Copyright © 1975, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

A PROGRAM TO COMPUTE THE REAL SCHUR FORM OF A REAL SQUARE MATRIX

by

B. N. Parlett and R. Feldman

Memorandum No. ERL-M526

June 1975

A PROGRAM TO COMPUTE THE REAL SCHUR FORM

OF A REAL SQUARE MATRIX

by

B. N. Parlett and R. Feldman

Memorandum No. ERL-M526

June 1975

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

A PROGRAM TO COMPUTE THE REAL SCHUR FORM

OF A REAL SQUARE MATRIX[†]

by

B.N. Parlett
Department of Mathematics
Department of Electrical Engineering and Computer Sciences
and the Electronics Research Laboratory
University of California
Berkeley, California 94720

and

R. Feldman
Department of Mathematics
University of California
Berkeley, California 94720

June 1975

Abstract

A Fortran program is presented which will obtain the real Schur form of a real $n \times n$ matrix in $10n^3 + 30n^2$ multiplications (approximately).

Key Phrases: Schur Form, real matrix

[†]Research sponsored by Office of Naval Research Contract N00014-69-A-0200-1017.

The algorithm is described at three different levels.

Level 1 is for a busy colleague.

Level 2 is for publication.

Level 3 is for the programmer.

Table of Contents

1. The Schur Form

A result in matrix theory, often called Schur's lemma, states that any square matrix B, whether real or complex, is unitarily similar to an upper triangular complex matrix S:

$$B = PSP^* , \qquad PP^* = P^*P = 1 \quad .$$

Here P^* denotes the conjugate transpose of P. Using slightly different language the lemma states that there is an orthonormal basis in the vector space on which B acts such that B's representation in this basis is upper triangular. Thus S may be regarded as a <u>canonical form</u> for B acting on Euclidean space.

Because S is triangular its eigenvalues $\{\lambda_1,\lambda_2,\ldots,\lambda_n\}$ lie revealed on the diagonal. In fact the eigenvalues may be taken in any desired order down the diagonal. Even when this order is fixed the matrix S is still not uniquely determined by B. However, the possible variations in S are rather trivial because $|s_{ij}|$, $i < j$, is fixed whenever $\lambda_i = s_{ii} \neq \lambda_j = s_{jj}$.

Discovery of S solves the eigenvalue problem for B and facilitates the computation of eigenvectors. Another use of S is in the formation of an analytic function ϕ of B since $\phi(B) = P\phi(S)P^*$.

From a practical point of view one defect of the Schur Form S is that S may be complex even when B is real. So we ask for the canonical form of B in real Euclidean space. The answer is an easy modification of S, called the real Schur Form $\hat{S}$ which is quasi-triangular. That is, $\hat{S}$ is <u>block</u> upper triangular and the diagonal blocks are either 1×1 or 2×2. To each complex conjugate pair of eigenvalues λ and $\bar{\lambda}$ in S there corresponds a real 2×2 diagonal block in $\hat{S}$ whose eigenvalues are λ and $\bar{\lambda}$. Sometimes it is convenient to standardize the real Schur form by requiring that the 2×2 diagonal blocks have the form

$$\begin{bmatrix} \rho & \beta \\ \gamma & \rho \end{bmatrix}, \quad \gamma > 0, \quad \beta < 0, \quad -\beta\gamma = \mu^2$$

where $\lambda = \rho + i\mu$, $\mu > 0$, and $i^2 = -1$. In general it is not possible to arrange that $\gamma = -\beta = \mu$.

An example of a standardized real Schur Form is

$$\begin{bmatrix} 3 & 1 & 1 & 0 & -1 & 0 \\ 0 & 1 & -3 & 2 & 3 & -1 \\ 0 & 2 & 1 & 1 & 0 & 4 \\ 0 & 0 & 0 & 2 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Our purpose here is to compute the real Schur form, not to praise it. Algorithms for the complex case are available in EISPACK Release 2.

2. The Algorithm (described at Level 1)

It is not difficult to compute P and $\hat{S}$ from B, the techniques we use are quite standard. B is reduced to upper Hessenberg form H by means of orthogonal similarity transformations and then H is reduced to $\hat{S}$ by the double QR algorithm. The product of all the orthogonal matrices used in the process are accumulated to form P.

We make use of a few devices to keep the number of QR transformations fairly low.

3. The Algorithm (described at Level 2)

The process has three steps:

Step 1: The routine PERMS, a modification of the EISPACK [1] routine BALANC, performs a sequence of row and column interchanges which detect when B is a permutation of a block triangular matrix and put it in the standard form

$$B_2 = P_1^* B P_1$$

where P_1 is a permutation matrix and

$$B_2 = \begin{pmatrix} B_{11} & B_{12} & B_{13} \\ 0 & B_{22} & B_{23} \\ 0 & 0 & B_{33} \end{pmatrix},$$

Figure 1

with B_{11} and B_{33} upper triangular. PERMS also acts on B_{22}. The goal is to bring rows with excessive norms to the top in order to get the QR algorithm off to a good start.

More precisely rows (and columns) are exchanged if the ratio of their ℓ_1-norms exceeds two.

In most cases $B_2 = B_{22}$, but the normalization which PERMS effects is rapid and is a necessary preparation for the routines which follow.

Step 2: The (2,2) block of B_2 is reduced to upper Hessenberg form by ORTHAN, a modification of the EISPACK routine ORTHES, and the product of the sequence of reflections is accumulated to yield P_2 such that

$$B_3 = P_2^* B_2 P_2$$

is in upper Hessenberg form.

Step 3: The (2,2) block of B_3 is reduced to quasi-triangular form by HQR3, a modification of the EISPACK routine HQR2. $\hat{S} = P_3^* B_3 P_3$

No effort is made to compute the eigenvectors of $\hat{S}$, but WI, which contains the imaginary parts of the eigenvalues, is retained, to indicate the presence of a 2×2 block on the main diagonal of $\hat{S}$. The array $\hat{S}$ is forced explicitly to be block upper triangular in case the user wishes to have it printed out (i.e., $\hat{S}$ is zero below the block diagonal).

In addition HQR3 performs a supplementary plane rotation after a pair of complex conjugate eigenvalues, $\lambda \pm i\mu$, has been recorded in the course of the QR algorithm. The transformation of the diagonal block is

$$\begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \begin{pmatrix} c & -s \\ s & c \end{pmatrix} = \begin{pmatrix} \lambda & \theta \\ \xi & \lambda \end{pmatrix}$$

where $\xi\theta = -\mu^2$. (This device is not used in HQR2.)

Note that it is not in general possible to transform

$$\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \rightarrow \begin{pmatrix} \lambda & -\mu \\ \mu & \lambda \end{pmatrix}$$

using <u>orthogonal</u> similarity transformations.

The purpose of the transformation is to yield a simple solution to certain systems of linear equations which sometimes must be solved. The supplementary plane rotation is done at the stage when the imaginary parts of the eigenvalues are being recorded in WI. We want to choose $c = \cos\theta$ and $s = \sin\theta$ so that

$$\alpha c^2 + (\beta+\gamma)cs + \delta s^2 = \delta c^2 - (\beta+\gamma)cs + \alpha s^2 \quad .$$

Hence

$$\tan 2\theta = \frac{2sc}{c^2-s^2} = -\frac{2p}{\sigma} = \frac{2|p|}{|\sigma|}\,\mathrm{sign}(-p\sigma) \ ,$$

$$\sigma = \beta+\gamma, \quad p = (\alpha-\delta)/2 \quad .$$

Let

$$\tau = \sqrt{\sigma^2+4p^2} \quad .$$

Then

$$\cos\theta = q = \sqrt{\tfrac{1}{2}(1+\cos 2\theta)} = \sqrt{(1+|\sigma|/\tau)/2} \ ,$$

$$\sin\theta = \sin 2\theta/2\cos\theta = |p|\,\mathrm{sign}(-p\sigma)/\tau q \quad .$$

Our program does not force the subdiagonal element of a 2×2 diagonal block to be positive.

```
      SUBROUTINE PERMS(NM,N,A,P,LOW,IGH,RABS,SCALE)
      DIMENSION A(NM,N), P(NM,N), RABS(N)
      INTEGER SCALE(N)
C*********************INITIALIZE SCALE*********************************
      DO 10 K = 1,N
   10 SCALE(K) = K
      K = 1
      L = N
      GOTO 100
C****************IN-LINE PROCEDURE FOR ROW AND
C                     COLUMN INTERCHANGE************************************
   20 IF (J.EQ.M) GOTO 60
C****************RECORD TRANSFORMATION IN SCALE*************************
      I = SCALE(J)
      SCALE(J) = SCALE(M)
      SCALE(M) = I
C
C******   EXCHANGE COLUMN J AND M
      DO 40 I = 1,L
         F = A(I,J)
         A(I,J) = A(I,M)
         A(I,M) = F
   40 CONTINUE
C*******EXCHANGE ROW J AND M
      DO 50 I = K,N
        F = A(J,I)
        A(J,I) = A(M,I)
        A(M,I) = F
   50 CONTINUE
C
   60 GOTO (80,130,220),IEXC
C ******   SEARCH FOR ROWS ISOLATING AN EIGENVALUE
C          AND PUSH THEM DOWN****************************
   80 IF (L.EQ.1) GOTO 300
      L = L - 1
C******** FOR J = L STEP -1 UNTIL 1 DO
  100 DO 120 JJ = 1 , L
        J = L + 1 - JJ
        DO 110 I = 1,L
           IF (I.EQ.J) GOTO 110
           IF (A(J,I).NE.0.0) GOTO 120
  110   CONTINUE
C***********THE JTH ROW ISOLATES AN EIGENVALUE,EXCHANGE IT WITH ROW L
C
        M = L
        IEXC = 1
        GOTO 20
  120 CONTINUE
C
      GOTO 140
C   *********** SEARCH FOR COMUMNS ISOLATING AN EIGENVALUE
C               AND PUSH THEM LEFT *****************
  130 K = K + 1


C
  140 DO 170 J = K , L
        DO 150 I = K,L
           IF (I.EQ.J) GOTO 150
           IF (A(I,J).NE.0.0) GOTO 170
  150   CONTINUE
C***** COLMUMN J ISOLATES AN EIGENVALUE,EXCHANGE IT WITH COMUMN K
        M = K
        IEXC = 2
        GOTO 20
  170 CONTINUE
```

4. Programs and Facing Comments

PERMS is an adaptation of the EISPACK routine BALANC. See BALANC where no comments are given.

A contains the matrix to be reduced to Schur form. The transformations (elementary permutations) are gathered in P. SCALE is an integer vector used as working space to record the transformations. For RABS (also working space) WI can be used (see Section 5).

Statement	
10+1	K will become LOW and so starts at 1. L will become IGH and so starts at N.
40-4 50-4	The indices of the DO loops take into account that the matrix already has the block upper triangular structure shown in Figure 1 (Section 3).
80	If L reaches 1, the matrix is upper triangular and we need not search the columns.
100+2	We only need search the submatrix in columns 1 through L.
140	We only need to search the submatrix in rows K through L.

```
C**************BRING ROWS OF LARGE 1 NORM TO THE TOP********************
C
C*****************COMPUTE 1 NORM OF ROWS FROM LOW TO IGH***************
      DO 210 I = K,L
         RABS(I) = 0.0
            DO 200 J = K,L
  200       RABS(I) = RABS(I) + ABS(A(I,J))
  210 CONTINUE
C
      IEXC = 3
C
C**********BUBBLE SORTING,EXCHANGING ROWS J+1 AND J IF
C              RABS(J+1).GT.2*RABS(J)****************************************
      LOWP1 = K + 1
C     FOR IP = IGH - 1 STEP -1 UNTIL LOW DO
      DO 230 IQ = LOWP1,L
         IP = K + L - IQ
         KCOUNT = 0
         DO 220 J = K,IP
            M = J + 1
            IF (RABS(M).LE.2.0*RABS(J)) GOTO 220
            F = RABS(M)
            RABS(M) = RABS(J)
            RABS(J) = F
            KCOUNT = KCOUNT + 1
            GOTO 20
  220    CONTINUE
         IF (KCOUNT.EQ.0.0) GOTO 300
  230 CONTINUE
C***************FORM PERMUTATION MATRIX P      **************
  300 DO 320 J = 1,N
         DO 310 I = 1,N
           P(I,J) = 0.0
  310    CONTINUE
         M = SCALE(J)
        P(M,J) = 1.0
  320 CONTINUE
C
      LOW = K
      IGH = L
      RETURN
      END
```

Statement

200 The 1 norm is computed only for the vector in columns LOW through IGH of the given row.

210+2 In regular "bubble" sorting, at the end of the IPth step, the smallest element among elements LOW,LOW+1,...,IP+1, ends up in position IP + 1. Here an exchange is made only when RABS(J+1) .GT. 2*RAB(J), i.e. a factor 2 is inserted. This factor can be changed if the user desires. KCOUNT indicates the number of exchanges made in the IPth step. If KCOUNT = 0, no exchanges have been made and we stop the sort. If lines J and M are exchanged, the corresponding interchange must be made in RABS. Since IEXC = 3, the inline procedure returns to 220 after an exchange. The indices K and L are already correctly set for the in-line procedure.

300 The (SCALE(J),J) element of the permutation matrix P is set to 1.

320 Note that L > K, unless A has been permuted into an upper triangular matrix, in which case L = K = 1.

```
      SUBROUTINE ORTHAN(NM,N,LOW,IGH,A,P,ORT)
      REAL A(NM,N),P(NM,N),ORT(IGH)
C
      LA = IGH - 1
      KP1 = LOW + 1
      IF (LA.LT.KP1) GOTO 300
C
      DO 200 M = KP1,LA
         H = 0.0
         ORT(M) = 0.0
         SCALE = 0.0
C*********SCALE COLUMN ( ALGOL TOL THEN NOT NEEDED)*************
         DO 90 I = M,IGH
   90    SCALE = SCALE + ABS(A(I,M-1))
      IF (SCALE.EQ.0.0) GOTO 200
         MP = M + IGH
C*********FOR I = IGH STEP -1 UNTIL M DO ..*****************************
         DO 100 II = M,IGH
            I = MP - II
            ORT(I) = A(I,M-1)/SCALE
            H = H + ORT(I)*ORT(I)
  100    CONTINUE
         G = -SIGN(SQRT(H),ORT(M))
         H = H - ORT(M)*G
         ORT(M) = ORT(M) - G
C********* FORM (I-U*UT)/H) * A *************************
         DO 130 J = M,N
            F = 0.0
C***************FOR I = IGH STEP -1 UNTIL M DO .. ****************
            DO 110 II = M,IGH
               I = MP - II
               F = F + ORT(I)*A(I,J)
  110       CONTINUE
C
            F = F/H
C
            DO 120 I = M,IGH
  120       A(I,J) = A(I,J) - F*ORT(I)
C
  130    CONTINUE
C  ***** FORM (I-(U*UT)/H)*A*(I-(U*UT)/H) ***********
         DO 160 I = 1,IGH
            F = 0.0
C***********FOR J = IGH STEP -1 UNTIL M DO ****************
            DO 140 JJ = M,IGH
               J = MP - JJ
               F = F + A(I,J)*ORT(J)
  140       CONTINUE
C
            F = F/H
C
            DO 150 J = M,IGH
  150       A(I,J) = A(I,J) - F*ORT(J)

  160    CONTINUE
C***************ACCUMULATE TRANSFORMATION  ***************************
         DO 190 I = 1,N
            F = 0.0
            DO 170 J = M,IGH
  170       F = F + P(I,J)*ORT(J)
C
            F = F/H
C
            DO 180 J = M,IGH
  180       P(I,J) = P(I,J) - F*ORT(J)
  190    CONTINUE
         A(M,M-1) = SCALE*G
  200 CONTINUE
  300 RETURN
      END
```

In this adaptation of the EISPACK routine ORTHES the transformations are post multiplied into P, which on input contains the output of PERMS. The array WI can be used for ORT. See Section 5. Since the matrix A is block triangular, the index I at 130+1 need only run to IGH, whereas at 160+1 the index I runs to N since P is not of this structure.

```
      SUBROUTINE HQR3(NM,N,LOW,IGH,H,V,WI,IERR)
      DIMENSION H(NM,N),V(NM,N),WI(N)
      REAL NORM,MACHEP
      INTEGER EN,ENM2
      LOGICAL NOTLAS
      DATA MACHEP     /01642400000000000000/
C  POSTMULTIPLY TRANSFORMATIONS,I.E. SCHUR FORM = VT   A V
C
C  SET WI TO ZERO AND CHECK FOR TRIANGULARITY
C
      DO 50 I = 1,N
   50 WI(I) = 0.0
      IF (LOW.EQ.IGH) GOTO 400
      IERR = 0
C
      EN = IGH
      T = 0.0
C
C   SEARCH FOR NEXT EIGENVALUES
C
C           TEST FOR END CONDITION
   60 IF  (EN.LT.LOW) GOTO 400
      ITS = 0
      NA = EN - 1
      ENM2 = NA - 1
C
C   LOOK FOR SINGLE SMALL SUB-DIAGONAL ELEMENT
C      FOR L=EN STEP -1 UNTIL LOW DO
C
   70 IF (EN.EQ.LOW) GOTO 90
      DO 80 LL=LOW,NA
         L=EN+LOW-LL
         IF(ABS(H(L,L-1)).LE.MACHEP*(ABS(H(L-1,L-1))
     X            + ABS(H(L,L))))GO TO 100
   80 CONTINUE
   90 L = LOW
C
C      FORM SHIFT
C
  100 X = H(EN,EN)
      IF (L.EQ.EN) GOTO 270
      Y = H(NA,NA)
      W = H(EN,NA) * H(NA,EN)
      IF (L.EQ.NA) GOTO 300
      IF (ITS.EQ.30) GOTO 1000
      IF (ITS.NE.10 .AND. ITS.NE.20) GOTO 130
C
C     FORM EXCEPTIONAL SHIFT
      Y = X
      S = ABS(H(EN,NA)) + ABS(H(NA,ENM2))
C
      T = T + X
C

      DO 120 I = LOW ,EN
  120 H(I,I) = H(I,I) - X
      X = 0.75 * S
      W = -0.4275*S*S
  130 ITS = ITS + 1
```

This is an adaptation of the EISPACK routine HQR2. It is the matrix to be reduced to Schur form. The transformations are post multiplied into V, which on input contains the output P of ORTHAN. WI contains the imaginary parts of the eigenvalues. For a complex eigenvalue, the positive imaginary part appears first. The use of WI is to indicate when there is a non-zero subdiagonal element (in which case WI(J) > 0) of the Schur form.

<u>Statement</u>

50	WI must be initialized to zero for the case when LOW = IGH = 1, i.e., when the matrix is already upper triangular. In this case, no QR steps need be performed and we go directly to 400.

```
C     LOOK FOR TWO CONSECUTIVE SMALL SUB-DIAGONAL
C       ELEMENTS.   FOR M=EN-2 STEP -1 UNTIL L DO
C
        DO 140 MM = L,ENM2
              M = ENM2 + L - MM
              ZZ = H(M,M)
              R = X - ZZ
              S = Y - ZZ
              P = (R*S -W)/H(M+1,M)    +    H(M,M+1)
              Q = H(M+1,M+1) - ZZ - R - S
              R = H(M+2,M+1)
              S = ABS(P) + ABS(Q) + ABS(R)
              P = P/S
              Q = Q/S
              R = R/S
              IF (M.EQ.L) GOTO 150
              IF (ABS(H(M,M-1))*(ABS(Q) + ABS(R)).LE.MACHEP*ABS(P)
     X        *(ABS(H(M-1,M-1)) + ABS(ZZ) + ABS(H(M+1,M+1)))) GOTO 150
  140 CONTINUE
C
  150 MP2 = M + 2
C
        DO 160 I = MP2,EN
              H(I,I-2) = 0.0
              IF (I.EQ.MP2) GOTO 160
              H(I,I-3) = 0.0
  160 CONTINUE
C
C     DOUBLE QR STEP INVOLVING ROWS L TO EN
C       AND COLUMNS M TO EN.
C
        DO 260 K = M,NA
              NOTLAS = K.NE.NA
              IF (K.EQ.M) GOTO 170
              P = H(K,K-1)
              Q = H(K+1,K-1)
              R = 0.0
              IF (NOTLAS) R = H(K+2,K-1)
              X = ABS(P) + ABS(Q) + ABS(R)
              IF (X.EQ.0.0) GOTO 260
              P = P/X
              Q = Q/X
              R = R/X
  170         S = SIGN(SQRT(P*P + Q*Q + R*R),P)
              IF (K.EQ.M) GOTO 180
              H(K,K-1) = -S*X
              GOTO 190
  180         IF (L.NE.M) H(K,K-1) = -H(K,K-1)


  190         P = P + S
              X = P/S
              Y = Q/S
              ZZ = R/S
              Q = Q/P
              R = R/P
```

No comments for this section. See EISPACK.

```
C     ROW MODIFICATION
C
             DO 210 J = K,N
                P = H(K,J) + Q*H(K+1,J)
                IF (.NOT.NOTLAS) GOTO 200
                P = P + R*H(K+2,J)
                H(K+2,J) = H(K+2,J) - P*ZZ
  200           H(K+1,J) = H(K+1,J) - P*Y
                H(K,J) = H(K,J) - P*X
  210        CONTINUE
C
             J = MIN0(EN,K+3)
C
C     COLUMN MODIFICATION
C
             DO 230 I = 1,J
                P = X*H(I,K) + Y*H(I,K+1)
                IF (.NOT.NOTLAS) GOTO 220
                P = P + ZZ*H(I,K+2)
                H(I,K+2) = H(I,K+2) - P*R
  220           H(I,K+1) = H(I,K+1) - P*Q
                H(I,K) = H(I,K) - P
  230        CONTINUE
C
C     ACCUMULATE TRANSFORMATIONS
C
          DO 250 I = 1,N
                P = X*V(I,K) + Y*V(I,K+1)
                IF (.NOT.NOTLAS) GOTO 240
                P = P + ZZ*V(I,K+2)
                V(I,K+2) = V(I,K+2) - P*R
  240           V(I,K+1) = V(I,K+1) - P*Q
                V(I,K) = V(I,K) - P
  250        CONTINUE
  260 CONTINUE
      GO TO 70
C
C    ONE ROOT FOUND
C
  270 H(EN,EN)=X+T
      WI(EN)=0.0
C
  290 EN = NA
      GOTO 60
```

Statement

200 The indices J and I at 200-5 and 210+2 take into account the fact that H is upper Hessenberg, whereas at 230+1 the index I runs from 1 to N, since V has no special structure.

```
C   TWO ROOTS FOUND
C

  300 P = (Y-X)/2.0
      Q = P*P + W
      ZZ = SQRT(ABS(Q))
      H(EN,EN) = X + T
      X = H(EN,EN)
      H(NA,NA) = Y + T
      IF (Q.LT.0.0) GOTO 310
      ZZ = P + SIGN(ZZ,P)
C
C     REAL PAIR
C
      WI(NA) = 0.0
      WI(EN) = 0.0
      X = H(EN,NA)
      R = SQRT(X*X + ZZ*ZZ)
      P = X/R
      Q = ZZ/R
      GOTO 320
C
C    COMPLEX PAIR
C
  310 WI(NA) = ZZ
      WI(EN) = -ZZ
C
C   MAKE DIAGONAL ELEMENTS EQUAL
C
      IF (P.EQ.0.0) GOTO 380
      BPC = H(EN,NA) + H(NA,EN)
      TX = SQRT(BPC*BPC + 4.0* P*P)
      Q = SQRT(.5 * (1.0 + ABS(BPC)/TX))
      P = SIGN(P/(Q*TX),-BPC*P)
C
C     ROW MODIFICATION
C
  320 DO 330 J = NA,N
          ZZ = H(NA,J)
          H(NA,J) = Q*ZZ + P*H(EN,J)
          H(EN,J) = Q*H(EN,J) - P*ZZ
  330 CONTINUE
C
C      COLUMN MODIFICATION
C
      DO 340 I = 1,EN
          ZZ = H(I,NA)
          H(I,NA) = Q*ZZ + P*H(I,EN)
          H(I,EN) = Q*H(I,EN) - P*ZZ
  340 CONTINUE
C
C     ACCUMULATE TRANSFORMATIONS
C
      DO 350 I = 1,N
          ZZ = V(I,NA)
          V(I,NA) = Q*ZZ + P*V(I,EN)
          V(I,EN) = Q*V(I,EN) - P*ZZ

  350 CONTINUE
```

Statement

310+2 See Section 3, step 3, where the rotation for making the diagonal elements of the 2 by 2 block equal is explained.

The section of program from 320 to 350 performs the plane rotation for either of two cases: when a real pair is found and H(EN,NA) is to be zeroed, or when the diagonal elements of a complex block are being made equal. In the former case P and Q are set at 310-3, in the latter at 320-2.

The limits of the J index at 320 and the I indices at 330+1 and 340+1 take into account the fact that H is upper Hessenberg whereas V is not.

```
  380 EN = ENM2
      GOTO 60
C
C     ZERO H BELOW BLOCK DIAGONAL
C
  400 IF (N.LT.3) RETURN
      IF (WI(N-1).EQ.0.0) H(N,N-1) = 0.0
      DO 420 J = 3,N
         JM2 = J - 2
         IF (WI(JM2).LE.0.0) H (J-1,JM2) = 0.0
         DO 410 I = J,N
  410    H(I,JM2) = 0.0
  420 CONTINUE
      RETURN
 1000 IERR = EN
      RETURN
      END
```

The section of program from 400 to 420 which zeroes H below the block diagonal takes into account the block structure of the matrix.

<u>Statement</u>

400	If N < 3 there is nothing to be done and an out of range index for WI must be avoided.
1000	If IERR > 0, after 30 iterations, the IERRth eigenvalue is not isolated and the Schur form is not found, but WI(J) is correct for J = IERR+1,...,N.

5. Usage

```
DIMENSION H(24,24),P(24,24),WI(24)
INTEGER SCALE(24)
NM = 24
N = 6
```

Enter H

```
CALL PERMS(NM,N,H,P,LOW,IGH,WI,SCALE)
CALL ORTHAN(NM,N,LOW,IGH,H,P,WI)
CALL HQR3(NM,N,LOW,IGH,H,P,WI,IERR)
```

Operation Count

One operation means a multiplication or division followed by an addition or subtraction. Counts are taken from the program

PERMS: no arithmetic operations, only comparisons

ORTHAN: At the m^{th} major step column m-1 is reduced to Hessenberg form.

Formation of the vector u in $1-\gamma uu^T$: $n-m+1$

Row operations: $\sum_{j=m}^{n} \{2(n-m+1)+1\} = (n-m+1)[2(n-m+1)+1]$

Column operations: $\sum_{j=1}^{n} \{2(n-m+1)+1\} = n[2(n-m+1)+1]$

Accumulate transforms: $\sum_{j=1}^{n} \{2(n-m+1)+1\} = n[2(n-m+1)+1]$

Set element (m,m-1): 1

Summing these quantities for $m = 2,\dots,n-1$ yields

$$\sum_{\ell=2}^{n-1} [\ell + (\ell+2n)(2\ell+1)+1] = \frac{8}{3}n^3 - 3n^2 + O(n) \ .$$

HQR3: A typical QR transformation acts on the leading $j \times j$ submatrix of a Hessenberg matrix. To restore column k to Hessenberg form requires the following calculations:

Computation	Key values	Rows	Columns	Accumulate
Count	9	$\sum_{\ell=k}^{n} 5$	$\sum_{\ell=k}^{\min(k+3,j)} 5$	$\sum_{\ell=1}^{n} 5$

Subtotal for the $j \times j$ submatrix:

$$\sum [9 + 5(n-k+1) + 5(k+3) + 5n] = 10nj + 29j$$

Assuming b iterations per eigenvalue the total is

$$[5n^3 + 20n^2 + 0(n)]b \ .$$

Realistic value for b is about 1.5.

<u>GRAND TOTAL</u> (for the real Schur form): $10n^3 + 30n^2 + 0(n)$

Input H

-9.0000	21.0000	-15.0000	4.0000	2.0000	0.
-10.0000	21.0000	-14.0000	4.0000	2.0000	0.
-8.0000	16.0000	-11.0000	4.0000	2.0000	0.
-6.0000	12.0000	-9.0000	3.0000	3.0000	0.
-4.0000	8.0000	-6.0000	0.	5.0000	0.
-2.0000	4.0000	-3.0000	0.	1.0000	3.0000

Output H of PERMS

3.0000	4.0000	-3.0000	0.	-2.0000	1.0000
0.	21.0000	-14.0000	4.0000	-10.0000	2.0000
0.	16.0000	-11.0000	4.0000	-8.0000	2.0000
0.	12.0000	-9.0000	3.0000	-6.0000	3.0000
0.	21.0000	-15.0000	4.0000	-9.0000	2.0000
0.	8.0000	-6.0000	0.	-4.0000	5.0000

Output P of PERMS

0.	0.	0.	0.	1.0000	0.
0.	1.0000	0.	0.	0.	0.
0.	0.	1.0000	0.	0.	0.
0.	0.	0.	1.0000	0.	0.
0.	0.	0.	0.	0.	1.0000
1.0000	0.	0.	0.	0.	0.

Schur Form $\hat{S}$

3.0000	.1124	-2.4164	-4.1948	2.4259	-.8165
0.	2.0000	22.9544	35.7438	-19.0808	14.2995
0.	-.0436	2.0000	.6376	-1.5265	.0143
0.	0.	0.	1.0000	1.7778	.5158
0.	0.	0.	0.	3.0000	2.5997
0.	0.	0.	0.	0.	1.0000

Final Transformation Matrix P

0.	-.6003	.7997	.0000	-.0000	.0000
0.	-.5442	-.4085	-.7328	0.	0.
0.	-.4353	-.3268	.5054	-.5307	.4082
0.	-.3265	-.2451	.3790	.1516	-.8165
0.	-.2177	-.1634	.2527	.8339	.4082
1.0000	0.	0.	0.	0.	0.

$P\hat{S}P^T = H$

-9.0000	21.0000	-15.0000	4.0000	2.0000	0.
-10.0000	21.0000	-14.0000	4.0000	2.0000	0.
-8.0000	16.0000	-11.0000	4.0000	2.0000	0.
-6.0000	12.0000	-9.0000	3.0000	3.0000	0.
-4.0000	8.0000	-6.0000	-.0000	5.0000	0.
-2.0000	4.0000	-3.0000	-.0000	1.0000	3.0000

$PP^T = I$

1.0000	.0000	.0000	.0000	.0000	0.
.0000	1.0000	-.0000	-.0000	-.0000	0.
.0000	-.0000	1.0000	-.0000	-.0000	0.
.0000	-.0000	-.0000	1.0000	-.0000	0.
.0000	-.0000	-.0000	-.0000	1.0000	0.
0.	0.	0.	0.	0.	1.0000

6. Numerical Example

This example [2] demonstrates the use of PERMS and HQR3. First the isolated eigenvalue of the last column is detected, and the first and sixth columns are exchanged. Hence LOW = 2 and IGH = 6. Since the last row's norm is now twice the fifth rows, these are exchanged.

The eigenvalues are 3, 2+i, 2-i, 1, 3, 1, in order given along the block diagonal of $\hat{S}$. The standardized two by two block appears in second position along the diagonal. As a check we have also computed $PSP^T = H$ and $PP^T = I$.

References

[1] EISPACK Guide, Lecture Notes in Computer Science No. 6, Springer-Verlag (1974).

[2] Gregory, T. and Karney, L. A Collection of Matrices for Testing Computational Algorithms, Example 5.26, p. 108.

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*

Computer Science Division
University of California
Berkeley, California 94720

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

3. REPORT TITLE

A PROGRAM TO COMPUTE THE REAL SCHUR FORM OF A REAL SQUARE MATRIX

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Scientific Final

5. AUTHOR(S) *(First name, middle initial, last name)*

B.N. Parlett
R. Feldman

6. REPORT DATE

June 1975

7a. TOTAL NO. OF PAGES

28

7b. NO. OF REFS

2

8a. CONTRACT OR GRANT NO.

ONR-N00014-69-A-0200-1017

b. PROJECT NO.

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

Electronics Research Laboratory
Memorandum M-526

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY

Mathematics Branch
Office of Naval Research
Washington, D.C. 20360

13. ABSTRACT

A Fortran program is presented which will obtain the real Schur form of a real $n \times n$ matrix in $10n^3 + 30n^2$ multiplications (approximately).