Chapter 3

Hermitian Matrices

3.1 Introduction

The FORTRAN codes in this chapter address the question of computing distinct eigenvalues and corresponding eigenvectors of Hermitian matrices, using a single-vector Lanczos procedure. For a given Hermitian matrix A, these codes compute real scalars λ and corresponding complex vectors $x \neq 0$ such that

$$Ax = \lambda x. \tag{3.1.1}$$

Definition 2 *A complex nxn matrix A, $A \equiv (a_{ij}), 1 \leq i,j \leq n$, is a Hermitian matrix if and only if for every i and j, $a_{ij} = \overline{a_{ji}}$, where the overbar denotes the complex conjugate of the complex-valued entry a_{ij}.*

It is straight-forward to demonstrate from Definition 2 that for any Hermitian matrix $A = B + Ci$, where B and C are real matrices and $i = \sqrt{-1}$, that B must be a real symmetric matrix and C must be a skew symmetric matrix. That is, $B^T = B$ and $C^T = -C$. Futhermore, it is not difficult to see that Hermitian matrices must have real diagonal entries and real eigenvalues. However, the eigenvectors are complex-valued. Any Hermitian matrix can be transformed into a real symmetric tridiagonal matrix for the purposes of computing the eigenvalues of the Hermitian matrix, Stewart [24]. In fact, the Lanczos recursion which we use in the codes in this chapter transforms the given Hermitian matrix A into a family of real symmetric tridiagonal matrices rather than into a family of Hermitian tridiagonal matrices.

Hermitian matrices possess the 'same' properties as real symmetric matrices do, except that these properties are defined with respect to the complex or Hermitian norm, rather than with respect to the Euclidean norm, see Stewart [24]. The Hermitian norm of a given complex-valued vector $x \equiv (x(i)), 1 \leq i \leq n$, is defined as $\|x\|_C^2 \equiv \sum_{i=1}^{n} \overline{x(i)}x(i)$. Three properties which we use are:

1. Hermitian matrices have complete eigensystems. That is, the dimension of the eigenspace corresponding to any given eigenvalue of a Hermitian matrix is the same as the multiplicity of that eigenvalue as a root of the characteristic polynomial of that matrix.

2. For any two distinct eigenvalues λ, μ and corresponding eigenvectors x, y, $x^H y = 0$, where the superscript H denotes the complex conjugate transpose of the vector x. The complex conjugate transpose of a column vector x is the row vector whose i^{th} component is $\overline{x(i)}$. There is a complete set of eigenvectors $X_n \equiv (x_1, \ldots, x_n)$ such that X is a unitary matrix.

3. Small Hermitian perturbations in a Hermitian matrix cause only small perturbations in the eigenvalues.

The single-vector Lanczos codes in this chapter can be used to compute either a very few or very many of the distinct eigenvalues of the given Hermitian matrix. The documentation for these codes is contained in Chapter 2, Section 2.2. As in the real symmetric case, the A-multiplicity of a given computed 'good' Lanczos eigenvalue can be obtained only with additional computation, and the modifications required to do this additional computation are not included in these versions of the codes. This implementation uses a Hermitian analog of the basic Lanczos recursion contained in Eqns (1.2.1) and (1.2.2) to generate a family of real symmetric tridiagonal matrices whose sizes are specified by the user. There is no reorthogonalization of the Lanczos vectors at any stage in any of the computations.

The Hermitian version of the Lanczos recursion which we use is given below. For $i = 1, 2, ..., m$ and a randomly-generated complex starting vector v_1 with $\|v_1\|_C = 1$, generate Lanczos vectors v_i using the following recursion.

$$\beta_{i+1} v_{i+1} = A v_i - \alpha_i v_i - \beta_i v_{i-1}, \tag{3.1.2}$$

where

$$\alpha_i \equiv v_i^H A v_i, \quad \beta_{i+1} = \|A v_i - \alpha_i v_i - \beta_i v_{i-1}\|_C \tag{3.1.3}$$

We see from Eqns(3.1.3) that the Hermitian inner product is used. This is the 'natural' inner product for Hermitian matrices. Gram-Schmidt orthogonalization is used, unlike the real symmetric case where a modified Gram-Schmidt orthogonalization was used. This change in the local orthogonalization procedure increases the storage requirements for the implementation of the Lanczos recursion by one additional complex vector of length equal to the order of the original A-matrix. Modified Gram-Schmidt orthogonalization cannot be used in the Hermitian case because corrections to the α_i defined by this modification are complex-valued not real, and it would not be legitimate to accept the real portions of these corrections and simply ignore the complex portions.

It is easy to demonstrate that as we stated earlier, each Lanczos matrix (T-matrix) generated by this Hermitian recursion is a real symmetric tridiagonal matrix. In particular, we see from the formulas in Eqn(3.1.3) that the diagonal entries of each of these matrices are Rayleigh quotients of the given Hermitian matrix A, and therefore must all be real-valued. Furthermore by construction, the nonzero off-diagonal entries β_{i+1} are all real-valued. This use of real-valued β_i requires some justification. This justification is given in Section 4.9 of Chapter 4 of Volume 1 of this book.

HLEVAL, the main program for the Hermitian eigenvalue computations, calls the subroutine BISEC to compute eigenvalues of the specified tridiagonal Lanczos matrices on the user-specified intervals. BISEC simultaneously computes these T-eigenvalues with their T-multiplicities and sorts the computed T-eigenvalues into two classes, the 'good' T-eigenvalues and the 'spurious' T-eigenvalues. The 'good' T-eigenvalues are accepted as approximations to eigenvalues of the user-specified matrix A. The accuracy of these 'good' T-eigenvalues as eigenvalues of A is then estimated using error estimates computed by subroutine INVERR. Error estimates are computed only for isolated 'good' T-eigenvalues. All other 'good' T-eigenvalues are assumed to have converged. Convergence is then checked. If convergence has not yet occurred and a larger T-matrix has been specified by the user, the program will continue on to the larger T-matrix, repeating the above procedure on this larger matrix.

Once the eigenvalues have been computed accurately enough, the user can select a subset of the 'converged' eigenvalues for which eigenvectors are to be computed. The main program HLEVEC, for computing eigenvectors of Hermitian matrices, is then used to compute these desired eigenvectors.

The computations in the Lanczos recursion are a mixture of double precision real arithmetic and of double precision complex arithmetic. Once the Lanczos matrices have been computed, the remaining computations are all done in double precision real arithmetic, using the same subroutines that are used in the real symmetric case. In addition to the programs and subroutines provided here, the user must supply a subroutine USPEC which defines and initializes the user-specified matrix A and a subroutine CMATV which computes matrix-vector multiplies Ax for any given vector x. These subroutines must be constructed in such a way as to take advantage of the sparsity (and/or structure) of the user-supplied A-matrix and such that these computations are done accurately.

3.2 HLEVAL: Main Program, Eigenvalue Computations

```
C-----HLEVAL  (EIGENVALUES OF HERMITIAN MATRICES)----------------------HHL00010
C  Authors:  Jane Cullum and Ralph A. Willoughby (deceased)            HHL00020
C            Los Alamos National Laboratory                            HHL00030
C            Los Alamos, New Mexico 87544                              HHL00040
C            cullumj@lanl.gov                                          HHL00045
c                                                                      HHL00050
c  These codes are copyrighted by the authors.  These codes            HHL00060
c  and modifications of them or portions of them are NOT to be         HHL00070
c  incorporated into any commercial codes without legal agreements     HHL00080
c  with the authors.  If these codes or portions of them               HHL00090
c  are used in other scientific or engineering research works          HHL00100
c  the names of the authors of these codes and appropriate             HHL00110
c  references to their written work are to be incorporated in the      HHL00120
c  derivative works.                                                   HHL00130
c                                                                      HHL00140
c  This header is not to be removed from these codes.                  HHL00150
C                                                                      HHL00155
C       REFERENCE: Cullum and Willoughby, Chapters 1,2,3,4             HHL00160
C       Lanczos Algorithms for Large Symmetric Eigenvalue ComputationsHHL00165
C       VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in    HHL00166
C       Applied Mathematics, 2002. SIAM Publications,                  HHL00167
C       Philadelphia, PA. USA                                          HHL00168
C                                                                      HHL00169
C     CONTAINS MAIN PROGRAM FOR COMPUTING DISTINCT EIGENVALUES OF      HHL00170
C     A HERMITIAN MATRIX USING LANCZOS TRIDIAGONALIZATION WITHOUT      HHL00180
C     REORTHOGONALIZATION                                              HHL00190
C                                                                      HHL00200
C     PORTABILITY:                                                     HHL00210
C     THIS PROGRAM IS NOT PORTABLE DUE TO THE USE OF COMPLEX*16        HHL00220
C     VARIABLES.  MOREOVER, THE PFORT VERIFIER IDENTIFIED THE          HHL00230
C     FOLLOWING ADDITIONAL NONPORTABLE CONSTRUCTIONS:                  HHL00240
C                                                                      HHL00250
C     1.  DATA/MACHEP/ STATEMENT                                       HHL00260
C     2.  ALL READ(5,*) STATEMENTS (FREE FORMAT)                       HHL00270
C     3.  FORMAT(20A4) USED WITH EXPLANATORY HEADER EXPLAN.            HHL00280
C     4.  HEXADECIMAL FORMAT (4Z20) USED IN ALPHA/BETA FILES 1 AND 2.  HHL00290
C                                                                      HHL00300
C----------------------------------------------------------------------HHL00310
C     SPECIFY DIMENSIONS OF ARRAYS NEEDED BY LANCZOS ROUTINES          HHL00320
C                                                                      HHL00330
C     USER SPECIFIES THE FOLLOWING:                                    HHL00340
C     OTHER ARRAY DIMENSIONS ARE COMPUTED IN PARAMETER STATEMENTS      HHL00350
C     N = DIMENSION OF THE MATRIX EIGENVALUE PROBLEM                   HHL00360
C     KMAX =  MAXIMUM SIZE OF LANCZOS MATRICES TO BE USED              HHL00370
C     NSINT >= NUMBER OF SUBINTERVALS SPECIFIED IN INPUT FILE 5        HHL00380
C     NTMATS >= NUMBER OF LANCZOS MATRICES SPECIFIED IN INPUT FILE 5   HHL00390
C     BELOW WE ASSUME THAT NO MORE THAN KMAX/2 EIGENVALUES             HHL00400
C     ARE COMPUTED IN ANY ONE OF THE SUBINTERVALS (LB(J),UB(J))        HHL00410
C     SUPPLIED BY THE USER.  V2 WILL BE USED FOR BOTH UPPER AND        HHL00420
C     LOWER BOUNDS ON THE EIGENVALUES AS THEY ARE COMPUTED SO          HHL00430
C     IF MORE THAN KMAX/2 EIGENVALUES ARE TO BE COMPUTED IN ANY        HHL00440
```

```
C     ONE SUBINTERVAL, THE DIMENSION OF V2 MUST BE ADJUSTED
C     ACCORDINGLY.  FOR EXAMPLE IF THE USER WANTS ALL THE EIGENVALUES
C     OF THE LANCZOS MATRIX THEN KV2 MUST BE > MAX(KMAX,N)
C     BECAUSE OF THE INTEGER ARITHMETIC IT IS NECESSARY TO ADD AN
C     EXTRA 1 TO THE EXPRESSIONS.
C
C     TO AVOID USING MAX(I,J) IN THE PARAMETER LISTING WE HAVE USED
C     THE FOLLOWING EQUIVALENT RELATIONSHIP
C
C     MAX(I,J) = ( 2*I/(I+J))*I  + (2*J/(I+J))*J
C
C     PARAMETER ( N = 81, KMAX = 100, NSINT = 20, NTMATS = 20)
      PARAMETER ( N = 625, KMAX = 1500, NSINT = 20, NTMATS = 20)
C
      PARAMETER ( KMAX1 = KMAX+1, KMAX2 = 2*KMAX, NKMAX = N+KMAX )
      PARAMETER ( KMAXP2 = KMAX + 2)
      PARAMETER ( N2 = 2*N, N2KMAX = N2+KMAX, NKMAX2=N+KMAX2)
      PARAMETER ( KMAXPO2 = KMAXP2/2, KMAX1O2 = KMAX1/2 )
      PARAMETER ( NKMAX12 = N+KMAX1O2, NKMAXPO = N+KMAXPO2)
      PARAMETER (KVS = ((2*N2)/N2KMAX)*N2 + ((2*KMAX)/N2KMAX)*KMAX )
      PARAMETER (KV1 =((2*N)/NKMAXPO)*N+((2*KMAXPO2)/NKMAXPO)*KMAXPO2)
C     PARAMETER (KV2 = ((2*N)/NKMAX12)*N +((2*KMAX1O2)/NKMAX12)*KMAX1O2)
C  BELOW GOES WITH COMPUTING ALL EIGENVALUES OF LANCZOS MATRIX
      PARAMETER (KV2 = ((2*N)/NKMAX)*N + ((2*KMAX)/NKMAX)*KMAX)
      PARAMETER (KG = ((2*KMAX2)/NKMAX2)*KMAX2 +((2*N)/NKMAX2)*N )
C
C-----------------------------------------------------------------------
C
      DOUBLE PRECISION  ALPHA(KMAX),BETA(KMAX1),VS(KVS)
      COMPLEX*16  V1(KV1),V2(KV2)
      DOUBLE PRECISION  GR(N),GC(N),LB(NSINT),UB(NSINT)
      DOUBLE PRECISION  BTOL,GAPTOL,TTOL,MACHEP,EPSM,RELTOL
      DOUBLE PRECISION  SCALE1,SCALE2,SCALE3,SCALE4,BISTOL,CONTOL,MULTOL
      DOUBLE PRECISION  ONE,ZERO,TEMP,TKMAX,BETAM,BKMIN,T0,T1
      REAL  G(KG),EXPLAN(20)
      INTEGER  MP(KMAX),NMEV(NTMATS)
      INTEGER  SVSEED,RHSEED,SVSOLD
      INTEGER IABS
      REAL  ABS
      DOUBLE PRECISION  DABS, DSQRT, DFLOAT
      EXTERNAL CMATV
C
C-----------------------------------------------------------------------
      DATA MACHEP/Z3410000000000000/
      EPSM = 2.0D0*MACHEP
C-----------------------------------------------------------------------
      WRITE(6,1) N,KMAX,NSINT,NTMATS
    1 FORMAT(' N,KMAX,NSINT,NTMATS ='/4I10)
      WRITE(6,2) KMAX1,KMAX2,N2,N2KMAX,NKMAX2
    2 FORMAT(' KMAX1,KMAX2,N2,N2KMAX,NKMAX2 ='/5I10)
      WRITE(6,3) KMAXPO2,KMAX1O2,NKMAXPO,NKMAX12
    3 FORMAT(' KMAXPO2,KMAX1O2,NKMAXPO,NKMAX12 ='/4I10)
      WRITE(6,4) KVS,KV1,KV2,KG
    4 FORMAT(' KVS,KV1,KV2,KG ='/4I10)
C
```

```
C     THE ARRAYS V1 AND V2 ARE DEFINED AS COMPLEX*16 IN THE MAIN PROGRAMHHL01000
C     AND IN THE SUBROUTINE LANCZS.  HOWEVER, IN THE OTHER SUBROUTINES  HHL01010
C     THEY ARE DECLARED AS DOUBLE PRECISION ARRAYS. NOTE THAT THE       HHL01020
C     DIMENSION OF V2 ASSUMES THAT NO MORE THAN KMAX/2 EIGENVALUES OF   HHL01030
C     THE T-MATRICES ARE BEING COMPUTED IN ANY ONE OF THE SUB-INTERVALS HHL01040
C     BEING CONSIDERED.  V2 MUST CONTAIN UPPER AND LOWER BOUNDS         HHL01050
C     ON EACH T-EIGENVALUE COMPUTED BY BISEC IN ANY ONE GIVEN INTERVAL. HHL01060
C                                                                       HHL01070
C     ARRAYS MUST BE DIMENSIONED AS FOLLOWS:                            HHL01080
C          1. ALPHA:   >= KMAX.  BETA:  >= (KMAX+1)                     HHL01090
C          2. V1:  >= MAX(N,(KMAX+1)/2).   V2:  >= MAX(N,KMAX/2)        HHL01100
C          3. VS:  >= MAX(2*N,KMAX).                                    HHL01110
C          4. GR,GC:  >=  N                                             HHL01120
C          5. G:  >= MAX(2*KMAX,N)                                      HHL01130
C          6. MP:   >= KMAX                                             HHL01140
C          7. LB,UB:  >= NUMBER OF SUB-INTERVALS SPECIFIED              HHL01150
C          8. NMEV:  >= NUMBER OF T-MATRICES SPECIFIED                  HHL01160
C          9. EXPLAN:  DIMENSION IS 20.                                 HHL01170
C                                                                       HHL01180
C                                                                       HHL01190
C     IMPORTANT TOLERANCES OR SCALES THAT ARE USED REPEATEDLY           HHL01200
C     THROUGHOUT THE PROGRAM ARE THE FOLLOWING:                         HHL01210
C     SCALED MACHINE EPSILON:  TTOL = TKMAX*EPSM WHERE                  HHL01220
C     EPSM = 2*MACHINE EPSILON AND                                      HHL01230
C     TKMAX = MAX(|ALPHA(J)|,BETA(J), J = 1,MEV)                        HHL01240
C     BISEC CONVERGENCE TOLERANCE:  BISTOL = DSQRT(1000+MEV)*TTOL       HHL01250
C     BISEC MULTIPLICITY TOLERANCE:  MULTOL = (1000+MEV)*TTOL           HHL01260
C     LANCZOS CONVERGENCE TOLERANCE:   CONTOL = BETA(MEV+1)*1.D-10      HHL01270
C-----------------------------------------------------------------------HHL01280
C     OUTPUT HEADER                                                     HHL01290
      WRITE(6,10)                                                       HHL01300
   10 FORMAT(/' LANCZOS PROCEDURE FOR HERMITIAN MATRICES'/)             HHL01310
C                                                                       HHL01320
C     SET PROGRAM PARAMETERS                                            HHL01330
C     SCALEK ARE USED IN TOLERANCES NEEDED IN SUBROUTINES LUMP,         HHL01340
C     ISOEV AND PRTEST.  USER MUST NOT MODIFY THESE SCALES.             HHL01350
      SCALE1 = 5.0D2                                                    HHL01360
      SCALE2 = 5.0D0                                                    HHL01370
      SCALE3 = 5.0D0                                                    HHL01380
      SCALE4 = 1.0D4                                                    HHL01390
      ONE  = 1.0D0                                                      HHL01400
      ZERO = 0.0D0                                                      HHL01410
      BTOL = EPSM                                                       HHL01420
C     BTOL = 1.0D-8                                                     HHL01430
      GAPTOL = 1.0D-8                                                   HHL01440
      ICONV = 0                                                         HHL01450
      MOLD = 0                                                          HHL01460
      MOLD1 = 1                                                         HHL01470
      ICT = 0                                                           HHL01480
      MMB = 0                                                           HHL01490
      IPROJ = 0                                                         HHL01500
C                                                                       HHL01510
C     READ USER-SPECIFIED PARAMETERS FROM INPUT FILE 5 (FREE FORMAT)    HHL01520
C                                                                       HHL01530
C     READ USER-PROVIDED HEADER FOR RUN                                 HHL01540
```

```
      READ(5,20) EXPLAN
      WRITE(6,20) EXPLAN
      READ(5,20) EXPLAN
      WRITE(6,20) EXPLAN
   20 FORMAT(20A4)
C
C  MODIFIED 4/16/93, N AND KMAX SET IN PARAMETER LIST.
C XXXXREAD ORDER OF MATRICES (N) , MAXIMUM ORDER OF T-MATRIX (KMAX),
C     NUMBER OF T-MATRICES ALLOWED (NMEVS), AND MATRIX IDENTIFICATION
C     NUMBERS (MATNO)
      READ(5,20) EXPLAN
      READ(5,*) NMEVS,MATNO
C     READ(5,*) N,KMAX,NMEVS,MATNO
C
C     READ SEEDS FOR LANCZS AND INVERR SUBROUTINES (SVSEED AND RHSEED)
C     READ MAXIMUM NUMBER OF ITERATIONS ALLOWED FOR EACH INVERSE
C     ITERATION (MXINIT) AND MAXIMUM NUMBER OF STURM SEQUENCES
C     ALLOWED (MXSTUR)
      READ(5,20) EXPLAN
      READ(5,*) SVSEED,RHSEED,MXINIT,MXSTUR
C
C     ISTART = (0,1):  ISTART = 0 MEANS ALPHA/BETA FILE IS NOT
C     AVAILABLE.  ISTART = 1 MEANS ALPHA/BETA FILE IS AVAILABLE ON
C     FILE 2.
C     ISTOP = (0,1):  ISTOP = 0 MEANS PROCEDURE GENERATES ALPHA/BETA
C     FILE AND THEN TERMINATES.  ISTOP = 1 MEANS PROCEDURE GENERATES
C     ALPHAS/BETAS IF NEEDED AND THEN COMPUTES EIGENVALUES AND ERROR
C     ESTIMATES AND THEN TERMINATES.
      READ(5,20) EXPLAN
      READ(5,*) ISTART,ISTOP
C
C     IHIS = (0,1):  IHIS = 0 MEANS ALPHA/BETA FILE IS NOT WRITTEN
C     TO FILE 1.  IHIS = 1 MEANS ALPHA/BETA FILE IS WRITTEN TO FILE 1.
C     IDIST = (0,1):  IDIST = 0 MEANS DISTINCT T-EIGENVALUES
C     ARE NOT WRITTEN TO FILE 11.  IDIST = 1 MEANS DISTINCT
C     T-EIGENVALUES ARE WRITTEN TO FILE 11.
C     IWRITE = (0,1):  IWRITE = 0 MEANS NO INTERMEDIATE OUTPUT
C     FROM THE COMPUTATIONS IS WRITTEN TO FILE 6.  IWRITE = 1 MEANS
C     T-EIGENVALUES AND ERROR ESTIMATES ARE WRITTEN TO FILE 6
C     AS THEY ARE COMPUTED.
      READ(5,20) EXPLAN
      READ(5,*) IHIS,IDIST,IWRITE
C
C     READ IN THE RELATIVE TOLERANCE (RELTOL) FOR USE IN THE
C     SPURIOUS, T-MULTIPLICITY, AND PRTESTS.
      READ(5,20) EXPLAN
      READ(5,*) RELTOL
C
C     READ IN THE SIZES OF THE T-MATRICES TO BE CONSIDERED.
      READ(5,20) EXPLAN
      READ(5,*) (NMEV(J), J=1,NMEVS)
C
C     READ IN THE NUMBER OF SUBINTERVALS TO BE CONSIDERED.
      READ(5,20) EXPLAN
      READ(5,*) NINT
```

```
C
C     READ IN THE LEFT-END POINTS OF THE SUBINTERVALS TO BE CONSIDERED.
C     THESE MUST BE IN ALGEBRAICALLY-INCREASING ORDER
      READ(5,20) EXPLAN
      READ(5,*) (LB(J), J=1,NINT)
C
C     READ IN THE RIGHT-END POINTS OF THE SUBINTERVALS TO BE CONSIDERED.
C     THESE MUST BE IN ALGEBRAICALLY-INCREASING ORDER
      READ(5,20) EXPLAN
      READ(5,*) (UB(J), J=1,NINT)
C
C-----------------------------------------------------------------------
C
C     INITIALIZE THE ARRAYS FOR THE USER-SPECIFIED MATRIX
C     AND PASS THE STORAGE LOCATIONS OF THESE ARRAYS TO THE
C     MATRIX-VECTOR MULTIPLY SUBROUTINE CMATV.
C
      CALL USPEC(N,MATNO)
C
C-----------------------------------------------------------------------
C
C     MASK UNDERFLOW AND OVERFLOW
C
      CALL MASK
C
C-----------------------------------------------------------------------
C
C     WRITE TO FILE 6, A SUMMARY OF THE PARAMETERS FOR THIS RUN
C
      WRITE(6,30) MATNO,N,KMAX
   30 FORMAT(/3X,'MATRIX ID',4X,'ORDER OF A',4X,'MAX ORDER OF T'/
     1 I12,I14,I18/)
C
      WRITE(6,40) ISTART,ISTOP
   40 FORMAT(/2X,'ISTART',3X,'ISTOP'/2I8/)
C
      WRITE(6,50) IHIS,IDIST,IWRITE
   50 FORMAT(/4X,'IHIS',3X,'IDIST',2X,'IWRITE'/3I8/)
C
      WRITE(6,60) SVSEED,RHSEED
   60 FORMAT(/' SEEDS FOR RANDOM NUMBER GENERATOR'//
     1 4X,'LANCZS SEED',4X,'INVERR SEED'/2I15/)
C
      WRITE(6,70) (NMEV(J), J=1,NMEVS)
   70 FORMAT(/' SIZES OF T-MATRICES TO BE CONSIDERED'/(6I12))
C
      WRITE(6,80) RELTOL,GAPTOL,BTOL
   80 FORMAT(/' RELATIVE TOLERANCE USED TO COMBINE COMPUTED T-EIGENVALUE
     1S'/E15.3/' RELATIVE GAP TOLERANCES USED IN INVERSE ITERATION'/
     1E15.3/' RELATIVE TOLERANCE FOR CHECK ON SIZE OF BETAS'/E15.3/)
C
      WRITE(6,90) (J,LB(J),UB(J), J=1,NINT)
   90 FORMAT(/' BISEC WILL BE USED ON THE FOLLOWING INTERVALS'/
     1 (I6,2E20.6))
C
```

```
      IF (ISTART.EQ.0) GO TO 140                                        HHL02650
C                                                                       HHL02660
C     READ IN ALPHA BETA HISTORY                                        HHL02670
C                                                                       HHL02680
      READ(2,100)MOLD,NOLD,SVSOLD,MATOLD                                HHL02690
  100 FORMAT(2I6,I12,I8)                                                HHL02700
C                                                                       HHL02710
C  CHANGED KMAX TO PARAMETER VARIABLE SO BELOW NO LONGER ALLOWED        HHL02720
C  SO DEFAULT TO TERMINATE IF HISTORY FILE IS NOT LONG ENOUGH           HHL02730
C     IF (KMAX.LT.MOLD) KMAX = MOLD                                     HHL02740
C     KMAX1 = KMAX + 1                                                  HHL02750
C                                                                       HHL02760
      IF (KMAX.LT.MOLD) WRITE(6,115) KMAX,MOLD                          HHL02770
      IF (KMAX.LT.MOLD) GO TO 640                                       HHL02780
  115 FORMAT(/' PROGRAM TERMINATES FOR USER TO RESET KMAX.  CURRENT VALUHHL02790
     1E',I6/' IS LARGER THAN THE SIZE',I6,' OF THE TRIDIAGONAL MATRIX ONHHL02800
     1FILE 2'/)                                                         HHL02810
C                                                                       HHL02820
C     CHECK THAT ORDER N, MATRIX ID MATNO, AND RANDOM SEED SVSEED       HHL02830
C     AGREE WITH THOSE IN THE HISTORY FILE.  IF NOT PROCEDURE STOPS.    HHL02840
C                                                                       HHL02850
      ITEMP = (NOLD-N)**2+(MATNO-MATOLD)**2+(SVSEED-SVSOLD)**2          HHL02860
C                                                                       HHL02870
      IF (ITEMP.EQ.0) GO TO 120                                         HHL02880
C                                                                       HHL02890
      WRITE(6,110)                                                      HHL02900
  110 FORMAT(' PROGRAM TERMINATES'/    ' READ FROM FILE 2 CORRESPONDS TOHHL02910
     1 DIFFERENT MATRIX THAN MATRIX SPECIFIED'/)                        HHL02920
      GO TO 640                                                         HHL02930
C                                                                       HHL02940
  120 CONTINUE                                                          HHL02950
      MOLD1 = MOLD+1                                                    HHL02960
C                                                                       HHL02970
      READ(2,130)(ALPHA(J), J=1,MOLD)                                   HHL02980
      READ(2,130)(BETA(J), J=1,MOLD1)                                   HHL02990
  130 FORMAT(4Z20)                                                      HHL03000
C                                                                       HHL03010
      IF (KMAX.EQ.MOLD) GO TO 160                                       HHL03020
C                                                                       HHL03030
      READ(2,130)(V1(J), J=1,N)                                         HHL03040
      READ(2,130)(V2(J), J=1,N)                                         HHL03050
C                                                                       HHL03060
  140 CONTINUE                                                          HHL03070
      IIX = SVSEED                                                      HHL03080
C                                                                       HHL03090
C-----------------------------------------------------------------------HHL03100
C                                                                       HHL03110
      CALL LANCZS(CMATV,V1,V2,VS,ALPHA,BETA,GR,GC,G,KMAX,MOLD1,N,IIX)   HHL03120
C                                                                       HHL03130
C-----------------------------------------------------------------------HHL03140
C                                                                       HHL03150
C  COMMENTED OUT BELOW BECAUSE KMAX1 IS NOW SET IN PARAMETER LIST       HHL03160
C     KMAX1 = KMAX + 1                                                  HHL03170
C                                                                       HHL03180
      IF (IHIS.EQ.0.AND.ISTOP.GT.0) GO TO 160                           HHL03190
```

```
C
      WRITE(1,150) KMAX,N,SVSEED,MATNO
  150 FORMAT(2I6,I12,I8,' = KMAX,N,SVSEED,MATNO')
C
C     TO AVOID PERTURBATIONS CAUSED BY HEX TO DECIMAL AND DECIMAL TO HEX
C     CONVERSIONS, THE ALPHA AND BETA MUST BE WRITTEN OUT IN HEX.
      WRITE(1,130)(ALPHA(I), I=1,KMAX)
      WRITE(1,130)(BETA(I), I=1,KMAX1)
C     WRITE(1,135)(ALPHA(I), I=1,N)
C     WRITE(1,135)(BETA(I), I=1,N)
  135 FORMAT(4E20.12)
C
C     WRITE(1,130)(V1(I), I=1,N)
C     WRITE(1,130)(V2(I), I=1,N)
C
      IF (ISTOP.EQ.0) GO TO 540
C
  160 CONTINUE
      BKMIN = BTOL
      WRITE(6,170)
  170 FORMAT(/' T-MATRICES (ALPHA AND BETA) ARE NOW AVAILABLE'/)
C
C-----------------------------------------------------------------------
C     SUBROUTINE TNORM CHECKS MIN(BETA)/(ESTIMATED NORM(A)) > BTOL .
C     IF THIS IS VIOLATED IB IS SET EQUAL TO THE NEGATIVE OF THE INDEX
C     OF THE MINIMAL BETA.  IF(IB < 0) THEN SUBROUTINE TNORM IS
C     CALLED FOR EACH VALUE OF MEV TO DETERMINE WHETHER OR NOT THERE
C     IS A BETA IN THE T-MATRIX SPECIFIED THAT VIOLATES THIS TEST.
C     IF THERE IS SUCH A BETA THE PROGRAM TERMINATES FOR THE USER
C     TO DECIDE WHAT TO DO.  THIS TEST CAN BE OVER-RIDDEN BY
C     SIMPLY MAKING BTOL SMALLER, BUT THEN THERE IS THE POSSIBILITY
C     THAT LOSSES IN THE LOCAL ORTHOGONALITY MAY HURT THE COMPUTATIONS.
C     BTOL = 1.D-8 IS HOWEVER A CONSERVATIVE CHOICE FOR BTOL.
C
C     TNORM ALSO COMPUTES TKMAX = MAX(|ALPHA(K)|,BETA(K), K=1,KMAX).
C     TKMAX IS USED TO SCALE THE TOLERANCES USED IN THE
C     T-MULTIPLICITY AND SPURIOUS TESTS IN BISEC. TKMAX IS ALSO USED IN
C     THE PROJECTION TEST FOR HIDDEN EIGENVALUES THAT HAD 'TOO SMALL'
C     A PROJECTION ON THE STARTING VECTOR.
C
      CALL TNORM(ALPHA,BETA,BKMIN,TKMAX,KMAX,IB)
C
C-----------------------------------------------------------------------
C
      TTOL = EPSM*TKMAX
C
C     LOOP ON THE SIZE OF THE T-MATRIX
C
  180 CONTINUE
      MMB = MMB + 1
      MEV = NMEV(MMB)
C     IS MEV TOO LARGE ?
      IF(MEV.LE.KMAX) GO TO 200
      WRITE(6,190) MMB, MEV, KMAX
  190 FORMAT(/' TERMINATE PRIOR TO CONSIDERING THE',I6,'TH T-MATRIX'/
```

```
     1' BECAUSE THE SIZE REQUESTED',I6,' IS GREATER THAN THE MAXIMUM SIZHHL03750
     1E ALLOWED',I6/)                                                  HHL03760
      GO TO 540                                                        HHL03770
C                                                                      HHL03780
  200 MP1 = MEV + 1                                                    HHL03790
      BETAM = BETA(MP1)                                                HHL03800
C                                                                      HHL03810
      IF (IB.GE.0) GO TO 210                                           HHL03820
C                                                                      HHL03830
      T0 = BTOL                                                        HHL03840
C                                                                      HHL03850
C-----------------------------------------------------------------------HHL03860
C                                                                      HHL03870
      CALL TNORM(ALPHA,BETA,T0,T1,MEV,IBMEV)                           HHL03880
C                                                                      HHL03890
C-----------------------------------------------------------------------HHL03900
C                                                                      HHL03910
      TEMP = T0/TKMAX                                                  HHL03920
      IBMEV = IABS(IBMEV)                                              HHL03930
      IF (TEMP.GE.BTOL) GO TO 210                                      HHL03940
      IBMEV = -IBMEV                                                   HHL03950
      GO TO 600                                                        HHL03960
C                                                                      HHL03970
  210 CONTINUE                                                         HHL03980
      IC = MXSTUR-ICT                                                  HHL03990
C                                                                      HHL04000
C-----------------------------------------------------------------------HHL04010
C     BISEC LOOP. THE SUBROUTINE BISEC INCORPORATES DIRECTLY THE       HHL04020
C     T-MULTIPLICITY AND SPURIOUS TESTS. T-EIGENVALUES WILL BE         HHL04030
C     CALCULATED BY BISEC SEQUENTIALLY ON INTERVALS                    HHL04040
C     (LB(J),UB(J)), J = 1,NINT).                                      HHL04050
C                                                                      HHL04060
C     ON RETURN FROM BISEC                                             HHL04070
C     NDIS = NUMBER OF DISTINCT EIGENVALUES OF T(1,MEV) ON UNION       HHL04080
C            OF THE (LB,UB) INTERVALS                                  HHL04090
C     VS = DISTINCT T-EIGENVALUES IN ALGEBRAICALLY INCREASING ORDER    HHL04100
C     MP = MULTIPLICITIES OF THE T-EIGENVALUES IN VS                   HHL04110
C     MP(I) = (0,1,MI), MI>1, I=1,NDIS  MEANS:                         HHL04120
C        (0)  VS(I) IS SPURIOUS                                        HHL04130
C        (1)  VS(I) IS T-SIMPLE AND GOOD                               HHL04140
C        (MI) VS(I) IS MULTIPLE AND IS THEREFORE NOT ONLY GOOD BUT     HHL04150
C             ALSO A CONVERGED GOOD T-EIGENVALUE.                      HHL04160
C     WITHIN BISEC V1 AND V2 ARE DEFINED AS DOUBLE PRECISION ARRAYS    HHL04170
C                                                                      HHL04180
C                                                                      HHL04190
      CALL BISEC(ALPHA,BETA,V1,V2,VS,LB,UB,EPSM,TTOL,MP,NINT,          HHL04200
     1 MEV,NDIS,IC,IWRITE)                                             HHL04210
C                                                                      HHL04220
C-----------------------------------------------------------------------HHL04230
C                                                                      HHL04240
      IF (NDIS.EQ.0) GO TO 620                                         HHL04250
C                                                                      HHL04260
C     COMPUTE THE TOTAL NUMBER OF STURM SEQUENCES USED TO DATE         HHL04270
C     COMPUTE THE BISEC CONVERGENCE AND T-MULTIPLICITY TOLERANCES USED. HHL04280
C     COMPUTE THE CONVERGENCE TOLERANCE FOR EIGENVALUES OF A.          HHL04290
```

```
      ICT = ICT + IC
      TEMP = DFLOAT(MEV+1000)
      MULTOL = TEMP*TTOL
      TEMP = DSQRT(TEMP)
      BISTOL = TTOL*TEMP
      CONTOL = BETAM*1.D-10
C
C-----------------------------------------------------------------------
C    SUBROUTINE LUMP 'COMBINES' T-EIGENVALUES THAT ARE 'TOO CLOSE'.
C    NOTE HOWEVER THAT CLOSE SPURIOUS T-EIGENVALUES ARE NOT AVERAGED
C    WITH GOOD ONES. HOWEVER, THEY MAY BE USED TO INCREASE THE
C    MULTIPLICITY OF A GOOD T-EIGENVALUE.
C
      LOOP = NDIS
      CALL LUMP(VS,RELTOL,MULTOL,SCALE2,MP,LOOP)
C
C-----------------------------------------------------------------------
C
      IF(NDIS.EQ.LOOP) GO TO 230
C
      WRITE(6,220) NDIS, MEV, LOOP
  220 FORMAT(/I6,' DISTINCT T-EIGENVALUES WERE COMPUTED IN BISEC AT MEV
     1',I6/ 2X,' LUMP SUBROUTINE REDUCES NUMBER OF DISTINCT EIGENVALUES
     1TO',I6)
C
  230 CONTINUE
      NDIS = LOOP
      BETA(MP1) = BETAM
C
C-----------------------------------------------------------------------
C     THE SUBROUTINE ISOEV LABELS THOSE SIMPLE EIGENVALUES OF T(1,MEV)
C     WITH VERY SMALL GAPS BETWEEN NEIGHBORING EIGENVALUES OF T(1,MEV)
C     TO AVOID COMPUTING ERROR ESTIMATES FOR ANY SIMPLE GOOD
C     T-EIGENVALUE THAT IS TOO CLOSE TO A SPURIOUS T-EIGENVALUE.
C     ON RETURN FROM ISOEV, G CONTAINS CODED MINIMAL GAPS
C     BETWEEN THE DISTINCT EIGENVALUES OF T(1,MEV). (G IS REAL).
C     G(I) < 0 MEANS MINGAP IS DUE TO LEFT GAP G(I) > 0 MEANS DUE TO
C     RIGHT GAP. MP(I) = -1 MEANS THAT THE GOOD T-EIGENVALUE IS SIMPLE
C     AND HAS A VERY SMALL MINGAP IN T(1,MEV) DUE TO A SPURIOUS
C     T-EIGENVALUE.   NG = NUMBER OF GOOD EIGENVALUES.
C     NISO = NUMBER OF ISOLATED GOOD T-EIGENVALUES.
C
      CALL ISOEV(VS,GAPTOL,MULTOL,SCALE1,G,MP,NDIS,NG,NISO)
C
C-----------------------------------------------------------------------
C
      WRITE(6,240)NG,NISO,NDIS
  240 FORMAT(/I6,' GOOD T-EIGENVALUES HAVE BEEN COMPUTED'/
     1 I6,' OF THESE ARE T-ISOLATED'/
     2 I6,' = NUMBER OF DISTINCT T-EIGENVALUES COMPUTED'/)
C
C     DO WE WRITE DISTINCT EIGENVALUES OF T-MATRIX TO FILE 4?
      IF (IDIST.EQ.0) GO TO 280
C
      WRITE(11,250) NDIS,NISO,MEV,N,SVSEED,MATNO
```

```
  250 FORMAT(/4I6,I12,I8,' = NDIS,NISO,MEV,N,SVSEED,MATNO'/)
C
      WRITE(11,260) (MP(I),VS(I),G(I), I=1,NDIS)
  260 FORMAT(2(I3,E25.16,E12.3))
C
      WRITE(11,270) NDIS, (MP(I), I=1,NDIS)
  270 FORMAT(/I6,' = NDIS, T-MULTIPLICITIES (0 MEANS  SPURIOUS)'/(20I4))
C
  280 CONTINUE
C
      IF (NISO.NE.0) GO TO 310
C
      WRITE(4,290) MEV
  290 FORMAT(/' AT MEV = ',I6,' THERE ARE NO ISOLATED T-EIGENVALUES'/
     1' SO NO ERROR ESTIMATES WERE COMPUTED/')
C
      WRITE(6,300)
  300 FORMAT(/' ALL COMPUTED GOOD T-EIGENVALUES ARE MULTIPLE'/
     1 ' THEREFORE ALL SUCH EIGENVALUES ARE ASSUMED TO HAVE CONVERGED')
C
      ICONV = 1
      GO TO 350
C
  310 CONTINUE
C
C-----------------------------------------------------------------------
C     SUBROUTINE INVERR COMPUTES ERROR ESTIMATES FOR ISOLATED GOOD
C     T-EIGENVALUES USING INVERSE ITERATION ON T(1,MEV). ON RETURN
C     G(J) = MINIMUM GAP IN T(1,MEV) FOR EACH VS(J), J=1,NDIS
C     G(MEV+I) = BETAM*|U(MEV)| = ERROR ESTIMATE FOR ISOLATED GOOD
C              T-EIGENVALUES, WHERE I = 1, NISO AND  BETAM = BETA(MEV+1)
C              U(MEV) IS MEVTH COMPONENT OF THE UNIT EIGENVECTOR OF T
C              CORRESPONDING TO THE ITH ISOLATED GOOD T-EIGENVALUE.
C     A NEGATIVE ERROR ESTIMATE MEANS THAT FOR THAT PARTICULAR
C     EIGENVALUE THE INVERSE ITERATION DID NOT CONVERGE IN <= MXINIT
C     STEPS AND THAT THE CORRESPONDING ERROR ESTIMATE IS QUESTIONABLE.
C
C     V2 CONTAINS THE ISOLATED GOOD T-EIGENVALUES
C     V1 CONTAINS THE MINGAPS TO THE NEAREST DISTINCT  EIGENVALUE
C        OF T(1,MEV) FOR EACH ISOLATED GOOD EIGENVALUE IN V2.
C     VS CONTAINS THE NDIS DISTINCT EIGENVALUES OF T(1,MEV)
C     MP CONTAINS THE CORRESPONDING CODED T-MULTIPLICITIES
C     WITHIN INVERR V1 AND V2 ARE DOUBLE PRECISION ARRAYS
C
      IT = MXINIT
      CALL INVERR(ALPHA,BETA,V1,V2,VS,EPSM,G,MP,MEV,MMB,NDIS,NISO,N,
     1 RHSEED,IT,IWRITE)
C
C-----------------------------------------------------------------------
C
C     SIMPLE CHECK FOR CONVERGENCE. CHECKS TO SEE IF ALL OF THE ERROR
C     ESTIMATES ARE SMALLER THAN CONTOL.
C     IF THIS TEST IS SATISFIED, THEN CONVERGENCE FLAG, ICONV IS SET
C     TO 1.  TYPICALLY ERROR ESTIMATES ARE VERY CONSERVATIVE.
C
```

```
      WRITE(6,320) CONTOL
  320 FORMAT(/' CONVERGENCE IS TESTED USING THE CONVERGENCE TOLERANCE',
     1E13.4/)
C
      II = MEV +1
      IF = MEV+NISO
      DO 330 I = II,IF
      IF (ABS(G(I)).GT.CONTOL) GO TO 350
  330 CONTINUE
      ICONV = 1
      MMB = NMEVS
C
      WRITE(6,340) CONTOL
  340 FORMAT(' ALL COMPUTED ERROR ESTIMATES WERE LESS THAN',E15.4/
     1 ' THEREFORE PROCEDURE TERMINATES'/)
C
  350 CONTINUE
C
C     IF CONVERGENCE IS INDICATED, THAT IS ICONV = 1 ,THEN
C     THE SUBROUTINE PRTEST IS CALLED TO CHECK FOR ANY CONVERGED
C     EIGENVALUES THAT HAVE BEEN MISLABELLED AS SPURIOUS BECAUSE
C     THE PROJECTION OF THEIR EIGENVECTOR(S) ON THE STARTING
C     VECTOR WERE TOO SMALL.
C     NUMERICAL TESTS INDICATE THAT SUCH EIGENVALUES ARE RARE.
C     IF FOR SOME REASON MANY OF THESE HIDDEN EIGENVALUES APPEAR
C     ON SOME RUN, YOU CAN BE CERTAIN THAT SOMETHING IS FOULED UP.
C
      IF (ICONV.EQ.0) GO TO 480
C
C-----------------------------------------------------------------------
C
      CALL PRTEST (ALPHA,BETA,VS,TKMAX,EPSM,RELTOL,SCALE3,SCALE4,
     1 MP,NDIS,MEV,IPROJ)
C
C-----------------------------------------------------------------------
C
      IF(IPROJ.EQ.0) GO TO 470
C
      IF(IDIST.EQ.1)  WRITE(11,360) IPROJ
  360 FORMAT(' SUBROUTINE PRTEST WANTS TO RELABEL',I6,' SPURIOUS EIGENVA
     1LUES'/' WE ACCEPT RELABELLING ONLY IF LAST COMPONENT OF T-EIGENVEC
     1TOR IS L.T. 1.D-10'/)
C
      IIX = RHSEED
C
C-----------------------------------------------------------------------
C
      CALL GENRAN(IIX,G,MEV)
C
C-----------------------------------------------------------------------
C
      ITEN = -10
      NISOM = NISO + MEV
      IWRITO = IWRITE
      IWRITE = 0
```

```
C                                                                       HHL05950
      DO 390 J = 1,NDIS                                                 HHL05960
      IF(MP(J).NE.ITEN) GO TO 390                                       HHL05970
      T0 = VS(J)                                                        HHL05980
C                                                                       HHL05990
C-----------------------------------------------------------------------HHL06000
C                                                                       HHL06010
      IT = MXINIT                                                       HHL06020
      CALL INVERM(ALPHA,BETA,V1,V2,T0,TEMP,T1,EPSM,G,MEV,IT,IWRITE)     HHL06030
C                                                                       HHL06040
C-----------------------------------------------------------------------HHL06050
C                                                                       HHL06060
      IF(TEMP.LE.1.D-10) GO TO 380                                      HHL06070
C     ERROR ESTIMATE WAS NOT SMALL REJECT RELABELLING OF THIS EIGENVALUEHHL06080
      IF(IDIST.EQ.1)  WRITE(11,370) J,T0,TEMP                           HHL06090
  370 FORMAT(/' LAST COMPONENT FOR',I6,'TH EIGENVALUE',E20.12/' IS TOO LHHL06100
     1ARGE = ',E15.6,' SO DO NOT ACCEPT PRTEST RELABELLING'/)           HHL06110
      MP(J) = 0                                                         HHL06120
      IPROJ = IPROJ - 1                                                 HHL06130
      GO TO 390                                                         HHL06140
C     RELABELLING ACCEPTED                                              HHL06150
  380 NISOM = NISOM + 1                                                 HHL06160
      G(NISOM) = BETAM*TEMP                                             HHL06170
  390 CONTINUE                                                          HHL06180
      IWRITE = IWRITO                                                   HHL06190
C                                                                       HHL06200
      IF(IPROJ.EQ.0) GO TO 430                                          HHL06210
      WRITE(6,400) IPROJ                                                HHL06220
  400 FORMAT(/I6,' T-EIGENVALUES WERE RECLASSIFIED AS GOOD.'/           HHL06230
     1' THESE ARE IDENTIFIED IN FILE 3 BY A T-MULTIPLICITY OF -10'/' USEHHL06240
     2R SHOULD INSPECT EACH TO MAKE SURE NEIGHBORS HAVE CONVERGED'/)    HHL06250
C                                                                       HHL06260
      IF(IDIST.EQ.1)  WRITE(11,410) IPROJ                               HHL06270
  410 FORMAT(/I6,' T-EIGENVALUES WERE RELABELLED AS GOOD'/              HHL06280
     1' BELOW IS CORRECTED T-MULTIPLICITY PATTERN'/)                    HHL06290
C                                                                       HHL06300
      WRITE(6,420) NDIS, (MP(I), I=1,NDIS)                              HHL06310
      IF(IDIST.EQ.1)  WRITE(11,420) NDIS, (MP(I), I=1,NDIS)             HHL06320
  420 FORMAT(/I6,' = NDIS, T-MULTIPLICITIES (0 MEANS  SPURIOUS)'/       HHL06330
     1 6X, ' (-10) MEANS SPURIOUS T-EIGENVALUE RELABELLED AS GOOD'/(20I4HHL06340
     1))                                                                HHL06350
C                                                                       HHL06360
C     RECALCULATE MINGAPS FOR DISTINCT T(1,MEV) EIGENVALUES.            HHL06370
  430 NM1 = NDIS - 1                                                    HHL06380
      G(NDIS) = VS(NM1)-VS(NDIS)                                        HHL06390
      G(1) = VS(2)-VS(1)                                                HHL06400
C                                                                       HHL06410
      DO 440 J = 2,NM1                                                  HHL06420
      T0 = VS(J)-VS(J-1)                                                HHL06430
      T1 = VS(J+1)-VS(J)                                                HHL06440
      G(J) = T1                                                         HHL06450
      IF (T0.LT.T1) G(J) = -T0                                          HHL06460
  440 CONTINUE                                                          HHL06470
      IF(IPROJ.EQ.0) GO TO 470                                          HHL06480
C     WRITE TO FILE 4 ERROR ESTIMATES FOR THOSE T-EIGENVALUES RELABELLEDHHL06490
```

```fortran
      NGOOD = 0
      DO 450 J = 1,NDIS
      IF(MP(J).EQ.0) GO TO 450
      NGOOD = NGOOD + 1
      IF(MP(J).NE.ITEN) GO TO 450
      T0 = VS(J)
      NISO = NISO + 1
      NISOM = MEV + NISO
      WRITE(4,460) NGOOD,T0,G(NISOM),G(J)
  450 CONTINUE
  460 FORMAT(I10,E25.16,2E14.3)
C
  470 CONTINUE
C
C     WRITE THE GOOD T-EIGENVALUES TO FILE 3.  FIRST TRANSFER THEM
C     TO V2 AND THEIR T-MULTIPLICITIES TO THE CORRESPONDING POSITIONS
C     IN MP AND COMPUTE THE A-MINGAPS, THE MINIMAL GAPS BETWEEN THE
C     GOOD T-EIGENVALUES.  THESE GAPS WILL BE PUT IN THE ARRAY G.
C     SINCE G CURRENTLY CONTAINS THE MINIMAL GAPS BETWEEN THE DISTINCT
C     EIGENVALUES OF THE T-MATRIX, THESE GAPS WILL FIRST BE
C     TRANSFERRED TO GC.  NOTE THAT GC<0 MEANS THAT THAT MINIMAL GAP
C     IN THE T-MATRIX IS DUE TO A SPURIOUS T-EIGENVALUE.
C     ALL THIS INFORMATION IS PRINTED TO FILE 3
C
  480 CONTINUE
C
      NG = 0
      DO 490 I = 1,NDIS
      IF (MP(I).EQ.0) GO TO 490
      NG = NG+1
      MP(NG) = MP(I)
      GR(NG) = VS(I)
      TEMP = G(I)
      TEMP = DABS(TEMP)
      J = I+1
      IF (G(I).LT.ZERO) J = I-1
      IF (MP(J).EQ.0) TEMP = -TEMP
      GC(NG) = TEMP
  490 CONTINUE
C
      WRITE(6,500)MEV
  500 FORMAT(//' T-EIGENVALUE CALCULATION AT MEV = ',I6,'    IS COMPLETE
     1')
C
C     NG = NUMBER OF COMPUTED DISTINCT GOOD T-EIGENVALUES.  NEXT
C     GENERATE GAPS BETWEEN GOOD T-EIGENVALUES (AMINGAPS) AND PUT THEM
C     IN G.  G(J) < 0 MEANS THE AMINGAP IS DUE TO THE LEFT-HAND GAP.
C
      NGM1 = NG - 1
      G(NG) = GR(NGM1)-GR(NG)
      G(1) = GR(2)-GR(1)
C
      DO 510 J = 2,NGM1
      T0 = GR(J)-GR(J-1)
      T1 = GR(J+1)-GR(J)
```

```
      G(J) = T1                                                         HHL07050
      IF (T0.LT.T1) G(J) = -T0                                          HHL07060
  510 CONTINUE                                                          HHL07070
C                                                                       HHL07080
C     WRITE GOOD T-EIGENVALUES OUT TO FILE 3.                           HHL07090
C                                                                       HHL07100
      WRITE(3,520)NG,NDIS,MEV,N,SVSEED,MATNO,MULTOL,IB,BTOL             HHL07110
  520 FORMAT(4I6,I12,I8,' = NG,NDIS,MEV,N,SVEED,MATNO'/                 HHL07120
     1 E20.12,I6,E13.4,' = MUTOL,INDEX MINIMAL BETA,BTOL'/              HHL07130
     1' EV NO',2X,'TMULT',7X,'GOOD T-EIGENVALUE',7X,'TMINGAP',7X,'AMINGAHHL07140
     1P')                                                               HHL07150
C                                                                       HHL07160
      WRITE(3,530)(I,MP(I),GR(I),GC(I),G(I), I=1,NG)                    HHL07170
  530 FORMAT(2I6,E25.16,2E14.3)                                         HHL07180
C                                                                       HHL07190
C     IF CONVERGENCE FLAG ICONV.NE.1 AND NUMBER OF T-MATRICES           HHL07200
C     CONSIDERED TO DATE IS LESS THAN NUMBER ALLOWED, INCREMENT MEV.    HHL07210
C     AND LOOP BACK TO 210 TO REPEAT COMPUTATIONS.  RESTORE BETA(MEV+1).HHL07220
C                                                                       HHL07230
      BETA(MP1) = BETAM                                                 HHL07240
C                                                                       HHL07250
      IF (MMB.LT.NMEVS.AND.ICONV.NE.1) GO TO 180                        HHL07260
C                                                                       HHL07270
C     END OF LOOP ON DIFFERENT SIZE T-MATRICES ALLOWED.                 HHL07280
C                                                                       HHL07290
  540 CONTINUE                                                          HHL07300
C                                                                       HHL07310
      IF(ISTOP.EQ.0)  WRITE(6,550)                                      HHL07320
  550 FORMAT(/' T-MATRICES (ALPHA AND BETA) ARE NOW AVAILABLE, TERMINATEHHL07330
     1')                                                                HHL07340
      IF (IHIS.EQ.1.AND.KMAX.NE.MOLD) WRITE(1,560)                      HHL07350
  560 FORMAT(/' ABOVE ARE THE FOLLOWING VECTORS '/                      HHL07360
     1 '  ALPHA(I), I = 1,KMAX'/                                        HHL07370
     2 '  BETA(I), I = 1,KMAX+1'/                                       HHL07380
     3 ' FINAL TWO LANCZOS VECTORS OF ORDER N FOR I = KMAX,KMAX+1'/     HHL07390
     4 ' ALL ENTRIES IN THIS FILE HAVE FORMAT 4Z20 '/                   HHL07400
     5 ' ----- END OF FILE 1 NEW ALPHA, BETA HISTORY---------------'///)HHL07410
C                                                                       HHL07420
      IF (ISTOP.EQ.0) GO TO 640                                         HHL07430
C                                                                       HHL07440
      WRITE(3,570)                                                      HHL07450
  570 FORMAT(/' ABOVE ARE COMPUTED GOOD T-EIGENVALUES'/                 HHL07460
     1 ' NG = NUMBER OF GOOD T-EIGENVALUES COMPUTED'/                   HHL07470
     2 ' NDIS = NUMBER OF COMPUTED DISTINCT EIGENVALUES OF T(1,MEV)'/   HHL07480
     3 ' N = ORDER OF A,  MATNO = MATRIX IDENT'/                        HHL07490
     4 ' MULTOL = MULTIPLICITY TOLERANCE FOR T-EIGENVALUES IN BISEC'/   HHL07500
     4 ' TMULT IS THE T-MULTIPLICITY OF GOOD T-EIGENVALUE'/             HHL07510
     5 ' TMULT = -1 MEANS SPURIOUS T-EIGENVALUE TOO CLOSE'/             HHL07520
     6 ' DO NOT COMPUTE ERROR ESTIMATES FOR SUCH EIGENVALUES'/          HHL07530
     7 ' AMINGAP = MINIMAL GAP BETWEEN THE COMPUTED A-EIGENVALUES'/     HHL07540
     8 ' AMINGAP .LT. 0. MEANS MINIMAL GAP IS DUE TO LEFT-HAND GAP'/    HHL07550
     9 ' TMINGAP= MINIMAL GAP W.R.T.  DISTINCT EIGENVALUES IN T(1,MEV)'/HHL07560
     1 ' TMINGAP .LT. 0. MEANS MINGAP IS DUE TO SPURIOUS EIGENVALUE'/   HHL07570
     2 ' ----- END OF FILE 3 GOODEIGENVALUES-----------------------'///)HHL07580
C                                                                       HHL07590
```

```
      IF (IDIST.EQ.1) WRITE(11,580)                                     HHL07600
  580 FORMAT(/' ABOVE ARE THE DISTINCT EIGENVALUES OF T(1,MEV).'/       HHL07610
     2 ' THE FORMAT IS      T-MULTIPLICITY    EIGENVALUE   TMINGAP'/    HHL07620
     3 '        THIS FORMAT IS REPEATED TWICE ON EACH LINE.'/           HHL07630
     4 ' T-MULTIPLICITY = -1 MEANS THAT THE SUBROUTINE ISOEV HAS TAGGED'HHL07640
     5 /'   THIS SIMPLE T-EIGENVALUE AS HAVING A VERY CLOSE SPURIOUS'/  HHL07650
     6 '   T-EIGENVALUE SO THAT NO ERROR ESTIMATE WILL BE COMPUTED'/    HHL07660
     7 '   FOR THAT EIGENVALUE IN SUBROUTINE INVERR.'/                  HHL07670
     8 ' TMINGAP .LT. 0, TMINGAP IS DUE TO LEFT GAP .GT. 0, RIGHT GAP.'/HHL07680
     9 ' EACH OF THE DISTINCT T-EIGENVALUE TABLES IS FOLLOWED'/         HHL07690
     9 ' BY THE T-MULTIPLICITY PATTERN.'/                               HHL07700
     1 ' NDIS = NUMBER OF COMPUTED DISTINCT EIGENVALUES OF T(1,MEV).'/  HHL07710
     2 ' NG = NUMBER OF GOOD T-EIGENVALUES. '/                          HHL07720
     3 ' NISO = NUMBER OF ISOLATED GOOD T-EIGENVALUES. '/               HHL07730
     4 ' NISO ALSO IS THE COUNT OF +1 ENTRIES IN T-MULTIPLICITY PATTERN.HHL07740
     5 '/' ----- END OF FILE 4 DISTINCT T-EIGENVALUES----------------'//HHL07750
     6 /)                                                               HHL07760
C                                                                       HHL07770
      IF(NISO.NE.0)  WRITE(4,590)                                       HHL07780
  590 FORMAT(/' ABOVE ARE THE ERROR ESTIMATES OBTAINED FOR THE ISOLATED HHL07790
     1GOOD EIGENVALUES'/                                                HHL07800
     1' OBTAINED VIA INVERSE ITERATION IN THE SUBROUTINE INVERR.'/      HHL07810
     1' ALL OTHER GOOD EIGENVALUES HAVE CONVERGED.'/                    HHL07820
     2' ERROR ESTIMATE = BETAM*ABS(UM)'/                                HHL07830
     2' WHERE BETAM = BETA(MEV+1) AND UM = U(MEV).'/                    HHL07840
     3' U = UNIT EIGENVECTOR OF T WHERE T*U = EV*U AND EV = ISOLATED GOOHHL07850
     3D EIGENVALUE.'/                                                   HHL07860
     4' TMINGAP = GAP TO NEAREST DISTINCT EIGENVALUE OF T(1,MEV).'/     HHL07870
     5' TMINGAP .LT. 0. MEANS MINGAP IS DUE TO LEFT NEIGHBOR.'/         HHL07880
     6' ERROR ESTIMATE L.T. 0 MEANS INVERSE ITERATION DID NOT CONVERGE'/HHL07890
     7' ------ END OF FILE 7 ERRINV ------------------------------'//) HHL07900
      GO TO 640                                                         HHL07910
C                                                                       HHL07920
  600 CONTINUE                                                          HHL07930
C                                                                       HHL07940
      IBB = IABS(IBMEV)                                                 HHL07950
      IF (IBMEV.LT.0) WRITE(6,610) MEV,IBB,BETA(IBB)                    HHL07960
  610 FORMAT(/' PROGRAM TERMINATES BECAUSE MEV REQUESTED = ',I6,' IS .GTHHL07970
     1',I6/' AT WHICH AN ABNORMALLY SMALL BETA = ' , E13.4,' OCCURRED'/)HHL07980
      GO TO 640                                                         HHL07990
C                                                                       HHL08000
  620 IF (NDIS.EQ.0.AND.ISTOP.GT.0) WRITE(6,630)                        HHL08010
  630 FORMAT(/' INTERVALS SPECIFIED FOR BISECT DID NOT CONTAIN ANY EIGENHHL08020
     1VALUES'/' PROGRAM TERMINATES')                                    HHL08030
C                                                                       HHL08040
  640 CONTINUE                                                          HHL08050
C                                                                       HHL08060
      STOP                                                              HHL08070
C-----END OF MAIN PROGRAM FOR LANCZOS HERMITIAN EIGENVALUE COMPUTATIONS-HHL08080
      END                                                               HHL08090
```

3.3 HLEVEC: Main Program, Eigenvector Computations

```
C     MODIFIED 8/16/83 ( and 4/27/93 to change array dimensioning)      HHL00010
C                                                                       HHL00020
C-----HLEVEC (EIGENVECTORS OF HERMITIAN MATRICES)-----------------------HHL00030
C                                                                       HHL00040
C  Authors:  Jane Cullum and Ralph A. Willoughby (deceased)             HHL00050
C            Los Alamos National Laboratory                             HHL00060
C            Los Alamos, New Mexico 87544                               HHL00070
C            E-Mail: cullumj@lanl.gov                                   HHL00075
c                                                                       HHL00080
c  These codes are copyrighted by the authors.  These codes             HHL00090
c  and modifications of them or portions of them are NOT to be          HHL00100
c  incorporated into any commercial codes without legal agreements      HHL00110
c  with the authors.  If these codes or portions of them                HHL00120
c  are used in other scientific or engineering research works          HHL00130
c  the names of the authors of these codes and appropriate              HHL00140
c  references to their written work are to be incorporated in the      HHL00150
c  derivative works.                                                    HHL00160
c                                                                       HHL00170
c  This header is not to be removed from these codes.                   HHL00180
C                                                                       HHL00181
C        REFERENCE: Cullum and Willoughby, Chapters 1,2,3,4             HHL00182
C        Lanczos Algorithms for Large Symmetric Eigenvalue ComputationsHHL00183
C        VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in    HHL00184
C        Applied Mathematics, 2002. SIAM Publications,                  HHL00185
C        Philadelphia, PA. USA                                          HHL00186
C                                                                       HHL00187
c                                                                       HHL00190
C     CONTAINS MAIN PROGRAM FOR COMPUTING AN EIGENVECTOR CORRESPONDING  HHL00200
C     TO EACH OF A SET OF EIGENVALUES THAT HAVE BEEN COMPUTED           HHL00210
C     ACCURATELY BY THE CORRESPONDING LANCZOS EIGENVALUE PROGRAM        HHL00220
C     (HLEVAL) FOR HERMITIAN MATRICES.  THIS PROGRAM COULD BE           HHL00230
C     MODIFIED TO COMPUTE ADDITIONAL EIGENVECTORS FOR ANY               HHL00240
C     MULTIPLE EIGENVALUE OF THE GIVEN A-MATRIX.  THE AMOUNT OF         HHL00250
C     ADDITIONAL COMPUTATION REQUIRED BY SUCH A MODIFICATION WOULD      HHL00260
C     DEPEND UPON THE GIVEN MATRIX AND UPON WHICH PART OF THE           HHL00270
C     SPECTRUM WAS INVOLVED.                                            HHL00280
C                                                                       HHL00290
C     THE LANCZOS EIGENVECTOR COMPUTATIONS ASSUME THAT EACH             HHL00300
C     EIGENVALUE THAT IS BEING CONSIDERED HAS CONVERGED AS AN           HHL00310
C     EIGENVALUE OF THE LANCZOS TRIDIAGONAL MATRICES.                   HHL00320
C                                                                       HHL00330
C     PORTABILITY:                                                      HHL00340
C     THIS PROGRAM IS NOT PORTABLE BECAUSE OF THE USE OF COMPLEX*16     HHL00350
C     VARIABLES.  MOREOVER, THE PFORT VERIFIER IDENTIFIED THE           HHL00360
C     FOLLOWING ADDITIONAL NONPORTABLE CONSTRUCTIONS:                   HHL00370
C                                                                       HHL00380
C     1.  DATA/MACHEP/ STATEMENT                                        HHL00390
C     2.  ALL READ(5,*) STATEMENTS (FREE FORMAT)                        HHL00400
C     3.  FORMAT(20A4) USED WITH THE EXPLANATORY HEADER, EXPLAN         HHL00410
C     4.  HEXADECIMAL FORMAT (4Z20) USED IN ALPHA/BETA FILES 1 AND 2.   HHL00420
C                                                                       HHL00430
```

```
C     IMPORTANT NOTE:  PROGRAM ALLOWS ENLARGEMENT OF THE ALPHA, BETA    HHL00440
C     ARRAYS.  IN PARTICULAR, IF ANY ONE OF THE EIGENVALUES SUPPLIED    HHL00450
C     IS T-SIMPLE AND NOT CLOSE TO A SPURIOUS T-EIGENVALUE, THE PROGRAM HHL00460
C     REQUIRES THAT KMAX BE AT LEAST 11*MEV/8 + 12.  IF KMAX IS NOT     HHL00470
C     THIS LARGE, THEN THE PROGRAM WILL RESET KMAX TO THIS SIZE         HHL00480
C     AND EXTEND THE ALPHA, BETA HISTORY IF REQUIRED.                   HHL00490
C     THUS, THE DIMENSIONS OF THE ALPHA AND BETA ARRAYS MUST BE         HHL00500
C     LARGE ENOUGH TO ALLOW FOR THIS POSSIBILITY.                       HHL00510
C     REMEMBER THAT THE BETA ARRAY, BETA(J), IS SUCH THAT               HHL00520
C     J = 1,..., KMAX+1.  SO IF THE KMAX USED BY THE PROGRAM            HHL00530
C     IS TO BE 3000, THEN BETA MUST BE OF LENGTH AT LEAST 3001.         HHL00540
C                                                                       HHL00550
C     TO AVOID USING MAX(I,J) IN THE PARAMETER LISTING WE HAVE USED     HHL00560
C     THE FOLLOWING EQUIVALENT RELATIONSHIP                             HHL00570
C                                                                       HHL00580
C     MAX(I,J) = ( 2*I/(I+J))*I  + (2*J/(I+J))*J                        HHL00590
C                                                                       HHL00600
      parameter (n=625,mev=1500,ngood= 3,nngood=n*ngood)                HHL00610
      parameter (kmaxn = (3*mev)/2 + 12, kmaxn1=kmaxn+1)                HHL00620
      parameter(nkmaxn = ngood*kmaxn)                                   HHL00630
      PARAMETER ( KMAXn1 = KMAXn+1, KMAXn12 = KMAXn1/2 )                HHL00640
      PARAMETER ( NKMAXn2 = N+KMAXn12, NPKMAXn = N+KMAXn)               HHL00650
      PARAMETER (KVS = ((2*N)/NPKMAXn)*N + ((2*KMAXn)/NPKMAXn)*KMAXn )  HHL00660
      PARAMETER (KV2 = ((2*N)/NKMAXn2)*N +((2*KMAXn12)/NKMAXn2)*KMAXn12)HHL00670
C-----------------------------------------------------------------------HHL00680
      COMPLEX*16  V1(kv2),V2(n),VS(n),RITVEC(nngood),ZEROC,TEMPC        HHL00690
      DOUBLE PRECISION  ALPHA(kmaxn),BETA(kmaxn1),GR(n),GC(n)           HHL00700
      DOUBLE PRECISION  TVEC(nkmaxm),GOODEV(ngood),EVNEW(ngood)         HHL00710
      DOUBLE PRECISION  EVAL,EVALN,TOLN,TTOL,ERTOL,ALFA,BATA            HHL00720
      DOUBLE PRECISION  MULTOL,SCALE0,STUTOL,BTOL,LB,UB                 HHL00730
      DOUBLE PRECISION  ONE,ZERO,MACHEP,EPSM,TEMP,SUM,ERRMIN,BKMIN      HHL00740
      DOUBLE PRECISION  RELTOL,ERROR,TERROR,TLAST(ngood)                HHL00750
      REAL  G(kvs),AMINGP(ngood),TMINGP(ngood),EXPLAN(20)               HHL00760
      REAL  TERR(ngood),ERR(ngood),ERRDGP(ngood),RNORM(ngood)           HHL00770
      real TBETA(ngood)                                                 HHL00780
      INTEGER  MP(ngood),M1(ngood),M2(ngood),MA(ngood)                  HHL00790
      integer ML(ngood),MINT(ngood),MFIN(ngood)                         HHL00800
      INTEGER  SVSEED,SVSOLD,RHSEED,IDELTA(ngood),MULEVA(ngood)         HHL00810
      INTEGER  MBOUND,NTVCON,SVTVEC,TVSTOP,LVCONT,ERCONT,TFLAG          HHL00820
      DOUBLE PRECISION  DABS, DMAX1, DSQRT, DFLOAT                      HHL00830
      REAL ABS                                                          HHL00840
      INTEGER  IABS                                                     HHL00850
C-----------------------------------------------------------------------HHL00860
      EXTERNAL CMATV                                                    HHL00870
      DATA MACHEP/Z3410000000000000/                                    HHL00880
      EPSM = 2.D0*MACHEP                                                HHL00890
C-----------------------------------------------------------------------HHL00900
C                                                                       HHL00910
C     ARRAYS MUST BE DIMENSIONED AS FOLLOWS:                            HHL00920
C     1.  ALPHA:  >= KMAXN,  BETA: >= (KMAXN+1) WHERE KMAXN, THE        HHL00930
C                 LARGEST SIZE T-MATRIX CONSIDERED BY THE PROGRAM,      HHL00940
C                 IS THE LARGER OF THE SIZE OF THE ALPHA, BETA HISTORY  HHL00950
C                 PROVIDED ON FILE 2 (IF ANY ) AND THE SIZE WHICH THE   HHL00960
C                 PROGRAM SPECIFIES INTERNALLY, THIS LATTER IS ALWAYS   HHL00970
C                 < = 11*MEV / 8  +  12, WHERE MEV IS THE SIZE          HHL00980
```

```
C                T-MATRIX THAT WAS USED IN THE CORRESPONDING EIGENVALUE
C                COMPUTATIONS.
C     2.  V1:  >= MAX(N,KMAX/2)
C     3.  V2, VS:   >= N
C     4.  G:  >= MAX(N,KMAX).   GR, GC:  >= N
C     5.  RITVEC:  >= N*NGOOD, WHERE NGOOD IS NUMBER OF EIGENVALUES
C                  SUPPLIED TO THIS PROGRAM.
C     6.  TVEC:  >= CUMULATIVE LENGTH OF ALL THE T-EIGENVECTORS
C         NEEDED TO GENERATE THE DESIRED RITZ VECTORS.  AN EDUCATED
C         GUESS AT AN APPROPRIATE LENGTH CAN BE OBTAINED BY RUNNING THE
C         PROGRAM WITH THE FLAG MBOUND = 1 AND MULTIPLYING THE
C         RESULTING SIZE BY 5/4.
C     7.  GOODEV, EVNEW, AMINGP, TMINGP, TERR, ERR, ERRGDP, RNORM, TBETA
C         TLAST, MP, MA, M1, M2, MINT, MFIN, MULEVA, AND IDELTA ALL
C         MUST BE AT LEAST NGOOD.
C
C-----------------------------------------------------------------------
C     OUTPUT HEADER
      WRITE(6,10)
   10 FORMAT(/' LANCZOS EIGENVECTOR PROCEDURE FOR HERMITIAN MATRICES'/)
C
C     SET PROGRAM PARAMETERS
C     USER MUST NOT MODIFY SCALE0
      SCALE0 = 5.0D0
      ZERO = 0.0D0
      ZEROC = DCMPLX(ZERO,ZERO)
      ONE = 1.0D0
      MPMIN = -1000
C     CONVERGENCE TOLERANCE FOR T-EIGENVECTORS FOR RITZ VECTORS
      ERTOL = 1.D-10
      ISREAL = 0
C
C     READ USER-SPECIFIED PARAMETER FROM INPUT FILE 5 (FREE FORMAT)
C
C     READ USER-PROVIDED HEADER FOR RUN
      READ(5,20) EXPLAN
      WRITE(6,20) EXPLAN
   20 FORMAT(20A4)
C
C     READ IN THE MAXIMUM PERMISSIBLE DIMENSIONS FOR THE TVEC ARRAY
C     (MDIMTV), FOR THE RITVEC ARRAY (MDIMRV), AND FOR THE BETA
C     ARRAY (MBETA).
C
      READ(5,20) EXPLAN
      READ(5,*) MDIMTV, MDIMRV, MBETA
C
C     READ IN RELATIVE TOLERANCE (RELTOL) USED IN DETERMINING
C     APPROPRIATE SIZES FOR THE T-MATRICES USED IN THE EIGENVECTOR
C     COMPUTATIONS
C
      READ(5,20) EXPLAN
      READ(5,*) RELTOL
C
C
C     SET FLAGS TO 0 OR 1:
```

```
C     MBOUND = 1:  PROGRAM TERMINATES AFTER COMPUTING 1ST GUESSES       HHL01540
C                  ON APPROPRIATE T-SIZES FOR USE IN THE RITZ VECTOR    HHL01550
C                  COMPUTATIONS                                         HHL01560
C     NTVCON = 0:  PROGRAM TERMINATES IF THE TVEC ARRAY IS NOT          HHL01570
C                  LARGE ENOUGH TO HOLD ALL THE T-EIGENVECTORS REQUIRED.HHL01580
C     SVTVEC = 0:  THE T-EIGENVECTORS ARE NOT WRITTEN TO FILE 11        HHL01590
C                  UNLESS TVSTOP = 1                                    HHL01600
C     SVTVEC = 1:  WRITE THE T-EIGENVECTORS TO FILE 11.                 HHL01610
C     TVSTOP = 1:  PROGRAM TERMINATES AFTER COMPUTING THE               HHL01620
C                  T-EIGENVECTORS                                       HHL01630
C     LVCONT = 0:  PROGRAM TERMINATES IF THE NUMBER OF T-EIGENVECTORS   HHL01640
C                  COMPUTED IS NOT EQUAL TO THE NUMBER OF RITZ          HHL01650
C                  VECTORS REQUESTED.                                   HHL01660
C     ERCONT = 0:  MEANS FOR ANY GIVEN EIGENVALUE, A RITZ VECTOR        HHL01670
C                  WILL NOT BE COMPUTED FOR THAT EIGENVALUE UNLESS      HHL01680
C                  A T-EIGENVECTOR HAS BEEN IDENTIFIED WITH A LAST      HHL01690
C                  COMPONENT WHICH SATISFIES THE SPECIFIED              HHL01700
C                  CONVERGENCE CRITERION.                               HHL01710
C     ERCONT = 1:  MEANS FOR ANY GIVEN EIGENVALUE, A RITZ VECTOR        HHL01720
C                  WILL BE COMPUTED.  IF A T-EIGENVECTOR CANNOT         HHL01730
C                  BE IDENTIFIED WHICH SATISFIES THE LAST               HHL01740
C                  COMPONENT CRITERION, THEN THE PROGRAM WILL           HHL01750
C                  USE THE T-VECTOR THAT CAME CLOSEST TO                HHL01760
C                  SATISFYING THE CRITERION                             HHL01770
C     IWRITE = 1:  EXTENDED OUTPUT OF INTERMEDIATE COMPUTATIONS         HHL01780
C                  IS WRITTEN TO FILE 6                                 HHL01790
C     IREAD = 0:   ALPHA/BETA FILE IS REGENERATED                       HHL01800
C     IREAD = 1:   ALPHA/BETA FILE USED IN EIGENVALUE COMPUTATIONS      HHL01810
C                  IS READ IN AND EXTENDED IF NECESSARY.  IN BOTH       HHL01820
C                  CASES IREAD = 0 OR 1, THE LANCZOS VECTORS ARE        HHL01830
C                  ALWAYS REGENERATED FOR THE RITZ VECTOR               HHL01840
C                  COMPUTATIONS                                         HHL01850
C                                                                       HHL01860
      READ(5,20) EXPLAN                                                 HHL01870
      READ(5,*) MBOUND,NTVCON,SVTVEC,IREAD                              HHL01880
C                                                                       HHL01890
      READ(5,20) EXPLAN                                                 HHL01900
      READ(5,*) TVSTOP,LVCONT,ERCONT,IWRITE                             HHL01910
      IF (TVSTOP.EQ.1) SVTVEC = 1                                       HHL01920
C                                                                       HHL01930
C     READ IN SEED FOR GENERATING RANDOM STARTING VECTOR FOR THE        HHL01940
C     INVERSE ITERATION ON THE T-MATRICES.                              HHL01950
C                                                                       HHL01960
      READ(5,20) EXPLAN                                                 HHL01970
      READ(5,*) RHSEED                                                  HHL01980
C                                                                       HHL01990
C     READ IN MATNO = MATRIX/RUN IDENTIFICATION NUMBER AND              HHL02000
C     N = ORDER OF A-MATRIX                                             HHL02010
C                                                                       HHL02020
      READ(5,20) EXPLAN                                                 HHL02030
      READ(5,*) MATNO,N                                                 HHL02040
C     IF MATNO < 0, THEN MATRIX SUPPLIED IS REAL AND USER WANTS TO      HHL02050
C     CHECK ON THE T-MULTIPLICITY OF THE EIGENVALUES OF GIVEN MATRIX    HHL02060
      IF(MATNO.GT.0) GO TO 30                                           HHL02070
      ISREAL = 1                                                        HHL02080
```

```
      MATNO = - MATNO                                                   HHL02090
   30 CONTINUE                                                          HHL02100
C                                                                       HHL02110
C-----------------------------------------------------------------------HHL02120
C     INITIALIZE THE ARRAYS FOR THE USER-SPECIFIED MATRIX               HHL02130
C     AND PASS THE STORAGE LOCATIONS OF THESE ARRAYS TO THE             HHL02140
C     MATRIX-VECTOR MULTIPLY SUBROUTINE CMATV.                          HHL02150
C                                                                       HHL02160
      CALL USPEC(N,MATNO)                                               HHL02170
C                                                                       HHL02180
C-----------------------------------------------------------------------HHL02190
C     MASK UNDERFLOW AND OVERFLOW                                       HHL02200
      CALL MASK                                                         HHL02210
C                                                                       HHL02220
C-----------------------------------------------------------------------HHL02230
C                                                                       HHL02240
C     WRITE RUN PARAMETERS OUT TO FILE 6                                HHL02250
C                                                                       HHL02260
      WRITE(6,40) MATNO,N                                               HHL02270
   40 FORMAT(/' MATRIX IDENTIFICATION NO. = ',I10,' ORDER OF A = ',I5)  HHL02280
C                                                                       HHL02290
      WRITE(6,50) MBOUND,NTVCON,SVTVEC,IREAD                            HHL02300
   50 FORMAT(/3X,'MBOUND',3X,'NTVCON',3X,'SVTVEC',3X,'IREAD'/3I9,I8)    HHL02310
C                                                                       HHL02320
      WRITE(6,60) TVSTOP,LVCONT,ERCONT,IWRITE                           HHL02330
   60 FORMAT(/3X,'TVSTOP',3X,'LVCONT',3X,'ERCONT',3X,'IWRITE'/4I9)      HHL02340
C                                                                       HHL02350
      WRITE(6,70) MDIMTV,MDIMRV,MBETA                                   HHL02360
   70 FORMAT(/3X,'MDIMTV',3X,'MDIMRV',3X,'MBETA'/2I9,I8)                HHL02370
C                                                                       HHL02380
      WRITE(6,80) RELTOL,RHSEED                                         HHL02390
   80 FORMAT(/7X,'RELTOL',3X,'RHSEED'/E13.4,I9)                         HHL02400
C                                                                       HHL02410
C                                                                       HHL02420
C     FROM FILE 3 READ IN THE NUMBER OF EIGENVALUES (NGOOD) FOR WHICH   HHL02430
C     EIGENVECTORS ARE REQUESTED, THE ORDER (MEV) OF THE LANCZOS        HHL02440
C     TRIDIAGONAL MATRIX USED IN COMPUTING THESE EIGENVALUES, THE       HHL02450
C     ORDER (NOLD) OF THE USER-SPECIFIED MATRIX USED IN THE EIGENVALUE  HHL02460
C     COMPUTATIONS, THE SEED (SVSEED) USED FOR GENERATING THE STARTING  HHL02470
C     VECTOR THAT WAS USED IN THOSE LANCZOS EIGENVALUE COMPUTATIONS,    HHL02480
C     AND THE MATRIX/RUN IDENTIFICATION NUMBER (MATOLD) USED IN THOSE   HHL02490
C     COMPUTATIONS.  ALSO READ IN THE NUMBER (NDIS) OF DISTINCT         HHL02500
C     EIGENVALUES OF T(1,MEV) THAT WERE COMPUTED BUT THIS VALUE IS      HHL02510
C     NOT USED IN THE EIGENVECTOR COMPUTATIONS.                         HHL02520
C                                                                       HHL02530
      READ(3,90) NGOOD,NDIS,MEV,NOLD,SVSEED,MATOLD                      HHL02540
   90 FORMAT(4I6,I12,I8)                                                HHL02550
C                                                                       HHL02560
C     READ IN THE T-MULTIPLICITY TOLERANCE USED IN THE BISEC SUBROUTINE HHL02570
C     DURING THE COMPUTATION OF THE GIVEN EIGENVALUES.                  HHL02580
C     ALSO READ IN THE FLAG IB.  IF IB < 0, THEN SOME BETA(I) IN THE    HHL02590
C     T-MATRIX FILE PROVIDED ON FILE 2 FAILED THE ORTHOGONALITY         HHL02600
C     TEST IN THE TNORM SUBROUTINE.  USER SHOULD NOTE THAT THIS         HHL02610
C     VECTOR PROGRAM PROCEEDS INDEPENDENTLY OF THE SIZE OF THE BETA.    HHL02620
C                                                                       HHL02630
```

```
      READ(3,100) MULTOL,IB,BTOL                                        HHL02640
  100 FORMAT(E20.12,I6,E13.4)                                           HHL02650
C                                                                       HHL02660
      TEMP = DFLOAT(MEV+1000)                                           HHL02670
      TTOL = MULTOL/TEMP                                                HHL02680
      WRITE(6,110) MULTOL,TTOL                                          HHL02690
  110 FORMAT(/' T-MULTIPLICITY TOLERANCE USED IN THE EIGENVALUE COMPUTATHHL02700
     1IONS WAS',E13.4/' SCALED MACHINE EPSILON IS',E13.4)               HHL02710
C                                                                       HHL02720
C     CONTINUE WRITE TO FILE 6 OF THE PARAMETERS FOR THIS RUN           HHL02730
C                                                                       HHL02740
      WRITE(6,120)NGOOD,NDIS,MEV,NOLD,MATOLD,SVSEED,MULTOL,IB,BTOL      HHL02750
  120 FORMAT(/' EIGENVALUES SUPPLIED ARE READ IN FROM FILE 3'/' FILE 3  HHL02760
     1HEADER IS'/4X,'NG',2X,'NDIS',3X,'MEV',2X,'NOLD',2X,'MATOLD',4X,   HHL02770
     1'SVSEED',6X,'MULTOL',6X,'IB',9X,'BTOL'/4I6,I8,I10,E12.3,I8,E13.4/)HHL02780
C                                                                       HHL02790
C     IS THE ARRAY RITVEC LONG ENOUGH TO HOLD ALL OF THE DESIRED        HHL02800
C     RITZ VECTORS (APPROXIMATE EIGENVECTORS)?                          HHL02810
      NMAX = NGOOD*N                                                    HHL02820
      IF(MBOUND.NE.0) GO TO 130                                         HHL02830
      IF(TVSTOP.NE.1.AND.NMAX.GT.MDIMRV) GO TO 1390                     HHL02840
C                                                                       HHL02850
C     CHECK THAT THE ORDER N AND THE MATRIX IDENTIFICATION NUMBER       HHL02860
C     MATNO SPECIFIED BY THE USER AGREE WITH THOSE READ IN FROM         HHL02870
C     FILE 3.                                                           HHL02880
  130 ITEMP = (NOLD-N)**2+(MATOLD-MATNO)**2                             HHL02890
      IF (ITEMP.NE.0) GO TO 1410                                        HHL02900
C                                                                       HHL02910
C                                                                       HHL02920
C     THE LANCZOS EIGENVECTOR COMPUTATIONS ASSUME THAT EACH             HHL02930
C     EIGENVALUE THAT IS BEING CONSIDERED HAS CONVERGED AS AN           HHL02940
C     EIGENVALUE OF THE LANCZOS TRIDIAGONAL MATRICES.                   HHL02950
C                                                                       HHL02960
C     READ IN FROM FILE 3, THE T-MULTIPLICITIES OF THE EIGENVALUES      HHL02970
C     WHOSE EIGENVECTORS ARE TO BE COMPUTED, THE VALUES OF THESE        HHL02980
C     EIGENVALUES AND THEIR MINIMAL GAPS AS EIGENVALUES OF THE          HHL02990
C     USER-SPECIFIED MATRIX AND AS EIGENVALUES OF THE T-MATRIX.         HHL03000
C                                                                       HHL03010
      READ(3,20) EXPLAN                                                 HHL03020
      READ(3,140) (MP(J),GOODEV(J),TMINGP(J),AMINGP(J), J=1,NGOOD)      HHL03030
  140 FORMAT(6X,I6,E25.16,2E14.3)                                       HHL03040
C                                                                       HHL03050
      WRITE(6,150) (J,GOODEV(J),MP(J),TMINGP(J),AMINGP(J), J=1,NGOOD)   HHL03060
  150 FORMAT(/' EIGENVALUES READ IN, T-MULTIPLICITIES, T-GAPS AND A-GAPSHHL03070
     1 '/4X,' J ',5X,'GOOD EIGENVALUE',5X,'MULT',4X,'  TMINGAP ',4X,    HHL03080
     1'  AMINGAP '/(I6,E25.16,I4,2E15.4))                               HHL03090
C                                                                       HHL03100
C     READ IN ERROR ESTIMATES                                           HHL03110
      WRITE(6,180) MEV,SVSEED                                           HHL03120
C     CHECK WHETHER OR NOT THERE ARE ANY ISOLATED T-EIGENVALUES IN      HHL03130
C     THE EIGENVALUES PROVIDED                                          HHL03140
      DO 160 J=1,NGOOD                                                  HHL03150
      IF(MP(J).EQ.1) GO TO 170                                          HHL03160
  160 CONTINUE                                                          HHL03170
      GO TO 200                                                         HHL03180
```

```
  170 READ(4,20) EXPLAN                                                 HHL03190
      READ(4,20) EXPLAN                                                 HHL03200
      READ(4,20) EXPLAN                                                 HHL03210
  180 FORMAT(/' THESE EIGENVALUES WERE COMPUTED USING A T-MATRIX OF     HHL03220
     1ORDER ',I5/' AND SEED FOR RANDOM NUMBER GENERATOR =',I12)         HHL03230
      READ(4,190) NISO                                                  HHL03240
  190 FORMAT(18X,I6)                                                    HHL03250
      READ(4,20) EXPLAN                                                 HHL03260
      READ(4,20) EXPLAN                                                 HHL03270
      READ(4,20) EXPLAN                                                 HHL03280
  200 DO 230 J=1,NGOOD                                                  HHL03290
      ERR(J) = 0.D0                                                     HHL03300
      IF(MP(J).NE.1) GO TO 230                                          HHL03310
      READ(4,210) EVAL, ERR(J)                                          HHL03320
  210 FORMAT(10X,E25.16,E14.3)                                          HHL03330
      IF(DABS(EVAL - GOODEV(J)).LT.1.D-10) GO TO 230                    HHL03340
      WRITE(6,220) EVAL,GOODEV(J)                                       HHL03350
  220 FORMAT(' PROBLEM WITH READ IN OF ERROR ESTIMATES'/' EIGENVALUE REAHHL03360
     1D IN',E20.12,' DOES NOT MATCH GOODEV(J) ='/E20.12)                HHL03370
      GO TO 1630                                                        HHL03380
C                                                                       HHL03390
  230 CONTINUE                                                          HHL03400
C                                                                       HHL03410
      WRITE(6,240) (J,GOODEV(J),ERR(J), J=1,NGOOD)                      HHL03420
  240 FORMAT(' ERROR ESTIMATES ='/4X,' J',5X,'EIGENVALUE',10X,'ESTIMATE'HHL03430
     1 /(I6,E20.12,E14.3))                                              HHL03440
C                                                                       HHL03450
      IF(IREAD.EQ.0) GO TO 340                                          HHL03460
C                                                                       HHL03470
C     READ IN THE SIZE OF THE T-MATRIX PROVIDED ON FILE 2.  READ IN     HHL03480
C     THE ORDER OF THE USER-SPECIFIED MATRIX , THE SEED FOR THE         HHL03490
C     RANDOM NUMBER GENERATOR, AND THE MATRIX/TEST IDENTIFICATION       HHL03500
C     NUMBER THAT WERE USED IN THE LANCZOS EIGENVALUE COMPUTATIONS.     HHL03510
C     IF FLAG IREAD = 0, REGENERATE HISTORY.  HISTORY MUST BE           HHL03520
C     STORED IN HEXADECIMAL FORMAT TO AVOID ERRORS INCURRED IN          HHL03530
C     INPUT/OUTPUT CONVERSIONS.                                         HHL03540
C                                                                       HHL03550
      READ(2,250) KMAX,NOLD,SVSOLD,MATOLD                               HHL03560
  250 FORMAT(2I6,I12,I8)                                                HHL03570
C                                                                       HHL03580
      WRITE(6,260) KMAX,NOLD,SVSOLD,MATOLD                              HHL03590
  260 FORMAT(/' READ IN THE T-MATRICES STORED ON FILE 2'/' FILE 2 HEADERHHL03600
     1 IS'/2X,'KMAX',2X,'NOLD',6X,'SVSOLD',2X,'MATOLD'/2I6,I12,I8/)     HHL03610
C                                                                       HHL03620
C     CHECK THAT THE ORDER, THE MATRIX/TEST IDENTIFICATION NUMBER       HHL03630
C     AND THE SEED FOR THE RANDOM NUMBER GENERATOR USED IN THE          HHL03640
C     LANCZOS COMPUTATIONS THAT GENERATED THE HISTORY FILE              HHL03650
C     BEING USED AGREE WITH WHAT THE USER HAS SPECIFIED.                HHL03660
      IF (NOLD.NE.N.OR.MATOLD.NE.MATNO.OR.SVSOLD.NE.SVSEED) GO TO 1430  HHL03670
C                                                                       HHL03680
      KMAX1 = KMAX + 1                                                  HHL03690
C                                                                       HHL03700
C     READ IN THE T-MATRICES FROM FILE 2.  THESE ARE USED TO GENERATE   HHL03710
C     THE T-EIGENVECTORS THAT WILL BE USED IN THE RITZ VECTOR           HHL03720
C     COMPUTATIONS.  ALPHA/BETA HISTORY MUST BE STORED IN               HHL03730
```

```
C     MACHINE FORMAT, ((4Z20) FOR IBM/3081).
C
      READ(2,270) (ALPHA(J), J=1,KMAX)
      READ(2,270) (BETA(J), J=1,KMAX1)
  270 FORMAT(4Z20)
C
      READ(2,270) (V1(J), J=1,N)
      READ(2,270) (V2(J), J=1,N)
C
C     ENLARGE KMAX IF THE SIZE AT WHICH THE EIGENVALUE
C     COMPUTATIONS WERE PERFORMED IS ESSENTIALLY KMAX AND
C     THERE IS AT LEAST ONE EIGENVALUE THAT IS T-SIMPLE AND
C     T-ISOLATED, IN THE SENSE THAT IF ITS NEAREST T-NEIGHBOR IS TOO
C     CLOSE THAT NEIGHBOR IS A 'GOOD' T-EIGENVALUE.
      DO 280 J = 1,NGOOD
      IF(MP(J).EQ.1) GO TO 300
  280 CONTINUE
      WRITE(6,290)
  290 FORMAT(/' ALL EIGENVALUES USED ARE T-MULTIPLE OR CLOSE TO SPURIOUS
     1 T-EIGENVALUES'/' SO DO NOT CHANGE KMAX')
      IF(KMAX.LT.MEV) GO TO 1450
      GO TO 320
C
  300 KMAXN= 11*MEV/8 + 12
      IF(MBETA.LE.KMAXN) GO TO 1610
      IF(KMAX.GE.KMAXN )  GO TO 320
      WRITE(6,310) KMAX, KMAXN
  310 FORMAT(' ENLARGE KMAX FROM ',I6,' TO ',I6)
      MOLD1 = KMAX + 1
      KMAX = KMAXN
      GO TO 390
C
  320 WRITE(6,330) KMAX
  330 FORMAT(/' T-MATRICES HAVE BEEN READ IN FROM FILE 2'/' THE LARGEST
     1SIZE T-MATRIX ALLOWED IS',I6/)
C
      IF(IREAD.EQ.1) GO TO 410
C
C     REGENERATE THE ALPHA AND BETA
C
  340 MOLD1 = 1
C
      DO 350 J = 1,NGOOD
      IF(MP(J).EQ.1) GO TO 370
  350 CONTINUE
      KMAX = MEV + 12
      WRITE(6,360) KMAX
  360 FORMAT(/' ALL EIGENVALUES FOR WHICH EIGENVECTORS ARE TO BE COMPUTE
     1D ARE EITHER T-MULTIPLE OR CLOSE TO'/' A SPURIOUS EIGENVALUE. THER
     1EFORE SET KMAX = MEV + 12 = ',I7)
      GO TO 390
C
  370 KMAXN = 11*MEV/8 + 12
      IF(MBETA.LE.KMAXN) GO TO 1610
      WRITE(6,380) KMAXN
```

```
  380 FORMAT(' SET KMAX EQUAL TO ',I6)
      KMAX = KMAXN
C
  390 WRITE(6,400) MOLD1,KMAX
  400 FORMAT(/' LANCZS SUBROUTINE GENERATES ALPHA(J), BETA(J+1), J =',
     1 I6,' TO ', I6/)
C
C-----------------------------------------------------------------------
C
      CALL LANCZS(CMATV,V1,V2,VS,ALPHA,BETA,GR,GC,G,KMAX,MOLD1,N,SVSEED)
C
C-----------------------------------------------------------------------
C
  410 CONTINUE
C
C     THE SUBROUTINE STURMI DETERMINES THE SMALLEST SIZE T-MATRIX FOR
C     WHICH THE EIGENVALUE IN QUESTION IS AN EIGENVALUE (TO WITHIN A
C     GIVEN TOLERANCE) AND IF POSSIBLE THE SMALLEST SIZE T-MATRIX
C     FOR WHICH IT IS A DOUBLE EIGENVALUE (TO WITHIN THE SAME
C     TOLERANCE).  THE SIZE T-MATRIX USED IN THE EIGENVECTOR
C     COMPUTATIONS IS THEN DETERMINED BY LOOPING ON THE SIZES OF THE
C     T-EIGENVECTORS, USING THE INFORMATION FROM STURMI TO OBTAIN
C     STARTING GUESSES AT THE T-SIZES.
C
C
      STUTOL = SCALE0*MULTOL
      IF(IWRITE.EQ.1) WRITE(6,420)
  420 FORMAT(' FROM STURMI')
      DO 460 J = 1,NGOOD
      EVAL = GOODEV(J)
C     COMPUTE THE TOLERANCES USED BY STURMI TO DETERMINE AN INTERVAL
C     CONTAINING THE EIGENVALUE EVAL.
      TEMP = DABS(EVAL)*RELTOL
      TOLN = DMAX1(TEMP,STUTOL)
C
C-----------------------------------------------------------------------
C
      CALL STURMI(ALPHA,BETA,EVAL,TOLN,EPSM,KMAX,MK1,MK2,IC,IWRITE)
C
C-----------------------------------------------------------------------
C
C     STORE THE COMPUTED ORDERS OF T-MATRICES FOR LATER PRINTOUT
      M1(J) = MK1
      M2(J) = MK2
      ML(J) = (MK1 + 3*MK2)/4
      IF(MK2.EQ.KMAX)  ML(J) = KMAX
C
      IF(IC.GT.0) GO TO 440
C     IC = 0 MEANS THERE WAS NO T-EIGENVALUE IN THE DESIGNATED INTERVAL
C     BY T-SIZE KMAX.  THIS MEANS THAT THE T-EIGENVALUE PROVIDED HAS
C     NOT YET CONVERGED AS AN EIGENVALUE OF THE TRIDIAGONAL MATRICES
C     SO PROGRAM SHOULD NOT COMPUTE ITS EIGENVECTOR.
      WRITE(6,430) J,GOODEV(J),MK1,MK2
  430 FORMAT(I6,'TH EIGENVALUE',E20.12,' HAS NOT CONVERGED '/
     1' SO DO NOT COMPUTE ANY T-EIGENVECTOR OR RITZ VECTOR FOR IT'
```

```
     1/' MK1 AND MK2 FOR THIS EIGENVALUE WERE',2I6)
      MP(J) = MPMIN
      MA(J) = -2*KMAX
      GO TO 460
C     COMPUTE AN APPROPRIATE SIZE T-MATRIX FOR THE GIVEN EIGENVALUE.
  440 IF(M2(J).EQ.KMAX) GO TO 450
C     M1 AND M2 WERE BOTH DETERMINED
      MA(J) = (3*M1(J) + M2(J))/4  + 1
      GO TO 460
C     M2 NOT DETERMINED
  450 MA(J) = 5*M1(J)/4  + 1
C
  460 CONTINUE
C
      IF (IWRITE.EQ.1) WRITE(6,470) (MA(JJ), JJ=1,NGOOD)
  470 FORMAT(/' 1ST GUESS AT APPROPRIATE SIZE T-MATRICES'/
     1 ' ACTUAL VALUES WILL PROBABLY BE 1/4 AGAIN AS MUCH'/(13I6))
C
C     PRINT OUT TO FILE 10 1ST GUESSES AT SIZES OF T-MATRICES TO
C     BE USED IN THE EIGENVECTOR COMPUTATIONS.
C     ACTUAL SIZES MAY BE 1/4 OR MORE LARGER THAN THESE SIZES.
      WRITE(10,480) N,KMAX
  480 FORMAT(2I8,' = ORDER OF USER MATRIX AND MAX ORDER OF T(1,MEV)')
C
      WRITE(10,490)
  490 FORMAT(/' 1ST GUESS AT APPROPRIATE SIZE T-MATRICES'/
     1 ' ACTUAL VALUES WILL PROBABLY BE 1/4 AGAIN AS MUCH'/)
C
      WRITE(10,500)
  500 FORMAT(4X,'J',7X,'GOODEV(J)',4X,'M1(J)',1X,'M2(J)',1X,'MA(J)')
C
      WRITE(10,510) (J,GOODEV(J),M1(J),M2(J), MA(J), J=1,NGOOD)
  510 FORMAT(I5,E19.12,3I6)
C
      IF(MBOUND.EQ.1) WRITE(10,520)
  520 FORMAT(/' GOODEV(J) IS A GOOD EIGENVALUE OF T(1,MEV)'/
     1 ' M1 = SMALLEST VALUE OF M SUCH THAT T(1,M) HAS AT LEAST'/
     1 '      ONE EIGENVALUE IN THE INTERVAL (EV-TOLN,EV+TOLN)'/
     1 '      TOLN(J) = DMAX1(GOODEV(J)*RELTOL, SCALE0*MULTOL)'/
     1 ' M2 = SMALLEST M (IF ANY) SUCH THAT IN THE ABOVE INTERVAL'/
     1 '      T(1,M) HAS AT LEAST TWO EIGENVALUES '/
     1 ' INITIAL VALUE OF MA(J) IS CHOSEN HEURISTICALLY'/
     1 ' PROGRAM LOOPS ON SIZE OF T-MATRIX TO GET BETTER SIZE'/
     1 ' END OF SIZES OF T-MATRICES FILE 10'///)
C
C
C     TERMINATE AFTER COMPUTING 1ST GUESSES AT  SIZES OF THE
C     T-MATRICES REQUIRED FOR THE GIVEN EIGENVALUES?
      IF(MBOUND.EQ.1) GO TO 1470
C
C
C     IS THERE ROOM FOR ALL OF THE REQUESTED T-EIGENVECTORS?
      MTOL = 0
      DO 530 J = 1,NGOOD
      IF(MP(J).EQ.MPMIN) GO TO 530
```

```
      MTOL = MTOL + IABS(MA(J))
  530 CONTINUE
      MTOL = (5*MTOL)/4
      IF(MTOL.GT.MDIMTV.AND.NTVCON.EQ.0) GO TO 1490
C
C-----------------------------------------------------------------------
C     GENERATE A RANDOM VECTOR TO BE USED REPEATEDLY BY
C     SUBROUTINE INVERM
C
      IIL = RHSEED
      CALL GENRAN(IIL,G,KMAX)
C
C-----------------------------------------------------------------------
C
C     LOOP ON GIVEN EIGENVALUES TO COMPUTE THE CORRESPONDING
C     T-EIGENVECTOR.
C
      MTOL = 0
      NTVEC = 0
      ILBIS = 0
      DO 720 J = 1,NGOOD
      ICOUNT = 0
      ERRMIN = 10.D0
      MABEST = MPMIN
      IF(MP(J).EQ.MPMIN) GO TO 720
      TFLAG = 0
      EVAL = GOODEV(J)
      TEMP = RELTOL* DABS(EVAL)
      UB = EVAL + DMAX1(STUTOL,TEMP)
      LB = EVAL - DMAX1(STUTOL,TEMP)
  540 KMAXU = IABS(MA(J))
C
C     SELECT A SUITABLE INCREMENT FOR THE ORDERS OF THE T-MATRICES
C     TO BE CONSIDERED IN DETERMINING APPROPRIATE SIZES FOR THE RITZ
C     VECTOR COMPUTATIONS.
      IF(ICOUNT.GT.0) GO TO 560
C     SELECT IDELTA(J) BASED UPON THE T-MULTIPLICITY OBTAINED
      IF(M2(J).EQ.KMAX) GO TO 550
C     M2 DETERMINED
      IDELTA(J) = ((3*M1(J) + 5*M2(J))/8  +  1 - IABS(MA(J)))/10 + 1
      GO TO 560
C     M2 NOT DETERMINED
  550 MAMAX = MIN0((11*MEV)/8 + 12, (13*M1(J))/8 + 1)
      IDELTA(J) = (MAMAX - IABS(MA(J)))/10  +  1
  560 ICOUNT = ICOUNT + 1
C
C-----------------------------------------------------------------------
C     TO MIMIMIZE THE EFFECT OF THE ONE-SIDED ACCEPTANCE TEST FOR
C     T-EIGENVALUES IN THE BISEC SUBROUTINE, RECOMPUTE THE GIVEN
C     EIGENVALUE AT THE SPECIFIED KMAXU
C
      CALL LBISEC(ALPHA,BETA,EPSM,EVAL,EVALN,LB,UB,TTOL,KMAXU,NEVT)
C
C-----------------------------------------------------------------------
C
```

```
C     CHECK WHETHER OR NOT GIVEN T-MATRIX HAS AN EIGENVALUE IN THE      HHL05940
C     SPECIFIED INTERVAL AND IF SO WHAT ITS T-MULTIPLICITY IS.          HHL05950
C                                                                       HHL05960
      IF(NEVT.EQ.1) GO TO 600                                           HHL05970
      IF(NEVT.NE.0) GO TO 580                                           HHL05980
      ILBIS = 1                                                         HHL05990
      WRITE(6,570) EVAL,KMAXU                                           HHL06000
  570 FORMAT(/' PROBLEM ENCOUNTERED IN RECOMPUTATION OF USER-SUPPLIED EIHHL06010
     1GENVALUE',E20.12/' THE SIZE T-MATRIX SPECIFIED',I6,' DOES NOT     HHL06020
     1HAVE AN EIGENVALUE IN THE INTERVAL SPECIFIED'/' THEREFORE NO EIGENHHL06030
     1VECTOR WILL BE COMPUTED FOR THIS PARTICULAR EIGENVALUE'/)         HHL06040
      GO TO 620                                                         HHL06050
C                                                                       HHL06060
  580 IF(NEVT.GT.1)  WRITE(6,590) EVAL,KMAXU                            HHL06070
  590 FORMAT(/' PROBLEM ENCOUNTERED IN RECOMPUTATION OF USER-SUPPLIED   HHL06080
     1EIGENVALUE',E20.12/' FOR THE SIZE T-MATRIX SPECIFIED =',I6,' THE  HHL06090
     1GIVEN EIGENVALUE IS T-MULTIPLE IN THE INTERVAL SPECIFIED'/' SOMETHHHL06100
     1ING IS WRONG, THEREFORE NO EIGENVECTOR WILL BE COMPUTED FOR THIS EHHL06110
     1IGENVALUE'/)                                                      HHL06120
C                                                                       HHL06130
      MP(J) = MPMIN                                                     HHL06140
      MA(J) = -2*KMAX                                                   HHL06150
      GO TO 720                                                         HHL06160
C                                                                       HHL06170
  600 CONTINUE                                                          HHL06180
      ILBIS = 0                                                         HHL06190
C                                                                       HHL06200
      EVNEW(J) = EVALN                                                  HHL06210
      EVAL = EVALN                                                      HHL06220
      MTOL = MTOL+KMAXU                                                 HHL06230
C                                                                       HHL06240
C     IS THERE ROOM IN TVEC ARRAY FOR THE NEXT T-EIGENVECTOR?           HHL06250
C     IF NOT, SKIP TO RITZ VECTOR COMPUTATIONS.                         HHL06260
      IF (MTOL.GT.MDIMTV) GO TO 730                                     HHL06270
C                                                                       HHL06280
      IT = 3                                                            HHL06290
      KINT = MTOL - KMAXU +1                                            HHL06300
C                                                                       HHL06310
C     RECORD THE BEGINNING AND END OF THE T-EIGENVECTOR BEING COMPUTED  HHL06320
      MINT(J) = KINT                                                    HHL06330
      MFIN(J) = MTOL                                                    HHL06340
C                                                                       HHL06350
C-----------------------------------------------------------------------HHL06360
C     SUBROUTINE INVERM DOES INVERSE ITERATION, I.E. SOLVES             HHL06370
C     (T(1,KMAXU) - EVAL)*U = RHS  FOR EACH EIGENVALUE TO OBTAIN THE    HHL06380
C     DESIRED T-EIGENVECTOR.                                            HHL06390
C                                                                       HHL06400
      IF(IWRITE.EQ.1)  WRITE(6,610) J                                   HHL06410
  610 FORMAT(/I6,'TH EIGENVALUE')                                       HHL06420
C                                                                       HHL06430
      CALL INVERM(ALPHA,BETA,V1,TVEC(KINT),EVAL,ERROR,TERROR,EPSM,      HHL06440
     1 G,KMAXU,IT,IWRITE)                                               HHL06450
C                                                                       HHL06460
C-----------------------------------------------------------------------HHL06470
C                                                                       HHL06480
```

```
      TERR(J) = TERROR                                                  HHL06490
      TLAST(J) = ERROR                                                  HHL06500
      KMAXU1 = KMAXU + 1                                                HHL06510
      TBETA(J) = BETA(KMAXU1)*ERROR                                     HHL06520
C                                                                       HHL06530
C     AFTER COMPUTING EACH OF THE T-EIGENVECTORS,                       HHL06540
C     CHECK THE SIZE OF THE ERROR ESTIMATE, ERROR.                      HHL06550
C     IF THIS ESTIMATE IS NOT AS SMALL AS DESIRED AND                   HHL06560
C     |MA(J)| < ML(J), ATTEMPT TO INCREASE THE SIZE OF |MA(J)|          HHL06570
C     AND REPEAT THE T-EIGENVECTOR COMPUTATIONS.                        HHL06580
C                                                                       HHL06590
      IF(ERROR.LT.ERTOL.OR.TFLAG.EQ.1)  GO TO 710                       HHL06600
C                                                                       HHL06610
      IF(ERROR.GE.ERRMIN) GO TO 620                                     HHL06620
C     LAST COMPONENT IS LESS THAN MINIMAL TO DATE                       HHL06630
      ERRMIN = ERROR                                                    HHL06640
      MABEST = MA(J)                                                    HHL06650
  620 CONTINUE                                                          HHL06660
C                                                                       HHL06670
      IF(MA(J).GT.0)  ITEST = MA(J) + IDELTA(J)                         HHL06680
      IF(MA(J).LT.0)  ITEST = -(IABS(MA(J)) + IDELTA(J))                HHL06690
      IF(IABS(ITEST).LE.ML(J).AND.ICOUNT.LE.10) GO TO 640               HHL06700
C     NEW MA(J) IS GREATER THAN MAXIMUM ALLOWED.                        HHL06710
      IF(ERCONT.EQ.0.OR.MABEST.EQ.MPMIN) GO TO 660                      HHL06720
      TFLAG = 1                                                         HHL06730
      MA(J) = MABEST                                                    HHL06740
      IF(ILBIS.EQ.0)  MTOL = MTOL - KMAXU                               HHL06750
      WRITE(6,630) MA(J)                                                HHL06760
  630 FORMAT(' 10 ORDERS WERE CONSIDERED.  NONE SATISFIED THE ERROR TESTHHL06770
     1'/' THEREFORE USE THE BEST ORDER OBTAINED FOR THE EIGENVECTORS'   HHL06780
     1,I6)                                                              HHL06790
      GO TO 540                                                         HHL06800
C                                                                       HHL06810
  640 MA(J) = ITEST                                                     HHL06820
C                                                                       HHL06830
      MT = IABS(MA(J))                                                  HHL06840
      IF(IWRITE.EQ.1) WRITE(6,650)  MT                                  HHL06850
  650 FORMAT(/' CHANGE SIZE OF T-MATRIX TO ',I6,' RECOMPUTE T-EIGENVECTOHHL06860
     1R')                                                               HHL06870
C                                                                       HHL06880
      IF(ILBIS.EQ.0)  MTOL = MTOL - KMAXU                               HHL06890
C                                                                       HHL06900
      GO TO 540                                                         HHL06910
C                                                                       HHL06920
C       APPROPRIATE SIZE T-MATRIX WAS NOT OBTAINED                      HHL06930
  660 CONTINUE                                                          HHL06940
      WRITE(10,670) J,EVAL,MP(J)                                        HHL06950
  670 FORMAT(/' ON 10 INCREMENTS NOT ABLE TO IDENTIFY APPROPRIATE SIZE  HHL06960
     1T-MATRIX FOR'/                                                    HHL06970
     1' EIGENVALUE(',I4,') = ',E20.12,' T-MULTIPLICITY =',I4/)          HHL06980
      IF(M2(J).EQ.KMAX) WRITE(10,680)                                   HHL06990
      IF(M2(J).LT.KMAX) WRITE(10,690)                                   HHL07000
  680 FORMAT(/' ORDERS TESTED RANGED FROM 5*M1(J)/4 TO APPROXIMATELY'/  HHL07010
     1'   MIN(11*MEV/8, 13*M1(J)/8)'/)                                  HHL07020
  690 FORMAT(/' ORDERS TESTED RANGED FROM (3*M1(J)+M2(J)/4 TO APPROXIMATHHL07030
```

```
     1ELY'/'  (3*M1(J) + 5*M2(J))/8'/)
      WRITE(10,700)
  700 FORMAT(' ALLOWING LARGER ORDERS FOR THE T-MATRICES MAY RESULT IN
     1 SUCCESS'/' BUT PROBABLY WILL NOT.  PROBLEM IS PROBABLY DUE TO'
     1 /' LACK OF CONVERGENCE OF GIVEN EIGENVALUE, CHECK THE ERROR ESTIM
     1ATE')
      MP(J) = MPMIN
      IF(ILBIS.EQ.0) MTOL = MTOL - KMAXU
      GO TO 720
  710 NTVEC = NTVEC + 1
C
  720 CONTINUE
      NGOODC = NGOOD
      GO TO 750
C
C     COME HERE IF THERE IS NOT ENOUGH ROOM FOR ALL OF T-EIGENVECTORS
  730 NGOODC = J-1
      WRITE(6,740) J,MTOL,MDIMTV
  740 FORMAT(/' NOT ENOUGH ROOM IN TVEC ARRAY FOR ',I4,'TH T-EIGENVECTOR
     1'/'  TVEC DIMENSION REQUESTED = ',I6,' BUT TVEC HAS DIMENSION ',I6
     1/)
      IF(NGOODC.EQ.0)  GO TO 1510
      MTOL = MTOL-KMAXU
C
  750 CONTINUE
C
C     THE LOOP ON T-EIGENVECTOR COMPUTATIONS IS COMPLETE.
C     WRITE OUT THE SIZE T-MATRICES THAT WILL BE USED FOR
C     THE RITZ VECTOR COMPUTATIONS.
C
      WRITE(10,760)
  760 FORMAT(/' SIZES OF T-MATRICES THAT WILL BE USED IN THE RITZ COMPUT
     1ATIONS'/5X,'J',16X,'GOODEV(J)',1X,'MA(J)')
C
      WRITE(10,770)   (J,GOODEV(J),MA(J), J=1,NGOOD)
  770 FORMAT(I6,E25.14,I6)
      WRITE(10,520)
C
      WRITE(6,780) MTOL
  780 FORMAT(/' THE CUMULATIVE LENGTH OF THE T-EIGENVECTORS IS',I18)
C
      WRITE(6,790) NTVEC,NGOOD
  790 FORMAT(/I6,' T-EIGENVECTORS OUT OF',I6,' REQUESTED WERE COMPUTED')
C
C     SAVE THE T-EIGENVECTORS ON FILE 11?
      IF(TVSTOP.NE.1.AND.SVTVEC.EQ.0) GO TO 850
C
      WRITE(11,800) NTVEC,MTOL,MATNO,SVSEED
  800 FORMAT(I6,3I12,' = NTVEC,MTOL,MATNO,SVSEED')
C
      DO 830 J=1,NGOODC
C     IF MP(J) = MPMIN THEN NO SUITABLE T-EIGENVECTOR IS AVAILABLE
C     FOR THAT EIGENVALUE.
      IF(MP(J).EQ.MPMIN) WRITE(11,810) J,MA(J),GOODEV(J),MP(J)
  810 FORMAT(2I6,E20.12,I6/' TH EIGVAL,T-SIZE,EVALUE,FLAG,NO EIGVEC')
```

```
      IF(MP(J).NE.MPMIN) WRITE(11,820) J,MA(J),GOODEV(J),MP(J)          HHL07590
  820 FORMAT(I6,I6,E20.12,I6/' T-EIGVEC,SIZE T,EVALUE OF A,MP(J)')      HHL07600
      IF(MP(J).EQ.MPMIN) GO TO 830                                      HHL07610
      KI = MINT(J)                                                      HHL07620
      KF = MFIN(J)                                                      HHL07630
C                                                                       HHL07640
      WRITE(11,270) (TVEC(K), K=KI,KF)                                  HHL07650
C                                                                       HHL07660
  830 CONTINUE                                                          HHL07670
C                                                                       HHL07680
      IF(TVSTOP.NE.1) GO TO 850                                         HHL07690
C                                                                       HHL07700
      WRITE(6,840) TVSTOP, NTVEC,NGOOD                                  HHL07710
  840 FORMAT(/' USER SET TVSTOP = ',I1/                                 HHL07720
     1' THEREFORE PROGRAM TERMINATES AFTER T-EIGENVECTOR COMPUTATIONS'/ HHL07730
     1' T-EIGENVECTORS THAT WERE COMPUTED ARE SAVED ON FILE 11'/        HHL07740
     1I8,' T-EIGENVECTORS WERE COMPUTED OUT OF',I7,' REQUESTED'/)       HHL07750
C                                                                       HHL07760
      GO TO 1630                                                        HHL07770
C                                                                       HHL07780
  850 CONTINUE                                                          HHL07790
C     IF NOT ABLE TO COMPUTE ALL THE REQUESTED T-EIGENVECTORS           HHL07800
C     CONTINUE WITH THE LANCZOS VECTOR COMPUTATIONS ANYWAY?             HHL07810
      IF(NTVEC.NE.NGOOD.AND.LVCONT.EQ.0) GO TO 1530                     HHL07820
C                                                                       HHL07830
C     COMPUTE THE MAXIMUM SIZE OF THE T-MATRIX USED FOR THOSE           HHL07840
C     EIGENVALUES WITH GOOD ERROR ESTIMATES.                            HHL07850
C                                                                       HHL07860
      KMAXU = 0                                                         HHL07870
      DO 860 J = 1,NGOODC                                               HHL07880
      MT = IABS(MA(J))                                                  HHL07890
      IF(MT.LT.KMAXU.OR.MP(J).EQ.MPMIN) GO TO 860                       HHL07900
      KMAXU = MT                                                        HHL07910
  860 CONTINUE                                                          HHL07920
C                                                                       HHL07930
      IF(KMAXU.EQ.0) GO TO 1570                                         HHL07940
C                                                                       HHL07950
      WRITE(6,870) KMAXU                                                HHL07960
  870 FORMAT(/I6,' = LARGEST SIZE T-MATRIX TO BE USED IN THE RITZ VECTORHHL07970
     1 COMPUTATIONS')                                                   HHL07980
C                                                                       HHL07990
C     COUNT THE NUMBER OF RITZ VECTORS NOT BEING COMPUTED               HHL08000
      MREJEC = 0                                                        HHL08010
      DO 880 J=1,NGOODC                                                 HHL08020
  880 IF(MP(J).EQ.MPMIN)  MREJEC = MREJEC + 1                           HHL08030
      MREJET = MREJEC + (NGOOD-NGOODC)                                  HHL08040
      IF(MREJET.NE.0) WRITE(6,890) MREJET                               HHL08050
  890 FORMAT(/' RITZ VECTORS ARE NOT COMPUTED FOR',I6,' OF THE EIGENVALUHHL08060
     1ES'/)                                                             HHL08070
      NACT = NGOODC - MREJEC                                            HHL08080
      WRITE(6,900) NGOOD,NTVEC,NACT                                     HHL08090
  900 FORMAT(/I6,' RITZ VECTORS WERE REQUESTED'/I6,' T-EIGENVECTORS WEREHHL08100
     1 COMPUTED'/I6,' RITZ VECTORS WILL BE COMPUTED'/)                  HHL08110
C     CHECK IF THERE ARE ANY RITZ VECTORS TO COMPUTE                    HHL08120
      IF(MREJEC.EQ.NGOODC) GO TO 1550                                   HHL08130
```

```
C                                                                       HHL08140
C     CONTINUE WITH THE LANCZOS VECTOR COMPUTATIONS?                    HHL08150
      IF(LVCONT.EQ.0.AND.MREJEC.NE.0) GO TO 1530                        HHL08160
C                                                                       HHL08170
C     NOW COMPUTE THE RITZ VECTORS.  REGENERATE THE                     HHL08180
C     LANCZOS VECTORS.                                                  HHL08190
C                                                                       HHL08200
      DO 910 I = 1,NMAX                                                 HHL08210
  910 RITVEC(I) = ZEROC                                                 HHL08220
C                                                                       HHL08230
C     REGENERATE THE STARTING VECTOR. THIS MUST BE GENERATED AND        HHL08240
C     NORMALIZED PRECISELY THE WAY IT WAS DONE IN THE EIGENVALUE        HHL08250
C     COMPUTATIONS, OTHERWISE THERE WILL BE A MISMATCH BETWEEN          HHL08260
C     THE T-EIGENVECTORS THAT HAVE BEEN COMPUTED FROM THE T-MATRICES    HHL08270
C     READ IN FROM FILE 2 AND THE LANCZOS VECTORS THAT ARE              HHL08280
C     BEING REGENERATED.                                                HHL08290
C                                                                       HHL08300
C-----------------------------------------------------------------------HHL08310
C                                                                       HHL08320
      IIL = SVSEED                                                      HHL08330
      CALL GENRAN(IIL,G,N)                                              HHL08340
C                                                                       HHL08350
C-----------------------------------------------------------------------HHL08360
C                                                                       HHL08370
      DO 920 I = 1,N                                                    HHL08380
  920 GR(I) = G(I)                                                      HHL08390
C                                                                       HHL08400
C-----------------------------------------------------------------------HHL08410
C                                                                       HHL08420
      CALL GENRAN(IIL,G,N)                                              HHL08430
C                                                                       HHL08440
C-----------------------------------------------------------------------HHL08450
C                                                                       HHL08460
      DO 930 I = 1,N                                                    HHL08470
  930 GC(I) = G(I)                                                      HHL08480
C                                                                       HHL08490
      DO 940 I = 1,N                                                    HHL08500
  940 V2(I) = DCMPLX(GR(I),GC(I))                                       HHL08510
C                                                                       HHL08520
C-----------------------------------------------------------------------HHL08530
      CALL CINPRD(V2,V2,SUM,N)                                          HHL08540
C-----------------------------------------------------------------------HHL08550
C                                                                       HHL08560
      SUM = ONE/DSQRT(SUM)                                              HHL08570
      DO 950 I = 1,N                                                    HHL08580
      V1(I) = ZEROC                                                     HHL08590
  950 V2(I) = V2(I)*SUM                                                 HHL08600
C                                                                       HHL08610
C     LOOP FOR GENERATING REQUIRED RITZ VECTORS (IVEC = 1,KMAXU)        HHL08620
C     USES GRAM-SCHMIDT ORTHOGONALIZATION WITHOUT MODIFICATION          HHL08630
C                                                                       HHL08640
      IVEC = 1                                                          HHL08650
      BATA = ZERO                                                       HHL08660
C                                                                       HHL08670
      GO TO 1010                                                        HHL08680
```

```
C                                                                       HHL08690
  960 CONTINUE                                                          HHL08700
C                                                                       HHL08710
C-----------------------------------------------------------------------HHL08720
C     CMATV(V2,VS,SUM) CALCULATES  VS = A*V2 - SUM*VS                   HHL08730
      SUM = ZERO                                                        HHL08740
      CALL CMATV(V2,VS,SUM)                                             HHL08750
      CALL CINPRD(V2,VS,ALFA,N)                                         HHL08760
C                                                                       HHL08770
C-----------------------------------------------------------------------HHL08780
C                                                                       HHL08790
      DO 970 J=1,N                                                      HHL08800
  970 V1(J) = (VS(J) - BATA*V1(J)) - ALFA*V2(J)                         HHL08810
C                                                                       HHL08820
C-----------------------------------------------------------------------HHL08830
      CALL CINPRD(V1,V1,BATA,N)                                         HHL08840
C-----------------------------------------------------------------------HHL08850
C                                                                       HHL08860
      BATA = DSQRT(BATA)                                                HHL08870
      SUM = ONE/BATA                                                    HHL08880
C                                                                       HHL08890
      TEMP = BETA(IVEC)                                                 HHL08900
      TEMP = DABS(BATA - TEMP)/TEMP                                     HHL08910
      IF (TEMP.LT.1.0D-10)GO TO 990                                     HHL08920
C                                                                       HHL08930
C     THE BETA BEING REGENERATED DO NOT MATCH THE HISTORY FILE          HHL08940
C     SOMETHING IS WRONG IN THE LANCZOS VECTOR GENERATION               HHL08950
C     PROGRAM TERMINATES FOR USER TO CORRECT THE PROBLEM                HHL08960
C     WHICH MUST BE IN THE STARTING VECTOR GENERATION OR IN             HHL08970
C     THE MATRIX-VECTOR MULTIPLY SUBROUTINE CMATV SUPPLIED.             HHL08980
C     THIS SUBROUTINE MUST BE THE SAME ONE USED IN THE                  HHL08990
C     EIGENVALUE COMPUTATIONS OR A MISMATCH WILL ENSUE.                 HHL09000
C                                                                       HHL09010
      WRITE(6,980) IVEC,BATA,BETA(IVEC),TEMP                            HHL09020
  980 FORMAT(/2X,'IVEC',16X,'BATA',10X,'BETA(IVEC)',14X,'RELDIF'/I6,    HHL09030
     13E20.12/' IN LANCZOS VECTOR REGENERATION THE ENTRIES OF THE TRIDIAHHL09040
     1GONAL MATRICES BEING'/' GENERATED ARE NOT THE SAME AS THOSE IN THEHHL09050
     1 MATRIX SUPPLIED ON FILE 2.'/' THEREFORE SOMETHING IS BEING INITIAHHL09060
     1LIZED OR COMPUTED DIFFERENTLY FROM THE WAY'/' IT WAS COMPUTED IN THHL09070
     1HE EIGENVALUE COMPUTATIONS'/' THE PROGRAM TERMINATES FOR THE USER HHL09080
     1TO DETERMINE WHAT THE PROBLEM IS'/)                               HHL09090
      GO TO 1630                                                        HHL09100
C                                                                       HHL09110
C                                                                       HHL09120
  990 CONTINUE                                                          HHL09130
      DO 1000 J = 1,N                                                   HHL09140
      TEMPC = SUM*V1(J)                                                 HHL09150
      V1(J) = V2(J)                                                     HHL09160
 1000 V2(J) = TEMPC                                                     HHL09170
C                                                                       HHL09180
 1010 CONTINUE                                                          HHL09190
C                                                                       HHL09200
      LFIN = 0                                                          HHL09210
      DO 1030 J = 1,NGOODC                                              HHL09220
      LL = LFIN                                                         HHL09230
```

```
      LFIN = LFIN + N                                                   HHL09240
C                                                                       HHL09250
      IF(IABS(MA(J)).LT.IVEC.OR.MP(J).EQ.MPMIN) GO TO 1030              HHL09260
      II = IVEC + MINT(J) - 1                                           HHL09270
      TEMP = TVEC(II)                                                   HHL09280
C     II IS THE (IVEC)TH COMPONENT OF THE T-EIGENVECTOR CONTAINED       HHL09290
C     IN TVEC(MINT(J)).                                                 HHL09300
C                                                                       HHL09310
      DO 1020 K = 1,N                                                   HHL09320
      LL = LL + 1                                                       HHL09330
 1020 RITVEC(LL) = TEMP*V2(K) + RITVEC(LL)                              HHL09340
C                                                                       HHL09350
 1030 CONTINUE                                                          HHL09360
C                                                                       HHL09370
      IVEC = IVEC + 1                                                   HHL09380
      IF (IVEC.LE.KMAXU) GO TO 960                                      HHL09390
C                                                                       HHL09400
C                                                                       HHL09410
C     RITZVECTOR GENERATION IS COMPLETE. NORMALIZE EACH RITZVECTOR.     HHL09420
C     NOTE THAT IF CERTAIN RITZ VECTORS WERE NOT COMPUTED THEN THAT     HHL09430
C     PORTION OF THE RITVEC ARRAY WAS NOT UTILIZED.                     HHL09440
C                                                                       HHL09450
      LFIN = 0                                                          HHL09460
      DO 1130 J = 1,NGOODC                                              HHL09470
C                                                                       HHL09480
      KK = LFIN                                                         HHL09490
      LFIN = LFIN + N                                                   HHL09500
      IF(MP(J).EQ.MPMIN) GO TO 1130                                     HHL09510
C                                                                       HHL09520
      DO 1040 K = 1,N                                                   HHL09530
      KK = KK + 1                                                       HHL09540
 1040 V2(K) = RITVEC(KK)                                                HHL09550
C                                                                       HHL09560
C-----------------------------------------------------------------------HHL09570
      CALL CINPRD(V2,V2,SUM,N)                                          HHL09580
C-----------------------------------------------------------------------HHL09590
C                                                                       HHL09600
      SUM = DSQRT(SUM)                                                  HHL09610
      RNORM(J) = SUM                                                    HHL09620
      TEMP = DABS(ONE-SUM)                                              HHL09630
      SUM = ONE/SUM                                                     HHL09640
C                                                                       HHL09650
      KK = LFIN - N                                                     HHL09660
      DO 1050 K = 1,N                                                   HHL09670
      KK = KK + 1                                                       HHL09680
      V2(K) = SUM*V2(K)                                                 HHL09690
 1050 RITVEC(KK) = V2(K)                                                HHL09700
C                                                                       HHL09710
C    ONLY ENTER NEXT PORTION IF GIVEN MATRIX IS REAL.                   HHL09720
      IF(ISREAL.NE.1) GO TO 1100                                        HHL09730
C                                                                       HHL09740
C    AT THIS POINT RITZ VECTOR IS IN V2.                                HHL09750
C    THIS PROGRAM CAN BE USED ON REAL MATRICES TO DETERMINE             HHL09760
C    WHICH IF ANY EIGENVALUES ARE A-MULTIPLE AND IF SO TO COMPUTE       HHL09770
C    TWO EIGENVECTORS FOR THOSE EIGENVALUES THAT ARE MULTIPLE AND ONE   HHL09780
```

```
C    FOR THOSE THAT ARE NOT MULTIPLE. HERE ONLY IDENTIFIES WHETHER
C    EIGENVALUE IS AT LEAST DOUBLE.  THIS IS DONE BY CHECKING THE
C    RATIOS OF SUCCEEDING REAL AND IMAGINARY PARTS OF THE COMPUTED
C    RITZ VECTORS.
C
      SUM = DIMAG(V2(1))/DREAL(V2(1))
      DO 1060 K=2,N
      TEMP = DREAL(V2(K))
      IF(DABS(TEMP).LT.1.D-9) GO TO 1060
      TEMP = DIMAG(V2(K))/DREAL(V2(K))
      IF(DABS(TEMP - SUM).LE.1.D-6) GO TO 1060
      MULEVA(J) = 2
      GO TO 1070
 1060 CONTINUE
      MULEVA(J) = 1
 1070 IF(MULEVA(J).EQ.2) WRITE(6,1090) J,GOODEV(J)
      IF(MULEVA(J).EQ.1) WRITE(6,1080) J,GOODEV(J)
 1080 FORMAT(I6,'TH EIGENVALUE CONSIDERED =',E20.12,' IS SIMPLE')
 1090 FORMAT(I6,'TH EIGENVALUE CONSIDERED =',E20.12,' IS MULTIPLE')
C
 1100 CONTINUE
C
      IF (IWRITE.NE.0) WRITE(6,1110) J,GOODEV(J)
 1110 FORMAT(/I5,' TH EIGENVALUE CONSIDERED = ',E20.12/)
C
      IF (IWRITE.NE.0) WRITE(6,1120) TERR(J),TBETA(J),TEMP
 1120 FORMAT(' NORM OF ERROR IN T-EIGENVECTOR = ',E14.3/
     1 ' BETA(MA(J)+1)*U(MA(J))  = ',E14.3/
     1 ' ABS(NORM(RITVEC) - 1.0)  = ',E14.3/)
C
      LINT = LFIN - N + 1
      EVAL = EVNEW(J)
C
C-----------------------------------------------------------------------
C
      CALL CMATV(RITVEC(LINT),V2,EVAL)
C
C-----------------------------------------------------------------------
C
C     COMPUTE ERROR IN RITZ VECTOR CONSIDERED AS A EIGENVECTOR OF A.
C     V2 = A*RITVEC - EVAL*RITVEC
C
C-----------------------------------------------------------------------
      CALL CINPRD(V2,V2,SUM,N)
C-----------------------------------------------------------------------
C
      SUM = DSQRT(SUM)
      ERR(J) = SUM
      GAP = ABS(AMINGP(J))
      ERRDGP(J) = SUM/GAP
C
 1130 CONTINUE
C
C
C     RITZVECTORS ARE NORMALIZED AND ERROR ESTIMATES ARE IN ERR ARRAY
```

```
C     AND IN ERRDGP ARRAY. STORE EVERYTHING
C
C
      WRITE(9,1140)
 1140 FORMAT(6X,'GOODEV(J)',1X,'MA(J)',4X,'A MINGAP',6X,'AERROR',2X,
     1 'AERROR/GAP',6X,'TERROR')
C
      WRITE(13,1150)
 1150 FORMAT(16X,'GOODEV(J)',5X,'RITZNORM',6X,'AMINGAP',5X,
     1 'TBETA(J)',5X,'TLAST(J)')
C
      DO 1180 J=1,NGOODC
C
      IF(MP(J).EQ.MPMIN) GO TO 1180
C
      WRITE(9,1160)EVNEW(J),MA(J),AMINGP(J),ERR(J),ERRDGP(J),TERR(J)
 1160 FORMAT(E15.8,I6,4E12.4)
C
      WRITE(13,1170) EVNEW(J),RNORM(J),AMINGP(J),TBETA(J),TLAST(J)
 1170 FORMAT(E25.14,4E13.5)
C
 1180 CONTINUE
C
      IF(MREJEC.EQ.0) GO TO 1260
      WRITE(9,1190)
 1190 FORMAT(/' RITZ VECTORS WERE NOT COMPUTED FOR THE FOLLOWING EIGENVA
     1LUES'/' EITHER BECAUSE THEY HAD NOT CONVERGED OR BECAUSE THE ERROR
     1 ESTIMATE'/'  WAS NOT AS SMALL AS DESIRED'/)
C
      DO 1250 J = 1,NGOODC
      IF(MP(J).NE.MPMIN) GO TO 1250
C     WRITE OUT MESSAGE FOR EACH EIGENVALUE FOR WHICH NO EIGENVECTOR
C     WAS COMPUTED.
C
      WRITE(9,1200)
 1200 FORMAT(6X,'GOODEV(J)',3X,'MA(J)',5X,'AMINGP(J)',6X,'TLAST(J)',3X,
     1'MP(J)')
      WRITE(9,1210) GOODEV(J),MA(J),AMINGP(J),TBETA(J),MP(J)
 1210 FORMAT(E15.8,I8,2E14.4,I8)
C
      WRITE(13,1220)
 1220 FORMAT(/' RITZ VECTORS WERE NOT COMPUTED FOR THE FOLLOWING EIGENVA
     1LUES'/' EITHER BECAUSE THEY HAD NOT CONVERGED OR BECAUSE'/' THE ER
     1ROR ESTIMATE WAS NOT AS SMALL AS DESIRED'/)
C
      WRITE(13,1230)
 1230 FORMAT(6X,'GOODEV(J)',3X,'MA(J)',3X,'M1(J)',3X,'M2(J)',3X,'MP(J)'
     1/)
      WRITE(13,1240) GOODEV(J),MA(J),M1(J),M2(J),MP(J)
 1240 FORMAT(E15.8,4I8)
C
 1250 CONTINUE
 1260 CONTINUE
C
      WRITE(9,1270)
```

```
 1270 FORMAT(/' ABOVE ARE ERROR ESTIMATES FOR THE A AND T EIGENVECTORS'/HHL10890
     1 ' ASSOCIATED WITH THE GOODEV LISTED IN COLUMN 1'/                HHL10900
     1 ' AERROR = NORM(A*X - EV*X)  TERROR = NORM(T*Y - EV*Y) '/        HHL10910
     1 ' WHERE T = T(1,MA(J))    X = RITZ VECTOR = V*Y  V = SUCCESSIVE'/HHL10920
     1 ' LANCZOS VECTORS. A MINGAP = GAP TO NEAREST A-EIGENVALUE'//)    HHL10930
C                                                                       HHL10940
      WRITE(13,1280)                                                    HHL10950
 1280 FORMAT(/' ABOVE ARE ERROR ESTIMATES ASSOCIATED WITH THE GOODEV'/  HHL10960
     1 ' RITZNORM = NORM(RITZ VECTOR)'/                                 HHL10970
     1 ' TBETA(J) = CDABS(BETA(MA(J)+1)*Y(MA(J))),  T*Y = GOODEV*Y'/    HHL10980
     1 ' TLAST(J) = CDABS(Y(MA(J))'/                                    HHL10990
     1 ' AMINGAP = DISTANCE TO CLOSEST COMPUTED GOOD T-EIGENVALUE'/)    HHL11000
C                                                                       HHL11010
C     NUMBER OF RITZ VECTORS COMPUTED                                   HHL11020
      NCOMPU = NGOODC - MREJEC                                          HHL11030
      WRITE(12,1290) N,NCOMPU,NGOODC,MATNO                              HHL11040
 1290 FORMAT(3I6,I12,' SIZE A, NO.RITZVECS, NO.EVALUES,MATNO')          HHL11050
C                                                                       HHL11060
      LFIN = 0                                                          HHL11070
      DO 1350 J = 1,NGOODC                                              HHL11080
      LINT = LFIN + 1                                                   HHL11090
      LFIN = LFIN + N                                                   HHL11100
C                                                                       HHL11110
      IF(MP(J).EQ.MPMIN) GO TO 1330                                     HHL11120
C     RITZ VECTOR WAS COMPUTED                                          HHL11130
      WRITE(12,1300) J, GOODEV(J), MP(J)                                HHL11140
 1300 FORMAT(I6,4X,E20.12,I6,' J, EIGENVAL, MP(J)')                     HHL11150
C                                                                       HHL11160
      WRITE(12,1310) ERR(J),ERRDGP(J)                                   HHL11170
 1310 FORMAT(2E15.5,' = NORM(A*Z-EVAL*Z) AND  NORM(A*Z-EVAL*Z)/MINGAP') HHL11180
C                                                                       HHL11190
      WRITE(12,1320) (RITVEC(LL), LL=LINT,LFIN)                         HHL11200
 1320 FORMAT(4E20.12)                                                   HHL11210
      GO TO 1350                                                        HHL11220
C     NO RITZ VECTOR WAS COMPUTED FOR THIS EIGENVALUE                   HHL11230
 1330 WRITE(12,1340) J,GOODEV(J),MP(J)                                  HHL11240
 1340 FORMAT(I6,4X,E20.12,I6,' J,EIGVALUE,NO RITZ VECTOR COMPUTED')     HHL11250
C                                                                       HHL11260
 1350 CONTINUE                                                          HHL11270
C                                                                       HHL11280
C     DID ANY T-MATRICES INCLUDE OFF-DIAGONAL ENTRIES SMALLER THAN      HHL11290
C     DESIRED, AS SPECIFIED BY BTOL?                                    HHL11300
C                                                                       HHL11310
      IF(IB.GT.0) GO TO 1380                                            HHL11320
      WRITE(6,1360) KMAXU                                               HHL11330
 1360 FORMAT(/' FOR LARGEST T-MATRIX CONSIDERED',I7,' CHECK THE SIZE OF HHL11340
     1BETAS')                                                           HHL11350
C                                                                       HHL11360
C-----------------------------------------------------------------------HHL11370
C                                                                       HHL11380
      CALL TNORM(ALPHA,BETA,BKMIN,TEMP,KMAXU,IBMT)                      HHL11390
C                                                                       HHL11400
C-----------------------------------------------------------------------HHL11410
C                                                                       HHL11420
      IF(IBMT.LT.0) WRITE (6,1370)                                      HHL11430
```

```
 1370 FORMAT(/' WARNING THE T-MATRICES FOR ONE OR MORE OF THE EIGENVALUEHHL11440
     1S CONSIDERED'/' HAD AN OFF-DIAGONAL ENTRY THAT WAS SMALLER THAN THHHL11450
     1E BETA TOLERANCE THAT WAS SPECIFIED'/)                            HHL11460
 1380 CONTINUE                                                          HHL11470
C                                                                       HHL11480
       GO TO 1630                                                       HHL11490
C                                                                       HHL11500
 1390 WRITE(6,1400) NGOOD,NMAX,MDIMRV                                   HHL11510
 1400 FORMAT(/I4,' RITZ VECTORS WERE REQUESTED BUT THE REQUIRED DIMENSIOHHL11520
     1N',I6/' IS LARGER THAN THE USER-SPECIFIED DIMENSION OF RITVEC',I6 HHL11530
     1/' THEREFORE, THE EIGENVECTOR PROCEDURE TERMINATES FOR THE USER TOHHL11540
     1 INTERVENE')                                                      HHL11550
C                                                                       HHL11560
      GO TO 1630                                                        HHL11570
C                                                                       HHL11580
 1410 WRITE(6,1420) NOLD,N,MATOLD,MATNO                                 HHL11590
 1420 FORMAT(/' PARAMETERS READ FROM FILE 3 DO NOT AGREE WITH THOSE SPECHHL11600
     1 IFIED'/' BY THE USER.   NOLD,N,MATOLD,MATNO = '/2I6,2I12/        HHL11610
     1' THEREFORE, PROGRAM TERMINATES FOR USER TO RESOLVE THE DIFFERENCEHHL11620
     1S'/)                                                              HHL11630
C                                                                       HHL11640
      GO TO 1630                                                        HHL11650
C                                                                       HHL11660
 1430 WRITE(6,1440)                                                     HHL11670
 1440 FORMAT(/' PARAMETERS IN ALPHA,BETA FILE READ IN DO NOT AGREE WITH HHL11680
     1 THOSE'/' SPECIFIED BY THE USER.  THEREFORE, THE PROCEDURE TERMINAHHL11690
     1TES'/' FOR THE USER TO RESOLVE THE DIFFERENCES.'/)                HHL11700
C                                                                       HHL11710
      GO TO 1630                                                        HHL11720
C                                                                       HHL11730
 1450 WRITE(6,1460) KMAX,MEV                                            HHL11740
 1460 FORMAT(/' ON ALPHA,BETA HEADER KMAX = ',I6/                       HHL11750
     1' BUT EIGENVALUES WERE COMPUTED AT MEV = ',I6,' PROGRAM STOPS'/)  HHL11760
C                                                                       HHL11770
      GO TO 1630                                                        HHL11780
C                                                                       HHL11790
 1470 WRITE(6,1480)                                                     HHL11800
 1480 FORMAT(/' PROGRAM COMPUTED 1ST GUESSES ON T-MATRIX SIZES, READ THEHHL11810
     1M TO FILE 10'/' THEN TERMINATED AS REQUESTED.')                   HHL11820
      GO TO 1630                                                        HHL11830
C                                                                       HHL11840
 1490 WRITE(6,1500) MTOL, MDIMTV                                        HHL11850
 1500 FORMAT(/' PROGRAM TERMINATES BECAUSE THE TVEC DIMENSION ANTICIPATEHHL11860
     1D',I7/'  IS LARGER THAN THE TVEC DIMENSION',I7,' SPECIFIED BY THE HHL11870
     1USER.'/'  USER MAY RESET THE TVEC DIMENSION AND RESTART THE PROGRAHHL11880
     1M')                                                               HHL11890
      GO TO 1630                                                        HHL11900
C                                                                       HHL11910
 1510 WRITE(6,1520)                                                     HHL11920
 1520 FORMAT(/' PROGRAM TERMINATES BECAUSE NO SUITABLE T-EIGENVECTORS WEHHL11930
     1RE IDENTIFIED'/'  FOR ANY OF THE EIGENVALUES SUPPLIED.  PROBLEM COHHL11940
     1ULD BE CAUSED'/'  BY TOO SMALL A TVEC DIMENSION OR SIMPLY BE THAT HHL11950
     1IT WAS NOT POSSIBLE'/'  TO IDENTIFY T-VECTORS.  USER SHOULD CHECK HHL11960
     1OUTPUT'/)                                                         HHL11970
      GO TO 1630                                                        HHL11980
```

```
C                                                                       HHL11990
 1530 WRITE(6,1540) LVCONT,NTVEC,NGOOD                                  HHL12000
 1540 FORMAT(/' LVCONT FLAG =',I2,' AND NUMBER ',I5,' OF T-EIGENVECTORS HHL12010
     1 COMPUTED N.E.'/' NUMBER',I5,' REQUESTED SO PROGRAM TERMINATES'/) HHL12020
      GO TO 1630                                                        HHL12030
 1550 WRITE(6,1560)                                                     HHL12040
 1560 FORMAT(/' PROGRAM TERMINATES WITHOUT COMPUTING ANY RITZ VECTORS'/ HHL12050
     1 ' BECAUSE ALL T-EIGENVECTORS WERE REJECTED AS NOT SUITABLE'/     HHL12060
     1 ' PROBABLE CAUSE IS LACK OF CONVERGENCE OF THE EIGENVALUES'/)    HHL12070
       GO TO 1630                                                       HHL12080
C                                                                       HHL12090
 1570 WRITE(6,1580)                                                     HHL12100
 1580 FORMAT(/' PROGRAM INDICATES THAT IT IS NOT POSSIBLE TO COMPUTE ANYHHL12110
     1 OF THE'/' REQUESTED EIGENVECTORS. THEREFORE PROGRAM TERMINATES') HHL12120
      DO 1590 J=1,NGOODC                                                HHL12130
 1590 WRITE(6,1600)  J,GOODEV(J),MP(J)                                  HHL12140
 1600 FORMAT(/4X,' J',11X,'GOODEV(J)',4X,'MP(J)'/I6,E20.12,I9)          HHL12150
      GO TO 1630                                                        HHL12160
C                                                                       HHL12170
 1610 WRITE(6,1620) MBETA,KMAXN                                         HHL12180
 1620 FORMAT(/' PROGRAM TERMINATES BECAUSE THE STORAGE ALLOTTED FOR THE HHL12190
     1BETA ARRAY',I8/' IS NOT SUFFICIENT FOR THE ENLARGED KMAX =',I8,' THHL12200
     1HAT THE PROGRAM WANTS'/' USER CAN ENLARGE THE ALPHA AND BETA ARRAYHHL12210
     1S AND RERUN THE PROGRAM.'/)                                       HHL12220
C                                                                       HHL12230
 1630 CONTINUE                                                          HHL12240
C                                                                       HHL12250
      STOP                                                              HHL12260
C-----END OF MAIN PROGRAM FOR LANCZOS HERMITIAN EIGENVECTOR COMPUTATIONSHHL12270
      END                                                               HHL12280
```

3.4 HLEMULT: LANCZS and Sample Matrix-Vector Multiply Subroutines

```
C-----HLEMULT----HERMITIAN MATRICES-------------------------------------
C  Authors:  Jane Cullum and Ralph A. Willoughby (deceased)
C            Los Alamos National Laboratory
C            Los Alamos, New Mexico 87544
C            cullumj@lanl.gov
c
c  These codes are copyrighted by the authors.  These codes
c  and modifications of them or portions of them are NOT to be
c  incorporated into any commercial codes without legal agreements
c  with the authors.  If these codes or portions of them
c  are used in other scientific or engineering research works
c  the names of the authors of these codes and appropriate
c  references to their written work are to be incorporated in the
c  derivative works.
c
c  This header is not to be removed from these codes.
C
C         REFERENCE: Cullum and Willoughby, Chapters 1,2,3,4
C         Lanczos Algorithms for Large Symmetric Eigenvalue Computations
C         VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in
C         Applied Mathematics, 2002. SIAM Publications,
C         Philadelphia, PA. USA
C
C     CONTAINS SUBROUTINE LANCZS AND SAMPLE USPEC, CMATV
C     USED BY THE HERMITIAN VERSION OF THE LANCZOS ALGORITHMS
C
C     PORTABILITY:
C     THESE PROGRAMS ARE NOT PORTABLE DUE TO THE USE OF COMPLEX*16
C     VARIABLES.  MOREOVER, THE PFORT VERIFIER IDENTIFIED THE
C     FOLLOWING ADDITIONAL NONPORTABLE CONSTRUCTIONS:
C     1.  THE ENTRY MECHANISM USED TO PASS THE STORAGE
C         LOCATIONS OF THE USER-SPECIFIED MATRIX FROM THE
C         SUBROUTINE USPEC TO THE MATRIX-VECTOR SUBROUTINE CMATV.
C     2.  IN THE PROGRAMS PROVIDED FOR 'HERMITIAN POISSON' TEST MATRICES
C         USPEC CONTAINS FREE FORMAT (8,*), AND FORMAT (20A4); AND
C         EXACT ERROR SUBROUTINE CONTAINS DATA/MACHEP DEFINITION.
C
C
C-----LANCZS-COMPUTE THE LANCZOS TRIDIAGONAL MATRICES-------------------
C
C     GRAM-SCHMIDT ORTHOGONALIZATION WITHOUT MODIFICATION
C     REQUIRES EXTRA VECTOR VS IN LANCZS.  MODIFICATION IS NOT
C     PERMISSIBLE IN THE HERMITIAN CASE BECAUSE COMPLEX PORTION
C     OF THE MODIFICATION COULD NOT BE INCORPORATED.
C
      SUBROUTINE LANCZS(MATVEC,V1,V2,VS,ALPHA,BETA,GR,GC,G,KMAX,MOLD1,N,
     1 IIX)
C
C-----------------------------------------------------------------------
      COMPLEX*16 V1(1), V2(1), VS(1), ZEROC, TEMP
```

```
      DOUBLE PRECISION  ALPHA(1), BETA(1), BATA, SUM, ONE, ZERO         HHL00300
      DOUBLE PRECISION GR(1),GC(1)                                      HHL00310
      REAL  G(1)                                                        HHL00320
      EXTERNAL MATVEC                                                   HHL00330
      DOUBLE PRECISION  DSQRT                                           HHL00340
C-----------------------------------------------------------------------HHL00350
C                                                                       HHL00360
      ZERO = 0.D0                                                       HHL00370
      ONE = 1.D0                                                        HHL00380
      ZEROC = DCMPLX(ZERO,ZERO)                                         HHL00390
C                                                                       HHL00400
      IF(MOLD1.GT.1)GO TO 50                                            HHL00410
C                                                                       HHL00420
C     ALPHA/BETA GENERATION STARTS AT I = 1                             HHL00430
C     MOLD1 = 1 SET V1 = 0. AND V2 = RANDOM UNIT VECTOR                 HHL00440
      IIL=IIX                                                           HHL00450
C                                                                       HHL00460
C-----------------------------------------------------------------------HHL00470
      CALL GENRAN(IIL,G,N)                                              HHL00480
C-----------------------------------------------------------------------HHL00490
C                                                                       HHL00500
      DO 10 I = 1,N                                                     HHL00510
   10 GR(I) = G(I)                                                      HHL00520
C                                                                       HHL00530
C-----------------------------------------------------------------------HHL00540
      CALL GENRAN(IIL,G,N)                                              HHL00550
C-----------------------------------------------------------------------HHL00560
C                                                                       HHL00570
      DO 20 I = 1,N                                                     HHL00580
   20 GC(I) = G(I)                                                      HHL00590
C                                                                       HHL00600
      DO 30 I = 1,N                                                     HHL00610
   30 V2(I) = DCMPLX(GR(I),GC(I))                                       HHL00620
C                                                                       HHL00630
C-----------------------------------------------------------------------HHL00640
      CALL CINPRD(V2,V2,SUM,N)                                          HHL00650
C-----------------------------------------------------------------------HHL00660
C                                                                       HHL00670
      SUM = ONE/DSQRT(SUM)                                              HHL00680
      DO 40 I = 1,N                                                     HHL00690
      V1(I) = ZEROC                                                     HHL00700
   40 V2(I) = V2(I)*SUM                                                 HHL00710
      BETA(1) = ZERO                                                    HHL00720
C                                                                       HHL00730
C     ALPHA BETA GENERATION LOOP                                        HHL00740
   50 CONTINUE                                                          HHL00750
C                                                                       HHL00760
      DO 80 I=MOLD1,KMAX                                                HHL00770
      SUM = ZERO                                                        HHL00780
C                                                                       HHL00790
C-----------------------------------------------------------------------HHL00800
C     MATVEC(V2,VS,SUM) CALCULATES  VS = A*V2 - SUM*VS                  HHL00810
      CALL MATVEC(V2,VS,SUM)                                            HHL00820
      CALL CINPRD(V2,VS,SUM,N)                                          HHL00830
C-----------------------------------------------------------------------HHL00840
```

```
C
      ALPHA(I) = SUM
      BATA = BETA(I)
      DO 60 J=1,N
   60 V1(J) = (VS(J)-BATA*V1(J)) - SUM*V2(J)
C
C-----------------------------------------------------------------------
      CALL CINPRD(V1,V1,SUM,N)
C-----------------------------------------------------------------------
C
      IN = I+1
      BETA(IN) = DSQRT(SUM)
      SUM = ONE/BETA(IN)
      DO 70 J=1,N
      TEMP = SUM*V1(J)
      V1(J) = V2(J)
   70 V2(J) = TEMP
   80 CONTINUE
C     END ALPHA, BETA GENERATION LOOP
C
C-----END OF LANCZS-----------------------------------------------------
C
      RETURN
      END
C
C-----USPEC-GENERAL SPARSE, HERMITIAN MATRIX----------------------------
C
C     SUBROUTINE USPEC(N,MATNO)
      SUBROUTINE GUSPEC(N,MATNO)
C
C-----------------------------------------------------------------------
      COMPLEX*16 A(3000)
      DOUBLE PRECISION  AD(1000)
      INTEGER  IROW(3000),ICOL(1000)
C-----------------------------------------------------------------------
C    DIMENSION ARRAYS NEEDED TO DEFINE MATRIX, READ IN VALUES FOR
C    ARRAYS AND THEN PASS THE STORAGE LOCATIONS OF THESE ARRAYS TO
C    THE MATRIX-VECTOR MULTIPLY SUBROUTINE CMATV.
C
C    USER-SUPPLIED MATRIX IS STORED IN FOLLOWING SPARSE FORMAT:
C    N = ORDER OF A-MATRIX
C    NZS = NUMBER OF NONZERO SUBDIAGONAL ENTRIES IN A
C    NZL = INDEX OF LAST COLUMN CONTAINING NONZERO SUBDIAGONAL ENTRIES
C    ICOL(J), J=1,NZL IS THE NUMBER OF NONZERO SUBDIAGONAL ELEMENTS
C             IN COLUMN J.
C    IROW(K), K = 1,NZS, IS THE ROW INDEX FOR CORRESPONDING A(K).
C    AD(I), I=1,N ARE DIAGONAL ENTRIES (INCLUDING ANY 0 DIAGONAL
C           ENTRIES)
C    A(K), K=1,NZS ARE NONZERO SUBDIAGONAL ENTRIES, LISTED BY COLUMN.
C    FOR J > NZL THERE ARE NO NONZERO SUBDIAGONAL ELEMENTS IN COLUMN J.
C    ICOL(J) = 0 IS ALLOWED
C
C-----------------------------------------------------------------------
C     IN THIS SAMPLE SUBROUTINE THE ARRAYS ARE READ IN FROM FILE 8
C
```

```
      READ(8,10) NZS,NOLD,NZL,MATOLD                                    HHL01400
   10 FORMAT(I10,2I6,I8)                                                HHL01410
C                                                                       HHL01420
      WRITE(6,20) NZS,NOLD,NZL,MATOLD                                   HHL01430
   20 FORMAT(I10,2I6,I8,' = NZS,NOLD,NZL,MATOLD'/)                      HHL01440
C                                                                       HHL01450
C     TEST OF PARAMETER CORRECTNESS                                     HHL01460
      ITEMP = (NOLD-N)**2 + (MATNO-MATOLD)**2                           HHL01470
C                                                                       HHL01480
      IF(ITEMP.EQ.0) GO TO 40                                           HHL01490
C                                                                       HHL01500
      WRITE(6,30)                                                       HHL01510
   30 FORMAT(/' PROGRAM TERMINATES BECAUSE EITHER ORDERS OF OR LABELS FOHHL01520
     1R MATRIX DISAGREE'/)                                              HHL01530
      GO TO 80                                                          HHL01540
C                                                                       HHL01550
   40 CONTINUE                                                          HHL01560
C                                                                       HHL01570
C     NUMBER OF NONZERO SUBDIAGONAL ENTRIES IN EACH COLUMN IS READ      HHL01580
C     THEN THE CORRESPONDING ROW INDEX FOR EACH SUCH ENTRY IS READ      HHL01590
      READ(8,50) (ICOL(K), K=1,NZL)                                     HHL01600
      READ(8,50) (IROW(K), K=1,NZS)                                     HHL01610
   50 FORMAT(13I6)                                                      HHL01620
C     DIAGONAL IS READ FIRST, THEN NONZERO BELOW DIAGONAL ENTRIES       HHL01630
      READ(8,60) (AD(K), K=1,N)                                         HHL01640
   60 FORMAT(4E20.12)                                                   HHL01650
      READ(8,70) (A(K), K=1,NZS)                                        HHL01660
C  50 FORMAT(4Z20)                                                      HHL01670
   70 FORMAT(4E20.12)                                                   HHL01680
C                                                                       HHL01690
C-----------------------------------------------------------------------HHL01700
C     PASS STORAGE LOCATIONS OF ARRAYS THAT DEFINE THE MATRIX TO        HHL01710
C     THE MATRIX-VECTOR MULTIPLY SUBROUTINE CMATV                       HHL01720
      CALL CMATVE(A,AD,ICOL,IROW,N,NZL)                                 HHL01730
C-----------------------------------------------------------------------HHL01740
C                                                                       HHL01750
      RETURN                                                            HHL01760
   80 STOP                                                              HHL01770
C                                                                       HHL01780
C-----END OF USPEC FOR GENERAL, SPARSE HERMITIAN MATRICES---------------HHL01790
      END                                                               HHL01800
C                                                                       HHL01810
C-----START OF MATRIX-VECTOR MULTIPLY-GENERAL SPARSE HERMITIAN--------- HHL01820
C                                                                       HHL01830
C     SUBROUTINE CMATV(W,U,SUM)                                         HHL01840
      SUBROUTINE GCMATV(W,U,SUM)                                        HHL01850
C                                                                       HHL01860
C-----------------------------------------------------------------------HHL01870
      COMPLEX*16  U(1),W(1),A(1)                                        HHL01880
      DOUBLE PRECISION  AD(1),SUM                                       HHL01890
      INTEGER  IROW(1),ICOL(1)                                          HHL01900
C-----------------------------------------------------------------------HHL01910
C    SPARSE MATRIX-VECTOR MULTIPLY FOR LANCZS  U = A*W - SUM*U          HHL01920
C    SEE USPEC SUBROUTINE FOR DESCRIPTION OF THE ARRAYS THAT DEFINE     HHL01930
C    THE MATRIX                                                         HHL01940
```

```
C                                                                       HHL01950
      GO TO 3                                                           HHL01960
C                                                                       HHL01970
C-----------------------------------------------------------------------HHL01980
C     STORAGE LOCATIONS OF ARRAYS ARE PASSED TO CMATV FROM USPEC        HHL01990
      ENTRY CMATVE(A,AD,ICOL,IROW,N,NZL)                                HHL02000
C-----------------------------------------------------------------------HHL02010
C                                                                       HHL02020
      GO TO 4                                                           HHL02030
    3 CONTINUE                                                          HHL02040
C                                                                       HHL02050
C     COMPUTE THE DIAGONAL TERMS                                        HHL02060
      DO 10 I = 1,N                                                     HHL02070
   10 U(I) = AD(I)*W(I)-SUM*U(I)                                        HHL02080
C                                                                       HHL02090
C     COMPUTE BY COLUMN                                                 HHL02100
      LLAST = 0                                                         HHL02110
      DO 30 J = 1,NZL                                                   HHL02120
C                                                                       HHL02130
      IF (ICOL(J).EQ.0) GO TO 30                                        HHL02140
      LFIRST = LLAST + 1                                                HHL02150
      LLAST = LLAST + ICOL(J)                                           HHL02160
C                                                                       HHL02170
      DO 20 L = LFIRST,LLAST                                            HHL02180
      I = IROW(L)                                                       HHL02190
C                                                                       HHL02200
      U(I) = U(I) + A(L)*W(J)                                           HHL02210
      U(J) = U(J) + DCONJG(A(L))*W(I)                                   HHL02220
C                                                                       HHL02230
   20 CONTINUE                                                          HHL02240
C                                                                       HHL02250
   30 CONTINUE                                                          HHL02260
C                                                                       HHL02270
    4 RETURN                                                            HHL02280
C                                                                       HHL02290
C-----END OF CMATV-GENERAL, SPARSE, HERMITIAN MATRICES -----------------HHL02300
      END                                                               HHL02310
C                                                                       HHL02320
C-----USPEC, CMATV, EXEVG, AND HEXVEC FOR HERMITIAN 'POISSON' MATRICES--HHL02330
C                                                                       HHL02340
C-----USPEC (HERMITIAN POISSON MATRICES)--------------------------------HHL02350
C                                                                       HHL02360
      SUBROUTINE HUSPEC(N,MATNO)                                        HHL02370
C     SUBROUTINE USPEC(N,MATNO)                                         HHL02380
C                                                                       HHL02390
C-----------------------------------------------------------------------HHL02400
      DOUBLE PRECISION  C0,C1,C2,HALF,ONE,SCR,SCI,ANGLE,TEMP            HHL02410
      COMPLEX*16 SC,TC,CL0,CL1,CL3,CL4                                  HHL02420
      REAL EXPLAN(20)                                                   HHL02430
      DOUBLE PRECISION EIGVAL(1000)                                     HHL02440
      REAL GAPS(1000)                                                   HHL02450
      INTEGER MULTS(1000)                                               HHL02460
C-----------------------------------------------------------------------HHL02470
      HALF = 0.5D0                                                      HHL02480
      ONE  = 1.0D0                                                      HHL02490
```

```
C                                                                       HHL02500
C     READ IN PARAMETERS TO DEFINE MATRIX                               HHL02510
C     MATRIX IS COMPLEX DIAGONAL SIMILITARY TRANSFORM OF REAL SYMMETRIC HHL02520
C     POISSON MATRIX WHICH HAS SYMMETRIC TOEPLITZ BLOCKS ALONG          HHL02530
C     THE DIAGONAL, EACH ONE OF WHICH HAS THE PARAMETER C2 ALONG THE    HHL02540
C     DIAGONAL AND -C0 ABOVE AND BELOW THE DIAGONAL, AND OFF-DIAGONAL   HHL02550
C     BLOCKS THAT ARE DIAGONAL WITH DIAGONAL ENTRIES -C1.  EACH BLOCK   HHL02560
C     IS KX*KX AND THERE ARE KY BLOCKS.  THE HERMITIAN VERSION IS       HHL02570
C     OBTAINED BY APPLYING A DIAGONAL SIMILARITY TRANSFORM TO THE       HHL02580
C     REAL MATRIX WHERE THIS TRANSFORMATION IS SUCH THAT ITS            HHL02590
C     DIAGONAL ENTRIES ARE (SC)**(K-1), K = 1,...,N, WHERE SC           HHL02600
C     HAS MODULUS 1.                                                    HHL02610
C                                                                       HHL02620
      READ(8,10) EXPLAN                                                 HHL02630
      WRITE(6,10) EXPLAN                                                HHL02640
      READ(8,10) EXPLAN                                                 HHL02650
   10 FORMAT(20A4)                                                      HHL02660
C     IF MTYPE = 0 WE HAVE ZERO BOUNDARY CONDITIONS                     HHL02670
C     IF MTYPE = 1 WE HAVE NORMAL DERIVATIVE BOUNDARY CONDITIONS        HHL02680
C     NOTE THAT SUBROUTINES EXEVG AND HEXVEC ARE VALID ONLY FOR         HHL02690
C     MTYPE = 0.                                                        HHL02700
      READ(8,*) NOLD,MATOLD,IVEC,MTYPE                                  HHL02710
      WRITE(6,20) NOLD,MATOLD                                           HHL02720
   20 FORMAT(' ORDER OF MATRIX READ FROM FILE =',I6/' MATRIX NUMBER =', HHL02730
     1I8/)                                                              HHL02740
      IF(MTYPE.EQ.0) WRITE(6,30)                                        HHL02750
   30 FORMAT(/' HERMITIAN POISSON CORRESPONDING TO ZERO BOUNDARY CONDITIHHL02760
     1ONS'/)                                                            HHL02770
      IF(MTYPE.EQ.1) WRITE(6,40)                                        HHL02780
   40 FORMAT(/' HERMITIAN POISSON CORRESPONDING TO NORMAL DERIVATIVE BOUHHL02790
     1NDARY CONDITIONS'/)                                               HHL02800
      IF(IVEC.NE.0.AND.MTYPE.EQ.0) WRITE(6,50)                          HHL02810
   50 FORMAT(' COMPUTE THE TRUE EIGENVALUES AND PUT IN FN TRUEEVAL'/)   HHL02820
C                                                                       HHL02830
C     TEST OF PARAMETER CORRECTNESS                                     HHL02840
      ITEMP = (NOLD-N)**2 + (MATNO-MATOLD)**2                           HHL02850
C                                                                       HHL02860
      IF(ITEMP.EQ.0) GO TO 70                                           HHL02870
C                                                                       HHL02880
      WRITE(6,60)                                                       HHL02890
   60 FORMAT(' PROGRAM TERMINATES BECAUSE EITHER ORDERS OF OR LABELS FORHHL02900
     1 MATRIX DISAGREE')                                                HHL02910
      GO TO 150                                                         HHL02920
C                                                                       HHL02930
   70 CONTINUE                                                          HHL02940
C                                                                       HHL02950
      READ(8,10) EXPLAN                                                 HHL02960
      READ(8,*)  C0,KX,KY                                               HHL02970
      IF (KX.GT.4.AND.KY.GT.4) GO TO 90                                 HHL02980
      WRITE(6,80) KX,KY                                                 HHL02990
   80 FORMAT(2I6,' = KX KY  ONE OR BOTH OF KX KY TOO SMALL SO STOP'/)   HHL03000
      GO TO 150                                                         HHL03010
   90 CONTINUE                                                          HHL03020
      READ(8,10) EXPLAN                                                 HHL03030
C     BELOW SC = COS(ANGLE) + I SIN(ANGLE)                              HHL03040
```

```
C     READ IN DESIRED COSINE, COMPUTE ANGLE, THEN SINE                  HHL03050
      READ(8,*) SCR                                                     HHL03060
      ANGLE = DACOS(SCR)                                                HHL03070
      SCI = DSIN(ANGLE)                                                 HHL03080
      SC = DCMPLX(SCR,SCI)                                              HHL03090
      WRITE(6,100) SC                                                   HHL03100
C     IF (IVEC.NE.0.AND.MTYPE.EQ.0) WRITE(9,7) SC                       HHL03110
  100 FORMAT(' GENERATOR OF DIAGONAL TRANSFORMATION ='/2E20.12)         HHL03120
C                                                                       HHL03130
      TC = SC                                                           HHL03140
      DO 110 J=2,KX                                                     HHL03150
  110 TC = SC*TC                                                        HHL03160
      WRITE(6,120) TC                                                   HHL03170
  120 FORMAT(' TC = ',2E20.12)                                          HHL03180
C                                                                       HHL03190
      N = KX*KY                                                         HHL03200
      C2 = ONE                                                          HHL03210
      C1 = HALF-C0                                                      HHL03220
      TEMP = DSQRT(2.0D0)                                               HHL03230
      IF (MTYPE.EQ.0) TEMP = ONE                                        HHL03240
      CL0 = -SC*C0                                                      HHL03250
      CL1 = -TC*C1                                                      HHL03260
      CL3 = -SC*C0*TEMP                                                 HHL03270
      CL4 = -TC*C1*TEMP                                                 HHL03280
C                                                                       HHL03290
      WRITE(6,130) N,MTYPE,KX,KY,C2,C0,C1                               HHL03300
  130 FORMAT(/5X,'N',1X,'MTYPE',4X,'KX',4X,'KY',7X,'DIAGONAL',          HHL03310
     1 3X,'X-CODIAGONAL',3X,'Y-CODIAGONAL'/4I6,3E15.8/)                 HHL03320
C                                                                       HHL03330
C-----------------------------------------------------------------------HHL03340
       CALL HMATVE(C2,CL0,CL1,CL3,CL4,KX,KY)                            HHL03350
C-----------------------------------------------------------------------HHL03360
C                                                                       HHL03370
      IF(IVEC.EQ.0.OR.MTYPE.NE.0) GO TO 140                             HHL03380
C                                                                       HHL03390
C     COMPUTE THE EXACT EIGENVALUES                                     HHL03400
C                                                                       HHL03410
C-----------------------------------------------------------------------HHL03420
      CALL EXEVG(EIGVAL,C0,C1,C2,GAPS,MULTS,KX,KY)                      HHL03430
C-----------------------------------------------------------------------HHL03440
C                                                                       HHL03450
      IF(IVEC.LT.0) GO TO 150                                           HHL03460
C                                                                       HHL03470
  140 CONTINUE                                                          HHL03480
      RETURN                                                            HHL03490
C                                                                       HHL03500
C-----END OF USPEC------------------------------------------------------HHL03510
  150 STOP                                                              HHL03520
      END                                                               HHL03530
C                                                                       HHL03540
C-----START OF CMATV FOR HERMITIAN POISSON MATRICES---------------------HHL03550
C                                                                       HHL03560
      SUBROUTINE HMATV(W,U,SUM)                                         HHL03570
C     SUBROUTINE CMATV(W,U,SUM)                                         HHL03580
C                                                                       HHL03590
```

```
C-----------------------------------------------------------------------
      DOUBLE PRECISION  C2,SUM
      COMPLEX*16 U(1),W(1)
      COMPLEX*16 CL0,CL1,CL3,CL4,CR0,CR1,CR3,CR4
C-----------------------------------------------------------------------
C     CALCULATES U = A*W - SUM*U
C
      GO TO 3
C
      ENTRY HMATVE(C2,CL0,CL1,CL3,CL4,KK,LL)
C
      GO TO 4
C
    3 CONTINUE
C
      N = KK*LL
      CR0 = DCONJG(CL0)
      CR1 = DCONJG(CL1)
      CR3 = DCONJG(CL3)
      CR4 = DCONJG(CL4)
C
C-----------------------------------------------------------------------
C    FIRST AND LAST BLOCKS
      J = 1
      U(J)=(C2*W(J)+CR3*W(J+1)+CR1*W(J+KK)) - SUM*U(J)
      J = 2
      U(J)=(C2*W(J)+CL3*W(J-1)+CR0*W(J+1)+CR1*W(J+KK))-SUM*U(J)
      J = KK
      U(J)=(C2*W(J)+CL3*W(J-1)+CR1*W(J+KK))-SUM*U(J)
      J = KK - 1
      U(J)=(C2*W(J)+CR3*W(J+1)+CL0*W(J-1)+CR1*W(J+KK))-SUM*U(J)
      J = N - KK + 1
      U(J)=(C2*W(J)+CR3*W(J+1)+CL4*W(J-KK))-SUM*U(J)
      J = N - KK + 2
      U(J)=(C2*W(J)+CL3*W(J-1)+CR0*W(J+1)+CL4*W(J-KK))-SUM*U(J)
      J = N
      U(J)=(C2*W(J)+CL3*W(J-1)+CL4*W(J-KK))-SUM*U(J)
      J = N - 1
      U(J)=(C2*W(J)+CL0*W(J-1)+CR3*W(J+1)+CL4*W(J-KK))-SUM*U(J)
C
      KK2 = KK - 2
      DO 10 JJ = 3,KK2
      J = JJ
      U(J)=(C2*W(J)+CL0*W(J-1)+CR0*W(J+1)+CR1*W(J+KK))-SUM*U(J)
      J = N - KK + JJ
   10 U(J)=(C2*W(J)+CL0*W(J-1)+CR0*W(J+1)+CL4*W(J-KK))-SUM*U(J)
C
C    START BLOCKS 2 AND LL-1
      J = KK + 1
      U(J)=(C2*W(J)+CR3*W(J+1)+CL1*W(J-KK)+CR1*W(J+KK))-SUM*U(J)
      J = KK + 2
      U(J)=(C2*W(J)+CL3*W(J-1)+CR0*W(J+1)+CL1*W(J-KK)+CR1*W(J+KK))
     1 -SUM*U(J)
      J = KK + KK
      U(J)=(C2*W(J)+CL3*W(J-1)+CL1*W(J-KK)+CR1*W(J+KK))-SUM*U(J)
```

```
      J = KK + KK - 1
      U(J)=(C2*W(J)+CR3*W(J+1)+CL0*W(J-1)+CL1*W(J-KK)+CR1*W(J+KK))
     1  -SUM*U(J)
      J = N - 2*KK + 1
      U(J)=(C2*W(J)+CR3*W(J+1)+CR4*W(J+KK)+CL1*W(J-KK))
     1 -SUM*U(J)
      J = N - 2*KK + 2
      U(J)=(C2*W(J)+CL3*W(J-1)+CR0*W(J+1)+CR4*W(J+KK)+CL1*W(J-KK))
     1 -SUM*U(J)
      J = N - KK
      U(J)=(C2*W(J)+CL3*W(J-1)+CR4*W(J+KK)+CL1*W(J-KK))-SUM*U(J)
      J = N - KK - 1
      U(J)=(C2*W(J)+CR3*W(J+1)+CL0*W(J-1)+CR4*W(J+KK)+CL1*W(J-KK))
     1 -SUM*U(J)
C
      DO 20 JJ = 3,KK2
      J = KK + JJ
      U(J)=(C2*W(J)+CL0*W(J-1)+CR0*W(J+1)+CL1*W(J-KK)+CR1*W(J+KK))
     1 -SUM*U(J)
      J = N - 2*KK + JJ
      U(J)=(C2*W(J)+CL0*W(J-1)+CR0*W(J+1)+CR4*W(J+KK)+CL1*W(J-KK))
     1 -SUM*U(J)
   20 CONTINUE
C
C    MIDDLE BLOCKS
      LL2 = LL - 2
      JP = KK
      DO 40 JJ = 3,LL2
      JP = JP + KK
C     JP = (JJ-1)*KK
      J = JP + 1
      U(J)=(C2*W(J)+CR3*W(J+1)+CL1*W(J-KK)+CR1*W(J+KK))-SUM*U(J)
      J = J + 1
      U(J)=(C2*W(J)+CL3*W(J-1)+CR0*W(J+1)+CL1*W(J-KK)+
     1 CR1*W(J+KK))-SUM*U(J)
      J = J + KK - 2
      U(J) = (C2*W(J)+CL3*W(J-1)+CL1*W(J-KK)+CR1*W(J+KK))-SUM*U(J)
      J = J - 1
      U(J)=(C2*W(J)+CR3*W(J+1)+CL0*W(J-1)+CL1*W(J-KK)+
     1 CR1*W(J+KK))-SUM*U(J)
C
      DO 30 II = 3,KK2
      J = JP + II
      U(J)=(C2*W(J)+CL0*W(J-1)+CR0*W(J+1)+CL1*W(J-KK)+CR1*W(J+KK))
     1 -SUM*U(J)
   30 CONTINUE
C
   40 CONTINUE
C
    4 RETURN
C
C-----END OF HMATV-----------------------------------------------------
      END
C
C-----START OF EXEVG---------------------------------------------------
```

```
C
C     FOR MTYPE = 0, ZERO BOUNDARY CONDITIONS:
C     COMPUTES EXACT EIGENVALUES OF HERMITIAN POISSON MATRIX,
C     THEIR MULTIPLICITIES, AND THE GAPS BETWEEN THE EIGENVALUES AND
C     PUTS THEM RESPECTIVELY INTO VECTORS U, MP, AND G.  THESE
C     QUANTITIES ARE ALL WRITTEN TO FILE 9.
C
      SUBROUTINE EXEVG(U,C0,C1,C2,G,MP,KX,KY)
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION  U(*),MACHEP
      DOUBLE PRECISION  EPSM,C0,C1,C2,T0,T1,PIK,PIL,ONE,TWO,ATOLN,EE
      REAL G(1)
      INTEGER MP(1)
C-----------------------------------------------------------------------
      DATA MACHEP/Z3410000000000000/
      EPSM = 2.0D0*MACHEP
C-----------------------------------------------------------------------
      N = KX*KY
      ONE  = 1.0D0
      TWO  = 2.0D0
      T0 = DACOS(-ONE)
      T1 = DFLOAT(KX+1)
      PIK = T0/T1
      T1 = DFLOAT(KY+1)
      PIL = T0/T1
C     GENERATE EXACT EIGENVALUES
      KP = 0
      DO 20 J = 1,KY
      T1 = PIL*DFLOAT(J)
      T0 = C2 - TWO*C1*DCOS(T1)
      DO 10 I = 1,KX
      KP = KP+1
      T1 = PIK*DFLOAT(I)
   10 U(KP) = T0 - TWO*C0*DCOS(T1)
   20 CONTINUE
C
C     ORDER U VECTOR BY INCREASING ALGEBRAIC SIZE
      DO 40 K = 2,N
      KM1 = K-1
      DO 30 L = 1,KM1
      JJ = K-L
      IF (U(JJ+1).GE.U(JJ)) GO TO 40
      T0 = U(JJ)
      U(JJ) = U(JJ+1)
   30 U(JJ+1) = T0
   40 CONTINUE
      ATOLN = DMAX1(DABS(U(1)),DABS(U(N)))*EPSM
C
      WRITE(9,50)
   50 FORMAT(' TRUE EIGENVALUES FOR HERMITIAN POISSON')
C
      WRITE(9,60)N,KX,KY,C2,C0,C1,ATOLN
      WRITE(6,60) N,KX,KY,C2,C0,C1,ATOLN
   60 FORMAT(1X,'A-SIZE',2X,'X-DIM',2X,'Y-DIM'/3I7/
```

```
     1 5X,'A-DIAGONAL',3X,'X-CODIAGONAL',3X,'Y-CODIAGONAL',10X,'ATOLN'/
     2 4E15.8)
C
C      DETERMINE TRUE MULTIPLICITIES FOR EXACT EIGENVALUES
      I = 1
      IDEX = 1
      J = 1
      NEXACT = 0
   70 J = J+1
      IF (J.GT.N) GO TO 80
      EE = DABS(U(J)-U(I))
      IF (EE.GT.ATOLN) GO TO 80
      IDEX = IDEX+1
      GO TO 70
   80 NEXACT = NEXACT+1
      U(NEXACT) = U(I)
      MP(NEXACT) = IDEX
C     MP(K) = MULTIPLICITY OF KTH EIGENVALUE CLUSTER FOR A
      IDEX = 1
      I = J
      IF (I.GT.N) GO TO 90
      GO TO 70
   90 CONTINUE
C
C     MULTIPLICITIES HAVE BEEN DETERMINED
C     NEXACT = NUMBER OF DISTINCT A-EIGENVALUES
C
      WRITE(9,100)NEXACT
      WRITE(6,100)NEXACT
  100 FORMAT(I6,' = NUMBER OF TRUE A-EIGENVALUES WHICH ARE DISTINCT'/)
C
C     MINGAP CALCULATION FOR DISTINCT A-EIGENVALUES
      NM1 = NEXACT - 1
      G(NEXACT) = U(NM1)-U(NEXACT)
      G(1) = U(2)-U(1)
C
      DO 110 J = 2,NM1
      T0 = U(J)-U(J-1)
      T1 = U(J+1)-U(J)
      G(J) = T1
      IF (T0.LT.T1) G(J) = -T0
  110 CONTINUE
C
C     NEXACT DISTINCT A-EIGENVALUES ARE IN U IN ASCENDING ORDER
C     MP = MULTIPLICITIES OF THE DISTINCT EIGENVALUES OF A
C     G = TRUE MINIMUM GAP IN A FOR EACH OF THESE EIGENVALUES
C     G < 0 INDICATES THE LEFT-HAND GAP WAS MINIMAL.
C     OUTPUT MULTIPLICITIES, DISTINCT EVS, AND MINGAPS TO FILE 11
C
      WRITE(9,120)
  120 FORMAT(5X,'I',1X,'AMULT',5X,'TRUE A-EIGENVALUE(I)',
     1 3X,'A-MINGAP(I)')
C
      WRITE(9,130)(J,MP(J),U(J),G(J), J=1,NEXACT)
  130 FORMAT(2I6,E25.16,E14.3)
```

```
C
      WRITE(9,140)
  140 FORMAT(' NEXACT DISTINCT A-EIGENVALUES ARE IN ASCENDING ORDER'/
     1 ' AMULT = MULTIPLICITIES OF THE DISTINCT EIGENVALUES OF A.'/
     2 ' A-MINGAP(I) = TRUE MINIMUM GAP IN A FOR EACH EIGENVALUE.'/
     3 ' A-MINGAP(I) LT 0 INDICATES THE LEFT-HAND GAP WAS MINIMAL.'//)
C
C     WE ORDER U VECTOR BY INCREASING SIZE OF THE GAPS
C
      DO 150 K = 1,N
  150 MP(K) = K
C
      DO 170 K = 2,N
      KM1 = K-1
C
      DO 160 L = 1,KM1
      JJ = K - L
      IF (ABS(G(JJ+1)).GE.ABS(G(JJ))) GO TO 170
      EE = U(JJ)
      U(JJ) = U(JJ+1)
      U(JJ+1) = EE
      GG = G(JJ)
      G(JJ) = G(JJ+1)
      G(JJ+1) = GG
      IEE = MP(JJ)
      MP(JJ) = MP(JJ+1)
  160 MP(JJ+1) = IEE
C
  170 CONTINUE
C
      WRITE(9,180)
  180 FORMAT(5X,'K',6X,'A-MINGAP',5X,'TRUE A-EIGENVALUE(I)',2X,'A-EVNO')
C
      WRITE(9,190)(J,G(J),U(J),MP(J), J=1,NEXACT)
  190 FORMAT(I6,E14.3,E25.16,I8)
C
      WRITE(9,200)
  200 FORMAT(' NEXACT DISTINCT A-EIGENVALUES. GAPS IN ASCENDING ORDER'/
     2 ' A-MINGAP(I) = TRUE MINIMUM GAP IN A FOR EACH EIGENVALUE.'/
     3 ' A-MINGAP(I) LT 0 INDICATES THE LEFT-HAND GAP WAS MINIMAL.'/
     3 ' A-MATRIX IS BLOCK TRIDIAGONAL AND EACH DIAGONAL BLOCK IS OF ORD
     3ER NX.'/
     4 ' NX = NUMBER OF POINTS ON EACH X-LINE. THERE ARE NY DIAGONAL BLO
     4CKS.'/
     5 ' NY = NUMBER OF POINTS ON EACH Y-LINE.'/
     5 ' A-DIAGONAL   = A(K,K)'/
     6 ' X-CODIAGONAL = A(I,I+1)'/
     7 ' Y-CODIAGONAL = A(I,I+NX)'/
     8 ' ----- END OF FILE 9 EXACTEV----------------------------'//)
C
C-----END OF EXEVG-----------------------------------------------------
C
      RETURN
      END
C
```

```
C-----START OF HEXVEC---------------------------------------------------
C
C     FOR THE HERMITIAN POISSON TEST CASES WITH MTYPE = 0 ONLY:
C     FOR A GIVEN RITZ VECTOR V AND EIGENVALUE X1, COMPUTES
C     THE CLOSEST TRUE EIGENVALUE Y1 AND CORRESPONDING TRUE
C     EIGENVECTOR Z, CALCULATES THE NORM OF V-Z AND THE MAXIMAL
C     DIFFERENCE OF THE COMPONENTS.  USER WOULD HAVE TO
C     INCORPORATE ENTRY AND CALL TO THIS SUBROUTINE INTO
C     HLEVEC PROGRAM IF THESE QUANTITIES ARE DESIRED.
C     U CONTAINS THE COMPUTED TRUE EIGENVALUES.
C     W CONTAINS THE TRUE EIGENVECTOR FOR THE REAL POISSON MATRIX
C
      SUBROUTINE HEXVEC(Z,V,U,W,X1,Y1,MP,JNUM)
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION  U(*),W(*)
      DOUBLE PRECISION WI(110),WJ(110),WII(110)
      DOUBLE PRECISION X1,Y1,EV,EE,WS,PIK,PIL,SUM,TEMP
      DOUBLE PRECISION ATOLN,EPSM,ZERO,HALF,ONE,TWO,MACHEP
      DOUBLE PRECISION C0,C1,C2,T0,T1,T2
      COMPLEX*16 CONE,S,SB,STEMP,V(1),Z(1)
      INTEGER MP(1)
C-----------------------------------------------------------------------
      DATA MACHEP/Z3410000000000000/
      EPSM = 2.0D0*MACHEP
C-----------------------------------------------------------------------
C     THIS PROGRAM CALCULATES THE EXACT EIGENVALUES AND EIGENVECTORS
C     OF THE HERMITIAN POISSON MATRIX A OF ORDER  N = KX BY KY
C     A CONSISTS OF KY TRIDIAGONAL BLOCKS OF ORDER KX
C     KX = X-DIMENSION     KY = Y-DIMENSION.
C
C     C2 = DIAGONAL OF KX BY KX MATRIX
C    -C0 = C0-DIAGONAL OF THE KX BY KX MATRIX.
C    -C1 = Y-CODIAGONAL.
C
C     NOTE THAT THE VECTORS WI,WJ,WII ARE DIMENSIONED INTERNALLY
C     THEY ARE USED JUST TO KEEP FROM REGENERATING INFORMATION.
C     WI,WII = REAL*8 ARRAYS OF DIMENSION AT LEAST KX
C     WJ     = REAL*8 ARRAY  OF DIMENSION AT LEAST KY.
C
C     NOTATION USED IN PROGRAM
C
C     PIK = ARCOS(-1)/(KX+1)    PIL = ARCOS(-1)/(KY+1)
C     WI(I) = PIK*I        WJ(J) = PIL*J
C
C     T0 = C2 - 2*C1*COS(PIL*J)    EV(I,J) = T0 - 2*C0*COS(PIK*I)
C     I = 1,KX    J = 1,KY    KP = (J-1)*KX + I
C
C     W(KV) = SIN(PIK*I*IK)*SIN(PIL*J*JK)
C     IK = 1,KX    JK = 1,KY    KV = (JK-1)*KX + IK
C     W IS UNSCALED EIGENVECTOR FOR EV(I,J)
C     WS = 1/||W||: ||W|| = .5*DSQRT(T2*T3)  T2 = KX+1  T3 = KY+1
C     U(K) IS A-EV ORDERED BY INCREASING SIZE, K = 1,N
C
C     GIVEN X1 FIND Y1 AND KVEC SUCH THAT
```

```
C           Y1 = EV(KVEC) AND |X1-Y1| = MIN
C           ALSO GIVEN UNIT RITZ VECTOR ASSOCIATED WITH X1
C           CALCULATE UNIT EIGENVECTOR W, A*W = Y1*W
C           T2 = ||V-W||  T1 = MAX(|V(K)-W(K)|, K= 1,N)
C           MAX OCCURS FIRST AT K = KK
C
C-----------------------------------------------------------------------
C     C2 = A(K,K)
C     C0 = A(K,K+1) = A(K+1,K)
C     C1 = A(K,K+KX) = A(K+KX,K)
C     C0 + C1 = HALF
C
      GO TO 3
C
C-----------------------------------------------------------------------
      ENTRY EXVECP(SB,C0,C1,C2,KX,KY)
C-----------------------------------------------------------------------
      GO TO 4
C
    3 CONTINUE
C
C     SPECIFY PARAMETERS
      N = KX*KY
      ZERO = 0.0D0
      HALF = 0.5D0
      ONE  = 1.0D0
      TWO  = 2.0D0
      T0 = DACOS(-ONE)
      T1 = DFLOAT(KX+1)
      PIK = T0/T1
      T2 = DFLOAT(KY+1)
      PIL = T0/T2
      WS = TWO/DSQRT(T1*T2)
C
C     GENERATE WI WJ VECTORS
      KP = 0
      DO 20 J = 1,KY
      T1 = PIL*DFLOAT(J)
      WJ(J) = T1
      T0 = C2 - TWO*C1*DCOS(T1)
      DO 10 I = 1,KX
      KP = KP+1
      T1 = PIK*DFLOAT(I)
      WI(I) = T1
   10 U(KP) = T0 - TWO*C0*DCOS(T1)
   20 CONTINUE
C     U(KP) = EV(I,J) = C2 - 2*C1*COS(PIL*J) - 2*C0*COS(PIK*I)
C
C     INITIALIZE MP VECTOR
      DO 30 K = 1,N
   30 MP(K) = K
C
C     WE ORDER U VECTOR BY INCREASING SIZE OF THE EVS
      DO 50 K = 2,N
      KM1 = K-1
```

```
C                                                                       HHL07450
      DO 40 L = 1,KM1                                                   HHL07460
      JJ = K - L                                                        HHL07470
      IF (U(JJ+1).GE.U(JJ)) GO TO 50                                    HHL07480
      EE = U(JJ)                                                        HHL07490
      U(JJ) = U(JJ+1)                                                   HHL07500
      U(JJ+1) = EE                                                      HHL07510
      IEE = MP(JJ)                                                      HHL07520
      MP(JJ) = MP(JJ+1)                                                 HHL07530
   40 MP(JJ+1) = IEE                                                    HHL07540
C                                                                       HHL07550
   50 CONTINUE                                                          HHL07560
C                                                                       HHL07570
      ATOLN = DMAX1(DABS(U(1)),DABS(U(N)))*EPSM                         HHL07580
C                                                                       HHL07590
      WRITE(6,60) N,KX,KY,C2,C0,C1,ATOLN                                HHL07600
   60 FORMAT(/' EXACT ERRORS FOR CONVERGED GOODEV'/                     HHL07610
     1 4I6,' = N KX KY'//                                               HHL07620
     1 4E12.5,' = C2 C0 C1 ATOLN'//)                                    HHL07630
C                                                                       HHL07640
C     KP = MP(K) MEANS EIGENVALUE U(K) CORRESPONDS TO EIGENVECTOR W(KP) HHL07650
C     COMPUTE TOLERANCE USED IN COMPUTING TRUE MULTIPLICITIES           HHL07660
C                                                                       HHL07670
C     X1 IS AN INPUT PARAMETER. WE CALCULATE EXACT                      HHL07680
C     A-EIGENVALUE WHICH IS CLOSEST TO X1, LABEL IT Y1 AND CALCULATE    HHL07690
C     UNIT EIGENVECTOR OF A ASSOCIATED WITH Y1. A*W = Y1*W, ||W|| = 1.  HHL07700
C     Y1 = U(KEV). EIGENVALUES OF A ARE ORDERED BY INCREASING SIZE.     HHL07710
C     V = COMPLEX RITZ VECTOR ASSOCIATED WITH GOODEV X1                 HHL07720
C     WE SHOULD HAVE V = D*W WHERE D = DIAG(D(1),D(2),..,D(N))          HHL07730
C     D(1) = ONE, D(K+1)/D(K) = SB, |SB| = ONE                          HHL07740
C                                                                       HHL07750
      KX1 = 0                                                           HHL07760
      IF (X1.LE.U(1)) KX1 = 1                                           HHL07770
      IF (X1.GE.U(N)) KX1 = N                                           HHL07780
      NM1 = N-1                                                         HHL07790
      IF (KX1.NE.0) GO TO 80                                            HHL07800
C                                                                       HHL07810
      DO 70 KVEC = 2,N                                                  HHL07820
      IF (X1.GE.U(KVEC)) GO TO 70                                       HHL07830
C     U(KVEC-1).LE.X1.LT.U(KVEC)                                        HHL07840
      T1 = X1 - U(KVEC-1)                                               HHL07850
      T2 = U(KVEC) - X1                                                 HHL07860
      KX1 = KVEC - 1                                                    HHL07870
      IF (T1.GT.T2) KX1 = KVEC                                          HHL07880
      GO TO 80                                                          HHL07890
   70 CONTINUE                                                          HHL07900
C                                                                       HHL07910
   80 Y1 = U(KX1)                                                       HHL07920
C                                                                       HHL07930
      IF (KX1.EQ.1) EE = U(2) - U(1)                                    HHL07940
      IF (KX1.EQ.N) EE = U(N) - U(NM1)                                  HHL07950
      IF (KX1.EQ.1.OR.KX1.EQ.N) GO TO 90                                HHL07960
      EE = DMIN1(U(KX1+1)-U(KX1),U(KX1)-U(KX1-1))                       HHL07970
   90 CONTINUE                                                          HHL07980
C                                                                       HHL07990
```

```
      T0 = DABS(ONE - X1/Y1)                                            HHL08000
C                                                                       HHL08010
      WRITE(6,100) N,KX1,JNUM,Y1,X1,T0,EE                               HHL08020
  100 FORMAT(3I8,' = N, A-EV NUMBER,GOODEV NO'//                        HHL08030
     1 18X,'EXACTEV',19X,'GOODEV',4X,'RELERROR',4X,'A-MINGAP'/          HHL08040
     1 2E25.16,2E12.3/)                                                 HHL08050
C                                                                       HHL08060
      IF (EE.GT.ATOLN) GO TO 120                                        HHL08070
C                                                                       HHL08080
      WRITE(6,110)                                                      HHL08090
  110 FORMAT(' Y1 IS A MULTIPLE EIGENVALUE OF A SO WE EXIT'/)           HHL08100
C                                                                       HHL08110
      GO TO 200                                                         HHL08120
C                                                                       HHL08130
C     Y1 IS TOEPLITZ EIGENVALUE CLOSEST TO X1.                          HHL08140
C     CALCULATION OF EIGENVECTOR ASSOCIATED WITH EIGENVALUE Y1          HHL08150
C     A*W = Y1*W                                                        HHL08160
C                                                                       HHL08170
C     DETERMINE I J FROM K: MP(K) = KP = (J-1)*KX+I                     HHL08180
  120 CONTINUE                                                          HHL08190
      K = KX1                                                           HHL08200
      KP = MP(K)                                                        HHL08210
      I = MOD(KP,KX)                                                    HHL08220
      IF (I.EQ.0) I = KX                                                HHL08230
      T1 = WI(I)                                                        HHL08240
      J = 1 + (KP-1)/KX                                                 HHL08250
      T2 = WJ(J)                                                        HHL08260
C                                                                       HHL08270
      DO 130 II = 1,KX                                                  HHL08280
      T0 = T1*DFLOAT(II)                                                HHL08290
  130 WII(II) = WS*DSIN(T0)                                             HHL08300
C                                                                       HHL08310
      KV = 0                                                            HHL08320
      DO 150 JJ = 1,KY                                                  HHL08330
      T0 = T2*DFLOAT(JJ)                                                HHL08340
      T0 = DSIN(T0)                                                     HHL08350
C                                                                       HHL08360
      DO 140 II = 1,KX                                                  HHL08370
      KV = KV + 1                                                       HHL08380
  140 W(KV) = T0*WII(II)                                                HHL08390
C                                                                       HHL08400
  150 CONTINUE                                                          HHL08410
C                                                                       HHL08420
C     W IS UNIT EXACT EIGENVECTOR OF A ASSOCIATED WITH Y1               HHL08430
C     V IS UNIT COMPLEX RITZVECTOR OF B ASSOCIATED WITH X1              HHL08440
C                                                                       HHL08450
      CONE = DCMPLX(ONE,ZERO)                                           HHL08460
      STEMP = CONE                                                      HHL08470
      DO 160 K = 1,N                                                    HHL08480
      Z(K) = STEMP*W(K)                                                 HHL08490
  160 STEMP = STEMP*SB                                                  HHL08500
C                                                                       HHL08510
      T1 = ZERO                                                         HHL08520
      S = ONE                                                           HHL08530
      KK = 0                                                            HHL08540
```

```
      DO 170 K = 1,N
      IF (CDABS(Z(K)).LE.T1) GO TO 170
      T1 = CDABS(Z(K))
      KK = K
  170 CONTINUE
C
      S = V(KK)/Z(KK)
C
      KK = 0
      T1 = ZERO
      T2 = ZERO
      DO 180 K = 1,N
      TEMP = CDABS(S*Z(K) - V(K))
      T2 = T2 + TEMP**2
      IF (TEMP.LE.T1) GO TO 180
      KK = K
      T1 = TEMP
  180 CONTINUE
C
      T2 = DSQRT(T2)
      WRITE(6,190) KK,T1,T2
  190 FORMAT(' EIGENVECTOR ERROR. MAX ERROR AT COMPONENT = ',I6/
     1 ' MAX CDABS(EXACTVEC(K)-RITZVEC(K)) = ',E12.5/
     1 ' NORM(EXACTVEC-RITZVEC) = ',E12.5/)
C
  200 CONTINUE
C
C-----END OF HEXVEC----------------------------------------------------
    4 RETURN
      END
C
C-----USPEC (TRIDIAGONAL HERMITIAN MATRICES)---------------------------
C
C     SUBROUTINE USPEC(N,MATNO)
      SUBROUTINE TSPEC(N,MATNO)
C
C---------------------------------------------------------------------
      DOUBLE PRECISION  D(100), DAR(100),DAI(100), PI, EIGVAL(100)
      DOUBLE PRECISION SPACE
      COMPLEX*16   DA(100),DB(100)
      REAL EXPLAN(20)
C---------------------------------------------------------------------
C    DIMENSION ARRAYS NEEDED TO DEFINE MATRIX.  THEN
C    PASS THE STORAGE LOCATIONS OF THESE ARRAYS TO THE MATRIX-VECTOR
C    MULTIPLY SUBROUTINE CMATV.
C
C    DIAGONAL ENTRY = D, ABOVE DIAGONAL ENTRY = DA, BELOW DIAGONAL = DB.
C
      READ(8,10) EXPLAN
   10 FORMAT(20A4)
      READ(8,*) NOLD,MATOLD
C
      WRITE(6,20) N,MATOLD
   20 FORMAT(I10,2I6,I8,' = N,MATOLD'/)
C
```

```
C     TEST OF PARAMETER CORRECTNESS
      ITEMP = (NOLD-N)**2 + (MATNO-MATOLD)**2
C
      IF(ITEMP.EQ.0) GO TO 40
C
      WRITE(6,30)
   30 FORMAT(' PROGRAM TERMINATES BECAUSE EITHER ORDERS OF OR LABELS FOR
     1 MATRIX DISAGREE')
      GO TO 250
C
   40 CONTINUE
C
C     IF ITOEP = 1 THEN MATRIX IS TOEPLITZ AND WE PRINT OUT TRUE
C     EIGENVALUES
      READ(8,10) EXPLAN
      READ(8,*) ITOEP
      READ(8,10) EXPLAN
C
      IF(ITOEP.EQ.1) WRITE(6,50)
   50 FORMAT(/' TEST MATRIX IS HERMITIAN TOEPLITZ'/)
      IF(ITOEP.NE.1) GO TO 110
C
      READ(8,*) DAR(1),DAI(1),D(1)
      DA(1) = DCMPLX(DAR(1),DAI(1))
      DB(1) = DCONJG(DA(1))
      DO 60 J=2,N
      D(J) = D(1)
      DA(J) = DA(1)
   60 DB(J) = DB(1)
      WRITE(6,70) DB(1),D(1),DA(1)
      WRITE(9,70) DB(1),D(1),DA(1)
   70 FORMAT(' HERMITIAN TOEPLITZ MATRIX IS USED.'/' BELOW DIAGONAL ENTR
     1Y = ',2E12.3/' DIAGONAL ENTRY = ',E12.3/' ABOVE DIAGONAL ENTRY ='
     1,2E12.3)
C
C     COMPUTE THE TRUE EIGENVALUES.  FORMULA IS CORRECT ONLY FOR THOSE
C     MATRICES WHOSE DIAGONAL = 2., ABOVE DIAGONAL = A, BELOW DIAGONAL
C     = A-CONJUGATE, AND A HAS NORM 1.
C
      PI = DACOS(-1.D0)
      DO 80 J=1,N
   80 EIGVAL(J) = 2.D0 * (1.D0 -DCOS(PI*DFLOAT(J)/DFLOAT(N+1)))
      WRITE(9,90) N
   90 FORMAT(I6, ' = ORDER OF MATRIX'/' TRUE EIGENVALUES ARE'/)
      WRITE(9,100) (J, EIGVAL(J), J=1,N)
  100 FORMAT(I5,4X,E25.16,6X,I5,4X,E25.16)
      GO TO 240
C
C     NONTOEPLITZ HERMITIAN.  DIAGONAL ENTRIES ARE EQUALLY-SPACED.
C     ABOVE DIAGONAL ENTRIES ARE GENERATED BY GENERATING EQUALLY-SPACED
C     REAL PARTS, AND EQUALLY-SPACED IMAGINARY PARTS.  THE BELOW
C     DIAGONAL ENTRIES ARE THEN OBTAINED BY TAKING THE COMPLEX CONJUGATE
C     OF THE ABOVE DIAGONAL ENTRIES
C
  110 READ(8,*) D(1), SPACE
```

```
      WRITE(6,120) D(1),SPACE
  120 FORMAT(' 1ST DIAGONAL ENTRY =',E20.12,' SPACING =',E20.12)
      DO 130 J=2,N
  130 D(J) = D(J-1) + SPACE
      WRITE(6,140) (D(J), J=1,3)
  140 FORMAT(' 1ST THREE DIAGONAL ENTRIES ='/(2E20.12))
      READ(8,10) EXPLAN
      READ(8,*) DAR(1), SPACE
      WRITE(6,150) DAR(1),SPACE
  150 FORMAT(' REAL PART OF 1ST ABOVE DIAGONAL ENTRY =',E20.12,/
     1' SPACING = ',E20.12)
      DO 160 J=2,N
  160 DAR(J) = DAR(J-1) + SPACE
      WRITE(6,170) (DAR(J), J=1,3)
  170 FORMAT(' REAL PARTS OF 1ST THREE ABOVE DIAGONAL ENTRIES ='/
     1(2E20.12))
      READ(8,10) EXPLAN
      READ(8,*) DAI(1), SPACE
      WRITE(6,180) DAI(1),SPACE
  180 FORMAT(' IMAGINARY PART OF 1ST ABOVE =',E20.12,/' SPACING =',
     1 E20.12)
      DO 190 J=2,N
  190 DAI(J) = DAI(J-1) + SPACE
      WRITE(6,200) (DAI(J), J = 1,3)
  200 FORMAT(' IMAGINARY PARTS OF 1ST THREE ABOVE DIAGONAL ENTRIES ='/
     1 (2E20.12))
      DO 210 J=1,N
      DA(J) = DCMPLX(DAR(J),DAI(J))
  210 DB(J) = DCONJG(DA(J))
C
      WRITE(9,220) (D(J), J=1,N)
  220 FORMAT(' DIAGONAL ENTRIES ='/(4E20.12))
      WRITE(9,230) (DA(J), J=1,N)
  230 FORMAT(' ABOVE DIAGONAL ENTRIES'/(4E20.12))
C
C     PASS STORAGE LOCATIONS OF ARRAYS THAT DEFINE THE MATRIX TO
C     THE MATRIX-VECTOR MULTIPLY SUBROUTINE CMATV
C
  240 CONTINUE
C
C-----------------------------------------------------------------------
      CALL TMATVE(DA,DB,D,N)
C-----------------------------------------------------------------------
C
      RETURN
  250 STOP
C
C-----END OF USPEC------------------------------------------------------
      END
C
C-----START OF MATRIX-VECTOR MULTIPLY (HERMITIAN TRIDIAGONAL)-----------
C
C     SUBROUTINE CMATV(W,U,SUM)
      SUBROUTINE TMATV(W,U,SUM)
C
```

```
C-----------------------------------------------------------------------
      COMPLEX*16   U(1),W(*),DA(1),DB(1)
      DOUBLE PRECISION  D(1),SUM
C-----------------------------------------------------------------------
C    HERMITIAN MATRIX-VECTOR MULTIPLY FOR LANCZS  U = A*W - SUM*U
C    MATRIX IS TRIDIAGONAL HERMITIAN TOEPLITZ
C-----------------------------------------------------------------------
C
C     COMPUTE A*W - SUM*U
C
      GO TO 3
C-----------------------------------------------------------------------
C    STORAGE LOCATIONS ARE PASSED TO CMATV FROM USPEC
      ENTRY TMATVE(DA,DB,D,N)
      GO TO 4
C-----------------------------------------------------------------------
    3 CONTINUE
C
      U(1) = D(1)*W(1) + DA(1)*W(2) - SUM*U(1)
      N1 = N-1
      DO 10 I = 2,N1
   10 U(I) = DB(I-1)*W(I-1)+D(I)*W(I) + DA(I)*W(I+1) -SUM*U(I)
      U(N) = DB(N-1)*W(N-1) + D(N)*W(N) - SUM*U(N)
C
    4 RETURN
C
C-----END OF CMATV------------------------------------------------------
      END
C-----DUMMY USPEC DOES NOTHING------------------------------------------
C
      SUBROUTINE USPEC(N,MATNO)
C     SUBROUTINE CUSPEC(N,MATNO)
C
C-----------------------------------------------------------------------
      RETURN
      END
```

3.5 HLEVAL: HLEVEC: File Definitions, Sample Input Files

Below is a listing of the input/output files definitions which are accessed by the Hermitian Lanczos eigenvalue program, HLEVAL. Included also is a sample of the input file which HLEVAL requires on file 5. The parameters are supplied in free format. HLEVAL computes eigenvalues of Hermitian matrices A on user-specified intervals which must be supplied in ascending order. File 8 is assumed to contain the data which defines the Hermitian nxn matrix A.

```
Sample Specifications of the input/output files for HLEVAL
----------------------------------------------------------------
 HLEVAL EXEC HERMITIAN EIGENVALUE CALCULATION
FI 06 TERM
FILEDEF  1 DISK &1        NHISTORY  A (RECFM F LRECL 80 BLOCK 80
FILEDEF  2 DISK &1        HISTORY   A (RECFM F LRECL 80 BLOCK 80
FILEDEF  3 DISK &1        GOODEV    A (RECFM F LRECL 80 BLOCK 80
FILEDEF  4 DISK &1        ERRINV    A (RECFM F LRECL 80 BLOCK 80
FILEDEF  5 DISK HLEVAL    INPUT     A (RECFM F LRECL 80 BLOCK 80
FILEDEF  8 DISK &1        INPUT     A (RECFM F LRECL 80 BLOCK 80
FILEDEF 11 DISK &1        DISTINCT  A (RECFM F LRECL 80 BLOCK 80
LOAD   HLEVAL   LESUB   HLEMULT
----------------------------------------------------------------
```

```
Sample Input File for HLEVAL
-----------------------------------------------------------------
 HLEVAL INPUT EIGENVALUE COMPUTATION, NO REORTHOGONALIZATION
 HERMITIAN TEST MATRIX
LINE 1      N    KMAX    NMEVS     MATNO
          528    1600        3    721830
LINE 2   SVSEED    RHSEED      MXINIT    MXSTUR
       49302312   5731029           5    100000
LINE 3   ISTART     ISTOP
              0         1
LINE 4     IHIS     IDIST  IWRITE
              1         0       1
LINE 5   RELTOL (RELATIVE TOLERANCE IN 'COMBINING' GOODEV)
    .0000000001
LINE 6   MB(1)   MB(2)   MB(3)   MB(4)    (ORDERS OF T(1,MEV) )
           528    1056    1584
LINE 7    NINT     (NUMBER OF SUB-INTERVALS FOR BISEC)
             1
LINE 8    LB(1)    LB(2)   LB(3)   LB(4)  (INTERVAL LOWER BOUNDS)
           1.0
LINE 9    UB(1)    UB(2)   UB(3)   UB(4)  (INTERVAL UPPER BOUNDS)
           2.0
-----------------------------------------------------------------
```

Below is a listing of the input/output files definitions which are accessed by the Hermitian Lanczos eigenvector program, HLEVEC. Included also is a sample of the input file which HLEVEC requires on file 5. The parameters are supplied in free format. HLEVEC computes eigenvectors for each of a user-specified subset of the eigenvalues computed by the companion code HLEVEC. Eigenvector approximations will be computed only for eigenvalue approximations which have converged.

```
Sample Specifications of the Input/Output Files for HLEVEC
-------------------------------------------------------------------
 HLEVEC EXEC TO RUN LANCZOS EIGENVECTOR PROGRAM, HERMITIAN MATRICES
FI 06 TERM
FILEDEF  2 DISK &1       HISTORY   A (RECFM F LRECL 80 BLOCK 80
FILEDEF  3 DISK &1       GOODEV    A (RECFM F LRECL 80 BLOCK 80
FILEDEF  4 DISK &1       ERRINV    A (RECFM F LRECL 80 BLOCK 80
FILEDEF  5 DISK HLEVEC   INPUT     A (RECFM F LRECL 80 BLOCK 80
FILEDEF  8 DISK &1       INPUT     A (RECFM F LRECL 80 BLOCK 80
FILEDEF  9 DISK &1       ERREST    A (RECFM F LRECL 80 BLOCK 80
FILEDEF 10 DISK &1       BOUNDS    A (RECFM F LRECL 80 BLOCK 80
FILEDEF 11 DISK &1       TEIGVECS  A (RECFM F LRECL 80 BLOCK 80
FILEDEF 12 DISK &1       RITZVECS  A (RECFM F LRECL 80 BLOCK 80
FILEDEF 13 DISK &1       PAIGE     A (RECFM F LRECL 80 BLOCK 80
LOAD  HLEVEC  LESUB  HLEMULT
-------------------------------------------------------------------
```

```
Sample Input File for HLEVEC
---------------------------------------------------------------------
 HLEVEC EIGENVECTORS OF HERMITIAN MATRIX, NO REORTHOGONALIZATION
LINE 1  MDIMTV    MDIMRV  MBETA(MAX.DIMENSIONS, TVEC, RITVEC AND BETA
         10000     10000   2000
LINE 2       RELTOL
        .0000000001
LINE 3  MBOUND   NTVCON SVTVEC IREAD (FLAGS
             0        1      0     1
LINE 4  TVSTOP   LVCONT ERCONT  IWRITE (FLAGS
             0        1      1       1
LINE 5    RHSEED  (RANDOM GENERATOR SEED FOR STARTING VECTOR IN INVERM
        45329517
LINE 6  MATNO      N
          100    100
----------------------------------------------------------------------
```