ALGORITHM 496
The LZ Algorithm To Solve the Generalized Eigenvalue Problem for Complex Matrices [F2]

LINDA KAUFMAN

University of Colorado

Key Words and Phrases: eigenvalues, generalized eigenvalue problem
CR Categories: 5.14
Language: Fortran

DESCRIPTION

The program given here is an implementation of the LZ algorithm [1] for finding $\mathbf{x}$ and λ such that $A\mathbf{x} = \lambda B\mathbf{x}$, where A and B are $n \times n$ matrices. The LZ algorithm does not require matrix inversion and may be used when either or both matrices are singular. It is a generalization of Rutishauser's LR method [3] for the standard eigenvalue problem $A\mathbf{x} = \lambda\mathbf{x}$ and closely resembles the QZ algorithm given by Moler and Stewart [2] for the generalized problem given above. Since the LZ algorithm uses elementary transformations and the QZ algorithm uses orthogonal transformations, the LZ algorithm is probably more efficient when either A or B is complex.

The LZ algorithm is based on three observations:

(1) If L and M are nonsingular matrices, the eigenvalue problems $LAM\mathbf{y} = \lambda LBM\mathbf{y}$ and $A\mathbf{x} = \lambda B\mathbf{x}$ have the same eigenvalues and their eigenvectors are related by $\mathbf{x} = M\mathbf{y}$.

(2) If A is a triangular matrix with diagonal elements α_i, and B is a triangular matrix with diagonal elements β_i, then for each i, $i = 1, 2, \ldots, n$, α_i/β_i is an eigenvalue of the generalized problem if $\beta_i \neq 0$.

(3) There exist matrices L and M such that LAM and LBM are upper triangular and L and M are the products of lower triangular and permutation matrices.

The aim of the LZ algorithm is to find the matrices L and M which reduce A and B simultaneously to triangular form. It accomplishes this aim in two phases.

(1) The first phase simultaneously reduces A to upper Hessenberg form (i.e. $a_{i,j} = 0$ for $i > j$) and B to upper triangular form by (a) reducing B to triangular form as in Gaussian elimination and applying the transformations to A, and (b) reducing A to upper Hessenberg form while preserving the triangularity of B by

Received 21 June 1973, 6 December 1974, and 15 April 1975.

Author's address: Department of Computer Science, University of Colorado, Boulder, CO 80302.

using row transformations to zero the elements below the subdiagonal of A and column transformations to zero the subdiagonal elements of B introduced by the row transformations.

(2) The second phase iteratively reduces A to upper triangular form while preserving the triangularity of B where each iteration is given by (a) finding a shift λ_k, (b) finding a stabilized transformation $\bar{L}_k$ such that $\bar{L}_k(A_k - \lambda_k B_k)$ is upper triangular, (c) finding a stabilized transformation $\bar{M}_k$ such that $\bar{L}_k B_k \bar{M}_k$ is upper triangular, (d) setting $A_{k+1} = \bar{L}_k A_k \bar{M}_k$ and $B_{k+1} = \bar{L}_k B_k \bar{M}_k$.

If $\bar{L}_k$ and $\bar{M}_k$ are constructed appropriately, A_{k+1} will be upper Hessenberg and hopefully some of its subdiagonal elements will be small enough to be considered zero.

The shift λ_k is used to hasten the convergence of the algorithm. In practice it is set to the solution of the lowest 2×2 subproblem on the diagonal of $A_k - \lambda B_k$ which has not been triangularized.

Each matrix $\bar{L}_k$ is a product of matrices $L_{k,n-1}L_{k,n-2} \ldots L_{k,2}L_{k,1}$ and each $\bar{M}_k$ is a product of matrices $M_{k,1}M_{k,2} \ldots M_{k,n-1}$, where each $L_{k,i}$ and each $M_{k,i}$ is a stabilized elementary transformation (see [4]). The matrix $L_{k,1}$ is designed to annihilate the (2, 1) element of $A_k - \lambda_k B_k$. For $i \geq 1$, $M_{k,i}$ zeros the ith subdiagonal element of B introduced by $L_{k,i}$, and for $i > 1$, $L_{k,i}$ zeros the $(i + 1, i - 1)$-th element of A introduced by $M_{k,i-1}$.

The LZ algorithm determines the eigenvectors of the problem by computing the eigenvectors of the triangularized A and B and multiplying them by the product of all the column transformations.

The Fortran program given here is designed to find the eigensystem for two complex matrices A and B and consists of two subroutines which must be called separately. A user should first issue a call to LZHES to reduce A to upper Hessenberg form and B to triangular form and then issue a call to LZHES to determine the eigensystem of the reduced matrices. If A and B have already been reduced to Hessenberg and triangular form, then calling LZHES is unnecessary. In this situation, if eigenvectors are requested, the matrix X should be set to the identity matrix. The subroutine LZIT does not return the eigenvalues of the problem but the diagonal elements of the triangularized matrices. The ith eigenvalue may be found by dividing EIGA(i) by EIGB(i). The user is warned that EIGB(i) may be zero. The eigenvectors produced by LZIT are normalized so that the modulus of their largest component is 1.

Entrance to LZHES is attained by the statement

```
CALL LZHES(N,A,NA,B,NB,X,NX,WANTX)
```

where the input parameters are

N	order of the A and B matrices,
A	$n \times n$ complex matrix,
NA	row dimension of A,
B	$n \times n$ complex matrix,
NB	row dimension of B,
NX	row dimension of X (see output parameters)
WANTX	logical variable which should be set to .TRUE. if eigenvectors are wanted and otherwise set to .FALSE.,

and the output parameters are

A complex upper Hessenberg matrix; the original A matrix is destroyed,
B complex upper triangular matrix; the original B matrix is destroyed,
X $n \times n$ complex array containing the transformations needed to compute the eigenvectors of the problem if WANTX was .TRUE. .

Entrance to LZIT is attained by the statement

CALL LZIT(N,A,NA,B,NB,X,NX,WANTX,ITER,EIGA,EIGB)

where the input parameters are

N order of the A and B matrices,
A $n \times n$ complex upper Hessenberg matrix,
NA row dimension of the A matrix,
B $n \times n$ complex upper triangular matrix,
NB row dimension of the B matrix,
X $n \times n$ complex array containing the transformations to determine the eigenvectors of the original problem if WANTX has been set to .TRUE. ,
NX row dimension of the X matrix,
WANTX logical variable which should be set to .TRUE. if eigenvectors are wanted and otherwise set to .FALSE. ,

and the output parameters are

X $n \times n$ complex array whose ith column contains the ith eigenvectors if WANTX was set to .TRUE. ,
ITER integer array of length n whose ith entry contains the number of iterations needed to find the ith eigenvalue; for any i, if iter$(i) = -1$, then after 30 iterations there was not a sufficient decrease in the last subdiagonal element of A to continue iterating so that not all the eigenvalues have been isolated,
EIGA complex array of length n containing the diagonal of the triangularized A,
EIGB complex array of length n containing the diagonal of B.

Since $|x|$, where x is a complex scalar, must be computed at least $8n$ times per iteration, the execution time of the method is dependent on the algorithm used to compute $|x|$. Replacing the true modulus by an approximation has little effect on the numerical properties but may result in a significantly more efficient code. Several versions of the LZ algorithm with different methods for computing $|x|$ were run on the IBM 360 model 67 with the Fortran H compiler opt=2. The results were interesting:

(1) The procedure CABS supplied by the manufacturer to compute $|x|$ was by far the slowest.

(2) Substituting the procedure RABS given by

```
REAL FUNCTION RABS(X)
COMPLEX X, XX
REAL T(2)
EQUIVALENCE (XX, T(1))
XX = X
RABS = ABS(T(1)) + ABS(T(2))
RETURN
END
```

for CABS decreased the time by about 20 percent when no eigenvectors were computed. When using RABS, changing the precision of the program requires very little effort.

(3) Computing $|x|$ as ABS(REAL(X)) + ABS(AIMAG(X)) as in the program given here decreased execution times from version 2 by roughly 10 percent when no eigenvectors were computed.

Although theoretically numerical growth can be quite large, our experiments have shown that the algorithm behaves like most other methods which use stabilized elementary transformations: numerical growth rarely occurs and errors rarely accumulate. The eigenvalues are usually determined to the accuracy justified by the condition of the problem.

```
THE  MATRIX A:

 -238+  -344I     86+   178I    164+   240I   -166+  -308I     56+   158I
   76+   152I    -96+  -128I     40+   -32I     60+   184I    -60+  -136I
  118+   284I     55+  -182I    -13+   460I     34+  -192I   -176+  -214I
 -314+  -160I    132+    78I    114+   296I    -90+  -164I   -424+  -374I
  -54+   -24I   -205+  -400I    109+   148I    158+   312I    -38+   -96I

THE MATRIX B:

  388+    94I   -386+  -122I   -250+   -14I    556+   130I   -396+   -62I
 -304+   -76I    384+    64I   -160+    16I   -240+   -92I    240+    68I
 -658+  -136I    -73+   100I   -109+  -250I   -118+   100I    406+    96I
 -640+   -10I    204+   -42I   -692+   -90I    288+    66I   -192+   154I
 -162+   -72I    631+   158I    131+    52I   -758+  -184I    278+    76I


   TRUE EIGENVALUE
 1     7.6470588235294E-01+     9.4117647058823E-01 I
 2    -1.0000000000000E+00+    -1.3333333333333E+00 I
 3    -3.5294117647059E-01+    -4.1176470588235E-01 I
 4    -3.5294117647059E-01+    -4.1176470588235E-01 I
 5    -3.5294117647059E-01+     4.1176470588235E-01 I
ALPHA
 1    -3.1059278060929E+02+    -1.2380967798794E+03 I
 2     3.9432113020788E+01+     1.8552048946044E+02 I
 3     3.4303578717787E+02+     3.6267003407891E+01 I
 4     4.0331923195321E+01+    -4.1475768275521E+02 I
 5    -2.6483904149705E+01+    -1.2383248679301E+02 I
BETA
 1    -9.5389018503961E+02+    -4.4503094594735E+02 I
 2    -1.0324539562850E+02+    -4.7859961955782E+01 I
 3    -4.6241674938452E+02+     4.3672969795956E+02 I
 4     5.3226244802292E+02+     5.5417391177970E+02 I
 5    -1.4158479653056E+02+     1.8567644996121E+02 I
   COMPUTED EIGENVALUE
 1     7.6470588235296E-01+     9.4117647058821E-01 I
 2    -1.0000000000000E+00+    -1.3333333333333E+00 I
 3    -3.5294117647059E-01+    -4.1176470588233E-01 I
 4    -3.5294117647058E-01+    -4.1176470588236E-01 I
 5    -3.5294117647058E-01+     4.1176470588232E-01 I
          RELATIVE ERROR      RELATIVE RESIDUAL      NO. OF ITERATIONS
 1     2.7009971587330E-14     4.3138589276375E-15          0
 2     6.0291550413457E-15     6.7886927362343E-15          1
 3     3.9441522601135E-14     2.9742630452889E-15          0
 4     1.1809767484717E-14     6.0057593010225E-15          3
 5     5.6542864065059E-14     8.3080870692588E-15          1
```

Fig. 1

The example shown in Figure 1 was generated in integer arithmetic by multiplying two diagonal matrices by nonsingular transformations. It was run on the CDC Run compiler. The relative residual is the quantity,

$$|| \beta_i A\mathbf{x}_i - \alpha_i B\mathbf{x}_i ||_\infty / (|| \beta_i || \, || A ||_\infty + || \alpha_i || \, || B ||_\infty)$$

where α_i and β_i are the ith diagnonal elements of the triangularized A and B matrices and $\mathbf{x}_i$ is the ith eigenvector.

REFERENCES

1. KAUFMAN, L.C. The LZ algorithm to solve the generalized eigenvalue problem. *SIAM J. Numer. Anal. 11* (Oct. 1974), 997–1023.
2. MOLER, C.B , AND STEWART, G W. An algorithm for the generalized matrix eigenvalue problem. *SIAM J. Numer. Anal. 10* (April 1973), 241–256.
3. RUTISHAUSER, H. *Solution of the eigenvalue problem with the LR transformation.* Nat Bur. Standards Appl Math. Ser 49, U.S. Govt. Printing Office, Washington, D.C., Jan. 1958.
4. WILKINSON, J.H. *The Algebraic Eigenvalue Problem.* Oxford U. Press, New York, 1965.

ALGORITHM

```
      SUBROUTINE LZHES(N, A, NA, B, NB, X, NX, WANTX)                    LZH   10
C THIS SUBROUTINE REDUCES THE COMPLEX MATRIX A TO UPPER                   LZH   20
C HESSENBERG FORM AND REDUCES THE COMPLEX MATRIX B TO                     LZH   30
C TRIANGULAR FORM                                                         LZH   40
C INPUT PARAMETERS..                                                      LZH   50
C N   THE ORDER OF THE A AND B MATRICES                                   LZH   60
C A   A COMPLEX MATRIX                                                    LZH   70
C NA  THE ROW DIMENSION OF THE A MATRIX                                   LZH   80
C B   A COMPLEX MATRIX                                                    LZH   90
C NB  THE ROW DIMENSION OF THE B MATRIX                                   LZH  100
C NX  THE ROW DIMENSION OF THE X MATRIX                                   LZH  110
C WANTX A LOGICAL VARIABLE WHICH IS SET TO .TRUE. IF                      LZH  120
C       THE EIGENVECTORS ARE WANTED. OTHERWISE IT SHOULD                  LZH  130
C     BE SET TO .FALSE.                                                   LZH  140
C OUTPUT PARAMETERS..                                                     LZH  150
C A  ON OUTPUT A IS AN UPPER HESSENBERG MATRIX, THE                       LZH  160
C    ORIGINAL MATRIX HAS BEEN DESTROYED                                   LZH  170
C B  AN UPPER TRIANGULAR MATRIX, THE ORIGINAL MATRIX                      LZH  180
C    HAS BEEN DESTROYED                                                   LZH  190
C X  CONTAINS THE TRANSFORMATIONS NEEDED TO COMPUTE                       LZH  200
C    THE EIGENVECTORS OF THE ORIGINAL SYSTEM                              LZH  210
      COMPLEX Y, W, Z, A(NA,N), B(NB,N), X(NX,N)                          LZH  220
      REAL C, D                                                           LZH  230
      LOGICAL WANTX                                                       LZH  240
      NM1 = N - 1                                                         LZH  250
C REDUCE B TO TRIANGULAR FORM USING ELEMENTARY                            LZH  260
C TRANSFORMATIONS                                                         LZH  270
      DO 80 I=1,NM1                                                       LZH  280
        D = 0.00                                                          LZH  290
        IP1 = I + 1                                                       LZH  300
        DO 10 K=IP1,N                                                     LZH  310
          Y = B(K,I)                                                      LZH  320
          C = ABS(REAL(Y)) + ABS(AIMAG(Y))                                LZH  330
          IF (C.LE.D) GO TO 10                                            LZH  340
          D = C                                                           LZH  350
          II = K                                                          LZH  360
   10   CONTINUE                                                         LZH  370
        IF (D.EQ.0.0) GO TO 80                                            LZH  380
        Y = B(I,I)                                                        LZH  390
        IF (D.LE.ABS(REAL(Y))+ABS(AIMAG(Y))) GO TO 40                     LZH  400
```

```
C MUST INTERCHANGE                                                      LZH  410
      DO 20 J=1,N                                                       LZH  420
        Y = A(I,J)                                                      LZH  430
        A(I,J) = A(II,J)                                                LZH  440
        A(II,J) = Y                                                     LZH  450
   20   CONTINUE                                                        LZH  460
      DO 30 J=I,N                                                       LZH  470
        Y = B(I,J)                                                      LZH  480
        B(I,J) = B(II,J)                                                LZH  490
        B(II,J) = Y                                                     LZH  500
   30   CONTINUE                                                        LZH  510
   40   DO 70 J=IP1,N                                                   LZH  520
        Y = B(J,I)/B(I,I)                                               LZH  530
        IF (REAL(Y).EQ.0.0 .AND. AIMAG(Y).EQ.0.0) GO TO 70              LZH  540
        DO 50 K=1,N                                                     LZH  550
          A(J,K) = A(J,K) - Y*A(I,K)                                    LZH  560
   50   CONTINUE                                                        LZH  570
        DO 60 K=IP1,N                                                   LZH  580
          B(J,K) = B(J,K) - Y*B(I,K)                                    LZH  590
   60   CONTINUE                                                        LZH  600
   70   CONTINUE                                                        LZH  610
      B(IP1,I) = (0.0,0.0)                                              LZH  620
   80 CONTINUE                                                          LZH  630
C INITIALIZE X                                                          LZH  640
      IF (.NOT.WANTX) GO TO 110                                         LZH  650
      DO 100 I=1,N                                                      LZH  660
        DO 90 J=1,N                                                     LZH  670
          X(I,J) = (0.0,0.0)                                            LZH  680
   90   CONTINUE                                                        LZH  690
        X(I,I) = (1.0,0.00)                                             LZH  700
  100 CONTINUE                                                          LZH  710
C REDUCE A TO UPPER HESSENBERG FORM                                     LZH  720
  110 NM2 = N - 2                                                       LZH  730
      IF (NM2.LT.1) GO TO 270                                           LZH  740
      DO 260 J=1,NM2                                                    LZH  750
        JM2 = NM1 - J                                                   LZH  760
        JP1 = J + 1                                                     LZH  770
        DO 250 II=1,JM2                                                 LZH  780
          I = N + 1 - II                                                LZH  790
          IM1 = I - 1                                                   LZH  800
          IMJ = I - J                                                   LZH  810
          W = A(I,J)                                                    LZH  820
          Z = A(IM1,J)                                                  LZH  830
          IF (ABS(REAL(W))+ABS(AIMAG(W)).LE.ABS(REAL(Z))                LZH  840
     *      +ABS(AIMAG(Z))) GO TO 140                                   LZH  850
C MUST INTERCHANGE ROWS                                                 LZH  860
          DO 120 K=J,N                                                  LZH  870
            Y = A(I,K)                                                  LZH  880
            A(I,K) = A(IM1,K)                                           LZH  890
            A(IM1,K) = Y                                                LZH  900
  120     CONTINUE                                                      LZH  910
          DO 130 K=IM1,N                                                LZH  920
            Y = B(I,K)                                                  LZH  930
            B(I,K) = B(IM1,K)                                           LZH  940
            B(IM1,K) = Y                                                LZH  950
  130     CONTINUE                                                      LZH  960
  140     Z = A(I,J)                                                    LZH  970
          IF (REAL(Z).EQ.0.0 .AND. AIMAG(Z).EQ.0.0) GO TO 170           LZH  980
          Y = Z/A(IM1,J)                                                LZH  990
          DO 150 K=JP1,N                                                LZH 1000
            A(I,K) = A(I,K) - Y*A(IM1,K)                                LZH 1010
  150     CONTINUE                                                      LZH 1020
          DO 160 K=IM1,N                                                LZH 1030
            B(I,K) = B(I,K) - Y*B(IM1,K)                                LZH 1040
  160     CONTINUE                                                      LZH 1050
C TRANSFORMATION FROM THE RIGHT                                         LZH 1060
  170     W = B(I,IM1)                                                  LZH 1070
          Z = B(I,I)                                                    LZH 1080
          IF (ABS(REAL(W))+ABS(AIMAG(W)).LE.ABS(REAL(Z))                LZH 1090
     *      +ABS(AIMAG(Z))) GO TO 210                                   LZH 1100
C MUST INTERCHANGE COLUMNS                                              LZH 1110
          DO 180 K=1,I                                                  LZH 1120
            Y = B(K,I)                                                  LZH 1130
            B(K,I) = B(K,IM1)                                           LZH 1140
            B(K,IM1) = Y                                                LZH 1150
```

```
  180     CONTINUE                                                    LZH 1160
          DO 190 K=1,N                                                LZH 1170
            Y = A(K,I)                                                LZH 1180
            A(K,I) = A(K,IM1)                                         LZH 1190
            A(K,IM1) = Y                                              LZH 1200
  190     CONTINUE                                                    LZH 1210
          IF (.NOT.WANTX) GO TO 210                                   LZH 1220
          DO 200 K=IMJ,N                                              LZH 1230
            Y = X(K,I)                                                LZH 1240
            X(K,I) = X(K,IM1)                                         LZH 1250
            X(K,IM1) = Y                                              LZH 1260
  200     CONTINUE                                                    LZH 1270
  210     Z = B(I,IM1)                                                LZH 1280
          IF (REAL(Z).EQ.0.0 .AND. AIMAG(Z).EQ.0.0) GO TO 250         LZH 1290
          Y = Z/B(I,I)                                                LZH 1300
          DO 220 k=1.IM1                                              LZH 1310
            B(K,IM1) = B(K,IM1) - Y*B(K,I)                            LZH 1320
  220     CONTINUE                                                    LZH 1330
          B(I,IM1) = (0.0,0.0)                                        LZH 1340
          DO 230 K=1,N                                                LZH 1350
            A(K,IM1) = A(K,IM1) - Y*A(K,I)                            LZH 1360
  230     CONTINUE                                                    LZH 1370
          IF (.NOT.WANTX) GO TO 250                                   LZH 1380
          DO 240 K=IMJ,N                                              LZH 1390
            X(K,IM1) = X(K,IM1) - Y*X(K,I)                            LZH 1400
  240     CONTINUE                                                    LZH 1410
  250   CONTINUE                                                      LZH 1420
        A(JP1+1,J) = (0.0,0.0)                                        LZH 1430
  260 CONTINUE                                                        LZH 1440
  270 RETURN                                                          LZH 1450
      END                                                             LZH 1460
```

```
      SUBROUTINE LZIT(N, A, NA, B, NB, X, NX, WANTX, ITER, EIGA,      LZI   10
     * EIGB)                                                          LZI   20
C THIS SUBROUTINE SOLVES THE GENERALIZED EIGENVALUE PROBLEM           LZI   30
C               A X  = LAMBDA B X                                     LZI   40
C WHERE A IS A COMPLEX UPPER HESSENBERG MATRIX OF                     LZI   50
C ORDER N AND B IS A COMPLEX UPPER TRIANGULAR MATRIX OF ORDER N        LZI   60
C INPUT PARAMETERS                                                    LZI   70
C N       ORDER OF A AND B                                            LZI   80
C A       AN N X N UPPER HESSENBERG COMPLEX MATRIX                    LZI   90
C NA      THE ROW DIMENSION OF THE A MATRIX                           LZI  100
C B       AN N X N UPPER TRIANGULAR COMPLEX MATRIX                    LZI  110
C NB      THE ROW DIMENSION OF THE B MATRIX                           LZI  120
C X       CONTAINS TRANSFORMATIONS TO OBTAIN EIGENVECTORS OF          LZI  130
C         ORIGINAL SYSTEM. IF EIGENVECTORS ARE REQUESTED AND QZHES    LZI  140
C         IS NOT CALLED, X SHOULD BE SET TO THE IDENTITY MATRIX       LZI  150
C NX      THE ROW DIMENSION OF THE X MATRIX                           LZI  160
C WANTX   LOGICAL VARIABLE WHICH SHOULD BE SET TO .TRUE.              LZI  170
C         IF EIGENVECTORS ARE WANTED. OTHERWISE IT                    LZI  180
C         SHOULD BE SET TO .FALSE.                                    LZI  190
C OUTPUT PARAMETERS                                                   LZI  200
C X       THE ITH COLUMN CONTAINS THE ITH EIGENVECTOR                 LZI  210
C         IF EIGENVECTORS ARE REQUESTED                               LZI  220
C ITER    AN INTEGER ARRAY OF LENGTH N WHOSE ITH ENTRY                LZI  230
C         CONTAINS THE NUMBER OF ITERATIONS NEEDED TO FIND            LZI  240
C         THE ITH EIGENVALUE. FOR ANY I IF ITER(I) =-1 THEN           LZI  250
C         AFTER 30 ITERATIONS THERE HAS NOT BEEN A SUFFICIENT         LZI  260
C         DECREASE IN THE LAST SUBDIAGONAL ELEMENT OF A               LZI  270
C         TO CONTINUE ITERATING.                                      LZI  280
C EIGA    A COMPLEX ARRAY OF LENGTH N CONTAINING THE DIAGONAL OF A    LZI  290
C EIGB    A COMPLEX ARRAY OF LENGTH N CONTAINING THE DIAGONAL OF B    LZI  300
C THE ITH EIGENVALUE CAN BE FOUND BY DIVIDING EIGA(I) BY              LZI  310
C EIGB(I). WATCH OUT FOR EIGB(I) BEING ZERO                           LZI  320
      COMPLEX A(NA,N), B(NB,N), EIGA(N), EIGB(N)                      LZI  330
      COMPLEX S, W, Y, Z, CSQRT                                       LZI  340
      COMPLEX X(NX,N)                                                 LZI  350
      INTEGER ITER(N)                                                 LZI  360
      COMPLEX ANNM1, ALFM, BETM, D, SL, DEN, NUM, ANM1M1              LZI  370
      REAL EPSA, EPSB, SS, R, ANORM, BNORM, ANI, BNI, C               LZI  380
      REAL D0, D1, D2, E0, E1                                         LZI  390
      LOGICAL WAN1X                                                   LZI  400
      NN = N                                                          LZI  410
```

```
C COMPUTE THE MACHINE PRECISION TIMES THE NORM OF A AND B                LZI  420
      ANORM = 0.                                                         LZI  430
      BNORM = 0.                                                         LZI  440
      DO 30 I=1,N                                                        LZI  450
        ANI = 0.                                                         LZI  460
        IF (I.EQ.1) GO TO 10                                             LZI  470
        Y = A(I,I-1)                                                     LZI  480
        ANI = ANI + ABS(REAL(Y)) + ABS(AIMAG(Y))                         LZI  490
   10   BNI = 0.                                                         LZI  500
        DO 20 J=I,N                                                      LZI  510
          ANI = ANI + ABS(REAL(A(I,J))) + ABS(AIMAG(A(I,J)))             LZI  520
          BNI = BNI + ABS(REAL(B(I,J))) + ABS(AIMAG(B(I,J)))             LZI  530
   20   CONTINUE                                                         LZI  540
        IF (ANI.GT.ANORM) ANORM = ANI                                    LZI  550
        IF (BNI.GT.BNORM) BNORM = BNI                                    LZI  560
   30 CONTINUE                                                           LZI  570
      IF (ANORM.EQ.0.) ANORM = 1.0                                       LZI  580
      IF (BNORM.EQ.0.) BNORM = 1.0                                       LZI  590
      EPSB = BNORM                                                       LZI  600
      EPSA = ANORM                                                       LZI  610
   40 EPSA = EPSA/2.0                                                    LZI  620
      EPSB = EPSB/2.0                                                    LZI  630
      C = ANORM + EPSA                                                   LZI  640
      IF (C.GT.ANORM) GO TO 40                                           LZI  650
      IF (N.LE.1) GO TO 320                                              LZI  660
   50 ITS = 0                                                            LZI  670
      NM1 = NN - 1                                                       LZI  680
C CHECK FOR NEGLIGIBLE SUBDIAGONAL ELEMENTS                              LZI  690
   60 D2 = ABS(REAL(A(NN,NN))) + ABS(AIMAG(A(NN,NN)))                    LZI  700
      DO 70 LB=2,NN                                                      LZI  710
        L = NN + 2 - LB                                                  LZI  720
        SS = D2                                                          LZI  730
        Y = A(L-1,L-1)                                                   LZI  740
        D2 = ABS(REAL(Y)) + ABS(AIMAG(Y))                                LZI  750
        SS = SS + D2                                                     LZI  760
        Y = A(L,L-1)                                                     LZI  770
        R = SS + ABS(REAL(Y)) + ABS(AIMAG(Y))                            LZI  780
        IF (R.EQ.SS) GO TO 80                                            LZI  790
   70 CONTINUE                                                           LZI  800
      L = 1                                                              LZI  810
   80 IF (L.EQ.NN) GO TO 320                                             LZI  820
      IF (ITS.LT.30) GO TO 90                                            LZI  830
      ITER(NN) = -1                                                      LZI  840
      IF (ABS(REAL(A(NN,NM1)))+ABS(AIMAG(A(NN,NM1))).GT.0.8*             LZI  850
     * ABS(REAL(ANNM1))+ABS(AIMAG(ANNM1))) RETURN                        LZI  860
   90 IF (ITS.EQ.10 .OR. ITS.EQ.20) GO TO 110                            LZI  870
C COMPUTE SHIFT AS EIGENVALUE OF LOWER 2 BY 2                            LZI  880
      ANNM1 = A(NN,NM1)                                                  LZI  890
      ANM1M1 = A(NM1,NM1)                                                LZI  900
      S = A(NN,NN)*B(NM1,NM1) - ANNM1*B(NM1,NN)                          LZI  910
      W = ANNM1*B(NN,NN)*(A(NM1,NN)*B(NM1,NM1)-B(NM1,NN)*ANM1M1)         LZI  920
      Y = (ANM1M1*B(NN,NN)-S)/2.                                         LZI  930
      Z = CSQRT(Y*Y+W)                                                   LZI  940
      IF (REAL(Z).EQ.0.0 .AND. AIMAG(Z).EQ.0.0) GO TO 100                LZI  950
      D0 = REAL(Y/Z)                                                     LZI  960
      IF (D0.LT.0.0) Z = -Z                                              LZI  970
  100 DEN = (Y+Z)*B(NM1,NM1)*B(NN,NN)                                    LZI  980
      IF (REAL(DEN).EQ.0.0 .AND. AIMAG(DEN).EQ.0.0) DEN =                LZI  990
     * CMPLX(EPSA,0.0)                                                   LZI 1000
      NUM = (Y+Z)*S - W                                                  LZI 1010
      GO TO 120                                                          LZI 1020
C AD-HOC SHIFT                                                           LZI 1030
  110 Y = A(NM1,NN-2)                                                    LZI 1040
      NUM = CMPLX(ABS(REAL(ANNM1))+ABS(AIMAG(ANNM1)),ABS(REAL(Y)))       LZI 1050
     * +ABS(AIMAG(Y)))                                                  LZI 1060
      DEN = (1.0,0.0)                                                    LZI 1070
C CHECK FOR 2 CONSECUTIVE SMALL SUBDIAGONAL ELEMENTS                     LZI 1080
  120 IF (NN.EQ.L+1) GO TO 140                                           LZI 1090
      D2 = ABS(REAL(A(NM1,NM1))) + ABS(AIMAG(A(NM1,NM1)))                LZI 1100
      E1 = ABS(REAL(ANNM1)) + ABS(AIMAG(ANNM1))                          LZI 1110
      D1 = ABS(REAL(A(NN,NN))) + ABS(AIMAG(A(NN,NN)))                    LZI 1120
      NL = NN - (L+1)                                                    LZI 1130
      DO 130 MB=1,NL                                                     LZI 1140
        M = NN - MB                                                      LZI 1150
        E0 = E1                                                          LZI 1160
```

```
      Y = A(M,M-1)                                                    LZI 1170
      E1 = ABS(REAL(Y)) + ABS(AIMAG(Y))                               LZI 1180
      D0 = D1                                                         LZI 1190
      D1 = D2                                                         LZI 1200
      Y = A(M-1,M-1)                                                  LZI 1210
      D2 = ABS(REAL(Y)) + ABS(AIMAG(Y))                               LZI 1220
      Y = A(M,M)*DEN - B(M,M)*NUM                                     LZI 1230
      D0 = (D0+D1+D2)*(ABS(REAL(Y))+ABS(AIMAG(Y)))                    LZI 1240
      E0 = E0*E1*(ABS(REAL(DEN))+ABS(AIMAG(DEN))) + D0                LZI 1250
      IF (E0.EQ.D0) GO TO 150                                         LZI 1260
  130 CONTINUE                                                        LZI 1270
  140 M = L                                                           LZI 1280
  150 CONTINUE                                                        LZI 1290
      ITS = ITS + 1                                                   LZI 1300
      W = A(M,M)*DEN - B(M,M)*NUM                                     LZI 1310
      Z = A(M+1,M)*DEN                                                LZI 1320
      D1 = ABS(REAL(Z)) + ABS(AIMAG(Z))                               LZI 1330
      D2 = ABS(REAL(W)) + ABS(AIMAG(W))                               LZI 1340
C FIND L AND M AND SET A=LAM AND B=LBM                                LZI 1350
      NP1 = N + 1                                                     LZI 1360
      LOR1 = L                                                        LZI 1370
      NNORN = NN                                                      LZI 1380
      IF (.NOT.WANTX) GO TO 160                                       LZI 1390
      LOR1 = 1                                                        LZI 1400
      NNORN = N                                                       LZI 1410
  160 DO 310 I=M,NM1                                                  LZI 1420
        J = I + 1                                                     LZI 1430
C FIND ROW TRANSFORMATIONS TO RESTORE A TO                            LZI 1440
C UPPER HESSENBERG FORM. APPLY TRANSFORMATIONS                        LZI 1450
C TO A AND B                                                          LZI 1460
        IF (I.EQ.M) GO TO 170                                         LZI 1470
        W = A(I,I-1)                                                  LZI 1480
        Z = A(J,I-1)                                                  LZI 1490
        D1 = ABS(REAL(Z)) + ABS(AIMAG(Z))                             LZI 1500
        D2 = ABS(REAL(W)) + ABS(AIMAG(W))                             LZI 1510
        IF (D1.EQ.0.0) GO TO 60                                       LZI 1520
  170   IF (D2.GT.D1) GO TO 190                                       LZI 1530
C MUST INTERCHANGE ROWS                                               LZI 1540
        DO 180 K=I,NNORN                                              LZI 1550
          Y = A(I,K)                                                  LZI 1560
          A(I,K) = A(J,K)                                             LZI 1570
          A(J,K) = Y                                                  LZI 1580
          Y = B(I,K)                                                  LZI 1590
          B(I,K) = B(J,K)                                             LZI 1600
          B(J,K) = Y                                                  LZI 1610
  180   CONTINUE                                                      LZI 1620
        IF (I.GT.M) A(I,I-1) = A(J,I-1)                               LZI 1630
        IF (D2.EQ.0.0) GO TO 220                                      LZI 1640
C THE SCALING OF W AND Z IS DESIGNED TO AVOID A DIVISION BY ZERO      LZI 1650
C WHEN THE DENOMINATOR IS SMALL                                       LZI 1660
        Y = CMPLX(REAL(W)/D1,AIMAG(W)/D1)/CMPLX(REAL(Z)/D1,AIMAG(Z)/  LZI 1670
     *   D1)                                                          LZI 1680
        GO TO 200                                                     LZI 1690
  190   Y = CMPLX(REAL(Z)/D2,AIMAG(Z)/D2)/CMPLX(REAL(W)/D2,AIMAG(W)/  LZI 1700
     *   D2)                                                          LZI 1710
  200   DO 210 K=I,NNORN                                              LZI 1720
          A(J,K) = A(J,K) - Y*A(I,K)                                  LZI 1730
          B(J,K) = B(J,K) - Y*B(I,K)                                  LZI 1740
  210   CONTINUE                                                      LZI 1750
        IF (I.GT.M) A(J,I-1) = (0.0,0.0)                              LZI 1760
C PERFORM TRANSFORMATIONS FROM RIGHT TO RESTORE B TO                  LZI 1770
C   TRIANGLULAR FORM                                                  LZI 1780
C APPLY TRANSFORMATIONS TO A                                          LZI 1790
  220   Z = B(J,I)                                                    LZI 1800
        W = B(J,J)                                                    LZI 1810
        D2 = ABS(REAL(W)) + ABS(AIMAG(W))                             LZI 1820
        D1 = ABS(REAL(Z)) + ABS(AIMAG(Z))                             LZI 1830
        IF (D1.EQ.0.0) GO TO 60                                       LZI 1840
        IF (D2.GT.D1) GO TO 270                                       LZI 1850
C MUST INTERCHANGE COLUMNS                                            LZI 1860
        DO 230 K=LOR1,J                                               LZI 1870
          Y = A(K,J)                                                  LZI 1880
          A(K,J) = A(K,I)                                             LZI 1890
          A(K,I) = Y                                                  LZI 1900
          Y = B(K,J)                                                  LZI 1910
          B(K,J) = B(K,I)                                             LZI 1920
          B(K,I) = Y                                                  LZI 1930
```

```
  230   CONTINUE                                                        LZI 1940
        IF (I.EQ.NM1) GO TO 240                                         LZI 1950
        Y = A(J+1,J)                                                    LZI 1960
        A(J+1,J) = A(J+1,I)                                             LZI 1970
        A(J+1,I) = Y                                                    LZI 1980
  240   IF (.NOT.WANTX) GO TO 260                                       LZI 1990
        DO 250 K=1,N                                                    LZI 2000
          Y = X(K,J)                                                    LZI 2010
          X(K,J) = X(K,I)                                               LZI 2020
          X(K,I) = Y                                                    LZI 2030
  250   CONTINUE                                                        LZI 2040
  260   IF (D2.EQ.0.0) GO TO 310                                        LZI 2050
        Z = CMPLX(REAL(W)/D1,AIMAG(W)/D1)/CMPLX(REAL(Z)/D1,AIMAG(Z)/    LZI 2060
     *   D1)                                                            LZI 2070
        GO TO 280                                                       LZI 2080
  270   Z = CMPLX(REAL(Z)/D2,AIMAG(Z)/D2)/CMPLX(REAL(W)/D2,AIMAG(W)/    LZI 2090
     *   D2)                                                            LZI 2100
  280   DO 290 K=LOR1,J                                                 LZI 2110
          A(K,I) = A(K,I) - Z*A(K,J)                                    LZI 2120
          B(K,I) = B(K,I) - Z*B(K,J)                                    LZI 2130
  290   CONTINUE                                                        LZI 2140
        B(J,I) = (0.0,0.0)                                              LZI 2150
        IF (I.LT.NM1) A(I+2,I) = A(I+2,I) - Z*A(I+2,J)                  LZI 2160
        IF (.NOT.WANTX) GO TO 310                                       LZI 2170
        DO 300 K=1,N                                                    LZI 2180
          X(K,I) = X(K,I) - Z*X(K,J)                                    LZI 2190
  300   CONTINUE                                                        LZI 2200
  310 CONTINUE                                                          LZI 2210
      GO TO 60                                                          LZI 2220
  320 CONTINUE                                                          LZI 2230
      EIGA(NN) = A(NN,NN)                                               LZI 2240
      EIGB(NN) = B(NN,NN)                                               LZI 2250
      IF (NN.EQ.1) GO TO 330                                            LZI 2260
      ITER(NN) = ITS                                                    LZI 2270
      NN = NM1                                                          LZI 2280
      IF (NN.GT.1) GO TO 50                                             LZI 2290
      ITER(1) = 0                                                       LZI 2300
      GO TO 320                                                         LZI 2310
C FIND EIGENVECTORS USING B FOR INTERMEDIATE STORAGE                    LZI 2320
  330 IF (.NOT.WANTX) RETURN                                            LZI 2330
      M = N                                                             LZI 2340
  340 CONTINUE                                                          LZI 2350
      ALFM = A(M,M)                                                     LZI 2360
      BETM = B(M,M)                                                     LZI 2370
      B(M,M) = (1.0,0.0)                                                LZI 2380
      L = M - 1                                                         LZI 2390
      IF (L.EQ.0) GO TO 370                                             LZI 2400
  350 CONTINUE                                                          LZI 2410
      L1 = L + 1                                                        LZI 2420
      SL = (0.0,0.0)                                                    LZI 2430
      DO 360 J=L1,M                                                     LZI 2440
        SL = SL + (BETM*A(L,J)-ALFM*B(L,J))*B(J,M)                      LZI 2450
  360 CONTINUE                                                          LZI 2460
      Y = BETM*A(L,L) - ALFM*B(L,L)                                     LZI 2470
      IF (REAL(Y).EQ.0.0 .AND. AIMAG(Y).EQ.0.0) Y =                     LZI 2480
     * CMPLX((EPSA+EPSB)/2.0,0.0)                                      LZI 2490
      B(L,M) = -SL/Y                                                    LZI 2500
      L = L - 1                                                         LZI 2510
  370 IF (L.GT.0) GO TO 350                                             LZI 2520
      M = M - 1                                                         LZI 2530
      IF (M.GT.0) GO TO 340                                             LZI 2540
C TRANSFORM TO ORIGINAL COORDINATE SYSTEM                               LZI 2550
      M = N                                                             LZI 2560
  380 CONTINUE                                                          LZI 2570
      DO 400 I=1,N                                                      LZI 2580
        S = (0.0,0.0)                                                   LZI 2590
        DO 390 J=1,M                                                    LZI 2600
          S = S + X(I,J)*B(J,M)                                         LZI 2610
  390   CONTINUE                                                        LZI 2620
        X(I,M) = S                                                      LZI 2630
  400 CONTINUE                                                          LZI 2640
      M = M - 1                                                         LZI 2650
      IF (M.GT.0) GO TO 380                                             LZI 2660
C NORMALIZE SO THAT LARGEST COMPONENT = 1.                              LZI 2670
      M = N                                                             LZI 2680
```

```
410 CONTINUE                                                          LZI 2690
    SS = 0.                                                           LZI 2700
    DO 420 I=1,N                                                      LZI 2710
      R = ABS(REAL(X(I,M))) + ABS(AIMAG(X(I,M)))                      LZI 2720
      IF (R.LT.SS) GO TO 420                                          LZI 2730
      SS = R                                                          LZI 2740
      D = X(I,M)                                                      LZI 2750
420 CONTINUE                                                          LZI 2760
    IF (SS.EQ.0.0) GO TO 440                                          LZI 2770
    DO 430 I=1,N                                                      LZI 2780
      X(I,M) = X(I,M)/D                                               LZI 2790
430 CONTINUE                                                          LZI 2800
440 M = M - 1                                                         LZI 2810
    IF (M.GT.0) GO TO 410                                             LZI 2820
    RETURN                                                            LZI 2830
    END                                                               LZI 2840
```