THE EIGENVALUE PROBLEM OF TWO MATRICES

THE EIGENVALUE PROBLEM OF

TWO TYPES OF COMPOUND MATRICES

A THESIS

Presented to

The Faculty of Graduate Studies

by

Hans Ralph Bastel

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Mathematics

McMaster University

June, 1966

ACKNOWLEDGMENTS

The research reported in this thesis was done while the author was a graduate student of mathematics at McMaster University. It was directed by Professor S. Charmonman of the Mathematics Department.

The author acknowledges with deep gratitude his indebtedness to Professor Charmonman for suggesting this problem and for his invaluable advice, criticism and guidance throughout the entire period in which this research was carried out.

My sincere thanks are also due to Professor Kenworthy. Through his comments and criticisms, a number of significant improvements resulted in the computer programming part of this thesis.

At last, a vote of thanks to my wife, Erika, who unflinchingly typed pages of equations and uncomplainingly made revision after revision. Also, the author wishes to acknowledge the extensive help of Mrs. M. Hruboska in typing the final draft of the manuscript and carrying through the many arduous tasks necessary to the successful completion of this thesis.

ABSTRACT

B. Friedman proved in Eigenvalues of compound matrices (New York University, Mathematics Research Group, Research Rept. No. TW-16 (1951)) that if A and B are real square matrices of order n and $S = \begin{bmatrix} A & B \\ B & A \end{bmatrix}$, then λ (S), the eigenvalues of S, is the set of 2n numbers λ (P) and λ (Q) where P = A + B, and Q = A - B. In the present paper we give a simple proof and three extensions: (I) An algorithm to find the eigenvectors of S in terms of the eigenvectors of P and Q. (II) If S is a cyclic permutation matrix of order $2^i n$, where i is a positive integer, then λ (S) is the set of $2^i n$ eigenvalues of matrices of the type Q of orders $2^{i-1}n$, $2^{i-2}n$, 2n, n, and a matrix of the type P of order n. (III) If $T = \begin{bmatrix} A & B \\ B^T & A \end{bmatrix}$ then λ (T) is the set of 2n numbers λ (A + C) and λ (A - C), where $C^2 = B^T B$. For the coefficient matrix of the five-point difference equation approximating Laplace's equation in a rectangular domain, C is obtained by inspection. We found that the use of the smaller matrices P and Q is superior to the use of the original matrix in view of the effect of round-off error as well as the computer storage and the number of computational operations required.

TABLE OF CONTENTS

LIST OF TABLES

SECTION 1

INTRODUCTION

As Bernard Friedman has pointed out in [I], the problem of finding the eigenvalues of certain matrices can be simplified by taking into account the special patterns of submatrices of these matrices. He investigated matrices of the type

$$(I.1) \qquad \begin{bmatrix} A_1 & A_2 \\ A_2 & A_1 \end{bmatrix}$$

and of a more general system as shown below:

$$(I.2) \qquad \begin{bmatrix} A_1 & A_2 & ------ & A_n \\ A_n & A_1 & A_2 --- & A_{n-1} \\ A_2 & ------- & A_n - & A_1 \end{bmatrix}$$

where the A_i $(i=1,2,\ldots\ldots,n)$ are nxn matrices.

In this paper, system (I.1) was investigated from the point of view of the complete eigenvalue problem (that is the determination of eigenvalues as well as eigenvectors).

The main purpose of the investigation was to find out whether it is possible to extend Friedman's result to the case of eigenvectors.

In Section (2) of this paper we define certain basic concepts and give proofs of relevant theorems. Various algorithms used for solving the eigenvalue problem are described in Section (3). In Section (4) we give an error analysis of Gaussian elimination and Householder's method. We look at a special compound matrix in Section (5) and show how we can facilitate the eigenvalue problem by means of a solution by inspection. This particular matrix occurs in the numerical solution of partial differential equations. In Section (6) the methods discussed in Section (3) are utilized in the investigation of system (I.1). This section also sets forth our conclusions. In Appendix A we present a few more theorems, and the Fortran IV programs used in our investigations are exhibited in Appendix B.

This paper was presented as Srisakdi Charmonman and H. R. Bastel, "Eigenvalue problems of a 2nx2n matrix", 632^{nd} Meeting, AMS, New York, N-Y April 4-7, 1966.

SECTION 2

DEFINITIONS AND THEOREMS

Preliminaries. In most of these definitions and theorems we shall follow closely the notation of Friedman [1] and MacDuffee [2].

DEFINITIONS: A rectangular array containing m rows and n columns of elements of a field F is called a mxn matrix over F. A matrix whose elements are again matrices is called compound matrix. The function "det" whose domain is the set of all nxn matrices over F and whose range is a subset of F is called a determinant provided "det" satisfies the following three properties:

(1) det is a linear function of each column, that is, for any $k = 1, 2, \ldots, n$ and all $b, c \in F$, if $A_k = bB_k + cC_k$, then

$$\det (A_1, \ldots, bB_k + cC_k, \ldots A_n)$$
$$= b \det (A_1, ., B_k, ., A_n) + c \det (A_1, \ldots, C_k, \ldots A_n)$$

(2) if two adjacent columns of A are equal, then det A = 0; and

(3) det I = 1, where I is the identity matrix and n is the unity element of F.

The polynomial $\det (A - \lambda I)$ is called the characteristic polynomial of the matrix A. The equation $\det (A - \lambda I) = 0$ is called the characteristic equation of A. The eigenvalues of a matrix A are the roots of the characteristic equation of A. To determine the eigenvectors associated with the eigenvalue λ_i we solve for the vector X_i from

$$(A - \lambda_i I) X_i = 0$$

$$\text{or } AX_i = \lambda_i X_i$$

An nxn matrix A is said to be non - singular if and only if $\det(A) \neq 0$, otherwise A is said to be singular. Two nxn matrices A and B are said to be similar if and only if

$$A = PBP^{-1}$$

for some non-singular matrix P.

If $A = (a_{ij})$; $i = 1,\ldots, m$; $j = 1,\ldots, m$ then A is the identity matrix I if $a_{ij} = 1$ when $i=j$ and 0 otherwise. A is said to be a diagonal matrix if $a_{ij} = 0$ when $i \neq j$.

A is said to be upper triangular if $a_{ij} = 0$ when $i > j$. A triangular matrix for which $a_{ii} = 0$ is called strictly triangular.

Given $A = (a_{ij})$, the transpose of A, denoted by A^T is defined by

$$A^T = (b_{ij}); \text{ where } a_{ij} = b_{ji}$$

An nxn matrix is said to be symmetric if and only if

$$A = A^T$$

B is the inverse of A if and only if

$$A\,B = I = B\,A$$

Here we denote B by A^{-1}.

We define the left direct product by

$$A \cdot\!\times B = \begin{bmatrix} A\,b_{11} & \cdots & A\,b_{1S} \\ A\,b_{21} & \cdots & A\,b_{2S} \\ A\,b_{r1} & \cdots & A\,b_{rS} \end{bmatrix}$$

where A is an nxn matrix and B is an rxS matrix.

A matrix will be called a permutation matrix if the elements of any row are a permutation of the elements of the first row. If the k^{th} row of a matrix consists of the elements of the first row shifted cyclically $(k - 1)\,q$ places to the right, we call such a matrix a q - cycle permutation matrix.

A ring A is an algebraic system having operations of additions (+) and multiplication (·) and satisfying the following conditions:

(a) A is an abelian group under addition;

(b) multiplication is associative;

(c) multiplication is distributive relative to addition;

THEOREMS

THEOREM 1. If A and B are square matrices of order r, and C is a nxm matrix, then

$$(A+B).xC = A.xC+B.xC$$

where "+" denotes ordinary matrix addition.

Proof: Let $A = (a_{ij})$, $B = (b_{ij})$; $i,j = 1,2,\cdots, r$ and $C = (c_{st})$; $s,t = 1,2,..,n$

$(A+B)\cdot xC$

$$= \begin{bmatrix} (A+B)c_{11} & (A+B)c_{12} & \cdots & (A+B)c_{1n} \\ (A+B)c_{21} & (A+B)c_{22} & \cdots & (A+B)c_{2n} \\ " & " & & " \\ " & " & & " \\ " & " & & " \\ (A+B)c_{n1} & (A+B)c_{n2} & \cdots & (A+B)c_{nn} \end{bmatrix}$$

$$= \begin{bmatrix} Ac_{11} & Ac_{12} & \cdot\cdot & Ac_{1n} \\ Ac_{21} & Ac_{22} & \cdot\cdot & Ac_{2n} \\ " & " & & " \\ " & " & & " \\ " & " & & " \\ Acn_1 & Ac_{n2} & \cdot\cdot & Ac_{nn} \end{bmatrix} + \begin{bmatrix} Bc_{11} & Bc_{12} & \cdots & Bc_{1n} \\ Bc_{21} & Bc_{22} & \cdots & Bc_{2n} \\ " & " & & " \\ " & " & & " \\ " & " & & " \\ Bc_{n1} & Bc_{n2} & \cdots & Bc_{nn} \end{bmatrix}$$

$= A\cdot xC + B\cdot xC$

THEOREM 2. If A is of order r and B is of order n and if $A\cdot xB = C = (c_{ij})$, then

$$c_{ij} = a_{i_1 j_1} b_{i_2 j_2}$$

where $i = r(i_2 - 1) + i_1$, $1 \leq i_1 \leq r$, $1 \leq i_2 < n$

and $j = r(j_2 - 1) + j_1$, $1 \leq j_1 \leq r$, $1 \leq j_2 < n$

The proof follows directly from the definition.

THEOREM 3. $(AC)\cdot x(BD) = (A\cdot xB)(C\cdot xD)$

Let $k = r\ (k_2 - 1) + k_1$; $1 \leq k_1 < r$

Proof: $(A\cdot xB)(C\cdot xD)$

$$= (a_{i_1 j_1} b_{i_2 j_2})\ (c_{i_1 j_1} d_{i_2 j_2})$$

$$= \sum_k a_{i_1 k_1} b_{i_2 k_2} c_{k_1 j_1} d_{k_2 j_2}$$

$$= \sum_{k_1} a_{i_1 k_1} c_{k_1 j_2} \sum_{k_2} b_{i_2 k_2} d_{k_2 j_2}$$

$$= (AC)\cdot x(BD)$$

THEOREM 4. If λ is an eigenvalue of A and μ is an eigenvalue of B then $\lambda\mu$ is an eigenvalue of $A\cdot xB$.

Proof: Let X [Y] be an eigenvector of A[B] with respect to $\lambda[\mu]$ and let Z be the vector with components $X_k\ Y_j$. If $i = r(i_2-1)+i_1$, then the i^{th} component of $(A\cdot xB)Z$ is

$$\sum_{j=1}^{n} \left(\sum_{k=1}^{r} a_{i_1 k}\, b_{i_2 j}\, X_k Y_j \right)$$

$$= \left(\sum_{k=1}^{r} a_{i_1 k} X_k\right)\left(\sum_{j=1}^{n} b_{i_2 j} Y_j\right)$$

$$= (\lambda X_{i_1})(\mu Y_{i_2})$$

$$= \lambda\mu X_{i_1} Y_{i_2}$$

$$= \lambda\mu Z_i$$

Hence $(A \cdot xB)Z = \lambda\mu Z$, and $\lambda\mu$ is an eigenvalue of $A \cdot xB$

THEOREM 5. Let $A_1, A_2, \ldots, A_p$ be n - dimensional square matrices and $B_1, \ldots, B_p$ be r - dimensional square matrices. Suppose that $B_1, \ldots, B_p$ have a common eigenvector, X, and that the corresponding eigenvalues are $u_1, \ldots, u_p$ respectively, then the eigenvalues of

$$u_1 A_1 + \ldots\ldots + u_p A_p \tag{2.1}$$

will be eigenvalues of

$$C = A_1 \cdot xB_1 + A_2 \cdot xB_2 + \cdots + A_p \cdot xB_p \tag{2.2}$$

Proof: Let Y be an eigenvector of (2.1) corresponding to the eigenvalue v then

$$(u_1 A_1 + u_2 A_2 + \cdots + u_p A_p)\, Y = v\, Y \tag{2.3}$$

From the definition of X we have

$$B_1 X = u_1 X,\ B_2 X = u_2 X,\ \ldots,\ B_p X = u_p X.$$

Then by (2.3) it follows that

$$(A_1 \cdot xB_1 + A_2 \cdot xB_2 + \ldots + A_p \cdot xB_p)\,(Y \cdot xX)$$
$$= (A_1 \cdot xB_1)(Y \cdot xX) + \ldots + (A_p \cdot xB_p)(Y \cdot xX)$$
$$= (A_1\, Y) \cdot x(B_1 X) + \ldots + (A_p Y) \cdot x(B_p X)$$
$$= (A_1 Y) \cdot x(u_1 X) + \ldots + (A_p Y) \cdot x(u_p X)$$
$$= (u_1 A_1 Y + \ldots\ldots + u_p A_p Y) \cdot xX$$
$$= vY \cdot xX$$

This proves that $Y \cdot xX$ is an eigenvector of (2.2) corresponding to the eigenvalue v.

THEOREM 6. Let C be the matrix considered in Theorem 5. Suppose that the ring generated by the matrices $B_1, \ldots, B_p$ has an m - dimensional representation ($m<r$) in which the matrix B_k is represented by M_k. Then every eigenvalue of the n m-dimensional matrix

$$D = A_1 \cdot xM_1 + \ldots + A_p \cdot xM_p$$

is an eigenvalue of the nr-dimensional matrix

$$C = A_1 \cdot xB_1 + \ldots + A_p \cdot xB_p$$

The proof is given in [1].

THEOREM 7. Let C be a p-cycle permutation matrix, then the eigenvalues of any of the following matrices will be eigenvalues of C.

$T_o = (S_o)$

$$T_1 = \begin{bmatrix} 0 & 0 & 0 & \ldots & 0 & 0 & S_p^{t-1} \\ S_1 & 0 & 0 & \ldots & 0 & 0 & 0 \\ 0 & S_p & 0 & \ldots & 0 & 0 & 0 \\ " & & & & & " & \\ " & & & & & " & \\ " & & & & & " & \\ 0 & 0 & 0 & \ldots & S_p^{t-3} & 0 & 0 \\ 0 & 0 & 0 & \ldots & 0 & S_p^{t-2} & 0 \end{bmatrix}$$

$$T_{a_1} = \begin{bmatrix} 0 & 0 & \ldots\ldots\ldots\ldots & & S_{a_1 p}^{t-1} \\ S_{a_1} & 0 & \ldots\ldots\ldots\ldots & & 0 \\ 0 & S_{a_1 p} & 0 \ldots\ldots\ldots & & 0 \\ 0 & 0 & \ldots\ldots\ldots\ldots & 0 & 0 \\ 0 & & \ldots\ldots\ldots\ldots & S_{a_1 p}^{t-2} & 0 \end{bmatrix}$$

where

$$S_k = A_1 + \ell^k A_2 + \ell^{2k} A_3 + \ldots + \ell^{k(r-1)} A_r$$

$$k = 0, 1, \ldots r-1; \pmod r$$

and ℓ is a primitive r^{th} root of unity.

The proof follows [1].

COROLLARY. If we want to obtain the eigenvalues of $S = \begin{bmatrix} A & B \\ B & A \end{bmatrix}$ where A, B are square matrices of order n, we apply Theorem 7 and note that the eigenvalues of S are the eigenvalues of A + B and A - B.

THEOREM 8. The eigenvalues of a matrix are invariant under a similarity transformation.

Proof: If x is the eigenvector of A belonging to the eigenvalue λ, then

$$Ax = \lambda x$$

Premultiplication by H^{-1} gives

$$H^{-1} Ax = \lambda H^{-1} x$$

then

$$H^{-1} A (HH^{-1}) x = \lambda H^{-1} x$$

and

$$(H^{-1} AH) H^{-1} x = \lambda H^{-1} x$$

The eigenvalues are therefore preserved and the eigenvectors are multiplied by H^{-1}.

THEOREM 9. Let λ_i; $i=1,\cdots,2n$, be the eigenvalues of $S = \begin{bmatrix} A & B \\ B & A \end{bmatrix}$, μ_i, $i=1,\cdots,n_1$ the eigenvalues of $P = A+B$, and η_j, $j = n+1,\cdots,2n$ the eigenvalues of $Q = A-B$. Then the eigenvectors $X_i = \begin{bmatrix} X_{1i} \\ X_{2i} \end{bmatrix}$ corresponding to the μ_i are:

a) if any of the μ_i are equal to any of the η_j, then

$$X_{1i} = Z_i + Y_i$$

$$X_{2i} = -Z_i + Y_i$$

where
$$(P-\mu_i I)\ Y_i = 0$$

$$(Q-\mu_i I)\ Z_i = 0$$

$$(P-\eta_j I)\ Z_j = 0$$

b) if $\mu_i \neq \eta_j$, then

$$X_{1i} = X_{2i} = Y_i$$

Similarly, the eigenvectors corresponding to the η_j are:

a) if $\eta_j = \mu_i$, then

$$X_{1i} = Z_j + Y_i$$

$$X_{2i} = Z_j - Y_i$$

b) if $\eta_j \neq \mu_i$

then $X_{1i} = -X_{2i} = Z_j$

Proof: We consider the matrix equation

$$\begin{bmatrix} A & B \\ B & A \end{bmatrix} \begin{bmatrix} X_{1i} \\ X_{2i} \end{bmatrix} = \lambda_i \begin{bmatrix} X_{1i} \\ X_{2i} \end{bmatrix}$$

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

Multiplying out and equating terms gives

(2.6) $A\,X_{1i} + B\,X_{2i} = \lambda_i X_{1i}$

(2.7) $B\,X_{1i} + A\,X_{2i} = \lambda_i X_{2i}$

We add (2.6) and (2.7) and get

$$(A + B)(X_{1i} + X_{2i}) = \lambda_i (X_{1i} + X_{2i})$$

hence

(2.8) $P\,Y_i = \mu_i Y_i$: $i = 1,2,\cdots,n$

where $Y_i = X_{1i} + X_{2i}$

Similarly, by subtracting we get

(2.9) $Q\,Y_j = \eta_j$; $j = n+1,\cdots,2n$

where $Y_j = X_{1i} - X_{2i}$

We note that Y_i and Y_j may be obtained from (2.8) and (2.9), if the λ_i are known. Now we develop a method by means of which we can determine all eigenvectors (unique to a constant multiplier) corresponding to the μ_i. For this we go back to consider S.

$$\begin{bmatrix} A - \mu_i I & B \\ B & A - \mu_i I \end{bmatrix} \begin{bmatrix} X_{1i} \\ X_{2i} \end{bmatrix} = 0$$

Multiplying out the above matrix equation we get

(2.10) $(A - \mu_i I)\,X_{1i} + B\,X_{2i} = 0$

(2.11) $B\,X_{1i} + (A - \mu_i I)\,X_{2i} = 0$

We subtract (2.11) from (2.10) and get

$$(A - B - \mu_i I)\,(X_{1i} - X_{2i}) = 0$$

or

(2.12) $(Q - \mu_i I)\,Z_i = 0$

where $Z_i = X_{1i} - X_{2i}$

First we note that if

$$\left| Q - \mu_i I \right| = 0 \text{ then}$$

μ_i is an eigenvalue of Q as well as of P.

Hence, if any of the μ_i are equal to any of the η_j then

$$Z_i = k\,Y_j$$

for any constant k.

Thus

$$Z_i = (X_{1i} - X_{2i})$$

also $\quad Y_i = X_{1i} + X_{2i}$

Hence

$$X_{1i} = 1/2\ (Z_i + Y_i)$$

$$X_{1i} = 1/2\ (Y_i - Z_i)$$

or

$$X_{1i} = kY_j + Y_i$$

$$X_{2i} = Y_i - kY_j$$

but if $\mu_i \neq \eta_j$, with (i : $i<n$, $i \varepsilon$ positive integers) and (j : $n+1 \leq j \leq 2n$, $j \varepsilon$ positive integers), then

$$|Q-\mu_i I| \neq 0$$

Thus (2.12) implies that $Z_i = 0$, which by definition gives

$$X_{1i} = X_{2i}$$

But $$Y_i = X_{1i} + X_{2i}$$

Thus $$X_{1i} = X_{2i} = Yi$$

The eigenvalues corresponding to η_j are obtained in the same manner.

SECTION 3

NUMERICAL METHODS

HOUSEHOLDER'S METHOD FOR THE SOLUTION OF THE SYMMETRIC EIGENVALUE PROBLEM

Preliminaries: Householder suggested the use of symmetric matrices P, defined by

(3.10) $P = I - 2\, ww^T$

where w is a column vector such that

(3.2) $w^T w = 1$

The matrix, P, is symmetric and we have

(3.3) $$\begin{aligned} P^T P &= (I - 2\, ww^T)\,(I - 2\, ww^T) \\ &= I - 4\, ww^T + 4\, w\,(w^T w)\, w^T \\ &= I - 4\, ww^T + 4\, ww^T \\ &= I \end{aligned}$$

Hence P is also orthogonal.

We define w_r to be a vector with its first (r-1) components equal to zero, so that

(3.4) $w^T_r = (0, 0, \ldots, 0, x_r, \ldots, x_n)$, and, P_r, to be a matrix of the form, P, with $w = w_r$. From

(3.2) we get that

(3.5) $x^2_r + x^2_{r+1} + \ldots + x^2_n = 1$

The transformation to triple diagonal form, as shown later on in this section, is effected by (n-2) orthogonal similarity transformations with matrices $P_2, P_3, \ldots, P_{n-1}$ respectively.

The first transformation produces zeros in the 1^{st} row and the 1^{st} column, except those in the tridiagonal section. The second transformation produces zeros in the 2^{nd} column and the 2^{nd} row, except those in the tridiagonal section, and so on.

We denote the original matrix by $A^{(1)}$ and define $A^{(r)}$ by the relation

$$A^{(r)} = P_r \, A^{(r-1)} \, P_r \tag{3.6}$$

where $A^{(r-1)}$ contains (n-r) elements in row (r-1) which are to be reduced to zero by the transformation with P_r. This gives us (n-r) equations to be satisfied by the (n-r+1) elements of w_r. These equations, in addition to equation (3.5) above, determine the elements, but not quite uniquely. We are free to make that choice which will give the greatest numerical convenience.

Householder Method. The transformation with the matrices $P_2, \ldots, P_{n-1}$ are performed successively. A typical stage may be illustrated on hand of a 5x5 matrix after applying P_2 and P_3, the matrix $A^{(3)}$ is

$$A^{(3)} = \begin{bmatrix} \alpha_1 & \beta_2 & 0 & 0 & 0 \\ \beta_2 & \alpha_2 & \beta_3 & 0 & 0 \\ 0 & \beta_3 & Y & Y & \bar{Y} \\ 0 & 0 & Y & Y & Y \\ 0 & 0 & \bar{Y} & Y & Y \end{bmatrix}$$

In the transformation with P_4 only the elements of the 3x3 matrix, denoted by Y's are modified. The barred Y's are to be reduced to zeros. It is not hard to show that the general step in the Householder method is typified by the first. Thus the whole process may be illustrated by considering the first step in the reduction of a 4x4 matrix.

We let

$$A = \begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ b_1 & b_2 & c_2 & d_2 \\ c_1 & c_2 & c_3 & d_3 \\ d_1 & d_2 & d_3 & d_4 \end{bmatrix}$$

Now we want to determine P_2 such that the transformation P_2 A P_2 will introduce zeros into positions (1,3), (1,4) and thus in (3.1) and(4.1). We define w_2 by

(3.7) $$w_2^T = (0, x_2, x_3, x_4)$$

By (3.5) we get

(3.8) $$x_2^2 + x_3^2 + x_4^2 = 1$$

The first row of any matrix is unaltered by multiplication on the left by P_2, so that P_2 A P_2 will have zeros in positions (1.3) and 1.4) if and only if AP_2 has zeros in these positions. Thus we must choose w_2 so that this condition is satisfied. Now we have

(3.9) $$AP_2 = A - 2\ A\ w_2 w_2^T$$

If we write

(3.10) $$Aw_2 = p$$

where

(3.11) $$p^T = (p_1, p_2, p_3, p_4)$$

then the elements of the first row of AP_2 are

$$a_1,\ b_1 - 2p_1x_2,\ c_1 - 2\ p_1x_3,\ d_1 - 2p_1x_4$$

where

(3.12) $$p_1 = b_1x_2 + c_1x_3 + d_1x_4$$

We must have

$$c_1 - 2p_1x_3 = 0 \tag{3.13}$$
$$d_1 - 2p_1x_4 = 0$$

if

$$S = b_1^2 + c_1^2 + d_1^2 \tag{3.14}$$

we must also have

$$b_1 - 2p_1x_2 = \pm S^{1/2} \tag{3.15}$$

since the sum of the squares of the elements in any row must be invariant. Multiplying (3.15) by x_2 and equations (3.13) by x_3 and x_4 we get

$$p_1 - 2p_1 (x_2^2 + x_3^2 + x_4^2) = \pm x_2 S^{1/2}$$

Thus

$$p_1 = \pm x_2 S^{1/2} \tag{3.16}$$

Equation (3.15) therefore gives

$$b_1 \pm 2x_2^2 S^{1/2} = \pm S^{1/2}$$
$$x_2^2 = \tfrac{1}{2}\{1 \pm b_1/S^{1/2}\} \tag{3.17}$$

From equations (3.13) and (3.16) we get

$$x_3 = \mp c_1/2x_2S^{1/2} \tag{3.18}$$

and

$$x_4 = \mp d_1/2x_2S^{1/2} \tag{3.19}$$

where the upper and lower signs in (3.15), (3.17), (3.18) and (3.19) go together. Wilkinson points out in [3] that it is advantageous to use the following sign conventions:

$$x_2^2 = \tfrac{1}{2}\{1 + b_1 (\text{sign } b_1) / S^{1/2}\}$$
$$x_3 = c_1 (\text{sign } b_1) / 2x_2S^{1/2}$$
$$x_4 = d_1 (\text{sign } b_1) / 2x_2S^{1/2}$$

We now obtain an explicit expression for $P_2 A P_2$ in terms of w_2 and p. In the following calculation we omit the subscript 2 in w_2.

$$
\begin{aligned}
(P_2 A P_2) &= (I - 2\, ww^T)\,(A)\,(I - 2\, ww^T) \\
&= A - 2\, ww^T A - 2\, A\, ww^T + 4\, w\,(w^T A w)\, w^T \\
&= A - 2\, w\,(w^T A - (w^T A w)\, w^T) \\
&\qquad - 2\,(Aw - w\,(w^T A w))\, w^T \\
&= A - 2\, wq^T - 2\, qw^T
\end{aligned}
$$

where
$$
\begin{aligned}
q &= Aw - (w^T A w)\, w \\
&= p - Kw
\end{aligned}
$$

$K = (w^T A w) = (w^T p)$, is a scalar.

For the general case of order n there are $(n-1)^2$ multiplications in the calculation of $(p_2, \ldots, p_n)$ and $n(n-1)$ multiplications in the calculation of $(- 2wq^T - 2qw^T)$ if we take advantage of symmetry. The rest of the computation requires a number of multiplications which is of order n. The total number of multiplications in the reduction to tri-diagonal form is therefore

$$2[n^2 + (n-1)^2 + \ldots + 2^2] \approx 2/3 n^3$$

The number required in the Givens transformation is $4/3 n^3$. Also, there are approximately 2n square roots to be evaluated in Householder's method while there are $\frac{1}{2} n^2$ in Givens method.

THE QR TRANSFORMATION

Preliminaries. The QR algorithm is a modification of the well known LR method by Rutishauser [4]. It was developed by J. G. F. Francis [5] in 1959 in order to overcome the possible numerical instability of Rutishauser's algorithm. The transformations on which the QR method is based are orthogonal* and thus it can be expected that they are numerically stable. In this section we shall first state the QR algorithm and then prove the main theorems connected with it.

QR Algorithm. The method consists of forming a sequence of matrices $A^{(S)}$ where $A^{(1)} = A$. The matrix $A^{(S)}$ is then decomposed into the product of an orthogonal matrix $Q^{(S)}$ and an upper triangular matrix $R^{(S)}$. This is achieved by pre-multiplying $A^{(S)}$ by an orthogonal matrix $Q^{(S)^T} = (Q^{(S)})^{-1}$ chosen so as to reduce $A^{(S)}$ to an upper triangular matrix. $A^{(S+1)}$ is then formed by post-multiplying $R^{(S)}$ by $Q^{(S)}$. Thus

$$(3.20) \qquad \begin{cases} A^{(1)} = A \\ A^{(S)} = Q^{(S)}\,R^{(S)},\ A^{(S+1)} = R^{(S)}\,Q^{(S)},\ (S=1,2,\ldots) \end{cases}$$

The matrix $Q^{(S)}$ may be found explicitly or may only exist as a product of simple factors. We can also write the algorithm as a similarity transformation, e.g.

$$(3.21) \quad A^{(S+1)} = Q^{(S)^T}\,Q^{(S-1)^T}\ldots Q^{(1)^T}\,AQ^{(1)}\,Q^{(2)}\ldots Q^{(S)}$$

In this discussion we make use of some special notational conventions as outlined below.

---

* In this discussion we assume that our matrices are real. The results, however, can be generalized to apply to Hermitian matrices as well. (e.g. see [3], [4], [5], [6].)

(a) Lower case letters with bar underneath (e.g. $\underline{x}$) denote column vectors or columns of matrices. They usually have a suffix giving their position in the array.

(b) If we write a hat (e.g. $\wedge$) over a square matrix we mean the rectangular matrix obtained by omitting its first column. Similarly a hat over a vector indicates that its first element has been omitted.

(c) The transpose of A is written as A^T. By $\hat{A}^T$ we mean $(\hat{A})^T$. Row vectors always have a superscript T.

(d) The identity matrix I has columns $e_1, e_2, \ldots, e_n$ if I is of order n.

(e) If a matrix has a suffix in brackets the same affix usually appears in brackets with its columns and elements. For example

$$A^{(S)} = [\underline{a}_1^{(S)}, \underline{a}_2^{(S)}, \ldots, \underline{a}_n^{(S)}] = [a_{ij}^{(S)}]$$

$$\text{or } \hat{B}_i = [\underline{b}_{(i)2} \ldots, \underline{b}_{(i)n}]$$

$$\text{or } \underline{\hat{c}}_i^{(k)T} = \left[\underline{c}_{2i}^{(k)}, \underline{c}_{3i}^{(k)}, \ldots, \underline{c}_{ni}^{(k)}\right]$$

Theorems: In the following discussion we shall show that the ortho-triangular decomposition of any square matrix exists. The diagonal elements of the triangular matrix can always be made positive, and if this is so, and the matrix is non-singular, then the decomposition is unique. This will be shown in Theorem 3. We shall also show, that if certain conditions are satisfied, the matrix $A^{(S)}$ tends to an upper triangular matrix as $s \to \infty$, the diagonal elements of which are the eigen-values of A. But first we shall show that any matrix can be reduced to a triangular matrix by a similarity transformation using a suitable orthogonal matrix.

Theorem 1. For an arbitrary matrix A there exists an orthogonal transformation Q such that

$$Q^T A Q = T$$

where T is triangular.

Proof:* We shall prove this theorem by induction. If n=1, the theorem is true since a matrix of order 1 is triangular. Suppose the theorem is true for matrices of order (n-1) and let A be a matrix of order n. Let $\underline{v}$ be an eigenvector of A with modulus 1 corresponding to any eigenvalue, say λ_1. Let $\underline{v}_1$, $\underline{u}_1$, $\ldots \underline{u}_{n-1}$ be an orthonormal set of vectors.** If Q is the matrix whose columns are $\underline{v}_1$, $\underline{u}_1$, ..., $\underline{u}_{n-1}$ we have

$$A_1 = Q_1^T A Q_1 = \begin{bmatrix} \lambda_1 & \underline{w}^T \\ \underline{0} & B \end{bmatrix} \tag{3.22}$$

By the induction hypothesis there exists an orthogonal matrix P of order (n-1) such that $P^T B P = T_{n-1}$

Now let

$$Q_2 = \begin{bmatrix} 1 & \underline{0}^T \\ \underline{0} & P \end{bmatrix} \tag{3.23}$$

so that Q_2 is orthogonal of order n. Then from (3.22) and (3.23)

$$\begin{aligned} Q_2^T Q_1^T A Q_1 Q_2 &= Q_2^T \begin{bmatrix} \lambda_1 & \underline{w}^T \\ \underline{0} & B \end{bmatrix} Q_2 \\ &= \begin{bmatrix} \lambda_1 & \underline{w}^T P \\ \underline{0} & T_{n-1} \end{bmatrix} \\ &= T_n \end{aligned}$$

---

* This theorem in its more general form (e.g. matrices could be complex) is due to Schur.

** e.g. $\underline{v}_1^T \underline{u}_i = 0$, $(i=1,\ldots,n-1)$ $\underline{u}_i^T \underline{u}_j = \delta_{ij}$, $(i,j=1,\ldots,n-1)$

Where T_n is triangular of order n. Setting $Q = Q_1Q_2$ proves the theorem.

Theorem 2. For any vector $\underline{b}$ with, say m elements, an orthogonal matrix M exists such that $M^T\underline{b} = ||\underline{b}||e_1$. (This implies $\hat{M}^T\underline{b} = 0$, and $\underline{m}_1{}^T\underline{b} = ||\underline{b}||$)

Proof:* When $\underline{b}$ is zero, M can be any orthogonal matrix. We shall assume $\underline{b} \neq \underline{0}$, in which case the first column of M is uniquely determined.

By an elementary orthogonal matrix we mean a matrix which differs from the identity matrix by at most in one principal (2x2) submatrix. This submatrix (say of the matrix T) is of the form

$$\begin{bmatrix} t_{ii} & t_{ij} \\ t_{ji} & t_{jj} \end{bmatrix} = \begin{bmatrix} \cos\Theta & -\sin\Theta \\ \sin\Theta & \cos\Theta \end{bmatrix}$$

where Θ is real.

An orthogonal matrix M^T, such that $M^T\underline{b} = ||\underline{b}||e_1$ can conveniently be constructed out of a series of elementary orthogonal matrices,

$$M^T = T_m\, T_{m-1} \cdots T_1 .$$

If $\underline{b}$ is multiplied in turn by the T_i, $(i=1, \ldots, m)$, then T_1 makes the first element b_1 of $\underline{b}$ non-negative, and the other transformations eliminate in turn the remaining elements b_i $(i=2,3, \ldots, m)$. We define $T_r = I$ if $b_p = 0$ for all $p \leq r$. Otherwise the elements of T_r are given by $t_{ij}^{(r)} = \delta_{ij}$, except for $t_{r1}^{(r)}$, $t_{1r}^{(r)}$, $t_{11}^{(r)}$, and $t_{rr}^{(r)}$.

For r = 1, $t_{11}^{(r)} = b_1/|b_1|$

(T_1 is a diagonal matrix)

For r = 2, 3,..., m

$$t_{11}^{(r)} = t_{rr}^{(r)} = \left(1 - |b_r^2| \Big/ \sum_{p\leq r} |b_p|^2\right)^{1/2}$$

$$t_{r1}^{(r)} = -\, t_{1r}^{(r)} = -\, b_r \Big/ \left(\sum_{p\leq r} |b_p|^2\right)^{1/2}$$

---

* This proof is due to Francis [5].

The first column $\underline{m}_1$ of M is unique, for since

$$M^T\underline{b} = ||b||\underline{e}_1$$

we have $$M\underline{e}_1 = \underline{m}_1 = (1/||b||)\underline{b}$$

Theorem 3. For any Matrix A, of order n, there exists an orthogonal matrix Q such that A = QR where R is an upper triangular matrix which has real, non-negative diagonal elements. Moreover, Q is unique if A is non-singular.

Proof:*

(a) EXISTENCE:

By Theorem 2 we can transform A so as to eliminate the elements below the main diagonal column by column starting on the left. Thus we reduce it to a triangular matrix. We define $B_1 = A$ and form $B_{i+1} = \hat{M}_i^T B_i$, $(i = 1, 2, \ldots, n-1)$. M_i is determined for $i = 1, \ldots, n$, such that $M_i^T \underline{b}_{(i)1} = ||\underline{b}_{(i)1}||\underline{e}_i$

where $\underline{b}_{(i)1}$ is the first column of B_i.

Now if

$$N_i = \begin{bmatrix} I_{i-1} & 0 \\ 0 & M_i \end{bmatrix}$$

where M_i is of order $n-i+1$, then $N_n^T N_{n-1}^T \ldots N_1^T A = R$

where $r_{ii} = ||b_{(i)1}||$ and $r_{ij} = 0$ for $i>j$.

Thus if $Q = N_1 N_2 \ldots N_n$ we have $A = QR$.

(b) UNIQUENESS:

Now suppose that we have two ortho-triangular decompositions $A = Q_1R_1 = Q_2R_2$. If A is non-singular then so are R_1 and R_2; we then have $R_1R_2^{-1} = Q_1^TQ_2$ and, as $(Q_1^TQ_2)$ is orthogonal, $(R_1R_2^{-1})^{-1} = (R_1R_2^{-1})^T$,

---

* Proof due to Francis [5].

which shows that $R_1R_2^{-1}$ is diagonal, since the left-hand side is upper-tridiagonal and the right-hand side is lower-tridiagonal. Furthermore, if we consider the diagonal,

$$r_{(2)ii}/r_{(1)ii} = \hat{r}_{(1)ii}/\hat{r}_{(2)ii} \text{ and thus, as the}$$

$r_{(1)ii}$ and $r_{(2)ii}$ are real and positive $r_{(1)ii} = r_{(r)ii}$. Hence $R_1R_2^{-1} = I$, so that $Q_1 = Q_2$ and thus Q is unique.

Theorem 4. If A is non-singular the matrix $P^{(S)} = Q^{(1)} \ldots Q^{(S)}$, such that $A^{(S+1)} = P^{(S)T}AP^{(S)}$ can be arrived at from the ortho-triangular decomposition of $A^{(S)}$ via $A^{(S)} = P^{(S)}S^{(S)}$ where $S^{(S)} = R^{(S)}R^{(S-1)} \ldots R^{(1)}$ and $P^{(S)} = Q^{(1)} \ldots Q^{(S)}$.

Proof:* Equation (3.21) gives $Q^{(1)} \ldots Q^{(S-1)} A^{(S)} = AQ^{(1)} \ldots Q^{(S-1)}$ and, as $A^{(S)} = Q^{(S)} R^{(S)}$, we get

$$Q^{(1)} \ldots Q^{(S)} R^{(S)} = AQ^{(1)} \ldots Q^{(S-1)}$$

$$Q^{(1)} \ldots Q^{(S-1)}R^{(S-1)} = AQ^{(1)} \ldots Q^{(S-2)}$$

$$- -$$

and so on.

Now, if we write $P^{(S)} = Q^{(1)} \ldots Q^{(S)}$ and $S^{(S)} = R^{(S)} R^{(S-1)} \ldots R^{(1)}$ then

$$\begin{aligned} P^{(S)} S^{(S)} &= Q^{(1)} Q^{(2)} \ldots Q^{(S)} R^{(S)} R^{(S-1)} \ldots R^{(1)} \\ &= Q^{(1)} Q^{(2)} \ldots Q^{(S-1)} A R^{(S-1)} \ldots R^{(1)} \\ &= Q^{(1)} Q^{(2)} \ldots Q^{(S-2)} A R^{(S-2)} \ldots R^{(1)} \\ &- \\ &= A^{(S)} \end{aligned}$$

Since $P^{(S)}$ is orthogonal and $S^{(S)}$ is triangular and, by Theorem 3, the ortho-triangular decomposition of a non-singular matrix is unique, the theorem is proved.

---

* Proof due to Francis [5].

Theorem 5. If the eigenvalues of A are such that $|\lambda_1|>|\lambda_2|>\ldots>|\lambda_n|$ and if the A^k, (k=1, ..., n) are non-singular then $A^{(S)}$ converges to an upper-triangular matrix.

Proof:* Since an A, as defined above, has linear divisors we may write

$$A^k = X\mathrm{diag}(\lambda_i{}^k)X^{-1} = XD^kY .$$

We define matrices Q, R, L, S by the relations

$$(X = QR), (Y = LS)$$

where R and S are upper-triangular, L is unit lower-triangular and Q is orthogonal. We note that all four matrices thus defined are independent of k. Since X is non-singular it follows that R is non-singular, too. Furthermore, we note that the QR decomposition always exists (Theorem 1) but a triangular decomposition of Y exists only if all its leading principal minors are non-zero. We have

$$(3.24) \qquad A^k = Q\ R\ D^kL\ S = Q\ R\ (D^kL\ D^{-k})\ D^kS$$

so that $D^kL\ D^{-k}$ is a unit lower-triangular matrix. Its $(i,j)^{th}$ element is given by $e_{ij}(\lambda_i/\lambda_j)^k$ when i>j and thus we may write

$$D^kL\ D^{-k} = I + E^{(S)} \text{ where } E^{(S)}\to 0 \text{ as } s\to\infty$$

Equation (3.24) now gives

$$\begin{aligned} A^k &= QR\ (I + E^{(S)})D^kS \\ &= Q\ (I + RE^{(S)}R^{-1})\ R\ D^kS \\ &= Q\ (I + F^{(S)})\ R\ D^kS \end{aligned}$$

Where $F^{(S)}\to 0$ as $S\to\infty$. Now $(I + F^{(S)})$ may be factorized into the product of an orthogonal matrix $\bar{Q}^{(S)}$ and an upper-triangular matrix $\bar{R}^{(S)}$ and since $F^{(S)}\to 0$, $\bar{Q}^{(S)}$ and $\bar{R}^{(S)}$ both tend to I. Hence we get

---

* Proof is due to Wilkinson [3]. But other more sophisticated proofs have been obtained by Kublanovskaya [7] and Householder [8].

(3.25) $A^k = (Q\,\bar{Q}^{(S)})\ (\bar{R}^{(S)}R\ D^kS)$

The first factor in (3.25) is orthogonal and the second is upper-triangular. Since A^k is non-singular its factorization into such an expression is unique and therefore $P^{(S)}$ in Theorem 4 is equal to $Q\,\bar{Q}^{(S)}$ apart possibly from a post-multiplying diagonal orthogonal matrix. Hence $P^{(S)}$ converges to Q. If we insist that all $R^{(S)}$ have positive diagonal elements we can find the ortho-diagonal factor from (3.25) writing

(3.26) $D = |D|D_1\ ,\ S = D_2(D_2^{-1}S)$

where D_1 and D_2 are orthogonal diagonal matrices and $D_2^{-1}S$ has positive diagonal elements, ($\bar{R}^{(S)}$ and R already have positive diagonal elements), we obtain from (3.25)

$$A^k = Q\,\bar{Q}^{(S)}D_2D_1^{\,k}\ \{\ (D_2D_1^{\,k})^{-1}\ \bar{R}^{(S)}R(D_2D_1^{\,k})\ D^{\ k}(D_2^{-1}S)\}.$$

The matrix in braces is upper-triangular with positive diagonal elements and thus $P^{(S)}$ approaches $QD_2D_1^{\,k}$ showing that ultimately $Q^{(S)}$ becomes D_1.

ELEMENTARY TRANSFORMATION

The reduction of a matrix of general form to a condensed form such as Hessenberg or triangular, can be achieved by performing a sequence of simple similarity transformations. The matrices employed to perform such transformations are called ELEMENTARY MATRICES. The transformations based on these matrices are referred to as ELEMENTARY TRANSFORMATIONS. Gaussian elimination can be looked upon as such an elementary transformation. As will be shown below, the transformation matrices in this particular case are a sequence of matrices P_i where the i^{th} column written as a row vector is

$$(0,\ 0,\ \ldots,\ 0,\ 1,\ p_{i+1},\ i,\ \ldots,\ n_{ni}).$$

Gaussian Elimination.

Consider a non-singular system of equations

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= a_{1,n+1} \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= a_{2,n+1} \\
&\cdots\cdots \\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= a_{n,n+1}
\end{aligned}
\tag{3.27}
$$

where for notational simplicity we have written the right hand side vector components as $a_{j,n+1}$; $(j=1, \ldots, n)$. Now suppose $a_{11} \neq 0$. We subtract the multiple a_{i1}/a_{11} of the first equation from the i^{th} equation $(i = 2, \ldots, n)$ to get

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= a_{1,n+1} \\
a_{22}^{(1)}x_2 + \cdots + a_{2n}^{(1)}x_n &= a_{2,n+1}^{(1)} \\
&\cdots\cdots \\
a_{n2}^{(1)}x_2 + \cdots + a_{nn}^{(1)}x_n &= a_{n,n+1}^{(1)}
\end{aligned}
\tag{3.28}
$$

The new coefficients $a_{ij}^{(1)}$ are given by the following relation:

$$
a_{ij}^{(1)} = a_{ij} - (a_{i1}/a_{11})a_{1j} \tag{3.29}
$$

where $(i = 2, \ldots, n)$ and $(j = 2, \ldots, n+1)$.

Now if $a_{22}^{(1)}$ in (3.28) is non-zero, we subtract $a_{i2}^{(1)}/a_{22}^{(1)}$ times the second equation from the i^{th} equation in (3.28) $(i = 3, \ldots, n)$ and get

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= a_{1,n+1} \\
a_{22}^{(1)}x_2 + \cdots + a_{2n}^{(1)}x_n &= a_{2,n+1}^{(1)} \\
a_{33}^{(2)}x_3 + \cdots + a_{3n}^{(2)}x_n &= a_{3,n+1}^{(2)} \\
&\cdots\cdots \\
a_{n3}^{(2)}x_3 + \cdots + a_{nn}^{(2)}x_n &= a_{n,n+1}^{(2)}
\end{aligned}
\tag{3.30}
$$

with

$$a_{ij}^{(2)} = a_{ij}^{(1)} - \frac{a_{i2}^{(1)}}{a_{22}^{(1)}} a_{2j}^{(1)} \tag{3.31}$$

where $(i = 3, \ldots, n)$ and $(j = 3, \ldots, n+1)$.

Again, if $a_{33}^{(2)} \neq 0$, we may eliminate all elements a_{13} $(i = 4, \ldots, n)$. Continuing with this process through $(n-1)$ steps we arrive at the final system (3.32).

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= a_{1,n+1} \\ a_{22}^{(1)}x_2 + \cdots + a_{2n}^{(1)}x_n &= a_{2\ n+1}^{(1)} \\ a_{33}^{(2)}x_3 + \cdots + a_{3n}^{(2)}x_n &= a_{3,n+1}^{(2)} \\ \cdots\cdots\cdots\cdots\cdots\cdots & \\ a_{nn}^{(n-1)}x_n &= a_{n,n+1}^{(n-1)} \end{aligned} \tag{3.32}$$

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} a_{kj}^{(k-1)} \tag{3.33}$$

$$k = 1, \ldots, n-1$$

$$j = k+1, \ldots n+1$$

$$i = k+1, \ldots, n$$

$$a_{ij}^{(0)} = a_{ij}$$

From (3.32), the back substitution process is carried out by the use of

$$x_i = \frac{1}{a_{ii}^{(i-1)}} \left\{ a_{i,n+1}^{(i-1)} - \sum_{j=i+1}^{n} a_{ij}^{(i-1)} x_j \right\} (i=n, \ldots, 1) \tag{3.34}$$

The process leading to (3.32) is called Forward Elimination while the calculation of the solution (3.34) is called Back-substitution.

The diagonal elements a_{11}, $a_{22}^{(1)}$, ..., $a_{nn}^{(n-1)}$ are called PIVOTS. If at any stage one of these pivots vanishes, we attempt to rearrange the remaing rows so as to obtain a non-vanishing pivot.

If this is impossible, then our system (3.27) is singular, and hence it has no solution.

For some systems, though a pivot is not zero, it may be small compared to other elements in the column being eliminated at that stage. In such cases the multipliers (e.g. a_{12}/a_{11} etc.) will be larger than unity in magnitude. The use of larger multipliers will, as we shall see in the error analysis part, lead to a possible increase in errors both during the elimination and during the backsubstitution phase of the process. This magnification of errors can be reduced if we interchange the rows such that the pivot at any stage is larger in magnitude than any remaining element in that column. Gaussian elimination when modified in this way is called PARTIAL PIVOTING or PIVOTAL CONDENSATION. Similarly at any stage, say the r^{th} stage, we may select as pivot the element of largest magnitude in the whole of the remaining n+1-r square array. This then is called COMPLETE PIVOTING.

Matrix Equivalent. It is possible to express Gaussian elimination in a more compact form by dealing with matrices rather than with individual elements.

We write:

(3.35) $$Ax = b$$

where

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ " & " & & " \\ " & " & & " \\ " & " & & " \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

$$x^T = (x_1, x_2, \ldots, x_n)$$

$$b^T = (b_1, b_2, \ldots, b_n)$$

Now, if we take pivots down the diagonal, it is easy to show (by straight matrix multiplication) that the first condensed (reduced) set of equations (3.28) has the matrix representation

$$P_1 Ax = P_1 b, \tag{3.36}$$

where

$$P_1 = \begin{bmatrix} 1 & 0 & 0 \ldots 0 & 0 \\ p_{21} & 1 & 0 \quad\; 0 & 0 \\ p_{31} & 0 & 1 \quad\; 0 & 0 \\ '' & & & \\ '' & & & \\ '' & & & \\ p_{n1} & 0 & 0 \ldots 0 & 1 \end{bmatrix} \tag{3.37}$$

and

$$p_{r1} = - a_{r1}/a_{11}.$$

The second reduction (3.29) has the matrix representation

$$P_2 P_1 Ax = P_2 P_1 b$$

where

$$P_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \ldots & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & p_{32} & 1 & 0 & 0 & 0 \\ 0 & p_{42} & 0 & 1 & 0 & 0 \\ '' & & & & & \\ '' & & & & & \\ '' & & & & & \\ 0 & p_{n2} & 0 & 0 \ldots & 0 & 1 \end{bmatrix} \tag{3.38}$$

The final set of equations is given by

(3.39) $$P_{n-1}\, P_{n-2} \cdots P_1 Ax = P_{n-1} \cdots P_1 b$$

where all P_j's $(j=n-1, \ldots, 1)$ are lower triangular matrices. Again it is easy to show by matrix multiplication that the product of lower triangular matrices is itself a matrix of that type. Thus if we let

$$P_{n-1}\, P_{n-2} \cdots P_1 = P$$

(3.39) reduces to

(3.40) $$P\, Ax = Pb$$

But now we note that the final matrix (3.32) operating on x is an upper triangular matrix, say U. Thus on the left hand side of (3.40) we have carried out a process equivalent to

(3.41) $$P\, A = U$$

which implies that $A = P^{-1}\, U$ and since the inverse of a lower triangular matrix is again lower triangular we have obtained A in terms of the product of an upper and lower triangular matrix. The matrix P^{-1} is easily obtained by noting that

$$P^{-1} = P_1^{-1}\, P_2^{-1} \cdots P_{n-1}^{-1}$$

and that each P_i^{-1}, $(i=1, \ldots, n-1)$ is identical with P_i except that the sign of each element p_{ij} is changed. Thus we get

$$P^{-1} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ -p_{21} & 1 & 0 & \cdots & 0 & 0 \\ -p_{31} & -p_{32} & 1 & \cdots & 0 & 0 \\ " & & & & & \\ " & & & & & \\ " & & & & & \\ -p_{n1} & -p_{n2} & -p_{n3} & \cdots & -p_{n,n-1} & 1 \end{bmatrix}$$

Number of Operations. As a conclusion of this discussion we shall attempt to evaluate the number of simple arithmetical operations necessary for Gaussian elimination and backsubstitution. Since on most modern computers division and multiplication times are much greater than addition times, we shall disregard the count of additions.

Forward Elimination. In forward elimination A is reduced to an upper-triangular matrix. If pivots are taken down the diagonal, the reduction from k to k-1 equations involves the following steps:

(i) we form the reciprocal of the pivot,

(ii) we multiply the rest of the pivotal column by this reciprocal. This involves k-1 multiplications.

(iii) To obtain the new set of k-1 equations we have to perform one multiplication for each element. This will give us k-1 multiplicationson the right and k-1 multiplications on the left.

For the full forward elimination, we form n reciprocals, and

$$\sum_{1}^{n} \{(k-1)+(k-1)^2+(k-1)\} \text{ multiplications}$$

Using formulae from [9], e.g.

$$\sum_{1}^{n} k = n(n+1)/2 \text{ and } \sum_{1}^{n} k^2 = n(n+1)(2n+1)/6,$$

we obtain the following total

$$n\{(1/3n^2 - 1/3) + 1/2(n-1)\}$$

$$= n(1/3n^2 + 1/2n - 5/6)$$

Backsubstitution. In the backsubstitution process, for each right hand side the computation of x_n involves one multiplication by the reciprocal of the final pivot, for x_{n-1} we have two multiplications, down to x_1 which needs n multiplications. The total number of multiplications is

$$1+2+3+\ldots+n = 1/2n(n+1)$$

n reciprocals and $n(1/3n^2+n-1/3)$ multiplications.

EBERLEIN'S METHOD

Preliminaries. The Jacobi method for the calculation of eigenvalues of a symmetric matrix has been described by Ralston [6]. It consists of finding a sequence of two-dimensional orthogonal transformations U_i such that if $U = \Pi_i U_i$ then $U^* A U$ is approximately diagonal with the approximations to the eigenvalues appearing on the diagonal [1]. The U_i are determined at each step of the iteration by minimizing the function

$$\tau^2(A) = \sum_{i \neq j} |a_{ij}|^2 \tag{3.42}$$

Here we shall describe a generalization of Jacobi's method due to EBERLEIN [14]. We shall show that for an arbitrary complex matrix A, a matrix P may be generated from a seouence of two-diemensional transformations $P_i(k,m)$, where (k,m) is the pivot-pair, such that if $A_L = P^{-1}AP, (P = \Pi_i P_i)$ then the absolute value of every element of $(A_L A^*_L - A_L^* A_L)$ is arbitrarily small. Each P_i is of the form

$$p_{ij} = \delta_{ij} \tag{3.43}$$

$$p_{kk} = e^{-i\beta}\cos(Z),\ p_{km} = -e^{i\alpha}\sin(Z)$$

$$p_{mk} = e^{-i\alpha}\sin(Z),\ p_{mm} = e^{i\beta}\cos(Z)$$

---

1 We denote the complex conjugate of A by A*. The real and complex parts of A are denoted by $\mathcal{R}(A)$ and $I_m(A)$ respectively.

where α and β are real and $z = x + i\,y$.

The theorem underlying EBERLEIN'S method is due to MIRSKY [15],

(3.44) $$\underset{P}{\text{infimum}}\ N^2(P^{-1}AP) = \sum_{i=1}^{n} |\lambda_i|^2$$

where P is non-singular, λ_i are the eigenvalues of the nxn matrix A and

(3.45) $$N^2(A) = \sum_{i,j} |a_{ij}|^2$$

Before proving some lemmas and a theorem we consider the effect of a transformation (3.43), $P_i(k,m)$ with pivot (k,m) on an arbitrary matrix. We let $A^1 = P_i^{-1}AP_i$ be the transformed matrix. We find

(3.46)
$$a^1_{ij} = a_{ij} \quad (i,j \neq k,m)$$
$$a^1_{ki} = e^{i\beta}a_{ki}\cos(Z) + e^{i\alpha}a_{mi}\sin(Z)$$
$$a^1_{ik} = e^{-i\beta}a_{ik}\cos(Z) + e^{-i\alpha}a_{im}\sin(Z) \quad (i\neq k,m)$$
$$a^1_{mi} = e^{-i\beta}a_{mi}\cos(Z) - e^{i\alpha}a_{ki}\sin(Z)$$
$$a^1_{im} = e^{i\beta}a_{im}\cos(Z) - e^{i\alpha}a_{ik}\sin(Z)$$
$$a^1_{kk} = 1/2\ \{(a_{kk}+a_{mm}) + D_{km}\cos(2Z) + \xi_{km}\sin(2Z)\}$$
$$a^1_{km} = 1/2\ e^{i(\alpha+\beta)}\{+\eta_{km} - D_{km}\sin(2Z) + \xi_{km}\cos(2Z)\}$$
$$a^1_{mk} = 1/2\ e^{-i(\alpha+\beta)}\{-\eta_{km} - D_{km}\sin(2Z) + \xi_{km}\cos(2Z)\}$$
$$a^1_{mm} = 1/2\ \{(a_{kk}+a_{mm}) - D_{km}\cos(2Z) - \xi_{km}\sin(2Z)\}$$

where

(3.47)
$$D_{km} = a_{kk} - a_{mm},\ B_{km} = a_{km} + a_{mk}$$
$$E_{km} = a_{km} - a_{mk}$$
$$\xi_{km} = B_{km}\cos(\alpha-\beta) - iE_{km}\sin(\alpha-\beta)$$
$$\eta_{km} = E_{km}\cos(\alpha-\beta) - iB_{km}\sin(\alpha-\beta)$$

Since we are considering the effect of a single transformation, we shall omit the subscripts k and m unless they are needed to avoid ambiguity. The effect of $N^2(A)$ is found by straightforward but tedious calculations:

$$\Delta A\ N^2\ (y,\alpha-\beta) \equiv N^2\ (A) - N^2(A^1)$$

(3.48) $$= G(1 - \cosh(2y)) - H(\sinh(2y))$$

$$+ 1/2\ (|D|^2 + |\xi|^2)\ (1 - \cosh(4y)) + 1/2\ i(D\xi^* - D^*\xi)\sinh(4y)$$

where

(3.49) $$G = G_{km} = \sum_{i\neq k,m} \{\ |a_{ki}{}^2| + |a_{ik}{}^2| + |a^2{}_{mi}| + |a^2{}_{im}|\ \}$$

$$H = H_{km} = -\mathcal{R}(K)\sin(\alpha-\beta) + I_m(K)\cos(\alpha-\beta)$$

and

$$K = 2 \sum_{i\neq i,m} (a_{ki}a^*{}_{mi} - a^*{}_{ik}a_{im})$$

In the following lemmas and theorem we assume that A has been normalized so that $N^2(A)\leq 1$. We also assume that $C = A\ A^* = A^*\ A$.

Lemma 1. For fixed (k,m) and arbitrary x and β, let[3] $A^1 = S^{-1}AS$ where S is defined by (3.43). Define α and y by

(3.50) $$\tan(\alpha-\beta) = -\frac{\mathcal{R}(C_{km})}{I_m(C_{km})}$$

$$\tanh(y) = \frac{\sin(\alpha-\beta)\mathcal{R}(C_{km}) - \cos(\alpha-\beta)I_m\ (C_{km})}{G + 2\ (|\xi|^2 + |D|^2)}$$

Then

(3.51) $$\Delta N^2(A) \geq \frac{4}{3}\ \frac{|C_{km}|^2}{G + 2\ (|\xi|^2 + |D|^2)}$$

$$\geq \frac{1}{3}\ |C_{km}|$$

---

3 We use notation S in Lemma 1 and R in Lemma 2 instead of P to distinguish between different choices of the parameters in (3.43).

Proof: By (3.49) and (3.47) we have

$$i(D\xi^* - D^*\xi) - H$$

(3.52) $$= \sin(\alpha-\beta)\,(\mathcal{R}(K) - (E\,D^* + E^*\,D))$$

$$- \cos(\alpha-\beta)\,(I_m(K) + i(BD^* - B^*D))$$

$$= 2(\sin(\alpha-\beta)\mathcal{R}(C_{km}) - \cos(\alpha-\beta)\,I_m(C_{km}))$$

Thus we need to establish the required inequality (3.51) for

(3.53) $$\tanh(y) = 1/2\,\frac{i(D\xi^*-D^*\xi)-H}{G+2(|\xi|^2+|D|^2)}$$

(we note that $|\tanh(y)| \leq 1/2$ since $|H| \leq G$ and

$$|i(D\xi^* - D^*\xi)| \leq |\xi|^2+|D|^2)$$

From (3.48) we have

$$\Delta N^2(A) = -\,H\sinh(2y)$$

$$+\ (\sinh(2y))\,(\cosh(2y))\,i(D\xi^*-D^*\xi)$$

$$-\ G(\cosh(2y)-1) - 1/2(\cosh(4y)-1)\,(|D|^2+|\xi|^2)$$

$$\geq \sinh(2y)\{i(D\xi^*-D^*\xi)\cosh(2y) - H - 1/2(G+2(|D|^2+|\xi|^2))\sinh(2y)\}$$

since $\cosh(2y) - 1 \leq 1/4\,(\cosh(4y)-1)$

$$= 1/2\,\sinh^2(2y)$$

Using $\sinh(2y) = 2\tanh(y)\cosh^2(y)$ and the definition of $\tanh(y)$ in (3.53)

we get $$\Delta N^2(A) \geq (2\tanh(y))(\cosh^2(2y))\{i(D\xi^* - D^*\xi)\cosh(2y)$$

$$-\ H - 1/2(i(D\xi^* - D^*\xi) - H)\cosh^2(y)\}$$

Letting $2r = i(D\xi^* - D^*\xi)$

$$\Delta N^2(A) \geq \frac{(2r - H)}{G+2(|\xi|^2+|D|^2)}\,\cosh^2(y)\,\{2r\cosh(2y)$$

(3.54) $$-\ H - 1/4(\cosh(2y)+1)(2r - H)\}$$

$$= \frac{(2r-H)}{G+2\,(|\xi|^2+|D|^2)}\,\cosh^2(y)\,\{2r[1/2+3/4\,(\cosh(2y)-1)]$$

$$-\ H\,[1/2-1/4\,(\cosh(2y)-1)]\,\}$$

$$= \frac{1/2\ (2r-H)^2}{G+2(|\xi|^2+|D|^2)}\ \cosh^2(y)\ \left\{1+\sinh^2(y)\ \frac{(6r+H)}{(2r-H)}\right\}$$

$$\geq \frac{1/2\ (2r-H)^2}{G+2(|\xi|^2+|D|^2)}\ \cosh^2(y)\ \left\{1-\sinh^2(y)\left|\ \frac{6r+H}{2r-H}\right|\right\}$$

We have

$$\sinh^2(y) = \frac{\tanh^2(y)}{1-\tanh^2(y)} \qquad = \frac{1/4\ (2r-H)^2}{[G+2(|\xi|^2+|D|^2)]^2 - 1/4(2r-H)^2}$$

and hence

$$(3.55) \qquad \sinh^2(y)\left|\ \frac{6r+h}{2r-H}\ \right|$$

$$= \frac{1/4|2r-H|\ |6r+H|}{[G+2(|\xi|^2+|D|^2)]^2 - 1/4(2r-H)^2}$$

Now since $|2r| = |i(D\xi^* - D^*\xi)| \leq |D|^2 + |\xi|^2$

and $|H| \leq G$, we have

$$|2r-H| \leq |D|^2+|\xi|^2+G$$

and

$$|6r+H| \leq 3(|D|^2+|\xi|^2)+G$$

Hence the numerator of (3.55)

$$1/4\ |r-H|\ |6r\ H|$$

$$\leq 1/4\ [G+(|D|^2+|\xi|^2)]\ [G+3(|D|^2+|\xi|^2)]$$

$$\leq 1/4\ [G+2(|\xi|^2+|D|^2)]^2$$

and the denominator of (3.55) are

$$\geq 3/4\ [G+2(|\xi|^2+|D|^2)]^2$$

From these last two inequalities and from (3.55) we have

$$\sinh^2(y)\ \ |(6r+H)/(2r-H)| < 1/3$$

Using this expression in (3.54) we obtain

$$(3.56) \qquad \Delta N^2(A) \geq 1/3\ \frac{(2r-H)^2}{G+2(|\xi|^2+|D|^2)}$$

$$= \frac{4}{3}\ \frac{|c_{km}|^2}{G+2(|\xi|^2+|D|^2)}$$

by (3.50), (3.52) and the definition of 2r. Since the denominator is less than $4\ N^2(A)$ we have

$$\Delta N^2(A) \geq 1/3 \ \frac{|c_{km}|^2}{N^2(A)} \geq 1/3\ |c_{km}|^2$$

Lemma 2. Let R be a real rotation, obtained from (3.43) by setting $\alpha=\beta=\gamma=0$. Let $A^1 = R^{-1}AR$ and

$$\tan(2x) = -\frac{c_{kk} - c_{mm}}{2\mathcal{R}(c_{km})} \qquad (3.57)$$

Then $c^1_{kk} - c^1_{mm} = 0$

$$I_m(c^1_{km}) = I_m(c_{km}) \text{ and}$$

$$2\mathcal{R}(c^1_{km}) = (2\mathcal{R}(c_{km})\cos(2x) - (c_{kk} - c_{mm})\sin(2x)$$

(we note that c_{ii} is real since $C = A\ A^* - A^*\ A$ is Hermitian).

Proof: This lemma follows immediately upon computing c^1_{km} and $c^1_{kk} - c^1_{mm} = (c_{kk} - c_{mm})\cos(2x) + (2\mathcal{R}(c_{km})\sin(2x)$.

Lemma 3. Let (k,m) be chosen so that $4|c_{km}|^2 + (c_{kk} - c_{mm})^2$

$$\geq \frac{2}{n(n-1)} \sum_{i<j} \{4|c_{ij}|^2+(c_{ii} - c_{jj})^2\} \qquad (3.58)$$

Then $4|c_{km}|^2 + (c_{kk} - c_{mm})^2 \geq \frac{4}{n(n-1)} N^2(c)$

Proof: We have $\sum_{i<j} (c_{ii} - c_{jj})^2 = (n-1) \sum_i c^2_{ii} - 2 \sum_{i<j} c_{ii}c_{jj}$

But since trace $(A\ A^* - A^*\ A) = 0$

e.g.

$$\sum_i c_{ii} = 0$$

$$(\sum_i c_{ii})^2 = \sum_i c^2_{ii} + 2 \sum_{i<j} c_{ii}c_{jj} = 0$$

Thus

$$\sum_{i<j} (c_{ii} - c_{jj})^2 = n \sum_i c^2_{ii} \geq 2 \sum_i c^2_{ii},\ (n\geq 2)$$

Using (3.58) we have $4|c_{km}|^2 + (c_{kk} - c_{mm})^2$

$$\geq \frac{2}{n(n-1)} \sum_{i<j} \{4|c_{ij}|^2 + 2c_{ii}^2\} = \frac{2}{n(n-1)} N^2(C)$$

Theorem. Let $A_o = A$ with $N^2(A)<1$. Let $A_{i+1} = O_i^{-1} A O_i$ where $O_i(k_i, m_i) = R_i S_i$ and the pair (k_i, m_i) is chosen so that $4|c^{(i)}_{k_i m_i}|^2 + (c^{(i)}_{k_i k_i} - c^{(i)}_{m_i m_i})^2$ is at least average in magnitude of all such quantities. The transformation R_i and S_i are each of the form (3.43) with parameters defined as follows: [2]

$$R: \quad \tan(2x)_R = - \frac{c_{kk} - c_{mm}}{2\mathcal{R}(c_{km})}$$

$$\alpha_R = \beta_R = y_R = 0$$

$$S: \quad \tan(\alpha_S - \beta_S) = -1/2 \frac{(\mathcal{R}(c_{km})\cos(2x)_R - (c_{kk} - c_{mm})\sin 2(x)_R}{I_m(c_{km})}$$

$$\tanh(y)_S = \frac{1/2\sin(\alpha_S-\beta_S)\{(\mathcal{R}(c_{km})\cos(2x)_R-(c_{kk}-c_{mm})\sin(2x)_R - \cos(\alpha_S-\beta_S)I_m(c_{km})}{G_{km} + 2(|\xi^1|^2 + |D^1_{km}|^2)}$$

where

$$\xi^1 = (B_{km}\cos(2x)_R - D\sin(2x)_R)\cos(\alpha_S-\beta_S) - iE_{km}\sin(\alpha_S-\beta_S),$$

$D^1_{km} = D_{km}\cos(2x)_R + B_{km}\sin(2x)_R$; β_S aand x_R are arbitrary. Then $\lim_{i\to\infty}\left(N^2(c_i)\right) = 0$; i. e., for i sufficiently large, A_i is arbitrarily close to being normal.

Proof: Let $A^1_{i+1} = R_i^{-1} A_i R_i$

$$A_{i+1} = S_i^{-1} A_i S_i$$

---

2 We omit subscript or subscripts i.

Then by Lemma 2

$$2\mathcal{R}(c^1_{km}) = \pm [4\mathcal{R}(c^2_{km}) + (c_{kk} - c_{mm})^2]^{1/2},$$
$$I_m(c^1_{km}) = I_m(c_{km})$$

we have

$$\tan(\alpha_S - \beta_S) = -\frac{\mathcal{R}(c^1_{km})}{I_m(c^1_{km})}$$

and

$$\tanh(y)_S = \frac{S_{in}(\alpha_S-\beta_S)\mathcal{R}(c^1_{km})-\cos(\alpha_S-\beta_S)I_m(c^1_{km})}{G_{km} + 2(|\xi^1_{km}|^2 + |D^1_{km}|^2)}$$

By Lemma 1 and the invariance of N^2 under rotations

$$\Delta N^2(A_i) = \Delta N^2(A^1_i) \geq 1/3|c^{(1)}_{km}|^2$$
$$= \frac{1}{12} [4|c^{(i)}_{km}|^2 + (c^{(i)}_{kk} - c^{(i)}_{mm})^2]$$

Hence $\Delta N^2(A_i) \geq 1/3(1/n(n-1))N^2(C_i)$ by Lemma 3. But since $N^2(A_i)$ is a decreasing monotone function bounded below by $\sum_j |\lambda_j|^2$

$\Delta N^2(A_i) \to 0$ as $i \to \infty$ and so does $N^2(C_i)$.

SECTION 4

AN ERROR ANALYSIS OF SOME SPECIAL NUMERICAL METHODS

Preliminaries: Considerable attention has been given to the effect of rounding errors on the numerical solutions of problems in linear algebra. Some fundamental contributions in this field have been made by J. H. Wilkinson ([3], [11]). In this paper we shall give a short account of Wilkinson's work, including some applications pertaining to the eigenvalue problem. In particular, we shall give an error analysis of the Gaussian elimination and of Householder's method. The former method is of importance in the reduction of an unsymmetric matrix to the Hessenberg form. (Then in order to utilize this special form in the calculation of eigenvalues we may apply the QR method to it).

The basic idea behind Wilkinson's work is simple. He set out to show that the computed results of a problem may be obtained by exact calculations from a perturbed problem. Then (upper) bounds are obtained for the various perturbations. This type of analysis Wilkinson calls "backward" analysis in contradistinction to "forward" analysis.*

---

* We calculate a mathematical expression given by $y=f(x_1, \ldots, x_n)$. Then the "backward" analysis shows that the computed y does not satisfy the equation above but another equation of the form $Y=f(x_1+e_1, \ldots, x_n+e_n)$ where the e_i, $(i=1, \ldots, n)$ are perturbations for which bounds are usually given. "Forward" analysis attempts to trace the forward propagation of individual rounding errors and then compares the computed answer to that of the exact answer.

It was found that in many applications a backward analysis is much easier to perform than a forward analysis. One way in which backward analysis can be applied is to compare two numerical methods. If a method, A, say, has smaller bounds for the e_i than another method, B, then we should select A as the better method provided all other factors are equal.

VECTOR AND MATRIX NORMS. The norm of a vector x is denoted by $||x||$ and has the following properties:

(4.1) $||x|| > 0$

$||kx|| - |k|\ ||x||$; k is a complex constant

$||x + y|| \leqq ||x|| + ||y||$

From (4.1)

$$||x + y|| \leq ||x|| + ||y||$$

let $y = z - x$

$$\rightarrow ||x + z - x|| \leq ||x|| + ||z - x||$$

$$\rightarrow ||z|| \leq ||x|| + ||z - x||$$

$$\rightarrow ||z - x|| \geq ||z|| - ||x||$$

if we let $z = x$ and $x = y$ we have

(4.2) $||x - y|| \geq ||x|| - ||y||$

The three vector norms most commonly used are special cases of the HOELDER norm

$$||x||_k = \left[\sum_{i=1}^{n} |x_i|^k\right]^{1/k} ;\ k\geq 1$$

They are obtained by letting $k = 1$, 2 and ∞. Thus we get for

$k = 1$: $||x||_1 = |x_1| + |x_2| + \ldots + |x_n|$

$k = 2$: $||x||_2 = \{|x_1|^2 + |x_2|^2 + \ldots + |x_n|^2\}^{1/2}$

$k = \infty$: $||x||_\infty = \max |x_i|$; $i=1, \ldots, n$

We note that when $k = 2$ our norm is the Euclidian length of the vector x. This is frequently written as $||x||_E$. In a similar way we denote the norm of a matrix A by $||A||$. The matrix norm satisfies properties (4.1) and (4.2) with vectors x and y replaced by matrices A and B. In addition it satisfies the multiplicative property (4.3).

$$(4.3) \qquad ||A|| < ||A||.||B||$$

Since vectors and matrices often appear together we have to set up a relation between them.

$$(4.4) \qquad ||Ax|| \leq ||A||.||x||$$

Now we can show* that the following relations hold true:

$$(4.5) \qquad ||A||_1 = \max_j \sum_{i=1}^{n} |a_{ij}|$$

$$||A||_2 = (\text{max eigenvalue of } A^H A)^{1/2}$$

$$||A||_\infty = \max_i \sum_{j=1}^{n} |a_{ij}|$$

where A^H denotes the Hermitian conjugate of A and (a_{ij}) is the $(i,j)^{th}$ element of A. For error analysis a norm which is consistent with $||A||_2$ but easier to compute is the Schur norm,

$$||A||_E = (\sum_i \sum_j |a_{ij}|^2)^{1/2}$$

ARITHMETIC OPERATIONS. We shall now briefly discuss some of the basic floating-point arithmetic operations. Any binary number x can be represented in the form

$$x = a.2^b$$

where a is called the mantissa and b the exponent. The exponent is a negative or positive integer while the mantissa is a fraction in binary machine where all numbers are normalized.

---

* Proof given in Appendix.

(4.6) $$1/2 \leqslant |a| < 1$$

We shall let t denote the number of binary digits allocated to the mantissa. In order to differentiate between (a) mathematical relations and (b) computation equations we use the equal sign for (a) and the equivalence sign ($\equiv$) for (b). Following Wilkinson's notation we use fl (x.y) to denote the computed result of multiplying together two floating point numbers.

If the mantissa is normalized to lie in the range (4.6) we get the following computational equations according to Wilkinson

$$fl\ (x \pm y) \equiv (x \mp y)\ (1+\varepsilon_1)$$
$$fl\ (x.y) \equiv (x.y)\ (1+\varepsilon_2)$$
$$fl\ (x/y) \equiv (x/y)\ (1+\varepsilon_3)$$

where $|\varepsilon_i| \leqslant 2^{-t}$.

For extended floating point operations we get:*

$$fl\ (x_1.x_2 \ldots x_n) \equiv x_1 x_2 \ldots x_n (1+\varepsilon)$$
$$fl\ (x_1 + \ldots + x_n) \equiv x_1(1+\varepsilon_1) + \ldots + x_n(1+\varepsilon_n)$$
$$fl\ (x_1 y_1 + x_2 y_2 + \ldots + x_n y_n) \equiv x_1 y_1 (1+\eta_1) + \ldots + x_n y_n (1+\eta_n)$$

where:

$$(1 - 2^{-t})^{n-1} \leqslant 1+\varepsilon \leqslant (1+2^{-t})^{n-1}$$
$$(1 - 2^{-t})^{n-1} \leqslant 1+\varepsilon_1 \leqslant (1+2^{-t})n-1$$
$$(1 - 2^{-t})^{n+1-r} \leqslant 1+\varepsilon_r \leqslant (1+2^{-t})^{n+1-r};\ (r \geqslant 2)$$
$$(1 - 2^{-t})^{n} \leqslant 1+\eta_1 \leqslant (1+2^{-t})^{n}$$
$$(1 - 2^{-t})^{n-r+2} \leqslant 1+\eta_r \leqslant (1+2^{-t})^{n-r+2};\ (r \geqslant 2)$$

---

* Proof given in appendix.

<u>ERROR ANALYSIS OF GAUSSIAN ELIMINATION.</u> Gaussian elimination has already been discussed in Section (3) of this paper. For this reason we shall refrain from going into details regarding the method itself in this section. We denote the original set of equations by

$$A^{(1)}x = b^{(1)}$$

then (n-1) equivalent sets of equations

$$A^{(r)}x = b^{(r)}, \ (r = 2; \ldots, n)$$

are produced. $A^{(n)}$ of the final equation is in an upper triangular form. The matrix $A^{(r+1)}$ is obtained from $A^{(r)}$ by subtracting a multiple m_{ir} of the r^{th} row from the i^{th} row for values of i from (r+1) to n. The m_{ir}'s are defined by $m_{ir} = a^{(r)}_{ir}/a^{(r)}_{rr}$. Now we consider two cases (i and ii) depending on whether $i \leq j$ or $i > j$.

Case (i). $(i \leq j)$: The element is changed in each transformation until we reach $A^{(i)}$ after which it remains constant. Thus we get

$$\begin{aligned} a^{(2)}_{ij} &\equiv a^{(1)}_{ij} - m_{i1}a^{(1)}_{1j} + \varepsilon^{(2)}_{ij} \\ a^{(3)}_{ij} &\equiv a^{(2)}_{ij} - m_{i2}a^{(2)}_{2j} + \varepsilon^{(3)}_{ij} \\ &- \\ a^{(i)}_{ij} &\equiv a^{(i-1)}_{ij} - m_{i,i-1}a^{(i-1)}_{i-1,j} + \varepsilon^{(i)}_{ij} \end{aligned} \tag{4.7}$$

where the $a^{(k)}_{ij}$ and m_{ik} are computed values and $\varepsilon^{(k)}_{ij}$ is the difference between the accepted and the exact values of $a^{(k)}_{ij}$. Summing (4.7) we get

$$a^{(i)}_{ij} = a^{(1)}_{ij} - m_{i1}a^{(1)}_{1j} - m_{i2}a^{(2)}_{2j} - \cdots - m_{i,i-1}a^{(i-1)}_{i-1,j} + e_{ij}$$

where

$$e_{ij} = \varepsilon^{(1)}_{ij} + \ldots + \varepsilon^{(i)}_{ij} \tag{4.8}$$

Case (ii). $(i > j)$: As in case i the element is changed until $A^{(i)}$ is obtained. $a^{(i)}_{ij}$ is then used to compute m_{ij}. $a^{(j+1)}_{ij}$ up to and including $a^{(n)}_{ij}$ are considered to be exactly zero. The computed m_{ij} satisfy

$$m_{ij} = (a^{(j)}_{ij}/a^{(j)}_{jj}) + \eta_{ij}$$

where η_{ij} is the rounding error incurred by the division. The equations in this case are:

$$
\begin{aligned}
(4.10) \qquad & a^{(2)}_{ij} \equiv a^{(1)}_{ij} - m_{i1}a^{(1)}_{ij} + \varepsilon^{(2)}_{ij} \\
& a^{(3)}_{ij} \equiv a^{(2)}_{ij} - m_{i2}a^{(2)}_{2j} + \varepsilon^{(3)}_{ij} \\
& - \\
& a^{(j)}_{ij} \equiv a^{(j-1)}_{ij} - m_{i,j-1}a^{(j-1)}_{j-1,j} + \varepsilon^{(j)}_{ij} \\
& 0 \equiv a^{(j)}_{ij} - m_{ij}a^{(j)}_{jj} + \varepsilon^{(j+1)}_{ij} \\
& \varepsilon^{(j+1)}_{ij} = a^{(j)}_{jj}\eta_{ij}
\end{aligned}
$$

again summing we get

$$0 \equiv a^{(1)}_{ij} - m_{i1}a^{(1)}_{ij} - m_{i2}a^{(2)}_{2j} - \dots - m_{ij}a^{(j)}_{jj} + e_{ij}$$

where

$$(4.9) \qquad e_{ij} = \varepsilon^{(2)}_{ij} + \varepsilon^{(3)}_{ij} + \dots + \varepsilon^{(j+1)}_{ij}$$

Now we note that the two sets of equations (4.7) and (4.10) are equivalent to the single matrix equation

$$LU \equiv A^{(1)} + E$$

where

$$L = \begin{bmatrix} 1 & 0 & & \dots & 0 \\ m_{21} & 1 & 0 & \dots & 0 \\ " & & & & \\ " & & & & \\ " & & & & 0 \\ m_{n1} & m_{n2} & & & 1 \end{bmatrix} ; \quad U = \begin{bmatrix} a^{(1)}_{11} & a^{(1)}_{12} & \dots & a^{(1)}_{1n} \\ 0 & a^{(2)}_{22} & \dots & a^{(2)}_{2n} \\ " & & & \\ " & & & \\ " & & & \\ 0 & 0 & & a^{(n)}_{nn} \end{bmatrix}$$

E is defined by relations (4.8) and (4.9).

Now we shall attempt to establish bounds for E. In floating point arithmetic the computed $a^{(k)}_{ij}$ is defined by

$$a^{(k)}_{ij} \equiv fl\,(a^{(k-1)}_{ij} - m_{i,k-1}\, a^{(k-1)}_{k-1,j})$$
$$= [a^{(k-1)}_{ij} - m_{i,k-1}\, a^{(k-1)}_{k-1,j}\,(1+\varepsilon_1)]\,(1+\varepsilon_2)$$

Thus the difference between the exact and computed solutions are

$$\varepsilon^{(k)}_{ij} = a^{(k)}_{ij} - (a^{(k-1)}_{ij} - m_{i,k-1}\, a^{(k-1)}_{k-1,j})$$
$$= a^{(k)}_{ij} - \left(\frac{a^{(k)}_{ij}}{1+\varepsilon_2} + m_{i,k-1}\, a^{(k-1)}_{k-1,j}\varepsilon_1\right)$$

$$(4.11)\quad \varepsilon^{(k)}_{ij} = \frac{a^{(k)}_{ij}\,\varepsilon_2}{1+\varepsilon_2} - m_{i,k-1}\, a^{(k-1)}_{k-1,j}\,\varepsilon_1$$

Thus in order to get satisfactory bounds for $\varepsilon^{(k)}_{ij}$, we shall need reasonable bounds for m_{ik} and $a^{(k)}_{ij}$. In practice we shall attempt to get

$$|m_{ij}| \leq 1$$

This can be done by either partial or complete pivoting. Let us consider any kind of pivoting at the moment. Then we shall denote the maximum element of any $A^{(r)}$ by g. By taking into account scaling, let us assume that

$$|a^{(i)}_{ij}| \leq 1$$

From (4.11) we get

$$(4.12)\qquad |\varepsilon^{(k)}_{ij}| \leq \frac{g\,2^{-t}}{1-2^{-t}} + g\,2^{-t} < (2.01)g\,2^{-t}$$

This applies to all $\varepsilon^{(k)}_{ij}$ except $\varepsilon^{(j+1)}_{ij}$ for $i>j$. For these we use the relations established earlier, namely

$$m_{ij} = (a^{(j)}_{ij}/a^{(j)}_{jj}) + \eta_{ij}$$

Thus we have

$$m_{ij} \equiv fl\ (a^{(j)}_{ij}/a^{(j)}_{jj})$$
$$= a^{(j)}_{ij}/a^{(j)}_{jj}(1+\varepsilon)$$
$$\eta_{ij} = (a^{(j)}_{ij}/a^{(j)}_{jj})\varepsilon$$

Thus

$$|\varepsilon^{(j+1)}_{ij}| = |a^{(j)}_{jj}(a^{(j)}_{ij}/a^{(j)}_{jj})\varepsilon|$$
$$= a^{(j)}_{ij}\varepsilon$$
$$\leq g\ 2^{-t} \tag{4.13}$$
$$< (2.01)g\ 2^{-t}$$

Now combining (4.2), (4.13), (4.8) and (4.9) we get

$$|E| \leq (2.01)g\ 2^{-t} \begin{bmatrix} 0 & 0 & \dots & & & 0 \\ 1 & 1 & \dots & & & 1 \\ 1 & 2 & 2 & \dots & & 2 \\ 1 & 2 & 3 & \dots & & 3 \\ - & - & - & - & - & - \\ 1 & 2 & 3 & 4 & \dots (n-1) & (n-1) \end{bmatrix} \tag{4.14}$$

According to Wilkinson, g is usually of order unity if we use pivoting.

AN ERROR ANALYSIS OF HOUSEHOLDER'S METHOD FOR THE SYMMETRIC EIGENVALUE PROBLEM

Preliminaries: Again we describe in detail only the transformation from A_1 to A_2. We partition A_1 as follows:

$$A = A_1 = \begin{bmatrix} a_{11} & c^T \\ c & C \end{bmatrix}$$

where $c^T = (a_{12}, a_{13}, \dots, a_{1N})$

and

$$C = \begin{bmatrix} a_{22} & \cdots & a_{2N} \\ '' & & \\ '' & & \\ '' & & \\ a_{N2} & \cdots & a_{NN} \end{bmatrix}$$

A_2 is then obtained by the following computation:

$$s = (c^T c)^{1/2}$$

$$x_2 = [.5+.5(|a_{12}|/s)]^{1/2}$$

$$\eta = (2x_2)s$$

$$w = \begin{cases} 1 \text{ if } a_{12} \geq 0 \\ -1 \text{ if } a_{12} < 0 \end{cases}$$

$$x_i = wa_{1i}/\eta; \ (i = 3, \ldots, N)$$

$$p = Cu, \ u^T = (x_2, \ldots, x_N)$$

$$\gamma = u^T p$$

$$q = p - \gamma\mu$$

$$B = C - 2(q\mu^T + q\mu^T)$$

$$b^T = (-ws, 0, \ldots, 0)$$

$$A_2 = \begin{bmatrix} a_{11} & b^T \\ b & B \end{bmatrix}$$

In this section we do not assume anything about the nature of floating point arithmetic other than the existence of a number m such that

$$(4.15) \qquad |fl(xy) - xy| \leq m|xy|$$

$$(4.16) \qquad |fl(x/y) - x/y| \leq m|x|/|y|$$

$$(4.17) \qquad |fl(x \pm y) - (x \pm y)/ \leq m(|x|+|y|)$$

$$(4.18) \qquad |fl(x^{1/2}) - x^{1/2}| \leq 2mx^{1/2}$$

The number m will depend on the word length of the computer, whether the results of the computer operations are rounded or truncated and possibly other factors.

For the inner product of vectors we assume that

$$(4.19)\qquad |fl(v^Tw) - v^Tw| \leq m^{(n)}_p ||v||.||w||$$

for all vectors v and w of length n. In order to simplify the handling of error terms we shall assume that

$$(4.20)\qquad \begin{aligned} m &\leq 10^{-6} \\ m^{(n)}_p &\leq 10^{-4} \ (n=1,2, \ldots, N-1) \end{aligned}$$

Then we shall need bounds for the errors in various matrix computations. According to Ortega [12], we have

$$(4.21)\qquad \begin{aligned} ||fl(v\underline{+}w) - (v\underline{+}w)|| &= |||fl(v\underline{+}w) - (v\underline{+}w)||| \\ &\leq m(||v|| + ||w||) \end{aligned}$$

$$(4.22)\qquad ||fl(\alpha v) - \alpha v|| = |||fl(\alpha v) - \alpha v||| \leq m|\alpha|||v||$$

$$(4.23)\qquad ||fl(vw^T) - vw^T|| \leq |||fl(vw^T) - vw^T||| \leq m||v||.||w||$$

$$(4.24)\qquad ||fl(E\underline{+}F) - (E\underline{+}F)|| \leq |||fl(E\underline{+}F) - (E\underline{+}F)||| \leq m(|||E||| + |||F|||)$$

$$(4.25)\qquad ||fl(\alpha E) - \alpha E|| \leq |||fl(\alpha E) - \alpha E||| \leq m|\alpha||||E|||$$

$$(4.26)\qquad ||fl(Ev) - Ev|| \leq |||fl(Ev) - Ev||| \leq m^{(n)}_p n^{1/2}||E||.||v||$$

where α is a scalar, v and w are column vectors and E and F are nxn matrices.

ERROR BOUNDS FOR THE COMPUTED EIGENVALUES. Following our exposition of the Householder method in section 3 we base our analysis on the following observations. Let $\overline{A}_1 = A_1, \overline{A}_2, \ldots \overline{A}_{N-1}$ be the sequence of matrices actually machine computed by the Householder algorithm. Let $A^{(i+1)}$ be the matrix produced by exact arithmetic when the i^{th} step of the reduction is applied to $\overline{A}_i$ and let $X_i = A^{(i+1)} - \overline{A}_{i+1}$, $(i=1,2,\ldots, N-2)$. Then if $\lambda^{(i)}_1 \geq \ldots \geq \lambda^{(i)}_N$ and $\overline{\lambda}^{(i)}_1 \geq \ldots \geq \overline{\lambda}^{(i)}_N$ are the eigenvalues of $A^{(i)}$ and $\overline{A}_i$ respectively we have by the perturbation theorem that follows from the Courant-Fischer minimax representation [13]

$$\max_j \left|\overline{\lambda}^{(i+1)}_j - \lambda^{(i+1)}_j\right| \leq ||X_i||, \quad (i=1, \ldots, N-2)$$

Therefore, since $\overline{A}_i$ and $A^{(i+1)}$ are similar, we have

$$(4.27) \qquad \max_j \left|\overline{\lambda}^{(i+1)}_j - \overline{\lambda}^{(i)}_j\right| \leq ||X_i||, \quad (i=1, \ldots, N-2)$$

If we now let

$$\varepsilon = \max_j \left|\overline{\lambda}^{(1)}_j - \overline{\lambda}^{(N-1)}_j\right|$$

then ε is the maximum error in the eigenvalues of the computed tridiagonal matrix $\overline{A}_{N-1}$ and we have by (4.27)

$$(4.28) \quad \varepsilon \leq \max_j \left|\overline{\lambda}^{(1)}_j - \overline{\lambda}^{(2)}_j\right| + \max_j \left|\overline{\lambda}^{(2)}_j - \overline{\lambda}^{(3)}_j\right| + \ldots + \max_j \left|\overline{\lambda}^{(N-2)}_j - \overline{\lambda}^{(N-1)}_j\right|$$
$$\leq \sum_{i=1}^{N-2} ||X_i||$$

From (4.28) it is clear now that our objective is to obtain bounds for the $||X_i||$.

We begin by considering $X_1 = A^{(2)} - \overline{A}_2 = A_2 - \overline{A}_2$ and assume that the computation of $\overline{A}_2$ is carried out according to (3.10) in Section(3). We let barred letters denote the computed intermediate quantities. Thus if we let $\nabla = c^T c$, we have, from (4.19)

$$(4.29) \quad |\nabla - \overline{\nabla}| = |c^T c - fl(c^T c)| \leq m^{(N-1)}_n ||c||^2 = m_n \nabla$$

(here for reasons of simplicity, we shall omit the superscript of m).

Consequently from (4.18) we get

$$(4.30) \quad |s-\overline{s}| = |\nabla^{1/2} - \overline{\nabla}^{1/2} + \overline{\nabla}^{1/2} - fl(\overline{\nabla}^{1/2})| \leq |\nabla^{1/2} - \overline{\nabla}^{1/2}| + 2m\overline{\nabla}^{1/2}$$

and since from (4.29) we get

$$\nabla(1-m_p) \leq \bar{\nabla} \leq \nabla(1+m_p)$$

then by (4.20) we have

$$(4.31)\qquad \begin{aligned} \nabla^{1/2}(1-1/2m_p-1/2m_p^{\,2}) &\leq \nabla^{1/2}(1-m_p)^{1/2} \\ &\leq \bar{\nabla}^{1/2} \\ &\leq \nabla^{1/2}(1+m_p)^{1/2} \\ &\leq \nabla^{1/2}(1+1/2m_p) \end{aligned}$$

Therefore, we get

$$(4.32)\qquad \begin{aligned} \left|\nabla^{1/2} - \bar{\nabla}^{1/2}\right| &\leq (1/2m_p + 1/2m_p^{\,2})\nabla^{1/2} \\ &\leq .50005m_p\,\nabla^{1/2} \end{aligned}$$

Now combining (4.30), (4.31) and (4.32) we obtain

$$(4.33)\qquad \begin{aligned} |s-\bar{s}| &\leq .50005m_p\nabla^{1/2} + 2m(1+1/2m_p)\nabla^{1/2} \\ &\leq (.50005m_p + 2.00005m)s. \end{aligned}$$

We next need a bound for $|x_2 - \bar{x}_2|$. We let

$$\alpha = |a_{12}|/s,\ \beta = .5\alpha$$

and $\mu = .5 + \beta$ and obtain by straight forward computation, using (4.15), (4.15), (4.17), (4.18), (4.33) and the inequality

$$1/(1-.50005m_p - 2.00005m) \leq 1+.75m_p+3m$$

$$|\alpha - \bar{\alpha}| \leq (.50009m_p + 3.0002m)\alpha$$

$$|\beta - \bar{\beta}| \leq (.50009m_p + 4.0003m)\beta$$

$$|\mu - \bar{\mu}| \leq (5.0009m_p + 4.0004m)\mu$$

and

$$|x_2 - \bar{x}_2| \leq |\mu^{1/2} - \bar{\mu}^{1/2}| + 2m\bar{\mu}^{1/2}$$

By the same analysis that led to (4.32) we obtain here

$$|\mu^{1/2} - \bar{\mu}^{1/2}| \leq (2.501m_p + 2.0003m)\mu^{1/2}$$

and therefore

(4.34) $$|x_2 - \bar{x}_2| \leqslant (2.501m_p + 4.0005m)x_2$$

To obtain bounds for the errors in the other $\bar{x}_i$ we let

$$\xi = 2x_2 \text{ and } \eta = \xi s$$

Then using (4.15), (4.33) and (4.34) we obtain

$$|\xi - \bar{\xi}| \leqslant (.2501m_p + 5.0006m)\xi$$

and $|\eta - \bar{\eta}| \leqslant (.7502m_p + 8.0008m)\eta$

Therefore we get, using (4.16) and the inequality

$$1/(1-.7502m_p - 8.0008m) < 1+1.37m_p + 12m$$

(4.35) $$|x_i - \bar{x}_i| \leqslant (.7504m_p + 9.0009m)\ |x_i|$$

$$(i = 3, \ldots, N).$$

Thus recalling that $u^T = (x_2, \ldots, x_N)$ and letting $\bar{u}^T = (\bar{x}_2, \ldots, \bar{x}_N)$, we have

(4.36) $$||u - \bar{u}|| \leqslant (.7504m_p + 9.0009m)||u|| = (.7504m_p + 9.0009m)$$

since $||u|| = 1$.

Now we continue with the matrix portion of the calculation and since from (4.36)

(4.37) $$||\bar{u}|| \leqslant 1 + .7504m_p + 9.0009m$$

we have from (4.26), (4.36) and (4.37)

(4.38) $$\begin{aligned} ||p - \bar{p}|| &= ||Cu - C\bar{u} + C\bar{u} - fl(C\bar{u})|| \\ &\leqslant ||C||.||u - \bar{u}|| + m_p(N-1)^{1/2}\ ||C||.||\bar{u}|| \\ &\leqslant \{(.7504m_p + 9.0009m) + m_p(N-1)^{1/2}(1 + .7504m_p + 9.0009m)\}\ ||C|| \\ &\leqslant \{1.0001\ (N-1)^{1/2}m_p + .7504m_p + 9.0009m\}\ ||A_1|| \end{aligned}$$

since $||C|| \leq ||A_1||$. Now $||p|| \leq ||C||\ ||u|| \leq ||A_1||$ and hence from (4.19), (4.36) and (4.38) we obtain after a calculation similar to that of (4.38)

(4.39) $|\gamma - \bar{\gamma}| \leq \{1.0003\ (N-1)^{1/2} m_p + 2.5011 m_p + 18001m\}\ ||A_1||.$

We next need a bound for $||q - \bar{q}||$ and as an intermediate step we let $r = \gamma u$. Then since $|\gamma| \leq ||p||\ ||u|| \leq ||A_1||$ we obtain, using (4.22), (4.36) and (4.39)

$$|r - \bar{r}| \leq \{1.0004\ (N-1)^{1/2} m_p + 3.2518 m_p + 28.003m\}\ ||A_1||$$

Using this result and the fact that $||r|| \leq |\gamma|\ ||u|| \leq ||A_1||$ we then obtain from (4.21) and (4.38)

(4.40)

$$||q - \bar{q}|| \leq \{2.0006\ (N-1)^{1/2} m_p + 4.0023 m_p + 39.004m\}\ ||A_1||.$$

We now want a bound for $||B - \bar{B}||$ and as a first step we let $G = qu^T$. Then since $||q|| \leq ||p|| + ||\gamma u|| \leq 2||A_1||$ we have from $||\ |E|\ || \leq n^{1/2}||E||$, $||\ |vw^T|\ || = ||v||.||w||$, (4.23), (4.36) and (4.40)

(4.41)

$$\begin{aligned} ||G - \bar{G}|| \leq ||\ |G - \bar{G}|\ || &\leq ||\ |qu^T - \bar{q}u^T| + |\bar{q}u^T - \bar{q}\bar{u}^T| \\ &\qquad + |\bar{q}\bar{u}^T - fl(\bar{q}\bar{u}^T)|\ || \\ &\leq ||q - \bar{q}||.||u|| + ||q||\ ||u - \bar{u}|| + m||\bar{q}||\ ||\bar{u}|| \\ &\leq \{2.0008\ (N-1)^{1/2} m_p + 5.5035 m_p + 59.007m\}\ ||A_1|| \end{aligned}$$

and consequently from $||\ |E|\ || \leq ||\ |E - F| + |F|\ ||$
$\leq ||\ |E - F|\ || + ||\ |F|\ ||$

we get

(4.42) $||\ |\bar{G}|\ || \leq ||\ ||\ |\bar{G}| - |G|| + |G|\ || \leq ||\ |G - \bar{G}|\ || + ||\ |qu^T|\ ||$
$\leq \{2 + 2.0008\ (n-1)^{1/2} m_p + 5.5035 m_p + 59.007m\}\ ||A_1||$

Then, if we let $H = G+G^T$ we obtain from (4.24), (4.41) and (4.42)

$$(4.43)\quad \|H - \bar{H}\| \leqslant \|\,|H-\bar{H}|\,\|$$
$$\leqslant \|\,|G^T-\bar{G}^T| + |G-\bar{G}| + |\bar{G}+\bar{G}^T - fl(\bar{G}+\bar{G}^T)|\,\|$$
$$\leqslant 2\,\|\,|G-\bar{G}|\,\| + m(\|\,|\bar{G}|\,\| + \|\,|\bar{G}^T|\,\|)$$
$$\leqslant \{4.002\,(N-1)^{1/2}m_p + 11.008m_p + 122.015m\}\,\|A_1\|$$

and since $\|\,|H|\,\| \leqslant 2\,\|\,|G|\,\| \leqslant 2\,\|q\|.\|u\| \leqslant 4\,\|A_1\|$ we get

$$(4.44)\quad \|\,|\bar{H}|\,\| < \|\,|H|\,\| + \|\,|H-\bar{H}|\,\|$$
$$\leqslant \{4 + 4.002(N-1)^{1/2}m_p + 11.008m_p + 122.015m\}\,\|A_1\|$$

Next we let $Q = 2H$ and obtain from (4.25), (4.43) and (4.44)

$$(4.45)\quad \|Q-\bar{Q}\| \leqslant \|\,|Q-\bar{Q}|\,\| \leqslant \|\,|2H-2\bar{H}| + |2\bar{H}-fl(2\bar{H})|\,\|$$
$$\leqslant 2\,\|\,|H-\bar{H}|\,\| + 2m\|\,|\bar{H}|\,\|$$
$$\leqslant \{8.005(N-1)^{1/2}m_p + 22.02m_p + 252.04m\}\,\|A_1\|,$$

and

$$(4.46)\quad \|\,|\bar{Q}|\,\| \leqslant \|\,|Q|\,\| + \|\,|Q-\bar{Q}|\,\|$$
$$\leqslant \{8 + 8.005(N-1)^{1/2}m_p + 22.02m_p + 252.04m\}\,\|A_1\|$$

Therefore, since $\|\,|C|\,\| \leqslant (N-1)^{1/2}\,\|C\|$ we have from (4.24), (4.45) and (4.46)

$$(4.47)\quad \|B-\bar{B}\| = \|(C-Q) + (C-\bar{Q}) - fl(C-\bar{Q}) - (C-\bar{Q})\|$$
$$\leqslant \|Q-\bar{Q}\| + m(\|\,|C|\,\| + \|\,|\bar{Q}|\,\|)$$
$$\leqslant \{8.006(N-1)^{1/2}m_p + 22.02m_p + (N-1)^{1/2}m + 260.05m\}\,\|A_1\|$$

Finally

$$A_2 - \bar{A}_2 = \begin{bmatrix} a_{11} & b^T \\ b & B \end{bmatrix} - \begin{bmatrix} a_{11} & \bar{b}^T \\ \bar{b} & \bar{B} \end{bmatrix}$$
$$= \begin{bmatrix} 0 & (b-\bar{b})^T \\ b-\bar{b} & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & B-\bar{B} \end{bmatrix}$$

and since $(b-\bar{b})^T = (-w(s-\bar{s}), 0, \ldots, 0)$ and $s \le ||A_1||$ we have from (4.33) and (4.47)

(4.48) $$||X_1|| = ||A_1-\bar{A}_2|| \le 2^{1/2} |s-\bar{s}| + ||B-\bar{B}||$$
$$\le \{8.006(N-1)^{1/2}m_p^{(N-1)} + 22.75m_p^{(N-1)} + (N-1)^{1/2}m + 262.9m\} ||A_1||$$

In order to obtain bounds for the remaining $||X_i||$ we recall that the i^{th} step of the Householder reduction performs the same operations on the lower principal submatrix of order N-i+1 of $\bar{A}_i$ as the first step performs on A_1. The bound for $||X_i||$ will then have the same form as the bound for $||X_1||$; we need only replace N by N-i+1 and $||A_1||$ by $||\bar{A}_i||$. Therefore, we have

(4.49) $$||X_i|| \le \{8.006(N-i)^{1/2}m_p^{(N-i)} + 22.75m_p^{(N-i)} + (N-i)^{1/2}m + 262.9m\} ||\bar{A}_i|| \quad (i=1, \ldots, N-2).$$

Now let

(4.50) $$\Gamma_i = \{8.006(N-i)^{1/2}m_p^{(N-i)} + 22.75m_p^{(N-i)} + (N-i)^{1/2}m + 262.9m\},$$

and

(4.51) $$\Gamma = \sum_{i=1}^{N-2} \Gamma_i$$

Then since A_2 is orthogonally congruent to A_1, we have

$$||\bar{A}_2|| \le ||A_2 - \bar{A}_2|| + ||A_2||$$
$$= ||X_1|| + ||A_1||$$
$$\le (\Gamma_1 + 1) ||A_1||;$$

similarly, since $A^{(3)}$ is orthogonally congruent to $\bar{A}_2$

$$||\bar{A}_3|| \le ||A^{(3)} - \bar{A}_3|| + ||A^{(3)}|| = ||X_2|| + ||\bar{A}_2||$$
$$\le (\Gamma_2 + 1) ||\bar{A}_2|| \le (\Gamma_2 + 1)(\Gamma_1 + 1) ||A_1||$$

and in general

$$||\bar{A}_i|| \leq (\Gamma_{i-1}+1) \ldots (\Gamma_1+1) \, ||A_1||, \quad (i=2, \ldots N-1)$$

If we put this in (4.49) we get

$$\sum_{i=1}^{N-2} ||X_i|| \leq \{\Gamma_1+\Gamma_2(\Gamma_1+1) + \ldots + \Gamma_{N-2} \prod_{i=1}^{N-3} (\Gamma_i+1)\} \, ||A_1||$$

and since

$$\Gamma_1 + \Gamma_2(\Gamma_1+1) + \ldots + \Gamma_{N-2} \prod_{i=1}^{N-3} (\Gamma_i+1)$$

$$\leq \Gamma + \Gamma^2 + \ldots + \Gamma^{N-2}$$

we have from (4.28) the bound for the maximum error ε:

(4.52) $$\varepsilon \leq \sum_{i=1}^{N-2} ||X_i|| \leq (\Gamma + \ldots + \Gamma^{N-2}) \, ||A_1|| \leq \Gamma \, ||A_1||/(1-\Gamma),$$

provided that $\Gamma<1$

This inequality is our basic result. Once m and the $m_p^{(N-i)}$ are known Γ may be evaluated and (4.52) then gives the bound relative to the spectral norm e.g. $||E|| = \max ||Ev||$. $||v|| = 1$

We now evaluate Γ in terms of m for an important choice of the $m_p^{(N-i)}$.

We assume that

(4.53) $$m_p^{(N-i)} = (N-i+1)m, \quad (i=1, \ldots, N-2)$$

This corresponds to what Ortega calls the STANDARD INNER PRODUCT ROUTINE; that is, the inner product is formed in the usual way with no attempt to accumulate it exactly. We have then from (4.50) and (4.51)

(4.54) $$\Gamma = \sum_{i=1}^{N-2} \Gamma_i \leq \{8.006 \sum_{i=1}^{N-2} (N-i)^{1/2}(N-i+1) + 22.75 \sum_{i=1}^{N-2} (N-i+1) + \sum_{i=1}^{N-2} (N-i)^{1/2} + 262.9 \, (N-2)\} \, m$$

$$\leq (3.21N^{5/2} + 11.4N^2 + 6.01N^{3/2} + 275 \, N-628)m$$

where we have used

$$\sum_{i=1}^{N-2} (N-i)^{1/2} < \int_2^N x^{1/2} dx = \frac{2}{3} (N^{3/2} - 2^{3/2})$$

and

$$\sum_{i=1}^{N-2} (N-i)^{1/2}(N-i+1) < \int_2^N x^{1/2}(x+1) dx$$

$$= \frac{2}{5} N^{5/2} + \frac{2}{3} N^{3/2} - (44.2)^{1/2}/15$$

Therefore putting (4.54) in (4.52) we obtain the following bound

(4.55)

$$\frac{\varepsilon}{||A_1||} \leq \frac{(3.21N^{5/2} + 11.4N^2 + 6.01N^{3/2} + 275N - 628)m}{1 - (3.21N^{5/2} + 11.4N^2 + 6.01N^{3/2} + 275N - 628)m}$$

SECTION 5

THE MATRIX $\begin{bmatrix} A & B \\ B^T & A \end{bmatrix}$

In Section (1) we have referred to the eigenvalue problem of a matrix of type S. In this section we shall consider a similar matrix T, namely

$$T = \begin{bmatrix} A & B \\ B^T & A \end{bmatrix}$$

where T is 2n x 2n and A and B are both (nxn). Problems of this type arise in the field of numerical solutions of elliptic partial differential equations, where the largest eigenvalues have to be found in order to apply Young's overrelaxation method.

In this thesis we investigate a special type of T where A is tridiagonal and

$$B = \begin{bmatrix} 0_1 & 0_2 \\ X & 0_3 \end{bmatrix}$$

with 0_i being zero-submatrices and x the only non-zero element in position (n,1).

By observation we have found that the eigenvalues of T are the same as those of Y together with Z where

(5.1) $$Y = A + C, \; Z = A - C \text{ and}$$

$$C = \begin{bmatrix} 0_2 & 0_1 \\ 0_3 & X \end{bmatrix}$$

with 0_i again being zero-submatrices and x the non-zero element of B in position (n,n)

FUTURE PROSPECTS. We note that $C = BB^T$. Keeping this in mind we can generalize the problem somewhat. Let

$$B = \begin{bmatrix} 0_1 & 0_2 \\ D & 0_3 \end{bmatrix}$$

where D is a kxk symmetric submatrix, k<n and 0_i are zero-submatrices. By observation we find that

$$C^2 = \begin{bmatrix} 0 & 0 \\ 0 & D^2 \end{bmatrix} = BB^T$$

and

$$C = \begin{bmatrix} 0 & 0 \\ 0 & D \end{bmatrix}$$

Unfortunately C is not unique and the problem is to find the particular C which will assure that the eigenvalues of T are the same as those of Y and those of Z. The author has experimented with the five-diagonal matrix occurring when the 5-point formula is applied to Laplace's partial differential equation. In this case

$$B = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & & & \\ '' & '' & & & \\ '' & '' & & & \\ '' & '' & & & \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$

He found that (5.1) can be applied when C is taken as

$$C = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ '' & '' & & 1 & '' \\ '' & '' & & & '' \\ '' & '' & & & '' \\ 0 & 1 & \cdots & 0 & 0 \end{bmatrix}$$

We note here again $C^2 = BB^T$

Some more research will have to be done with respect to this problem. For example, a formal proof that the relationship $C^2 = BB^T$ has to hold does not yet exist.

SECTION 6

COMPUTATIONAL RESULTS AND CONCLUSIONS

In Section (2) we have shown that the 2n eigenvalues of two (nxn) matrices P and Q are the 2n eigenvalues of a(2n x 2n) matrix S, where

$$S = \begin{bmatrix} A & B \\ B & A \end{bmatrix}, \quad P = A+B, \ Q = A-B.$$

This type of matrix occurs in the theory of directional couplers of wave guides, the theory of overlapping polymer chains and the Ising model of ferromagnetic materials [1].

It was found that the matrix S can easily be generated when we permute a σ-cycle permutation matrix, provided we let $\sigma = 1$ or $\sigma = 2n-1$.

It was the objective of the computation part of this thesis to show, that the calculation of eigenvalues and eigenvectors of the (2n x 2n) matrix S by use of two (nxn) matrices P and Q is superior to the direct calculation. Superior, that is, with respect to computer time, accuracy and storage space utilization of the computer. Two types of matrices, symmetric and non-symmetric were used in these computations. For the symmetric matrices we employed three numerical methods, namely Householder's, QR and Eberlein's. For non-symmetric matrices the latter two were used. The dimensions of the matrices under investigation ranged from n = 4 to n = 30. While the storage requirements for the direct solution of S were of the order of $2n \times 2n = 4n^2$ those for the "partitioned" method were of the order of $2(nxn) = 2n^2$, an effective saving of storage space of 50%.

Thus, matrices of types S and T which formerly could not be handled by computers, due to lack of storage space, may now be processed by means of the "partitioned" method.

It was shown in Section (4) for the Householder method that the accuracy of the eigenvalue solution is a function of the dimension of the matrix. Formula (4.55) indicates that the error increases as the dimension of the matrix increases. Thus the eigenvalues obtained by use of P and Q will be, in general, more accurate than those calculated from S and T directly. This observation is borne out by calculations summerized in Table 4 for the QR method. In that case we investigated a symmetric tridiagonal matrix of type T for which an analytical solution* had been found. For a 60 x 60 matrix the direct method was accurate to 5 decimal places while the "partitioned" method was accurate to 6 decimal places. Since a tridiagonal matrix involves fewer calculations with respect to the eigenvalue problem than one with no zero elements, it can reasonably be assumed that the difference in accuracy between the two types of solutions is even more pronounced in problems summerized in Tables 1 and 2.

Now we come to the third aspect of our investigation, computer time. We note from Tables 1, 2 and 3 that in the case of 8x8 matrices there is no significant difference in time between the direct and "partitioned methods". In the case of the 28x28 matrix the difference already becomes apparent, but when dealing with the 60x60 matrices it is significant. It takes about twice as long to evaluate the eigenvalues by

---

* The analytical solution is $\lambda_k = a - 2b\cos(k\pi/n+1)$ where a are the diagonal and b the off-diagonal elements.

the QR method directly than it does by the "partitioned" algorithm, and more than four times as long when Eberlein's method is employed. Thus the time difference becomes more pronounced as the dimension of the matrices increase.

Thus we have succeeded in showing that for a matrix of types S or T the "partitioned" method of solving the eigenvalue problem is superior to the direct solution. In the course of our investigations, we have observed that of the three numerical methods employed, the Householder algorithm is inferior to Eberlein's, at least as far as computer time is concerned. From our tables we note that it is impossible to obtain a solution for a 28x28 matrix by Householder's method in less than ten minutes, while we obtain it by Eberlein's within that period of time. Similarly, we observe from Table 4 that the QR method is more accurate than Eberlein's method. The difference amounts to one decimal place for a 60x60 matrix. As far as storage space requirements are concerned no significant differences exist between the three methods. We note that Theorem 9 presents us with a method of calculating the eigenvectors of S utilizing the eigenvalues of P and Q. A similar algorithm for a matrix of type T will have to be found yet.

TABLE 1

COMPUTER TIME IN SYMMETRIC CASES

DIMENSION OF MATRIX	HOUSEHOLDER		EBERLEIN		QR	
	COMPLETE	SUB-DIVIDED	COMPLETE	SUB-DIVIDED	COMPLETE	SUB-DIVIDED
N = 8	1:23*	1:28	1:15	1:23	1:14	1:15
N = 16	2:48	1:26				
N = 28	> 10:00	>10:00	3:20	2:59	1:36	1:22
N = 60	> 10:00	>10:00	> 24:00	6:18	4:49	2:22

* Time in minutes and seconds required to obtain all the eigenvalues and eigenvectors of the matrices.

TABLE 2

COMPUTER TIME IN NONSYMMETRIC CASES

DIMENSION OF MATRIX	EBERLEIN		QR	
	COMPLETE	SUB-DIVIDED	COMPLETE	SUB-DIVIDED
N = 60	> 10:00*	> 10:00	4:58	2:36

* Time in minutes and seconds to obtain all the eigenvalues of the matrix.

TABLE 3

COMPUTER TIME IN TRIDIAGONAL SYMMETRIC CASES

DIMENSION OF MATRIX	EBERLEIN		QR		HOUSEHOLDER	
	COMPLETE	SUB-DIVIDED	COMPLETE	SUB-DIVIDED	COMPLETE	SUB-DIVIDED
N = 8	1:14*	1:25	1:16	1:14	1:28	1:36
N = 60	>10:00	5:21	4:36	2:12	>10:00	>10:00

* Time in minutes and seconds to obtain all the eigenvalues and eigenvectors of the matrices.

TABLE 4

ACCURACY

DIMENSION OF MATRIX	EBERLEIN		QR		HOUSEHOLDER	
	COMPLETE	SUB-DIVIDED	COMPLETE	SUB-DIVIDED	COMPLETE	SUB-DIVIDED
N = 8	6*	6	6	6	6	6
N = 60		5	5	6		

* The number of significant digits of the computed solution. The matrices tested are symmetric tridiagonal and therefore analytical solutions are available. For actual numerical results see Appendix B.

APPENDIX A

The proofs to the theorems given in this section are by Faddeev [16], Wilkinson [18] and Fox [17].

Theorem. $$||A||_\infty = \max_i \sum_{j=1}^{n} |a_{ij}|$$

Proof: Let $||X||_\infty = 1$. Then

$$||AX||_\infty = \max_i \left| \sum_{j=1}^{n} a_{ij}\ x_j \right| \leq \max_i \sum_{j=1}^{n} |a_{ij}| \cdot |x_j|$$

$$\leq \max_i \sum_{j=1}^{n} |a_{ij}|$$

Consequently

$$\max_{||X||=1} ||A\ X|| \leq \max_i \sum_{j=1}^{n} |\ a_{ij}|$$

We shall now prove that $\max_{||X||=1} ||A\ X||$ is equal to $\max_i \sum_{j=1}^{n} |a_{ij}|$. We construct a vector X_o such that $||X_o||_\infty = 1$ and $||A\ X_o|| = \max_i \sum_{j=1}^{n} |a_{ij}|$.

Namely, let $\sum_{j=1}^{n} |a_{ij}|$ attain the greatest value for i = k then as the component $x_j^{(o)}$ of X_o we take $x_j^{(o)} = |a_{kj}|/a_{kj}$, if $a_{kj} \neq 0$ and $x_j^{(o)} = 1$, if $a_{kj} = 0$. Thus, $||X_o|| = 1$

Moreover, $$\left| \sum_{j=1}^{n} a_{ij} x_j^{(o)} \right| \leq \sum_{j=1}^{n} |a_{ij}| \leq \sum_{j=1}^{n} |a_{kj}|$$

for $i \neq j$ and $$\left| \sum_{j=1}^{n} a_{kj} x_j^{(o)} \right| = \sum_{j=1}^{n} |a_{kj}|$$

Hence

$$\max_i \left| \sum_{j=1}^{n} a_{ij} x_j^{(o)} \right| = \sum_{j=1}^{n} |a_{kj}| = \max_i \sum_{j=1}^{n} |a_{ij}|$$

Thus $$||A\ X_o||_\infty = \max_i \sum_{j=1}^{n} |a_{ij}|$$

and $$||A||_\infty = \max_i \sum_{j=1}^{n} |a_{ij}|$$

Theorem. $$||A||_1 = \max_j \sum_{i=1}^{n} |a_{ij}|$$

Proof: $$||A||_1 = \max_{||X||=1} \sum_{i=1}^{n} \left| \sum_{j=1}^{n} a_{ij} x_j \right| \leq \sum_{j=1}^{n} \sum_{i=1}^{n} |a_{ij}| . |x_j|$$

$$\leq \max_j \sum_{i=1}^{n} |a_{ij}|$$

Since $\sum |x_j| = 1$, we can reach equality by choosing X to be zero except in the element corresponding to the value j for which $\sum |a_{ij}|$ is largest, and to have unity in this component. Then $||X|| = 1$ and

$$||A||_1 = \max_j \sum_{i=1}^{n} |a_{ij}|$$

Theorem. $$||A||_2 = \sqrt{\lambda_1}$$

if λ_1 is the greatest eigenvalue of the Matrix $A^H A$.

Proof: Let $|X|^2 = \sum_{j=1}^{n} |x_j|^2 = (XX)$ and $||A|| = \max_{|X|=1} |AX|$.

But $|AX|^2 = (AX, AX) = (X, A^H AX)$

The matrix $A^H A$ is Hermitian. Hence its eigenvalues are real. Let λ_1 be its greatest eigenvalue. Then for $|X| = 1$ $\max (X, A^H AX) = \lambda_1$. Consequently $||A||_2 = \sqrt{\lambda_1}$

Theorem. $fl\ (x_1\ x_2\ \ldots,\ x_n) \equiv x_1\ x_2\ \ldots\ x_n\ (1+E)$

if $(1-2^{-t})^{n-1} \leq 1+E < (1+2^{-t})^{n-1}$

Proof: We consider $p_n = fl(x_1 \ldots x_n)$. The quantities p_r are defined recursively by

$$p_1 = x_1$$

$$p_r = fl(p_{r-1}\ x_r) \equiv p_{r-1} x_r\ (1+\varepsilon_r)$$

$$(1)\quad |\varepsilon_r| \leq 2^{-t}$$

Hence, we have

$$(2)\quad p_n \equiv x_1 x_2 \ldots x_n (1+\varepsilon_2)(1+\varepsilon_3) \ldots (1+\varepsilon_n)$$

each ε_r satisfies (1). Equation (2) implies that

$$fl\ (x_1\ x_2\ \ldots\ x_n) \equiv x_1\ x_2\ \ldots\ x_n (1+\varepsilon)$$

where

$$(1-2^{-t})^{n-1} \leq 1+\ \varepsilon \leq (1+2^{-t})^{n-1}$$

APPENDIX B

COMPUTER PROGRAMS

Not all the results obtained and programs used by the author are exhibited in this section. Usually only a few sample answers are given where the presentation of all results would have taken up too much space. The results are compiled so as to facilitate comparisons. An answer obtained by different methods is written out in full only the first time it appears. At all other times, only the last two digits are written. If an answer differs by more than two digits from its first version, all differing digits are written out.

The Letter I is written in place of answers too small in magnitude for accurate comparisons

```
$JOB            000702HANS BASTEL        100    010    030
$IBJOB          NODECK
$IBFTC
C     PROGRAM TO CALCULATE EIGENVALUES AND EIGENVECTORS OF AN (2N*2N)
C     MATRIX SUBDIVIDED INTO (N*N) MATRICES BY HOUSEHOLDER'S METHOD
      DIMENSION A(62,62),B(62,62),E(30,30),A1(30,30),A2(30,30)
      DIMENSION D1(1000),D2(1000),X(30),Y(500),Z(30)
      READ(5,1) N
      READ(5,2) (A(1,J), J=1,N)
C
C     WE GENERATE REMAINDER OF MATRIX
C
      CALL GEN1(A,N)
      WRITE(6,9)
      WRITE(6,3) (A(1,J), J=1,N)
C
C     WE START CALCULATIONS
C
      N=N/2
      DO 30 I=1,N
      DO 30 J=1,N
      JP=J+N
      B(I,J)=A(I,J)+A(I,JP)
   30 E(I,J)=A(I,J)-A(I,JP)
      REWIND 0
```

```
      N2=0
      DO 40 I=1,N
      DO 40 J=I,N
      N2=N2+1
      D1(N2)=B(I,J)
   40 D2(N2)=E(I,J)
      CALL HOUSE2(N,N,1.E-06,D1,X,Y,N)
      REWIND 0
      DO 50 J=1,N
   50 READ(0) (A1(I,J),I=1,N)
      CALL HOUSE2(N,N,1.E-06,D2,Z,Y,N)
      REWIND 0
      DO 60 J=1,N
   60 READ(0) (A2(I,J),I=1,N)
C
C     WE CALCULATE ORIGINAL EIGENVECTORS
C
      ERR=.000001
      DO 70 I=1,N
      I1=I+N
      DO 80 J=1,N
      IF(ABS(X(I)-Z(J)).GE.ERR) GO TO 80
      DO 90 K=1,N
      K1=K+N
      A(K,I)=A1(K,I)+A2(K,J)
      A(K1,I)=A1(K,I)-A2(K,J)
```

```
      A(K,I1)=A1(K,I)-A2(K,J)
   90 A(K1,I1)=A1(K,I)+A2(K,J)
   80 CONTINUE
      DO 100 K=1,N
      K1=K+N
      A(K,I)=A1(K,I)*0.5
  100 A(K1,I)=A(K,I)
      DO 110 J=1,N
      IF(ABS(X(J)-Z(I)).LE.ERR) GO TO 70
  110 CONTINUE
      DO 120 K=1,N
      K1=K+N
      A(K,I1)=A2(K,I)*0.5
  120 A(K1,I1)=-A2(K,I)*0.5
   70 CONTINUE
      DO 71 I=1,N
      I1=I+N
      B(I,I)=X(I)
   71 B(I1,I1)=Z(I)
      N=2*N
      DO 130 J=1,N
      SUM=0.
      DO 140 I=1,N
  140 SUM=SUM+A(I,J)*A(I,J)
      SUM=1.0/(SQRT(SUM))
```

```
      DO 150 I=1,N
  150 A(I,J)=A(I,J)*SUM
  130 CONTINUE
      WRITE(6,77)
      DO 888 I=1,N
      WRITE(6,8) B(I,I)
  888 WRITE(6,7) (A(J,I), J=1,N)
    1 FORMAT(I4)
    2 FORMAT(8F5.1)
    3 FORMAT( 8F16.8/)
    7 FORMAT(2X, 8E16.8/)
    8 FORMAT( 10X, E20.10//)
    9 FORMAT(30X, 17H THE FIRST ROW IS      ///)
   77 FORMAT( 25X,34H EIGENVALUES AND EIGENVECTORS ARE            ///)
      STOP
      END
$IBFTC  HANS
      SUBROUTINE GEN1(A,N)
      DIMENSION A(62,62)
      NH=N/2
      N1=NH+1
      DO 10 I=2,NH
      IM=I-1
      IP=I+NH
      A(I,1)=  A(1,I)
```

```
      A(I,N1)=A(1,IP)
      DO 10J=2,NH
      JM=J-1
      JP=J+NH
      J1=JM+NH
      A(I,J)=A(IM,JM)
   10 A(I,JP)=A(IM,J1)
      DO 20I=1,NH
      IP=  I+NH
      DO 20 J=1,NH
      JP=J+NH
      A(IP,J)=A(I,JP)
   20 A(IP,JP)=A(I,J)
      RETURN
      END
$IBSYS
```

```
$JOB            000702HANS BASTEL        100    010    030
$IBJOB          NODECK
$IBFTC
C       PROGRAM TO CALCULATE EIGENVALUES AND EIGENVECTORS OF A
C       SYMMETRIC MATRIX BY HOUSEHOLDER'S METHOD
        DIMENSION A(60,60),D(2000),X(60),Y(500),A1(60,60)
        READ(5,1) N
        READ(5,2) (A(1,J), J=1,N)
C
C       WE GENERATE REMAINDER OF MATRIX
C
        CALL GEN1(A,N)
        WRITE(6,9)
        WRITE(6,3) (A(1,J), J=1,N)
        REWIND 0
        N2=0
        DO 40 I=1,N
        DO 40 J=I,N
        N2=N2+1
     40 D(N2)=A(I,J)
        CALL HOUSE2(N,N,1.E-06,D,X,Y,N)
        REWIND0
        DO 50 J=1,N
     50 READ(0) (A1(I,J),I=1,N)
        WRITE(6,77)
```

```
      DO 888 I=1,N
      WRITE(6,8) X(I)
  888 WRITE(6,7) (A1(J,I), J=1,N)
    1 FORMAT (I4)
    2 FORMAT(8F5.1)
    3 FORMAT( 8F16.8/)
    7 FORMAT(2X, 8E16.8/)
    8 FORMAT( 10X, E20.10//)
    9 FORMAT(30X, 17H THE FIRST ROW IS       ///)
   77 FORMAT( 25X,34H EIGENVALUES AND EIGENVECTORS ARE           ///)
      STOP
      END
$ENTRY
$IBSYS
```

```
$JOB            000702HANS BASTEL        100    010    030
$IBJOB          NODECK
$IBFTC
C      PROGRAM TO CALCULATE EIGENVALUES AND EIGENVECTORS
C      OF A SYMMETRIC MATRIX BY THE QR METHOD
       DIMENSION A(62,62), C(62), X(62), Y(62)
       READ(5,1) N
       READ(5,2) (A(1,J), J=1,N)
C
C      WE GENERATE REMAINDER OF MATRIX
C
       CALL GEN1(A,N)
       WRITE(6,9)
       WRITE(6,3) (A(1,J), J=1,N)
C
C      WE START CALCULATIONS
C
       CALL HESSEN(A,N,62,C)
       CALL QREIG(A,N,X,Y,62)
       WRITE(6,66)
       DO 30 J=1,N
       WRITE(6,4) J,X(J)
    30 CONTINUE
     1 FORMAT(I4)
     2 FORMAT(8F5.1)
```

```
  3 FORMAT( 8F16.8/)
  4 FORMAT(10X,I10,E20.10)
  9 FORMAT(30X, 17H THE FIRST ROW IS       ///)
 66 FORMAT( 30X, 21H THE EIGENVALUES ARE         ///)
    CALL EXIT
    END
$ENTRY
$IBSYS
```

```
$JOB            000702HANS BASTEL        100    010    030
$IBJOB          NODECK
$IBFTC
C      PROGRAM TO CALCULATE EIGENVALUES OF A (2N*2N) MATRIX
C      SUBDIVIDED INTO (N*N) MATRICES
C      BY QR METHOD
       DIMENSION A(62,62), C(30), X(30), Y(30), B(30,30), E(30,30)
       READ(5,1) N
       READ(5,2) (A(1,J), J=1,N)
C
C      WE GENERATE REMAINDER OF MATRIX
C
       CALL GEN1(A,N)
       WRITE(6,9)
       WRITE(6,3) (A(1,J), J=1,N)
       N=N/2
       DO 30 I=1,N
       DO 30 J=1,N
       JP=J+N
       B(I,J)=A(I,J)+A(I,JP)
    30 E(I,J)=A(I,J)-A(I,JP)
       CALL HESSEN( B,N,30,C)
       CALL QREIG (B,N,X,Y,30)
       WRITE(6,6)
       DO 40 J=1,N
```

```
   40 WRITE(6,7)     J,X(J)
      CALL HESSEN(E,N,30,C   )
      CALL QREIG (E,N,X,Y,30)
      WRITE(6,8)
      DO 50 J=1,N
   50 WRITE(6,7) J,X(J)
    1 FORMAT(I4)
    2 FORMAT(8F5.1)
    3 FORMAT( 8F16.8/)
    6 FORMAT(30X, 30H THE EIGENVALUES OF A+B ARE           ///)
    7 FORMAT(40X,I10, E20.10)
    8 FORMAT(30X, 30H THE EIGENVALUES OF A-B ARE          ///)
    9 FORMAT(30X, 17H THE FIRST ROW IS      ///)
      CALL EXIT
      END
$ENTRY
$IBSYS
```

```
$JOB            000702HANS BASTEL        100    010    030
$IBJOB          NODECK
$IBFTC
C       PROGRAM TO CALCULATE EIGENVALUES AND EIGENVECTORS OF A
C       MATRIX BY EBERLEIN'S METHOD
        DIMENSION A(62,62), B(62,62)
        READ(5,1) N
        READ(5,2) (A(1,J), J=1,N)
C
C       WE GENERATE REMAINDER OF MATRIX
C
        CALL GEN1(A,N)
        WRITE(6,9)
        WRITE(6,3) (A(1,J), J=1,N)
C
C       WE START CALCULATIONS
C
        DO 30 I=1,N
        DO 30 J=1,N
        B(I,J)=0.
        IF(I.EQ.J) B(I,J)=1.0
     30 CONTINUE
        CALL EBERVC(A,N,2,200,.01,.001,1.E03,62,B,1.)
        WRITE(6,77)
        DO 888 I=1,N
```

```
      WRITE(6,8) A(I,I)
  888 WRITE(6,7) (B(J,I), J=1,N)
    1 FORMAT(I4)
    2 FORMAT(8F5.1)
    3 FORMAT( 8F16.8/)
    7 FORMAT(2X, 8E16.8/)
    8 FORMAT( 10X, E20.10//)
    9 FORMAT(30X, 17H THE FIRST ROW IS      ///)
   77 FORMAT( 25X,34H EIGENVALUES AND EIGENVECTORS ARE          ///)
      STOP
      END
```

```
$JOB           000702HANS BASTEL        100    010    030
$IBJOB         NODECK
$IBFTC
C     PROGRAM TO CALCULATE EIGENVALUES AND EIGENVECTORS OF AN (2N*2N)
C     MATRIX SUBDIVIDED INTO (N*N) MATRICES BY EBERLEIN'S METHOD
      DIMENSION A(60,60), B(60,60), E(30,30),A1(60,60), A2(30,30)
      READ(5,1) N
      READ(5,2) (A(1,J), J=1,N)
C
C     WE GENERATE REMAINDER OF MATRIX
C
      CALL GEN1(A,N)
      WRITE(6,9)
      WRITE(6,3) (A(1,J), J=1,N)
C
C     WE START CALCULATIONS
C
      N=N/2
      DO 30 I=1,N
      DO 30 J=1,N
      JP=J+N
      B(I,J)=A(I,J)+A(I,JP)
   30 E(I,J)=A(I,J)-A(I,JP)
      DO 40 I=1,N
      DO 40 J=1,N
```

```
      A1(I,J)=0.0
      IF(I.EQ.J) A1(I,J)=1.0
   40 CONTINUE
      DO 50 I=1,N
      DO 50 J=1,N
   50 A2(I,J)=A1(I,J)
      CALL EBERVC(B,N,2,200,.01,.001,1.E03,60,A1,1.)
      CALL EBERVC(E,N,2,200,.01,.001,1.E03,30,A2,1.)
C
C     WE CALCULATE ORIGINAL EIGENVECTORS
C
      ERR=.000001
      DO 70 I=1,N
      I1=I+N
      DO 80 J=1,N
      IF(ABS(B(I,I)-E(J,J)).GE.ERR) GO TO 80
      DO 90 K=1,N
      K1=K+N
      A(K,I)=A1(K,I)+A2(K,J)
      A(K1,I)=A1(K,I)-A2(K,J)
      A(K,I1)=A1(K,I)-A2(K,J)
   90 A(K1,I1)=A1(K,I)+A2(K,J)
   80 CONTINUE
      DO 100 K=1,N
```

```
      K1=K+N
      A(K,I)=A1(K,I)*0.5
  100 A(K1,I)=A(K,I)
      DO 110 J=1,N
      IF(ABS(B(J,J)-E(I,I)).LE.ERR) GO TO 70
  110 CONTINUE
      DO 120 K=1,N
      K1=K+N
      A(K,I1)=A2(K,I)*0.5
  120 A(K1,I1)=-A2(K,I)*0.5
   70 CONTINUE
      DO 71 I=1,N
      I1=I+N
   71 B(I1,I1)=E(I,I)
      N=2*N
      DO 130 J=1,N
      SUM=0.
      DO 140 I=1,N
  140 SUM=SUM+A(I,J)*A(I,J)
      SUM=1.0/(SQRT(SUM))
      DO 150 I=1,N
  150 A(I,J)=A(I,J)*SUM
  130 CONTINUE
      WRITE(6,77)
      DO 888 I=1,N
```

```
      WRITE(6,8) B(I,I)
  888 WRITE(6,7) (A(J,I),J=1,N)
    1 FORMAT(I4)
    2 FORMAT(8F5.1)
    3 FORMAT( 8F16.8/)
    7 FORMAT(2X, 8E16.8/)
    8 FORMAT( 10X, E20.10//)
    9 FORMAT(30X, 17H THE FIRST ROW IS      ///)
   77 FORMAT( 25X,34H EIGENVALUES AND EIGENVECTORS ARE          ///)
      STOP
      END
$ENTRY
$IBSYS
```

EIGENVALUES FOR (8x8) SYMMETRIC MATRIX

ELEMENTS OF FIRST ROW: (1, 2, 3, 4, 5, 6, 7)

NO.	HOUSEHOLDER	HOUSEHOLDER PARTITIONED
1	.34110767E 02	... 69E 02
2	I	I
3	I	I
4	I	I
5	- .11715726E 01	... 27E 01
6	- .21107704E 01	... 02E 01
7	- .68284264E 01	... 66E 01
8	- .15999999E 02	... 99E 02

NO.	QR	QR PARTITIONED
1	... 55E 02	... 60E 02
2	I	I
3	I	I
4	I	I
5	... 23E 01	... 28E 01
6	... 693E 01	... 695E 01
7	... 34E 01	... 63E 01
8	... 93E 02	- .16000000 E 02

NO.	EBERLEIN	EBERLEIN PARTITIONED
1	... 36E 02	... 55E 02
2	I	I
3	I	I
4	I	I
5	... 17E 01	... 25E 01
6	... 681E 01	... 696E 01
7	... 11E 01	... 49E 01
8	... 87E 02	... 98E 02

EIGENVECTOR COMPONENTS CORRESPONDING TO SOME
EIGENVALUES OF ABOVE MATRIX

	HOUSEHOLDER PARTITIONED	HOUSEHOLDER
1	- .37256403E 00	... 04E 00
	- .33346073E 00	... 62E 00
	- .33346074E 00	... 71E 00
	- .37256401E 00	... 399E 00
	...	...
5	- .19134168E 00	... 74E 00
	.46193975E 00	... 80E 00
	- .46193980E 00	... 82E 00
	.19134172E 00	... 79E 00
	...	...
7	.46193979E 00	... 77E 00
	.19134167E 00	... 57E 00
	- .19134171E 00	... 69E 00
	- .46193977E 00	... 73E 00
	...	...

	EBERLEIN	EBERLEIN PARTITIONED
1	... 00E 00	... 01E 00
	... 76E 00	... 71E 00
	... 76E 00	... 71E 00
	... 01E 00	... 03E 00
5	... 72E 00	... 75E 00
	... 77E 00	... 76E 00
	... 77E 00	... 74E 00
	... 72E 00	... 73E 00
7	... 78E 00	... 77E 00
	... 70E 00	... 68E 00
	... 72E 00	... 70E 00
	... 77E 00	... 75E 00

EIGENVALUES FOR (16x16) SYMMETRIC MATRIX

ELEMENTS OF FIRST ROW: (1, 2, ..., 16)

NO.	HOUSEHOLDER	HOUSEHOLDER PARTITIONED
1	.12265008E 03	... 08E 03
2	I	I
3	I	I
4	I	I
5	I	I
6	I	I
7	I	I
8	I	I
9	I	I
10	- .10395662E 01	... 66E 01
11	- .11744020E 01	... 23E 01
12	- .14464627E 01	... 36E 01
13	- .20285420E 01	... 28E 01
14	- .32398278E 01	... 85E 01
15	- .74471486E 02	... 92E 01
16	- .26274141E 02	... 41E 01

EIGENVALUES FOR (60x60) SYMMETRIC MATRIX

ELEMENTS OF FIRST ROW: (1,2,...,32,12,14,10,10,11,19,17,16,1,2,...16)

NO.	QR	QR PARTITIONED
1	- 0.74010339E 00	...1767E 00
2	- 0.10394783E 01	... 942E 01
3	- 0.15621691E 01	... 500E 01
4	- 0.23406509E 01	... 42E 01
5	0.24752292E 01	... 78E 01
6	- 0.40809060E 01	... 12E 01
7	0.28229825E 01	... 902E 01
8	0.29623596E 01	... 70E 01
9	0.30852689E 01	... 707E 01
10	0.31122335E 01	... 85E 01
11	- 0.50563282E 01	... 97E 01
12	- 0.55733818E 01	... 937E 01
13	- 0.57107912E 01	...8016E 01
14	- 0.59185702E 01	... 702E 01
15	0.55557301E 01	... 98E 01
16	- 0.72569279E 01	... 342E 01
17	0.57271207E 01	... 207E 01
18	- 0.96647612E 01	... 732E 01
19	- 0.10287984E 02	...8005E 02
20	0.81377675E 01	... 838E 01
21	- 0.10454824E 02	... 44E 02
22	0.93455066E 01	... 192E 01
23	0.94106247E 01	... 491E 01
24	0.13448830E 02	... 32E 02
25	- 0.14586916E 02	... 916E 02
26	0.13534069E 02	... 103E 02
27	- 0.18088952E 02	... 85E 02
28	- 0.18874990E 02	...5043E 02
29	- 0.19463553E 02	... 93E 02
30	- 0.19683034E 02	... 99E 02
31	- 0.20324141E 02	... 76E 02
32	- 0.20470955E 02	... 98E 02

33	0.17706660E 02	... 700E 02
34	0.17778306E 02	... 60E 02
35	0.18014909E 02	... 58E 02
36	0.18565599E 02	... 656E 02
37	0.28443414E 02	... 76E 02
38	- 0.29363776E 02	... 828E 02
39	0.32845773E 02	... 825E 02
40	- 0.34926480E 02	... 523E 02
41	0.34260713E 02	... 61E 02
42	- 0.35660363E 02	... 412E 02
43	- 0.40799458E 02	... 516E 02
44	- 0.44497778E 02	... 884E 02
45	0.43542106E 02	... 95E 02
46	- 0.51090260E 02	... 375E 02
47	0.46945260E 02	... 358E 02
48	0.49888831E 02	... 910E 02
49	0.52071063E 02	... 180E 02
50	- 0.61444711E 02	... 811E 02
51	0.69762852E 02	... 995E 02
52	- 0.70190845E 02	... 900E 02
53	0.84972948E 02	...3112E 02
54	0.86891199E 02	... 413E 02
55	- 0.90884388E 02	... 601E 02
56	- 0.95245156E 02	... 349E 02
57	- 0.12312038E 03	... 42E 03
58	- 0.18870613E 03	... 30E 03
59	- 0.26456400E 03	... 15E 03
60	0.71036588E 03	... 653E 03

```
$JOB            000702HANS BASTEL        100    010    030
$IBJOB          NODECK
$IBFTC
C      PROGRAM TO CALCULATE EIGENVALUES AND EIGENVECTORS
C      OF A NON-SYMMETRIC MATRIX BY THE QR METHOD
       DIMENSION A(62,62), C(62), X(62), Y(62)
       READ(5,1) N
       READ(5,2) (A(1,J), J=1,N)
C
C      WE GENERATE REMAINDER OF MATRIX
C
       CALL GEN1(A,N)
       WRITE(6,9)
       WRITE(6,3) (A(1,J), J=1,N)
C
C      WE START CALCULATIONS
C
       CALL HESSEN(A,N,62,C)
       CALL QREIG(A,N,X,Y,62)
       WRITE(6,66)
       DO 30 J=1,N
       WRITE(6,4) J,X(J)
    30 CONTINUE
     1 FORMAT(I4)
     2 FORMAT(8F5.1)
```

```
    3 FORMAT( 8F16.8/)
    4 FORMAT(10X,I10,E20.10)
    9 FORMAT(30X, 17H THE FIRST ROW IS       ///)
   66 FORMAT( 30X, 21H THE EIGENVALUES ARE        ///)
      CALL EXIT
      END
$IBFTC  HANS
      SUBROUTINE GEN1(A,N)
      DIMENSION A(62,62)
      DO 10 I=2,N
      IM=I-1
      A(I,1)=A(IM,N)
      DO 10 J=2,N
      JM=J-1
   10 A(I,J)=A(IM,JM)
      RETURN
      END
$ENTRY
$IBSYS
```

SOME EIGENVALUES FOR (60x60) NON-SYMMETRIC MATRIX

ELEMENTS OF FIRST ROW: (1,2,...,32,12,14,10,10,11,19,17,16,1,2,...16)

NO.	QR	QR PARTITIONED
1	- 0.34999966E 01	... 35E 01
2	- 0.34999966E 01	... 35E 01
3	- 0.46698596E 01	... 684E 01
4	- 0.46698596E 01	... 684E 01
5	- 0.91073592E 01	... 705E 01
6	- 0.91073592E 01	... 705E 01
7	- 0.10024090E 02	... 101E 02
8	- 0.10024090E 02	... 101E 02
9	- 0.50000079E 00	... 83E 00
10	- 0.50000079E 00	... 83E 00
11	0.19431574E 01	... 608E 01
12	0.19431574E 01	... 608E 01
13	- 0.13330078E 02	... 106E 02
14	- 0.13330078E 02	... 106E 02
15	- 0.88804265E 01	... 410E 01
16	- 0.88804265E 01	... 410E 01
17	- 0.13601749E 02	... 61E 02
18	- 0.13601749E 02	... 61E 02
19	- 0.13836123E 02	... 54E 02
20	- 0.13836123E 02	... 65E 02
21	- 0.19488786E 02	... 869E 02
22	- 0.19488786E 02	... 869E 02
23	- 0.20663009E 02	... 42E 02
24	- 0.20663009E 02	... 42E 02
25	0.17269850E 00	... 70252E 00
26	0.17269850E 00	... 70252E 00
27	- 0.12836801E 02	... 33E 02
28	- 0.12836801E 02	... 33E 02
29	0.10278338E 02	... 65E 02
30	0.10278338E 02	... 65E 02
31	0.97297672E 01	... 8010E 01
32	0.97297672E 01	... 8010E 01

```
$JOB            000702HANS BASTEL        100    010    030
$IBJOB          NODECK
$IBFTC
C      PROGRAM TO CALCULATE EIGENVALUES OF A TRIDIAGONAL MATRIX
C      ANALYTICAL SOLUTION
       DIMENSION EIV(60)
       READ(5,1) N
       READ(5,2) AA,BB,CC
       S=SQRT(BB*CC)
       PI=3.141593/FLOAT(N+1)
       DO 10 I=1,N
       FI=FLOAT(I)
    10 EIV(I)=AA-2.0*S*COS(FI*PI)
       WRITE(6,3)
       DO 20 J=1,N
    20 WRITE(6,4) J,EIV(J)
     1 FORMAT(I4)
     2 FORMAT(3F5.1)
     3 FORMAT(30X, 22H THE EIGENVALUES ARE       ///)
     4 FORMAT(40X, I10, E20.10)
       STOP
       END
$ENTRY
$IBSYS
```

```
$JOB            000702HANS BASTEL        100    010    030
$IBJOB          NODECK
$IBFTC
C       PROGRAM TO CALCULATE EIGENVALUES AND EIGENVECTORS
C       OF A SYMMETRIC TRIDIAGONAL MATRIX BY THE QR METHOD
        DIMENSION A(62,62), C(62), X(62), Y(62)
        READ(5,1) N
        READ(5,2) AA,BB,CC
C
C       WE GENERATE REMAINDER OF MATRIX
C
        CALL GEN1(A,N,AA,BB,CC)
        WRITE(6,9)
        WRITE(6,3) (A(1,J), J=1,N)
C
C       WE START CALCULATIONS
C
        CALL HESSEN(A,N,62,C)
        CALL QREIG(A,N,X,Y,62)
        WRITE(6,66)
        DO 30 J=1,N
        WRITE(6,4) J,X(J)
     30 CONTINUE
      1 FORMAT(I4)
      2 FORMAT(8F5.1)
```

```
   3 FORMAT( 8F16.8/)
   4 FORMAT(10X,I10,E20.10)
   9 FORMAT(30X, 17H THE FIRST ROW IS      ///)
  66 FORMAT( 30X, 21H THE EIGENVALUES ARE        ///)
     CALL EXIT
     END
$IBFTC  HANS
$IBFTC  HANS
     SUBROUTINE GEN1(A,N,AA,BB,CC)
     DIMENSION A(62,62)
     DO 10 I=1,N
     DO 10 J=1,N
  10 A(I,J)=0.0
     DO 20 I=1,N
  20 A(I,I)=AA
     N1=N-1
     DO 21 I=1,N1
  21 A(I,I+1)=BB
     DO 22 I=2,N
  22 A(I,I-1)=CC
     RETURN
     END
$ENTRY
$IBSYS
```

```
$JOB           000702HANS BASTEL        100    010    030
$IBJOB         NODECK
$IBFTC
C      PROGRAM TO CALCULATE EIGENVALUES OF A (2N*2N) MATRIX
C      SUBDIVIDED INTO (N*N) MATRICES
C      BY QR METHOD
C      MATRIX IS TRIDIAGONAL
       DIMENSION A(62,62), C(30), X(30), Y(30), B(30,30), E(30,30)
       READ(5,1) N
       READ(5,2) AA,BB,CC
C
C      WE GENERATE REMAINDER OF MATRIX
C
       CALL GEN1(A,N,AA,BB,CC)
       WRITE(6,9)
       WRITE(6,3) (A(1,J), J=1,N)
       N=N/2
       DO 30 I=1,N
       DO 30 J=1,N
       B(I,J)=A(I,J)
    30 E(I,J)=B(I,J)
       B(N,N)=B(N,N)+A(N,N+1)
       E(N,N)=E(N,N)-A(N,N+1)
       CALL HESSEN( B,N,30,C)
       CALL QREIG (B,N,X,Y,30)
```

```
      WRITE(6,6)
      DO 40 J=1,N
   40 WRITE(6,7)     J,X(J)
      CALL HESSEN(E,N,30,C   )
      CALL QREIG (E,N,X,Y,30)
      WRITE(6,8)
      DO 50 J=1,N
   50 WRITE(6,7) J,X(J)
    1 FORMAT(I4)
    2 FORMAT(8F5.1)
    3 FORMAT( 8F16.8/)
    6 FORMAT(30X, 30H THE EIGENVALUES OF A+B ARE            ///)
    7 FORMAT(40X,I10, E20.10)
    8 FORMAT(30X, 30H THE EIGENVALUES OF A-B ARE          ///)
    9 FORMAT(30X, 17H THE FIRST ROW IS      ///)
      CALL EXIT
      END
$ENTRY
$IBSYS
```

SOME EIGENVALUES FOR (60x60)TRIDIAGONAL MATRIX

DIAGONAL ELEMENTS: 2
CO-DIAGONAL ELEMENTS: -1

NO.	ANALYTIC SOLUTION	QR	QR PARTITIONED
1	0.26518255E-02	... 108E-02	... 12E-02
2	0.10600254E-01	... 15E-01	... 50E-01
3	0.23824215E-01	... 133E-01	... 196E-01
4	0.42288646E-01	... 531E-01	... 580E-01
5	0.65944567E-01	... 356E-01	... 459E-01
6	0.94729260E-01	...8954E-01	... 126E-01
7	0.12856640E 00	... 590E 00	... 13E 00
8	0.16736624E 00	... 564E 00	... 24E 00
9	0.21102589E 00	... 512E 00	... 49E 00
10	0.25942958E 00	... 858E 00	... 07E 00
11	0.31244895E 00	... 767E 00	... 47E 00
12	0.36994341E 00	... 198E 00	... 403E 00
13	0.43176049E 00	...5885E 00	...5967E 00
14	0.49773625E 00	... 442E 00	... 522E 00
15	0.56769575E 00	... 323E 00	... 575E 00
16	0.64145344E 00	... 125E 00	... 266E 00
17	0.71881379E 00	... 379E 00	... 231E 00
18	0.79957160E 00	...6913E 00	...6994E 00
19	0.88351274E 00	...0931E 00	... 429E 00
20	0.97041459E 00	... 111E 00	... 271E 00

```
$JOB            000702HANS BASTEL        100    010    030
$IBFTC
$IBJOB          NODECK
      SUBROUTINE GAUS1(A,B,C,X)
      DIMENSION A(10,10),B(10,10),C(10,10),X(10,10)
      READ(5,41) ((B(I,J),I=1,8),J=1,8)
   41 FORMAT(8F5.2)
      READ(5,42) (A(K,K),K=1,5)
   42 FORMAT(5F16.8)
      DO 43 I=1,3
      I1=I+5
   43 A(I1,I1)=A(I,I)
      N=8
      N1=N-1
      DO 49 I=1,N
      DO 49 J=1,N
   49 X(I,J)=0.
      DO 100 M=1,N
      DO 30 I=1,N
      DO 40 J=1,N
   40 C(I,J)=B(I,J)
   30 C(I,I)=C(I,I)-A(M,M)
      DO 31 J=1,N1
      J1=J+1
      N3=J
```

```
      S=0.
      DO 32 I=J,N1
      K=I+1
      S=AMAX1(ABS(C(I,J)),S)
      IF(S.LT.ABS(C(K,J))) N3=K
   32 CONTINUE
      IF(N3.EQ.J) GO TO 33
      DO 34 I=J,N
      S=C(J,I)
      C(J,I)=C(N3,I)
   34 C(N3,I)=S
   33 C(J,J)=1./C(J,J)
      DO 35 I=J1,N
   35 C(J,I)=C(J,I)*C(J,J)
      DO 36 I=J1,N
      I1=I-1
      IF(M.GT.2) GO TO 47
      IF(J.GT.2) GO TO 47
      WRITE(6,45) (C(I1,K1),K1=1,N)
      WRITE(6,45) (C(I,K1),K1 =1,N)
   47 DO 46 K1=J1,N
   46 C(I,K1)=C(I,K1)-C(J,K1)*C(I,J)
   36 CONTINUE
   31 CONTINUE
      X(N,M)=1.
```

```
      DO 37 I=1,N1
      K=N-I
      X(K,M)=0.
      DO 37 J=K,N1
      J1=J+1
   37 X(K,M)=X(K,M)-C(K,J1)*X(J1,M)
      X(N,M)=1.
      S=0.
      DO 39 J=1,N
   39 S=S+X(J,M)*X(J,M)
      S=1./S**0.5
      DO 38 J=1,N
   38 X(J,M)=X(J,M)*S
      IF(M.GT.3) GO TO 48
  100 CONTINUE
   48 WRITE(6,44) (A(J,J),(X(I,J),I=1,N),J=1,N)
   44 FORMAT (8(E20.8/8E16.8/))
   45 FORMAT(1X,8E16.8)
      STOP
      END
$ENTRY
$IBSYS
```

```
$IBFTC EBERVC
      SUBROUTINE EBERVC(A,N,IN,NBMAX,EPS,EPS1,EF,AV,IND)
      DIMENSION A(30,30),AV(30,30)
     DO 16 II=1,IN
     EPS=EPS/EF
     EPS1=EPS1/EF
     NB=0
  18 DR=0.0
     DI=0.0
     DO 17 I=2,N
     IJ=I-1
     DO 17 J=1,IJ
     C=A(I,J)+A(J,I)
     D=A(I,I)-A(J,J)
     IF(EPS.LE.ABS(C)) GO TO 20
  21 CC=1.0
     SS=0.0
     GO TO 22
  23 CC=D/C
     SIG=SIGN(1.,CC)
     COT=CC+SIG*SQRT(1.0+CC*CC)
     SS=SIG/SQRT(1.0+COT*COT)
     CC=SS*COT
     DR=DR+1.0
  22 E=A(I,J)-A(J,I)
```

```
      IF(EPS.GT.ABS(E)) GO TO 31
      CO=CC*CC-SS*SS
      SI=2.0*SS*CC
      H=0.0
      G=0.0
      HJ=0.0
      DO 40 K=1,N
      IF(K.EQ.I) GO TO 40
      IF(K.EQ.J) GO TO 40
      H=H+A(I,K)*A(J,K)-A(K,I)*A(K,J)
      S1=A(I,K)*A(I,K)+A(K,J)*A(K,J)
      S2=A(J,K)*A(J,K)+A(K,I)*A(K,I)
      G=G+S1+S2
      HJ=HJ+S1-S2
   40 CONTINUE
      D=D*CO+C*SI
      H=2.0*H*CO-HJ*SI
      F=(2.0*E*D-H)/(4.0*(E*E+D*D)+2.0*G)
      IF(EPS1.GT.ABS(F)) GO TO 31
      CH=1.0/SQRT(1.0-F*F)
      SH=F*CH
      DI=DI+1.0
      GO TO 36
   31 CH=1.0
      SH=0.0
```

```
36 C1=CH*CC-SH*SS
   C2=CH*CC+SH*SS
   S1=CH*SS+SH*CC
   S2=SH*CC-CH*SS
   IF((ABS(S1)+ABS(S2)).EQ.0.0) GO TO 17
   DO 52 L=1,N
   A1=A(L,I)
   A2=A(L,J)
   A(L,I)=C2*A1-S2*A2
   A(L,J)=C1*A2-S1*A1
    IF(IND.LT.0) GO TO 52
    A1=AV(L,I)
    A2=AV(L,J)
    AV(L,I)=C2*A1-S2*A2
    AV(L,J)=C1*A2-S1*A1
52  CONTINUE
   DO 53 L=1,N
   A1=A(I,L)
   A2=A(J,L)
   A(I,L)=C1*A1+S1*A2
   A(J,L)=C2*A2+S2*A1
    IF(IND.GT.0) GO TO 53
    A1=AV(I,L)
    A2=AV(J,L)
    AV(I,L)=C1*A1+S1*A2
```

```
      AV(J,L)=C2*A2+S2*A1
53    CONTINUE
17   CONTINUE
     IF((DR+DI).LT.0.5)  GO TO 49
     NB=NB+1
     IF(NB.NE.NBMAX) GO TO 18
16   CONTINUE
     EPS=EPS*EF**IN
     EPS1=EPS1*EF**IN
      IF(IND.LE.0) GO TO 70
      DO 80 I=1,N
      SUM=0.
      DO 81 J=1,N
81    SUM=SUM+AV(J,I)**2
      SUM=SQRT(SUM)
      DO 82 J=1,N
82    AV(J,I)=AV(J,I)/SUM
80    CONTINUE
      RETURN
70    DO 90 I=1,N
      SUM=0.
      DO 91 J=1,N
91    SUM=SUM+AV(I,J)**2
```

```
      SUM=SQRT(SUM)
      DO 92 J=1,N
   92 AV(I,J)=AV(I,J)/SUM
   90 CONTINUE
      RETURN
      END
```

```
C       SUBROUTINE TO PUT MATRIX IN UPPER HESSENBERG FORM.
        SUBROUTINE HESSEN(A,M)
         DIMENSION A(50,50),B(49)
        DOUBLE PRECISION SUM
        IF (M - 2) 30,30,32
     32 DO 40 LC = 3,M
        N = M - LC + 3
        N1 = N - 1
        N2 = N - 2
        NI = N1
        DIV = ABS(A(N,N-1))
        DO 2 J = 1,N2
        IF(ABS(A(N,J))- DIV) 2,2,1
      1 NI = J
        DIV = ABS(A(N,J))
      2 CONTINUE
        IF(DIV) 3,40,3
      3 IF(NI - N1) 4, 7,4
      4 DO 5 J = 1,N
        DIV = A(J,NI)
        A(J,NI) = A(J,N1)
      5 A(J,N1) = DIV
        DO 6 J = 1,M
        DIV = A(NI,J)
        A(NI,J) = A(N1,J)
```

```
  6 A(N1,J) = DIV
  7 DO 26 K = 1, N1
 26 B(K) = A(N,K)/A(N,N-1)
    DO 45 J = 1,M
    SUM = 0.0
    IF (J - N1) 46,43,43
 46 IF(B(J))  41,43,41
 41 A(N,J) = 0.0
    DO 42 K = 1,N1
    A(K,J) = A(K,J) - A(K,N1)*B(J)
 42 SUM = SUM + A(K,J)*B(K)
    GO TO 45
 43 DO 44 K = 1,N1
 44 SUM = SUM + A(K,J)*B(K)
 45 A(N1,J) = SUM
 40 CONTINUE
 30 RETURN
     END
    SUBROUTINE QRT(A,N,R,SIG,D)
     DIMENSION A(50,50),PSI(2),G(3)
    N1 = N - 1
    IA = N - 2
    IP = IA
    IF(N-3) 101,10,60
60  DO 12 J = 3,N1
```

```
      J1 = N - J
      IF(ABS(A(J1+1,J1))-D)    10,10,11
   11 DEN = A(J1+1,J1+1)*(A(J1+1,J1+1)-SIG)+A(J1+1,J1+2)*A(J1+2,J1+1)
      IF(DEN) 61,12,61
 61   IF(ABS(A(J1+1,J1)*A(J1+2,J1+1)*(ABS(A(J1+1,J1+1)+A(J1+2,J1+2)
     1-SIG)+ABS(A(J1+3,J1+2)))/DEN)-D)  10,10,12
 12   IP=J1
 10   DO  14  J=1,IP
      J1=IP-J+1
      IF(ABS(A(J1+1,J1))-D)    13,13,14
 14   IQ=J1
 13   DO  100  I=IP,N1
      IF(I-IP)   16,15,16
 15   G(1)=A(IP,IP)*(A(IP,IP)-SIG)+A(IP,IP+1)*A(IP+1,IP)+R
      G(2)=A(IP+1,IP)*(A(IP,IP)+A(IP+1,IP+1)-SIG)
      G(3)=A(IP+1,IP)*A(IP+2,IP+1)
      A(IP+2,IP)=0.0
      GO TO 19
 16   G(1)=A(I,I-1)
      G(2)=A(I+1,I-1)
      IF(I-IA)   17,17,18
 17   G(3)=A(I+2,I-1)
      GO TO 19
 18   G(3)=0.0
   19 XK = SIGN(SQRT(G(1)**2 + G(2)**2 + G(3)**2), G(1))
```

```
22    IF(XK)    23,24,23
23    AL=G(1)/XK+1.0
      PSI(1)=G(2)/(G(1)+XK)
      PSI(2)=G(3)/(G(1)+XK)
      GO TO 25
24    AL=2.0
      PSI(1)=0.0
      PSI(2)=0.0
25    IF(I-IQ)    26,27,26
26    IF(I-IP)     29,28,29
28    A(I,I-1)=-A(I,I-1)
      GO TO 27
29    A(I,I-1)=-XK
27    DO 30   J=I,N
      IF(I-IA)  31,31,32
31    C=PSI(2)*A(I+2,J)
      GO TO 33
32    C=0.0
33    E=AL*(A(I,J)+PSI(1)*A(I+1,J)+C)
      A(I,J)=A(I,J)-E
      A(I+1,J)=A(I+1,J)-PSI(1)*E
      IF(I-IA)    34,34,30
34    A(I+2,J)=A(I+2,J)-PSI(2)*E
30    CONTINUE
      IF(I-IA)    35,35,36
```

```
 35    L=I+2
       GO TO  37
 36    L=N
 37    DO  40  J=IQ,L
       IF(I-IA)  38,38,39
 38    C=PSI(2)*A(J,I+2)
       GO TO  41
 39    C=0.0
 41    E=AL*(A(J,I)+PSI(1)*A(J,I+1)+C)
       A(J,I)=A(J,I)-E
       A(J,I+1)=A(J,I+1)-PSI(1)*E
       IF(I-IA)  42,42,40
 42    A(J,I+2)=A(J,I+2)-PSI(2)*E
 40    CONTINUE
       IF(I-N+3)   43,43,1000
 43    E=AL*PSI(2)*A(I+3,I+2)
       A(I+3,I)=-E
       A(I+3,I+1)=-PSI(1)*E
       A(I+3,I+2)=A(I+3,I+2)-PSI(2)*E
1000    CALL WRITE(A,N)
 100   CONTINUE
 101   RETURN
        END
C        PROGRAM TO CALL QR TRANSFORMATION, MAXIMUM ITER IS 50.
       SUBROUTINE QREIG(A,M,ROOTR,ROOTI,IPRNT)
```

```
      DIMENSION A(50,50),ROOTR(50),ROOTI(50)
      COMMON IP,IQ
      NCOUNT=0
      N = M
      IF(IPRNT)  80,81,80
80    WRITE (6,104)
81    ZERO = 0.0
      JJ=1
177   XNN=0.0
      XN2=0.0
      AA = 0.0
      B = 0.0
      C = 0.0
      DD = 0.0
      R=0.0
      SIG=0.0
      ITER = 0
17    IF(N-2)     13,14,12
13    IF(IPRNT)  82,83,82
82    WRITE (6,105)A(1,1)
83    ROOTR(1) = A(1,1)
      ROOTI(1) = 0.0
1     RETURN
14    JJ=-1
12    X = (A(N-1,N-1) - A(N,N))**2
```

```
      S = 4.0*A(N,N-1)*A(N-1,N)
      ITER = ITER + 1
      IF(X .EQ. 0.0 .OR. ABS(S/X) .GT. 1.0E-8) GO TO 15
   16 IF(ABS(A(N-1,N-1))-ABS(A(N,N)))  32,32,31
   31 E = A(N-1,N-1)
      G = A(N,N)
      GO TO 33
   32 G = A(N-1,N-1)
      E = A(N,N)
   33 F = 0.
      H = 0.
      GO TO 24
   15 S = X + S
      X = A(N-1,N-1) + A(N,N)
      IF(S)    18,19,19
19    SQ=SQRT(S)
      F=0.0
      H=0.0
      IF (X) 21,21,22
21    E=(X-SQ)/2.0
      G=(X+SQ)/2.0
      GO TO 24
22    G=(X-SQ)/2.0
      E=(X+SQ)/2.0
      GO TO 24
```

```
   18 F = SQRT(-S)/2.0
      E=X/2.0
      G=E
      H=-F
24    IF(JJ)  28,70,70
   70 D = 1.0E-10*(ABS(G) + F)
      IF(ABS(A(N-1,N-2)) .GT. D)  GO TO 26
28    IF(IPRNT)  84,85,84
84    WRITE (6,105)E,F, ITER
      WRITE (6,105)G,H
85    ROOTR(N) = E
      ROOTI(N) = F
      ROOTR(N-1) = G
      ROOTI(N-1) = H
      N=N-2
      IF(JJ)  1,177,177
   26 IF(ABS(A(N,N-1)) .GT. 1.0E-10*ABS(A(N,N)))  GO TO 50
29    IF(IPRNT)  86,87,86
86    WRITE (6,105)A(N,N), ZERO, ITER
87    ROOTR(N) = A(N,N)
      ROOTI(N) = 0.0
      N=N-1
      GO TO 177
50    IF(ABS(ABS(XNN/A(N,N-1))-1.0)-1.0E-6)  63,63,62
62    IF(ABS(ABS(XN2/A(N-1,N-2))-1.0)-1.0E-6)  63,63,700
```

```
 63 VQ=ABS(A(N,N-1))-ABS(A(N-1,N-2))
    IF (ITER-15) 53,164,64
164 IF(VQ) 165,165,166
165 R = A(N-1,N-2)**2
    SIG = 2.0*A(N-1,N-2)
    GO TO 60
166 R = A(N,N-1)**2
    SIG = 2.0*A(N,N-1)
    GO TO 60
64  IF(VQ)   67,67,66
66  IF(IPRNT)  88,85,88
88  WRITE (6,107)A(N-1,N-2)
    GO TO 84
67  IF(IPRNT)  89,87,89
89  WRITE (6,107)A(N,N-1)
    GO TO 86
700 IF(ITER .GT. 50) GO TO 63
    IF(ITER .GT. 5 ) GO TO 53
701 Z1=          ((E-AA)**2+(F-B)**2)/(E*E+F*F)
    Z2=       ((G-C)**2+(H-DD)**2)/(G*G+H*H)
    IF(Z1-0.25)  51,51,52
51  IF(Z2-0.25)   53,53,54
53  R=E*G-F*H
    SIG=E+G
    GO TO 60
```

```
54    R=E*E
      SIG=E+E
      GO TO 60
52    IF(Z2-0.25) 55,55,601
55    R=G*G
      SIG=G+G
      GO TO 60
 601  R = 0.0
      SIG = 0.0
60    XNN=A(N,N-1)
      XN2=A(N-1,N-2)
      CALL QRT(A,N,R,SIG,D)
       NCOUNT=NCOUNT+1
       IF(NCOUNT.GT.8) RETURN
      AA=E
      B=F
      C=G
      DD=H
      GO TO 12
104   FORMAT(////1X, 9HREAL PART  6X  14HIMAGINARY PART, 26X
     1  13HTAKEN AS ZERO 6X 4HITER //)
105   FORMAT(1X,E15.8,3X,E15.8, 42X I3)
107   FORMAT(56X  E13.8)
       END
```

BIBLIOGRAPHY

(1) Friedman, Berhard; Eigenvalues of Compound Matrices, Research Report No. TW-16, New York University, New York, (1951).

(2) MacDuffee, C. C.; The Theory of Matrices. Chelsea Publishing Co., New York, (1946).

(3) Wilkinson, J. H.; The Algebraic Eigenvalues Problem. Clarendon Press, Oxford, (1965).

(4) Rutishauser, H.; "Solution of Eigenvalue Problems with the LR - Transformation". Appl. Math. Ser. nat. Bur. Stand. (1958).

(5) Francis, J. G. F.; "The QR Transformation". Computer J. 4 1 (1961).

(6) Ralson, A.; A First Course in Numerical Analysis. McGraw-Hill, New York, (1965).

(7) Kublanovskaya, V. N.; "On Some Algorithms for the Solution of the Complete Eigenvalue Problem". Zh. Vych. mat. 1 (1961).

(8) Householder, A. S.; The Theory of Matrices in Numerical Analysis. Blaisdell, New York (1964).

(9) Miller, G. A.; Theory and Applications of Finite Groups. Wiley, New York, (1916).

(10) Conte, S. D.; Elementary Numerical Analysis. McGraw-Hill, New York, (1965).

(11) Wilkinson, J. H.; Rounding Errors in Algebraic Processes. Prentice-Hall, Inc., Englewood Cliffs, N. J. (1963).

(12) Ortega, J. M.; "An Error Analysis of Householder's Method for the Symmetric Eigenvalue Problem". Numerische Math. 5, (1963).

(13) Forsythe, G. E., and Wasow, W. R.; Finite Difference Methods for Partial Differential Equations. Wiley, New York, (1960).

(14) Eberlein, P. J.; "A Jacobi-like Method for the Automatic Computation of Eigenvalues and Eigenvectors of an Arbitrary Matrix." J. Soc. Indust. Appl. Math. 10, (1962).

(15) Mirsky, L.; "On the Minimization of Matrix Norms". Amer. Math. Monthly, 65, (1958).

(16) Faddeev, D. K. and Faddeeva, V. N.; Computational Methods for Linear Algebra. W. H. Freeman, San Francisco, (1963).

(17) Fox, L.; An Introduction to Numerical Linear Algebra. Oxford University Press, London, (1964).

(18) Wilkinson, J. H.; "Rigorous Error Bounds for Computed Eigensystems". Computer J. 4., (1961).

OTHER REFERENCES USED

Bellman, R., Introduction to Matrix Analysis. McGraw-Hill, New York (1960).

Rall, L. B., Errors in Digital Computation. Vol. 1, Wiley, New York, (1965).

Finkbeiner, D. T., Introduction to Matrices and Linear Transformations. W. H. Freeman, San Francisco, (1960).

Wilkinson, J. H. "Householder's Method for the Solution of the Algebraic Eigenproblem". Computer J. 3, (1960).

Henrici, P., "Bounds for Eigenvalues of Certain Tridiagonal Matrices". J. Soc. Indust. Appl. Math. 5, (1963).

Kpoal, Z., Numerical Analysis. Wiley, New York, (1955).

Todd, J., Survey of Numerical Analysis. McGraw-Hill, New York, (1962).

Johnson, R. E., First Course in Abstract Algebra. Prentice-Hall, Englewood Cliffs, N. J. (1961).