General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

BELLCOMM, INC.
955 L'ENFANT PLAZA NORTH, S.W. WASHINGTON, D. C. 20024

B69 07092

SUBJECT: Eigenvalues and Eigenvectors of Symmetric Matrices - Case 320

DATE: July 31, 1969

FROM: J. S. Vandergraft

ABSTRACT

A FORTRAN IV subroutine has been written which, when used in conjunction with the subroutine TRIDMX in the UNIVAC 1108 MATH-PACK, will find the eigenvalues, and a set of orthogonal eigenvectors, for any real symmetric matrix.* The subroutine applies the QR algorithm to a symmetric tri-diagonal matrix. This algorithm finds a sequence of matrices, which are orthogonally similar to the original matrix, and which converges to a diagonal matrix. The product of the similarity transformations converges to the matrix of eigenvectors; hence the algorithm produces orthogonal eigenvectors, even when some eigenvalues are multiple.

Also included is a subroutine which uses the output of TRIDMX to transform the eigenvectors of the tri-diagonal matrix into the eigenvectors of the original matrix.

* The subroutine TRIDMX uses Householders' method to transform a symmetric matrix into tri-diagonal form.

BELLCOMM, INC.
955 L'ENFANT PLAZA NORTH, S.W. WASHINGTON, D. C. 20024

B69 07092

SUBJECT: Eigenvalues and Eigenvectors of Symmetric Matrices - Case 320

DATE: July 31, 1969

FROM: J. S. Vandergraft

MEMORANDUM FOR FILE

1.0 INTRODUCTION

The QR method was developed by Francis[1] as a method for finding the real and complex eigenvalues of an arbitrary matrix. When applied to a symmetric matrix, the algorithm also produces a complete set of orthogonal eigenvectors. Comparison with the procedures given by Wilkinson[2,3] for solving this same problem using Householder's reduction to tri-diagonal form, the Sturm sequence method, and inverse iteration, shows that the QR algorithm is about 60% faster[4]. Moreover, the eigenvectors produced by Wilkinson's routines are often not orthogonal, and in fact, if multiple eigenvalues exist, special techniques must be used to obtain a full set of eigenvectors.

The QR method should not be used, however, if only a few selected eigenvalues and eigenvectors are needed. For this problem, the Wilkinson techniques are superior.

If TRIDMX has been used to transform the matrix into tri-diagonal form, prior to applying the QR method, the eigenvectors produced by QR must be transformed into eigenvectors of the original matrix. A separate subroutine has been provided to do this transformation.

2.0 THE QR ALGORITHM

Let A be any symmetric matrix. It can be shown (see Section 2.4) that there is an orthogonal matrix Q and an upper triangular matrix R such that $A = Q \cdot R$. Let A_1 be the matrix $R \cdot Q$, and decompose A_1 into the product $Q_1 \cdot R_1$, where Q_1 is orthogonal, R_1 is upper triangular. Let $A_2 = R_1 \cdot Q_1$ and repeat this process to obtain a sequence of matrices $A_1, A_2, \cdots$. The k-th step is:

> Given A_{k-1}, find an orthogonal matrix Q_{k-1}, and an upper triangular matrix R_{k-1} so that $A_{k-1} = Q_{k-1} \cdot R_{k-1}$. Then, let $A_k = R_{k-1} \cdot Q_{k-1}$.

Since

$$A_k = R_{k-1} \cdot Q_{k-1} = Q_{k-1}^T \; Q_{k-1} \cdot R_{k-1} \cdot Q_{k-1}$$

$$= Q_{k-1}^T \; A_{k-1} \; Q_{k-1}$$

it follows that A_k is similar to A_{k-1}, and hence by induction, all of the matrices A, $A_1, A_2, \cdots$ are similar and therefore have the same eigenvalues.

2.1 Convergence Theorem

Let A be any real symmetric matrix, and let $A_1, A_2 \cdots$ be the sequence of matrices defined above. Then this sequence converges to a diagonal matrix, where the diagonal elements are the eigenvalues of A. Moreover, if λ_i denotes the i-th diagonal element, then $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|$.

In the special case when all of the eigenvalues have distinct moduli, (i.e., $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n|$,) it can be shown that the (i,j) element, i>j, tends to zero like

$$\left|\frac{\lambda_i}{\lambda_j}\right|^k$$

Because of symmetry the (j,i) element also tends to zero. If the eigenvalues do not all have distinct moduli, the off-diagonal elements still tend to zero, but in a more complicated manner[7].

2.2 Shift of Origin

To increase the rate at which the off-diagonal elements tend to zero, the matrix A_k is replaced by $A_k - S_k I$, where S_k is a scalar. Since the eigenvalues of $A_k - S_k I$ are $\lambda_1 - S_k$, $\lambda_2 - S_k, \cdots, \lambda_n - S_n$, the (i,j) element, i>j, tends to zero like

$$\left|\frac{\lambda_i - S_k}{\lambda_j - S_k}\right|^k$$

Hence, if S_k is chosen to be close to λ_i, the ratio $|\lambda_i - S_k|/|\lambda_j - S_k|$ is very small, and convergence to zero is accelerated. The method for choosing S_k is described in Section 2.3.

In order to preserve the similarity of the matrices $A_1, A_2, \cdots$, and still incorporate the shift of origin idea, the basic algorithm is replaced by:

$$\left(A_k - S_k I\right) = Q_k \cdot R_k$$

$$A_{k+1} = R_k\, Q_k + S_k I$$

It should be observed that the use of origin shifts may destroy the ordering of the eigenvalues along the diagonal.

2.3 QR Applied to Tri-diagonal Matrices

It is easily seen that if A is symmetric and tri-diagonal, then so also are $A_1, A_2, \cdots$. Hence a preliminary reduction to tri-diagonal form, using Householder's method for example, results in a drastic reduction in computation time, program complexity, and storage requirements. Furthermore, if A is tri-diagonal, then the last row of A_k contains only two non-zero elements, $a_{n,n-1}^{(k)}$, $a_{n,n}^{(k)}$. By the Gerschgorin Theorem[5], the element $a_{nn}^{(k)}$ differs from an eigenvalue by less than $|a_{n,n-1}^{(k)}|$, provided $|a_{n,n-1}^{(k)}|$, is small. If this is true, then $a_{n,n}^{(k)}$ is close to λ_n and can be effectively used as the shift parameter S_k, defined in the previous section. In this case, the (n,n-1) element will tend to zero very rapidly. As soon as this element is suitably small, $a_{nn}^{(k)}$ can be accepted as an eigenvalue, and the last row and column can be dropped from the matrix. The algorithm is then applied to the resulting (n-1) x (n-1) matrix.

A somewhat better choice for S_k is to use the smallest eigenvalue of the 2x2 matrix

$$\begin{pmatrix} a_{n-1,n-1}^{(k)} & a_{n-1,n}^{(k)} \\ a_{n,n-1}^{(k)} & a_{nn}^{(k)} \end{pmatrix}$$

See [2].

2.4 Calculation of A_{k+1}

The matrices A_k, A_{k+1} are related by

$$A_{k+1} = Q_k^T A_k Q_k$$

where Q_k is an orthogonal matrix, such that $Q_k^T A_k$ is upper triangular. (For simplicity, in this section we will assume $S_k = 0$.) Let U_1 be the rotation matrix

$$U_1 = \begin{bmatrix} \cos\theta & \sin\theta & 0 \cdots & 0 \\ -\sin\theta & \cos\theta & 0 & \\ 0 & 0 & 1 & \\ \vdots & & \ddots & 0 \\ 0 & \cdots\cdots & 0 & 1 \end{bmatrix}$$

where θ is chosen so that the (2,1) element of $U_1 \cdot A_k$ is zero. That is,

$$\cos\theta = \frac{a_{11}^{(k)}}{r} \quad , \quad \sin\theta = \frac{a_{21}^{(k)}}{r}$$

$$r = \left(a_{11}^{\;2} + a_{21}^{\;2}\right)^{1/2}$$

Similarly, let

$$U_i = \begin{array}{cc} & \begin{array}{c} i^{\underline{th}} \text{ column} \\ \downarrow \end{array} \\ \left[\begin{array}{cccccc} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & & \cos\theta & \sin\theta & \\ & & & -\sin\theta & \cos\theta & \\ & & & & & 1 \\ & & & & & & \ddots \\ & & & & & & & 1 \end{array}\right] & \leftarrow i^{\underline{th}} \text{ row} \end{array}$$

where

$$\cos\theta = \frac{a_{ii}^{(k)}}{r} \quad , \quad \sin\theta = \frac{a_{i+1,i}^{(k)}}{r}$$

$$r = \left(a_{ii}^2 + a_{i+1,i}^2\right)^{1/2}$$

Then, if A_k is tri-diagonal, and

$$U_{n-1} \cdot U_{n-2} \cdots U_2 \cdot U_1 \cdot A_k = R_k$$

R_k will be upper triangular, and $Q_k^T = U_{n-1} \cdot U_{n-2} \cdots U_1$ is orthogonal.

2.5 Program Details

The subroutine QR is essentially a FORTRAN IV version of the algorithm QR 2, as described in [6], and including the modification suggested in [2]. The logic has been changed slightly, and the "zero" tolerance has been set to $\varepsilon = 10^{-8} \; ||A||_\infty$ where

$$||A||_\infty = \max_{i=1,\cdots,n} \sum_{j=1}^{n} |a_{ij}| \quad .$$

The matrix is assumed to be in tri-diagonal form, with diagonal elements A(1),···,A(N), off-diagonal elements B(2),···B(N). The two-dimensional array X is initially set equal to the N x N identity matrix, and the transformations $Q_1, Q_2, \cdots$ are applied to

X as they are generated. Since Q_k is a product of simple plane rotations, they need not be stored as a two-dimensional array; hence, the only two-dimensional array which is needed is the array X which will finally contain all of the eigenvectors.

2.6 Accuracy

All of the transformations involved in the QR algorithm are stable with respect to round-off error. Hence, good accuracy can be expected, even for very large problems. In practice, it is found that the largest eigenvalues are accurate to at least seven significant figures, and the corresponding eigenvectors to six significant figures. The smaller eigenvalues will have fewer accurate significant figures because, in general, all eigenvalues have the same <u>absolute</u> accuracy.

2.7 Test Problems

The subroutine was tested on the following problems.

A) The matrix W_{21}^{+}, defined in Wilkinson[4], page 308. This is a 21 x 21 symmetric tri-diagonal matrix, which has three pairs of eigenvalues which agree to 8 figures. The subroutine found all eigenvalues and vectors accurate to at least 7 figures. The maximum element of the matrix $I - X^T X$, where X is the matrix of computed eigenvectors, was less than 10^{-7}.

B) A 5 x 5 symmetric matrix, given in [6]. The matrix was first reduced to tri-diagonal form, using TRIDMX. The eigenvalues and vectors were found using QR, and the vectors were transformed using TRANSF (see Section 4). The answers were correct to 7 figures, and the orthogonality test, used in problem A, was 10^{-7}.

C) 120 x 120 symmetric matrix, produced by S. N. Hou. This matrix has eigenvalues of the order 10^7 , and zero is an eigenvalue of multiplicity three. The three smallest calculated eigenvalues were of order .1, and the orthogonality test was 10^{-6}. As a further check, the maximum element of the matrix AX-DX was computed. Here X is the matrix of computed eigenvectors, D is the diagonal matrix of eigenvalues. This quantity, divided by the maximum element of A, was $\sim 10^{-6}$.

2.8 Calling Sequence

CALL QR (N, A, B, E, X, W1, W2, W3, M)

N : Dimension of all matrices and vectors.

A : A one dimensional array, containing the diagonal elements of a symmetric tri-diagonal matrix.

B : A one dimensional array, containing the off-diagonal elements of the tri-diagonal matrix, in locations B(2),··,B(N). The subroutine sets B(1) = 0.

X : A two dimensional array, which is used to store the eigenvectors. The eigenvector corresponding to the k-th eigenvalue is stored in X(1,K), X(2,K), ···,X(N,K). The subroutine initializes this array so that X(I,J) = δ_{IJ}.

E : A one dimensional array which is used to store the eigenvalues.

M : The maximum value that N can assume.

W1,W2, W3 : One dimensional working arrays.

3.0 TRANSFORMATION SUBROUTINE

If the FORTRAN statement

```
CALL TRIDMX (N, M, T, A, B)
```

is used to transform the symmetric matrix T into tri-diagonal form, then the transformation matrix is stored in the lower triangular part of T, but in the following form:

$$Q = (I-2W_2W_2^T)(I-2W_3W_3^T)\cdots(I-2W_{n-1}W_{n-1}^T)$$

where

$$W_r = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ T(r,r-1) \\ T(r+1,r-1) \\ \vdots \\ T(N,r-1) \end{bmatrix}$$

The subroutine TRANSF applies the transformation Q tc the eigenvectors of the tri-diagonal matrix to transform them into eigenvalues of the original matrix.

3.1 Calling Sequence

CALL TRANSF (N, T, X, W, M)

N: Dimension of all matrices and vectors.

T: The two dimensional array which results from using TRIDMX to transform T into tri-diagonal form.

X: A two-dimensional array which contains the eigenvectors of the tri-diagonal matrix. That is, X(1,K), X(2,K), ···, X(N,K) is the k-th eigenvector.

W: A one dimensional working array.

M: The maximum value of N.

3.2 Example

The statements

```
CALL TRIDMX (N, M, T, A, B)
CALL QR (N, A, B, E, X, W1, W2, W3, M)
CALL TRANSF (N, T, X, W1, M)
```

can be used to find a matrix X of orthogonal eigenvectors, and a vector E of eigenvalues, for the symmetric matrix T. In addition to the two-dimensional arrays X,T, five one-dimensional working arrays A,B,W1,W2,W3, and the vector E, are required.

J. S. Vandergraft

2031:JSV:jct J. S. Vandergraft

Attachment

```
      SUBROUTINE QR(N,A,B,E,X,SN,CS,C,L)
C   THIS SUBROUTINE FINDS THE EIGENVALUES AND EIGENVECTORS OF A
C   SYMMETRIC TRIDIAGONAL MATRIX.N IS THE DIMENSION,A(1),...A(N) THE
C   DIAGONAL,B(2),...B(N)THE OFF-DIAGONAL,E(1),...E(N) THE EIGENVALUES,
C   X(1,K),...,X(N,K) IS THE EIGENVECTOR CORRESPONDING TO E(K),AND SN,
C   CS,C ARE ONE DIMENSIONAL WORKING ARRAYS.
      DIMENSION A(L),B(L),E(L),X(L ,L),SN(L),CS(L),C(L)
      REAL NORM ,MU,LAM
C     SET THE X ARRAY EQUAL TO THE NXN IDENTITY
      DO 200 I=1,N
      DO 201 J=I,N
      X(I,J)=0.
 201  X(J,I)=0.
 200  X(I,I)=1.
      B(1)=0.0
      NORM = ABS(B(N))+ABS(A(N))
      N1=N-1
      DO 10  I=1,N1
      SUM=ABS(A(I))+ABS(B(I))+ABS(B(I+1))
      IF (SUM .GT.NORM) NORM=SUM
 10   CONTINUE
      EPS= NORM * (10.E-8)
      MU=0.
      M=N
 15   IF (M.LE.0) GO TO 500
C     CHECK FOR POSSIBLE DECOUPLING OF THE MATRIX
 20   IF (ABS(B(M)).GT. EPS) GO TO 40
```

```
      E(M)=A(M)
      M=M-1
      GO TO 15
40    M1=M-1
      K=M1
41    IF(ABS(B(K)).LE.EPS) GO TO 42
      K=K-1
      GO TO 41
C     DETERMINE THE SHIFT OF ORIGIN
42    B0=B(M)**2
      A1=SQRT((A(M1)-A(M))**2+4.*B0)
      T=A(M1)*A(M)-B0
      A0=A(M1)+A(M)
      FACT=1.0
      IF(A0 .LT.0.) FACT=-1.0
      LAM=0.5*(A0+FACT*A1)
      T=T/LAM
      IF (ABS(T-MU)-0.5*ABS(T)) 70,80,80
70    MU=T
      LAM=T
      GO TO 90
80    IF (ABS(LAM-MU)-0.5*ABS(LAM)) 81,82,82
81    MU=LAM
      GO TO 90
82    MU=T
      LAM=0.
90    A(K)=A(K)-LAM
      BETA=B(K+1)
C     DO THE TRANSFORMATION ON THE LEFT
```

```
      DO 100 J=K,M1
      A0=A(J)
      A1=A(J+1)-LAM
      B0=B(J+1)
      T=SQRT(A0**2+BETA**2)
      COSE=A0/T
      CS(J)=COSE
      SINE =BETA/T
      SN(J)=SINE
      A(J)=COSE*A0+SINE*BETA
      A(J+1)=-SINE*B0+COSE*A1
      B(J+1)=COSE*B0+SINE*A1
      BETA=B(J+2)
      B(J+2)=COSE*BETA
      C(J+1)=SINE*BETA
100   CONTINUE
C     DO THE TRANSFORMATION ON THE RIGHT
      B(K)=0.
      C(K  )=0.
      DO 110  J=K,M1
      SINE=SN(J)
      COSE=CS(J)
      A0=A(J)
      B0=B(J+1)
      B(J  )_B(J  )*COSE + C(J  )*SINE
      A(J)=A0*COSE+B0*SINE+LAM
      B(J+1)=-A0*SINE+B0*COSE
      A(J+1)=A(J+1)*COSE
```

```
C     APPLY THE TRANSFORMATIONS TO THE X MATRIX
          DO 120 I=1,N
          X0=X(I,J)
          X1=X(I,J+1)
          X(I,J)=X0*COSE + X1*SINE
          X(I,J+1)=-X0*SINE +X1*COSE
  120     CONTINUE
  110     CONTINUE
          A(M)=A(M)+LAM
          GO TO 15
  500     RETURN
          END
@     FOR,IS  TRANSF,TRANSF
          SUBROUTINE TRANSF(N,A,X,C,M)
C     THIS SUBROUTINE TRANSFORMS THE EIGENVECTORS OF A TRIDIAGONAL
C     MATRIX INTO THE EIGENVECTORS OF THE ORIGINAL MATRIX.
C     A IS THE MATRIX WHICH WAS USED AS INPUT TO TRIDMX, AND
C     X IS THE MATRIX OF EIGENVECTORS
          DIMENSION A(M,M),X(M,M),C(M)
          N2=N-2
          DO 102 K1=1,N2
          K=N-K1
          K2=K-1
          DO 103 J=1,N
          SUM =0
          DO 104 I=K,N
  104     SUM =SUM+A(I,K2)*X(I,J)
  103     C(J)=2.*SUM
          DO 105 I=K,N
```

```
      DO 105 J=1,N
105   X(I,J)=X(I,J)-A(I,K2)*C(J)
102   CONTINUE
      RETURN
      END
```

REFERENCES

1. J. G. F. Francis, The QR Transformation - A Unitary Analogue to the LR Transformation, Parts 1 & 2, Comp. Journal, 4, 265-271 and 332-345, (1961/62).

2. J. H. Welsch, Certification of Algorithm 254: Eigenvalues and Eigenvectors of a Real Symmetric Matrix by the QR Method, Comm. ACM, Vol. 10, No. 6, 376-377 (1967).

3. J. H. Wilkinson, Handbook Series Linear Algebra, Numer. Math 4, 354-376 (1963).

4. J. H. Wilkinson, The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, 1965.

5. A. S. Householder, The Theory of Matrices in Numerical Analysis, Blaisdell, New York, 1964.

6. P. A. Businger, Eigenvalues and Eigenvectors of a Real Symmetric Matrix by the QR Method, Algorithm 254, Comm. A.C.M. Vol 8, No. 4, 218-219, (1965).

7. B. N. Parlett, Convergence of the QR Algorithm, Numer. Math, 1, 187-193, (1965).