FILE: a2by2matrix.F

```
> diagonalization of a 2x2 matrix
>
>
>      | 0     T   |
>  H = |           |
>      | T   DELTA |
>
> diagonalization routine: DIAG(IA,N,A,R,E,A,IFAIL)
> N = dimension of matrix A,  IFAIL = check varaible
> before diagonalization : A = H
> after  diagobalization : A = set of eigenvectore
> eigen values           : R

================================================================================
C     * TEST WITH DIAGONALIZATION *
      PARAMETER (D=2)
      REAL    A(D,D),R(D),E(D)
      INTEGER IA,N,IV,IFAIL
      INTEGER I,J,K,L
      REAL    DELTA,T
C
      IA    = D
      N     = D
      IV    = D
      IFAIL = 0
C
      PRINT *,'* DIAGONALIZATION OF A 2X2 MATRIX *'
C
      WRITE(6,*)
      WRITE(6,*) 'ENTER DELTA AND T :'
      READ(5,*) DELTA,T
C
      DO 15 K=1,D
      R(K) = 0.
      E(K) = 0.
      DO 20 L=1,D
      A(K,L) = 0.
20    CONTINUE
15    CONTINUE
C
      A(1,1)    = 0.
      A(2,2)    = DELTA
      A(2,1)    = T
C
      CALL DIAG(IA,N,A,R,E,A,IFAIL)
      IF (IFAIL.EQ.0) GO TO 30
      WRITE(6,40) IFAIL
40    FORMAT(' ERROR IN DIAG,  IFAIL =',I2)
      WRITE(6,*) ' '
      WRITE(6,*) '************************************'
      WRITE(6,*) '     ERROR IN DIAGONALIZATION'
      WRITE(6,*) '************************************'
      STOP
30    CONTINUE
C
      WRITE(6,*)
      DO 80 I=1,N
      WRITE(6,*)
      WRITE(6,*) 'EIGEN VALUE NR : ',I,'            ENERGY : ',R(I)
      WRITE(6,*) 'EIGEN VECTOR   : '
      DO 90 J=1,N
```

```
      WRITE(6,*) '                  ', A(J,I)
90    CONTINUE
80    CONTINUE
C
      WRITE(6,*) '***********************************'
      WRITE(6,*) '  DIAGONALIZATION PROGRAM FINISHED'
      WRITE(6,*) '***********************************'
      STOP
      END
C
C
C
      SUBROUTINE DIAG(KMX,NLS,VECT,EIG,FK,C,M)
      DIMENSION VECT(KMX,KMX),EIG(KMX),FK(KMX),C(KMX,KMX)
      CALL TRED2 (KMX,NLS,C,EIG,FK,VECT)
      CALL TQL2  (KMX,NLS,EIG,FK,C,M)
      RETURN
      END

C#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#

      SUBROUTINE TRED2(NM, N, A, D, E, Z)
      DIMENSION A(NM,N), D(N), E(N), Z(NM,N)
      DO 20 I=1,N
C
         DO 10 J=1,I
            Z(I,J) = A(I,J)
 10      CONTINUE
 20   CONTINUE
C
      IF (N.EQ.1) GO TO 160
C     ********** FOR I=N STEP -1 UNTIL 2 DO -- **********
      DO 150 II=2,N
         I = N + 2 - II
         L = I - 1
         H = 0.0
         SCALE = 0.0
         IF (L.LT.2) GO TO 40
C     ********** SCALE ROW (ALGOL TOL THEN NOT NEEDED) **********
         DO 30 K=1,L
            SCALE = SCALE + ABS(Z(I,K))
 30      CONTINUE
C
         IF (SCALE.NE.0.0) GO TO 50
 40      E(I) = Z(I,L)
         GO TO 140
C
 50      DO 60 K=1,L
            Z(I,K) = Z(I,K)/SCALE
            H = H + Z(I,K)*Z(I,K)
 60      CONTINUE
C
         F = Z(I,L)
         G = -SIGN(SQRT(H),F)
         E(I) = SCALE*G
         H = H - F*G
         Z(I,L) = F - G
         F = 0.0
C
         DO 100 J=1,L
            Z(J,I) = Z(I,J)/(SCALE*H)
            G = 0.0
C     ********** FORM ELEMENT OF A*U **********
```

```
            DO 70 K=1,J
               G = G + Z(J,K)*Z(I,K)
 70         CONTINUE
C
            JP1 = J + 1
            IF (L.LT.JP1) GO TO 90
C
            DO 80 K=JP1,L
               G = G + Z(K,J)*Z(I,K)
 80         CONTINUE
C     ********** FORM ELEMENT OF P **********
 90         E(J) = G/H
            F = F + E(J)*Z(I,J)
 100     CONTINUE
C
         HH = F/(H+H)
C     ********** FORM REDUCED A **********
         DO 120 J=1,L
            F = Z(I,J)
            G = E(J) - HH*F
            E(J) = G
C
            DO 110 K=1,J
               Z(J,K) = Z(J,K) - F*E(K) - G*Z(I,K)
 110        CONTINUE
 120     CONTINUE
C
         DO 130 K=1,L
            Z(I,K) = SCALE*Z(I,K)
 130     CONTINUE
C
 140     D(I) = H
 150  CONTINUE
C
 160  D(1) = 0.0
      E(1) = 0.0
C     ********** ACCUMULATION OF TRANSFORMATION MATRICES **********
      DO 220 I=1,N
         L = I - 1
         IF (D(I).EQ.0.0) GO TO 200
C
         DO 190 J=1,L
            G = 0.0
C
            DO 170 K=1,L
               G = G + Z(I,K)*Z(K,J)
 170        CONTINUE
C
            DO 180 K=1,L
               Z(K,J) = Z(K,J) - G*Z(K,I)
 180        CONTINUE
 190     CONTINUE
C
 200     D(I) = Z(I,I)
         Z(I,I) = 1.0
         IF (L.LT.1) GO TO 220
C
         DO 210 J=1,L
            Z(I,J) = 0.0
            Z(J,I) = 0.0
 210     CONTINUE
C
 220  CONTINUE
```

```
C
      RETURN
      END

C #*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#*#

      SUBROUTINE TQL2(NM, N, D, E, Z, IERR)
      DIMENSION D(N), E(N), Z(NM,N)
      REAL MACHEP
C               **********
      MACHEP = 2.**(-46)
C     ********** MACHEP IS A MACHINE DEPENDENT PARAMETER SPECIFYING
C                THE RELATIVE PRECISION OF FLOATING POINT ARITHMETIC.
C
      IERR = 0
      IF (N.EQ.1) GO TO 160
C
      DO 10 I=2,N
         E(I-1) = E(I)
 10   CONTINUE
C
      F = 0.0
      B = 0.0
      E(N) = 0.0
C
      DO 110 L=1,N
         J = 0
         H = MACHEP*(ABS(D(L))+ABS(E(L)))
         IF (B.LT.H) B = H
C     ********** LOOK FOR SMALL SUB-DIAGONAL ELEMENT **********
         DO 20 M=L,N
            IF (ABS(E(M)).LE.B) GO TO 30
C     ********** E(N) IS ALWAYS ZERO, SO THERE IS NO EXIT
C                THROUGH THE BOTTOM OF THE LOOP **********
 20      CONTINUE
 30      IF (M.EQ.L) GO TO 100
 40      IF (J.EQ.30) GO TO 150
         J = J + 1
C     ********** FORM SHIFT **********
         P = (D(L+1)-D(L))/(2.0*E(L))
         R = SQRT(P*P+1.0)
         H = D(L) - E(L)/(P+SIGN(R,P))
C
         DO 50 I=L,N
            D(I) = D(I) - H
 50      CONTINUE
C
         F = F + H
C     ********** QL TRANSFORMATION **********
         P = D(M)
         C = 1.0
         S = 0.0
         MML = M - L
C     ********** FOR I=M-1 STEP -1 UNTIL L DO -- **********
         DO 90 II=1,MML
            I = M - II
            G = C*E(I)
            H = C*P
            IF (ABS(P).LT.ABS(E(I))) GO TO 60
            C = E(I)/P
            R = SQRT(C*C+1.0)
            E(I+1) = S*P*R
            S = C/R
```

```
            C = 1.0/R
            GO TO 70
 60         C = P/E(I)
            R = SQRT(C*C+1.0)
            E(I+1) = S*E(I)*R
            S = 1.0/R
            C = C*S
 70         P = C*D(I) - S*G
            D(I+1) = H + S*(C*G+S*D(I))
C     ********** FORM VECTOR **********
            DO 80 K=1,N
               H = Z(K,I+1)
               Z(K,I+1) = S*Z(K,I) + C*H
               Z(K,I) = C*Z(K,I) - S*H
 80         CONTINUE
C
 90      CONTINUE
C
         E(L) = S*P
         D(L) = C*P
         IF (ABS(E(L)).GT.B) GO TO 40
 100     D(L) = D(L) + F
 110  CONTINUE
C     ********** ORDER EIGENVALUES AND EIGENVECTORS **********
      DO 140 II=2,N
         I = II - 1
         K = I
         P = D(I)
C
         DO 120 J=II,N
            IF (D(J).GE.P) GO TO 120
            K = J
            P = D(J)
 120     CONTINUE
C
         IF (K.EQ.I) GO TO 140
         D(K) = D(I)
         D(I) = P
C
         DO 130 J=1,N
            P = Z(J,I)
            Z(J,I) = Z(J,K)
            Z(J,K) = P
 130     CONTINUE
C
 140  CONTINUE
C
      GO TO 160
C     ********** SET ERROR -- NO CONVERGENCE TO AN
C                EIGENVALUE AFTER 30 ITERATIONS **********
 150  IERR = L
 160  RETURN
C     ********** LAST CARD OF TQL2 **********
      END
```