$

The simultaneous solution of (3.12) is denoted by $\alpha_i^{(m)}$ $(i=1,2,\ldots,p)$. The corresponding value of $\vec{g}$ as given by (3.4) with $\alpha_i=\alpha_i^{(m)}$ $(i=1,2,\ldots,p)$ is denoted by $\vec{g}_m$. The $\vec{g}_m$ vector gives a good direction in which to change the vector $\vec{x}_m$.

A technique for solving the system (3.12) will now be developed. Defining

$$P = (\vec{z}_1, \vec{z}_2, \ldots, \vec{z}_p) \qquad (3.13)$$

which is an n x p matrix and for clarity omitting all subscripts that denote the mth step, then the system (3.12) can be rewritten as

$$P'Q\vec{g} = -P'Q\vec{x} \qquad (3.14)$$

where $\vec{g} = P\vec{\alpha}$ and $\vec{\alpha}$ is the p-tuple, $\vec{\alpha}=(\alpha_1,\alpha_2,\ldots,\alpha_p)'$. Letting

$$B = P'QP \text{ and } \vec{c} = -PQ\vec{x}, \qquad (3.15)$$

the system to be solved for $\vec{\alpha}$ becomes

$$B\vec{\alpha} = \vec{c} \qquad (3.16)$$

and the resulting solution defines $\vec{g}_m$ at the mth step by (3.4).

The system (3.16) will be solved by an escalator technique since this allows one to accept a new direction vector only if the solution of (3.16) is well-determined and to reject a new direction vector if the solution becomes ill-determined as evidenced by the fact that the coefficient matrix B approaches singularity. Suppose that at the mth step $k<p(m)$ direction vectors have been successfully brought into the system (3.16) and that the system has been solved by calculating the inverse of the coefficient matrix. The quantities defined by Equations (3.13) and (3.15) clearly depend on the number of direction vectors. Since the assumption is that k direction vectors have been brought into the system, this dependence will be denoted by adding subscripts in terms of k to all the quantities that depend on the number of direction vectors. That is, at the kth stage of the escalation procedure the solution to (3.16) is given by

$$\vec{\alpha}_k = B_k^{-1} \vec{c}_k \tag{3.17}$$

where

$$B_k = P_k' Q P_k, \quad \vec{c}_k = -P_k' Q \vec{x}, \text{ and } P_k = (\vec{z}_1, \vec{z}_2, \ldots, \vec{z}_k). \tag{3.18}$$

Here B_k is a k x k matrix, $\vec{c}_k$ is a vector with k components, and P_k is an n x k matrix. The quantity $\vec{\alpha}_k$ is the k-tuple,

$\vec{\alpha}_k = (\alpha_{k_1}, \alpha_{k_2}, \ldots, \alpha_{k_k})'$ where the double subscripts are used to indicate that the first (k-1) elements of $\vec{\alpha}_k$ are not necessarily the same as those of $\vec{\alpha}_{k-1}$. It is desired to bring into the system a (k+1)st vector, $\vec{z}_{k+1} \varepsilon Z_m(\nu)$, and to alter $\vec{\alpha}_k$ so that the solution to the larger system

$$B_{k+1}\vec{\alpha}_{k+1} = \vec{c}_{k+1} \tag{3.19}$$

is obtained.

Partition the matrix B_{k+1} formed by the direction vectors $\vec{z}_i \varepsilon Z_m(\nu)$ $(i=1,2,\ldots,k+1)$ in the form

$$B \equiv B_{k+1} = \begin{bmatrix} B_k & \vec{b}_{k+1} \\ \vec{b}'_{k+1} & \beta_{k+1} \end{bmatrix} \tag{3.20}$$

where B_k is given by (3.18), $\vec{b}_{k+1} = P'_k Q \vec{z}_{k+1}$ is a vector with k components, and $\beta_{k+1} = \vec{z}_{k+1} Q \vec{z}_{k+1}$ is a scaler. Upon the addition of $\vec{z}_{k+1}$ to the system, the right hand side of (3.19) is given by

$$\vec{c}_{k+1} = -P'_{k+1} Q \vec{x} \tag{3.21}$$

where $P_{k+1} = (\vec{z}_1, \vec{z}_2, \ldots, \vec{z}_{k+1})$. Note that $\vec{c}_k$ is identical to the first k elements of $\vec{c}_{k+1}$ and that the (k+1)st element of $\vec{c}_{k+1}$ is given by $-\vec{z}'_{k+1} Q \vec{x}$.

Since the system (3.17) was solved using B_k^{-1} then the system (3.19) can be solved by finding B_{k+1}^{-1} in the following manner. The fact that B is symmetric implies that B^{-1} will also be symmetric. So if X, $\vec{y}$, and ω can be found such that

$$\begin{bmatrix} B_k & \vec{b}_{k+1} \\ \vec{b}'_{k+1} & \beta_{k+1} \end{bmatrix} \begin{bmatrix} X & \vec{y} \\ \vec{y}' & \omega \end{bmatrix} = \begin{bmatrix} I & \vec{0} \\ \vec{0}' & 1 \end{bmatrix} \tag{3.22}$$

then $B_{k+1}^{-1} = \begin{bmatrix} X & \vec{y} \\ \vec{y}' & \omega \end{bmatrix}$. From (3.22) the following set of simultaneous equations is obtained:

$$B_k X + \vec{b}_{k+1}\vec{y}' = I, \tag{3.23}$$

$$\vec{b}'_{k+1}X + \beta_{k+1}\vec{y}' = \vec{0}', \tag{3.24}$$

$$B_k\vec{y} + \omega\vec{b}_{k+1} = \vec{0}, \tag{3.25}$$

$$\vec{b}'_{k+1}\vec{y} + \omega\beta_{k+1} = 1. \tag{3.26}$$

Equation (3.25) implies that

$$\vec{y} = -\omega B_k^{-1}\vec{b}_{k+1} \tag{3.27}$$

and using this in (3.26) gives

$$\omega = 1/(\beta_{k+1} - \vec{b}'_{k+1}B_k^{-1}\vec{b}_{k+1}) \tag{3.28}$$

and so $\vec{y}$ can be found from Equation (3.27). From Equation (3.23),

$$X = B_k^{-1}(I - \vec{b}_{k+1}\vec{y}') \tag{3.29}$$

and using (3.27)

$$X = B_k^{-1} + \omega B_k^{-1}\vec{b}_{k+1}\vec{b}'_{k+1}B_k^{-1} . \tag{3.30}$$

Now using (3.27), (3.28), and (3.30), $\vec{\alpha}_{k+1}$ can be found by

$$\vec{\alpha}_{k+1} = B_{k+1}^{-1}\vec{c}_{k+1}.$$

The preceding procedure for bringing into the system each new $\vec{z}_{k+1}$ can be successively repeated at the mth step until p(m) direction vectors have been brought into the system. The reason for using this method is to enable one to bring into the system $\vec{z}_{k+1}$ only if B_{k+1}^{-1} will be well-determined without losing the information already obtained for $\vec{z}_i$ (i=1,2,...,k). The ω defined by Equation (3.28) can be used as a measure of ill-conditioning for bringing in or leaving out of the system $\vec{z}_{k+1}$. Clearly, B_{k+1} is singular if ω is undefined, so if at the (k+1)st direction vector the magnitude of ω is exceptionally large as compared to the ω's that occurred at vectors 0 through k, then $\vec{z}_{k+1}$ would not be brought into the system. Two alternatives then exist, a different $\vec{z}_{k+1} \varepsilon Z_m(\nu)$ can be chosen, if possible, or $\vec{x}_m$ can be updated. This technique enables one to achieve maximum accuracy in the determination of the α's but it is at the expense of possibly fewer direction vectors. The preceding discussion indicates that p(m) should be considered only as an upper bound for the number of direction vectors desired at the mth iteration since it might not be possible to bring into the system exactly p(m) direction vectors without suffering a severe loss of accuracy.

Once $\vec{g}_m$ is determined, then to complete the mth iteration $\vec{x}_m$ is updated to give $\vec{x}_{m+1}$. Certainly if t_m is the optimal solution of $F_m'(t) = 0$ with $\vec{g}_m$ given by (3.4), then

$||R_{m+1}||^2 \leq ||R_m||^2$ where $\vec{x}_{m+1} = \vec{x}_m + t_m\vec{g}_m$. However, it will now be shown that the solution of this cubic at every stage of the iteration process can be avoided and that $\vec{x}_m$ can usually be updated by $\vec{x}_{m+1} = \vec{x}_m + \vec{g}_m$ with $||R_{m+1}||^2 \leq ||R_m||^2$ satisfied.

To show the relationship of $||R_{m+\frac{1}{2}}||^2$ of Equation (3.9) to $||R_{m+1}||^2$ as given by Equation (2.12), the bracketed expression on the right hand side of (3.9) is modified by replacing $\vec{g}$ by $t\vec{g}_m$ and $\vec{h}$ by $t\vec{h}_m$. The expression becomes

$$f(t) = t^2[(\vec{h}_m'\vec{h}_m)(\vec{x}_m'\vec{x}_m) - 2(\vec{h}_m'\vec{g}_m)(\vec{v}_m'\vec{x}_m) + (\vec{g}_m'\vec{g}_m)(\vec{v}_m'\vec{v}_m)]$$

$$+ 2t[(\vec{h}_m'\vec{v}_m)(\vec{x}_m'\vec{x}_m) - 2(\vec{g}_m'\vec{v}_m)(\vec{v}_m'\vec{x}_m) + (\vec{g}_m'\vec{x}_m)(\vec{v}_m'\vec{v}_m)]$$

$$= t^2\vec{g}_m'Q_m\vec{g}_m + 2t\vec{g}_m'Q_m\vec{x}_m \quad . \qquad (3.31)$$

In order to guarantee that f(1) is an absolute minimum of f(t) one must show that the coefficient of t^2 in Equation (3.31) is positive for that choice of $\vec{g}_m$ determined by (3.12) and (3.4) or in general show that it is positive for all choices of $\vec{g}_m$.

Lemma 3.1: The matrix Q_m is positive definite if $\vec{x}_m$ is not an eigenvector of the matrix A.

Proof: If $\vec{x}_m$ is an eigenvector of the matrix A corresponding to the eigenvalue λ, then $Q_m = (\vec{x}_m'\vec{x}_m)(A-\lambda I)^2$. So for non-zero $\vec{g}$, $\vec{g}'Q_m\vec{g} \geq 0$ with equality holding if and only

if $\vec{g}$ is an eigenvector of A corresponding to λ. In this case Q_m is positive semi-definite. If $\vec{x}_m$ is not an eigenvector of A then $Q_m \neq 0$ since A satisfies a minimal function of degree greater than two. So for all

$$\vec{g} \neq \vec{0},\ \vec{g}'Q_m\vec{g} = (\vec{x}_m'\vec{x}_m)(\vec{g}'A^2\vec{g}) - 2(\vec{x}_m'A\vec{x}_m)(\vec{g}'A\vec{g}) + (\vec{x}_m'A^2\vec{x}_m)(\vec{g}'\vec{g})$$

$$= ||A\vec{x}_m\vec{g}' - \vec{x}_m\vec{g}'A||^2 > 0$$

by Theorem 2.1. The last equality follows from

$$||A\vec{x}_m\vec{g}' - \vec{x}_m\vec{g}'A||^2 = tr[(\vec{g}\vec{x}_m'A - A\vec{g}\vec{x}_m')(A\vec{x}_m\vec{g}' - \vec{x}_m\vec{g}'A)]$$

$$= d_2 tr(\vec{g}\vec{g}') - d_1 tr(A\vec{g}\vec{g}') - d_1 tr(\vec{g}\vec{g}'A) + d_o tr(A\vec{g}\vec{g}'A)$$

$$= \vec{g}'Q_m\vec{g}$$

where $d_j = \vec{x}_m'A^j\vec{x}_m$ $(j=0,1,2)$.

Theorem 3.1: If $\vec{x}_m$ is not an eigenvector of A, the polynomial f(t) has an absolute minimum less than zero if and only if $\vec{g}'Q_m\vec{x}_m = c_1 \neq 0$.

Proof: If $\vec{x}_m$ is not an eigenvector of A and if $c_1 \neq 0$, then the form of f(t) as given in equation (3.31) and Lemma 3.1 imply that f(t) is concave upward and passes through the origin with non-zero slope. Thus f(t) has an absolute minimum less than zero. Conversely, if $\vec{x}_m$ is not an eigenvector of A and if f(t) has an absolute minimum less than

zero, then clearly this is possibly only if $c_1 \neq 0$.

Equation (3.9) can thus be written as

$$||R_{m+\frac{1}{2}}||^2 = ||R_m||^2 + f(1) \qquad (3.32)$$

where f(1) is the minimum value of the quadratic f(t). See Figure 3.1. As before if $R_{m+1} = A(\vec{x}_m+t\vec{g}_m)(\vec{x}_m+t\vec{g}_m)' - (\vec{x}_m+t\vec{g}_m)(\vec{x}_m+t\vec{g}_m)'A$, then Equation (2.4) can be written in the alternative form

$$||R_{m+1}||^2 = ||R_m||^2 + 2f(t) + g(t) \qquad (3.33)$$

where f(t) is given by (3.31) and

$$g(t) = c_4t^4 + c_3t^3 - 8(\vec{v}_m'\vec{g}_m)^2t^2 \qquad (3.34)$$

with c_4 given by (2.9) and c_3 given by (2.8).

Obviously, the polynomial 2f(t) + g(t) is identically the polynomial $F(t;\vec{x}_m,\vec{g}_m)$ of Equation (2.15) which was minimized as described in Chapter II by solving a cubic equation. Note that the polynomial g(t) has a double root at the origin and exactly one positive and one negative root. The general shape of the graph of g(t) is shown in Figure 3.2.

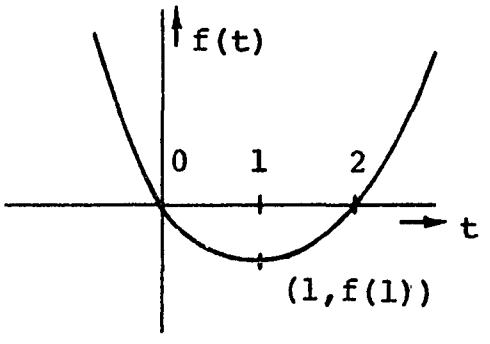

Figure 3.1. A graph of f(t)

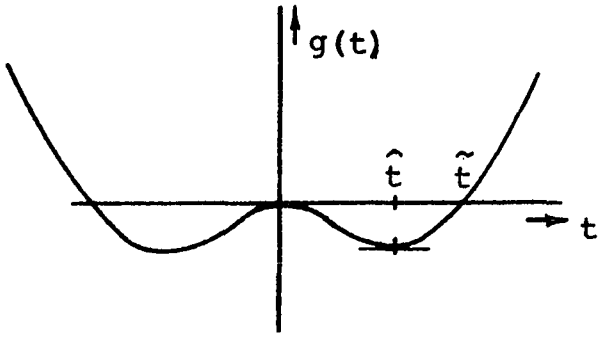

Figure 3.2. A graph of g(t)

As indicated in Figure 3.2, the point $\hat{t}$ is the positive value of t where g(t) has a minimum and the point $\tilde{t}$ is the positive root of g(t) = 0. Either of these points can be found by solving a quadratic equation in t because of the t^2 factor in g(t).

The value $\hat{t}$ can be tested to see if it is greater than or equal to one or less than one. If $\hat{t} < 1$, then $||R_{m+1}||^2$ is certainly less than $||R_m||^2$ for $t = \hat{t}$. If $\hat{t} \geq 1$, then $||R_{m+1}||^2$ is less than $||R_m||^2$ for t = 1 and is also less than $||R_{m+\frac{1}{2}}||^2$. These statements follow from the observation that when $\hat{t}<1$, both $f(\hat{t})$ and $g(\hat{t})$ are negative and when $\hat{t}\geq 1$, then both f(1) and g(1) are negative. Even when $\hat{t}<1$, g(1) might be negative and f(1) is always negative so the choice t=1 might provide a norm reduction. It has been the author's experience that this is usually the case.

With the previous discussion as background, the following generalized algorithm can be defined.

Generalized Algorithm:

For a given W_m, ν, and p perform the following steps:

Step 1 through Step 5 are identical to Algorithm 1.

Step 6.

(a) Set k = 1.

(b) Select $\vec{z}_k \varepsilon Z_m(\nu)$.

(c) Calculate the kth component of $\vec{c}_k$ by (3.21).

(d) If $k = 1$, set $B_k^{-1} = 1/B_1$ where $B_1 = \vec{z}_1' Q_m \vec{z}_1$ and go to (h). If $k \neq 1$, calculate $\vec{b}_k = P_{k-1} Q_m \vec{z}_k$ and $\beta_k = \vec{z}_k' Q_m \vec{z}_k$.

(e) Calculate $1/\omega$ by (3.28). If $1/\omega$ is not zero to τ decimal places (τ set initially) relative to B_1, proceed; otherwise, if possible to select a new $\vec{z}_k \varepsilon Z_m(\nu)$ go to (b), but if this is not possible, set $p = k-1$ and go to Step 7.

(f) Calculate B_k^{-1} using (3.27), (3.28) and (3.30).

(g) Calculate $\vec{\alpha}_k$.

(h) If $k<p$, set $k = k+1$ and go to (b), otherwise go to Step 7.

Step 7. Calculate $\vec{g}_m$ from (3.4).

Step 8. Calculate the coefficients of polynomials $f(t)$ and $g(t)$ from (3.31) and (3.34), respectively. Then solve $g'(t) = 0$ for the positive root $\hat{t}$ from the quadratic factor

$$4c_4 t^2 + 3c_3 t - 16(\vec{v}_m' \vec{g}_m)^2 = 0.$$

Step 9. Test $\hat{t}$ against 1 and select the appropriate branch. If $\hat{t}<1$, test to see if $2f(1) + g(1) \leq 2f(\hat{t}) + g(\hat{t})$ and if so set $\vec{x}_{m+1} = \vec{x}_m + \vec{g}_m$; otherwise, set $\vec{x}_{m+1} = \vec{x}_m + \hat{t}\vec{g}_m$. If $\hat{t} \geq 1$, set $\vec{x}_{m+1} = \vec{x}_m + \vec{g}_m$.

Step 10. Change m to m+1 and repeat from Step 3 iteratively until $||R_m||$ satisfies the test in Step 5.

Steps 8 and 9 of the Generalized Algorithm avoid solving the cubic equation $F_m'(t) = 0$ at each iteration. However, if one desires to choose t_m as the optimum solution of $F_m'(t) = 0$, then Steps 8 and 9 of the Generalized Algorithm can be replaced by Steps 8 and 9 of Algorithm 1. In practice this is usually not necessary since the rate of convergence appears to be the same in either case.

C. Special Methods

Clearly in the Generalized Algorithm $\vec{g}_m'\vec{x}_m = 0$ by the manner in which $Z_m(\nu)$ and $\vec{g}_m$ are constructed. However, nothing has been done to prevent $\vec{g}_m'\vec{\nabla}N_m$ from equaling zero which is contrary to the conditions of Theorem 2.4. If this should occur at the mth step for a given choice of W_m then this author recommends that W_m be chosen differently for this step so as to give a different $\vec{g}_m$. It has been this author's experience that the case $\vec{g}_m'\vec{\nabla}N_m = 0$ with $\vec{x}_m$ not an eigenvector will probably only occur in a hypothetical example constructed to give such a difficulty.

Several algorithms are now given which are based on the Generalized Algorithm of the last section.

Algorithm 2: Choose $W_m = \{A\vec{x}_m, A^2\vec{x}_m\}$, $\nu = 0$, $p(m) \leq 2$, and apply the Generalized Algorithm.

In this case the set $Z_m(0)$ corresponding to W_m is the same as that for Algorithm 1, however, in Algorithm 2, α_1 and α_2 are treated as parameters whereas in Algorithm 1 they were constants. In Algorithm 2, Step 6 can be simplified since α_1 and α_2 are determined as the solution of two linear equations in two unknowns.

The next algorithm is an obvious extension of Algorithm 2.

Algorithm 3: Choose $W_m = \{\vec{x}_m, A\vec{x}_m, A^2\vec{x}_m, \ldots\}$, $\nu = 0$, $p(m) \leq q$, and apply the Generalized Algorithm.

Let $r = r(\vec{x}_m)$ be the dimension of the space spanned by the vectors contained in W_m of Algorithm 3. Clearly, $r \leq n$ and the corresponding set $Z_m(0)$ contains at most r-1 independent vectors. Recall that q is the number of independent non-zero vectors in $Z_m(0)$. In this case the vectors in $Z_m(0)$ are of the form $A^i\vec{x}_m - \frac{\vec{x}_m' A^i \vec{x}_m}{\vec{x}_m' \vec{x}_m} \vec{x}_m$ $(i=1,2,\ldots,q)$ and each of these vectors represent an error vector converging to zero as $\vec{x}_m$ converges to an eigenvector. Computationally this latter property could lead to difficulty since eventually these vectors could be dominated by round-off error and become dependent.

Algorithm 4: Choose $W_m = \{\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_n\}$ where $\vec{e}_i$ is the ith column of the n x n identity matrix for i=1,2,...,n. Also choose $\nu = 1$, $p(m) \leq n-1$, and apply the Generalized Algorithm.

Numerical examples illustrating Algorithms 2 and 4 are given in Chapter VII. In Chapter VI an error analysis of the Generalized Algorithm is presented and this analysis indicates that difficulties can be encountered in some cases. Techniques to resolve these difficulties are proposed and analyzed. In Chapter VII, examples are given to illustrate the ideas of Chapter VI.

IV. GENERALIZED METHODS WITH n-1 DIRECTION VECTORS

In the last chapter at each step, m, of the iterative procedure a change vector $\vec{g}$ composed of p(m) direction vectors selected from the set $Z_m(\nu)$ is determined by solving the system

$$P'QP\vec{\alpha} = -P'Q\vec{x} \qquad (4.1)$$

for $\vec{\alpha}$ by an escalator technique. The set $Z_m(\nu)$ is generated from a given set of vectors W_m and $\vec{g}$ is determined from (3.4) using the $\vec{\alpha}$ calculated as the solution of (4.1). The quantities in Equation (4.1), at the kth stage of the escalation with $k \leq p$ are defined by Equations (3.10) and (3.18). For k=p, the change vector $\vec{g}$ can be written as $\vec{g} = P\vec{\alpha}$.

The objective of this chapter is to consider in detail the case when $Z_m(\nu)$ contains n-1 independent vectors and p(m) = n-1 because this is the maximum number of independent direction vectors that one could expect to find for any choice of the set W_m since there are only n-1 independent vectors orthogonal to $\vec{x}_m$. If the set $Z_m(\nu)$ is a set of n-1 independent vectors then the set $S = \{\vec{x}_m, \vec{z}_1, \vec{z}_2, \ldots, \vec{z}_{n-1}\}$ where $\vec{z}_i \varepsilon Z_m(\nu)$ (i=1,2,...,n-1) forms a basis for the space and so any other choice for the $\vec{z}$'s can be considered as a linear combination of the elements of S. This leads one to believe

that it might be possible to determine an optimum change vector $\vec{g}$ at the mth step independent of the choice of the set W_m and the corresponding set of $\vec{z}$'s defined by $Z_m(\nu)$. That this is indeed true will be shown in the remainder of this chapter. Suppose that $\vec{z}_i \varepsilon Z_m(\nu)$ $(i=1,2,\ldots,n-1)$ are normalized to unit length. If

$$P = (\vec{z}_1, \vec{z}_2, \ldots, \vec{z}_{n-1}) \tag{4.2}$$

then clearly P'P is the n-1 x n-1 identity matrix. To simplify the notation in the remainder of this chapter, let

$$s^2 = \vec{x}'\vec{x}. \tag{4.3}$$

The partioned matrix $(P,\vec{x}/s)$ is orthonormal, that is,

$$(P,\vec{x}/s)\begin{bmatrix} P' \\ \vec{x}'/s \end{bmatrix} = PP' + \vec{x}\vec{x}'/s^2 = I$$

and so

$$PP' = I - \vec{x}\vec{x}'/s^2. \tag{4.4}$$

Equation (4.1) can be written as $P'Q\vec{g} = -P'Q\vec{x}$ since $P\vec{\alpha} = \vec{g}$. This system with $\vec{g}$ unknown is underdetermined since it consists of n-1 equations in n unknowns. The additional independent restriction is $\vec{x}'\vec{g} = 0$. The resulting system is given by

$$P'Q\vec{g} = -P'Q\vec{x}$$

$$\vec{x}'\vec{g} = 0$$

or

$$\begin{bmatrix} P'Q \\ \vec{x}' \end{bmatrix}\vec{g} = \begin{bmatrix} -P'Q\vec{x} \\ 0 \end{bmatrix}. \tag{4.5}$$

Rewriting the matrix on the left in (4.5) as

$$\begin{bmatrix} P'Q \\ \vec{x}' \end{bmatrix} = \begin{bmatrix} P' \\ \vec{x}'/s \end{bmatrix} Q - \begin{bmatrix} 0 \\ (\vec{x}'/s)(Q-sI) \end{bmatrix} \tag{4.6}$$

and multiplying Equation (4.5) on the left by $(P,\vec{x}/s)$ gives

$$(P,\vec{x}/s)\left[\begin{pmatrix} P' \\ \vec{x}'/s \end{pmatrix} Q - \begin{pmatrix} 0 \\ (\vec{x}'/s)(Q-sI) \end{pmatrix}\right] = (P,\vec{x}/s)\begin{bmatrix} -P'Q\vec{x} \\ 0 \end{bmatrix} \tag{4.7}$$

$$= -PP'Q\vec{x}.$$

Performing the indicated multiplication in (4.7) and using (4.4), one obtains

$$[Q - (\vec{x}\vec{x}'/s^2)(Q-sI)]\vec{g} = (\vec{x}\vec{x}'/s^2-I)Q\vec{x} \tag{4.8}$$

which is independent of the $\vec{z}$'s and thus independent of the choice of W_m. If the matrix

$$H \equiv Q - (\vec{x}\vec{x}'/s^2)(Q-sI) \tag{4.9}$$

is non-singular, then

$$\vec{g} = H^{-1}(\vec{x}\vec{x}'/s^2-I)Q\vec{x}. \tag{4.10}$$

Theorem 4.1: The matrix H defined by (4.9) is non-singular if $\vec{x} \neq \vec{0}$ is not an eigenvector of A.

Proof: A result from linear algebra, for example Bodewig (1959, page 42), states that if C is non-singular and $D = \vec{a}\vec{b}'$ then $\det(C + D) = \det(C)(1+\mathrm{tr}C^{-1}D)$. By Lemma 3.1, Q is non-singular under this hypothesis and hence

$$\begin{aligned} \det(H) &= \det(Q)[1 + \mathrm{tr}\, Q^{-1}(\vec{x}\vec{x}'/s^2)(sI-Q)] \\ &= \det(Q)[1 + (\vec{x}'Q^{-1}\vec{x})/s - (\vec{x}'QQ^{-1}\vec{x})/s^2] \\ &= \det(Q)\,(\vec{x}'Q^{-1}\vec{x})/s \neq 0. \end{aligned}$$

This last expression is not equal to zero follows from the fact that Q is positive definite.

To determine the inverse of the matrix H, the following lemma will be used.

Lemma 4.1: The Sherman-Morrison and Bartlett Formula.

If C and C + D are non-singular with $C^{-1} = R$ and $D = \vec{a}\vec{b}'$, then $(C + D)^{-1} = R - \gamma RDR$ where $\gamma = 1/(1 + trRD)$.

For the proof of this lemma see Bodewig (1959, page 38). Using Lemma 4.1 it follows that

$$H^{-1} = Q^{-1} - \gamma Q^{-1}(\vec{x}\vec{x}'/s^2)(sI - Q)Q^{-1} \qquad (4.11)$$

where

$$\gamma^{-1} = 1 + tr[Q^{-1}(\vec{x}\vec{x}'/s^2)(sI - Q)]$$
$$= (\vec{x}'Q^{-1}\vec{x})/s. \qquad (4.12)$$

Therefore, using (4.12) in (4.11)

$$H^{-1} = Q^{-1} - \frac{s}{\vec{x}'Q^{-1}\vec{x}} Q^{-1} \frac{\vec{x}\vec{x}'}{s^2} (sI - Q)Q^{-1}. \qquad (4.13)$$

Substituting (4.13) into (4.10) and using (4.3) the solution to the system (4.5) is found to be

$$\vec{g} = \frac{\vec{x}'\vec{x}}{\vec{x}'Q^{-1}\vec{x}} Q^{-1}\vec{x} - \vec{x}. \qquad (4.14)$$

Equation (4.14) with $\vec{x} = \vec{x}_m$ determines $\vec{g}_m$ at the mth step of the iteration. Thus the following special algorithm for p(m) = n-1 direction vectors can be defined.

Algorithm 5: Generalized method with p(m) = n-1.

Step 1 through Step 5 and Step 8 through Step 10 are identical to the Generalized Algorithm.

Step 6. Calculate Q_m from (3.10) and determine Q_m^{-1}.

Step 7. Calculate $\vec{g}_m$ from (4.14).

This author has found this algorithm to be unsatisfactory in practice especially when convergence to an eigenvector is slow. This usually occurs for eigenvectors corresponding to close eigenvalues. The chief cause of difficulty is that Q_m approaches singularity as $\vec{x}_m$ approaches an eigenvector. This follows since if

$$\vec{x}_m = \vec{u} + \varepsilon\vec{v} \tag{4.15}$$

where $||\vec{v}|| = 1$ and $\vec{u}$ is an exact eigenvector of A with corresponding eigenvalue λ, then from (3.10)

$$Q = (\vec{u}'\vec{u})(A-\lambda I)^2 + 2\varepsilon(\vec{u}'\vec{v})(A-\lambda I)^2 + \varepsilon^2[A^2-2(\vec{v}'A\vec{v})A + (\vec{v}'A^2\vec{v})I]. \tag{4.16}$$

If it is assumed that A is scaled so that all of its eigenvalues have modulus less than or equal to one, then (4.16) implies that on a p decimal place computer Q will be singular if ε is zero to at least p/2 decimal places. To see this note that in this case the term in brackets in Equation (4.16) will not be seen by the computer but the computer actually sees Q as $(A-\lambda I)^2[(\vec{u}'\vec{u}) + 2\varepsilon(\vec{u}'\vec{v})]$ which is singular since $\det(A-\lambda I) = 0$. This means that the maximum

expected accuracy of $\vec{x}_m$ with $||\vec{x}_m|| = ||\vec{u}|| = 1$, is to p/2 decimal places if $\vec{g}_m$ is determined from Equation (4.14). For this reason Algorithm 5 is not suited for computational purposes.

V. CHOOSING THE STARTING VECTORS

Theoretically, the starting vector $\vec{x}_0$ can be chosen arbitrarily and the preceding algorithms will converge to an eigenvector. However, after the first eigenvector is determined, it is desirable to try to choose another starting value for $\vec{x}_0$ which will converge to a different eigenvector. For Algorithms 1 and 2 it is easily shown that if the starting vector, $\vec{x}_0$, is chosen orthogonal to the subspace spanned by the previously found eigenvectors then all the iterates $\vec{x}_m$, for m=1,2,3,... theoretically remain orthogonal to that subspace. This is proved with the aid of the following theorem:

Theorem 5.1: If $\vec{u}_1,\vec{u}_2,\ldots,\vec{u}_k$ are each distinct real eigenvectors of the n x n real symmetric matrix A, and if $\vec{y}$ is an arbitrary but nonzero real vector such that $\vec{u}_i\vec{y} = 0$ for i=1,2,...,k, then the vector $\vec{g} = \alpha_1(A^2\vec{y} - \frac{\vec{y}'A^2\vec{y}}{\vec{y}'\vec{y}}\vec{y}) + \alpha_2(A\vec{y} - \frac{\vec{y}'A\vec{y}}{\vec{y}'\vec{y}}\vec{y})$ is orthogonal to each of the vectors $\vec{u}_1,\vec{u}_2,\ldots,\vec{u}_k$ for arbitrary real scalars, α_1 and α_2.

Proof: $$\vec{u}_i'\vec{g} = \alpha_1(\vec{u}_i'A^2\vec{y} - \frac{\vec{y}'A^2\vec{y}}{\vec{y}'\vec{y}}\vec{u}_i'\vec{y}) + \alpha_2(\vec{u}_i'A\vec{y} - \frac{\vec{y}'A\vec{y}}{\vec{y}'\vec{y}}\vec{u}_i'\vec{y})$$

$$= \alpha_1(\lambda_i^2\vec{u}_i'\vec{y} - \frac{\vec{y}'A^2\vec{y}}{\vec{y}'\vec{y}}\vec{u}_i'\vec{y}) + \alpha_2(\lambda_i\vec{u}_i'\vec{y} - \frac{\vec{y}'A\vec{y}}{\vec{y}'\vec{y}}\vec{u}_i\vec{y})$$

$$= 0$$

since $\vec{u}_i'\vec{y} = 0$ for i=1,2,...,k by hypothesis. The number λ_i is the eigenvalue associated with $\vec{u}_i$.

Now observe that if the starting vector $\vec{x}_0$ has the properties of the vector $\vec{y}$ in Theorem 5.1 then the vector $\vec{x}_1 = \vec{x}_0 + t_o\vec{g}_o$ has these properties. The same is true for $\vec{x}_2, \vec{x}_3$, etc. so that each of the iterates $\vec{x}_m$, m=1,2,3,... remain orthogonal to the previously found eigenvectors $\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_k$.

The change vectors given in Algorithms 1 and 2 are special cases of the $\vec{g}$ defined in Theorem 5.1 and so all n eigenvectors of A can be determined by choosing n starting vectors satisfying the conditions of the theorem.

Clearly Theorem 5.1 can be generalized to the case when

$$\vec{g} = \sum_{i=1}^{p} \alpha_i \left(A^i\vec{y} - \frac{\vec{y}'A^i\vec{y}}{\vec{y}'\vec{y}}\,\vec{y}\right)$$

where α_i (i=1,2,...,p) are arbitrary. Thus the preceding result also holds for Algorithm 3.

A similar result also holds for Algorithm 5 since if $\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_k$ are distinct eigenvectors and if

$$\vec{u}_i'\vec{y} = 0 \quad (i=1,2,\ldots,k) \tag{5.1}$$

for arbitrary non-zero $\vec{y}$ not equal to an eigenvector, then from Equation (4.14)

$$\vec{u}_i'\vec{g} = \frac{\vec{y}'\vec{y}}{\vec{y}'Q^{-1}\vec{y}}\,\vec{u}_i'Q^{-1}\vec{y} - \vec{u}_i'\vec{y}$$

$$= \frac{\vec{y}'\vec{y}}{\vec{y}'Q^{-1}\vec{y}}\,\vec{u}_i'[(\vec{y}'\vec{y})A^2 - 2(\vec{y}'A\vec{y})A + (\vec{y}'A^2\vec{y})I]^{-1}\vec{y}$$

$$= \frac{\vec{y}'\vec{y}}{\vec{y}'Q^{-1}\vec{y}}\;\frac{\vec{u}_i'\vec{y}}{[(\vec{y}'\vec{y})\lambda_i^2 - 2(\vec{y}'A\vec{y})\lambda_i + (\vec{y}'A^2\vec{y})]}$$

$$= \frac{(\vec{y}'\vec{y})(\vec{u}_i'\vec{y})}{(\vec{y}'Q^{-1}\vec{y})[\vec{y}(A-\lambda_i I)^2\vec{y}]}$$

$$= 0 \qquad (i=1,2,\ldots,k). \qquad (5.2)$$

If $p(m) = n-1$ in Algorithm 4, then the analysis in Chapter IV indicates that Algorithm 4 and Algorithm 5 are equivalent. By (5.2), Algorithm 5 has the property that if $\vec{x}_0$ satisfies (5.1) with $\vec{y}$ replaced by $\vec{x}_0$ then each of the iterates $\vec{x}_m (m=0,1,2,\ldots)$ remain orthogonal to the previously found eigenvectors $\vec{u}_i (i=1,2,\ldots,k)$ and it follows that Algorithm 4 with $p(m) = n-1$ also has this property. It should be emphasized here that these two algorithms are theoretically identical but computationally they are not.

The choice of the starting vector appears to have a lot of influence on which eigenvector of A the norm reduction methods will converge. In most cases the norm reduction algorithms converge to that eigenvector of A in which the starting vector has its largest component although this is not universally true.

To facilitate choosing a vector $\vec{x}_0$ orthogonal to each of the previously found eigenvectors observe that $\vec{u}_i^{\,!}(\vec{v} - c_1\vec{u}_1 - c_2\vec{u}_2 - \ldots - c_k\vec{u}_k) = 0$ for i=1,2,...,k and arbitrary non-zero $\vec{v}$ if the c's are chosen by

$c_i = \dfrac{\vec{u}_i^{\,!}\vec{v}}{\vec{u}_i^{\,!}\vec{u}_i}$ (i=1,2,...,k). Let $\vec{w}=\vec{v}-\sum_{i=1}^{k}(c_i\vec{u}_i)=\vec{v}-\sum_{i=1}^{k}[\vec{u}_i(\dfrac{\vec{u}_i^{\,!}\vec{v}}{\vec{u}_i^{\,!}\vec{u}_i})]$

so that

$$\vec{w} = [I - \sum_{i=1}^{k} \frac{\vec{u}_i\vec{u}_i^{\,!}}{\vec{u}_i^{\,!}\vec{u}_i}]\vec{v}. \qquad (5.3)$$

If the arbitrary vector $\vec{v}$ is chosen successively as $\vec{e}_1,\vec{e}_2,\ldots,\vec{e}_n$ where $\vec{e}_j$ is the jth column of the identity matrix, j=1,2,...,n, the corresponding set of $\vec{w}$ vectors, $\vec{w}_j$, j=1,2,...,n, will contain n-k independent vectors each orthogonal to $\vec{u}_i$, i=1,2,...,k. Any one of these $\vec{w}$ vectors which is not zero is a suitable choice for $\vec{x}_0$.

There is no loss of generality in assuming that the previously found eigenvectors $\vec{u}_i$ are normalized so that the matrix in brackets in Equation (5.3) can be written

$$[I - \sum_{i=1}^{k} \vec{u}_i\vec{u}_i^{\,!}] \qquad (5.4)$$

since $\vec{u}_i^{\,!}\vec{u}_i = 1$, i=1,2,...,k. It is suggested that the matrix (5.4) be stored during the computation of all the eigenvectors. When each new one, say $\vec{u}_{k+1}$, is found and normalized, the simple product $\vec{u}_{k+1}\vec{u}_{k+1}^{\,!}$ is subtracted from the matrix given by (5.4) to update it. Since the trace of the matrix (5.4) is n-k, there must be some elements on its

diagonal which are greater than $\frac{n-k}{n}$ so that each diagonal element is tested in turn and the first one which is larger than $\frac{n-k}{n}$ determines the column of the matrix which is used for the new starting vector $\vec{x}_0$. This procedure assures that the starting vector is not approximately equal to the zero vector.

VI. ERROR ANALYSIS AND CONVERGENCE BEHAVIOR

The tolerance, ε, on $||R_m||$ should be chosen carefully since it determines bounds for the accuracy in the resulting eigenvalue and eigenvector. To obtain these bounds, let

$$\vec{r}_m = A\vec{x}_m - \lambda_m\vec{x}_m = \vec{v}_m - \lambda_m\vec{x}_m \tag{6.1}$$

where

$$\lambda_m = \frac{\vec{x}_m' A\vec{x}_m}{\vec{x}_m'\vec{x}_m} \tag{6.2}$$

and $\vec{v}_m = A\vec{x}_m$. Then $\vec{r}_m'\vec{r}_m = \{(\vec{v}_m'\vec{v}_m)(\vec{x}_m'\vec{x}_m) - (\vec{x}_m'\vec{v}_m)^2\}/(\vec{x}_m'\vec{x}_m)$ and so by (2.5),

$$||\vec{r}_m|| = \frac{\sqrt{2}}{2}\frac{||R_m||}{||\vec{x}_m||}\,. \tag{6.3}$$

Wilkinson (1961) shows that if $||\vec{x}_m|| = 1$, then there must be an eigenvalue λ of A such that

$$|\lambda - \lambda_m| \leq ||\vec{r}_m||. \tag{6.4}$$

If m is large enough so that $\varepsilon_m \equiv ||R_m|| < \varepsilon$ then in view of (6.3), $||\vec{r}_m|| < \varepsilon$ since $\sqrt{2}/2 < 1$ and $||\vec{x}_m|| \geq 1$.

If it is also known that for any other eigenvalue $\mu \neq \lambda$ that $|\mu - \lambda_m| \geq \delta$ where δ is appreciably greater than $\frac{\varepsilon_m\sqrt{2}}{2}$, then Wilkinson shows how to improve the error bound (6.4) to obtain

$$|\lambda - \lambda_m| \leq \frac{\varepsilon_m^2}{2\delta}/\left(1 - \frac{\varepsilon_m^2}{2\delta^2}\right). \tag{6.5}$$

If $\vec{u}$ is the exact eigenvector corresponding to the eigenvalue λ, then a bound for the error in $\vec{x}_m$ is given by

$$||\vec{x}_m - \vec{u}|| \leq \left(\frac{\varepsilon_m^4}{4\delta^4} + \frac{\varepsilon_m^2}{2\delta^2}\right)^{\frac{1}{2}} . \qquad (6.6)$$

Note that if $\delta >> \varepsilon_m$, then (6.5) and (6.6) give reasonably good error bounds for the eigenvalue and eigenvector, respectively, but as δ diminishes, this bound becomes poorer until when δ is of the order of magnitude of $\frac{\sqrt{2}\varepsilon_m}{2}$, (6.5) is no stronger than (6.4) and (6.6) no longer gives a useful result. This means that close eigenvalues in a symmetric matrix cause the determination of the corresponding eigenvectors to be ill-conditioned. However, it can be shown that coincident eigenvalues cause no difficulty in the determination of the corresponding eigenvectors.

If $\vec{x}_m = \sum_{i=1}^{n} a_i \vec{u}_i$ where $\vec{u}_i (i=1,2,\ldots,n)$ are an orthonormal set of eigenvectors of A corresponding to the real eigenvalues $\lambda_i (i=1,2,\ldots,n)$, then

$$|a_i| \leq \frac{\varepsilon_m \sqrt{2}}{2|\lambda_i - \lambda_m|} \qquad (i=1,2,\ldots,n; \lambda_i \neq \lambda) . \qquad (6.7)$$

where λ_m is the approximation to λ given by (6.2). The larger the value of $|\lambda_i - \lambda_m|$, the smaller is the bound for the component of the error in the direction of $\vec{u}_i$. For example if λ_1 and λ_2 are very close as compared to the remaining eigenvalues and if $\vec{x}_m$ is an approximation to $\vec{u}_1$, then the bound on $|a_2|$ will be large and the bound on $|a_j| = 0(\varepsilon_m)$ for $j=3,4,\ldots,n$. This means that eventually $\vec{x}_m$ is in the subspace spanned by $\vec{u}_1$ and $\vec{u}_2$ and that this subspace

is determined to $0(\varepsilon_m)$.

An example of Wilkinson's illustrating the preceding but modified to the present situation, is the matrix $A = \begin{bmatrix} a & \theta \\ \theta & a \end{bmatrix}$ where θ is small. Suppose that $\lambda = a$ and $\vec{x}' = (1,0)$ are taken as approximate eigenvalue and eigenvector, respectively, for A. Then $||R|| = \sqrt{2}\theta$. So if the tolerance, ε, on $||R||$ happens to be chosen greater than $\sqrt{2}\theta$, then $\vec{x}$ is taken as an eigenvector when actually (1,1) and (1,-1) corresponding to eigenvalues of $a+\theta$ and $a-\theta$, respectively, are the true eigenvectors. In this case, $\vec{x}$ in no sense approximates an eigenvector direction even though $||R||$ can be made arbitrarily small by choosing θ small enough. However, note that (6.4) implies that the eigenvalue approximation is always accurate for small $||R||$.

Wilkinson also mentions that the accuracy of the eigenvalue is twice that of the eigenvector when the eigenvalue is calculated by (6.2). It will now be shown that for close eigenvalues even better results can be obtained. Defining

$$\beta \equiv \frac{\varepsilon_m}{\sqrt{2}\delta}\left(1 + \frac{\varepsilon_m^2}{2\delta^2} + \frac{\varepsilon_m^4}{4\delta^4} + \ldots\right)^{\frac{1}{2}}$$

then Equation (6.5) can be rewritten as

$$|\lambda - \lambda_m| \leq \delta\beta^2 \tag{6.8}$$

and Equation (6.6) can be written as

$$||\vec{x}_m - \vec{u}|| \leq \frac{\varepsilon_m}{\sqrt{2}\delta}\left(1 + \frac{\varepsilon_m^2}{2\delta^2}\right)^{\frac{1}{2}} \leq \beta.$$

Thus if $\varepsilon_m < \sqrt{2}\delta$ and if the bound on $||\vec{x}_m - \vec{u}||$ is β, then a bound on $|\lambda - \lambda_m|$ is $\delta\beta^2$. So for $\delta<1$ the accuracy of the eigenvalue is more than twice that of the eigenvector.

Definition 6.1: The approximation $\vec{x}_m$ to the eigenvector $\vec{u}_j$ where $||\vec{x}_m|| = ||\vec{u}_j|| = 1$ is said to be accurate to τ decimal places if $||\vec{x}_m - \vec{u}_j|| \leq 10^{-\tau}$.

Suppose $\vec{x}_m = \sum_{i=1}^{n} a_i\vec{u}_i$ where $\vec{u}_i$ $(i=1,2,\ldots,n)$ are an orthonormal set of eigenvectors of A, then there is no loss of generality in assuming a_j to be positive, for if $a_j < 0$ then one may replace $\vec{x}_m$ by $-\vec{x}_m$ and all the previous results still hold. Thus if

$$||\vec{x}_m - \vec{u}_j||^2 = a_1^2 + a_2^2 + \ldots + (a_j-1)^2 + \ldots + a_n^2 \leq 10^{-2\tau}$$

then it follows that

$$|a_i| \leq 10^{-\tau} \quad (i\neq j;\ i=1,2,\ldots,n) \text{ and } |a_j-1| \leq 10^{-\tau}. \qquad (6.9)$$

The result given by (6.9) implies that each component of $\vec{x}_m$ is accurate to at least τ decimal places. If the error in $\vec{x}_m$ is known to be distributed equally among the components of $\vec{x}_m$, then the bound given in Definition 6.1 can be weakened to $||\vec{x}_m - \vec{u}_j|| \leq \sqrt{n}\ 10^{-\tau}$ with $\vec{x}_m$ still having τ decimal places of accuracy in each of its components.

For example, if $\vec{x}_m = a_1\vec{u}_1 + a_2\vec{u}_2$ is a unit vector with the component of error in the direction of $\vec{u}_j$ $(j=3,4,\ldots,n)$ negligible then from (6.2),

$$\lambda_m = a_1^2\lambda_1 + a_2^2\lambda_2 = \lambda_1[a_1^2 + a_2^2 \frac{\lambda_2}{\lambda_1}] \ . \qquad (6.10)$$

Suppose $\lambda_1 = 162.0000162$ and $\lambda_2 = 161.9999838$ and that $a_2 = .05$ and $a_1^2 = 1 - a_2^2$. These eigenvalues occur in an example given in the next chapter. By Definition 6.1 this means that $\vec{x}_m$ is an accurate approximation to $\vec{u}_1$ to less than two decimal places since it can be shown that $||\vec{x}_m - \vec{u}_1|| \doteq .05$. Now since $\lambda_1 - \lambda_2 = 324 \times 10^{-7}$ and from (6.10),

$$\lambda_m = \lambda_1[a_1^2 + a_2^2(1 - \frac{324 \times 10^{-7}}{\lambda_1})] \qquad (6.11)$$

$$\doteq \lambda_1 \ (1 - .0025(2) \times 10^{-7})$$

$$= \lambda_1 \ (1 - 5 \times 10^{-10}).$$

From (6.11), λ_m is an accurate approximation to λ_1 with an error of at most 5 in the 10th digit.

From the preceding analysis, if possible, ε should be chosen small enough so that even in the case of close eigenvalues accurate results can be obtained. If $\varepsilon_m << \delta$, then there are no problems in the determination of the eigenvalues and eigenvectors. In choosing the tolerance ε, it is important to realize that the round-off error in any of the proposed algorithms is not cumulative but depends only on each iteration separately.

The norm of the residual matrix is essentially obtained free at every step since it is composed of quantities that are

needed at the next iteration. However, when $\vec{x}_m$ becomes a good approximation to an eigenvector of A, then the subtraction $(\vec{v}_m'\vec{v}_m)(\vec{x}_m'\vec{x}_m)-(\vec{x}_m'\vec{v}_m)^2$ can yield erroneous results since all the significant figures could be subtracted out. So for good eigenvector approximations $||R_m||$ should be calculated by

$$||R_m||^2 = 2 \sum_{i=1}^{n-1} \sum_{j=i}^{n} r_{ij}^{(m)2} , \qquad (6.12)$$

where $r_{ij}^{(m)}$ is the ith row, jth column element of the mth residual matrix R_m.

For close eigenvalues it is possible to show that if ε is not chosen sufficiently small, then for some m, $||R_m||$ might be less than ε but the eigenvectors corresponding to the close eigenvalues might not be accurate to any decimal places. In order to show this, the following assumptions are made and they will be considered valid for the remainder of this chapter. Suppose

$$|\lambda_1 - \lambda_2| = \delta \qquad (6.13)$$

and that the separation of all the other eigenvalues is much greater than δ. Also assume that A is scaled so that if λ is an eigenvalue of A, then $|\lambda| \leq 1$. Suppose $\lambda_1 \neq 0$ and let

$$\frac{|\lambda_1 - \lambda_2|}{|\lambda_1|} = \delta_r \qquad (6.14)$$

be the relative separation of λ_1 and λ_2. If $\delta_r = 0(10^{-\tau})$, then λ_1 and λ_2 are identical to τ decimal places. Let $\vec{x}_m$ be an approximation to $\vec{u}_1$ such that $||\vec{x}_m|| = 1$. If

$$\vec{x}_m = \sum_{i=1}^{n} a_i \vec{u}_i \qquad (6.15)$$

where $\vec{u}_i$ (i=1,2,...,n) are an orthonormal set of eigenvectors of A corresponding to eigenvalues λ_i (i=1,2,...,n), it follows from (2.5) after some algebraic manipulation that

$$\frac{||R_m||^2}{2} = (\vec{x}_m'\vec{x}_m)(\vec{x}_m'A^2\vec{x}_m) - (\vec{x}_m'A\vec{x}_m)^2 \qquad (6.16)$$

$$= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} a_i^2 a_j^2 (\lambda_i - \lambda_j)^2.$$

Suppose that the subspace spanned by the ill-determined eigenvectors $\vec{u}_1$ and $\vec{u}_2$ is determined to $0(\varepsilon)$, that is, $a_j = 0(\varepsilon)$ (j=3,4,...,n). By (6.7), this is true for m sufficiently large. Then from (6.16),

$$\frac{||R_m||^2}{2} = a_1^2 a_2^2 (\lambda_1 - \lambda_2)^2 + 0(\varepsilon^2) \qquad (6.17)$$

$$= a_1^2 a_2^2 \lambda_1^2 \delta_r^2 + 0(\varepsilon^2).$$

If $\delta_r \leq \varepsilon$, then (6.17) implies that $||R_m||$ corresponding to $\vec{x}_m$ could be less than or equal to ε with $|a_2|$ large. That is, the convergence criteria might be satisfied with $\vec{u}_1$ and $\vec{u}_2$ indistinguishable and $\vec{x}_m$ any vector in the subspace spanned by $\vec{u}_1$ and $\vec{u}_2$. In terms of the number of decimal places, this means that if the close eigenvalues are identical to τ decimal places and the subspace spanned by the corresponding eigenvectors is determined to τ decimal places, then $||R_m||$ could be zero to τ decimal places with $\vec{x}_m$ an arbitrary

linear combination of $\vec{u}_1$ and $\vec{u}_2$. However, if ε is taken sufficiently smaller than δ_r, then $\vec{u}_1$ and $\vec{u}_2$ are distinguishable in terms of $||R_m||$ since the convergence criteria cannot be satisfied unless $|a_2|$ is also sufficiently small. If the computer has p decimal places and if the subspace is determined to $0(10^{-p})$, then $||R_m||$ is zero at best to $0(10^{-p})$. Now if $\delta_r = 0(10^{-\tau})$ then $|a_2|$ can be $0(10^{\tau-p})$ with $||R_m|| = 0(10^{-p})$ and therefore the maximum expected accuracy of $\vec{u}_1$ is to $p - \tau$ decimal places.

The preceding ill-conditioning is even more serious in terms of the polynomial $F_m(t)$ given by (2.15). Since $F_m(t) = c_1 t + c_2 t^2 + c_3 t^3 + c_4 t^4$, it follows that $F_m(t)$ is ill-determined if any of the c's are ill-determined.

If

$$\vec{g}_m = \sum_{i=1}^{n} b_i \vec{u}_i \tag{6.18}$$

is an arbitrary direction vector satisfying $||\vec{g}_m|| = 1$, then

$$A\vec{g}_m = \sum_{i=1}^{n} b_i \lambda_i \vec{u}_i. \tag{6.19}$$

Also from (6.15),

$$A\vec{x}_m = \sum_{i=1}^{n} a_i \lambda_i \vec{u}_i. \tag{6.20}$$

Under the preceding assumptions, if one uses (6.15), (6.18), (6.19) and (6.20) in (2.6) and combines all terms of order ε^2 into $0(\varepsilon^2)$, the result is

$$\frac{c_1}{2} = a_1 a_2 (a_1 b_2 + a_2 b_1)(\lambda_1 - \lambda_2)^2 + \sum_{j=3}^{n} b_j \varepsilon_j [a_1^2 (\lambda_1 - \lambda_j)^2] + a_2^2 (\lambda_2 - \lambda_j)^2] + 0(\varepsilon^2) \tag{6.21}$$

where ε_j is used in place of a_j $(j=3,4,\ldots,n)$ to emphasize that $a_j = 0(\varepsilon)$ $(j=3,4,\ldots,n)$. Using (6.14), Equation (6.21) can be rewritten as

$$\frac{c_1}{2} = a_1a_2(a_1b_2+a_2b_1)\lambda_1^2\delta_r^2 + \sum_{j=3}^{n} b_j\varepsilon_j[a_1^2(\lambda_1-\lambda_j)^2+a_2^2(\lambda_2-\lambda_j)^2]+0(\varepsilon^2). \tag{6.22}$$

Thus if $\delta_r^2 \leq \varepsilon$ then c_1 might not be well-determined which could conceivably cause the entire iteration procedure to break down. Again in terms of the number of decimal places, this means that if the close eigenvalues are identical to τ decimal places then the subspace spanned by the corresponding eigenvectors must be determined to more than 2τ decimal places if the iterative procedure is to be well defined and thus able to distinguish $\vec{u}_1$ and $\vec{u}_2$ and eventually meet the convergence criteria. Assuming that this is the case, then

$$||\vec{x}_m - \vec{u}_1||^2 = (a_1-1)^2 + a_3^2 + \ldots + a_n^2 \tag{6.23}$$

$$= 2(1 - a_1)$$

since $||\vec{x}_m|| = 1$. Let

$$\eta = \frac{||R_m||^4}{4\delta^4} + \frac{||R_m||^2}{2\delta^2}, \tag{6.24}$$

then from (6.6),

$$2(1 - a_1) \leq \eta . \tag{6.25}$$

Again, there is no loss of generality in assuming a_1 to be positive. Hence from (6.25),

$$a_1 \geq 1 - \frac{\eta}{2} . \qquad (6.26)$$

Since $||\vec{x}_m|| = 1$, it follows from (6.26) that

$$1 = \sum_{i=1}^{n} a_i^2 \geq a_1^2 + a_2^2 \geq (1 - \frac{\eta}{2})^2 + a_2^2$$

and from this it follows that

$$a_2^2 \leq 1 - (1 - \frac{\eta}{2})^2 = \eta(1 - \frac{\eta}{4}) \leq \eta. \qquad (6.27)$$

Equations (6.26) and (6.27) give bounds on a_1 and a_2 as they approach 1 and 0 respectively as $||R_m||$ approaches zero.

Now reconsider Equation (6.22) in view of (6.27) with the assumption that $\varepsilon < \delta_r^2$. Clearly, c_1 is well-determined until $a_2\delta_r^2 = 0(\varepsilon)$. At this point, $a_2 = 0(\varepsilon/\delta_r^2)$ and (6.27) implies that η might be as small as $0(\varepsilon^2/\delta_r^4)$ and still not violate (6.27). Then since $\delta^2 = \lambda_1^2\delta_r^2$, it follows from (6.24) that $||R_m|| = 0(\varepsilon/\delta_r)$. On a p decimal place computer, if $|a_j| = 0(10^{-p})$ $(j=3,4,\ldots,n)$ and if $\delta_r = 0(10^{-\tau})$ then the preceding disucssion implies that the iterative procedure can only be expected to reduce $||R_m||$ to $0(10^{\tau-p})$. Clearly, $\vec{u}_1$ and $\vec{u}_2$ can be separated by the iterative procedure only if $\tau < p/2$. Therefore, if $\varepsilon < 0(10^{\tau-p})$, or if $\tau \geq p/2$ then in order to reduce the component of error in the direction of $\vec{u}_2$ so that $||R_m|| < \varepsilon$ is satisfied it is necessary to resort to a different technique. An alternative technique will be developed and discussed later in this chapter.

It is not necessary to determine if the remaining c's

can also be ill-determined since the fact that c_1 is ill-determined implies that $F_m(t)$ is ill-determined.

In Algorithm 2, the manner of calculation of $\vec{z}_{1_m}$ and $\vec{z}_{2_m}$ is critical when $\vec{x}_m$ is almost an eigenvector, since these vectors represent error vectors that are approaching zero. These should be calculated by

$$\vec{z}_{1_m} = (A - \lambda_m I)\vec{x}_m \tag{6.28}$$

and

$$\vec{z}_{2_m} = (A - \hat{\lambda}_m I)[(A + \hat{\lambda}_m I)\vec{x}_m] \tag{6.29}$$

where $\lambda_m = \dfrac{\vec{x}_m'\vec{v}_m}{\vec{x}_m'\vec{x}_m}$ and $\hat{\lambda}_m = \left(\dfrac{\vec{v}_m'\vec{v}_m}{\vec{x}_m'\vec{x}_m}\right)^{\frac{1}{2}}$. Most of the significant figures of accuracy may be subtracted out if (3.2) is used directly. It is also possible that $\vec{z}_{1_m}$ and $\vec{z}_{2_m}$ could be dependent and so a check should be made to prevent erroneous calculations.

The presence of close eigenvalues might make the determination of the direction vectors ill-conditioned in the sense of being able to separate the corresponding eigenvectors. For example, consider

$$\vec{z}_{1_m} = A\vec{x}_m - \frac{\vec{x}_m'A\vec{x}_m}{\vec{x}_m'\vec{x}_m}\vec{x}_m \tag{6.30}$$

with $\vec{x}_m$ given by (6.15) and $a_j = 0(\varepsilon)$ $(j=3,4,\ldots,n)$.

Then

$$\vec{z}_{1_m} = \sum_{j=1}^{n} a_j(\lambda_j - \frac{\sum_{i=1}^{n} a_i^2\lambda_i}{\sum_{i=1}^{n} a_i^2})\vec{u}_j = \sum_{j=1}^{n} a_j \frac{\sum_{i=1}^{n} a_i^2(\lambda_j-\lambda_i)}{\sum_{i=1}^{n} a_i^2}\vec{u}_j. \tag{6.31}$$

Using the fact that $||\vec{x}_m|| = 1$ and combining all terms of order ε^2 into $0(\varepsilon^2)$, then Equation (6.31) becomes

$$\vec{z}_{1_m} = a_1[a_2^2(\lambda_1-\lambda_2) + 0(\varepsilon^2)]\vec{u}_1 + a_2[a_1^2(\lambda_2-\lambda_1) + 0(\varepsilon^2)]\vec{u}_2$$

$$+ \sum_{j=3}^{n} a_j[a_1^2(\lambda_j-\lambda_1) + a_2^2(\lambda_j-\lambda_2) + 0(\varepsilon^2)]\vec{u}_j .$$

Since $1 = a_1^2 + a_2^2 + 0(\varepsilon^2)$ the preceding equation can be simplified to

$$\vec{z}_{1_m} = a_1a_2^2(\lambda_1-\lambda_2)\vec{u}_1 + a_1^2a_2(\lambda_2-\lambda_1)\vec{u}_2 + \sum_{j=3}^{n} a_j(\lambda_j-a_1^2\lambda_1-a_2^2\lambda_2)\vec{u}_j$$

$$+ 0(\varepsilon^2)\vec{s} \tag{6.32}$$

where $\vec{s}$ is a vector in the space spanned by the vectors $\vec{u}_i$ (i=1,2,...,n). Using (6.13) and (6.14) and combining all terms of order ε, Equation (6.32) becomes

$$\vec{z}_{1_m} = a_1a_2(a_2\vec{u}_1 - a_1\vec{u}_2)\delta + 0(\varepsilon)\vec{s}_1 \tag{6.33}$$

$$= a_1a_2(a_2\vec{u}_1 - a_1\vec{u}_2)\lambda_1\delta_r + 0(\varepsilon)\vec{s}_1$$

under the assumption that $\lambda_1>\lambda_2$. Here $\vec{s}_1$ is a vector in the space spanned by the vectors $\vec{u}_i$ (i=1,2,...,n). Thus if $\delta_r<\varepsilon$, then $\vec{z}_{1_m}$ is not a meaningful direction vector since it cannot distinguish $\vec{u}_1$ and $\vec{u}_2$. Clearly, $\vec{z}_{1_m}$ can separate

$\vec{u}_1$ and $\vec{u}_2$ when the subspace spanned by these vectors is more accurately determined than $0(\delta_r)$.

The preceding analysis shows that numerous difficulties are encountered in the determination of the eigenvectors corresponding to close eigenvalues. In practice all known techniques encounter similar difficulties for accurately determining ill-conditioned eigenvectors corresponding to close eigenvalues. Rosser, Lanczos, Hestenes, and Karush (1951) and Wilkinson (1961) discuss certain techniques which do give some slight improvements in accuracy.

A separation technique will now be developed for accurately determining eigenvectors corresponding to close eigenvalues. This technique requires accurate approximations to the eigenvalues which can be achieved if the corresponding calculated eigenvectors are only slightly separated as indicated in the example given by Equations (6.10) and (6.11). Equation (6.22) indicates that on a p decimal place computer accurate eigenvalues can be obtained directly by using the iterative procedure if $\tau<p/2$ where the closest eigenvalues are identical to τ decimal places. In this case $F_m(t)$ becomes ill-determined when the norm of the residual matrix is of $0(10^{\tau-p})$. However, if $\tau\geq p/2$, Equation (6.22) also indicates that $F_m(t)$ and thus the entire iterative procedure can become ill-determined before the ill-determined eigenvectors are accurate to any decimal places. In this case

the norm of the residual matrix is greater than $0(10^{-p/2})$ and the corresponding eigenvalue might not be accurate. If this occurs it is proposed that the following search technique be used to reduce $||R_m||$ until the ill-determined eigenvectors are sufficiently separated so as to obtain accurate approximations to the corresponding eigenvalues. Choose $\vec{g}_m$ as usual but update $\vec{x}_m$ by $\vec{x}_{m+1} = \vec{x}_m + t'_m\vec{g}_m$ where t'_m minimizes $||R_m||$ and is found by a search technique. This search technique may be a simple trial and error procedure or quite an elaborate procedure. Methods for finding the minimum of a function in a given direction are discussed in Wilde (1964) and Wilde and Beightler (1967).

Now suppose $\vec{u}_j^{(c)}$ $(j=1,2,\ldots,n)$ are normalized calculated approximations to the exact normalized eigenvectors $\vec{u}_j$ $(j=1,2,\ldots,n)$ corresponding to eigenvalues λ_j $(j=1,2,\ldots,n)$. Since the eigenvectors of the matrix A form a basis for the space, there exist c_{ji}'s such that

$$\vec{u}_j^{(c)} = \sum_{i=1}^{n} c_{ji}\vec{u}_i \quad (j=1,2,\ldots,n). \tag{6.34}$$

If there are no ill-determined eigenvectors, then for each $\vec{u}_j^{(c)}$ it follows from (6.7) that $c_{ji} = 0(\varepsilon)$ for $j\neq i$ and $i=1,2,\ldots,n$. However, suppose that two eigenvalues are close, say $|\lambda_1 - \lambda_2| = \delta$, with $\lambda_1 > \lambda_2$, and that the separation of all the other eigenvalues is much greater than δ. By Equation (6.7), $|c_{12}|$ and $|c_{21}|$ are not necessarily small

and so $\vec{u}_1^{(c)}$ and $\vec{u}_2^{(c)}$ are not necessarily good approximations to $\vec{u}_1$ and $\vec{u}_2$, however, they are contained in the subspace spanned by $\vec{u}_1$ and $\vec{u}_2$ to good approximation. Suppose that the norms of the residual matrices corresponding to $\vec{u}_1^{(c)}$ and $\vec{u}_2^{(c)}$ did not meet the convergence criteria when $F_m(t)$ became ill-determined and, if necessary, the preceding search technique was applied until $\vec{u}_1^{(c)}$ and $\vec{u}_2^{(c)}$ were sufficiently separated to obtain accurate eigenvalues. Eventhough these vectors did not meet the convergence criteria they can still be used to generate new starting vectors with the properties discussed in Chapter V. This follows since the subspace is well-determined.

Suppose that A is scaled as before. Let $\lambda_2^{(c)}$ be the calculated value of λ_2. Define k_2 by

$$\lambda_2 - \lambda_2^{(c)} = k_2. \tag{6.35}$$

Note that k_2 is small by the preceding assumptions.

Consider the vector

$$\vec{v}_1 = (A - \lambda_2^{(c)} I)\vec{u}_1^{(c)} = c_{11}(\lambda_1-\lambda_2 + k_2)\vec{u}_1 + c_{12}k_2\vec{u}_2 + \sum_{i=3}^{n} c_{1i}(\lambda_1-\lambda_2 + k_2)\vec{u}_i. \tag{6.36}$$

If k_2 is sufficiently small so that $c_{12}k_2 = 0(\varepsilon)$ then one would expect $\vec{v}_1$ to be a more accurate approximation to $\vec{u}_1$ than $\vec{u}_1^{(c)}$ is since $c_{1i} = 0(\varepsilon)$ $(i=3,4,\ldots,n)$. However, upon closer examination one sees that if $|\lambda_i-\lambda_2|$ $(i=3,4,\ldots,n)$ is large, then the error component of $\vec{v}_1$ in the direction of

$\vec{u}_i$ (i=3,4,...,n) could conceivably be large enough to prevent one from obtaining results to the maximum expected accuracy of the computer. This suggests reorthogenalizing $\vec{v}_1$ with respect to $\vec{u}_j^{(c)}$ (j=3,4,...,n) by forming $\vec{w}_1$ given by

$$\vec{w}_1 = \vec{v}_1 - \sum_{j=3}^{n} (\vec{v}_1,\vec{u}_j^{(c)})\vec{u}_j^{(c)} . \qquad (6.37)$$

In order to determine if $\vec{w}_1$ is a better approximation to $\vec{u}_1$ than either $\vec{v}_1$ or $\vec{u}_1^{(c)}$, it is necessary to substitute (6.34), (6.35), and (6.36) into (6.37) which gives

$$\vec{w}_1 = c_{11}(\lambda_1-\lambda_2+k_2)\vec{u}_1 + c_{12}k_2\vec{u}_2 + \sum_{i=3}^{n} c_{1i}(\lambda_i-\lambda_2+k_2)\vec{u}_i$$

$$- \sum_{j=3}^{n} \{[c_{j1}c_{11}(\lambda_1-\lambda_2+k_2) + c_{12}c_{j2}k_2$$

$$+ \sum_{i=3}^{n} c_{ji}c_{1i}(\lambda_i-\lambda_2+k_2)] \; [\sum_{i=1}^{n} c_{ji}\vec{u}_i]\}. \qquad (6.38)$$

The only c_{ji}'s in Equation (6.38) that are not of $0(\varepsilon)$ are c_{ii} (i=1,2,...,n), c_{12}, and c_{21}. The significant features of Equation (6.38) are clearer if each c_{ji} of order ε is written as ε_{ji} and if all terms of order ε^2 are combined and denoted by $0(\varepsilon^2)$. If Equation (6.14) is also used, then Equation (6.38) becomes

$$\vec{w}_1 = [c_{11}(\lambda_1-\lambda_2+k_2) + 0(\varepsilon^2)]\vec{u}_1 + [c_{12}k_2 + 0(\varepsilon^2)]\vec{u}_2$$

$$+ \sum_{j=3}^{n} [\varepsilon_{1j}(\lambda_j-\lambda_2+k_2) - c_{11}c_{jj}\varepsilon_{j1}(\lambda_1-\lambda_2+k_2)$$

$$- c_{jj}^2\varepsilon_{1j}(\lambda_j-\lambda_2+k_2) + 0(\varepsilon^2)]\vec{u}_j$$

$$= [c_{11}(\lambda_1\delta_r+k_2) + 0(\varepsilon^2)]\vec{u}_1 + [c_{12}k_2 + 0(\varepsilon^2)]\vec{u}_2$$

$$+ \sum_{j=3}^{n} [\varepsilon_{1j}(\lambda_j-\lambda_2+k_2)(1-c_{jj}^2) - c_{11}c_{jj}\varepsilon_{j1}(\lambda_1\delta_r+k_2)$$

$$+ 0(\varepsilon^2)]\vec{u}_j \; . \qquad (6.39)$$

Equation (6.39) can be simplified further by noting that $\vec{u}_j^{(c)}$ (j=3,4,...,n) are of unit length and that the component of error in the direction of $\vec{u}_i$ ($i\neq j$; i=1,2,...,n) is of $0(\varepsilon)$. Then for j=3,4,...,n,

$$1 = \vec{u}_j^{(c)} \cdot \vec{u}_j^{(c)} = \sum_{i=1}^{n} c_{ji}^2 = c_{jj}^2 + \sum_{\substack{i=1 \\ i\neq j}}^{n} c_{ji}^2 = c_{jj}^2 + \sum_{\substack{i=1 \\ i\neq j}}^{n} \varepsilon_{ji}^2$$

$$= c_{jj}^2 + 0(\varepsilon^2)$$

and so

$$1 - c_{jj}^2 = 0(\varepsilon^2) \quad (j=3,4,...,n). \qquad (6.40)$$

Using Equation (6.40) in Equation (6.39) one obtains

$$\vec{w}_1 = [c_{11}(\lambda_1\delta_r+k_2) + 0(\varepsilon^2)]\vec{u}_1 + [c_{12}k_2 + 0(\varepsilon^2)]\vec{u}_2$$

$$- \sum_{j=3}^{n} [c_{11}c_{jj}\varepsilon_{j1}(\lambda_1\delta_r+k_2) + 0(\varepsilon^2)]\vec{u}_j \ . \qquad (6.41)$$

Equation (6.41) implies that $\vec{w}_1$ is a more accurate approximation to $\vec{u}_1$ then either $\vec{u}_1^{(c)}$ or $\vec{v}_1$ if $\delta_r > |c_{12}k_2|$. This conclusion follows since the component of error in the direction $\vec{u}_2$ of $\vec{u}_1^{(c)}$ is c_{12} which is not necessarily small and since the components of error in the directions $\vec{u}_j$ $(j=3,4,\ldots,n)$ of $\vec{v}_1$ are not necessarily small. Thus if on a p decimal place machine, the two closest eigenvalues are identical to τ decimal places and the corresponding eigenvectors can be sufficiently separated so that $c_{12}k_2$ is zero to p decimal places, then $\vec{w}_1$ is an accurate approximation of $\vec{u}_1$ to $p-\tau$ decimal places. If $c_{12}k_2$ is not zero to p decimal places, then (6.2), (6.36), and (6.37) can be recalculated with $\vec{u}_1^{(c)}$ replaced by $\vec{w}_1$ until $c_{12}k_2$ is zero to p decimal places. Further improvement of the accuracy of $\vec{w}_1$ is impossible by this method or by any method on a p decimal place computer. Only with the addition of more decimal places can better accuracy be obtained.

The preceding technique can be generalized to r>2 close eigenvalues, say $\lambda_1,\lambda_2, \ldots, \lambda_r$. Then to get a better approximation to $\vec{u}_1$, one would form

$$\vec{v}_1 = (A-\lambda_2^{(c)}I)(A-\lambda_3^{(c)}I)\ldots (A-\lambda_r^{(c)}I)\vec{u}_1^{(c)} \qquad (6.42)$$

where $\lambda_2^{(c)}$, $\lambda_3^{(c)}$, ..., $\lambda_r^{(c)}$ are accurate calculated values of λ_2, λ_3, ..., λ_r. Then reorthogonalizing $\vec{v}_1$ with respect to $\vec{u}_j^{(c)}$ (j=r+1, r+2,...,n), one forms

$$\vec{w}_1 = \vec{v}_1 - \sum_{j=r+1}^{n} (\vec{v}_1, \vec{u}_j^{(c)}) \vec{u}_j^{(c)} \qquad (6.43)$$

which can be shown to be a better approximation to $\vec{u}_1$ then either $\vec{u}_1^{(c)}$ or $\vec{v}_1$. In a similar manner one can obtain $\vec{w}_2$, $\vec{w}_3$, ..., $\vec{w}_r$ which would be more accurate approximations to $\vec{u}_2$, $\vec{u}_3$, ..., $\vec{u}_r$ than $\vec{u}_2^{(c)}$, $\vec{u}_3^{(c)}$, ..., $\vec{u}_r^{(c)}$.

For computational purposes the results of this chapter can be summarized as follows. If ε is sufficiently small than the only ill-conditioning one needs to consider is when $F_m(t)$ becomes ill-determined due to close eigenvalues. It is easy to test when this occurs since the resulting t_m might yield a residual norm increase, or the rate of convergence might become zero, or c_4 might be less than zero which is theoretically impossible.

Suppose that approximations to all of the eigenvectors have been calculated and that the norm of the residual matrix corresponding to each eigenvector approximation is less than $0(10^{-p/2})$ on a p decimal place computer. If the convergence criterion is not satisfied for several eigenvector approximations due to close eigenvalues, then the separation technique can be applied directly. However, if the norm of the residual matrix corresponding to some of the approximate eigenvectors is greater than $0(10^{-p/2})$ when $F_m(t)$ becomes ill-

determined, then there are close eigenvalues identical to at least p/2 decimal places. In this case the search technique described previously is applied until these eigenvectors are determined accurately enough to use the separation routine.

VII. COMPARISON OF METHODS AND EXAMPLES

A. Jacobi Method Versus Norm Reduction Methods

In this section the Jacobi method will be compared with the norm reduction methods proposed in this thesis. Computationally on small matrices any of the algorithms developed in this thesis seem to compare quite favorably with the Jacobi method. However, certain matrices can be found so that one method is superior to the other.

Consider the 3 x 3 matrix given by

$$A_0 = \begin{bmatrix} e & a & b \\ a & f & d \\ b & d & g \end{bmatrix}. \tag{7.1}$$

In the standard Jacobi method the first similarity transformation on A_0 reduces the (2,1) and (1,2) elements to zero. To do this it is necessary to calculate

$$\alpha = \frac{2a}{e-f}, \quad c^2 = \frac{1}{2}\left(1 + \frac{1}{\sqrt{\alpha^2+1}}\right), \text{ and } s^2 = \frac{1}{2}\left(1 - \frac{1}{\sqrt{\alpha^2+1}}\right). \tag{7.2}$$

If $e = f$, then $c = \cos(\frac{\pi}{4})$ and $s = \sin(\frac{\pi}{4})$. The first similarity transformation matrix is then given by

$$Y_0 = \begin{bmatrix} c & s & 0 \\ -s & c & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{7.3}$$

and

$$A_1 = Y_0' A_0 Y_0 = \begin{bmatrix} c^2e + 2csa + s^2f & 0 & cb + sd \\ 0 & s^2e - 2csa + c^2f & -sb + cd \\ cb + sd & -sb + cd & g \end{bmatrix}. \tag{7.4}$$

In a similar manner the (1,3) and (3,1) elements of A_1 can be reduced to zero. By applying this technique iteratively to the non-zero off-diagonal elements of A_k ($k=0,1,2,\ldots$) eventually, at say $k = M$, a diagonal matrix will result with negligible off-diagonal terms with the eigenvectors of A_0 given by $Y_0 Y_1 \ldots Y_M$. This is proved in Forsythe and Henrici (1960).

In particular, if A_0 is given by

$$A_0 = \begin{bmatrix} 10000. & .00002 & .00002 \\ .00002 & .0 & -.00002 \\ .00002 & -.00002 & 20000. \end{bmatrix} \tag{7.5}$$

then by (7.2), $\alpha = .00004/10000. = .4 \times 10^{-8}$ and $\alpha^2 = .16 \times 10^{-16}$. So in forming the quantity $\alpha^2 + 1$ on a sixteen decimal place machine the result will be 1 to sixteen decimal places with a loss of two significant figures in α^2. In this case, by (7.2), $c^2 = 1$ and $s^2 = 0$ and by (7.3), $Y_0 = I$, the identity matrix. Thus the loss of the two significant figures in forming $\alpha^2 + 1$ has made it impossible to reduce the (1,2) and (2,1) elements of A_0 to zero on a sixteen place computer. Similar difficulties are encountered if one attempts to make any other off-diagonal element of A_0

zero. However, some of these difficulties can be eliminated by combining and rationalizing the expression defining s^2 given by Equation (7.2).

From the preceding discussion it can be concluded that the Jacobi method will fail completely or will encounter a loss of accuracy in matrices with elements of widely varying orders of magnitude. In general, on a p decimal place computer the Jacobi method will work satisfactorily only if the elements of the matrix vary by less than p/2 orders of magnitude.

The matrix given by (7.5) was tried using the norm reduction technique defined by Algorithm 2. The results which are summarized in the following table were calculated in double precision (16 decimal places) on an IBM 360/65 computer.

It can be verified directly that these eigenvectors and eigenvalues are accurate to sixteen decimal places.

On a p decimal place computer the Jacobi method can directly obtain the eigenvectors corresponding to eigenvalues that are identical to more than p/2 decimal places. However, it was shown in Chapter VI that the norm reduction methods cannot determine these eigenvectors directly.

The eigenvalues and eigenvectors of matrices as large as 15 x 15 were calculated by Algorithms 2 and 4. In all cases, the results were comparable to the results obtained using the Jacobi method.

Table 7.1. Algorithm 2 applied to A_0 given by (7.5)

$\vec{u}_1^{(c)}$	$\vec{u}_2^{(c)}$	$\vec{u}_3^{(c)}$
1.000000000000000 $.200000000400 \times 10^{-8}$ $-.199999999600 \times 10^{-8}$	$-.200000000200 \times 10^{-8}$ 1.000000000000000 $.100000000200 \times 10^{-8}$	$.199999999800 \times 10^{-8}$ $-.099999999800 \times 10^{-8}$ 1.000000000000000
$\lambda_1^{(c)}$	$\lambda_2^{(c)}$	$\lambda_3^{(c)}$
9999.999999999984	$-.600000000800 \times 10^{-13}$	20000.00000000000
$\|\|R_7\|\| = .60 \times 10^{-16}$	$\|\|R_1\|\| = .15 \times 10^{-16}$	$\|\|R_0\|\| = .27 \times 10^{-16}$

B. Examples Using Norm Reduction Methods

To illustrate the methods that have been developed, consider the matrix

$$A = \begin{bmatrix} 468+36\delta & 18-36\delta & 24+33\delta & 192-60\delta & 120+3\delta \\ 18-36\delta & 468+36\delta & -24-33\delta & -192+60\delta & -120-3\delta \\ 24+33\delta & -24-33\delta & 86-80\delta & -8+8\delta & 100-76\delta \\ 192-60\delta & -192+60\delta & -8+8\delta & 398+64\delta & 200+40\delta \\ 120+3\delta & -120-3\delta & 100-76\delta & 200+40\delta & 182-56\delta \end{bmatrix}. \tag{7.6}$$

The exact eigenvalues and eigenvectors of this matrix for arbitrary δ are given by

$$\lambda_1=486, \quad \lambda_2=162(1-\delta), \quad \lambda_3=162(1+\delta), \quad \lambda_4=-18, \quad \lambda_5=810; \tag{7.7}$$

$$\vec{u}_1=\frac{\sqrt{2}}{2}\begin{bmatrix}1\\1\\0\\0\\0\end{bmatrix}, \quad \vec{u}_2=\frac{1}{18}\begin{bmatrix}3\\-3\\-13\\4\\-11\end{bmatrix}, \quad \vec{u}_3=\frac{1}{6}\begin{bmatrix}3\\-3\\1\\-4\\-1\end{bmatrix}, \quad \vec{u}_4=\frac{1}{3}\begin{bmatrix}0\\0\\2\\1\\-2\end{bmatrix}, \quad \vec{u}_5=\frac{\sqrt{2}}{18}\begin{bmatrix}6\\-6\\1\\8\\5\end{bmatrix}. \tag{7.8}$$

For δ sufficiently large the eigenvectors of A are well-determined but as δ approaches zero, the determination of the eigenvectors $\vec{u}_2$ and $\vec{u}_3$ becomes ill-conditioned as was discussed in Chapter VI. For $\delta=0$, the matrix A has two coincident eigenvalues of 162 and so any linear combination

of $\vec{u}_2$ and $\vec{u}_3$ is also an eigenvector of A.

The tolerance on the norm of the residual matrix was determined by the formula

$$\varepsilon = \frac{||A||\rho}{n} \tag{7.9}$$

where ρ is specified beforehand, A is an n x n matrix, and $||A||$ is the Euclidian norm of A. In double precision on the IBM 360/65 computer, if $\rho = 10^{-15}$ then results accurate to the precision of the computer will be obtained. All of the following examples were run in double precision.

In all cases the original starting vector was a unit vector with all components equal. The additional starting vectors needed to determine the remaining eigenvectors were generated as discussed in Chapter V.

Table 7.2 gives a comparison of the number of iterations for Algorithms 1 and 2 with $\delta=0$ and $\delta=.01$. The results were calculated with $\varepsilon=10^{-7}$ corresponding to $\rho = 5.14 \times 10^{-9}$. The results in Table 7.2 indicate that the number of iterations become large as δ approaches zero. The reason that this happens is that as $\vec{x}_m$ approaches an eigenvector, an oscillatory behavior occurs which slows down the convergence. To take advantage of this oscillatory behavior the following two acceleration procedures were applied at every sixth iteration. For Algorithm 1, the acceleration procedure was to choose $\vec{g}_m$ by

$$\vec{g}_m = \frac{1}{2} t_{m-1}\vec{g}_{m-1} + t_{m-2}\vec{g}_{m-2} + \frac{1}{2} t_{m-3}\vec{g}_{m-3} \qquad (7.10)$$

rather than by Equation (3.1). Here $\vec{g}_j$ (j=m-1,m-2,m-3) were chosen by (3.1). For Algorithm 2, the acceleration procedure was applied to the direction vectors $\vec{z}_{1_m}$ and $\vec{z}_{2_m}$ at every sixth iteration by choosing them as

$$\vec{z}_{i_m} = \frac{1}{2} \alpha_{i_{m-1}} \vec{z}_{i_{m-1}} + \alpha_{i_{m-2}} \vec{z}_{i_{m-2}} + \frac{1}{2} \alpha_{i_{m-3}} \vec{z}_{i_{m-3}} \quad (i=1,2) \qquad (7.11)$$

rather than by Equations (6.28) and (6.29). These simple acceleration techniques often remarkably reduced the number of iterations as can be seen in Table 7.2.

The actual calculated eigenvectors and eigenvalues corresponding to the results given in Table 7.2 will not be presented. In all cases the eigenvalues were accurate representations to 16 decimal places of the exact eigenvalues given by (7.7). The corresponding eigenvectors were accurate representation to 10 decimal places of the exact eigenvectors given by (7.8) except for the ill-determined eigenvectors when δ=.01. These vectors were accurate to 8 decimal places. The vectors corresponding to 162 that were calculated by Algorithms 1 and 2 with δ=0 were $27\vec{u}_3 - 117\vec{u}_2$ and $-13\vec{u}_3 - 3\vec{u}_2$ normalized to unit length. With δ=.01, Algorithm 1 with acceleration was unable to obtain satisfactory approximations to either $\vec{u}_2$ or $\vec{u}_3$ in 10,000 iterations and so the iteration was stopped. In this case, $\vec{u}_2$ through $\vec{u}_5$ were not obtained.

Table 7.2. Comparison of number of iterations

Vector	Algorithm 1			Algorithm 2		
	$\delta = 0$	$\delta=0$ with acceleration	$\delta=.01$ with acceleration	$\delta=0$	$\delta=.01$	$\delta=.01$ with acceleration
$\vec{u}_1$	26	26	26	17	15	9
$\vec{u}_2$	27	0	-	6	505	66
$\vec{u}_3$	134	33	-	5	6	6
$\vec{u}_4$	1	1	-	3	3	3
$\vec{u}_5$	12	34	-	1	1	1

Algorithm 4 with p(m) = 4 was also tried on the matrix A given by (7.6) with δ=.01 and $\varepsilon=10^{-7}$. In this case the number of iterations for the eigenvectors $\vec{u}_1$, $\vec{u}_2$, $\vec{u}_3$, $\vec{u}_4$, and $\vec{u}_5$ was 5, 6, 5, 3, and 0, respectively. However, the calculated eigenvectors were accurate to more decimal places than those obtained from Algorithm 2 due to the extremely rapid convergence of this method. For instance in the determination of $\vec{u}_1$ the norm of the residual matrix was $.21 \times 10^{-6}$ after the fourth iteration but after the fifth iteration it was $.23 \times 10^{-12}$.

To illustrate the ill-conditioning behavior that was analyzed in Chapter VI, two examples will be given and discussed in detail. In both of these examples Algorithm 4 with p(m) = n-1 direction vectors was used.

Example 1:

Matrix A given by (7.6) with $\delta = 10^{-3}$ was formed and then scaled by dividing each element of A by 810. The resulting matrix has eigenvalues 486/810, 162(1 + δ)/810, 162(1 - δ)/810, 1, and -18/810 with corresponding eigenvectors given in (7.8). The parameter ρ was set at 10^{-15} with the resulting tolerance on the norm of the residual matrix calculated from (7.9) to be $\varepsilon = .24 \times 10^{-15}$.

Using the original starting vector, after 5 iterations the results were

$$\vec{u}_1^{(c)} = \begin{bmatrix} .7071067811865474 \\ .7071067811865474 \\ -.557 \times 10^{-17} \\ .459 \times 10^{-16} \\ -.441 \times 10^{-17} \end{bmatrix}, \quad \begin{aligned} c_{11} &= .9999999999999997, \\ c_{12} &= .308 \times 10^{-16}, \\ c_{13} &= .109 \times 10^{-16}, \\ c_{14} &= .145 \times 10^{-16}, \\ c_{15} &= .683 \times 10^{-16}, \end{aligned}$$

$\lambda_1^{(c)} = .5999999999999998$, and $||R_5|| = .58 \times 10^{-16}$. Clearly these results are accurate to the precision of the computer.

The second starting vector which was chosen as discussed in Chapter V was such that the sequence $\{\vec{x}_m\}$ converged to an approximation to $\vec{u}_2$. In this case the iteration became ill-determined at the seventh iteration since $c_4 = 0$ and $c_3 < 0$. In theory if $c_4 = 0$ then c_3 must equal zero also. At m = 6, the results were

$$\vec{u}_2^{(c)} = \begin{bmatrix} -.1666658231953821 \\ .1666658231953821 \\ .7222225033782103 \\ -.2222233468499687 \\ .6111108299532255 \end{bmatrix}, \quad \begin{aligned} c_{21} &= .139 \times 10^{-16}, \\ c_{22} &= .9999999999985769, \\ c_{23} &= .1686942095 \times 10^{-5}, \\ c_{24} &= .167 \times 10^{-15}, \\ c_{25} &= .139 \times 10^{-16}, \end{aligned}$$

$\lambda_2^{(c)} = .1998000000000010$, and $||R_6|| = .95 \times 10^{-9}$.

The convergence criteria is not satisfied but the c_{2i}'s indicate that the subspace is well-determined and that $\vec{u}_2^{(c)}$ and $\vec{u}_3$ are slightly separated. Note that $\lambda_2^{(c)}$ is a very good approximation to λ_2 and so when the remaining eigenvectors

are found the separation technique of Chapter VI can be applied to obtain results as accurate as can be expected.

The third starting vector resulted in convergence to an approximation to $\vec{u}_3$ and again the iteration became ill-determined at the sixth iteration. At $m = 6$, $F_6(1)$ and $F_6(\hat{t})$ were both greater than zero. At this point

$$\vec{u}_3^{(c)} = \begin{bmatrix} -.5000000000018924 \\ .5000000000018924 \\ -.1666666666584655 \\ .6666666666641429 \\ .1666666666736062 \end{bmatrix}, \quad \begin{aligned} c_{31} &= -.139 \times 10^{-16}, \\ c_{32} &= -.1135556926268 \times 10^{-10}, \\ c_{33} &= -.9999999999999980, \\ c_{34} &= -.194 \times 10^{-15}, \\ c_{35} &= -.278 \times 10^{-16}, \end{aligned}$$

$\lambda_3^{(c)} = .2001999999999999$, and $||R_5|| = .64 \times 10^{-14}$.

No problems were encountered in determining the fourth and fifth eigenvectors. These are given by

$$\vec{u}_4^{(c)} = \begin{bmatrix} .538 \times 10^{-17} \\ .360 \times 10^{-17} \\ .6666666666666665 \\ .3333333333333332 \\ -.6666666666666665 \end{bmatrix}, \quad \begin{aligned} c_{41} &= .635 \times 10^{-17}, \\ c_{42} &= -.416 \times 10^{-16}, \\ c_{43} &= -.139 \times 10^{-16}, \\ c_{44} &= .9999999999999997, \\ c_{45} &= .0, \end{aligned}$$

$\lambda_4^{(c)} = -.0222222222222222$, and $||R_3|| = .24 \times 10^{-16}$,

and

$$\vec{u}_5^{(c)} = \begin{bmatrix} .4714045207910315 \\ -.4714045207910315 \\ .0785674201318384 \\ .6285393610547087 \\ .3928371006591928 \end{bmatrix}, \quad \begin{aligned} c_{51} &= -.694 \times 10^{-16}, \\ c_{52} &= .278 \times 10^{-16}, \\ c_{53} &= .0, \\ c_{54} &= .0, \\ c_{55} &= .9999999999999997, \end{aligned}$$

$\lambda_5^{(c)} = .99999999999999980$, and $||R_1|| = .56 \times 10^{-16}$.

The separation routine was applied to the vectors $\vec{u}_2^{(c)}$ and $\vec{u}_3^{(c)}$ to give

$$\vec{w}_2 = \begin{bmatrix} .1666666666666476 \\ -.1666666666666478 \\ -.7222222222222285 \\ .2222222222222471 \\ -.6111111111111046 \end{bmatrix}, \text{ and } \vec{w}_3 = \begin{bmatrix} -.4999999999999997 \\ .4999999999999997 \\ -.1666666666666673 \\ .6666666666666669 \\ .1666666666666658 \end{bmatrix}.$$

Clearly, $\vec{w}_2$ and $\vec{w}_3$ are accurate approximations of $\vec{u}_2$ and $\vec{u}_3$ to at least 13 decimal places which is the maximum expected accuracy as was shown in Chapter VI.

Example 2:

Matrix A with $\delta = 10^{-7}$ was formed and scaled as before. The tolerance, ε, was set the same as in Example 1. The vectors $\vec{u}_1^{(c)}$, $\vec{u}_4^{(c)}$, and $\vec{u}_5^{(c)}$ were accurate approximations to $\vec{u}_1$, $\vec{u}_4$, and $\vec{u}_5$ as in Example 1. For this reason these results will not be repeated here. In determining approximations to $\vec{u}_2$ and $\vec{u}_3$ the iteration procedure became ill-

determined at the seventh and sixth steps, respectively, because $c_4 < 0$ in each case. The results at this point were

$$\vec{u}_2^{(c)} = \begin{bmatrix} -.1563963619593657 \\ .1563963619593655 \\ .7254826753744356 \\ -.2358228294032591 \\ .6075712606728058 \end{bmatrix}, \quad \begin{aligned} c_{21} &= .111 \times 10^{-5}, \\ c_{22} &= .99979045204, \\ c_{23} &= .02047076009, \\ c_{24} &= .971 \times 10^{-16}, \\ c_{25} &= .555 \times 10^{-16}, \end{aligned}$$

$\lambda_2^{(c)} = .1999999800167619$, and $||R_6|| = .116 \times 10^{-8}$;

and

$$\vec{u}_3^{(c)} = \begin{bmatrix} -.5033070182308483 \\ .5033070182307284 \\ -.1518473094243255 \\ .6619779119401982 \\ .1791416465346731 \end{bmatrix}, \quad \begin{aligned} c_{31} &= -.849 \times 10^{-13}, \\ c_{32} &= -.02047075283, \\ c_{33} &= -.99979045218, \\ c_{34} &= .740 \times 10^{-11}, \\ c_{35} &= .915 \times 10^{-13}, \end{aligned}$$

$\lambda_3^{(c)} = .2000000199832376$, and $||R_5|| = .116 \times 10^{-8}$.

In this case $\lambda_2^{(c)}$ and $\lambda_3^{(c)}$ are accurate to only 10 decimal places. By Equation (6.8) this is all the accuracy that can be expected with the c_{2i}'s and c_{3i}'s given above. The corresponding $\vec{w}_2$ will have a component of error in the direction of $\vec{u}_3$ of about $.02 \times 10^{-10}$ and since the eigenvalues are identical to 7 decimal places the maximum expected accuracy of $\vec{w}_2$ is 5 decimal places. Similar remarks hold for $\vec{w}_3$. These comments are verified since

$$\vec{w}_2 = \begin{bmatrix} .1666709584238013 \\ -.1666709584238013 \\ -.7222207916078584 \\ .2222164998630029 \\ -.6111125416763567 \end{bmatrix} \text{ and } \vec{w}_3 = \begin{bmatrix} -.4999985693450006 \\ .4999985693450004 \\ -.1666728660857015 \\ .6666685741575444 \\ .1666614209930705 \end{bmatrix}. \quad (7.12)$$

If the search technique described in Chapter VI is applied to $\vec{u}_2^{(c)}$ and $\vec{u}_3^{(c)}$ for one iteration after $F_m(t)$ becomes ill-determined then the ill-conditioned eigenvectors can be further separated giving

$$\vec{u}_2^{(c)} = \begin{bmatrix} -.1664504977551612 \\ .1664504977551612 \\ .7222942058469743 \\ -.2225104059066580 \\ .6110390028936454 \end{bmatrix} \text{ and } \vec{u}_3^{(c)} = \begin{bmatrix} -.5000720054344616 \\ .5000720054346039 \\ -.1663544250659174 \\ .6665705348299262 \\ .1669308423396815 \end{bmatrix}. \quad (7.13)$$

The corresponding eigenvalues are $\lambda_2^{(c)} = .1999999800000073$ and $\lambda_3^{(c)} = .2000000199999922$. Also $c_{23} = .00043$ and $c_{32} = .00043$. Now using (7.13) to calculate $\vec{w}_2$ and $\vec{w}_3$, one obtains the following 9 decimal place accurate eigenvectors. This is the maximum expected accuracy.

$$\vec{w}_2 = \begin{bmatrix} .16666666663612827 \\ -.1666666663612827 \\ -.7222222223240166 \\ .2222222226294003 \\ -.6111111110093165 \end{bmatrix} \text{ and } \vec{w}_3 = \begin{bmatrix} -.5000000000363406 \\ .5000000000363406 \\ -.1666666665091893 \\ .6666666666182121 \\ .1666666667999163 \end{bmatrix}. \quad (7.14)$$

If the results given in (7.12) are relabeled $\vec{u}_2^{(c)}$ and $\vec{u}_3^{(c)}$ and if these are used in (6.12) to calculate new eigenvalues, then by recalculating $\vec{w}_2$ and $\vec{w}_3$, eigenvectors accurate to 9 decimal places similar to those given by (7.14) are obtained.

VIII. CONCLUSION

In this thesis a class of norm reduction algorithms were developed and analyzed. These algorithms have the advantage of very low round-off error since the computation is done by modifying approximations to the eigenvectors and the original matrix is left unchanged. Certain difficulties that these algorithms can encounter were explained and alternative techniques were proposed and analyzed. It was shown that on a p decimal place computer the norm reduction algorithms cannot directly separate the eigenvectors corresponding to close eigenvalues identical to more than p/2 decimal places, whereas,the Jacobi method cannot satisfactorily handle matrices with elements that vary by more than p/2 orders of magnitude. In all the examples that were tried, results to the maximum expected accuracy of the computer were obtained or could be obtained by choosing the tolerance on the norm of the residual matrix sufficiently small.

Our goal was to develop a technique that would give precise results on large matrices and that would not have the loss of accuracy that many present day algorithms have as was pointed out in Chapter I. The largest matrices tested by the author were well-conditioned 15 x 15 matrices. On this size of matrix, results of comparable accuracy could

also be obtained by the Jacobi method. More extensive comparisons with other methods, especially for matrices larger than 20 x 20, should be made.

The Generalized Algorithm with various choices of the parameters, W_m, ν, and p(m), should be analyzed more extensively on large matrices as well as on small matrices. Comparisons of the number of operations for various algorithms should also be made. In addition, an analysis of the round-off error per iteration should be made for various choices of these parameters.

Several aspects of convergence need to be considered more thoroughly. For instance, studies of the rate of convergence should be made for various choices of W_m, ν, and p(m). Computational experience indicates that the rate of convergence is quadratic. Also the relationship between the starting vector and the eigenvector to which the norm reduction algorithms converge should be determined.

In the Generalized Algorithm there is no guarantee that for some m, $\vec{g}_m' \vec{\nabla} N_m = 0$ with $\vec{x}_m$ not an eigenvector. Should this situation arise, a better technique than that proposed in Chapter III should be developed. A transformation on $\vec{g}_m$ rather than a new choice for W_m would be preferred.

Different iterative schemes might also be considered. For instance, the constraint that $\vec{g}_m' \vec{x}_m = 0$ might be replaced by the constraint $||\vec{x}_m|| = 1$. Another interesting possi-

bility is the following two stage iterative scheme:

Let $\vec{x}_0$ and $\vec{y}_0$ be arbitrary. For m=0,1,2,... perform the following two stage iteration:

Stage 1. Form $\vec{x}_{m+1} = \vec{x}_m + t_m\vec{g}_m$ by choosing t_m and $\vec{g}_m$ such that $||R_{m+\frac{1}{2}}|| < ||R_m||$ where $R_m = A\vec{x}_m\vec{y}_m' - \vec{x}_m\vec{y}_m'A$ and $R_{m+\frac{1}{2}} = A\vec{x}_{m+1}\vec{y}_m' - \vec{x}_{m+1}\vec{y}_m'A$.

Stage 2. Form $\vec{y}_{m+1} = \vec{y}_m + s_m\vec{h}_m$ by choosing s_m and $\vec{h}_m$ such that $||R_{m+1}|| < ||R_{m+\frac{1}{2}}||$ where $R_{m+1} = A\vec{x}_{m+1}\vec{y}_{m+1} - \vec{x}_{m+1}\vec{y}_{m+1}'A$.

In analyzing this two stage procedure certainly one of the first questions that one would consider is do the two sequences, $\{\vec{x}_m\}$ and $\{\vec{y}_m\}$, converge to the same vector and is that vector an eigenvector?

The possibility of using a different scheme for separating eigenvectors corresponding to exceptionally close eigenvalues rather than the search technique should be considered.

The essential features of a new approach for solving the algebraic eigenvalue problem has been presented in this thesis but the preceding discussion indicates that there are several possibilities for future work. This approach is certainly a step in the right direction toward developing methods that give accurate results for matrices of high order.

IX. BIBLIOGRAPHY

Bodewig, E.
1959 Matrix calculus. New York, New York, Interscience Publishers, Incorporated.

Causey, R. L. and Gregory, R. T.
1961 On Lanczos' algorithm for tridiagonalizing matrices. SIAM [Society of Industrial and Applied Mathematics] Review 3: 322-328.

Erisman, Albert M.
1967 Eigenvectors of a general complex matrix by norm reduction. Unpublished Master of Science thesis. Ames, Iowa, Library, Iowa State University.

Faddeev, D. K. and Faddeva, V. N.
1963 Computational methods of linear algebra. San Francisco, California, W. H. Freeman and Company.

Forsythe, G. E. and Henrici, P.
1960 The cyclic Jacobi method for computing the principle values of a complex matrix. American Mathematical Society Transactions 94: 1-23.

Fox, L.
1965 An introduction to numerical linear algebra. New York, New York, Oxford University Press.

Givens, W.
1953 A method of computing eigenvalues and eigenvectors suggested by classical results on symmetric matrices. United States Bureau of Standards Applied Mathematics Series 29: 117-122.

Goldstine, H. H., Murray, F. J., and von Neumann, J.
1959 The Jacobi method for real symmetric matrices. Association for Computing Machinery Journal 6: 59-96.

Hestenes, M. R. and Karush, W.
1951 A method of gradients for the calculation of latent roots and vectors of a real symmetric matrix. United States National Bureau of Standards Journal of Research 47: 45-61.

Hotelling, H.
1933 Analysis of a complex of statistical variables into principle components. Journal of Educational Psychology 24: 417-441, 498-520.

Householder, A. S.
1964 The theory of matrices in numerical analysis. New York, New York, Blaisdell.

Isaacson, Eugene and Keller, Herbert B.
1966 Analysis of numerical methods. New York, New York, John Wiley and Sons, Incorporated.

Lambert, R. J. and Sincovec, R. F.
1968 Precision calculation of eigenvectors by norm reduction. Multilithed. Ames, Iowa, Document Library, Ames Laboratory.

Lanczos, C.
1950 An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. United States National Bureau of Standards Journal of Research 45: 255-282.

Ortega, J. M.
1960 On Sturm sequences for tridiagonal matrices. Stanford University Applied Mathematics and Statistics Laboratories Technical Report 4.

Rosser, J. B., Lanczos, C., Hestenes, M. R., and Karush, W.
1951 Separation of close eigenvalues of a real symmetric matrix. United States National Bureau of Standards Journal of Research 47: 291-297.

Sincovec, Richard F.
1967 Precise eigenvector basis for a symmetric matrix. Unpublished Master of Science thesis. Ames, Iowa, Library, Iowa State University.

White, Paul A.
1958 The computation of eigenvalues and eigenvectors of a matrix. SIAM [Society for Industrial and Applied Mathematics] Journal 6: 393-437.

Wilde, Douglass J.
1964 Optimum seeking methods. Inglewood Cliffs, New Jersey, Prentice-Hall, Incorporated.

Wilde, Douglass J. and Beightler, Charles S.
1967 Foundations of optimization. Englewood Cliffs, New Jersey, Prentice-Hall, Incorporated.

Wilkinson, J. H.
1958a The calculation of the eigenvectors of codiagonal matrices. Computer Journal 1: 90-96.

Wilkinson, J. H.
1958b The calculation of eigenvectors by the method of Lanczos. Computer Journal 1: 148-152.

Wilkinson, J. H.
1960 Error analysis of floating-point computation. Numerische Mathematik 2: 319-340.

Wilkinson, J. H.
1961 Rigorous error bounds for computer eigensystems. Computer Journal 4: 230-241.

Wilkinson, J. H.
1962 Error analysis of eigenvalue techniques based on orthogonal transformations. SIAM [Society for Industrial and Applied Mathematics] Journal 10: 162-195.

Wilkinson, J. H.
1965 The algebraic eigenvalue problem. London, England, Oxford University Press.

X. ACKNOWLEDGMENTS

The author wishes to thank Dr. Robert J. Lambert for his guidance and advice in preparing this dissertation. The time and effort he spent in helping form this paper are greatly appreciated.

The author also wishes to thank Mrs. Mary Gonzalez for her programming help.