Chapter 4

Factored Inverses of Real Symmetric Matrices

4.1 Introduction

The FORTRAN codes in this chapter address the question of computing distinct eigenvalues and corresponding eigenvectors of a real symmetric matrix by applying a single-vector Lanczos procedure to the inverse of an associated matrix $B \equiv PCP^T$ where $C = (SCALE) * A + (SHIFT) * I$. The scalars $SCALE$ and $SHIFT$ are specified by the user, selected in such a way that the resulting matrix C (or B) has a reasonable numerical condition. The permutation matrix P is chosen so that for a sparse matrix A, the resulting factorization of B is also sparse.

For a given real symmetric matrix A, these codes compute real scalars λ and corresponding real-valued vectors $x \neq 0$ such that

$$B^{-1}x = \lambda x, \tag{4.1.1}$$

where B is as defined above. Note that the eigenvectors of B^{-1} are simple permutations of the eigenvectors of A. The eigenvalues of A are obtained from those of B by a simple scalar modification, which is incorporated in the codes. These codes do not require the matrix A. The Lanczos computations use only the user-supplied factorization of the associated matrix B, the scalars $SCALE$ and $SHIFT$, and the permutation P (if any).

Real symmetric matrices and factorizations of such matrices are discussed in Stewart [24]. See also Bunch and Kaufman [2] and George and Liu [10]. Chapter 2, Section 2.1, contains a brief summary of the properties of real symmetric matrices which we use in these codes.

Given a real symmetric matrix A, the user may decide to use the codes in this chapter rather than those in Chapter 2 if the eigenvalues to be computed are 'small' with 'small' gaps between them and the required factorization can be obtained with a reasonable amount of computation and storage. The user should note however that this type of transformation of the given matrix may not yield an eigenvalue distribution which is better for these Lanczos codes. Such a transformation will accelerate the Lanczos computations only if the desired eigenvalues either become larger in size relative to the other eigenvalues and/or the gaps between the desired eigenvalues become larger relative to the gaps between the other eigenvalues. This type of transformation can be very effective in compressing the big end of the spectrum of a given matrix and enhancing the small end of the spectrum. The Lanczos procedure, however, does not require large gaps between the desired eigenvalues, all it really requires is a reasonable overall gap ratio. That is, the ratio of the largest gap between two neighboring eigenvalues to the smallest such gap must be a

reasonable size.

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 real symmetric matrix. The documentation for these codes is contained in Chapter 2, Section 2.2. As in the direct real symmetric case (Chapter 2, Section 2.1), the A-multiplicity of a given computed 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 the basic Lanczos recursion contained in Eqns (1.2.1) and (1.2.2) to generate a family of real symmetric tridiagonal matrices (T-matrices) for the matrix B^{-1} , whose sizes are specified by the user. Specifically, for $i = 1, 2, ..., m$ and a randomly-generated starting vector v_1 with $\|v_1\| = 1$, generate Lanczos vectors v_i using the following recursion and Eqn(1.2.2) applied to the matrix B^{-1} .

$$\beta_{i+1} v_{i+1} = B^{-1} v_i - \alpha_i v_i - \beta_i v_{i-1}. \tag{4.1.2}$$

B is the matrix defined above in terms of the scalars $SCALE$ and $SHIFT$, and the permutation P, and each $B^{-1}v_i$ is evaluated by solving the system of equations $Bz = v_i$.

LIVAL, the main program for the factored inverse computations, calls the subroutine BISEC to compute eigenvalues of the specified Lanczos tridiagonal 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 B^{-1} matrix associated with the user-specified matrix A, scalars $SCALE$ and $SHIFT$, and the permutation matrix P (if any). The accuracy of these 'good' T-eigenvalues as eigenvalues of B^{-1} 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. After each T-matrix eigenvalue computation, the corresponding approximations to the eigenvalues of the user-specified matrix A are computed and included in the output.

Once the eigenvalues of B^{-1} 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 LIVEC, for computing eigenvectors of the inverse of a real symmetric matrix, given a factorization, is used to compute the desired eigenvectors. If the matrix B is a permutation of the matrix C, then LIVEC unwinds the permutation to obtain the corresponding eigenvectors of the user-supplied A-matrix.

All of the computations are done in double precision real arithmetic. Once the Lanczos T-matrices have been computed, the remaining computations use the same subroutines that are used in the real symmetric case discussed in Chapter 2. In addition to the programs and subroutines provided here, the user must supply a subroutine USPEC which defines and initializes the factorization of the scaled, shifted, and permuted version B of the original matrix A, and a subroutine BSOLV which computes matrix-vector multiplies $B^{-1}x$ 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.

The sample subroutines USPEC and BSOLV provided assume that the associated matrix B is positive definite and that its Cholesky factorization

$$B = LL^T, \tag{4.1.3}$$

where L is a lower triangular matrix, is used to compute $B^{-1}y$, for any given y. Thus, the sample USPEC subroutine provided for this chapter defines and initializes arrays which define the Cholesky factor L of the associated matrix B. The sample BSOLV subroutine provided computes the required matrix-vector

multiplies $u = B^{-1}y$ by solving sequentially the two equations $Lz = y$ and $L^T u = z$. These two equations are very easy to solve since L is a triangular matrix. The main portions of these Lanczos codes do not however require that the B-matrix be positive definite, only that a factorization be available. Therefore, the user could replace the sample USPEC and BSOLV subroutines by subroutines which use a more general factorization of B, for example $B = LDL^T$, where D is a diagonal matrix. All that is necessary is that the BSOLV subroutine provide the matrix-vector products $B^{-1}x$, rapidly and accurately. The information supplied to the Lanczos procedures about the matrix being processed must be consistent.

Several optional preprocessing programs are provided, PERMUT, LORDER, LFACT, and LTEST. PERMUT calls the SPARSPAK Library [9] to attempt to identify a reordering or permutation P of the given matrix A for which sparseness will be preserved under factorization of the permuted matrix. LORDER takes a given matrix C and permutation P and computes the sparse matrix format for the permuted matrix, $B \equiv PCP^T$. LFACT computes the Cholesky factors of a given positive definite matrix. LTEST performs a very crude check on the numerical condition of the matrix supplied to it, by solving a system of equations with and without iterative refinement LINPACK [7].

4.2 LIVAL: Main Program, Eigenvalue Computations

```
C-----LIVAL---(EIGENVALUES OF INVERSES OF REAL SYMMETRIC MATRICES)------LIV00010
C  Authors:  Jane Cullum and Ralph A. Willoughby (deceased)              LIV00020
C            Los Alamos National Laboratory                             LIV00030
C            Los Alamos, New Mexico 87544                               LIV00040
C                                                                       LIV00050
C            E-mail:  cullumj@lanl.gov                                  LIV00060
C                                                                       LIV00070
C  These codes are copyrighted by the authors.  These codes             LIV00080
C  and modifications of them or portions of them are NOT to be          LIV00090
C  incorporated into any commercial codes or used for any other         LIV00100
C  commercial purposes such as consulting for other companies,          LIV00110
C  without legal agreements with the authors of these Codes.            LIV00120
C  If these Codes or portions of them are used in other scientific or   LIV00130
C  engineering research works the names of the authors of these codes   LIV00140
C  and appropriate references to their written work are to be           LIV00150
C  incorporated in the derivative works.                                LIV00160
C                                                                       LIV00170
C  This header is not to be removed from these codes.                   LIV00180
C                                                                       LIV00190
C        REFERENCE: Cullum and Willoughby, Chapters 1,2,3,4             LIV00191
C        Lanczos Algorithms for Large Symmetric Eigenvalue ComputationsLIV00192
C        VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in    LIV00193
C        Applied Mathematics, 2002. SIAM Publications,                  LIV00194
C        Philadelphia, PA. USA                                          LIV00195
C                                                                       LIV00200
C     CONTAINS MAIN PROGRAM FOR COMPUTING DISTINCT EIGENVALUES OF       LIV00210
C     INVERSES OF REAL SYMMETRIC MATRICES USING REORDERING              LIV00220
C     AND SPARSE FACTORIZATION.  THE LANCZOS RECURSION IS APPLIED       LIV00230
C     TO A SCALED, SHIFTED, AND REORDERED VERSION B OF THE              LIV00240
C     ORIGINAL A-MATRIX.  THE PROCEDURE USES LANCZOS                    LIV00250
C     TRIDIAGONALIZATION WITHOUT REORTHOGONALIZATION                    LIV00260
C                                                                       LIV00270
C     PFORT VERIFIER IDENITIFIED THE FOLLOWING NONPORTABLE              LIV00280
C     CONSTRUCTIONS                                                     LIV00290
C                                                                       LIV00300
C     1.  DATA/MACHEP/ STATEMENT                                        LIV00310
C     2.  ALL READ(5,*) STATEMENTS (FREE FORMAT)                        LIV00320
C     3.  FORMAT(20A4) USED WITH EXPLANATORY HEADER EXPLAN.             LIV00330
C     4.  HEXADECIMAL FORMAT (4Z20) USED IN ALPHA/BETA FILES 1 AND 2.   LIV00340
C                                                                       LIV00350
C-----------------------------------------------------------------------LIV00360
C                                                                       LIV00370
      DOUBLE PRECISION  ALPHA(3000),BETA(3001)                          LIV00380
      DOUBLE PRECISION  V1(3001),V2(3000),VS(3000)                      LIV00390
      DOUBLE PRECISION  LB(20),UB(20)                                   LIV00400
      DOUBLE PRECISION  BTOL,GAPTOL,TTOL,MACHEP,EPSM,SHIFT,SHIFT0,RELTOLLIV00410
      DOUBLE PRECISION  SCALE1,SCALE2,SCALE3,SCALE4,BISTOL,CONTOL,MULTOLLIV00420
      DOUBLE PRECISION  ONE,ZERO,TEMP,TKMAX,BETAM,BKMIN,T0,T1,S0        LIV00430
      REAL  G(3000),GG(3000),EXPLAN(20)                                 LIV00440
      INTEGER  MP(3000),NMEV(20)                                        LIV00450
      INTEGER  SVSEED,RHSEED,SVSOLD                                     LIV00460
```

```
      INTEGER IABS                                                      LIV00470
      REAL  ABS                                                         LIV00480
      DOUBLE PRECISION  DABS, DSQRT, DFLOAT                             LIV00490
      EXTERNAL BSOLV                                                    LIV00500
C                                                                       LIV00510
C-----------------------------------------------------------------------LIV00520
      DATA MACHEP/Z3410000000000000/                                    LIV00530
      EPSM = 2.0D0*MACHEP                                               LIV00540
C-----------------------------------------------------------------------LIV00550
C                                                                       LIV00560
C     ARRAYS MUST BE DIMENSIONED AS FOLLOWS:                            LIV00570
C     1.  ALPHA: >= KMAX,   BETA: >= (KMAX+1) WHERE KMAX MAY            LIV00580
C         IS THE LARGEST SIZE T-MATRIX TO BE CONSIDERED.                LIV00590
C     2.  V1:  >= MAX(N,KMAX+1)                                         LIV00600
C     3.  V2,VS:   >= MAX(N,KMAX)                                       LIV00610
C     4.  GG:  >=  KMAX                                                 LIV00620
C     5.  G:  >= MAX(N,2*KMAX)                                          LIV00630
C     6.  MP:  >= KMAX                                                  LIV00640
C     7.  LB,UB:  >= NUMBER OF SUBINTERVALS SUPPLIED TO BISEC.          LIV00650
C     8.  NMEV:  >= NUMBER OF T-MATRICES ALLOWED.                       LIV00660
C     9.  EXPLAN:  DIMENSION IS 20.                                     LIV00670
C                                                                       LIV00680
C                                                                       LIV00690
C     IMPORTANT TOLERANCES OR SCALES THAT ARE USED REPEATEDLY           LIV00700
C     THROUGHOUT THE PROGRAM ARE THE FOLLOWING:                         LIV00710
C     SCALED MACHINE EPSILON:  TTOL = TKMAX*EPSM WHERE                  LIV00720
C     EPSM = 2*MACHINE EPSILON AND                                      LIV00730
C     TKMAX = MAX(|ALPHA(J)|,BETA(J), J = 1,MEV)                        LIV00740
C     BISEC CONVERGENCE TOLERANCE:  BISTOL  = DSQRT(1000+MEV)*TTOL      LIV00750
C     BISEC MULTIPLICITY TOLERANCE:  MULTOL = (1000+MEV)*TTOL           LIV00760
C     LANCZOS CONVERGENCE TOLERANCE: CONTOL = BETA(MEV+1)*1.D-10        LIV00770
C                                                                       LIV00780
C-----------------------------------------------------------------------LIV00790
C     OUTPUT HEADER                                                     LIV00800
      WRITE(6,10)                                                       LIV00810
   10 FORMAT(/' LANCZOS PROCEDURE FOR FACTORED INVERSES OF REAL SYMMETRILIV00820
     1C MATRICES')                                                      LIV00830
C                                                                       LIV00840
C     SET PROGRAM PARAMETERS                                            LIV00850
C     SCALEK ARE USED IN TOLERANCES NEEDED IN SUBROUTINES LUMP,         LIV00860
C     ISOEV AND PRTEST.  USER MUST NOT MODIFY THESE SCALES.             LIV00870
      SCALE1 = 5.0D2                                                    LIV00880
      SCALE2 = 5.0D0                                                    LIV00890
      SCALE3 = 5.0D0                                                    LIV00900
      SCALE4 = 1.0D4                                                    LIV00910
      ONE  = 1.0D0                                                      LIV00920
      ZERO = 0.0D0                                                      LIV00930
C     BTOL = 1.0D-8                                                     LIV00940
      BTOL = EPSM                                                       LIV00950
      GAPTOL = 1.0D-8                                                   LIV00960
      ICONV = 0                                                         LIV00970
      MOLD = 0                                                          LIV00980
      MOLD1 = 1                                                         LIV00990
      ICT = 0                                                           LIV01000
      MMB = 0                                                           LIV01010
```

```
      IPROJ = 0                                                          LIV01020
C-----------------------------------------------------------------------LIV01030
C     READ USER-SPECIFIED PARAMETERS FROM INPUT FILE 5 (FREE FORMAT)     LIV01040
C                                                                        LIV01050
C     READ USER-PROVIDED HEADER FOR RUN                                  LIV01060
      READ(5,20) EXPLAN                                                  LIV01070
      WRITE(6,20) EXPLAN                                                 LIV01080
      READ(5,20) EXPLAN                                                  LIV01090
      WRITE(6,20) EXPLAN                                                 LIV01100
   20 FORMAT(20A4)                                                       LIV01110
C                                                                        LIV01120
C     READ ORDER OF MATRICES (N) , MAXIMUM ORDER OF T-MATRIX (KMAX),     LIV01130
C     NUMBER OF T-MATRICES ALLOWED (NMEVS), AND MATRIX IDENTIFICATION    LIV01140
C     NUMBERS (MATNO), SHIFT APPLIED TO MATRIX (SHIFT) AND               LIV01150
C     SCALE (S0).                                                        LIV01160
      READ(5,20) EXPLAN                                                  LIV01170
      READ(5,*) N,KMAX,NMEVS,MATNO,S0,SHIFT                              LIV01180
C                                                                        LIV01190
C     READ SEEDS FOR LANCZS AND INVERR SUBROUTINES (SVSEED AND RHSEED)   LIV01200
C     READ MAXIMUM NUMBER OF ITERATIONS ALLOWED FOR EACH INVERSE         LIV01210
C     ITERATION (MXINIT) AND MAXIMUM NUMBER OF STURM SEQUENCES           LIV01220
C     ALLOWED (MXSTUR)                                                   LIV01230
      READ(5,20) EXPLAN                                                  LIV01240
      READ(5,*) SVSEED,RHSEED,MXINIT,MXSTUR                              LIV01250
C                                                                        LIV01260
C     ISTART = (0,1):  ISTART = 0 MEANS ALPHA/BETA FILE IS NOT           LIV01270
C     AVAILABLE.  ISTART = 1 MEANS ALPHA/BETA FILE IS AVAILABLE ON       LIV01280
C     FILE 2.                                                            LIV01290
C     ISTOP = (0,1):  ISTOP = 0 MEANS PROCEDURE GENERATES ALPHA/BETA     LIV01300
C     FILE AND THEN TERMINATES.  ISTOP = 1 MEANS PROCEDURE GENERATES     LIV01310
C     ALPHAS/BETAS IF NEEDED AND THEN COMPUTES EIGENVALUES AND ERROR     LIV01320
C     ESTIMATES AND THEN TERMINATES.                                     LIV01330
      READ(5,20) EXPLAN                                                  LIV01340
      READ(5,*) ISTART,ISTOP                                             LIV01350
C                                                                        LIV01360
C     IHIS = (0,1):  IHIS = 0 MEANS ALPHA/BETA FILE IS NOT WRITTEN       LIV01370
C     TO FILE 1.  IHIS = 1 MEANS ALPHA/BETA FILE IS WRITTEN TO FILE 1.   LIV01380
C     IDIST = (0,1):  IDIST = 0 MEANS DISTINCT T-EIGENVALUES             LIV01390
C     ARE NOT WRITTEN TO FILE 11.  IDIST = 1 MEANS DISTINCT              LIV01400
C     T-EIGENVALUES ARE WRITTEN TO FILE 11.                              LIV01410
C     IWRITE = (0,1):  IWRITE = 0 MEANS NO INTERMEDIATE OUTPUT           LIV01420
C     FROM THE COMPUTATIONS IS WRITTEN TO FILE 6.  IWRITE = 1 MEANS      LIV01430
C     T-EIGENVALUES AND ERROR ESTIMATES ARE WRITTEN TO FILE 6            LIV01440
C     AS THEY ARE COMPUTED.                                              LIV01450
      READ(5,20) EXPLAN                                                  LIV01460
      READ(5,*) IHIS,IDIST,IWRITE                                        LIV01470
C                                                                        LIV01480
C     READ IN THE RELATIVE TOLERANCE (RELTOL) FOR USE IN THE             LIV01490
C     SPURIOUS, T-MULTIPLICITY, AND PRTESTS.                             LIV01500
      READ(5,20) EXPLAN                                                  LIV01510
      READ(5,*) RELTOL                                                   LIV01520
C                                                                        LIV01530
C     READ IN THE SIZES OF THE T-MATRICES TO BE CONSIDERED.              LIV01540
      READ(5,20) EXPLAN                                                  LIV01550
      READ(5,*) (NMEV(J), J=1,NMEVS)                                     LIV01560
```

```
C                                                                       LIV01570
C     READ IN THE NUMBER OF SUBINTERVALS TO BE CONSIDERED.              LIV01580
      READ(5,20) EXPLAN                                                 LIV01590
      READ(5,*) NINT                                                    LIV01600
C                                                                       LIV01610
C     READ IN THE LEFT-END POINTS OF THE SUBINTERVALS TO BE CONSIDERED. LIV01620
C     THESE MUST BE IN ALGEBRAICALLY-INCREASING ORDER                   LIV01630
      READ(5,20) EXPLAN                                                 LIV01640
      READ(5,*) (LB(J), J=1,NINT)                                       LIV01650
C                                                                       LIV01660
C     READ IN THE RIGHT-END POINTS OF THE SUBINTERVALS TO BE CONSIDERED.LIV01670
C     THESE MUST BE IN ALGEBRAICALLY-INCREASING ORDER                   LIV01680
      READ(5,20) EXPLAN                                                 LIV01690
      READ(5,*) (UB(J), J=1,NINT)                                       LIV01700
C                                                                       LIV01710
C-----------------------------------------------------------------------LIV01720
C     INITIALIZE THE ARRAYS FOR THE FACTORIZATION OF THE ASSOCIATED     LIV01730
C     SCALED, SHIFTED AND PERMUTED VERSION OF THE A-MATRIX.             LIV01740
C     THE STORAGE LOCATIONS OF THESE ARRAYS ARE PASSED TO THE BSOLV     LIV01750
C     SUBROUTINE WHICH WILL BE CALLED FROM LANCZS FOR THE T-MATRIX      LIV01760
C     GENERATION.                                                       LIV01770
C                                                                       LIV01780
      CALL USPEC(N,MATNO)                                               LIV01790
C                                                                       LIV01800
C-----------------------------------------------------------------------LIV01810
C                                                                       LIV01820
C     MASKS UNDERFLOW AND OVERFLOW, USER MUST SUPPLY OR COMMENT OUT.    LIV01830
      CALL MASK                                                         LIV01840
C                                                                       LIV01850
C-----------------------------------------------------------------------LIV01860
C                                                                       LIV01870
C     WRITE TO FILE 6, A SUMMARY OF THE PARAMETERS FOR THIS RUN         LIV01880
C                                                                       LIV01890
      WRITE(6,30) MATNO,N,KMAX,SHIFT,SO                                 LIV01900
   30 FORMAT(/3X,'MATRIX ID',4X,'ORDER OF A',4X,'MAX ORDER OF T'//      LIV01910
     1 I12,I14,I18//8X,' SHIFT',8X,'SCALE'/2E15.6//                     LIV01920
     1 ' C = SCALE*A + SHIFT*I '/                                       LIV01930
     1 ' B = P*C*P-TRANSPOSE WHERE P IS A REORDERING OF C'/             LIV01940
     1 ' LANCZOS PROCEDURE USES THE FACTORIZATION OF B'/)               LIV01950
C                                                                       LIV01960
      WRITE(6,40) ISTART,ISTOP                                          LIV01970
   40 FORMAT(/2X,'ISTART',3X,'ISTOP'/2I8/)                              LIV01980
C                                                                       LIV01990
      WRITE(6,50) IHIS,IDIST,IWRITE                                     LIV02000
   50 FORMAT(/4X,'IHIS',3X,'IDIST',2X,'IWRITE'/3I8/)                    LIV02010
C                                                                       LIV02020
      WRITE(6,60) SVSEED,RHSEED                                         LIV02030
   60 FORMAT(/' SEEDS FOR RANDOM NUMBER GENERATOR'//                    LIV02040
     1 4X,'LANCZS SEED',4X,'INVERR SEED'/2I15/)                         LIV02050
C                                                                       LIV02060
      WRITE(6,70) (NMEV(J), J=1,NMEVS)                                  LIV02070
   70 FORMAT(/' SIZES OF T-MATRICES TO BE CONSIDERED'/(6I12))           LIV02080
C                                                                       LIV02090
      WRITE(6,80) RELTOL,GAPTOL,BTOL                                    LIV02100
   80 FORMAT(/' RELATIVE TOLERANCE USED TO COMBINE COMPUTED T-EIGENVALUELIV02110
```

```
     1S'/E15.3/' RELATIVE GAP TOLERANCES USED IN INVERSE ITERATION'/    LIV02120
     1E15.3/' RELATIVE TOLERANCE FOR CHECK ON SIZE OF BETAS'/E15.3/)    LIV02130
C                                                                       LIV02140
      WRITE(6,90) (J,LB(J),UB(J), J=1,NINT)                             LIV02150
   90 FORMAT(/' BISEC WILL BE USED ON THE FOLLOWING INTERVALS'/         LIV02160
     1 (I6,2E20.6))                                                     LIV02170
C                                                                       LIV02180
      IF (ISTART.EQ.0) GO TO 140                                        LIV02190
C                                                                       LIV02200
C     READ IN ALPHA BETA HISTORY                                        LIV02210
C                                                                       LIV02220
      READ(2,100)MOLD,NOLD,SVSOLD,MATOLD,SHIFTO                         LIV02230
  100 FORMAT(2I6,I12,I8,E13.4)                                          LIV02240
C                                                                       LIV02250
      IF (KMAX.LT.MOLD) KMAX = MOLD                                     LIV02260
      KMAX1 = KMAX + 1                                                  LIV02270
C                                                                       LIV02280
C     CHECK THAT ORDER N, MATRIX ID MATNO, AND RANDOM SEED SVSEED       LIV02290
C     AGREE WITH THOSE IN THE HISTORY FILE.  IF NOT PROCEDURE STOPS.    LIV02300
C                                                                       LIV02310
      ITEMP = (NOLD-N)**2+(MATNO-MATOLD)**2+(SVSEED-SVSOLD)**2          LIV02320
C                                                                       LIV02330
      IF (ITEMP.EQ.0.AND.SHIFT.EQ.SHIFTO) GO TO 120                     LIV02340
C                                                                       LIV02350
      WRITE(6,110)                                                      LIV02360
  110 FORMAT(' PROGRAM TERMINATES'/    ' READ FROM FILE 2 CORRESPONDS TOLIV02370
     1 DIFFERENT MATRIX THAN MATRIX SPECIFIED'/)                        LIV02380
      GO TO 700                                                         LIV02390
C                                                                       LIV02400
  120 CONTINUE                                                          LIV02410
      MOLD1 = MOLD+1                                                    LIV02420
C                                                                       LIV02430
      READ(2,130)(ALPHA(J), J=1,MOLD)                                   LIV02440
      READ(2,130)(BETA(J), J=1,MOLD1)                                   LIV02450
  130 FORMAT(4Z20)                                                      LIV02460
C                                                                       LIV02470
      IF (KMAX.EQ.MOLD) GO TO 170                                       LIV02480
C                                                                       LIV02490
      READ(2,130)(V1(J), J=1,N)                                         LIV02500
      READ(2,130)(V2(J), J=1,N)                                         LIV02510
C                                                                       LIV02520
  140 CONTINUE                                                          LIV02530
      IIX = SVSEED                                                      LIV02540
C                                                                       LIV02550
      WRITE(6,150)                                                      LIV02560
  150 FORMAT(' ENTERING LANCZS'/)                                       LIV02570
C                                                                       LIV02580
C-----------------------------------------------------------------------LIV02590
C                                                                       LIV02600
      CALL LANCZS(BSOLV,ALPHA,BETA,V1,V2,VS,G,KMAX,MOLD1,N,IIX)         LIV02610
C                                                                       LIV02620
C-----------------------------------------------------------------------LIV02630
C                                                                       LIV02640
C     ALPHA BETA WRITE                                                  LIV02650
      KMAX1 = KMAX + 1                                                  LIV02660
```

```
C                                                                       LIV02670
      IF(IHIS.EQ.0.AND.ISTOP.GT.0) GO TO 170                            LIV02680
C                                                                       LIV02690
      WRITE(1,160) KMAX,N,SVSEED,MATNO,SHIFT                            LIV02700
  160 FORMAT(2I6,I12,I8,E13.4,' = KMAX,N,SVSEED,MATNO,SHIFT')           LIV02710
C                                                                       LIV02720
      WRITE(1,130)(ALPHA(I), I=1,KMAX)                                  LIV02730
      WRITE(1,130)(BETA(I), I=1,KMAX1)                                  LIV02740
C                                                                       LIV02750
      WRITE(1,130)(V1(I), I=1,N)                                        LIV02760
      WRITE(1,130)(V2(I), I=1,N)                                        LIV02770
C                                                                       LIV02780
      IF (ISTOP.EQ.0) GO TO 600                                         LIV02790
C                                                                       LIV02800
  170 CONTINUE                                                          LIV02810
      KMAX1 = KMAX + 1                                                  LIV02820
      BKMIN = BTOL                                                      LIV02830
C                                                                       LIV02840
      WRITE(6,180)                                                      LIV02850
  180 FORMAT(/' T-MATRICES (ALPHA AND BETA) ARE NOW AVAILABLE'/)        LIV02860
C                                                                       LIV02870
C-----------------------------------------------------------------------LIV02880
C     SUBROUTINE TNORM CHECKS MIN(BETA)/(ESTIMATED NORM(A)) > BTOL .    LIV02890
C     IF THIS IS VIOLATED IB IS SET EQUAL TO THE NEGATIVE OF THE INDEX  LIV02900
C     OF THE MINIMAL BETA.  IF(IB < 0) THEN SUBROUTINE TNORM IS         LIV02910
C     CALLED FOR EACH VALUE OF MEV TO DETERMINE WHETHER OR NOT THERE    LIV02920
C     IS A BETA IN THE T-MATRIX SPECIFIED THAT VIOLATES THIS TEST.      LIV02930
C     IF THERE IS SUCH A BETA THE PROGRAM TERMINATES FOR THE USER       LIV02940
C     TO DECIDE WHAT TO DO.  THIS TEST CAN BE OVER-RIDDEN BY            LIV02950
C     SIMPLY MAKING BTOL SMALLER, BUT THEN THERE IS THE POSSIBILITY     LIV02960
C     THAT LOSSES IN THE LOCAL ORTHOGONALITY MAY HURT THE COMPUTATIONS. LIV02970
C     BTOL = 1.D-8 IS HOWEVER A CONSERVATIVE CHOICE FOR BTOL.           LIV02980
C                                                                       LIV02990
C     TNORM ALSO COMPUTES TKMAX = MAX(|ALPHA(K)|,BETA(K), K=1,KMAX).    LIV03000
C     TKMAX IS USED TO SCALE THE TOLERANCES USED IN THE                 LIV03010
C     T-MULTIPLICITY AND SPURIOUS TESTS IN BISEC. TKMAX IS ALSO USED IN LIV03020
C     THE PROJECTION TEST FOR HIDDEN EIGENVALUES THAT HAD 'TOO SMALL'   LIV03030
C     A PROJECTION ON THE STARTING VECTOR.                              LIV03040
C                                                                       LIV03050
      CALL TNORM(ALPHA,BETA,BKMIN,TKMAX,KMAX,IB)                        LIV03060
C                                                                       LIV03070
C-----------------------------------------------------------------------LIV03080
      TTOL = EPSM*TKMAX                                                 LIV03090
C                                                                       LIV03100
C     LOOP ON THE SIZE OF THE T-MATRIX                                  LIV03110
  190 CONTINUE                                                          LIV03120
      MMB = MMB + 1                                                     LIV03130
      MEV = NMEV(MMB)                                                   LIV03140
C     IS MEV TOO LARGE ?                                                LIV03150
      IF(MEV.LE.KMAX) GO TO 210                                         LIV03160
C                                                                       LIV03170
      WRITE(6,200) MMB, MEV, KMAX                                       LIV03180
  200 FORMAT(/' TERMINATE PRIOR TO CONSIDERING THE',I6,'TH T-MATRIX'/   LIV03190
     1' BECAUSE THE SIZE REQUESTED',I6,' IS GREATER THAN THE MAXIMUM SIZLIV03200
     1E ALLOWED',I6/)                                                   LIV03210
```

```
      GO TO 600                                                         LIV03220
C                                                                       LIV03230
  210 MP1 = MEV + 1                                                     LIV03240
      BETAM = BETA(MP1)                                                 LIV03250
      WRITE(6,220) MEV,MEV,BETA(MEV),MEV,BETAM                          LIV03260
  220 FORMAT(/' AT T-SIZE = ',I6,' BETA(',I4,') = ',E13.4/' BETA(',I4,'+LIV03270
     11) =',E13.4)                                                      LIV03280
      IF (IB.GE.0) GO TO 230                                            LIV03290
      T0 = BTOL                                                         LIV03300
C-----------------------------------------------------------------------LIV03310
C                                                                       LIV03320
      CALL TNORM(ALPHA,BETA,T0,T1,MEV,IBMEV)                            LIV03330
C                                                                       LIV03340
C-----------------------------------------------------------------------LIV03350
      TEMP = T0/TKMAX                                                   LIV03360
      IBMEV = IABS(IBMEV)                                               LIV03370
      IF (TEMP.GE.BTOL) GO TO 230                                       LIV03380
      IBMEV = -IBMEV                                                    LIV03390
      GO TO 660                                                         LIV03400
  230 CONTINUE                                                          LIV03410
      IC = MXSTUR-ICT                                                   LIV03420
C                                                                       LIV03430
C-----------------------------------------------------------------------LIV03440
C     BISEC LOOP. THE SUBROUTINE BISEC INCORPORATES DIRECTLY THE        LIV03450
C     T-MULTIPLICITY AND SPURIOUS TESTS. T-EIGENVALUES WILL BE          LIV03460
C     CALCULATED BY BISEC SEQUENTIALLY ON INTERVALS                     LIV03470
C     (LB(J),UB(J)), J = 1,NINT).                                       LIV03480
C                                                                       LIV03490
C     ON RETURN FROM BISEC                                              LIV03500
C     NDIS = NUMBER OF DISTINCT EIGENVALUES OF T(1,MEV) ON UNION        LIV03510
C            OF THE (LB,UB) INTERVALS                                   LIV03520
C     VS = DISTINCT T-EIGENVALUES IN ALGEBRAICALLY INCREASING ORDER     LIV03530
C     MP = T-MULTIPLICITIES OF THE T-EIGENVALUES IN VS                  LIV03540
C     MP(I) = (0,1,MI), MI>1, I=1,NDIS  MEANS:                          LIV03550
C        (0)  VS(I) IS SPURIOUS                                         LIV03560
C        (1)  VS(I) IS T-SIMPLE AND GOOD                                LIV03570
C        (MI) VS(I) IS MULTIPLE AND IS THEREFORE NOT ONLY GOOD BUT      LIV03580
C             ALSO A CONVERGED GOOD T-EIGENVALUE.                       LIV03590
C                                                                       LIV03600
      CALL BISEC(ALPHA,BETA,V1,V2,VS,LB,UB,EPSM,TTOL,MP,NINT,           LIV03610
     1 MEV,NDIS,IC,IWRITE)                                              LIV03620
C                                                                       LIV03630
C-----------------------------------------------------------------------LIV03640
      IF (NDIS.EQ.0) GO TO 680                                          LIV03650
C                                                                       LIV03660
C     COMPUTE THE TOTAL NUMBER OF STURM SEQUENCES USED TO DATE          LIV03670
C     COMPUTE THE BISEC CONVERGENCE AND T-MULTIPLICITY TOLERANCES USED. LIV03680
C     COMPUTE THE CONVERGENCE TOLERANCE FOR EIGENVALUES OF A.           LIV03690
      ICT = ICT + IC                                                    LIV03700
      TEMP = DFLOAT(MEV+1000)                                           LIV03710
      MULTOL = TEMP*TTOL                                                LIV03720
      TEMP = DSQRT(TEMP)                                                LIV03730
      BISTOL = TTOL*TEMP                                                LIV03740
      CONTOL = BETAM*1.D-10                                             LIV03750
C                                                                       LIV03760
```

```
C------------------------------------------------------------------------LIV03770
C    SUBROUTINE LUMP 'COMBINES' T-EIGENVALUES THAT ARE 'TOO CLOSE'.     LIV03780
C    NOTE HOWEVER THAT CLOSE SPURIOUS T-EIGENVALUES ARE NOT AVERAGED    LIV03790
C    WITH GOOD ONES. HOWEVER, THEY MAY BE USED TO INCREASE THE          LIV03800
C    T-MULTIPLICITY OF A GOOD T-EIGENVALUE.                             LIV03810
C                                                                       LIV03820
      LOOP = NDIS                                                       LIV03830
      CALL LUMP(VS,RELTOL,MULTOL,SCALE2,MP,LOOP)                        LIV03840
C                                                                       LIV03850
C------------------------------------------------------------------------LIV03860
      IF(NDIS.EQ.LOOP) GO TO 250                                        LIV03870
C                                                                       LIV03880
      WRITE(6,240) NDIS, MEV, LOOP                                      LIV03890
  240 FORMAT(/I6,' DISTINCT T-EIGENVALUES WERE COMPUTED IN BISEC AT MEV LIV03900
     1',I6/ 2X,' LUMP SUBROUTINE REDUCES NUMBER OF DISTINCT T-EIGENVALUELIV03910
     1S TO',I6)                                                         LIV03920
C                                                                       LIV03930
  250 CONTINUE                                                          LIV03940
      NDIS = LOOP                                                       LIV03950
      BETA(MP1) = BETAM                                                 LIV03960
C------------------------------------------------------------------------LIV03970
C     THE SUBROUTINE ISOEV LABELS THOSE SIMPLE EIGENVALUES OF T(1,MEV)  LIV03980
C     WITH VERY SMALL GAPS BETWEEN NEIGHBORING EIGENVALUES OF T(1,MEV)  LIV03990
C     TO AVOID COMPUTING ERROR ESTIMATES FOR ANY SIMPLE GOOD            LIV04000
C     T-EIGENVALUE THAT IS TOO CLOSE TO A SPURIOUS EIGENVALUE.          LIV04010
C     ON RETURN FROM ISOEV, G CONTAINS CODED MINIMAL GAPS               LIV04020
C     BETWEEN THE DISTINCT EIGENVALUES OF T(1,MEV). (G IS REAL).        LIV04030
C     G(I) < 0 MEANS MINGAP IS DUE TO LEFT GAP G(I) > 0 MEANS DUE TO    LIV04040
C     RIGHT GAP. MP(I) = -1 MEANS THAT THE GOOD T-EIGENVALUE IS SIMPLE  LIV04050
C     AND HAS A VERY SMALL MINGAP IN T(1,MEV) DUE TO A SPURIOUS         LIV04060
C     T-EIGENVALUE.  NG = NUMBER OF GOOD T-EIGENVALUES.                 LIV04070
C     NISO = NUMBER OF ISOLATED GOOD T-EIGENVALUES.                     LIV04080
C                                                                       LIV04090
      CALL ISOEV(VS,GAPTOL,MULTOL,SCALE1,G,MP,NDIS,NG,NISO)             LIV04100
C                                                                       LIV04110
C------------------------------------------------------------------------LIV04120
C                                                                       LIV04130
      WRITE(6,260)NG,NISO,NDIS                                          LIV04140
  260 FORMAT(/I6,' GOOD T-EIGENVALUES HAVE BEEN COMPUTED'/              LIV04150
     1 I6,' OF THESE ARE T-ISOLATED'/                                   LIV04160
     2 I6,' = NUMBER OF DISTINCT T-EIGENVALUES COMPUTED'/)              LIV04170
C                                                                       LIV04180
C     DO WE WRITE DISTINCT EIGENVALUES OF T-MATRIX TO FILE 11?          LIV04190
      IF (IDIST.EQ.0) GO TO 310                                         LIV04200
C                                                                       LIV04210
      WRITE(11,270) NDIS,NISO,MEV,N,SVSEED,MATNO                        LIV04220
  270 FORMAT(/4I6,I12,I8,' = NDIS,NISO,MEV,N,SVSEED,MATNO'/)            LIV04230
C                                                                       LIV04240
      WRITE(11,280)                                                     LIV04250
  280 FORMAT(/1X,'MP',21X,'EVBI',5X,'TMINGAP',1X,'MP',21X,'EVBI',5X,    LIV04260
     1'TMINGAP'/)                                                       LIV04270
C                                                                       LIV04280
      WRITE(11,290) (MP(I),VS(I),G(I), I=1,NDIS)                        LIV04290
  290 FORMAT(2(I3,E25.16,E12.3))                                        LIV04300
C                                                                       LIV04310
```

```
      WRITE(11,300) NDIS, (MP(I), I=1,NDIS)                              LIV04320
  300 FORMAT(/I6,' = NDIS, T-MULTIPLICITIES (0 MEANS  SPURIOUS)'/(20I4))LIV04330
C                                                                        LIV04340
  310 CONTINUE                                                           LIV04350
      IF (NISO.NE.0) GO TO 340                                           LIV04360
C                                                                        LIV04370
      WRITE(4,320) MEV                                                   LIV04380
  320 FORMAT(/' AT MEV = ',I6,' THERE ARE NO ISOLATED T-EIGENVALUES'/    LIV04390
     1' SO NO ERROR ESTIMATES WERE COMPUTED/')                           LIV04400
C                                                                        LIV04410
      WRITE(6,330)                                                       LIV04420
  330 FORMAT(/' ALL COMPUTED GOOD T-EIGENVALUES ARE MULTIPLE'/           LIV04430
     1 ' THEREFORE ALL SUCH EIGENVALUES ARE ASSUMED TO HAVE CONVERGED') LIV04440
C                                                                        LIV04450
      ICONV = 1                                                          LIV04460
      GO TO 380                                                          LIV04470
  340 CONTINUE                                                           LIV04480
C-----------------------------------------------------------------------LIV04490
C     SUBROUTINE INVERR COMPUTES ERROR ESTIMATES FOR ISOLATED GOOD       LIV04500
C     T-EIGENVALUES USING INVERSE ITERATION ON T(1,MEV). ON RETURN       LIV04510
C     G(J) = MINIMUM GAP IN T(1,MEV) FOR EACH VS(J), J=1,NDIS            LIV04520
C     G(MEV+I) = BETAM*|U(MEV)| = ERROR ESTIMATE FOR ISOLATED GOOD       LIV04530
C              T-EIGENVALUES, WHERE I = 1, NISO AND  BETAM = BETA(MEV+1)LIV04540
C              U(MEV) IS MEVTH COMPONENT OF THE UNIT EIGENVECTOR OF T    LIV04550
C              CORRESPONDING TO THE ITH ISOLATED GOOD T-EIGENVALUE.      LIV04560
C     A NEGATIVE ERROR ESTIMATE MEANS THAT FOR THAT PARTICULAR           LIV04570
C     EIGENVALUE THE INVERSE ITERATION DID NOT CONVERGE IN <= MXINIT     LIV04580
C     STEPS AND THAT THE CORRESPONDING ERROR ESTIMATE IS QUESTIONABLE.   LIV04590
C                                                                        LIV04600
C     V2 CONTAINS THE ISOLATED GOOD T-EIGENVALUES                        LIV04610
C     V1 CONTAINS THE MINGAPS TO THE NEAREST DISTINCT  EIGENVALUE        LIV04620
C        OF T(1,MEV) FOR EACH ISOLATED GOOD T-EIGENVALUE IN V2.          LIV04630
C     VS CONTAINS THE NDIS DISTINCT EIGENVALUES OF T(1,MEV)              LIV04640
C     MP CONTAINS THE CORRESPONDING CODED T-MULTIPLICITIES               LIV04650
C                                                                        LIV04660
      IT = MXINIT                                                        LIV04670
      CALL INVERR(ALPHA,BETA,V1,V2,VS,EPSM,G,MP,MEV,MMB,NDIS,NISO,N,     LIV04680
     1 RHSEED,IT,IWRITE)                                                 LIV04690
C                                                                        LIV04700
C-----------------------------------------------------------------------LIV04710
C     SIMPLE CHECK FOR CONVERGENCE. CHECKS TO SEE IF ALL OF THE          LIV04720
C     LAST COMPONENTS OF EIGENVECTORS ARE L.T. CONTOL.                   LIV04730
C     IF THIS TEST IS SATISFIED, THEN CONVERGENCE FLAG, ICONV IS SET     LIV04740
C     TO 1.  TYPICALLY ERROR ESTIMATES ARE VERY CONSERVATIVE.            LIV04750
C                                                                        LIV04760
      WRITE(6,350) CONTOL                                                LIV04770
  350 FORMAT(/' CONVERGENCE IS TESTED USING THE CONVERGENCE TOLERANCE', LIV04780
     1E13.4/)                                                            LIV04790
C                                                                        LIV04800
      II = MEV +1                                                        LIV04810
      IF = MEV+NISO                                                      LIV04820
      DO 360 I = II,IF                                                   LIV04830
      IF (ABS(G(I)).GT.CONTOL) GO TO 380                                 LIV04840
  360 CONTINUE                                                           LIV04850
      ICONV = 1                                                          LIV04860
```

```
      MMB = NMEVS                                                       LIV04870
C                                                                       LIV04880
      WRITE(6,370) CONTOL                                               LIV04890
  370 FORMAT(' ALL COMPUTED ERROR ESTIMATES WERE LESS THAN',E15.4/      LIV04900
     1 ' THEREFORE PROCEDURE TERMINATES'/)                              LIV04910
C                                                                       LIV04920
  380 CONTINUE                                                          LIV04930
C                                                                       LIV04940
      IF (ICONV.EQ.0) GO TO 510                                         LIV04950
C                                                                       LIV04960
C---------------------------------------------------------------------- LIV04970
C     IF CONVERGENCE IS INDICATED, THAT IS ICONV = 1 ,THEN              LIV04980
C     THE SUBROUTINE PRTEST IS CALLED TO CHECK FOR ANY CONVERGED        LIV04990
C     T-EIGENVALUES THAT HAVE BEEN MISLABELLED AS SPURIOUS BECAUSE      LIV05000
C     THE PROJECTION OF THEIR EIGENVECTOR(S) ON THE STARTING            LIV05010
C     VECTOR WAS(WERE) TOO SMALL.                                       LIV05020
C     NUMERICAL TESTS INDICATE THAT SUCH EIGENVALUES ARE RARE.          LIV05030
C     IF FOR SOME REASON MANY OF THESE HIDDEN EIGENVALUES APPEAR        LIV05040
C     ON SOME RUN, YOU CAN BE CERTAIN THAT SOMETHING IS FOULED UP.      LIV05050
C                                                                       LIV05060
      CALL PRTEST(ALPHA,BETA,VS,TKMAX,EPSM,RELTOL,SCALE3,SCALE4,        LIV05070
     1 MP,NDIS,MEV,IPROJ)                                               LIV05080
C                                                                       LIV05090
C-----------------------------------------------------------------------LIV05100
C                                                                       LIV05110
      IF(IPROJ.EQ.0) GO TO 500                                          LIV05120
C                                                                       LIV05130
      IF(IDIST.EQ.1)  WRITE(11,390) IPROJ                               LIV05140
  390 FORMAT(' SUBROUTINE PRTEST WANTS TO RELABEL',I6,' SPURIOUS T-EIGENLIV05150
     1VALUES'/' WE ACCEPT RELABELLING ONLY IF LAST COMPONENT OF T-EIGENVLIV05160
     1ECTOR IS L.T. 1.D-10'/)                                           LIV05170
C                                                                       LIV05180
      IIX = RHSEED                                                      LIV05190
C                                                                       LIV05200
C-----------------------------------------------------------------------LIV05210
C                                                                       LIV05220
      CALL GENRAN(IIX,G,MEV)                                            LIV05230
C                                                                       LIV05240
C-----------------------------------------------------------------------LIV05250
C                                                                       LIV05260
      ITEN = -10                                                        LIV05270
      NISOM = NISO + MEV                                                LIV05280
      IWRITO = IWRITE                                                   LIV05290
      IWRITE = 0                                                        LIV05300
C                                                                       LIV05310
      DO 420 J = 1,NDIS                                                 LIV05320
      IF(MP(J).NE.ITEN) GO TO 420                                       LIV05330
      TO = VS(J)                                                        LIV05340
C                                                                       LIV05350
C-----------------------------------------------------------------------LIV05360
C                                                                       LIV05370
      IT = MXINIT                                                       LIV05380
      CALL INVERM(ALPHA,BETA,V1,V2,TO,TEMP,T1,EPSM,G,MEV,IT,IWRITE)     LIV05390
C                                                                       LIV05400
C-----------------------------------------------------------------------LIV05410
```

```
C                                                                       LIV05420
      IF(TEMP.LE.1.D-10) GO TO 410                                      LIV05430
C     ERROR ESTIMATE WAS NOT SMALL REJECT RELABELLING OF THIS           LIV05440
C     T-EIGENVALUE                                                      LIV05450
      IF(IDIST.EQ.1)  WRITE(11,400) J,T0,TEMP                           LIV05460
  400 FORMAT(/' LAST COMPONENT FOR',I6,'TH T-EIGENVALUE',E20.12/' IS TOOLIV05470
     1 LARGE = ',E15.6,' SO DO NOT ACCEPT PRTEST RELABELLING'/)         LIV05480
      MP(J) = 0                                                         LIV05490
      IPROJ = IPROJ - 1                                                 LIV05500
      GO TO 420                                                         LIV05510
C     RELABELLING ACCEPTED                                              LIV05520
  410 NISOM = NISOM + 1                                                 LIV05530
      G(NISOM) = BETAM*TEMP                                             LIV05540
  420 CONTINUE                                                          LIV05550
      IWRITE = IWRIT0                                                   LIV05560
C                                                                       LIV05570
      IF(IPROJ.EQ.0) GO TO 460                                          LIV05580
      WRITE(6,430) IPROJ                                                LIV05590
  430 FORMAT(/I6,' T-EIGENVALUES WERE RECLASSIFIED AS GOOD.'/           LIV05600
     1' THESE ARE IDENTIFIED IN FILE 3 BY A T-MULTIPLICITY OF -10'/' USELIV05610
     2R SHOULD INSPECT EACH TO MAKE SURE NEIGHBORS HAVE CONVERGED')     LIV05620
C                                                                       LIV05630
      IF(IDIST.EQ.1)  WRITE(11,440) IPROJ                               LIV05640
  440 FORMAT(/I6,' T-EIGENVALUES WERE RELABELLED AS GOOD'/              LIV05650
     1' BELOW IS CORRECTED T-MULTIPLICITY PATTERN'/)                    LIV05660
C                                                                       LIV05670
      WRITE(6,450) NDIS, (MP(I), I=1,NDIS)                              LIV05680
      IF(IDIST.EQ.1)  WRITE(11,450) NDIS, (MP(I), I=1,NDIS)             LIV05690
  450 FORMAT(/I6,' = NDIS, T-MULTIPLICITIES (0 MEANS  SPURIOUS)'/       LIV05700
     1 6X, ' (-10) MEANS SPURIOUS T-EIGENVALUE RELABELLED AS GOOD'/(20I4LIV05710
     1))                                                                LIV05720
C                                                                       LIV05730
C     RECALCULATE MINGAPS FOR DISTINCT T(1,MEV) EIGENVALUES.            LIV05740
  460 NM1 = NDIS - 1                                                    LIV05750
      G(NDIS) = VS(NM1)-VS(NDIS)                                        LIV05760
      G(1) = VS(2)-VS(1)                                                LIV05770
C                                                                       LIV05780
      DO 470 J = 2,NM1                                                  LIV05790
      T0 = VS(J)-VS(J-1)                                                LIV05800
      T1 = VS(J+1)-VS(J)                                                LIV05810
      G(J) = T1                                                         LIV05820
      IF (T0.LT.T1) G(J) = -T0                                          LIV05830
  470 CONTINUE                                                          LIV05840
      IF(IPROJ.EQ.0) GO TO 500                                          LIV05850
C     WRITE TO FILE 4 ERROR ESTIMATES FOR THOSE T-EIGENVALUES RELABELLEDLIV05860
      NGOOD = 0                                                         LIV05870
      DO 480 J = 1,NDIS                                                 LIV05880
      IF(MP(J).EQ.0) GO TO 480                                          LIV05890
      NGOOD = NGOOD + 1                                                 LIV05900
      IF(MP(J).NE.ITEN) GO TO 480                                       LIV05910
      T0 = VS(J)                                                        LIV05920
      NISO = NISO + 1                                                   LIV05930
      NISOM = MEV + NISO                                                LIV05940
      WRITE(4,490) NGOOD,T0,G(NISOM),G(J)                               LIV05950
  480 CONTINUE                                                          LIV05960
```

```
  490 FORMAT(I10,E25.16,2E14.3)                                          LIV05970
C                                                                        LIV05980
  500 CONTINUE                                                           LIV05990
C                                                                        LIV06000
C     WRITE THE GOOD T-EIGENVALUES TO FILE 3.  FIRST TRANSFER THEM       LIV06010
C     TO V2 AND THEIR T-MULTIPLICITIES TO THE CORRESPONDING POSITIONS     LIV06020
C     IN MP AND COMPUTE THE A-MINGAPS, THE MINIMAL GAPS BETWEEN THE      LIV06030
C     GOOD T-EIGENVALUES.  THESE GAPS WILL BE PUT IN THE ARRAY G.        LIV06040
C     SINCE G CURRENTLY CONTAINS THE MINIMAL GAPS BETWEEN THE DISTINCT   LIV06050
C     EIGENVALUES OF THE T-MATRIX, THESE GAPS WILL FIRST BE              LIV06060
C     TRANSFERRED TO V1.  NOTE THAT V1<0 MEANS THAT THAT MINIMAL GAP     LIV06070
C     IN THE T-MATRIX IS DUE TO A SPURIOUS T-EIGENVALUE.                 LIV06080
C     ALL THIS INFORMATION IS PRINTED TO FILE 3                          LIV06090
C                                                                        LIV06100
  510 CONTINUE                                                           LIV06110
      NG = 0                                                             LIV06120
      DO 520 I = 1,NDIS                                                  LIV06130
      IF (MP(I).EQ.0) GO TO 520                                          LIV06140
      NG = NG+1                                                          LIV06150
      MP(NG) = MP(I)                                                     LIV06160
      V2(NG) = VS(I)                                                     LIV06170
      TEMP = G(I)                                                        LIV06180
      TEMP = DABS(TEMP)                                                  LIV06190
      J = I+1                                                            LIV06200
      IF (G(I).LT.ZERO) J = I-1                                          LIV06210
      IF (MP(J).EQ.0) TEMP = -TEMP                                       LIV06220
      V1(NG) = TEMP                                                      LIV06230
  520 CONTINUE                                                           LIV06240
C                                                                        LIV06250
      WRITE(6,530)MEV                                                    LIV06260
  530 FORMAT(//' T-EIGENVALUE CALCULATION AT MEV = ',I6,'  IS COMPLETE'/LIV06270
     1)                                                                  LIV06280
C                                                                        LIV06290
C     NG = NUMBER OF COMPUTED DISTINCT GOOD T-EIGENVALUES.  NEXT         LIV06300
C     GENERATE GAPS BETWEEN GOOD T-EIGENVALUES (BIMINGAPS) AND PUT THEM LIV06310
C     G.  G(J) < 0 MEANS THE MINIMAL GAP IS DUE TO THE LEFT-HAND GAP.    LIV06320
C                                                                        LIV06330
C     GG(J) = BIMINGAP FOR EIGENVALUES OF B-INVERSE MATRIX.              LIV06340
      NGM1 = NG - 1                                                      LIV06350
      GG(NG) = V2(NGM1)-V2(NG)                                           LIV06360
      GG(1) = V2(2)-V2(1)                                                LIV06370
C                                                                        LIV06380
      DO 540 J = 2,NGM1                                                  LIV06390
      T0 = V2(J)-V2(J-1)                                                 LIV06400
      T1 = V2(J+1)-V2(J)                                                 LIV06410
      GG(J) = T1                                                         LIV06420
      IF (T0.LT.T1) GG(J) = -T0                                          LIV06430
  540 CONTINUE                                                           LIV06440
C                                                                        LIV06450
C     WRITE GOOD BI EIGENVALUES TO FILE 3.                               LIV06460
      WRITE(3,550)NG,NDIS,MEV,N,SVSEED,MATNO,MULTOL,IB,BTOL,SHIFT        LIV06470
  550 FORMAT(4I6,I12,I8, ' = NG,NDIS,MEV,N,SVSEED,MATNO'/                LIV06480
     1 E20.12,I6,2E10.3,' = MULTOL,I(MINBETA),BTOL,SHIFT')               LIV06490
C                                                                        LIV06500
C     CALCULATE EIGENVALUES OF ORIGINAL INPUT MATRIX CORRESPONDING       LIV06510
```

```
C     TO COMPUTED GOOD T-EIGENVALUES.                                    LIV06520
      TEMP = -ONE/SO                                                     LIV06530
      DO 560 K = 1,NG                                                    LIV06540
      VS(K) = (SHIFT - (ONE/V2(K)))*TEMP                                 LIV06550
  560 CONTINUE                                                           LIV06560
C                                                                        LIV06570
      NGM1 = NG - 1                                                      LIV06580
      G(NG) = DABS(VS(NGM1)-VS(NG))                                      LIV06590
      G(1) = DABS(VS(2)-VS(1))                                           LIV06600
C                                                                        LIV06610
      DO 570 J = 2,NGM1                                                  LIV06620
      T0 = DABS(VS(J)-VS(J-1))                                           LIV06630
      T1 = DABS(VS(J+1)-VS(J))                                           LIV06640
      G(J) = T1                                                          LIV06650
      IF (T0.LT.T1) G(J)=-T0                                             LIV06660
  570 CONTINUE                                                           LIV06670
C                                                                        LIV06680
      WRITE(3,580)                                                       LIV06690
  580 FORMAT(' EVNO',1X,'TMULT',20X,'EVBI',5X,'BIGAP',6X,'AGAP',6X,      LIV06700
     1'TGAP',12X,'EVA')                                                  LIV06710
C                                                                        LIV06720
      WRITE(3,590)(I,MP(I),V2(I),GG(I),G(I),V1(I),VS(I),  I=1,NG)        LIV06730
  590 FORMAT(2I5,E25.16,3E10.3,E15.8)                                    LIV06740
C                                                                        LIV06750
C     IF CONVERGENCE FLAG ICONV.NE.1 AND NUMBER OF T-MATRICES            LIV06760
C     CONSIDERED TO DATE IS LESS THAN NUMBER ALLOWED, INCREMENT MEV.     LIV06770
C     AND LOOP BACK TO 210 TO REPEAT COMPUTATIONS.  RESTORE BETA(MEV+1).LIV06780
C                                                                        LIV06790
      BETA(MP1) = BETAM                                                  LIV06800
      IF (MMB.LT.NMEVS.AND.ICONV.NE.1) GO TO 190                         LIV06810
C     END OF LOOP ON DIFFERENT SIZE T-MATRICES ALLOWED.                  LIV06820
  600 CONTINUE                                                           LIV06830
C                                                                        LIV06840
      IF(ISTOP.EQ.0)  WRITE(6,610)                                       LIV06850
  610 FORMAT(/' T-MATRICES (ALPHA AND BETA) ARE NOW AVAILABLE, TERMINATELIV06860
     1')                                                                 LIV06870
      IF (IHIS.EQ.1.AND.KMAX.NE.MOLD) WRITE(1,620)                       LIV06880
  620 FORMAT(/' ABOVE ARE THE FOLLOWING VECTORS '/                       LIV06890
     1 '  ALPHA(I), I = 1,KMAX'/                                         LIV06900
     2 '  BETA(I), I = 1,KMAX+1'/                                        LIV06910
     3 ' FINAL TWO LANCZOS VECTORS OF ORDER N FOR I = KMAX,KMAX+1'/      LIV06920
     4 ' ALL VECTORS IN THIS FILE HAVE HEX FORMAT 4Z20'/                 LIV06930
     5 ' ----- END OF FILE 1 NEW ALPHA, BETA HISTORY--------------'///)LIV06940
C                                                                        LIV06950
      IF (ISTOP.EQ.0) GO TO 700                                          LIV06960
C                                                                        LIV06970
      WRITE(3,630)                                                       LIV06980
  630 FORMAT(/' ABOVE ARE COMPUTED GOOD T-EIGENVALUES'/                  LIV06990
     1 ' NG = NUMBER OF GOOD T-EIGENVALUES COMPUTED'/                    LIV07000
     2 ' NDIS = NUMBER OF COMPUTED DISTINCT EIGENVALUES OF T(1,MEV)'/    LIV07010
     3 ' N = ORDER OF A,  MATNO = MATRIX IDENT'/                         LIV07020
     3 ' THERE ARE TWO SETS OF EIGENVALUES, THOSE FOR A AND THOSE FOR'/ LIV07030
     3 ' B-INVERSE WHERE C=S0*A + SHIFT*I, B = P*C*P-TRANS = L*L-TRANS'/LIV07040
     3 ' THE LANCZOS RECURSIONS ARE APPLIED TO B-INVERSE, USING L'/      LIV07050
     3 ' IF EVBI IS A GOOD EIGENVALUE OF B-INVERSE, THEN EVA IS A'/      LIV07060
```

```
     3 ' GOOD EIGENVALUE OF A WHERE EVA = (SHIFT-ONE/EVBI)(-ONE/SO)'/   LIV07070
     4 ' MULTOL = T-MULTIPLICITY TOLERANCE FOR T-EIGENVALUES IN BISEC'/ LIV07080
     4 ' TMULT IS THE T-MULTIPLICITY OF GOOD T-EIGENVALUE'/             LIV07090
     5 ' TMULT = -1 MEANS SPURIOUS T-EIGENVALUE TOO CLOSE'/             LIV07100
     6 ' DO NOT COMPUTE ERROR ESTIMATES FOR SUCH EIGENVALUES'/          LIV07110
     7 ' AMINGAP = MINIMAL GAP BETWEEN THE COMPUTED A-EIGENVALUES'/     LIV07120
     8 ' AMINGAP .LT. 0. MEANS MINIMAL GAP IS DUE TO LEFT-HAND GAP'/    LIV07130
     9 ' TMINGAP= MINIMAL GAP W.R.T.  DISTINCT EIGENVALUES IN T(1,MEV)'/LIV07140
     1 ' TMINGAP .LT. 0. MEANS MINGAP IS DUE TO SPURIOUS T-EIGENVALUE'/ LIV07150
     2 ' ----- END OF FILE 3 GOODEIGENVALUES----------------------'///)LIV07160
C                                                                       LIV07170
      IF (IDIST.EQ.1) WRITE(11,640)                                     LIV07180
  640 FORMAT(/' ABOVE ARE THE DISTINCT EIGENVALUES OF T(1,MEV).'/       LIV07190
     2 ' THE FORMAT IS      T-MULTIPLICITY    T-EIGENVALUE   TMINGAP'/  LIV07200
     3 '        THIS FORMAT IS REPEATED TWICE ON EACH LINE.'/           LIV07210
     4 ' T-MULTIPLICITY = -1 MEANS THAT THE SUBROUTINE ISOEV HAS TAGGED'LIV07220
     5/'   THIS SIMPLE T-EIGENVALUE AS HAVING A VERY CLOSE SPURIOUS'/   LIV07230
     6 '   T-EIGENVALUE SO THAT NO ERROR ESTIMATE WILL BE COMPUTED'/    LIV07240
     7 '   FOR THAT EIGENVALUE IN SUBROUTINE INVERR.'/                  LIV07250
     8 ' TMINGAP .LT. 0, TMINGAP IS DUE TO LEFT GAP .GT. 0, RIGHT GAP.'/LIV07260
     9 ' EACH OF THE DISTINCT T-EIGENVALUE TABLES IS FOLLOWED'/         LIV07270
     9 ' BY THE T-MULTIPLICITY PATTERN.'/                               LIV07280
     1 ' NDIS = NUMBER OF COMPUTED DISTINCT EIGENVALUES OF T(1,MEV).'/  LIV07290
     2 ' NG = NUMBER OF GOOD T-EIGENVALUES. '/                          LIV07300
     3 ' NISO = NUMBER OF ISOLATED GOOD T-EIGENVALUES. '/               LIV07310
     4 ' NISO ALSO IS THE COUNT OF +1 ENTRIES IN T-MULTIPLICITY PATTERN.LIV07320
     5 '/' ----- END OF FILE 11 DISTINCT T-EIGENVALUES--------------'///LIV07330
     6)                                                                 LIV07340
C                                                                       LIV07350
      IF(NIOS.NE.0)  WRITE(4,650)                                       LIV07360
  650 FORMAT(/' ABOVE ARE THE ERROR ESTIMATES OBTAINED FOR THE ISOLATED LIV07370
     1GOOD T-EIGENVALUES'/                                              LIV07380
     1' OBTAINED VIA INVERSE ITERATION IN THE SUBROUTINE INVERR.'/      LIV07390
     1' ALL OTHER GOOD T-EIGENVALUES HAVE CONVERGED.'/                  LIV07400
     2' ERROR ESTIMATE = BETAM*ABS(UM)'/                                LIV07410
     2' WHERE BETAM = BETA(MEV+1) AND UM = U(MEV).'/                    LIV07420
     3' U = UNIT EIGENVECTOR OF T WHERE T*U = EV*U AND EV = ISOLATED GOOLIV07430
     3D T-EIGENVALUE.'/                                                 LIV07440
     4' TMINGAP = GAP TO NEAREST DISTINCT EIGENVALUE OF T(1,MEV).'/     LIV07450
     5' TMINGAP .LT. 0. MEANS MINGAP IS DUE TO LEFT NEIGHBOR'/          LIV07460
     6' ERROR ESTIMATE L.T. 0 MEANS INVERSE ITERATION DID NOT CONVERGE'/LIV07470
     7' ------ END OF FILE 4 ERRINV -------------------------------'//) LIV07480
      GO TO 700                                                         LIV07490
C                                                                       LIV07500
  660 CONTINUE                                                          LIV07510
C                                                                       LIV07520
      IBB = IABS(IBMEV)                                                 LIV07530
      IF (IBMEV.LT.0) WRITE(6,670) MEV,IBB,BETA(IBB)                    LIV07540
  670 FORMAT(/' PROGRAM TERMINATES BECAUSE MEV REQUESTED = ',I6,' IS .GTLIV07550
     1',I6/' AT WHICH AN ABNORMALLY SMALL BETA = ' , E13.4,' OCCURRED'/)LIV07560
      GO TO 700                                                         LIV07570
C                                                                       LIV07580
  680 IF (NDIS.EQ.0.AND.ISTOP.GT.0) WRITE(6,690)                        LIV07590
  690 FORMAT(/' INTERVALS SPECIFIED FOR BISECT DID NOT CONTAIN ANY T-EIGLIV07600
     1ENVALUES'/' PROGRAM TERMINATES')                                  LIV07610
```

```
C                                                                       LIV07620
  700 CONTINUE                                                          LIV07630
C                                                                       LIV07640
      STOP                                                              LIV07650
C-----END OF LIVAL (INVERSES OF REAL SYMMETRIC MATRICES)--------------- LIV07660
      END                                                               LIV07670
```

4.3 LIVEC: Main Program, Eigenvector Computations

```
C-----LIVEC (EIGENVECTORS OF INVERSES OF REAL SYMMETRIC MATRICES)-------LIV00010
C  Authors:  Jane Cullum and Ralph A. Willoughby (deceased)              LIV00020
C            Los Alamos National Laboratory                             LIV00030
C            Los Alamos, New Mexico 87544                               LIV00040
C                                                                       LIV00050
C            E-mail:  cullumj@lanl.gov                                  LIV00060
C                                                                       LIV00070
C  These codes are copyrighted by the authors.  These codes             LIV00080
C  and modifications of them or portions of them are NOT to be          LIV00090
C  incorporated into any commercial codes or used for any other         LIV00100
C  commercial purposes such as consulting for other companies,          LIV00110
C  without legal agreements with the authors of these Codes.            LIV00120
C  If these Codes or portions of them are used in other scientific or   LIV00130
C  engineering research works the names of the authors of these codes   LIV00140
C  and appropriate references to their written work are to be           LIV00150
C  incorporated in the derivative works.                                LIV00160
C                                                                       LIV00170
C  This header is not to be removed from these codes.                   LIV00180
C                                                                       LIV00190
C         REFERENCE: Cullum and Willoughby, Chapters 1,2,3,4            LIV00191
C         Lanczos Algorithms for Large Symmetric Eigenvalue ComputationsLIV00192
C         VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in   LIV00193
C         Applied Mathematics, 2002. SIAM Publications,                 LIV00194
C         Philadelphia, PA. USA                                         LIV00195
C                                                                       LIV00200
C     CONTAINS MAIN PROGRAM FOR COMPUTING AN EIGENVECTOR CORRESPONDING  LIV00210
C     TO EACH OF A SET OF EIGENVALUES WHICH HAVE BEEN COMPUTED          LIV00220
C     ACCURATELY BY THE CORRESPONDING LANCZOS EIGENVALUE PROGRAM        LIV00230
C     (LIVAL) FOR FACTORED INVERSES OF REAL, SYMMETRIC MATRICES.        LIV00240
C     THIS PROGRAM COULD BE MODIFIED TO COMPUTE ADDITIONAL EIGENVECTORS LIV00250
C     FOR ANY EIGENVALUES WHICH ARE MULTIPLE EIGENVALUES OF THE         LIV00260
C     A-MATRIX.  THE AMOUNT OF ADDITIONAL COMPUTATION REQUIRED BY       LIV00270
C     SUCH A MODIFICATION DEPENDS UPON THE GIVEN A-MATRIX AND UPON      LIV00280
C     WHICH PORTION OF THE SPECTRUM IS INVOLVED.                        LIV00290
C                                                                       LIV00300
C     THESE LANCZOS EIGENVECTOR COMPUTATIONS ASSUME THAT EACH           LIV00310
C     EIGENVALUE THAT IS BEING CONSIDERED HAS CONVERGED AS AN           LIV00320
C     EIGENVALUE OF THE LANCZOS TRIDIAGONAL MATRICES.                   LIV00330
C                                                                       LIV00340
C     PFORT VERIFIER IDENTIFIED THE FOLLOWING NONPORTABLE               LIV00350
C     CONSTRUCTIONS                                                     LIV00360
C                                                                       LIV00370
C     1.  DATA/MACHEP/ STATEMENT                                        LIV00380
C     2.  ALL READ(5,*) STATEMENTS (FREE FORMAT)                        LIV00390
C     3.  FORMAT(20A4) USED WITH THE EXPLANATORY HEADER, EXPLAN         LIV00400
C     4.  HEXADECIMAL FORMAT (4Z20) USED FOR ALPHA/BETA FILES 1 AND 2.  LIV00410
C                                                                       LIV00420
C     IMPORTANT NOTE:  PROGRAM ALLOWS ENLARGEMENT OF THE ALPHA, BETA    LIV00430
C     ARRAYS.  IN PARTICULAR, IF ANY ONE OF THE EIGENVALUES SUPPLIED    LIV00440
C     IS T-SIMPLE AND NOT CLOSE TO A SPURIOUS EIGENVALUE, THE PROGRAM   LIV00450
C     REQUIRES THAT KMAX BE AT LEAST 11*MEV/8 + 12.  IF KMAX IS NOT     LIV00460
```

```
C     THIS LARGE, THEN THE PROGRAM RESETS KMAX TO THIS SIZE             LIV00470
C     AND EXTENDS THE ALPHA, BETA HISTORY IF REQUIRED.                  LIV00480
C     THUS, THE DIMENSIONS OF THE ALPHA AND BETA ARRAYS MUST BE         LIV00490
C     LARGE ENOUGH TO ALLOW FOR THIS POSSIBILITY.                       LIV00500
C     REMEMBER THAT THE BETA ARRAY, BETA(J), IS SUCH THAT               LIV00510
C     J = 1,..., KMAX+1,  SO IF THE KMAX USED BY THE PROGRAM            LIV00520
C     IS TO BE 3000, THEN BETA MUST BE OF LENGTH AT LEAST 3001.         LIV00530
C                                                                       LIV00540
C-----------------------------------------------------------------------LIV00550
      DOUBLE PRECISION  ALPHA(1000),BETA(1001)                          LIV00560
      DOUBLE PRECISION  V1(2200),V2(2200),VS(2200)                      LIV00570
      DOUBLE PRECISION  RITVEC(40000),TVEC(5000)                        LIV00580
      DOUBLE PRECISION  GOODA(50),GOODBI(50),EVNEW(50),TLAST(50)        LIV00590
      DOUBLE PRECISION  EVAL,EVALN,TOLN,TTOL,ERTOL,ALFA,BATA            LIV00600
      DOUBLE PRECISION  MULTOL,SCALE0,STUTOL,BTOL,LB,UB,S0,RNORME       LIV00610
      DOUBLE PRECISION  ONE,ZERO,MACHEP,EPSM,TEMP,SUM,SHIFT,SHIFTO      LIV00620
      DOUBLE PRECISION  RELTOL,ERROR,TERROR,BKMIN,ERRMIN                LIV00630
      REAL G(5000),AMINGP(50),TMINGP(50),BIERR(50),BIEVER(50),BIERRG(50)LIV00640
      REAL TERR(50),RNORM(50),TBETA(50),BIMING(50)                      LIV00650
      REAL EXPLAN(20)                                                   LIV00660
      INTEGER  MP(50),IDELTA(50)                                        LIV00670
      INTEGER  M1(50),M2(50),MA(50),ML(50),MINT(50),MFIN(50)            LIV00680
      INTEGER  SVSEED,SVSOLD,RHSEED                                     LIV00690
      INTEGER  MBOUND,NTVCON,SVTVEC,TVSTOP,LVCONT,ERCONT,TFLAG          LIV00700
      DOUBLE PRECISION  FINPRO                                          LIV00710
      DOUBLE PRECISION  DABS, DMAX1, DSQRT, DFLOAT                      LIV00720
      REAL ABS                                                          LIV00730
      INTEGER  IABS                                                     LIV00740
      EXTERNAL BSOLV                                                    LIV00750
C-----------------------------------------------------------------------LIV00760
      DATA MACHEP/Z3410000000000000/                                    LIV00770
      EPSM = 2.D0*MACHEP                                                LIV00780
C-----------------------------------------------------------------------LIV00790
C     ARRAYS MUST BE DIMENSIONED AS FOLLOWS:                            LIV00800
C     1.  ALPHA:  >= KMAXN,  BETA: >= (KMAXN+1) WHERE KMAXN, THE        LIV00810
C                 LARGEST SIZE T-MATRIX CONSIDERED BY THE PROGRAM,      LIV00820
C                 IS THE LARGER OF THE SIZE OF THE ALPHA, BETA HISTORY  LIV00830
C                 PROVIDED ON FILE 2 (IF ANY ) AND THE SIZE WHICH THE   LIV00840
C                 PROGRAM SPECIFIES INTERNALLY, THIS LATTER IS ALWAYS   LIV00850
C                 < = 11*MEV / 8  +  12, WHERE MEV IS THE SIZE          LIV00860
C                 T-MATRIX THAT WAS USED IN THE CORRESPONDING EIGENVALUE LIV00870
C                 COMPUTATIONS.                                         LIV00880
C     2.  V1:  >= MAX(N,KMAX)                                           LIV00890
C     3.  V2, VS:   >= N                                                LIV00900
C     4.  G:  >= MAX(N,KMAX)                                            LIV00910
C     5.  RITVEC:  >= N*NGOOD, WHERE NGOOD IS NUMBER OF EIGENVALUES     LIV00920
C                  SUPPLIED TO THIS PROGRAM.                            LIV00930
C     6.  TVEC:  >= CUMULATIVE LENGTH OF ALL THE T-EIGENVECTORS         LIV00940
C         NEEDED TO GENERATE THE DESIRED RITZ VECTORS.  AN EDUCATED     LIV00950
C         GUESS AT AN APPROPRIATE LENGTH CAN BE OBTAINED BY RUNNING THE LIV00960
C         PROGRAM WITH THE FLAG MBOUND = 1 AND MULTIPLYING THE          LIV00970
C         RESULTING SIZE BY 5/4.                                        LIV00980
C     7.  GOODA, GOODBI, EVNEW, AMINGP, TMINGP, TERR, RNORM,            LIV00990
C         TBETA, TLAST, BIERR, BIERRG, MP, MA, M1, M2, MINT,            LIV01000
C         MFIN AND IDELTA MUST BE OF DIMENSION AT LEAST NGOOD.          LIV01010
```

```
C                                                                       LIV01020
C-----------------------------------------------------------------------LIV01030
C     OUTPUT HEADER                                                     LIV01040
      WRITE(6,10)                                                       LIV01050
   10 FORMAT(/' LANCZOS PROCEDURE FOR FACTORED INVERSES OF REAL SYMMETRILIV01060
     1C MATRICES'/'      COMPUTE EIGENVECTORS')                         LIV01070
C                                                                       LIV01080
C     SET PROGRAM PARAMETERS                                            LIV01090
C     USER MUST NOT MODIFY SCALE0                                       LIV01100
      SCALE0 = 5.0D0                                                    LIV01110
      ZER0 = 0.0D0                                                      LIV01120
      ONE = 1.0D0                                                       LIV01130
      MPMIN = -1000                                                     LIV01140
C     CONVERGENCE TOLERANCE FOR T-EIGENVECTORS FOR RITZ COMPUTATIONS    LIV01150
      ERTOL = 1.D-10                                                    LIV01160
C                                                                       LIV01170
C     READ USER-SPECIFIED PARAMETER FROM INPUT FILE 5 (FREE FORMAT)     LIV01180
C                                                                       LIV01190
C     READ USER-PROVIDED HEADER FOR RUN                                 LIV01200
      READ(5,20) EXPLAN                                                 LIV01210
      WRITE(6,20) EXPLAN                                                LIV01220
   20 FORMAT(20A4)                                                      LIV01230
C                                                                       LIV01240
C     READ IN MATN0 = MATRIX/RUN IDENTIFICATION NUMBER AND              LIV01250
C     N = ORDER OF A-MATRIX                                             LIV01260
C     READ IN SCALE (S0) AND SHIFT (SHIFT) APPLIED TO GIVEN             LIV01270
C     MATRIX AND FLAG JPERM.  JPERM = (0,1):                            LIV01280
C     JPERM = 1 MEANS THAT A-MATRIX HAS BEEN PERMUTED.                  LIV01290
C                                                                       LIV01300
      READ(5,20) EXPLAN                                                 LIV01310
      READ(5,*) MATN0,N,S0,SHIFT,JPERM                                  LIV01320
C                                                                       LIV01330
C     READ IN THE MAXIMUM PERMISSIBLE DIMENSIONS FOR THE TVEC ARRAY     LIV01340
C     (MDIMTV), FOR THE RITVEC ARRAY (MDIMRV), AND FOR THE BETA         LIV01350
C     ARRAY (MBETA).                                                    LIV01360
C                                                                       LIV01370
      READ(5,20) EXPLAN                                                 LIV01380
      READ(5,*) MDIMTV, MDIMRV, MBETA                                   LIV01390
C                                                                       LIV01400
C     READ IN RELATIVE TOLERANCE (RELTOL) USED IN DETERMINING           LIV01410
C     APPROPRIATE SIZES FOR THE T-MATRICES USED IN THE EIGENVECTOR      LIV01420
C     COMPUTATIONS                                                      LIV01430
C                                                                       LIV01440
      READ(5,20) EXPLAN                                                 LIV01450
      READ(5,*) RELTOL                                                  LIV01460
C                                                                       LIV01470
C     SET FLAGS TO 0 OR 1:                                              LIV01480
C     MBOUND = 1:  PROGRAM TERMINATES AFTER COMPUTING 1ST GUESSES       LIV01490
C                  ON APPROPRIATE T-SIZES FOR USE IN THE RITZ VECTOR    LIV01500
C                  COMPUTATIONS                                         LIV01510
C     NTVCON = 0:  PROGRAM TERMINATES IF THE TVEC ARRAY IS NOT          LIV01520
C                  LARGE ENOUGH TO HOLD ALL THE T-EIGENVECTORS REQUIRED.LIV01530
C     SVTVEC = 0:  THE T-EIGENVECTORS ARE NOT WRITTEN TO FILE 11        LIV01540
C                  UNLESS TVSTOP = 1                                    LIV01550
C     SVTVEC = 1:  WRITE THE T-EIGENVECTORS TO FILE 11.                 LIV01560
```

```
C     TVSTOP = 1:  PROGRAM TERMINATES AFTER COMPUTING THE              LIV01570
C                  T-EIGENVECTORS                                      LIV01580
C     LVCONT = 0:  PROGRAM TERMINATES IF THE NUMBER OF T-EIGENVECTORS  LIV01590
C                  COMPUTED IS NOT EQUAL TO THE NUMBER OF RITZ         LIV01600
C                  VECTORS REQUESTED.                                  LIV01610
C     ERCONT = 0:  MEANS FOR ANY GIVEN EIGENVALUE, A RITZ VECTOR       LIV01620
C                  WILL NOT BE COMPUTED FOR THAT EIGENVALUE UNLESS     LIV01630
C                  A T-EIGENVECTOR HAS BEEN IDENTIFIED WITH A LAST     LIV01640
C                  COMPONENT WHICH SATISFIES THE SPECIFIED             LIV01650
C                  CONVERGENCE CRITERION.                              LIV01660
C     ERCONT = 1:  MEANS FOR ANY GIVEN EIGENVALUE, A RITZ VECTOR       LIV01670
C                  WILL BE COMPUTED.  IF A T-EIGENVECTOR CANNOT        LIV01680
C                  BE IDENTIFIED WHICH SATISFIES THE LAST              LIV01690
C                  COMPONENT CRITERION, THEN THE PROGRAM WILL          LIV01700
C                  USE THE T-VECTOR THAT CAME CLOSEST TO               LIV01710
C                  SATISFYING THE CRITERION                            LIV01720
C     IWRITE = 1:  EXTENDED OUTPUT OF INTERMEDIATE COMPUTATIONS        LIV01730
C                  IS WRITTEN TO FILE 6                                LIV01740
C     IREAD = 0:   ALPHA/BETA FILE IS REGENERATED.                     LIV01750
C     IREAD = 1:   ALPHA/BETA FILE USED IN EIGENVALUE COMPUTATIONS     LIV01760
C                  IS READ IN AND EXTENDED IF NECESSARY.  IN BOTH      LIV01770
C                  CASES IREAD = 0 OR 1, THE LANCZOS VECTORS ARE       LIV01780
C                  ALWAYS REGENERATED FOR THE RITZ VECTOR              LIV01790
C                  COMPUTATIONS                                        LIV01800
C                                                                      LIV01810
      READ(5,20) EXPLAN                                                LIV01820
      READ(5,*) MBOUND,NTVCON,SVTVEC,IREAD                             LIV01830
C                                                                      LIV01840
      READ(5,20) EXPLAN                                                LIV01850
      READ(5,*) TVSTOP,LVCONT,ERCONT,IWRITE                            LIV01860
      IF (TVSTOP.EQ.1) SVTVEC = 1                                      LIV01870
C                                                                      LIV01880
C     READ IN SEED (RHSEED) FOR GENERATING RANDOM STARTING VECTOR      LIV01890
C     FOR THE INVERSE ITERATION ON THE T-MATRICES.                     LIV01900
C                                                                      LIV01910
      READ(5,20) EXPLAN                                                LIV01920
      READ(5,*) RHSEED                                                 LIV01930
C                                                                      LIV01940
C-----------------------------------------------------------------------LIV01950
C     INITIALIZE THE ARRAYS THAT DEFINE THE FACTORIZATION OF           LIV01960
C     THE B-MATRIX AND PASS THE STORAGE LOCATIONS OF THESE ARRAYS      LIV01970
C     TO THE SUBROUTINE BSOLV.                                         LIV01980
C                                                                      LIV01990
      CALL USPEC(N,MATNO)                                              LIV02000
C-----------------------------------------------------------------------LIV02010
C     MASK UNDERFLOW AND OVERFLOW                                      LIV02020
C                                                                      LIV02030
      CALL MASK                                                        LIV02040
C-----------------------------------------------------------------------LIV02050
C     WRITE RUN PARAMETERS OUT TO FILE 6                               LIV02060
C                                                                      LIV02070
      WRITE(6,30) MATNO,N,JPERM                                        LIV02080
   30 FORMAT(/4X,'MATRIX IDENTIFICATION NO.',4X,'SIZE OF A-MATRIX',4X, LIV02090
     1'JPERM'/I29,I21,I9)                                              LIV02100
C                                                                      LIV02110
```

```
      WRITE(6,40) SO,SHIFT                                              LIV02120
   40 FORMAT(/4X,'SCALE APPLIED TO MATRIX',4X,'SHIFT APPLIED TO MATRIX'/LIV02130
     1E27.4,E27.4)                                                      LIV02140
C                                                                       LIV02150
      WRITE(6,50) MBOUND,NTVCON,SVTVEC,IREAD                            LIV02160
   50 FORMAT(/3X,'MBOUND',3X,'NTVCON',3X,'SVTVEC',3X,'IREAD'/3I9,I8/)   LIV02170
C                                                                       LIV02180
      WRITE(6,60) TVSTOP,LVCONT,ERCONT,IWRITE                           LIV02190
   60 FORMAT(3X,'TVSTOP',3X,'LVCONT',3X,'ERCONT',3X,'IWRITE'/4I9)       LIV02200
C                                                                       LIV02210
      WRITE(6,70) MDIMTV,MDIMRV,MBETA                                   LIV02220
   70 FORMAT(/3X,'MDIMTV',3X,'MDIMRV',3X,'MBETA'/2I9,I8)                LIV02230
C                                                                       LIV02240
      WRITE(6,80) RELTOL,RHSEED                                         LIV02250
   80 FORMAT(/7X,'RELTOL',3X,'RHSEED'/E13.4,I9)                         LIV02260
C                                                                       LIV02270
C     FROM FILE 3 READ IN THE NUMBER OF EIGENVALUES (NGOOD) FOR WHICH   LIV02280
C     EIGENVECTORS ARE REQUESTED, THE ORDER (MEV) OF THE LANCZOS        LIV02290
C     TRIDIAGONAL MATRIX USED IN COMPUTING THESE EIGENVALUES, THE       LIV02300
C     ORDER (NOLD) OF THE USER-SPECIFIED MATRIX USED IN THE EIGENVALUE  LIV02310
C     COMPUTATIONS, THE SEED (SVSEED) USED FOR GENERATING THE STARTING  LIV02320
C     VECTOR THAT WAS USED IN THOSE LANCZOS EIGENVALUE COMPUTATIONS,    LIV02330
C     AND THE MATRIX/RUN IDENTIFICATION NUMBER (MATOLD) USED IN THOSE   LIV02340
C     COMPUTATIONS.  ALSO READ IN THE NUMBER (NDIS) OF DISTINCT         LIV02350
C     EIGENVALUES OF T(1,MEV) THAT WERE COMPUTED BUT THIS VALUE IS      LIV02360
C     NOT USED IN THE EIGENVECTOR COMPUTATIONS.                         LIV02370
C                                                                       LIV02380
      READ(3,90) NGOOD,NDIS,MEV,NOLD,SVSEED,MATOLD                      LIV02390
   90 FORMAT(4I6,I12,I8)                                                LIV02400
C                                                                       LIV02410
C     READ IN THE MULTIPLICITY TOLERANCE USED IN THE BISEC SUBROUTINE   LIV02420
C     DURING THE COMPUTATION OF THE GIVEN EIGENVALUES.                  LIV02430
C     ALSO READ IN THE FLAG IB.  IF IB < 0, THEN SOME BETA(I) IN THE    LIV02440
C     T-MATRIX FILE PROVIDED ON FILE 2 FAILED THE ORTHOGONALITY         LIV02450
C     TEST IN THE TNORM SUBROUTINE.  USER SHOULD NOTE THAT THIS VECTOR  LIV02460
C     PROGRAM PROCEEDS INDEPENDENTLY OF THE SIZE OF THE BETA USED.      LIV02470
C                                                                       LIV02480
      READ(3,100) MULTOL,IB,BTOL,SHIFTO                                 LIV02490
  100 FORMAT(E20.12,I6,2E10.3)                                          LIV02500
C                                                                       LIV02510
      TEMP = DFLOAT(MEV+1000)                                           LIV02520
      TTOL = MULTOL/TEMP                                                LIV02530
C                                                                       LIV02540
      WRITE(6,110) MULTOL,TTOL                                          LIV02550
  110 FORMAT(/' T-MULTIPLICITY TOLERANCE USED IN THE EIGENVALUE COMPUTATLIV02560
     1IONS WAS',E13.4/' SCALED MACHINE EPSILON TTOL IS',E13.4)          LIV02570
C                                                                       LIV02580
C     CONTINUE WRITE TO FILE 6 OF THE PARAMETERS FOR THIS RUN           LIV02590
C                                                                       LIV02600
      NG = NGOOD                                                        LIV02610
      WRITE(6,120)NG,NDIS,MEV,NOLD,MATOLD,SVSEED,IB,MULTOL,BTOL,SHIFTO  LIV02620
  120 FORMAT(/' EIGENVALUES ARE READ IN FROM FILE 3.  THE HEADER IS'/   LIV02630
     1 4X,'NG',2X,'NDIS',3X,'MEV',2X,'NOLD',2X,'MATOLD',6X,'SVSEED'/    LIV02640
     1 4I6,I8,I12/                                                      LIV02650
     1 6X,'IB',6X,'MULTOL',8X,'BTOL',6X,'SHIFTO'/                       LIV02660
```

```
     1 I8,E12.3,E12.3,E12.3/)                                          LIV02670
C                                                                      LIV02680
C     IS THE ARRAY RITVEC LONG ENOUGH TO HOLD ALL OF THE DESIRED       LIV02690
C     RITZ VECTORS (APPROXIMATE EIGENVECTORS)?                         LIV02700
C                                                                      LIV02710
      NMAX = NGOOD*N                                                   LIV02720
      IF(MBOUND.EQ.1) GO TO 130                                        LIV02730
      IF(TVSTOP.NE.1.AND.NMAX.GT.MDIMRV) GO TO 1430                    LIV02740
C                                                                      LIV02750
C     CHECK THAT THE ORDER N AND THE MATRIX IDENTIFICATION NUMBER      LIV02760
C     MATNO SPECIFIED BY THE USER AGREE WITH THOSE READ IN FROM FILE 3. LIV02770
C                                                                      LIV02780
  130 ITEMP = (NOLD-N)**2+(MATOLD-MATNO)**2                            LIV02790
      IF (ITEMP.NE.0.OR.SHIFTO.NE.SHIFT) GO TO 1450                    LIV02800
C                                                                      LIV02810
C     READ IN FROM FILE 3, THE T-MULTIPLICITIES OF THE EIGENVALUES     LIV02820
C     WHOSE EIGENVECTORS ARE TO BE COMPUTED, THE VALUES OF THESE       LIV02830
C     EIGENVALUES AND THEIR MINIMAL GAPS AS EIGENVALUES OF THE         LIV02840
C     USER-SPECIFIED MATRIX AND AS EIGENVALUES OF THE T-MATRIX.        LIV02850
C                                                                      LIV02860
      READ(3,20) EXPLAN                                                LIV02870
      READ(3,140) (MP(J),GOODBI(J),BIMING(J),AMINGP(J),TMINGP(J),      LIV02880
     1 J = 1,NGOOD)                                                    LIV02890
  140 FORMAT(5X,I5,E25.16,3E10.3)                                      LIV02900
C                                                                      LIV02910
C                                                                      LIV02920
      DO 150 J=1,NGOOD                                                 LIV02930
  150 GOODA(J) = (ONE/GOODBI(J) - SHIFT)/S0                            LIV02940
C                                                                      LIV02950
      WRITE(6,160) (J,GOODA(J),MP(J),GOODBI(J), J=1,NGOOD)             LIV02960
  160 FORMAT(/' EIGENVALUES READ IN, T-MULTIPLICITIES'/                LIV02970
     1 4X,' J ',5X,' A-EIGENVALUE',6X,'TMULT',3X,'B-INVERSE EIGENVALUE'/LIV02980
     1(I6,E25.16,I4,E25.16))                                           LIV02990
      WRITE(6,170) (J,GOODBI(J),TMINGP(J),BIMING(J), J=1,NGOOD)        LIV03000
  170 FORMAT(/' B(INVERSE) EIGENVALUES READ IN, T-GAPS AND B(INVERSE)-GALIV03010
     1PS'/4X,' J ',3X,'B-INVERSE EIGENVALUE',6X,'  TMINGAP ',6X,       LIV03020
     1' BIMINGAP '/(I6,E25.16,2E15.4))                                 LIV03030
      WRITE(6,180) (J,GOODA(J),AMINGP(J), J=1,NGOOD)                   LIV03040
  180 FORMAT(/' A-EIGENVALUES READ IN AND A-GAPS'/                     LIV03050
     1 4X,' J ',5X,'A-EIGENVALUE',10X,'  AMINGAP '                     LIV03060
     1/(I6,E25.16,E15.4))                                              LIV03070
C                                                                      LIV03080
C     READ IN ERROR ESTIMATES                                          LIV03090
      WRITE(6,210) MEV,SVSEED                                          LIV03100
C     CHECK WHETHER OR NOT THERE ARE ANY T-ISOLATED EIGENVALUES IN     LIV03110
C     THE EIGENVALUES PROVIDED                                         LIV03120
      DO 190 J=1,NGOOD                                                 LIV03130
      IF(MP(J).EQ.1) GO TO 200                                         LIV03140
  190 CONTINUE                                                         LIV03150
      GO TO 230                                                        LIV03160
  200 READ(4,20) EXPLAN                                                LIV03170
      READ(4,20) EXPLAN                                                LIV03180
      READ(4,20) EXPLAN                                                LIV03190
  210 FORMAT(/' THESE EIGENVALUES WERE COMPUTED USING A T-MATRIX OF    LIV03200
     1ORDER ',I5/' AND SEED FOR RANDOM NUMBER GENERATOR =',I12)        LIV03210
```

```
      READ(4,220) NISO                                                  LIV03220
  220 FORMAT(18X,I6)                                                    LIV03230
      READ(4,20) EXPLAN                                                 LIV03240
      READ(4,20) EXPLAN                                                 LIV03250
      READ(4,20) EXPLAN                                                 LIV03260
  230 DO 260 J=1,NGOOD                                                  LIV03270
      BIERR(J) = 0.D0                                                   LIV03280
      IF(MP(J).NE.1) GO TO 260                                          LIV03290
      READ(4,240) EVAL, BIERR(J)                                        LIV03300
  240 FORMAT(10X,E25.16,E14.3)                                          LIV03310
      IF(DABS(EVAL - GOODBI(J)).LT.1.D-10) GO TO 260                    LIV03320
      WRITE(6,250) EVAL,GOODBI(J)                                       LIV03330
  250 FORMAT(' PROBLEM WITH READ IN OF ERROR ESTIMATES'/' EIGENVALUE REALIV03340
     1D IN',E20.12,' DOES NOT MATCH GOODBI(J) ='/E20.12)                LIV03350
      GO TO 1670                                                        LIV03360
C                                                                       LIV03370
  260 CONTINUE                                                          LIV03380
C                                                                       LIV03390
      WRITE(6,270) (J,GOODBI(J),BIERR(J), J=1,NGOOD)                    LIV03400
  270 FORMAT(' B(INVERSE) ERROR ESTIMATES '/4X,' J',5X,'EIGENVALUE',10X LIV03410
     1,'ESTIMATE'/(I6,E20.12,E14.3))                                    LIV03420
C                                                                       LIV03430
      IF(IREAD.EQ.0)  GO TO 370                                         LIV03440
C                                                                       LIV03450
C     READ IN THE SIZE OF THE T-MATRIX PROVIDED ON FILE 2.  READ IN     LIV03460
C     THE ORDER OF THE USER-SPECIFIED MATRIX , THE SEED FOR THE         LIV03470
C     RANDOM NUMBER GENERATOR, AND THE MATRIX/TEST IDENTIFICATION       LIV03480
C     NUMBER THAT WERE USED IN THE LANCZOS EIGENVALUE COMPUTATIONS.     LIV03490
C     IF FLAG IREAD = 0, REGENERATE ALPHA,BETA ARRAYS                   LIV03500
C                                                                       LIV03510
      READ(2,280) KMAX,NOLD,SVSOLD,MATOLD,SHIFTO                        LIV03520
  280 FORMAT(2I6,I12,I8,E13.4)                                          LIV03530
C                                                                       LIV03540
      WRITE(6,290) KMAX,NOLD,SVSOLD,MATOLD,SHIFTO                       LIV03550
  290 FORMAT(/' READ IN THE T-MATRICES STORED ON FILE 2'/' FILE 2 HEADERLIV03560
     1 IS'/2X,'KMAX',2X,'NOLD',6X,'SVSOLD',2X,'MATOLD',4X,'SHIFTO'/     LIV03570
     1 2I6,I12,I8,E10.3/)                                               LIV03580
C                                                                       LIV03590
C     CHECK THAT THE ORDER, THE MATRIX/TEST IDENTIFICATION NUMBER       LIV03600
C     AND THE SEED FOR THE RANDOM NUMBER GENERATOR USED IN THE          LIV03610
C     LANCZOS COMPUTATIONS THAT GENERATED THE ALPHA,BETA FILE           LIV03620
C     BEING USED AGREE WITH WHAT THE USER HAS SPECIFIED.                LIV03630
      IF (NOLD.NE.N.OR.MATOLD.NE.MATNO.OR.SVSOLD.NE.SVSEED) GO TO 1470  LIV03640
C                                                                       LIV03650
      KMAX1 = KMAX + 1                                                  LIV03660
C                                                                       LIV03670
C     READ IN THE T-MATRICES FROM FILE 2.  THESE ARE USED TO GENERATE   LIV03680
C     THE T-EIGENVECTORS THAT WILL BE USED IN THE RITZ VECTOR           LIV03690
C     COMPUTATIONS.  ALPHA,BETA MUST BE STORED IN MACHINE FORMAT        LIV03700
C     ((4Z20) ON IBM/3081)                                              LIV03710
C                                                                       LIV03720
      READ(2,300) (ALPHA(J), J=1,KMAX)                                  LIV03730
      READ(2,300) (BETA(J), J=1,KMAX1)                                  LIV03740
  300 FORMAT(4Z20)                                                      LIV03750
C                                                                       LIV03760
```

```
      READ(2,300) (V1(J), J=1,N)                                        LIV03770
      READ(2,300) (V2(J), J=1,N)                                        LIV03780
C                                                                       LIV03790
C     ENLARGE KMAX IF THE SIZE AT WHICH THE EIGENVALUE                  LIV03800
C     COMPUTATIONS WERE PERFORMED IS ESSENTIALLY KMAX AND               LIV03810
C     THERE IS AT LEAST ONE EIGENVALUE THAT IS T-SIMPLE AND             LIV03820
C     T-ISOLATED IN THE SENSE THAT IF ITS NEAREST NEIGHBOR IS           LIV03830
C     TOO CLOSE THEN THAT NEIGHBOR IS A GOOD T-EIGENVALUE.              LIV03840
      DO 310 J = 1,NGOOD                                                LIV03850
      IF(MP(J).EQ.1) GO TO 330                                          LIV03860
  310 CONTINUE                                                          LIV03870
      WRITE(6,320)                                                      LIV03880
  320 FORMAT(/' ALL EIGENVALUES USED ARE T-MULTIPLE OR CLOSE TO SPURIOUSLIV03890
     1 T-EIGENVALUES'/' SO DO NOT CHANGE KMAX')                         LIV03900
      IF(KMAX.LT.MEV) GO TO 1490                                        LIV03910
      GO TO 350                                                         LIV03920
C                                                                       LIV03930
  330 KMAXN= 11*MEV/8 + 12                                              LIV03940
      IF(MBETA.LE.KMAXN) GO TO 1650                                     LIV03950
      IF(KMAX.GE.KMAXN )  GO TO 350                                     LIV03960
      WRITE(6,340) KMAX, KMAXN                                          LIV03970
  340 FORMAT(' ENLARGE KMAX FROM ',I6,' TO ',I6)                        LIV03980
      MOLD1 = KMAX + 1                                                  LIV03990
      KMAX = KMAXN                                                      LIV04000
      GO TO 420                                                         LIV04010
C                                                                       LIV04020
  350 WRITE(6,360) KMAX                                                 LIV04030
  360 FORMAT(/' T-MATRICES HAVE BEEN READ IN FROM FILE 2'/' THE LARGEST LIV04040
     1SIZE T-MATRIX ALLOWED IS',I6/)                                    LIV04050
C                                                                       LIV04060
      IF(IREAD.EQ.1) GO TO 440                                          LIV04070
C                                                                       LIV04080
C     REGENERATE THE ALPHA AND BETA                                     LIV04090
C                                                                       LIV04100
  370 MOLD1 = 1                                                         LIV04110
C                                                                       LIV04120
      DO 380 J = 1,NGOOD                                                LIV04130
      IF(MP(J).EQ.1) GO TO 400                                          LIV04140
  380 CONTINUE                                                          LIV04150
      KMAX = MEV + 12                                                   LIV04160
      WRITE(6,390) KMAX                                                 LIV04170
  390 FORMAT(/' ALL EIGENVALUES FOR WHICH EIGENVECTORS ARE TO BE COMPUTELIV04180
     1D ARE EITHER T-MULTIPLE OR CLOSE TO'/' A SPURIOUS T-EIGENVALUE. THLIV04190
     1EREFORE SET KMAX = MEV + 12 = ',I7)                               LIV04200
      GO TO 420                                                         LIV04210
C                                                                       LIV04220
  400 KMAXN = 11*MEV/8 + 12                                             LIV04230
      IF(MBETA.LE.KMAXN) GO TO 1650                                     LIV04240
      WRITE(6,410) KMAXN                                                LIV04250
  410 FORMAT(' SET KMAX EQUAL TO ',I6)                                  LIV04260
      KMAX = KMAXN                                                      LIV04270
C                                                                       LIV04280
  420 WRITE(6,430) MOLD1,KMAX                                           LIV04290
  430 FORMAT(/' LANCZS SUBROUTINE GENERATES ALPHA(J), BETA(J+1), J =',  LIV04300
     1 I6,' TO ', I6/)                                                  LIV04310
```

```
C                                                                       LIV04320
C-----------------------------------------------------------------------LIV04330
C                                                                       LIV04340
      CALL LANCZS(BSOLV,ALPHA,BETA,V1,V2,VS,G,KMAX,MOLD1,N,SVSEED)      LIV04350
C                                                                       LIV04360
C-----------------------------------------------------------------------LIV04370
C                                                                       LIV04380
  440 CONTINUE                                                          LIV04390
C                                                                       LIV04400
C     THE SUBROUTINE STURMI DETERMINES THE SMALLEST SIZE T-MATRIX FOR   LIV04410
C     WHICH THE EIGENVALUE IN QUESTION IS AN EIGENVALUE (TO WITHIN A    LIV04420
C     GIVEN TOLERANCE) AND IF POSSIBLE THE SMALLEST SIZE T-MATRIX       LIV04430
C     FOR WHICH IT IS A DOUBLE EIGENVALUE (TO WITHIN THE SAME           LIV04440
C     TOLERANCE).  THE SIZE T-MATRIX USED IN THE EIGENVECTOR            LIV04450
C     COMPUTATIONS IS THEN DETERMINED BY LOOPING ON SIZE OF THE         LIV04460
C     T-EIGENVECTORS, USING THE VALUES FROM STURMI TO DETERMINE         LIV04470
C     FIRST GUESSES AT THE APPROPRIATE T-SIZES.                         LIV04480
C                                                                       LIV04490
C                                                                       LIV04500
      STUTOL = SCALE0*MULTOL                                            LIV04510
      IF(IWRITE.EQ.1) WRITE(6,450)                                      LIV04520
  450 FORMAT(' FROM STURMI')                                            LIV04530
      DO 490 J = 1,NGOOD                                                LIV04540
      EVAL = GOODBI(J)                                                  LIV04550
C     COMPUTE THE TOLERANCES USED BY STURMI TO DETERMINE AN INTERVAL    LIV04560
C     CONTAINING THE EIGENVALUE EVAL.                                   LIV04570
      TEMP = DABS(EVAL)*RELTOL                                          LIV04580
      TOLN = DMAX1(TEMP,STUTOL)                                         LIV04590
C                                                                       LIV04600
C-----------------------------------------------------------------------LIV04610
C                                                                       LIV04620
      CALL STURMI(ALPHA,BETA,EVAL,TOLN,EPSM,KMAX,MK1,MK2,IC,IWRITE)     LIV04630
C                                                                       LIV04640
C-----------------------------------------------------------------------LIV04650
C                                                                       LIV04660
C     STORE THE COMPUTED ORDERS OF T-MATRICES FOR LATER PRINTOUT        LIV04670
      M1(J) = MK1                                                       LIV04680
      M2(J) = MK2                                                       LIV04690
      ML(J) = (MK1 + 3*MK2)/4                                           LIV04700
      IF(MK2.EQ.KMAX)  ML(J) = KMAX                                     LIV04710
C                                                                       LIV04720
      IF(IC.GT.0) GO TO 470                                             LIV04730
C     IC = 0 MEANS THERE WAS NO T-EIGENVALUE IN THE DESIGNATED INTERVAL LIV04740
C     BY T-SIZE KMAX.  THIS MEANS THAT THE EIGENVALUE PROVIDED HAS      LIV04750
C     NOT YET CONVERGED SO ITS EIGENVECTOR SHOULD NOT BE COMPUTED.      LIV04760
      WRITE(6,460) J,GOODBI(J),MK1,MK2                                  LIV04770
  460 FORMAT(I6,'TH EIGENVALUE',E20.12,' HAS NOT CONVERGED '/           LIV04780
     1' SO DO NOT COMPUTE ANY T-EIGENVECTOR OR RITZ VECTOR FOR IT'      LIV04790
     1/' MK1 AND MK2 FOR THIS EIGENVALUE WERE',2I6)                     LIV04800
      MP(J) = MPMIN                                                     LIV04810
      MA(J) = -2*KMAX                                                   LIV04820
      GO TO 490                                                         LIV04830
C     COMPUTE AN APPROPRIATE SIZE T-MATRIX FOR THE GIVEN EIGENVALUE.    LIV04840
  470 IF(M2(J).EQ.KMAX) GO TO 480                                       LIV04850
C     M1 AND M2 WERE BOTH DETERMINED                                    LIV04860
```

```
      MA(J) = (3*M1(J) + M2(J))/4 + 1                                   LIV04870
      GO TO 490                                                         LIV04880
C     M2 NOT DETERMINED                                                 LIV04890
  480 MA(J) = (5*M1(J))/4 + 1                                           LIV04900
C                                                                       LIV04910
  490 CONTINUE                                                          LIV04920
C                                                                       LIV04930
      IF (IWRITE.EQ.1) WRITE(6,500) (MA(JJ), JJ=1,NGOOD)                LIV04940
  500 FORMAT(/' 1ST GUESS AT APPROPRIATE SIZE T-MATRICES'/              LIV04950
     1 ' ACTUAL VALUES WILL PROBABLY BE 1/4 AGAIN AS MUCH'/(13I6))      LIV04960
C                                                                       LIV04970
C     PRINT OUT TO FILE 10 1ST GUESSES AT SIZES OF THE T-MATRICES TO    LIV04980
C     BE USED IN THE EIGENVECTOR COMPUTATIONS.                          LIV04990
C     ACTUAL VALUES USED MAY BE 1/4 OR MORE LARGER THAN THESE VALUES.   LIV05000
      WRITE(10,510) N,KMAX                                              LIV05010
  510 FORMAT(2I8,' = ORDER OF USER MATRIX AND MAX ORDER OF T(1,MEV)')   LIV05020
C                                                                       LIV05030
      WRITE(10,520)                                                     LIV05040
  520 FORMAT(/' 1ST GUESS AT APPROPRIATE SIZE T-MATRICES'/              LIV05050
     1 ' ACTUAL VALUES WILL PROBABLY BE 1/4 AGAIN AS MUCH'/)            LIV05060
C                                                                       LIV05070
      WRITE(10,530)                                                     LIV05080
  530 FORMAT(4X,'J',7X,'GOODBI(J)',4X,'M1(J)',1X,'M2(J)',1X,'MA(J)')    LIV05090
C                                                                       LIV05100
      WRITE(10,540) (J,GOODBI(J),M1(J),M2(J), MA(J), J=1,NGOOD)         LIV05110
  540 FORMAT(I5,E19.12,3I6)                                             LIV05120
C                                                                       LIV05130
      IF(MBOUND.EQ.1) WRITE(10,550)                                     LIV05140
  550 FORMAT(/' EV = GOODBI(J) IS A GOOD EIGENVALUE OF T(1,MEV)'/       LIV05150
     1 ' M1 = SMALLEST VALUE OF M SUCH THAT T(1,M) HAS AT LEAST'/       LIV05160
     1 '      ONE EIGENVALUE IN THE INTERVAL (EV-TOLN,EV+TOLN)'/        LIV05170
     1 ' TOLN(J) = DMAX1(GOODBI(J)*RELTOL, SCALE0*MULTOL)'/             LIV05180
     1 ' M2 = SMALLEST M (IF ANY) SUCH THAT IN THE ABOVE INTERVAL'/     LIV05190
     1 '      T(1,M) HAS AT LEAST TWO EIGENVALUES '/                    LIV05200
     1 ' INITIAL VALUE OF MA(J) IS CHOSEN HEURISTICALLY'/               LIV05210
     1 ' PROGRAM LOOPS ON SIZE OF T-MATRIX TO GET APPROPRIATE SIZE'/    LIV05220
     1 ' END OF SIZES OF T-MATRICES FILE 10'///)                        LIV05230
C                                                                       LIV05240
C                                                                       LIV05250
C     TERMINATE AFTER COMPUTING 1ST GUESSES AT  SIZES OF THE            LIV05260
C     T-MATRICES REQUIRED FOR THE GIVEN EIGENVALUES?                    LIV05270
      IF(MBOUND.EQ.1) GO TO 1510                                        LIV05280
C                                                                       LIV05290
C                                                                       LIV05300
C     WILL THERE BE ROOM FOR ALL OF THE REQUESTED T-EIGENVECTORS?       LIV05310
      MTOL = 0                                                          LIV05320
      DO 560 J = 1,NGOOD                                                LIV05330
      IF(MP(J).EQ.MPMIN) GO TO 560                                      LIV05340
      MTOL = MTOL + IABS(MA(J))                                         LIV05350
  560 CONTINUE                                                          LIV05360
      MTOL = (5*MTOL)/4                                                 LIV05370
      IF(MTOL.GT.MDIMTV.AND.NTVCON.EQ.0) GO TO 1530                     LIV05380
C                                                                       LIV05390
C-----------------------------------------------------------------------LIV05400
C     GENERATE A RANDOM VECTOR TO BE USED REPEATEDLY BY                 LIV05410
```

```
C     SUBROUTINE INVERM                                                 LIV05420
C                                                                       LIV05430
      IIL = RHSEED                                                      LIV05440
      CALL GENRAN(IIL,G,KMAX)                                           LIV05450
C                                                                       LIV05460
C-----------------------------------------------------------------------LIV05470
C                                                                       LIV05480
C     FOR EACH EIGENVALUE LOOP ON T-EIGENVECTOR COMPUTATIONS TO         LIV05490
C     COMPUTE AN APPROPRIATE T-EIGENVECTOR TO USE IN THE RITZ           LIV05500
C     VECTOR COMPUTATIONS.                                              LIV05510
C                                                                       LIV05520
      MTOL = 0                                                          LIV05530
      NTVEC = 0                                                         LIV05540
      ILBIS = 0                                                         LIV05550
      DO 750 J = 1,NGOOD                                                LIV05560
      ICOUNT = 0                                                        LIV05570
      ERRMIN = 10.D0                                                    LIV05580
      MABEST = MPMIN                                                    LIV05590
      IF(MP(J).EQ.MPMIN) GO TO 750                                      LIV05600
      TFLAG = 0                                                         LIV05610
      EVAL = GOODBI(J)                                                  LIV05620
      TEMP = RELTOL*DABS(EVAL)                                          LIV05630
      UB = EVAL + DMAX1(STUTOL,TEMP)                                    LIV05640
      LB = EVAL - DMAX1(STUTOL,TEMP)                                    LIV05650
  570 KMAXU = IABS(MA(J))                                               LIV05660
C                                                                       LIV05670
C     SELECT A SUITABLE INCREMENT FOR THE ORDERS OF THE T-MATRICES      LIV05680
C     TO BE CONSIDERED IN DETERMINING APPROPRIATE SIZES FOR THE RITZ    LIV05690
C     VECTOR COMPUTATIONS.                                              LIV05700
      IF(ICOUNT.GT.0) GO TO 590                                         LIV05710
C     SELECT IDELTA(J) BASED UPON THE T-MULTIPLICITY OBTAINED           LIV05720
      IF(M2(J).EQ.KMAX) GO TO 580                                       LIV05730
C     M2 DETERMINED                                                     LIV05740
      IDELTA(J) = ((3*M1(J) + 5*M2(J))/8  +  1 - IABS(MA(J)))/10 +  1   LIV05750
      GO TO 590                                                         LIV05760
C     M2 NOT DETERMINED                                                 LIV05770
  580 MAMAX = MIN0((11*MEV)/8 + 12, (13*M1(J))/8 + 1)                   LIV05780
      IDELTA(J) = (MAMAX - IABS(MA(J)))/10 +  1                         LIV05790
  590 ICOUNT = ICOUNT + 1                                               LIV05800
C                                                                       LIV05810
C-----------------------------------------------------------------------LIV05820
C     TO MIMIMIZE THE EFFECT OF THE ONE-SIDED ACCEPTANCE TEST FOR       LIV05830
C     EIGENVALUES IN THE BISEC SUBROUTINE, RECOMPUTE THE GIVEN          LIV05840
C     EIGENVALUE AT THE SPECIFIED KMAXU                                 LIV05850
C                                                                       LIV05860
      CALL LBISEC(ALPHA,BETA,EPSM,EVAL,EVALN,LB,UB,TTOL,KMAXU,NEVT)     LIV05870
C                                                                       LIV05880
C-----------------------------------------------------------------------LIV05890
C                                                                       LIV05900
C     CHECK WHETHER OR NOT GIVEN T-MATRIX HAS AN EIGENVALUE IN THE      LIV05910
C     SPECIFIED INTERVAL AND IF SO WHAT ITS T-MULTIPLICITY IS.          LIV05920
C                                                                       LIV05930
      IF(NEVT.EQ.1) GO TO 630                                           LIV05940
      IF(NEVT.NE.0) GO TO 610                                           LIV05950
      ILBIS = 1                                                         LIV05960
```

```
      WRITE(6,600) EVAL,KMAXU                                          LIV05970
  600 FORMAT(/' PROBLEM ENCOUNTERED IN RECOMPUTATION OF USER-SUPPLIED EILIV05980
     1GENVALUE',E20.12/' THE SIZE T-MATRIX SPECIFIED',I6,' DOES NOT     LIV05990
     1HAVE AN EIGENVALUE IN THE INTERVAL SPECIFIED'/' THEREFORE NO EIGENLIV06000
     1VECTOR WILL BE COMPUTED FOR THIS PARTICULAR EIGENVALUE')         LIV06010
      GO TO 650                                                        LIV06020
C                                                                      LIV06030
  610 IF(NEVT.GT.1)  WRITE(6,620) EVAL,KMAXU                           LIV06040
  620 FORMAT(/' PROBLEM ENCOUNTERED IN RECOMPUTATION OF USER-SUPPLIED   LIV06050
     1EIGENVALUE',E20.12/' FOR THE SIZE T-MATRIX SPECIFIED =',I6,' THE  LIV06060
     1GIVEN EIGENVALUE IS MULTIPLE IN THE INTERVAL SPECIFIED'/' SOMETHINLIV06070
     1G IS WRONG, THEREFORE NO EIGENVECTOR WILL BE COMPUTED FOR THIS EIGLIV06080
     1NVALUE')                                                         LIV06090
C                                                                      LIV06100
      MP(J) = MPMIN                                                    LIV06110
      MA(J) = -2*KMAX                                                  LIV06120
      GO TO 750                                                        LIV06130
C                                                                      LIV06140
  630 CONTINUE                                                         LIV06150
      ILBIS = 0                                                        LIV06160
C                                                                      LIV06170
      EVNEW(J) = EVALN                                                 LIV06180
      EVAL = EVALN                                                     LIV06190
      MTOL = MTOL+KMAXU                                                LIV06200
C                                                                      LIV06210
C     IS THERE ROOM IN TVEC ARRAY FOR THE NEXT T-EIGENVECTOR?          LIV06220
C     IF NOT, SKIP TO RITZ VECTOR COMPUTATIONS.                        LIV06230
      IF (MTOL.GT.MDIMTV) GO TO 760                                    LIV06240
C                                                                      LIV06250
      IT = 3                                                           LIV06260
      KINT = MTOL - KMAXU +1                                           LIV06270
C                                                                      LIV06280
C     RECORD THE BEGINNING AND END OF THE T-EIGENVECTOR BEING COMPUTED  LIV06290
      MINT(J) = KINT                                                   LIV06300
      MFIN(J) = MTOL                                                   LIV06310
C                                                                      LIV06320
C----------------------------------------------------------------------LIV06330
C     SUBROUTINE INVERM DOES INVERSE ITERATION, I.E. SOLVES            LIV06340
C     (T(1,KMAXU) - EVAL)*U = RHS  FOR EACH EIGENVALUE TO OBTAIN       LIV06350
C     THE DESIRED T-EIGENVECTOR.                                       LIV06360
C                                                                      LIV06370
      IF(IWRITE.EQ.1)   WRITE(6,640) J                                 LIV06380
  640 FORMAT(/I6,'TH EIGENVALUE')                                      LIV06390
C                                                                      LIV06400
      CALL INVERM(ALPHA,BETA,V1,TVEC(KINT),EVAL,ERROR,TERROR,EPSM,     LIV06410
     1 G,KMAXU,IT,IWRITE)                                              LIV06420
C                                                                      LIV06430
C----------------------------------------------------------------------LIV06440
C                                                                      LIV06450
      TERR(J) = TERROR                                                 LIV06460
      TLAST(J) = ERROR                                                 LIV06470
      KMAXU1 = KMAXU + 1                                               LIV06480
      TBETA(J) = BETA(KMAXU1)*ERROR                                    LIV06490
C                                                                      LIV06500
C     AFTER COMPUTING EACH OF THE T-EIGENVECTORS,                      LIV06510
```

```
C     CHECK THE SIZE OF THE ERROR ESTIMATE, ERROR.                      LIV06520
C     IF THIS ESTIMATE IS NOT AS SMALL AS DESIRED AND                   LIV06530
C     |MA(J)| < ML(J), ATTEMPT TO INCREASE THE SIZE OF |MA(J)|          LIV06540
C     AND REPEAT THE T-EIGENVECTOR COMPUTATIONS.                        LIV06550
C                                                                       LIV06560
      IF(ERROR.LT.ERTOL.OR.TFLAG.EQ.1)  GO TO 740                       LIV06570
C                                                                       LIV06580
      IF(ERROR.GE.ERRMIN) GO TO 650                                     LIV06590
C     LAST COMPONENT IS LESS THAN MINIMAL TO DATE                       LIV06600
      ERRMIN = ERROR                                                    LIV06610
      MABEST = MA(J)                                                    LIV06620
  650 CONTINUE                                                          LIV06630
C                                                                       LIV06640
      IF(MA(J).GT.0)  ITEST = MA(J) + IDELTA(J)                         LIV06650
      IF(MA(J).LT.0)  ITEST = -(IABS(MA(J)) + IDELTA(J))                LIV06660
      IF(IABS(ITEST).LE.ML(J).AND.ICOUNT.LE.10) GO TO 670               LIV06670
C     NEW MA(J) IS GREATER THAN MAXIMUM ALLOWED.                        LIV06680
      IF(ERCONT.EQ.0.OR.MABEST.EQ.MPMIN) GO TO 690                      LIV06690
      TFLAG = 1                                                         LIV06700
      MA(J) = MABEST                                                    LIV06710
      IF(ILBIS.EQ.0) MTOL = MTOL - KMAXU                                LIV06720
      WRITE(6,660) MA(J)                                                LIV06730
  660 FORMAT(' 10 ORDERS WERE CONSIDERED.  NONE SATISFIED THE ERROR TESTLIV06740
     1'/' THEREFORE USE THE BEST ORDER OBTAINED FOR THE EIGENVECTORS'   LIV06750
     1,I6)                                                              LIV06760
      GO TO 570                                                         LIV06770
C                                                                       LIV06780
  670 MA(J) = ITEST                                                     LIV06790
C                                                                       LIV06800
      MT = IABS(MA(J))                                                  LIV06810
      IF(IWRITE.EQ.1) WRITE(6,680)  MT                                  LIV06820
  680 FORMAT(/' CHANGE SIZE OF T-MATRIX TO ',I6,' RECOMPUTE T-EIGENVECTOLIV06830
     1R')                                                               LIV06840
C                                                                       LIV06850
      IF(ILBIS.EQ.0)  MTOL = MTOL - KMAXU                               LIV06860
C                                                                       LIV06870
      GO TO 570                                                         LIV06880
C                                                                       LIV06890
C     APPROPRIATE SIZE T-MATRIX WAS NOT OBTAINED                        LIV06900
  690 CONTINUE                                                          LIV06910
      WRITE(10,700) J,EVAL,MP(J)                                        LIV06920
  700 FORMAT(/' ON 10 INCREMENTS NOT ABLE TO IDENTIFY APPROPRIATE SIZE  LIV06930
     1T-MATRIX FOR'/                                                    LIV06940
     1' EIGENVALUE(',I4,') = ',E20.12,' T-MULTIPLICITY =',I4/)          LIV06950
      IF(M2(J).EQ.KMAX) WRITE(10,710)                                   LIV06960
      IF(M2(J).LT.KMAX) WRITE(10,720)                                   LIV06970
  710 FORMAT(' ORDERS TESTED RANGED FROM 5*M1(J)/4 TO APPROXIMATELY     LIV06980
     1 MIN(11*MEV/8,13*M1(J)/8)')                                       LIV06990
  720 FORMAT(' ORDERS TESTED RANGED FROM APPROX. (3*M1(J)+M2(J))/4 TO (3LIV07000
     1*M1(J)+5*M2(J))/8')                                               LIV07010
      WRITE(10,730)                                                     LIV07020
  730 FORMAT(' ALLOWING LARGER ORDERS FOR THE T-MATRICES MAY RESULT IN  LIV07030
     1 SUCCESS'/' BUT PROBABLY WILL NOT.  PROBLEM IS PROBABLY DUE TO'   LIV07040
     1 /' LACK OF CONVERGENCE OF GIVEN EIGENVALUE, CHECK THE ERROR ESTIMLIV07050
     1ATE')                                                             LIV07060
```

```
      MP(J) = MPMIN                                                     LIV07070
      IF(ILBIS.EQ.0)  MTOL = MTOL - KMAXU                               LIV07080
      GO TO 750                                                         LIV07090
  740 NTVEC = NTVEC + 1                                                 LIV07100
C                                                                       LIV07110
  750 CONTINUE                                                          LIV07120
      NGOODC = NGOOD                                                    LIV07130
      GO TO 780                                                         LIV07140
C                                                                       LIV07150
C     COME HERE IF THERE IS NOT ENOUGH ROOM FOR ALL OF T-EIGENVECTORS   LIV07160
  760 NGOODC = J-1                                                      LIV07170
      WRITE(6,770)  J,MTOL,MDIMTV                                       LIV07180
  770 FORMAT(/' NOT ENOUGH ROOM IN TVEC ARRAY FOR ',I4,'TH T-EIGENVECTORLIV07190
     1'/'  TVEC DIMENSION REQUESTED = ',I6,' BUT TVEC HAS DIMENSION ',I6LIV07200
     1/)                                                                LIV07210
      IF(NGOODC.EQ.0)  GO TO 1550                                       LIV07220
      MTOL = MTOL-KMAXU                                                 LIV07230
C                                                                       LIV07240
  780 CONTINUE                                                          LIV07250
C                                                                       LIV07260
C     THE LOOP ON T-EIGENVECTOR COMPUTATIONS IS COMPLETE.               LIV07270
C     WRITE OUT THE SIZE T-MATRICES THAT WILL BE USED FOR               LIV07280
C     THE RITZ VECTOR COMPUTATIONS.                                     LIV07290
C                                                                       LIV07300
      WRITE(10,790)                                                     LIV07310
  790 FORMAT(/' SIZES OF T-MATRICES THAT WILL BE USED IN THE RITZ COMPUTLIV07320
     1ATIONS'/5X,'J',8X,'  GOODBI(J) ',13X,'  GOODA(J) ',7X,'MA(J)')    LIV07330
C                                                                       LIV07340
      WRITE(10,800)   (J,GOODBI(J),GOODA(J),MA(J), J=1,NGOOD)           LIV07350
  800 FORMAT(I6,2E25.14,I6)                                             LIV07360
      WRITE(10,550)                                                     LIV07370
C                                                                       LIV07380
      WRITE(6,810) MTOL                                                 LIV07390
  810 FORMAT(/' THE CUMULATIVE LENGTH OF THE T-EIGENVECTORS IS',I18)    LIV07400
C                                                                       LIV07410
      WRITE(6,820) NTVEC,NGOOD                                          LIV07420
  820 FORMAT(/I6,' T-EIGENVECTORS OUT OF',I6,' REQUESTED WERE COMPUTED')LIV07430
C                                                                       LIV07440
C     SAVE THE T-EIGENVECTORS ON FILE 11?                               LIV07450
      IF(TVSTOP.NE.1.AND.SVTVEC.EQ.0) GO TO 880                         LIV07460
C                                                                       LIV07470
      WRITE(11,830) NTVEC,MTOL,MATNO,SVSEED                             LIV07480
  830 FORMAT(I6,3I12,' = NTVEC,MTOL,MATNO,SVSEED')                      LIV07490
C                                                                       LIV07500
      DO 860 J=1,NGOODC                                                 LIV07510
C     IF MP(J) = MPMIN THEN NO SUITABLE T-EIGENVECTOR IS AVAILABLE      LIV07520
C     FOR THAT EIGENVALUE.                                              LIV07530
      IF(MP(J).EQ.MPMIN) WRITE(11,840) J,MA(J),GOODBI(J),MP(J)          LIV07540
  840 FORMAT(2I6,E20.12,I6/' TH EIGVAL,T-SIZE,EVALUE,FLAG,NO EIGVEC')   LIV07550
      IF(MP(J).NE.MPMIN) WRITE(11,850) J,MA(J),GOODBI(J),MP(J)          LIV07560
  850 FORMAT(I6,I6,E20.12,I6/'  T-EIGENVECTOR, T-SIZE , BI-EIGENVALUE, TLIV07570
     1-MULTIPLICITY')                                                   LIV07580
      IF(MP(J).EQ.MPMIN) GO TO 860                                      LIV07590
      KI = MINT(J)                                                      LIV07600
      KF = MFIN(J)                                                      LIV07610
```

```
C                                                                       LIV07620
      WRITE(11,300) (TVEC(K), K=KI,KF)                                  LIV07630
C                                                                       LIV07640
  860 CONTINUE                                                          LIV07650
C                                                                       LIV07660
      IF(TVSTOP.NE.1) GO TO 880                                         LIV07670
C                                                                       LIV07680
      WRITE(6,870) TVSTOP, NTVEC,NGOOD                                  LIV07690
  870 FORMAT(/' USER SET TVSTOP = ',I1/                                 LIV07700
     1' THEREFORE PROGRAM TERMINATES AFTER T-EIGENVECTOR COMPUTATIONS'/ LIV07710
     1' T-EIGENVECTORS THAT WERE COMPUTED ARE SAVED ON FILE 11'/        LIV07720
     1I8,' T-EIGENVECTORS WERE COMPUTED OUT OF',I7,' REQUESTED'/)       LIV07730
C                                                                       LIV07740
      GO TO 1670                                                        LIV07750
C                                                                       LIV07760
  880 CONTINUE                                                          LIV07770
C     IF NOT ABLE TO COMPUTE ALL THE REQUESTED T-EIGENVECTORS           LIV07780
C     CONTINUE WITH THE LANCZOS VECTOR COMPUTATIONS ANYWAY?             LIV07790
      IF(NTVEC.NE.NGOOD.AND.LVCONT.EQ.0) GO TO 1570                     LIV07800
C                                                                       LIV07810
C     COMPUTE THE MAXIMUM SIZE OF THE T-MATRIX USED FOR THOSE           LIV07820
C     EIGENVALUES WITH GOOD ERROR ESTIMATES.                            LIV07830
C                                                                       LIV07840
      KMAXU = 0                                                         LIV07850
      DO 890 J = 1,NGOODC                                               LIV07860
      MT = IABS(MA(J))                                                  LIV07870
      IF(MT.LT.KMAXU.OR.MP(J).EQ.MPMIN) GO TO 890                       LIV07880
      KMAXU = MT                                                        LIV07890
  890 CONTINUE                                                          LIV07900
C                                                                       LIV07910
      IF(KMAXU.EQ.0) GO TO 1610                                         LIV07920
C                                                                       LIV07930
      WRITE(6,900) KMAXU                                                LIV07940
  900 FORMAT(/I6,' = LARGEST SIZE T-MATRIX TO BE USED IN THE RITZ VECTORLIV07950
     1 COMPUTATIONS')                                                   LIV07960
C                                                                       LIV07970
C     COUNT THE NUMBER OF RITZ VECTORS NOT BEING COMPUTED               LIV07980
      MREJEC = 0                                                        LIV07990
      DO 910 J=1,NGOODC                                                 LIV08000
  910 IF(MP(J).EQ.MPMIN)  MREJEC = MREJEC + 1                           LIV08010
      MREJET = MREJEC + (NGOOD-NGOODC)                                  LIV08020
      IF(MREJET.NE.0) WRITE(6,920) MREJET                               LIV08030
  920 FORMAT(/' RITZ VECTORS ARE NOT COMPUTED FOR',I6,' OF THE EIGNEVALULIV08040
     1ES'/)                                                             LIV08050
      NACT = NGOODC - MREJEC                                            LIV08060
      WRITE(6,930) NGOOD,NTVEC,NACT                                     LIV08070
  930 FORMAT(/I6,' RITZ VECTORS WERE REQUESTED'/I6,' T-EIGENVECTORS WERELIV08080
     1 COMPUTED'/I6,' RITZ VECTORS WILL BE COMPUTED'/)                  LIV08090
C     CHECK IF THERE ARE ANY RITZ VECTORS TO COMPUTE                    LIV08100
      IF(MREJEC.EQ.NGOODC) GO TO 1590                                   LIV08110
C                                                                       LIV08120
C     CONTINUE WITH THE LANCZOS VECTOR COMPUTATIONS?                    LIV08130
      IF(LVCONT.EQ.0.AND.MREJEC.NE.0) GO TO 1570                        LIV08140
C                                                                       LIV08150
C     NOW COMPUTE THE RITZ VECTORS.  REGENERATE THE                     LIV08160
```

```
C     LANCZOS VECTORS.                                                  LIV08170
C                                                                       LIV08180
      DO 940 I = 1,NMAX                                                 LIV08190
  940 RITVEC(I) = ZERO                                                  LIV08200
C                                                                       LIV08210
C-----------------------------------------------------------------------LIV08220
C     REGENERATE THE STARTING VECTOR. THIS MUST BE GENERATED AND        LIV08230
C     NORMALIZED PRECISELY THE WAY IT WAS DONE IN THE EIGENVALUE        LIV08240
C     COMPUTATIONS, OTHERWISE THERE WILL BE A MISMATCH BETWEEN          LIV08250
C     THE T-EIGENVECTORS THAT HAVE BEEN COMPUTED FROM THE T-MATRICES    LIV08260
C     READ IN FROM FILE 2 AND THE LANCZOS VECTORS THAT ARE              LIV08270
C     BEING REGENERATED.                                                LIV08280
C                                                                       LIV08290
      CALL GENRAN(SVSEED,G,N)                                           LIV08300
C                                                                       LIV08310
C-----------------------------------------------------------------------LIV08320
C                                                                       LIV08330
      DO 950 J = 1,N                                                    LIV08340
  950 V2(J) = G(J)                                                      LIV08350
C                                                                       LIV08360
C-----------------------------------------------------------------------LIV08370
      SUM = FINPRO(N,V2(1),1,V2(1),1)                                   LIV08380
C-----------------------------------------------------------------------LIV08390
C                                                                       LIV08400
      SUM = ONE/DSQRT(SUM)                                              LIV08410
C                                                                       LIV08420
      DO 960 I = 1,N                                                    LIV08430
      V1(I) = ZERO                                                      LIV08440
  960 V2(I) = V2(I)*SUM                                                 LIV08450
C                                                                       LIV08460
      WRITE(6,970)                                                      LIV08470
  970 FORMAT(' STARTING LANCZOS VECTOR HAS BEEN CALCULATED'/)           LIV08480
C                                                                       LIV08490
C     LOOP FOR GENERATING RITZ VECTORS  (IVEC = 1,KMAXU)                LIV08500
      IVEC = 1                                                          LIV08510
      BATA = ZERO                                                       LIV08520
C                                                                       LIV08530
      GO TO 1050                                                        LIV08540
C                                                                       LIV08550
  980 CONTINUE                                                          LIV08560
C                                                                       LIV08570
C     SOLVE B*VS = V2 FOR VS                                            LIV08580
      DO 990 K = 1,N                                                    LIV08590
  990 VS(K) = V2(K)                                                     LIV08600
C                                                                       LIV08610
C-----------------------------------------------------------------------LIV08620
      JBSOLV = 2                                                        LIV08630
      CALL BSOLV(VS,VS,JBSOLV)                                          LIV08640
C-----------------------------------------------------------------------LIV08650
C                                                                       LIV08660
C     VS = BI*V2   BI = B(INVERSE)                                      LIV08670
C     COMPUTE V1 = BI*V2 - BATA*V1                                      LIV08680
      DO 1000 K = 1,N                                                   LIV08690
 1000 V1(K) = VS(K) - BATA*V1(K)                                        LIV08700
C                                                                       LIV08710
```

```
C--------------------------------------------------------------------------LIV08720
      ALFA = FINPRO(N,V1(1),1,V2(1),1)                                     LIV08730
C--------------------------------------------------------------------------LIV08740
C                                                                          LIV08750
      DO 1010 J = 1,N                                                      LIV08760
 1010 V1(J) = V1(J)-ALFA*V2(J)                                             LIV08770
C                                                                          LIV08780
C--------------------------------------------------------------------------LIV08790
      BATA = FINPRO(N,V1(1),1,V1(1),1)                                     LIV08800
C--------------------------------------------------------------------------LIV08810
C                                                                          LIV08820
      BATA = DSQRT(BATA)                                                   LIV08830
      SUM = ONE/BATA                                                       LIV08840
C                                                                          LIV08850
      TEMP = BETA(IVEC)                                                    LIV08860
      TEMP = DABS(BATA - TEMP)/TEMP                                        LIV08870
      IF (TEMP.LT.1.0D-10)GO TO 1030                                       LIV08880
C                                                                          LIV08890
C     THE BETA BEING REGENERATED DO NOT MATCH THE HISTORY FILE             LIV08900
C     SOMETHING IS WRONG IN THE LANCZOS VECTOR GENERATION                  LIV08910
C     PROGRAM TERMINATES FOR USER TO CORRECT THE PROBLEM                   LIV08920
C     WHICH MUST BE IN THE STARTING VECTOR GENERATION OR IN                LIV08930
C     THE MATRIX-VECTOR MULTIPLY SUBROUTINE CMATV SUPPLIED.                LIV08940
C     THIS SUBROUTINE MUST BE THE SAME ONE USED IN THE                     LIV08950
C     EIGENVALUE COMPUTATIONS OR AGAIN A MISMATCH WILL ENSUE.              LIV08960
C                                                                          LIV08970
      WRITE(6,1020) IVEC,BATA,BETA(IVEC),TEMP                              LIV08980
 1020 FORMAT(/2X,'IVEC',16X,'BATA',10X,'BETA(IVEC)',14X,'RELDIF'/I6,       LIV08990
     13E20.12/' IN LANCZOS VECTOR REGENERATION THE ENTRIES OF THE TRIDIALIV09000
     1GONAL MATRICES BEING'/' GENERATED ARE NOT THE SAME AS THOSE IN THELIV09010
     1 MATRIX SUPPLIED ON FILE 2.'/' THEREFORE SOMETHING IS BEING INITIALIV09020
     1LIZED OR COMPUTED DIFFERENTLY FROM THE WAY'/' IT WAS COMPUTED IN TLIV09030
     1HE EIGENVALUE COMPUTATIONS'/' THE PROGRAM TERMINATES FOR THE USER LIV09040
     1TO DETERMINE WHAT THE PROBLEM IS'/)                                 LIV09050
      GO TO 1670                                                           LIV09060
C                                                                          LIV09070
 1030 CONTINUE                                                             LIV09080
      DO 1040 J = 1,N                                                      LIV09090
      TEMP = SUM*V1(J)                                                     LIV09100
      V1(J) = V2(J)                                                        LIV09110
 1040 V2(J) = TEMP                                                         LIV09120
C                                                                          LIV09130
 1050 CONTINUE                                                             LIV09140
C                                                                          LIV09150
      LFIN = 0                                                             LIV09160
      DO 1070 J = 1,NGOODC                                                 LIV09170
      LL = LFIN                                                            LIV09180
      LFIN = LFIN + N                                                      LIV09190
C                                                                          LIV09200
      IF(IABS(MA(J)).LT.IVEC.OR.MP(J).EQ.MPMIN) GO TO 1070                 LIV09210
      II = IVEC + MINT(J) - 1                                              LIV09220
      TEMP = TVEC(II)                                                      LIV09230
C     II IS THE (IVEC)TH COMPONENT OF THE T-EIGENVECTOR CONTAINED          LIV09240
C     IN TVEC(MINT(J)).                                                    LIV09250
C                                                                          LIV09260
```

```
      DO 1060 K = 1,N                                                   LIV09270
      LL = LL + 1                                                       LIV09280
 1060 RITVEC(LL) = TEMP*V2(K) + RITVEC(LL)                              LIV09290
C                                                                       LIV09300
 1070 CONTINUE                                                          LIV09310
C                                                                       LIV09320
      IVEC = IVEC + 1                                                   LIV09330
      IF (IVEC.LE.KMAXU) GO TO 980                                      LIV09340
C                                                                       LIV09350
C     RITZVECTOR GENERATION IS COMPLETE. NORMALIZE EACH RITZVECTOR.     LIV09360
C     NOTE THAT IF CERTAIN RITZ VECTORS WERE NOT COMPUTED THEN THAT     LIV09370
C     PORTION OF THE RITVEC ARRAY WAS NOT UTILIZED.                     LIV09380
C                                                                       LIV09390
      LFIN = 0                                                          LIV09400
      DO 1140 J = 1,NGOODC                                              LIV09410
C                                                                       LIV09420
      KK = LFIN                                                         LIV09430
      LFIN = LFIN + N                                                   LIV09440
      IF(MP(J).EQ.MPMIN) GO TO 1140                                     LIV09450
C                                                                       LIV09460
      DO 1080 K = 1,N                                                   LIV09470
      KK = KK + 1                                                       LIV09480
      V1(K) = RITVEC(KK)                                                LIV09490
 1080 VS(K) = V1(K)                                                     LIV09500
C                                                                       LIV09510
      IF(JPERM.EQ.0)  GO TO 1090                                        LIV09520
C                                                                       LIV09530
C-----------------------------------------------------------------------LIV09540
C     V2 = V1 = (L-TRANSPOSE)*V1                                        LIV09550
      IPERM = 2                                                         LIV09560
      CALL LPERM(V1,V2,IPERM)                                           LIV09570
C-----------------------------------------------------------------------LIV09580
C                                                                       LIV09590
C     V2 CONTAINS RITZ VECTOR FOR A, VS CONTAINS THE RITZ VECTOR FOR B  LIV09600
C                                                                       LIV09610
 1090 CONTINUE                                                          LIV09620
C                                                                       LIV09630
C-----------------------------------------------------------------------LIV09640
      SUM = FINPRO(N,V1(1),1,V1(1),1)                                   LIV09650
C-----------------------------------------------------------------------LIV09660
C                                                                       LIV09670
      SUM = DSQRT(SUM)                                                  LIV09680
      RNORM(J) = SUM                                                    LIV09690
      RNORME = DABS(ONE-SUM)                                            LIV09700
      SUM = ONE/SUM                                                     LIV09710
C                                                                       LIV09720
      KK = LFIN - N                                                     LIV09730
      DO 1100 K = 1,N                                                   LIV09740
      KK = KK + 1                                                       LIV09750
      VS(K) = SUM*VS(K)                                                 LIV09760
 1100 RITVEC(KK) = SUM*V1(K)                                            LIV09770
C                                                                       LIV09780
C     VS IS RITZ VECTOR FOR BI:  RITVEC IS RITZ VECTOR FOR A-MATRIX     LIV09790
C     B = S0*P*A*P' + SHIFT*I                                           LIV09800
C     BIERR = ||BI*VS - GOODBI(J)*VS||                                  LIV09810
```

```
C     BIEVER = |(VS-TRANS)*BI*VS - GOODBI(J)|                          LIV09820
C                                                                      LIV09830
C----------------------------------------------------------------------LIV09840
C     V1 = (B-INVERSE)*VS                                              LIV09850
      JBSOLV = 2                                                       LIV09860
      CALL BSOLV(VS,V1,JBSOLV)                                         LIV09870
C----------------------------------------------------------------------LIV09880
C                                                                      LIV09890
      EVALN = EVNEW(J)                                                 LIV09900
C                                                                      LIV09910
C----------------------------------------------------------------------LIV09920
      TEMP = FINPRO(N,V1(1),1,VS(1),1)                                 LIV09930
C----------------------------------------------------------------------LIV09940
C                                                                      LIV09950
      TEMP = DABS(TEMP - EVALN)                                        LIV09960
      BIEVER(J) = TEMP                                                 LIV09970
      DO 1110 K = 1,N                                                  LIV09980
 1110 V1(K) = V1(K) - EVALN*VS(K)                                      LIV09990
C                                                                      LIV10000
C----------------------------------------------------------------------LIV10010
      SUM = FINPRO(N,V1(1),1,V1(1),1)                                  LIV10020
C----------------------------------------------------------------------LIV10030
C                                                                      LIV10040
      SUM = DSQRT(SUM)                                                 LIV10050
      BIERR(J) = SUM                                                   LIV10060
      BIERRG(J) = SUM/ABS(BIMING(J))                                   LIV10070
C                                                                      LIV10080
      LINT = LFIN - N + 1                                              LIV10090
      EVAL = (ONE/EVALN - SHIFT)/SO                                    LIV10100
      GOODA(J) = EVAL                                                  LIV10110
      TEMP = BIEVER(J)                                                 LIV10120
C                                                                      LIV10130
      IF(IWRITE.EQ.0)  GO TO 1140                                      LIV10140
      WRITE(6,1120) J,GOODBI(J)                                        LIV10150
 1120 FORMAT(/I5,' TH B-INVERSE EIGENVALUE COMPUTED = ',E20.12/)       LIV10160
C                                                                      LIV10170
      WRITE(6,1130)  TERR(J),TBETA(J),RNORME                           LIV10180
 1130 FORMAT(' NORM OF ERROR IN T-EIGENVECTOR = ',E14.3/               LIV10190
     1' BETA(MA(J)+1)*U(MA(J)) = ',E14.3/                             LIV10200
     1' ABS(NORM(RITVEC) - 1.0) = ',E14.3/)                           LIV10210
C                                                                      LIV10220
 1140 CONTINUE                                                         LIV10230
C                                                                      LIV10240
C     RITZVECTORS ARE NORMALIZED AND ERROR ESTIMATES ARE IN BIERR      LIV10250
C     AND BIERRG ARRAYS. STORE EVERYTHING                              LIV10260
C                                                                      LIV10270
      WRITE(13,1150)                                                   LIV10280
 1150 FORMAT(6X,'BIEIGENVALUE',6X,'RITZNORM',7X,'TBETA',7X,'TLAST',5X, LIV10290
     1 'BIERROR',6X,'BIEVER')                                         LIV10300
C                                                                      LIV10310
      WRITE(9,1160)                                                    LIV10320
 1160 FORMAT(5X,'BIEIGENVALUE',4X,'MA(J)',4X,'BIMINGAP',5X,'BIERROR',3X LIV10330
     1 ,'BIERR/GAP',6X,'TERROR')                                      LIV10340
C                                                                      LIV10350
      DO 1190 J=1,NGOODC                                               LIV10360
```

```
C                                                                       LIV10370
      IF(MP(J).EQ.MPMIN) GO TO 1190                                     LIV10380
C                                                                       LIV10390
      WRITE(9,1170) GOODBI(J),MA(J),BIMING(J),BIERR(J),BIERRG(J),TERR(J)LIV10400
 1170 FORMAT(E20.12,I6,4E12.4)                                          LIV10410
C                                                                       LIV10420
      WRITE(13,1180) EVNEW(J),RNORM(J),TBETA(J),TLAST(J),BIERR(J),      LIV10430
     1 BIEVER(J)                                                        LIV10440
 1180 FORMAT(E20.12,5E12.4)                                             LIV10450
C                                                                       LIV10460
 1190 CONTINUE                                                          LIV10470
C                                                                       LIV10480
      WRITE(9,1200)                                                     LIV10490
 1200 FORMAT(/5X, 'J',7X,'AEIGENVALUE',3X,'MA(J)',5X,'AMINGAP')         LIV10500
C                                                                       LIV10510
      DO 1210 J = 1,NGOOD                                               LIV10520
      IF(MP(J).EQ.MPMIN) GO TO 1210                                     LIV10530
      WRITE(9,1220) J,GOODA(J),MA(J),AMINGP(J)                          LIV10540
 1210 CONTINUE                                                          LIV10550
 1220 FORMAT(I6,E20.12,I6,E12.4)                                        LIV10560
C                                                                       LIV10570
      IF (MREJEC.EQ.0) GO TO 1300                                       LIV10580
C                                                                       LIV10590
      WRITE(9,1230)                                                     LIV10600
 1230 FORMAT(/' RITZ VECTORS WERE NOT COMPUTED FOR THE FOLLOWING EIGENVALIV10610
     1LUES'/' EITHER BECAUSE THEY HAD NOT CONVERGED OR BECAUSE THE ERRORLIV10620
     1 ESTIMATE'/'  WAS NOT AS SMALL AS DESIRED'/)                      LIV10630
C                                                                       LIV10640
      WRITE(9,1240)                                                     LIV10650
 1240 FORMAT(6X,'GOODBI(J)',3X,'MA(J)',5X,'BIMING(J)',6X,'TBETA(J)',3X, LIV10660
     1'MP(J)')                                                          LIV10670
C                                                                       LIV10680
      WRITE(13,1250)                                                    LIV10690
 1250 FORMAT(/' RITZ VECTORS WERE NOT COMPUTED FOR THE FOLLOWING EIGENVALIV10700
     1LUES'/' EITHER BECAUSE THEY HAD NOT CONVERGED OR BECAUSE'/' THE ERLIV10710
     1ROR ESTIMATE WAS NOT AS SMALL AS DESIRED'/)                       LIV10720
C                                                                       LIV10730
      WRITE(13,1260)                                                    LIV10740
 1260 FORMAT(3X,'BIEIGENVALUE',3X,'MA(J)',3X,'M1(J)',3X,'M2(J)',3X,'MP(JLIV10750
     1)')                                                               LIV10760
C                                                                       LIV10770
      DO 1290 J = 1,NGOODC                                              LIV10780
C                                                                       LIV10790
      IF(MP(J).NE.MPMIN) GO TO 1290                                     LIV10800
C                                                                       LIV10810
C     WRITE OUT MESSAGE FOR EACH EIGENVALUE FOR WHICH NO EIGENVECTOR    LIV10820
C     WAS COMPUTED.                                                     LIV10830
C                                                                       LIV10840
      WRITE(9,1270) GOODBI(J),MA(J),BIMING(J),TBETA(J),MP(J)            LIV10850
 1270 FORMAT(E15.8,I8,2E14.4,I8)                                        LIV10860
C                                                                       LIV10870
      WRITE(13,1280) GOODBI(J),MA(J),M1(J),M2(J),MP(J)                  LIV10880
 1280 FORMAT(E15.8,4I8)                                                 LIV10890
C                                                                       LIV10900
 1290 CONTINUE                                                          LIV10910
```

```
C                                                                       LIV10920
 1300 CONTINUE                                                          LIV10930
C                                                                       LIV10940
      WRITE(9,1310)                                                     LIV10950
 1310 FORMAT(/' ABOVE ARE ERROR ESTIMATES FOR THE BI AND T EIGENVECTORS'LIV10960
     1/ ' ASSOCIATED WITH THE GOODBI LISTED, DENOTED BY EV '/           LIV10970
     1 ' BIERROR = NORM(BI*X-EV*X),  TERROR = NORM(T*Y - EV*Y)'/        LIV10980
     1 ' WHERE T = T(1,MA(J)),  P*X = RITZVEC = V*Y, T*Y = GOODBI*Y'/   LIV10990
     1 ' BIMINGAP = GAP TO NEAREST BI-EIGENVALUE'/)                     LIV11000
C                                                                       LIV11010
      WRITE(13,1320)                                                    LIV11020
 1320 FORMAT(/' ABOVE ARE ERROR ESTIMATES FOR THE EIGENVECTORS'/        LIV11030
     1 ' ASSOCIATED WITH THE BI-EIGENVALUES'/                           LIV11040
     1 ' RITZNORM = NORM(COMPUTED RITZ VECTOR FOR B-INVERSE)'/          LIV11050
     1 ' TBETA(J) = BETA(MA(J)+1)*Y(MA(J)),  T*Y = BIEVAL*Y'/           LIV11060
     1 ' TLAST(J) = DABS(Y(MA(J)))'/                                    LIV11070
     1 ' BIERROR = NORM(BI*X - BIEVAL*X) WHERE X = V*Y'/                LIV11080
     1 ' BIEVER = DABS(BIEIGENVALUE - (X-TRANSPOSE*BINVERSE*X))'/)      LIV11090
C                                                                       LIV11100
C     NUMBER OF RITZ VECTORS COMPUTED                                   LIV11110
      NCOMPU = NGOODC - MREJEC                                          LIV11120
      WRITE(12,1330) N,NCOMPU,NGOODC,MATNO                              LIV11130
 1330 FORMAT(3I6,I8,' = SIZE A, NO.RITZVECS, NO.GOODEVALUES,MATNO')     LIV11140
C                                                                       LIV11150
      LFIN = 0                                                          LIV11160
      DO 1390 J = 1,NGOODC                                              LIV11170
      LINT = LFIN + 1                                                   LIV11180
      LFIN = LFIN + N                                                   LIV11190
C                                                                       LIV11200
      IF(MP(J).EQ.MPMIN) GO TO 1370                                     LIV11210
C     RITZ VECTOR WAS COMPUTED                                          LIV11220
      WRITE(12,1340) J, EVNEW(J), GOODA(J),MP(J)                        LIV11230
 1340 FORMAT(I6,4X,2E20.12,I6,' J,GOODBI,GOODA,MP(J)')                  LIV11240
C                                                                       LIV11250
      WRITE(12,1350) BIERR(J), BIERRG(J), BIMING(J),AMINGP(J)           LIV11260
 1350 FORMAT(4X,' BIRESIDUAL  ',2X,'BIRESIDUAL/GAP',                    LIV11270
     12X,'BIMINGAP',3X,' AMINGAP'/                                      LIV11280
     1 E15.5,E16.5,2E11.3)                                              LIV11290
C                                                                       LIV11300
      WRITE(12,1360) (RITVEC(LL), LL=LINT,LFIN)                         LIV11310
 1360 FORMAT(4E20.12)                                                   LIV11320
      GO TO 1390                                                        LIV11330
C     NO RITZ VECTOR WAS COMPUTED FOR THIS EIGENVALUE                   LIV11340
 1370 CONTINUE                                                          LIV11350
      WRITE(12,1380) J,GOODBI(J),GOODA(J),MP(J)                         LIV11360
 1380 FORMAT(/I5,E20.12,E20.12,I6,' = J,GOODBI,GOODA,MP'/' NO RITZ VECTOLIV11370
     1R WAS COMPUTED FOR THIS EIGENVALUE'/)                             LIV11380
C                                                                       LIV11390
 1390 CONTINUE                                                          LIV11400
C                                                                       LIV11410
C     DID ANY T-MATRICES INCLUDE OFF-DIAGONAL ENTRIES SMALLER THAN      LIV11420
C     DESIRED, AS SPECIFIED BY BTOL?                                    LIV11430
C                                                                       LIV11440
      IF(IB.GT.0) GO TO 1420                                            LIV11450
      WRITE(6,1400) KMAXU                                               LIV11460
```

```
 1400 FORMAT(/' FOR LARGEST T-MATRIX CONSIDERED',I7,' CHECK THE SIZE OF LIV11470
     1BETAS')                                                          LIV11480
C                                                                      LIV11490
C-----------------------------------------------------------------------LIV11500
C                                                                      LIV11510
      CALL TNORM(ALPHA,BETA,BKMIN,TEMP,KMAXU,IBMT)                     LIV11520
C                                                                      LIV11530
C-----------------------------------------------------------------------LIV11540
C                                                                      LIV11550
      IF(IBMT.LT.0) WRITE(6,1410)                                      LIV11560
 1410 FORMAT(/' WARNING THE T-MATRICES FOR ONE OR MORE OF THE EIGENVALUELIV11570
     1S CONSIDERED'/' HAD AN OFF DIAGONAL ENTRY THAT WAS SMALLER THAN THLIV11580
     1E BETA TOLERANCE THAT WAS SPECIFIED')                           LIV11590
 1420 CONTINUE                                                         LIV11600
C                                                                      LIV11610
      GO TO 1670                                                       LIV11620
C                                                                      LIV11630
 1430 WRITE(6,1440) NGOOD,NMAX,MDIMRV                                  LIV11640
 1440 FORMAT(/I4,' RITZ VECTORS WERE REQUESTED BUT THE REQUIRED DIMENSIOLIV11650
     1N',I6/' IS LARGER THAN USER-SPECIFIED DIMENSION OF RITVEC',I6/    LIV11660
     1' THEREFORE, THE EIGENVECTOR PROCEDURE TERMINATES FOR THE USER TO LIV11670
     1 INTERVENE')                                                    LIV11680
C                                                                      LIV11690
      GO TO 1670                                                       LIV11700
C                                                                      LIV11710
 1450 WRITE(6,1460) NOLD,N,MATOLD,MATNO,SHIFTO,SHIFT                   LIV11720
 1460 FORMAT(/' PARAMETERS READ FROM FILE 3 DO NOT AGREE WITH WHAT USER LIV11730
     1SPECIFIED'/ ' NOLD,N,MATOLD,MATNO,SHIFTO,SHIFT = '/2I6,2I8,2E10.3 LIV11740
     1/' THEREFORE PROGRAM TERMINATES FOR USER TO RESOLVE THE DIFFERENCELIV11750
     1S')                                                             LIV11760
C                                                                      LIV11770
      GO TO 1670                                                       LIV11780
C                                                                      LIV11790
 1470 WRITE(6,1480)                                                    LIV11800
 1480 FORMAT(/' PARAMETERS READ FROM ALPHA,BETA FILE DO NOT AGREE WITH WLIV11810
     1HAT USER SPECIFIED'/' PROGRAM TERMINATES FOR USER TO RESOLVE THE DLIV11820
     1IFFERENCES')                                                    LIV11830
C                                                                      LIV11840
      GO TO 1670                                                       LIV11850
C                                                                      LIV11860
 1490 WRITE(6,1500) KMAX,MEV                                           LIV11870
 1500 FORMAT(/' IN ALPHA, BETA FILE KMAX = ',I6/                       LIV11880
     1' BUT EIGENVALUES WERE COMPUTED AT MEV =',I6,' PROGRAM STOPS')   LIV11890
C                                                                      LIV11900
      GO TO 1670                                                       LIV11910
C                                                                      LIV11920
 1510 WRITE(6,1520)                                                    LIV11930
 1520 FORMAT(/' PROGRAM COMPUTED 1ST GUESSES ON T-MATRIX SIZES AND READ LIV11940
     1THEM TO FILE 10'/' THEN TERMINATED AS REQUESTED.')              LIV11950
      GO TO 1670                                                       LIV11960
C                                                                      LIV11970
 1530 WRITE(6,1540) MTOL, MDIMTV                                       LIV11980
 1540 FORMAT(/' PROGRAM TERMINATES BECAUSE THE TVEC DIMENSION ANTICIPATELIV11990
     1D',I7/'  IS LARGER THAN THE TVEC DIMENSION',I7,' SPECIFIED BY THE LIV12000
     1USER.'/'  USER MAY RESET THE TVEC DIMENSION AND RESTART THE PROGRALIV12010
```

```
     1M')                                                              LIV12020
      GO TO 1670                                                       LIV12030
C                                                                      LIV12040
 1550 WRITE(6,1560)                                                    LIV12050
 1560 FORMAT(/' PROGRAM TERMINATES BECAUSE NO SUITABLE T-EIGENVECTORS WELIV12060
     1RE IDENTIFIED'/'  FOR ANY OF THE EIGENVALUES SUPPLIED.  PROBLEM COLIV12070
     1ULD BE CAUSED'/'  BY TOO SMALL A TVEC DIMENSION OR SIMPLY THAT SUILIV12080
     1TABLE T-VECTORS COULD'/' NOT BE IDENTIFIED.  USER SHOULD EXAMINE OLIV12090
     1UTPUT'/)                                                         LIV12100
      GO TO 1670                                                       LIV12110
C                                                                      LIV12120
 1570 WRITE(6,1580) LVCONT,NTVEC,NGOOD                                 LIV12130
 1580 FORMAT(/' LVCONT FLAG =',I2,' AND NUMBER ',I5,' OF T-EIGENVECTORS LIV12140
     1 COMPUTED N.E.'/' NUMBER',I5,' REQUESTED SO PROGRAM TERMINATES') LIV12150
      GO TO 1670                                                       LIV12160
C                                                                      LIV12170
 1590 WRITE(6,1600)                                                    LIV12180
 1600 FORMAT(/' PROGRAM TERMINATES WITHOUT COMPUTING RITZ VECTORS'/     LIV12190
     1' BECAUSE ALL T-EIGENVECTORS WERE REJECTED AS NOT SUITABLE FOR THELIV12200
     1RITZ VECTOR'/' COMPUTATIONS.  PROBABLE CAUSE IS LACK OF CONVERGENCLIV12210
     1E OF EIGENVALUES SUPPLIED')                                      LIV12220
       GO TO 1670                                                      LIV12230
C                                                                      LIV12240
 1610 WRITE(6,1620)                                                    LIV12250
 1620 FORMAT(/' PROGRAM INDICATES THAT IT IS NOT POSSIBLE TO COMPUTE ANYLIV12260
     1 OF THE REQUESTED EIGENVECTORS.'/' THEREFORE PROGRAM TERMINATES') LIV12270
      DO 1630 J=1,NGOODC                                               LIV12280
 1630 WRITE(6,1640)  J,GOODBI(J),MP(J)                                 LIV12290
 1640 FORMAT(/4X,' J',11X,'GOODBI(J)',4X,'MP(J)'/I6,E20.12,I9/)        LIV12300
      GO TO 1670                                                       LIV12310
C                                                                      LIV12320
 1650 WRITE(6,1660) MBETA,KMAXN                                        LIV12330
 1660 FORMAT(/' PROGRAM TERMINATES BECAUSE THE STORAGE ALLOTTED FOR THE LIV12340
     1BETA ARRAY',I8,/' IS NOT SUFFICIENT FOR THE ENLARGED KMAX =',I8,' LIV12350
     1THAT THE PROGRAM WANTS.'/' USER CAN ENLARGE THE ALPHA,BETA ARRAYS LIV12360
     1 AND RERUN THE PROGRAM'/)                                        LIV12370
C                                                                      LIV12380
 1670 CONTINUE                                                         LIV12390
C                                                                      LIV12400
      STOP                                                             LIV12410
C-----END EIGENVECTOR COMPUTATIONS FOR INVERSES OF REAL SYMMETRIC-------LIV12420
      END                                                              LIV12430
```

4.4 LIMULT: LANCZS and Sample Matrix-Vector Multiply Subroutines

```
C---LIMULT-(INVERSES OF REAL SYMMETRIC MATRICES)-----------------------LIM00010
C  Authors:  Jane Cullum and Ralph A. Willoughby (deceased)             LIM00020
C            Los Alamos National Laboratory                            LIM00030
C            Los Alamos, New Mexico 87544                              LIM00040
C                                                                      LIM00050
C            E-mail:  cullumj@lanl.gov                                 LIM00060
C                                                                      LIM00070
C  These codes are copyrighted by the authors.  These codes            LIM00080
C  and modifications of them or portions of them are NOT to be         LIM00090
C  incorporated into any commercial codes or used for any other        LIM00100
C  commercial purposes such as consulting for other companies,         LIM00110
C  without legal agreements with the authors of these Codes.           LIM00120
C  If these Codes or portions of them are used in other scientific or  LIM00130
C  engineering research works the names of the authors of these codes  LIM00140
C  and appropriate references to their written work are to be          LIM00150
C  incorporated in the derivative works.                               LIM00160
C                                                                      LIM00170
C  This header is not to be removed from these codes.                  LIM00180
C                                                                      LIM00190
C        REFERENCE: Cullum and Willoughby, Chapters 1,2,3,4            LIM00191
C        Lanczos Algorithms for Large Symmetric Eigenvalue ComputationsLIM00192
C        VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in   LIM00193
C        Applied Mathematics, 2002. SIAM Publications,                 LIM00194
C        Philadelphia, PA. USA                                         LIM00195
C                                                                      LIM00196
C                                                                      LIM00200
C     CONTAINS SUBROUTINE LANCZS AND SAMPLE USPEC AND BSOLV            LIM00210
C     USED BY THE VERSION OF THE LANCZOS ALGORITHMS FOR                LIM00220
C     FACTORED INVERSES OF REAL SYMMETRIC MATRICES, LIVAL AND LIVEC.   LIM00230
C                                                                      LIM00240
C     NONPORTABLE CONSTRUCTIONS:                                       LIM00250
C     1.  THE ENTRY MECHANISM USED TO PASS THE STORAGE LOCATIONS       LIM00260
C         OF THE FACTORIZATION OF THE MATRIX TO BE USED BY             LIM00270
C         LANCZS TO THE SOLVE SUBROUTINE BSOLV.                        LIM00280
C     2.  IN THE SAMPLE USPEC SUBROUTINES PROVIDED:                    LIM00290
C         THE FREE FORMAT (7,*) AND FORMATS (20A4) AND  (4Z20)         LIM00300
C         USED IN DEFINING THE MATRICES.                               LIM00310
C                                                                      LIM00320
C-----LANCZS-COMPUTE LANCZOS TRIDIAGONAL MATRICES----------------------LIM00330
C                                                                      LIM00340
      SUBROUTINE LANCZS(MATVEC,ALPHA,BETA,V1,V2,VS,G,KMAX,MOLD1,N,IIX) LIM00350
C                                                                      LIM00360
C----------------------------------------------------------------------LIM00370
      DOUBLE PRECISION  ALPHA(1), BETA(1), V1(1), V2(1), VS(1)         LIM00380
      DOUBLE PRECISION SUM, ONE, ZERO, TEMP                            LIM00390
      REAL  G(1)                                                       LIM00400
      EXTERNAL MATVEC                                                  LIM00410
      DOUBLE PRECISION  FINPRO, DSQRT                                  LIM00420
C----------------------------------------------------------------------LIM00430
C     ALPHA, BETA, LANCZOS VECTOR GENERATION                           LIM00440
C     ALPHA BETA GENERATION STARTS WITH IVEC = 1, BETA(1) = ZERO       LIM00450
```

```
C     V2 =  RANDOM UNIT VECTOR AND V1 = ZERO, OR EXTENDS                LIM00460
C     AN EXISTING ALPHA/BETA FILE.                                      LIM00470
C                                                                       LIM00480
      ZERO = 0.0D0                                                      LIM00490
      ONE  = 1.0D0                                                      LIM00500
      IF (MOLD1.GT.1) GO TO 30                                          LIM00510
      BETA(1) = ZERO                                                    LIM00520
      IIL = IIX                                                         LIM00530
C                                                                       LIM00540
C-----------------------------------------------------------------------LIM00550
      CALL GENRAN(IIL,G,N)                                              LIM00560
C-----------------------------------------------------------------------LIM00570
C                                                                       LIM00580
      DO 10 K = 1,N                                                     LIM00590
   10 V2(K) = G(K)                                                      LIM00600
C                                                                       LIM00610
C-----------------------------------------------------------------------LIM00620
      SUM = FINPRO(N,V2(1),1,V2(1),1)                                   LIM00630
C-----------------------------------------------------------------------LIM00640
C                                                                       LIM00650
      SUM = ONE/DSQRT(SUM)                                              LIM00660
C                                                                       LIM00670
      DO 20 K = 1,N                                                     LIM00680
      V1(K) = ZERO                                                      LIM00690
   20 V2(K) = SUM*V2(K)                                                 LIM00700
C                                                                       LIM00710
   30 CONTINUE                                                          LIM00720
C                                                                       LIM00730
      DO 80 IVEC = MOLD1,KMAX                                           LIM00740
C                                                                       LIM00750
      DO 40 K = 1,N                                                     LIM00760
   40 VS(K) = V2(K)                                                     LIM00770
C                                                                       LIM00780
C-----------------------------------------------------------------------LIM00790
      JBSOLV = 2                                                        LIM00800
      CALL MATVEC(VS,VS,JBSOLV)                                         LIM00810
C-----------------------------------------------------------------------LIM00820
C                                                                       LIM00830
C     VS = B(INVERSE)*V2                                                LIM00840
C                                                                       LIM00850
      SUM = BETA(IVEC)                                                  LIM00860
C                                                                       LIM00870
      DO 50 K = 1,N                                                     LIM00880
   50 V1(K) = VS(K)-SUM*V1(K)                                           LIM00890
C                                                                       LIM00900
C-----------------------------------------------------------------------LIM00910
      SUM = FINPRO(N,V1(1),1,V2(1),1)                                   LIM00920
C-----------------------------------------------------------------------LIM00930
C                                                                       LIM00940
      ALPHA(IVEC) = SUM                                                 LIM00950
C                                                                       LIM00960
      DO 60 K = 1,N                                                     LIM00970
   60 V1(K) = V1(K)-SUM*V2(K)                                           LIM00980
C                                                                       LIM00990
C-----------------------------------------------------------------------LIM01000
```

```
      SUM = FINPRO(N,V1(1),1,V1(1),1)                                   LIM01010
C-----------------------------------------------------------------------LIM01020
C                                                                       LIM01030
      IN = IVEC+1                                                       LIM01040
C                                                                       LIM01050
      BETA(IN) = DSQRT(SUM)                                             LIM01060
      SUM = ONE/BETA(IN)                                                LIM01070
C                                                                       LIM01080
      DO 70 K = 1,N                                                     LIM01090
      TEMP = SUM*V1(K)                                                  LIM01100
      V1(K) = V2(K)                                                     LIM01110
   70 V2(K) = TEMP                                                      LIM01120
C                                                                       LIM01130
   80 CONTINUE                                                          LIM01140
C                                                                       LIM01150
      RETURN                                                            LIM01160
C-----END LANCZS------------------------------------------------------LIM01170
      END                                                               LIM01180
C                                                                       LIM01190
C-----USPEC FOR FACTORED INVERSES OF REAL SYMMETRIC MATRICES-----------LIM01200
C                                                                       LIM01210
      SUBROUTINE CUSPEC(N,MATNO)                                        LIM01220
C     SUBROUTINE USPEC(N,MATNO)                                         LIM01230
C                                                                       LIM01240
C-----------------------------------------------------------------------LIM01250
      DOUBLE PRECISION  BD(2200),BSD(10000)                             LIM01260
      REAL EXPLAN(20)                                                   LIM01270
      INTEGER  KCOL(2200),KROW(10000),IPR(2200),IPT(2200)               LIM01280
C-----------------------------------------------------------------------LIM01290
C     NOTE THAT THIS SUBROUTINE ASSUMES THAT B IS POSITIVE DEFINITE.    LIM01300
C     USER COULD REPLACE THIS SUBROUTINE AND CORRESPONDING SAMPLE       LIM01310
C     USPEC SUBROUTINE BY ONE THAT WORKS WITH GENERAL FACTORIZATION.    LIM01320
C                                                                       LIM01330
C     DIMENSIONS ARRAYS NEEDED TO DEFINE CHOLESKY FACTOR OF B-MATRIX,   LIM01340
C     READS CHOLESKY FACTOR FROM FILE 7, AND THEN PASSES STORAGE        LIM01350
C     LOCATIONS OF THESE ARRAYS TO THE B-MATRIX SOLVE SUBROUTINE BSOLV. LIM01360
C                                                                       LIM01370
C     HERE WE HAVE B = P*C*P' = L*L'  WHERE  C = S0*A + SHIFT*I.        LIM01380
C     P IS A PERMUTATION MATRIX DEFINED BY THE VECTOR MAPS IPR AND IPT. LIM01390
C     THE ITH ROW OF B CORRESPONDS TO THE JTH ROW OF C (A) WHERE        LIM01400
C     J = IPR(I) AND I = IPT(J).   A IS THE ORIGINAL MATRIX.            LIM01410
C                                                                       LIM01420
C     THE B-CHOLESKY FACTOR IS STORED IN THE FOLLOWING SPARSE FORMAT:   LIM01430
C     N = ORDER OF THE B-MATRIX.                                        LIM01440
C     NZT = NUMBER OF NONZERO SUBDIAGONAL ENTRIES IN THE CHOLESKY       LIM01450
C           FACTOR, L.                                                  LIM01460
C     KCOL(J), J=1,N IS THE NUMBER OF NONZERO SUBDIAGONAL ELEMENTS IN   LIM01470
C              COLUMN J OF L.                                           LIM01480
C     KROW(K), K=1,NZT IS THE ROW INDEX FOR CORRESPONDING ENTRY BSD(K). LIM01490
C     BD(J), J = 1,N CONTAINS THE DIAGONAL ENTRIES OF L.                LIM01500
C     BSD(K), K =1,NZT CONTAINS THE NONZERO SUBDIAGONAL ENTRIES OF L    LIM01510
C             BY COLUMN.                                                LIM01520
C     JPERM = (0,1):  1 MEANS CHOLEKSY FACTOR CORRESPONDS TO            LIM01530
C                     PERMUTED C.  0 MEANS NO PERMUTATION WAS USED.     LIM01540
C-----------------------------------------------------------------------LIM01550
```

```
C     READ CHOLESKY FACTOR FROM FILE 7.  MUST BE STORED                 LIM01560
C     IN SPARSE MATRIX FORMAT.                                          LIM01570
C                                                                       LIM01580
      READ(7,5) EXPLAN                                                  LIM01590
    5 FORMAT(20A4)                                                      LIM01600
C                                                                       LIM01610
      READ(7,10) NZT,NOLD,NZL,MATOLD,JPERM                              LIM01620
   10 FORMAT(I10,2I6,I8,I6)                                             LIM01630
C                                                                       LIM01640
      WRITE(6,20) NZT,NZL,N,NOLD,MATOLD,JPERM                           LIM01650
   20 FORMAT(' HEADER, CHOLESKY FACTOR FILE'/                           LIM01660
     1 3X,'NZT',3X,'NZL',5X,'N',2X,'NOLD',2X,'MATOLD',1X,'JPERM'/       LIM01670
     1 4I6,I8,I6/)                                                      LIM01680
C                                                                       LIM01690
      IF (N.NE.NOLD.OR.MATNO.NE.MATOLD) GO TO 70                        LIM01700
C                                                                       LIM01710
      READ(7,5) EXPLAN                                                  LIM01720
C                                                                       LIM01730
      READ(7,30) (KCOL(K), K = 1,NZL)                                   LIM01740
   30 FORMAT(13I6)                                                      LIM01750
C                                                                       LIM01760
      READ(7,5) EXPLAN                                                  LIM01770
C                                                                       LIM01780
      READ(7,30) (KROW(K), K = 1,NZT)                                   LIM01790
C                                                                       LIM01800
      READ(7,5) EXPLAN                                                  LIM01810
C                                                                       LIM01820
      READ(7,40) (BD(K), K = 1,N)                                       LIM01830
   40 FORMAT(4Z20)                                                      LIM01840
C                                                                       LIM01850
      READ(7,5) EXPLAN                                                  LIM01860
C                                                                       LIM01870
      READ(7,40) (BSD(K), K = 1,NZT)                                    LIM01880
C                                                                       LIM01890
C     DOES CHOLESKY FACTOR CORRESPOND TO PERMUTED B?                    LIM01900
      IF(JPERM.EQ.0)  GO TO 60                                          LIM01910
C                                                                       LIM01920
      READ(7,5) EXPLAN                                                  LIM01930
C                                                                       LIM01940
      READ(7,30) (IPR(K), K = 1,N)                                      LIM01950
C                                                                       LIM01960
      DO 50 K = 1,N                                                     LIM01970
      J = IPR(K)                                                        LIM01980
   50 IPT(J) = K                                                        LIM01990
C                                                                       LIM02000
C-----------------------------------------------------------------------LIM02010
      CALL LPERME(IPR,IPT,N)                                            LIM02020
C-----------------------------------------------------------------------LIM02030
C                                                                       LIM02040
   60 CONTINUE                                                          LIM02050
C                                                                       LIM02060
C-----------------------------------------------------------------------LIM02070
C     PASS STORAGE LOCATIONS OF FACTORS TO INVERSION SUBROUTINE BSOLV   LIM02080
      CALL BSOLVE(BSD,BD,KCOL,KROW,N,NZT,NZL)                           LIM02090
C-----------------------------------------------------------------------LIM02100
```

```
C                                                                       LIM02110
      GO TO 90                                                          LIM02120
C                                                                       LIM02130
   70 CONTINUE                                                          LIM02140
C     DEFAULT EXIT                                                      LIM02150
      WRITE(6,80)                                                       LIM02160
   80 FORMAT(' TERMINATE.  PARAMETERS IN CHOLESKY FACTOR FILE'/         LIM02170
     1' DO NOT AGREE WITH THOSE SPECIFIED BY THE USER'/)                LIM02180
      STOP                                                              LIM02190
C                                                                       LIM02200
   90 CONTINUE                                                          LIM02210
C-----END OF USPEC-----------------------------------------------------LIM02220
      RETURN                                                            LIM02230
      END                                                               LIM02240
C                                                                       LIM02250
C-----BSOLV-(FACTORED INVERSE OR L*L-TRANS MULTIPLY)--------------------LIM02260
C           (FOR POSITIVE DEFINITE SYMMETRIC SPARSE MATRICES)           LIM02270
C                                                                       LIM02280
C     SUBROUTINE BSOLV(V,U,JBSOLV)                                      LIM02290
      SUBROUTINE CBSOLV(V,U,JBSOLV)                                     LIM02300
C                                                                       LIM02310
C-----------------------------------------------------------------------LIM02320
      DOUBLE PRECISION  BD(1),BSD(1),U(1),V(1),TEMP,ZERO,ONE            LIM02330
      INTEGER  KCOL(1),KROW(1)                                          LIM02340
C-----------------------------------------------------------------------LIM02350
      GO TO 3                                                           LIM02360
      ENTRY BSOLVE(BSD,BD,KCOL,KROW,N,NZT,NZL)                          LIM02370
      GO TO 4                                                           LIM02380
C-----------------------------------------------------------------------LIM02390
C     JBSOLV = 2 MEANS SOLVE B*U = V                                    LIM02400
C     JBSOLV = 1 MEANS COMPUTE U = B*V:  NOTE THAT IN THIS CASE V IS    LIM02410
C     DESTROYED.  LANCZOS PROGRAMS AS WRITTEN DO NOT USE JBSOLV = 1     LIM02420
C     PATH.                                                             LIM02430
    3 ZERO = 0.0D0                                                      LIM02440
      ONE  = 1.0D0                                                      LIM02450
      IF (JBSOLV .EQ.2) GO TO 60                                        LIM02460
C     U = B*V WHERE  B = L*L'                                           LIM02470
      KL = 0                                                            LIM02480
      DO 20 J = 1,N                                                     LIM02490
      TEMP = V(J)*BD(J)                                                 LIM02500
      IF (KCOL(J).EQ.0.OR.J.EQ.N) GO TO 20                              LIM02510
      KF = KL + 1                                                       LIM02520
      KL  = KL + KCOL(J)                                                LIM02530
      DO 10 K = KF,KL                                                   LIM02540
      IK = KROW(K)                                                      LIM02550
   10 TEMP = BSD(K)*V(IK) + TEMP                                        LIM02560
   20 V(J) = TEMP                                                       LIM02570
C     V = L'*V                                                          LIM02580
      DO 30 K = 1,N                                                     LIM02590
   30 U(K) = V(K)*BD(K)                                                 LIM02600
      KL = 0                                                            LIM02610
      DO 50 K = 1,N                                                     LIM02620
      TEMP = V(K)                                                       LIM02630
      IF (KCOL(K).EQ.0.OR.K.EQ.N) GO TO 50                              LIM02640
      KF = KL + 1                                                       LIM02650
```

```
      KL = KL + KCOL(K)                                                 LIM02660
      DO 40 KK = KF,KL                                                  LIM02670
      KR = KROW(KK)                                                     LIM02680
   40 U(KR) = U(KR) + TEMP*BSD(KK)                                      LIM02690
   50 CONTINUE                                                          LIM02700
      GO TO 120                                                         LIM02710
C     U = B*V                                                           LIM02720
C-----------------------------------------------------------------      LIM02730
   60 CONTINUE                                                          LIM02740
C     SOLVE B*U = V FOR U WHERE  B = L*L'                               LIM02750
C     SET U = V. FIRST SOLVE L*U = U FOR U, THEN SOLVE L'*U = U FOR U   LIM02760
      KL = 0                                                            LIM02770
      DO 70 K = 1,N                                                     LIM02780
   70 U(K) = V(K)                                                       LIM02790
      DO 90 K = 1,N                                                     LIM02800
      TEMP = U(K)/BD(K)                                                 LIM02810
      U(K) = TEMP                                                       LIM02820
      IF (KCOL(K).EQ.0.OR.K.EQ.N) GO TO 90                              LIM02830
      KF = KL + 1                                                       LIM02840
      KL = KL + KCOL(K)                                                 LIM02850
      DO 80 KK = KF,KL                                                  LIM02860
      KR = KROW(KK)                                                     LIM02870
   80 U(KR) = U(KR) - TEMP*BSD(KK)                                      LIM02880
   90 CONTINUE                                                          LIM02890
      NP1 = N+1                                                         LIM02900
      KF = NZT + 1                                                      LIM02910
      DO 110 K = 1,N                                                    LIM02920
      L = NP1 - K                                                       LIM02930
      TEMP = U(L)                                                       LIM02940
      IF (KCOL(L).EQ.0.OR.L.EQ.N) GO TO 110                             LIM02950
      KL = KF - 1                                                       LIM02960
      KF = KF - KCOL(L)                                                 LIM02970
      DO 100 LL = KF,KL                                                 LIM02980
      LR = KROW(LL)                                                     LIM02990
  100 TEMP = TEMP - BSD(LL)*U(LR)                                       LIM03000
  110 U(L) = TEMP/BD(L)                                                 LIM03010
  120 CONTINUE                                                          LIM03020
C                                                                       LIM03030
    4 RETURN                                                            LIM03040
C                                                                       LIM03050
C-----END OF BSOLV-----------------------------------------------------LIM03060
      END                                                               LIM03070
C                                                                       LIM03080
C-----SUBROUTINES FOR DIAGONAL TEST MATRICES---------------------------LIM03090
C                                                                       LIM03100
C     BSOLV AND USPEC SUBROUTINES FOR DIAGONAL TEST MATRICES            LIM03110
C                                                                       LIM03120
C-----BSOLV DIAGONAL TEST MATRIX---------------------------------------LIM03130
C                                                                       LIM03140
C     SUBROUTINE DBSOLV(V,U,JBSOLV)                                     LIM03150
      SUBROUTINE BSOLV(V,U,JBSOLV)                                      LIM03160
C                                                                       LIM03170
C-----------------------------------------------------------------------LIM03180
      DOUBLE PRECISION  V(1),U(1),D(1)                                  LIM03190
C-----------------------------------------------------------------------LIM03200
```

```
      GO TO 3                                                          LIM03210
C     BELOW ENTRY IS FOR A DIAGONAL TEST MATRIX                        LIM03220
      ENTRY DSOLVE(D,N)                                                LIM03230
      GO TO 4                                                          LIM03240
C-----------------------------------------------------------------------LIM03250
C     JBSOLV = 1, COMPUTE U = D*V. (NOTE THIS IS NOT USED)             LIM03260
C     JBSOLV = 2, COMPUTE U = (D-INVERSE)*V                            LIM03270
    3 IF(JBSOLV.EQ.2) GO TO 20                                         LIM03280
      DO 10 I=1,N                                                      LIM03290
   10 U(I) = D(I)*V(I)                                                 LIM03300
      GO TO 40                                                         LIM03310
C                                                                      LIM03320
   20 DO 30 I=1,N                                                      LIM03330
   30 U(I)= V(I)/D(I)                                                  LIM03340
C                                                                      LIM03350
   40 CONTINUE                                                         LIM03360
    4 RETURN                                                           LIM03370
C                                                                      LIM03380
C-----END OF BSOLV FOR DIAGONAL TEST MATRIX ----------------------------LIM03390
      END                                                              LIM03400
C                                                                      LIM03410
C-----START OF USPEC FOR DIAGONAL TEST MATRIX--------------------------LIM03420
C                                                                      LIM03430
      SUBROUTINE USPEC(N,MATNO)                                        LIM03440
C     SUBROUTINE DUSPEC(N,MATNO)                                       LIM03450
C                                                                      LIM03460
C-----------------------------------------------------------------------LIM03470
      DOUBLE PRECISION  D(1000), DI(1000), SHIFT, SPACE                LIM03480
      DOUBLE PRECISION  DABS, DFLOAT                                   LIM03490
      REAL  EXPLAN(20)                                                 LIM03500
C-----------------------------------------------------------------------LIM03510
C                                                                      LIM03520
      READ(7,10) EXPLAN                                                LIM03530
   10 FORMAT(20A4)                                                     LIM03540
      READ(7,*) NOLD,NUNIF,SPACE,D(1),SHIFT                            LIM03550
      NNUNIF = NOLD - NUNIF                                            LIM03560
      WRITE(6,20) NOLD,SPACE,NNUNIF,D(1),SHIFT                         LIM03570
   20 FORMAT(/' DIAGONAL TEST MATRIX, SIZE = ',I4/' IS THE INVERSE OF MALIM03580
     1TRIX WITH MOST ENTRIES',E10.3/' UNITS APART AND WITH ',I3,' ENTRIELIM03590
     1S IRREGULARLY SPACED'/' FIRST ENTRY WAS ',E13.4,' SHIFT = ',E10.3 LIM03600
     1/)                                                               LIM03610
C                                                                      LIM03620
      IF(N.NE.NOLD) GO TO 100                                          LIM03630
C     COMPUTE THE UNIFORM PORTION OF THE SPECTRUM                      LIM03640
      DO 30 J=2,NUNIF                                                  LIM03650
   30 D(J) = D(1) - DFLOAT(J-1)*SPACE                                  LIM03660
      NUNIF1=NUNIF + 1                                                 LIM03670
      READ(7,10)  EXPLAN                                               LIM03680
      DO 40 J=NUNIF1,N                                                 LIM03690
   40 READ(7,*) D(J)                                                   LIM03700
      NB = NUNIF - 2                                                   LIM03710
C                                                                      LIM03720
      IF(SHIFT.EQ.0.) GO TO 60                                         LIM03730
      DO 50 J=1,N                                                      LIM03740
   50 D(J) = D(J) + SHIFT                                              LIM03750
```

```
C                                                                       LIM03760
C     COMPUTE EIGENVALUES OF INVERSE FOR PRINTOUT ONLY                  LIM03770
   60 DO 70 J = 1,N                                                     LIM03780
   70 DI(J) = 1.D0/D(J)                                                 LIM03790
      WRITE(6,80) (DI(I), I=1,10 )                                      LIM03800
      WRITE(6,90) (DI(I), I = NB,N)                                     LIM03810
   80 FORMAT(/' INVERSE LANCZOS TEST, LANCZS USES INVERSE OF GIVEN MATRILIM03820
     1X'/' 1ST 10 ENTRIES OF INVERSE OF DIAGONAL TEST MATRIX = '/(3E22.1LIM03830
     14))                                                               LIM03840
   90 FORMAT(/' MIDDLE (ORIGINALLY UNIFORM) PORTION OF MATRIX IS NOT PRILIM03850
     1NTED OUT'/' END OF (UNIFORM) PLUS NONUNIFORM SECTION = '/(3E25.16)LIM03860
     1)                                                                 LIM03870
C                                                                       LIM03880
C     DIAGONAL GENERATION COMPLETE                                      LIM03890
C                                                                       LIM03900
C-----------------------------------------------------------------------LIM03910
C    PASS STORAGE LOCATIONS OF D AND N TO DSOLV SUBROUTINE              LIM03920
       CALL DSOLVE(D,N)                                                 LIM03930
C-----------------------------------------------------------------------LIM03940
C                                                                       LIM03950
      RETURN                                                            LIM03960
  100 WRITE(6,110) NOLD,N                                               LIM03970
  110 FORMAT(' PROGRAM TERMINATES BECAUSE NOLD = ',I5,'DOES NOT EQUAL N LIM03980
     1 =',I5)                                                           LIM03990
C-----END OF USPEC SUBROUTINE FOR DIAGONAL TEST MATRICES---------------LIM04000
      STOP                                                              LIM04010
      END                                                               LIM04020
```

4.5 PERMUT: LORDER: LFACT: LTEST: Optional Routines for Chapters 4, 5, 9

```
C-----PERMUT-(USES SPARSPAK PACKAGE)-----------------------------------PER00010
C  AUTHORS:  RALPH A. WILLOUGHBY (DECEASED)                            PER00020
C                                                                       PER00030
C                                                                       PER00040
C                                                                       PER00050
C                                                                       PER00060
C           E-mail:  cullumj@lanl.gov                                   PER00070
C                                                                       PER00080
C  These codes are copyrighted by the authors.  These codes             PER00090
C  and modifications of them or portions of them are NOT to be          PER00100
C  incorporated into any commercial codes or used for any other         PER00110
C  commercial purposes such as consulting for other companies,          PER00120
C  without legal agreements with the authors of these Codes.            PER00130
C  If these Codes or portions of them are used in other scientific or   PER00140
C  engineering research works the names of the authors of these codes   PER00150
C  and appropriate references to their written work are to be           PER00160
C  incorporated in the derivative works.                                PER00170
C                                                                       PER00180
C  This header is not to be removed from these codes.                   PER00190
C                                                                       PER00200
C                                                                       PER00210
C     OPTIONAL PREPROCESSING PROGRAM FOR USE WITH LANCZOS CODES.        PER00220
C     GIVEN A REAL SYMMETRIC A-MATRIX IN SPARSE MATRIX FORMAT, PERMUT   PER00230
C     CALLS THE SPARSPAK PACKAGE (A. GEORGE, J. LIU, E. NG, U. WATERLOO)PER00240
C     TO DETERMINE A REORDERING OF A, THAT IS A PERMUTATION MATRIX      PER00250
C     P, SUCH THAT SPARSITY IS PRESERVED IN THE FACTORIZATION OF        PER00260
C     THE PERMUTED MATRIX.  PERMUT ALSO MODIFIES THE GIVEN A-MATRIX     PER00270
C     TO FORM THE MATRIX C = S0*A + SHIFT*I, WHERE S0 AND SHIFT         PER00280
C     ARE SCALARS PROVIDED BY THE USER, AND THEN WRITES THIS            PER00290
C     C-MATRIX OUT TO FILE 9 ALONG WITH THE PERMUTATION P WHICH         PER00300
C     IS DEFINED BY THE VECTOR IPR.  IPR IS ALSO WRITTEN SEPARATELY     PER00310
C     TO FILE 14.                                                       PER00320
C                                                                       PER00330
C     NONPORTABLE CONSTRUCTIONS:                                        PER00340
C     1.  INTEGER*2 VARIABLE NPERM.  NOTE THAT THIS VARIABLE CANNOT     PER00350
C         BE CHANGED TO INTEGER*4.                                      PER00360
C     2.  FREE FORMAT (5,*) AND THE FORMAT (20A4).                      PER00370
C     3.  TO AVOID COMPOUNDING FORMAT CONVERSION ERRORS, THE MATRIX     PER00380
C         ENTRIES SHOULD BE STORED IN MACHINE FORMAT, ((4Z20) FOR       PER00390
C         IBM/3081)                                                     PER00400
C                                                                       PER00410
C-----------------------------------------------------------------------PER00420
C     SYMMETRIC A-MATRIX IS READ FROM FILE 8.  MATRIX IS STORED         PER00430
C     IN FOLLOWING SPARSE FORMAT:                                       PER00440
C                                                                       PER00450
C     NZL = INDEX OF LAST COLUMN CONTAINING NONZEROS BELOW THE DIAGONAL.PER00460
C     NZS = NUMBER OF NONZERO SUBDIAGONAL ENTRIES                       PER00470
C     ICOL(K), K=1,NZL CONTAINS THE NUMBER OF NONZERO SUBDIAGONAL       PER00480
C              ENTRIES IN COLUMN K.                                     PER00490
C     IROW(K), K=1,NZS CONTAINS ROW INDEX OF KTH NONZERO SUBDIAGONAL    PER00500
C              ENTRY, ENTRIES STORED COLUMN BY COLUMN.                  PER00510
```

```
C     AD(K), K=1,N CONTAINS THE DIAGONAL ENTRIES OF A, INCLUDING ANY    PER00520
C            ZERO ENTRIES.                                             PER00530
C     ASD(K), K=1,NZS CONTAINS THE NONZERO SUBDIAGONAL ENTRIES OF A,    PER00540
C             COLUMN BY COLUMN.                                        PER00550
C                                                                      PER00560
C-----INPUT/OUTPUT FILES ------------------------------------------------PER00570
C                                                                      PER00580
C    INPUT FILES:                                                      PER00590
C    FILE 5      CONTAINS THE PROGRAM PARAMETERS SET BY USER           PER00600
C    FILE 8      CONTAINS THE SPARSE A-MATRIX                          PER00610
C                                                                      PER00620
C    OUTPUT FILES:                                                     PER00630
C    FILE 6      INTERACTIVE TERMINAL FILE                             PER00640
C    FILE 9      CONTAINS THE SPARSE DATA FOR C = S0*A + SHIFT*I.      PER00650
C    FILE 14     CONTAINS PERMUTATION IPR DEFINING THE REORDERING.     PER00660
C                IN PARTICULAR J = IPR(I) MEANS ROW(COL) I OF          PER00670
C                B = P*C*(P-TRANSPOSE) CORRESPONDS TO ROW(COL) J       PER00680
C                OF THE A-MATRIX.                                      PER00690
C                                                                      PER00700
C-----SPARSPAK-----------------------------------------------------------PER00710
C     ARRAYS AND PARAMETERS THAT ARE REQUIRED BY SPARSPAK.             PER00720
C     NOTE THAT THE CALL FOR SPARSPAK IS SPRSPK.  SUBROUTINES          PER00730
C     IJBEGN, INIJ, IJEND, ORDRB5, AND PSTATS ARE SPARSPAK             PER00740
C     SUBROUTINES.                                                     PER00750
C                                                                      PER00760
C     S = VECTOR WHOSE ACTUAL DIMENSION IS DETERMINED BY SPARSPAK      PER00770
C         WHEN THE REORDERING IS OBTAINED.  USER SPECIFIES MAXIMUM     PER00780
C         DIMENSION MAXS ALLOWED; SPARSPAK DEFAULTS IF THIS MAXIMUM    PER00790
C         IS EXCEEDED.  SPARSPAK IS DESIGNED FOR SOLVING SYSTEMS       PER00800
C         OF EQUATIONS, THUS THE VECTOR S IS DESIGNED TO CONTAIN       PER00810
C         THE SOLUTION VECTOR IF THERE IS ONE, FOLLOWED BY THE         PER00820
C         PERMUTATION VECTOR IPR, FOLLOWED BY OTHER INFORMATION        PER00830
C         GENERATED BY SPARSPAK.  A CORRECT SIZE FOR MAXS CAN BE       PER00840
C         DETERMINED ONLY AFTER THE FACT.  AS A FIRST GUESS ONE        PER00850
C         CAN SET MAXS = K*N WHERE K >= 10.                            PER00860
C                                                                      PER00870
C     MSGLVL = CONTROL FOR WRITES TO FILE 6                            PER00880
C     NEQNS = ORDER OF A, THIS IS COMPUTED BY SPARSPAK                 PER00890
C     IERR = CONTROLS WRITING OF ERROR MESSAGES BY SPARSPAK.           PER00900
C     MAXS = USER-SPECIFIED MAXIMUM ALLOWED DIMENSION OF S-ARRAY.      PER00910
C                                                                      PER00920
C                                                                      PER00930
C-----------------------------------------------------------------------PER00940
      DOUBLE PRECISION AD(3000),ASD(10000),S0,SHIFT                    PER00950
      DOUBLE PRECISION S(30000),STEMP                                  PER00960
      REAL EXPLAN(20)                                                  PER00970
      INTEGER ICOL(3000),IROW(10000),IPR(3000)                         PER00980
      INTEGER*2 NPERM(4)                                               PER00990
      COMMON /SPKUSR/ MSGLVL,IERR,MAXS,NEQNS                           PER01000
      EQUIVALENCE (STEMP,NPERM(1))                                     PER01010
C-----------------------------------------------------------------------  PER01020
C                                                                      PER01030
C     ARRAYS MUST BE DIMENSIONED AS FOLLOWS:                           PER01040
C     1.  AD:  >= N, THE ORDER OF A-MATRIX.                            PER01050
C     2.  ASD: >= NZS, THE NUMBER OF NONZERO SUBDIAGONAL ENTRIES IN A.  PER01060
```

```
C     4.  ICOL:   >= N                                                  PER01070
C     5.  IROW:   >= NZS                                                PER01080
C     6.  IPR:    >= N + 4                                              PER01090
C     7.  S:   >= MAXS                                                  PER01100
C                                                                       PER01110
C-----------------------------------------------------------------------PER01120
C                                                                       PER01130
      WRITE(6,10)                                                       PER01140
   10 FORMAT(/' CALL SPARSPAK TO FIND REORDERING OF THE GIVEN MATRIX'/   PER01150
     1' THAT PRESERVES SPARSITY IN THE FACTORIZATION'/)                 PER01160
C                                                                       PER01170
C     READ IN USER-SPECIFIED PARAMETERS                                 PER01180
C     SPECIFY THE MAXIMUM DIMENSION (MAXS) ALLOWED FOR ARRAY S,         PER01190
C     AND WHETHER OR NOT A-MATRIX IS BEING SCALED BY S0 OR SHIFTED      PER01200
C     ISCALE = 0, THEN NO SCALING OR SHIFTING                           PER01210
      READ (5,20) EXPLAN                                                PER01220
      WRITE (6,20) EXPLAN                                               PER01230
      READ (5,20) EXPLAN                                                PER01240
   20 FORMAT(20A4)                                                      PER01250
      READ (5,*) MAXS,ISCALE,S0,SHIFT                                   PER01260
C                                                                       PER01270
C     READ IN INDICES FOR A-MATRIX STORED IN SPARSE FORMAT ON FILE 8.   PER01280
C     ONLY THE NONZERO STRUCTURE IS NEEDED TO OBTAIN THE PERMUTATION.   PER01290
C                                                                       PER01300
      READ(8,30) NZS,N,NZL,MATNO                                        PER01310
   30 FORMAT(I10,2I6,I8)                                                PER01320
C                                                                       PER01330
      WRITE(6,40) NZS,N,NZL,MATNO,MAXS,ISCALE,SHIFT,S0                  PER01340
   40 FORMAT(/I10,2I6,I10,I10,' = NZS,N,NZL,MATNO,MAXS'/                PER01350
     1 I6,2E12.5,' = ISCALE,SHIFT,S0'/)                                 PER01360
C                                                                       PER01370
      READ(8,50) (ICOL(K), K=1,NZL)                                     PER01380
      READ(8,50) (IROW(K), K=1,NZS)                                     PER01390
   50 FORMAT(13I6)                                                      PER01400
C                                                                       PER01410
C     DIAGONAL IS READ (INCLUDING ANY ZERO ENTRIES), THEN NONZERO       PER01420
C     BELOW DIAGONAL ENTRIES ARE READ IN                                PER01430
      READ(8,60) (AD(K), K=1,N)                                         PER01440
      READ(8,60) (ASD(K), K=1,NZS)                                      PER01450
   60 FORMAT(4E19.10)                                                   PER01460
C                                                                       PER01470
      IF (ISCALE.EQ.0) GO TO 90                                         PER01480
C                                                                       PER01490
C     CALCULATE C = S0*A + SHIFT*I AND PUT IN A-ARRAY                   PER01500
      DO 70 K = 1,N                                                     PER01510
   70 AD(K) = S0*AD(K) + SHIFT                                          PER01520
      DO 80 K = 1,NZS                                                   PER01530
   80 ASD(K) = S0*ASD(K)                                                PER01540
   90 CONTINUE                                                          PER01550
C                                                                       PER01560
C-----------------------------------------------------------------------PER01570
C     INPUT THE SPARSENESS STRUCTURE OF GIVEN A-MATRIX TO SPARSPAK      PER01580
      CALL SPRSPK                                                       PER01590
C-----------------------------------------------------------------------PER01600
C                                                                       PER01610
```

```
      MSGLVL = 4                                                        PER01620
C                                                                       PER01630
C-----------------------------------------------------------------------PER01640
      CALL IJBEGN                                                       PER01650
C-----------------------------------------------------------------------PER01660
C                                                                       PER01670
      LLAST = 0                                                         PER01680
      DO 110 J = 1,NZL                                                  PER01690
      IF (ICOL(J).EQ.0) GO TO 110                                       PER01700
      JJ = J                                                            PER01710
      LFIRST = LLAST + 1                                                PER01720
      LLAST = LLAST + ICOL(J)                                           PER01730
      DO 100 L = LFIRST,LLAST                                           PER01740
      II = IROW(L)                                                      PER01750
C                                                                       PER01760
C-----------------------------------------------------------------------PER01770
      CALL INIJ(II,JJ,S)                                                PER01780
C-----------------------------------------------------------------------PER01790
C                                                                       PER01800
  100 CONTINUE                                                          PER01810
C                                                                       PER01820
  110 CONTINUE                                                          PER01830
C                                                                       PER01840
C     SPARSENESS STRUCTURE HAS BEEN INPUTED TO SPARSPAK.                PER01850
C                                                                       PER01860
C-----------------------------------------------------------------------PER01870
      CALL IJEND(S)                                                     PER01880
C-----------------------------------------------------------------------PER01890
C                                                                       PER01900
      WRITE(6,120) N,NEQNS                                              PER01910
  120 FORMAT(/2I6,' = N,NEQNS'/)                                        PER01920
      IF (N.NE.NEQNS) GO TO 230                                         PER01930
C                                                                       PER01940
C-----------------------------------------------------------------------PER01950
C     USE SPARSPAK TO GENERATE REORDERING OF A THAT PRESERVES           PER01960
C     SPARSITY.  CORRESPONDING FACTORIZATION CAN BE COMPUTED BY         PER01970
C     PREPROCESSING PROGRAM LFACT WHEN C = S0*A + SHIFT*I IS POSITIVE   PER01980
C     DEFINITE. BELOW CALLS THE MINIMUM DEGREE ALGORITHM PROVIDED       PER01990
C     IN SPARSPAK.                                                      PER02000
      CALL ORDRB5(S)                                                    PER02010
      CALL PSTATS                                                       PER02020
C-----------------------------------------------------------------------PER02030
C                                                                       PER02040
C     EXTRACT THE REORDERING FROM SPARSPAK S VECTOR AND STORE IN FILE 14PER02050
      L = 1                                                             PER02060
      KNUM = N                                                          PER02070
      DO 130 K = 1,N                                                    PER02080
      KNUM = KNUM + 1                                                   PER02090
      STEMP = S(KNUM)                                                   PER02100
      IPR(L) = NPERM(1)                                                 PER02110
      IPR(L+1) = NPERM(2)                                               PER02120
      IPR(L+2) = NPERM(3)                                               PER02130
      IPR(L+3) = NPERM(4)                                               PER02140
      L = L+4                                                           PER02150
      IF (L.GT.N) GO TO 140                                             PER02160
```

```
  130 CONTINUE                                                          PER02170
  140 CONTINUE                                                          PER02180
C                                                                       PER02190
      WRITE(14,150) N,MATNO                                             PER02200
  150 FORMAT(I6,I8,' = N MATNO  K IPR(K)   A-MATRIX PERMUTATION')       PER02210
      WRITE(14,160) (K,IPR(K), K = 1,N)                                 PER02220
  160 FORMAT(6(1X,2I6))                                                 PER02230
C                                                                       PER02240
C                                                                       PER02250
C     WRITE C = S0*A + SHIFT*I WITH THE PERMUTATION IPR TO FILE 9.      PER02260
C                                                                       PER02270
      JPERM = 1                                                         PER02280
      WRITE(9,170) NZS,N,NZL,MATNO,JPERM                                PER02290
  170 FORMAT(I10,2I6,I8,I6,' = NZS,N,NZL,MATNO,JPERM. ACOMPAC')         PER02300
C                                                                       PER02310
C     NUMBER OF NONZERO SUBDIAGONAL ENTRIES IN EACH COLUMN IS WRITTEN   PER02320
C     THEN THE CORRESPONDING ROW INDEX FOR EACH SUCH ENTRY IS WRITTEN   PER02330
      WRITE(9,180) (ICOL(K), K=1,NZL)                                   PER02340
      WRITE(9,180) (IROW(K), K=1,NZS)                                   PER02350
  180 FORMAT(13I6)                                                      PER02360
C     DIAGONAL IS WRITTEN FIRST, THEN NONZERO BELOW DIAGONAL ENTRIES    PER02370
      WRITE(9,190) (AD(K), K=1,N)                                       PER02380
      WRITE(9,190) (ASD(K), K=1,NZS)                                    PER02390
  190 FORMAT(4E19.10)                                                   PER02400
      WRITE(9,180) (IPR(K), K=1,N)                                      PER02410
C                                                                       PER02420
      IF(ISCALE.NE.0) GO TO 200                                         PER02430
C     ISCALE = 0, SET DEFAULT VALUES OF S0 AND SHIFT                    PER02440
      S0 = 1.D0                                                         PER02450
      SHIFT = 0.D0                                                      PER02460
  200  WRITE(9,210) S0,SHIFT                                            PER02470
  210 FORMAT(2E12.5,' = S0 SHIFT'/                                      PER02480
     1 ' ABOVE IS SPARSE DATA FOLLOWED BY PERMUTATION IPR'/             PER02490
     1 ' FOR THE MATRIX C = S0*A+SHIFT*I '/                             PER02500
     1 ' B = P*C*PTRANS CAN BE GENERATED IN SUBROUTINE LORDER'/         PER02510
     1 ' ROW(COL) I OF B CORRESPONDS TO ROW(COL) J OF C, J = IPR(I)'/   PER02520
     1 ' NZS = TOTAL NUMBER OF SUBDIAGONAL NONZEROS IN C'/              PER02530
     1 ' KCOL(K) = NUMBER OF SUBDIAGONAL NONZEROS IN COL K OF C'/       PER02540
     1 ' KROW(K) = ROW INDEX OF SUBDIAGONAL NONZERO'/                   PER02550
     1 ' SUBDIAGONAL NONZEROS IN C ARE STORED COLUMN BY COLUMN'/        PER02560
     1 ' AD(K) = THE KTH DIAGONAL ELEMENT OF C'/                        PER02570
     1 ' ASD(K) =  KTH SUBDIAGONAL NONZERO IN C'/)                      PER02580
C                                                                       PER02590
      WRITE(6,220)                                                      PER02600
  220 FORMAT(/' PERMUT IS FINISHED MATRIX IS ON FILE 9'/)               PER02610
C                                                                       PER02620
  230 CONTINUE                                                          PER02630
C                                                                       PER02640
C-----END PERMUT-------------------------------------------------------PER02650
      STOP                                                              PER02660
      END                                                               PER02670
```

```
C-----LORDER-(STAND ALONE PROGRAM)--------------------------------------LOR00010
C  AUTHORS:  RALPH A. WILLOUGHBY (DECEASED)                             LOR00020
C                                                                       LOR00030
C                                                                       LOR00040
C                                                                       LOR00050
C            E-mail:  cullumj@lanl.gov                                  LOR00060
C                                                                       LOR00070
C  These codes are copyrighted by the authors.  These codes             LOR00080
C  and modifications of them or portions of them are NOT to be          LOR00090
C  incorporated into any commercial codes or used for any other         LOR00100
C  commercial purposes such as consulting for other companies,          LOR00110
C  without legal agreements with the authors of these Codes.            LOR00120
C  If these Codes or portions of them are used in other scientific or   LOR00130
C  engineering research works the names of the authors of these codes   LOR00140
C  and appropriate references to their written work are to be           LOR00150
C  incorporated in the derivative works.                                LOR00160
C                                                                       LOR00170
C  This header is not to be removed from these codes.                   LOR00180
C                                                                       LOR00190
C                                                                       LOR00200
C     ACCORDING TO THE PFORT VERIFIER THIS PROGRAM IS PORTABLE.         LOR00210
C     HOWEVER TO AVOID COMPOUNDING FORMAT CONVERSION ERRORS,            LOR00220
C     MATRIX ENTRIES SHOULD BE STORED IN MACHINE FORMAT, ((4Z20)        LOR00230
C     FOR IBM/3081).                                                    LOR00240
C                                                                       LOR00250
C     LORDER TAKES A SPARSE MATRIX C AND A PERMUTATION P GIVEN BY       LOR00260
C     THE VECTOR IPR AND COMPUTES THE PERMUTED MATRIX B = P*C*P' ,      LOR00270
C     AND THEN WRITES B TO FILE 9 ALONG WITH IPR AND ANY SCALE S0       LOR00280
C     AND SHIFT THAT WERE USED TO OBTAIN THE INPUT MATRIX C. (HERE      LOR00290
C     ROW(COL) I OF B CORRESPONDS TO ROW(COL) J OF A WHERE J = IPR(I),  LOR00300
C     AND INPUT MATRIX C = S0*A + SHIFT*I.                              LOR00310
C                                                                       LOR00320
C-----------------------------------------------------------------------LOR00330
      DOUBLE PRECISION ASD(10000),AD(3000),BSD(10000),BD(3000)          LOR00340
      DOUBLE PRECISION SHIFT,S0                                         LOR00350
      INTEGER  IPR(3000),IPT(3000)                                      LOR00360
      INTEGER  IROW(10000),INUM(10000),ICOL(3000)                       LOR00370
      INTEGER  KROW(10000),KNUM(10000),KCOL(3000)                       LOR00380
C-----------------------------------------------------------------------LOR00390
C                                                                       LOR00400
C     ARRAYS MUST BE DIMENSIONED AS FOLLOWS:                            LOR00410
C     1.  AD, BD:  >= N, THE ORDER OF C-MATRIX.                         LOR00420
C     2.  ASD: >= NZS, THE NUMBER OF NONZERO SUBDIAGONAL ENTRIES IN C.  LOR00430
C     3.  BSD:   >= NZS, THE NUMBER OF NONZERO SUBDIAGONAL ENTRIES IN   LOR00440
C                   B = P*C*P-TRANSPOSE                                 LOR00450
C     4.  IPR, IPT:   >= N                                              LOR00460
C     5.  ICOL, KCOL:   >= N                                            LOR00470
C     6.  IROW, KROW, INUM, KNUM:   >=  NZ = 2*NZS + N                  LOR00480
C                                                                       LOR00490
C-----------------------------------------------------------------------LOR00500
C     OUTPUT HEADER                                                     LOR00510
      WRITE(6,10)                                                       LOR00520
   10 FORMAT(/' LORDER PROGRAM, COMPUTE B = P*C*(P-TRANSPOSE), STORE ON LOR00530
     1FILE 9'/)                                                         LOR00540
C                                                                       LOR00550
```

```
C     READ NUMBER OF NONZERO SUBDIAGONAL ENTRIES (NZS), ORDER OF MATRIX LOR00560
C     (N),INDEX OF LAST COLUMN CONTAINING NONZERO ENTRIES BELOW THE     LOR00570
C     DIAGONAL (NZL), MATRIX IDENTIFICATION NUMBER (MATNO), PERMUTATION LOR00580
C     FLAG (JPERM).                                                     LOR00590
      READ(8,20) NZS,N,NZL,MATNO,JPERM                                  LOR00600
   20 FORMAT(I10,2I6,I8,I6)                                             LOR00610
C                                                                       LOR00620
      WRITE(6,30) NZS,N,NZL,MATNO,JPERM                                 LOR00630
   30 FORMAT(/I10,2I6,I8,I3,' = NZS,N,NZL,MATNO,JPERM'/)                LOR00640
C                                                                       LOR00650
C     NUMBER OF NONZERO SUBDIAGONAL ENTRIES IN EACH COLUMN IS READ      LOR00660
C     THEN THE CORRESPONDING ROW INDEX FOR EACH SUCH ENTRY IS READ      LOR00670
      READ(8,40) (ICOL(K), K=1,NZL)                                     LOR00680
      READ(8,40) (IROW(K), K=1,NZS)                                     LOR00690
   40 FORMAT(13I6)                                                      LOR00700
C                                                                       LOR00710
      NZL1 = NZL + 1                                                    LOR00720
      DO 50 K = NZL1,N                                                  LOR00730
   50 ICOL(K) = 0                                                       LOR00740
C                                                                       LOR00750
C     DIAGONAL OF C-MATRIX IS READ (INCLUDING ANY ZERO ENTRIES), THEN   LOR00760
C     NONZERO SUBDIAGONAL ENTRIES ARE READ IN                           LOR00770
      READ(8,60) (AD(K), K=1,N)                                         LOR00780
      READ(8,60) (ASD(K), K=1,NZS)                                      LOR00790
   60 FORMAT(4E19.10)                                                   LOR00800
C                                                                       LOR00810
      IF(JPERM.EQ.0) GO TO 390                                          LOR00820
C     READ PERMUTATION                                                  LOR00830
      READ(8,40) (IPR(K), K = 1,N)                                      LOR00840
C                                                                       LOR00850
      DO 70 K = 1,N                                                     LOR00860
      J = IPR(K)                                                        LOR00870
   70 IPT(J) = K                                                        LOR00880
C                                                                       LOR00890
      READ(8,80) S0,SHIFT                                               LOR00900
   80 FORMAT(2E12.5)                                                    LOR00910
C                                                                       LOR00920
      WRITE(6,90)                                                       LOR00930
   90 FORMAT(/' MATRIX HAS BEEN READ IN FROM FILE 8'/                   LOR00940
     1 ' PERMUTATION IPR HAS BEEN READ IN'/)                            LOR00950
C                                                                       LOR00960
C     EXPAND IROW AND ICOL TO INCLUDE DIAGONAL AND SUPER DIAGONAL       LOR00970
      KCOL(1) = 1 + ICOL(1)                                             LOR00980
      KNUM(1) = -1                                                      LOR00990
      KROW(1) = 1                                                       LOR01000
      IF (ICOL(1).EQ.0) GO TO 110                                       LOR01010
      KL = ICOL(1)                                                      LOR01020
      DO 100 K = 1,KL                                                   LOR01030
      KP1 = K+1                                                         LOR01040
      KROW(KP1) = IROW(K)                                               LOR01050
  100 KNUM(KP1) = K                                                     LOR01060
  110 KCOUNT = KCOL(1)                                                  LOR01070
C                                                                       LOR01080
      DO 160 K = 2,N                                                    LOR01090
      K1 = MIN(K-1,NZL)                                                 LOR01100
```

```
      JL = 0                                                            LOR01110
      JCOUNT = 0                                                        LOR01120
      DO 140 J = 1,K1                                                   LOR01130
      IF (ICOL(J).EQ.0) GO TO 140                                       LOR01140
      JF = JL + 1                                                       LOR01150
      JL = JL + ICOL(J)                                                 LOR01160
      DO 130 JJ = JF,JL                                                 LOR01170
      IF (IROW(JJ)-K) 130,120,140                                       LOR01180
  120 KCOUNT = KCOUNT + 1                                               LOR01190
      JCOUNT = JCOUNT + 1                                               LOR01200
      KROW(KCOUNT) = J                                                  LOR01210
      KNUM(KCOUNT) = JJ                                                 LOR01220
      GO TO 140                                                         LOR01230
  130 CONTINUE                                                          LOR01240
  140 CONTINUE                                                          LOR01250
      KCOUNT = KCOUNT + 1                                               LOR01260
      KROW(KCOUNT) = K                                                  LOR01270
      KNUM(KCOUNT) = -K                                                 LOR01280
      ITEMP = 0                                                         LOR01290
      IF (K.LE.NZL) ITEMP = ICOL(K)                                     LOR01300
      KCOL(K) = JCOUNT + 1 + ITEMP                                      LOR01310
      IF (K.GT.NZL.OR.ICOL(K).EQ.0) GO TO 160                           LOR01320
      KF = 1 + KL                                                       LOR01330
      KL = KL + ICOL(K)                                                 LOR01340
      DO 150 J = KF,KL                                                  LOR01350
      KCOUNT = KCOUNT + 1                                               LOR01360
      KROW(KCOUNT) = IROW(J)                                            LOR01370
  150 KNUM(KCOUNT) = J                                                  LOR01380
  160 CONTINUE                                                          LOR01390
C     NTOTAL = N + 2*NZS                                                LOR01400
C     A-MATRIX INDEX LISTS HAVE BEEN EXPANDED                           LOR01410
C                                                                       LOR01420
      WRITE(6,170)                                                      LOR01430
  170 FORMAT(/' EXPANSION OF INDEX LISTS FOR C-MATRIX IS COMPLETED'/)   LOR01440
C                                                                       LOR01450
C     DETERMINE STRUCTURE OF B = P*C*P-TRANSPOSE                        LOR01460
      IL = 0                                                            LOR01470
      KCOUNT = 0                                                        LOR01480
      DO 180 K = 1,N                                                    LOR01490
  180 ICOL(K) = 0                                                       LOR01500
      DO 270 K = 1,N                                                    LOR01510
      J = IPR(K)                                                        LOR01520
      JL = 0                                                            LOR01530
      IF (J.EQ.1) GO TO 200                                             LOR01540
      JM1 = J - 1                                                       LOR01550
      DO 190 JJ = 1,JM1                                                 LOR01560
  190 JL = JL + KCOL(JJ)                                                LOR01570
  200 CONTINUE                                                          LOR01580
      JF = JL + 1                                                       LOR01590
      JL = JL + KCOL(J)                                                 LOR01600
      ICOL(K) = KCOL(J)                                                 LOR01610
      IF = IL + 1                                                       LOR01620
      IL = IL + ICOL(K)                                                 LOR01630
C                                                                       LOR01640
      DO 210 JJ = JF,JL                                                 LOR01650
```

```
      KCOUNT = KCOUNT + 1                                              LOR01660
      JR = KROW(JJ)                                                    LOR01670
      JK = IPT(JR)                                                     LOR01680
      INUM(KCOUNT) = KNUM(JJ)                                          LOR01690
  210 IROW(KCOUNT) = JK                                                LOR01700
C                                                                      LOR01710
C     ORDER IROW VECTOR BY INCREASING SIZE                             LOR01720
      IF (IF.EQ.IL) GO TO 240                                          LOR01730
      IF1 = IF + 1                                                     LOR01740
      DO 230 I = IF1,IL                                                LOR01750
      IM1 = I-1                                                        LOR01760
      IMF = IM1 + IF                                                   LOR01770
      DO 220 L = IF,IM1                                                LOR01780
      II = IMF - L                                                     LOR01790
      IF (IROW(II+1).GE.IROW(II)) GO TO 230                            LOR01800
      I0 = IROW(II)                                                    LOR01810
      IROW(II) = IROW(II+1)                                            LOR01820
      IROW(II+1) = I0                                                  LOR01830
      I0 = INUM(II)                                                    LOR01840
      INUM(II) = INUM(II+1)                                            LOR01850
      INUM(II+1) = I0                                                  LOR01860
  220 CONTINUE                                                         LOR01870
  230 CONTINUE                                                         LOR01880
  240 CONTINUE                                                         LOR01890
C                                                                      LOR01900
      DO 250 I = IF,IL                                                 LOR01910
      IF (INUM(I).LT.0) GO TO 260                                      LOR01920
  250 CONTINUE                                                         LOR01930
  260 INUM(I) = -J                                                     LOR01940
  270 CONTINUE                                                         LOR01950
C                                                                      LOR01960
C     GENERATE SPARSE MATRIX REPRESENTATION OF B-MATRIX                LOR01970
      KCOUNT = 0                                                       LOR01980
      DO 280 K = 1,N                                                   LOR01990
  280 KCOL(K) = 0                                                      LOR02000
      DO 320 K = 1,N                                                   LOR02010
      KL = 0                                                           LOR02020
      DO 290 KK = 1,K                                                  LOR02030
  290 KL = KL + ICOL(KK)                                               LOR02040
      KK = KL+ 1                                                       LOR02050
  300 KK = KK - 1                                                      LOR02060
      IF (INUM(KK).GE.0) GO TO 300                                     LOR02070
      KCOL(K) = KL - KK                                                LOR02080
      J = IPR(K)                                                       LOR02090
      BD(K) = AD(J)                                                    LOR02100
      KF = KK + 1                                                      LOR02110
      IF (KCOL(K).EQ.0) GO TO 320                                      LOR02120
      DO 310 JJ = KF,KL                                                LOR02130
      KCOUNT = KCOUNT + 1                                              LOR02140
      KROW(KCOUNT) = IROW(JJ)                                          LOR02150
      KK = INUM(JJ)                                                    LOR02160
  310 BSD(KCOUNT) = ASD(KK)                                            LOR02170
  320 CONTINUE                                                         LOR02180
      NZL = 0                                                          LOR02190
      DO 330 K = 1,N                                                   LOR02200
```

```
      IF (KCOL(K).NE.0) NZL = K                                          LOR02210
  330 CONTINUE                                                           LOR02220
C     WE NOW HAVE B = P*A*P-TRANSPOSE IN SPARSE MATRIX FORMAT, WRITE TO  LOR02230
C     FILE 9                                                             LOR02240
C                                                                        LOR02250
      JPERM = 1                                                          LOR02260
      WRITE(9,340) NZS,N,NZL,MATNO,JPERM                                 LOR02270
  340 FORMAT(I10,2I6,I8,I6,' = NZS,N,NZL,MATNO,JPERM. BCOMPAC')          LOR02280
C                                                                        LOR02290
C     NUMBER OF NONZERO SUBDIAGONAL ENTRIES IN EACH COLUMN IS WRITTEN    LOR02300
C     THEN THE CORRESPONDING ROW INDEX FOR EACH SUCH ENTRY IS WRITTEN    LOR02310
      WRITE(9,350) (KCOL(K), K=1,NZL)                                    LOR02320
      WRITE(9,350) (KROW(K), K=1,NZS)                                    LOR02330
  350 FORMAT(13I6)                                                       LOR02340
C     DIAGONAL IS WRITTEN FIRST, THEN NONZERO BELOW DIAGONAL ENTRIES     LOR02350
      WRITE(9,360) (BD(K), K=1,N)                                        LOR02360
      WRITE(9,360) (BSD(K), K=1,NZS)                                     LOR02370
  360 FORMAT(4E19.10)                                                    LOR02380
C                                                                        LOR02390
C     WRITE PERMUTATION                                                  LOR02400
      WRITE(9,350) (IPR(K), K=1,N)                                       LOR02410
C                                                                        LOR02420
      WRITE(9,370) S0,SHIFT                                              LOR02430
  370 FORMAT(2E12.5,' = S0 SHIFT'/                                       LOR02440
     1 ' ABOVE IS REORDERED MATRIX, B'/                                  LOR02450
     1 ' INPUT MATRIX SUPPLIED WAS C = S0*A + SHIFT*I'/                  LOR02460
     1 ' B = P*C*(P-TRANSPOSE),  B IS STORED IN SPARSE MATRIX FORMAT'/   LOR02470
     1 ' ROW(COL) I OF B CORRESPONDS TO ROW(COL) J OF C, J = IPR(I)'/    LOR02480
     1 ' NZS = TOTAL NUMBER OF SUBDIAGONAL NONZEROS IN B-MATRIX'/        LOR02490
     1 ' KCOL(K) = NUMBER OF SUBDIAGONAL NONZEROS IN COL K OF B'/        LOR02500
     1 ' KROW(K) = ROW INDEX OF SUBDIAGONAL NONZERO'/                    LOR02510
     1 ' SUBDIAGONAL NONZEROS IN B ARE STORED COLUMN BY COLUMN'/         LOR02520
     1 ' BD(K) = THE KTH DIAGONAL ELEMENT OF B'/                         LOR02530
     1 ' BSD(K) = NUMERICAL VALUE OF KTH SUBDIAGONAL NONZERO IN B'/      LOR02540
     1 ' IPR(K) = J MEANS THAT ROW J OF C CORRESPONDS TO ROW K OF B')    LOR02550
C                                                                        LOR02560
      WRITE(6,380)                                                       LOR02570
  380 FORMAT(' SPARSE FORMAT FOR B-MATRIX HAS BEEN WRITTEN TO FILE 9') LOR02580
      GO TO 410                                                          LOR02590
C                                                                        LOR02600
  390 WRITE(6,400)                                                       LOR02610
  400 FORMAT(/' LORDER PROGRAM TERMINATES BECAUSE MATRIX FILE SUPPLIED DLOR02620
     1ID NOT'/' CONTAIN A PERMUTATION')                                  LOR02630
C                                                                        LOR02640
  410 CONTINUE                                                           LOR02650
C                                                                        LOR02660
C-----END OF LORDER---------------------------------------------------- LOR02670
      STOP                                                               LOR02680

      END                                                                LOR02690
```

```
C-----LFACT------------------------------------------------------------  LFA00010
C                                                                       LFA00020
C     NONPORTABLE CONSTRUCTIONS:                                        LFA00030
C     1.  FORMAT (4Z20).  TO AVOID COMPOUNDING FORMAT CONVERSION        LFA00040
C         ERRORS, THE MATRIX ENTRIES SHOULD BE IN MACHINE FORMAT,       LFA00050
C         (4Z20) FOR IBM/3081.                                          LFA00060
C                                                                       LFA00070
C     LFACT COMPUTES THE CHOLESKY FACTOR L FOR THE MATRIX B AND STORES  LFA00080
C     THIS FACTOR ON FILE 7.  B MUST BE A POSITIVE DEFINITE MATRIX.     LFA00090
C     THE PERMUTATION P (IN IPR), THE SCALE S0 AND THE SHIFT (IF ANY)   LFA00100
C     USED TO OBTAIN B FROM THE ORIGINAL MATRIX A ARE STORED AT THE END LFA00110
C     OF FILE 7.   THAT IS,  B = S0*P*A*P' + SHIFT*I.  THE PROGRAM      LFA00120
C     ASSUMES THAT THE DATA READ FROM FILE 9 IS FOR THE B-MATRIX.       LFA00130
C                                                                       LFA00140
C-----------------------------------------------------------------------LFA00150
C                                                                       LFA00160
C     ARRAYS MUST BE DIMENSIONED AS FOLLOWS:                            LFA00170
C     1.  AD:  >= N, THE ORDER OF A-MATRIX.                             LFA00180
C     3.  ASD:   >= NZT, THE NUMBER OF NONZERO SUBDIAGONAL ENTRIES      LFA00190
C                IN THE CHOLESKY FACTOR OF B.                           LFA00200
C     4.  ICOL,IPR:   >= N                                              LFA00210
C     5.  IROW:   >=  NZT                                               LFA00220
C                                                                       LFA00230
C-----------------------------------------------------------------------LFA00240
      DOUBLE PRECISION ASD(10000),AD(3000)                              LFA00250
      DOUBLE PRECISION ZERO,ONE,TEMP,S0,SHIFT                           LFA00260
      INTEGER IROW(10000),ICOL(3000),IPR(3000)                          LFA00270
      DOUBLE PRECISION DSQRT                                            LFA00280
C-----------------------------------------------------------------------LFA00290
C     OUTPUT HEADER                                                     LFA00300
      WRITE(6,5)                                                        LFA00310
    5 FORMAT(/' LFACT PROGRAM, COMPUTE CHOLESKY FACTOR FOR POSITIVED DEFLFA00320
     1INITE B-MATRIX'/' AND STORE THE FACTOR ON FILE 7'/)               LFA00330
C                                                                       LFA00340
C     SET PROGRAM PARAMETERS                                            LFA00350
      ONE = 1.0D0                                                       LFA00360
      ZERO = 0.0D0                                                      LFA00370
C                                                                       LFA00380
C     READ NUMBER OF NONZERO BELOW DIAGONAL ENTRIES, ORDER OF MATRIX,   LFA00390
C     INDEX OF LAST COLUMN CONTAINING NONZERO ENTRIES BELOW THE         LFA00400
C     DIAGONAL, MATRIX IDENTIFICATION NUMBER                            LFA00410
      READ(9,15) NZS,N,NZL,MATNO,JPERM                                  LFA00420
   15 FORMAT(I10,2I6,I8,I6)                                             LFA00430
C                                                                       LFA00440
      WRITE(6,20) NZS,N,NZL,JPERM,MATNO                                 LFA00450
   20 FORMAT(I10,3I6,I8,' = NZS,N,NZL,JPERM,MATNO'/)                    LFA00460
C                                                                       LFA00470
C     NUMBER OF NONZERO SUBDIAGONAL ENTRIES IN EACH COLUMN IS READ      LFA00480
C     THEN THE CORRESPONDING ROW INDEX FOR EACH SUCH ENTRY IS READ      LFA00490
      READ(9,30) (ICOL(K), K=1,NZL)                                     LFA00500
      READ(9,30) (IROW(K), K=1,NZS)                                     LFA00510
   30 FORMAT(13I6)                                                      LFA00520
C                                                                       LFA00530
C                                                                       LFA00540
      NZL1 = NZL + 1                                                    LFA00550
```

```
      DO 40 K = NZL1,N                                                  LFA00560
   40 ICOL(K) = 0                                                       LFA00570
C                                                                       LFA00580
C     DIAGONAL IS READ (INCLUDING ANY ZERO ENTRIES), THEN NONZERO       LFA00590
C     BELOW DIAGONAL ENTRIES ARE READ IN                                LFA00600
      READ(9,50) (AD(K), K=1,N)                                         LFA00610
      READ(9,50) (ASD(K), K=1,NZS)                                      LFA00620
   50 FORMAT(4E19.10)                                                   LFA00630
C  50 FORMAT(4Z20)                                                      LFA00640
C                                                                       LFA00650
      IF (JPERM.NE.0) READ(9,30) (IPR(K), K = 1,N)                      LFA00660
C                                                                       LFA00670
      READ(9,55) S0,SHIFT                                               LFA00680
   55 FORMAT(2E12.5)                                                    LFA00690
C                                                                       LFA00700
      WRITE(6,60)                                                       LFA00710
   60 FORMAT(/' B-MATRIX HAS BEEN READ IN FROM FILE 9'/)                LFA00720
C                                                                       LFA00730
      IF (JPERM.NE.0) WRITE(6,65)                                       LFA00740
   65 FORMAT( ' PERMUTATION IPR HAS BEEN READ IN'/)                     LFA00750
C                                                                       LFA00760
C     CALCULATE CHOLESKY FACTOR,  B = BL*(BL-TRANSPOSE)                 LFA00770
      NZT = NZS                                                         LFA00780
      NZL = N-1                                                         LFA00790
      KL = 0                                                            LFA00800
      DO 70 K = 1,N                                                     LFA00810
C     CALCULATE KTH PIVOT FOR BL                                        LFA00820
      TEMP = AD(K)                                                      LFA00830
C                                                                       LFA00840
      IF (AD(K).GT.ZERO) GO TO 80                                       LFA00850
C                                                                       LFA00860
      WRITE(6,90) K,AD(K)                                               LFA00870
   90 FORMAT(/I6,E15.8,' = K,AD(K)'/                                    LFA00880
     1' PIVOT IS NEGATIVE SO B-MATRIX IS NOT POSITIVE DEFINITE'/        LFA00890
     1' THEREFORE COMPUTATION OF CHOLESKY FACTOR TERMINATES'/)          LFA00900
      GO TO 240                                                         LFA00910
C                                                                       LFA00920
   80 CONTINUE                                                          LFA00930
      TEMP = DSQRT(TEMP)                                                LFA00940
      AD(K) = TEMP                                                      LFA00950
      TEMP = ONE/TEMP                                                   LFA00960
      IF(K.EQ.N.OR.ICOL(K).EQ.0) GO TO 70                               LFA00970
      KF = KL + 1                                                       LFA00980
      KL = KL + ICOL(K)                                                 LFA00990
      DO 100 KK = KF,KL                                                 LFA01000
      KR = IROW(KK)                                                     LFA01010
      ASD(KK) = TEMP*ASD(KK)                                            LFA01020
  100 AD(KR) = AD(KR) - ASD(KK)**2                                      LFA01030
      IF (KF.EQ.KL) GO TO 70                                            LFA01040
      K1 = K+1                                                          LFA01050
      DO 110 KK = KF,KL                                                 LFA01060
      KR = IROW(KK)                                                     LFA01070
      IF (KK.EQ.KL) GO TO 110                                           LFA01080
      KE = KL                                                           LFA01090
      DO 120 KC = K1,KR                                                 LFA01100
```

```
  120 KE= KE + ICOL(KC)                                                 LFA01110
      KB = KE - ICOL(KR) + 1                                            LFA01120
      KK1 = KK + 1                                                      LFA01130
      L = KB                                                            LFA01140
      DO 130 LL = KK1,KL                                                LFA01150
      LR = IROW(LL)                                                     LFA01160
      IF (ICOL(KR).EQ.0.OR.L.GT.KE) GO TO 140                           LFA01170
  150 LC = IROW(L)                                                      LFA01180
      IF (LC - LR) 160,170,140                                          LFA01190
  160 L = L + 1                                                         LFA01200
      IF (L.LE.KE) GO TO 150                                            LFA01210
C     NEW NONZERO IN CHOLESKY FACTOR L                                  LFA01220
  140 NZT = NZT + 1                                                     LFA01230
      L1 = L + 1                                                        LFA01240
      NT = NZT + L1                                                     LFA01250
      DO 180 KM = L1,NZT                                                LFA01260
      MK = NT - KM                                                      LFA01270
      ASD(MK) = ASD(MK-1)                                               LFA01280
  180 IROW(MK) = IROW(MK-1)                                             LFA01290
      ICOL(KR) = ICOL(KR) + 1                                           LFA01300
      KE = KE + 1                                                       LFA01310
      ASD(L) = -ASD(KK)*ASD(LL)                                         LFA01320
      IROW(L) = LR                                                      LFA01330
      GO TO 130                                                         LFA01340
C     UPDATE EXISTING ELEMENT                                           LFA01350
  170 ASD(L) = ASD(L) - ASD(KK)*ASD(LL)                                 LFA01360
  130 L = L + 1                                                         LFA01370
  110 CONTINUE                                                          LFA01380
   70 CONTINUE                                                          LFA01390
C                                                                       LFA01400
C                                                                       LFA01410
C     FACTOR L HAS BEEN COMPUTED, STORE IN SPARSE FORMAT ON FILE 7      LFA01420
C                                                                       LFA01430
      WRITE(7,190) NZT,N,NZL,MATNO,JPERM                                LFA01440
  190 FORMAT(I10,2I6,I8,I6,' = NZT,N,NZL,MATNO,JPERM. LCOMPAC')         LFA01450
C                                                                       LFA01460
C     NUMBER OF NONZERO SUBDIAGONAL ENTRIES IN EACH COLUMN IS WRITTEN   LFA01470
C     THEN THE CORRESPONDING ROW INDEX FOR EACH SUCH ENTRY IS WRITTEN   LFA01480
      WRITE(7,200) (ICOL(K), K=1,NZL)                                   LFA01490
      WRITE(7,200) (IROW(K), K=1,NZT)                                   LFA01500
  200 FORMAT(13I6)                                                      LFA01510
C     DIAGONAL IS WRITTEN FIRST, THEN NONZERO BELOW DIAGONAL ENTRIES    LFA01520
      WRITE(7,210) (AD(K), K=1,N)                                       LFA01530
      WRITE(7,210) (ASD(K), K=1,NZT)                                    LFA01540
  210 FORMAT(4Z20)                                                      LFA01550
C 210 FORMAT(3E25.16)                                                   LFA01560
      IF (JPERM.NE.0) WRITE(7,200) (IPR(K), K=1,N)                      LFA01570
C                                                                       LFA01580
      WRITE(7,220) S0,SHIFT                                             LFA01590
  220 FORMAT(2E12.5,' = S0 SHIFT'/                                      LFA01600
     1 ' ABOVE IS CHOLESKY FACTOR FOR B-MATRIX'/                        LFA01610
     1 ' IF JPERM = 0, THEN P = I.  C = S0*A * SHIFT*I'/                LFA01620
     1 ' B = P*C*P-TRANS = L*L-TRANS,  L IS STORED IN SPARSE FORMAT'/   LFA01630
     1 ' ROW(COL) I OF B CORRESPONDS TO ROW(COL) J OF C, J = IPR(I)'/   LFA01640
     1 ' NZT = TOTAL NUMBER OF SUBDIAGONAL NONZEROS IN L'/              LFA01650
```

```
     1 ' ICOL(K) = NUMBER OF SUBDIAGONAL NONZEROS IN COL K OF L'/       LFA01660
     1 ' IROW(K) = ROW INDEX OF SUBDIAGONAL NONZERO'/                   LFA01670
     1 ' SUBDIAGONAL NONZEROS IN L ARE STORED COLUMN BY COLUMN'/        LFA01680
     1 ' AD(K) = KTH DIAGONAL ELEMENT OF L'/                           LFA01690
     1 ' ASD(K) = KTH SUBDIAGONAL NONZERO IN L'/)                      LFA01700
C                                                                      LFA01710
      WRITE(6,230)                                                     LFA01720
  230 FORMAT(' CHOLESKY FACTOR HAS BEEN WRITTEN TO FILE 7 ')           LFA01730
C                                                                      LFA01740
  240 CONTINUE                                                         LFA01750
C                                                                      LFA01760
C-----END OF LFACT---------------------------------------------------  LFA01770
      STOP                                                             LFA01780
      END                                                              LFA01790
```

```
C-----LTEST------------------------------------------------------------ LTE00010
C                                                                       LTE00020
C     CONTAINS MAIN PROGRAM LTEST AND SAMPLE CMATS, CMATV, BSOLV        LTE00030
C     LTEST ALSO REQUIRES A RANDOM NUMBER GENERATOR.                    LTE00040
C                                                                       LTE00050
C     LTEST GIVES A ROUGH CHECK ON THE CONDITION OF A MATRIX B BY       LTE00060
C     SOLVING B*X = B*V1 FOR X WHERE V1 IS A KNOWN, RANDOMLY-GENERATED  LTE00070
C     VECTOR.  SOLVING IS DONE, WITH AND WITHOUT ITERATIVE REFINEMENT.  LTE00080
C     IN BOTH CASES, X IS COMPARED WITH V1 AND THE ERRORS ARE           LTE00090
C     WRITTEN TO FILE 6.                                                LTE00100
C                                                                       LTE00110
C     VECTORS V0, V1, V2, VS, AND G ARE USED IN THE COMPUTATIONS.       LTE00120
C     NOTE THAT THE SUBROUTINE CMATS USED TO COMPUTE THE RESIDUAL       LTE00130
C     IN EXTENDED PRECISION FOR THE ITERATIVE REFINEMENT CALCULATION    LTE00140
C     REQUIRES AN EXTRA LONG V0 VECTOR OF LENGTH TWICE THE SIZE OF B.   LTE00150
C                                                                       LTE00160
C     NONPORTABLE CONSTRUCTIONS:                                        LTE00170
C     1.  THE ENTRY MECHANISM WHICH PASSES THE STORAGE LOCATIONS OF     LTE00180
C         ARRAYS AND PARAMETERS THAT DEFINE THE B-MATRIX TO THE         LTE00190
C         SUBROUTINES CMATV, CMATS, AND BSOLV.                          LTE00200
C     2.  FORMATS (20A4) AND (4Z20).  TO AVOID COMPOUNDING FORMAT       LTE00210
C         CONVERSION ERRORS, MATRIX ENTRIES SHOULD BE STORED IN         LTE00220
C         MACHINE FORMAT, ((4Z20) FOR IBM/3081). ALSO FREE FORMAT       LTE00230
C         (5,*).                                                        LTE00240
C     3.  REAL*16 VARIABLES IN CMATS SUBROUTINE.                        LTE00250
C                                                                       LTE00260
C                                                                       LTE00270
C-----------------------------------------------------------------------LTE00280
      DOUBLE PRECISION ASD(10000),AD(3000),BSD(20000),BD(3000)          LTE00290
      DOUBLE PRECISION V0(6000),V1(3000),V2(3000),VS(3000)              LTE00300
      DOUBLE PRECISION ZERO,ONE,TEMP,SUM                                LTE00310
      DOUBLE PRECISION ERROR0,ERROR1,ENORM0,ENORM1                      LTE00320
      REAL EXPLAN(20),G(3000)                                           LTE00330
      INTEGER  IROW(20000),ICOL(3000),KROW(30000),KCOL(3000),SVSEED     LTE00340
      DOUBLE PRECISION FINPRO                                           LTE00350
      DOUBLE PRECISION DABS, DMAX1, DSQRT                               LTE00360
C-----------------------------------------------------------------------LTE00370
C                                                                       LTE00380
C     ARRAYS MUST BE DIMENSIONED AS FOLLOWS:                            LTE00390
C     1.  AD, BD:  >= N, THE ORDER OF A-MATRIX.                         LTE00400
C     2.  ASD:   >= NZS, THE NUMBER OF NONZERO SUBDIAGONAL ENTRIES IN B.LTE00410
C     3.  BSD:   >= NZT, THE NUMBER OF NONZERO SUBDIAGONAL ENTRIES      LTE00420
C                IN THE CHOLESKY FACTOR OF B.                           LTE00430
C     5.  ICOL, KCOL:   >= N                                            LTE00440
C     6.  KROW:   >=  NZS                                               LTE00450
C     7.  IROW:   >=  NZT                                               LTE00460
C     8.  V1,V2,VS:   >=   N                                            LTE00470
C     9.  V0:   >=   2*N                                                LTE00480
C                                                                       LTE00490
C-----------------------------------------------------------------------LTE00500
C     OUTPUT HEADER                                                     LTE00510
      WRITE(6,10)                                                       LTE00520
   10 FORMAT(/' LTEST PROGRAM, ROUGH CHECK ON NUMERICAL CONDITION OF GIVLTE00530
     1EN MATRIX'/)                                                      LTE00540
C                                                                       LTE00550
```

```
C     SET PROGRAM PARAMETERS                                          LTE00560
      ONE = 1.0D0                                                     LTE00570
      ZERO = 0.0D0                                                    LTE00580
C                                                                     LTE00590
C     READ INPUT HEADER                                               LTE00600
      READ(5,20) EXPLAN                                               LTE00610
      WRITE(6,20) EXPLAN                                              LTE00620
   20 FORMAT(20A4)                                                    LTE00630
C                                                                     LTE00640
C     READ IN IN FREE FORMAT USER-SPECIFIED PARAMETERS FROM FILE 5    LTE00650
      READ(5,20) EXPLAN                                               LTE00660
      READ(5,*) SVSEED                                                LTE00670
C                                                                     LTE00680
C     READ NUMBER OF NONZERO BELOW DIAGONAL ENTRIES, ORDER OF MATRIX, LTE00690
C     INDEX OF LAST COLUMN CONTAINING NONZERO ENTRIES BELOW THE        LTE00700
C     DIAGONAL, MATRIX IDENTIFICATION NUMBER                          LTE00710
      READ(9,30) NZS,N,NZL,MATNO,JPERM                                LTE00720
   30 FORMAT(I10,2I6,I8,I6)                                           LTE00730
C                                                                     LTE00740
      WRITE(6,40) NZS,N,NZL,JPERM,MATNO,SVSEED                        LTE00750
   40 FORMAT(I10,3I6,' = NZS,N,NZL,JPERM'/                            LTE00760
     1 I8,I12,' = MATNO,SVSEED'/)                                     LTE00770
C                                                                     LTE00780
C     NUMBER OF NONZERO SUBDIAGONAL ENTRIES IN EACH COLUMN IS READ    LTE00790
C     THEN THE CORRESPONDING ROW INDEX FOR EACH SUCH ENTRY IS READ    LTE00800
      READ(9,50) (KCOL(K), K=1,NZL)                                   LTE00810
      READ(9,50) (KROW(K), K=1,NZS)                                   LTE00820
   50 FORMAT(13I6)                                                    LTE00830
C                                                                     LTE00840
C                                                                     LTE00850
      NZL1 = NZL + 1                                                  LTE00860
      DO 60 K = NZL1,N                                                LTE00870
   60 KCOL(K) = 0                                                     LTE00880
C                                                                     LTE00890
C     DIAGONAL IS READ (INCLUDING ANY ZERO ENTRIES), THEN NONZERO     LTE00900
C     BELOW DIAGONAL ENTRIES ARE READ IN                              LTE00910
      READ(9,70) (AD(K), K=1,N)                                       LTE00920
      READ(9,70) (ASD(K), K=1,NZS)                                    LTE00930
   70 FORMAT(4E19.10)                                                 LTE00940
C                                                                     LTE00950
      WRITE(6,80)                                                     LTE00960
   80 FORMAT(/' B-MATRIX HAS BEEN READ IN FROM FILE 9'/)              LTE00970
C                                                                     LTE00980
C-----------------------------------------------------------------------LTE00990
C     ENTRIES TO CMATS AND CMATV SUBROUTINES                          LTE01000
      CALL CMATSE(ASD,AD,KCOL,KROW,N,NZL)                             LTE01010
      CALL CMATVE(ASD,AD,KCOL,KROW,N,NZL)                             LTE01020
C-----------------------------------------------------------------------LTE01030
C                                                                     LTE01040
C     READ CHOLESKY FACTOR FROM FILE 7                                LTE01050
C                                                                     LTE01060
      READ(7,90) NZT,N,NZL,MATNO,JPERM                                LTE01070
   90 FORMAT(I10,2I6,I8,I6)                                           LTE01080
C                                                                     LTE01090
C     NUMBER OF NONZERO SUBDIAGONAL ENTRIES IN EACH COLUMN IS READ    LTE01100
```

```
C     THEN THE CORRESPONDING ROW INDEX FOR EACH SUCH ENTRY IS READ      LTE01110
      READ(7,100) (ICOL(K), K=1,NZL)                                    LTE01120
      READ(7,100) (IROW(K), K=1,NZT)                                    LTE01130
  100 FORMAT(13I6)                                                      LTE01140
C     DIAGONAL IS READ FIRST, THEN NONZERO BELOW DIAGONAL ENTRIES       LTE01150
      READ(7,110) (BD(K), K=1,N)                                        LTE01160
      READ(7,110) (BSD(K), K=1,NZT)                                     LTE01170
  110 FORMAT(4Z20)                                                      LTE01180
C  90 FORMAT(3E25.16)                                                   LTE01190
C                                                                       LTE01200
C-----------------------------------------------------------------------LTE01210
C     ENTRY TO BSOLV SUBROUTINE, PASS FACTOR OF B                       LTE01220
      CALL BSOLVE(BSD,BD,ICOL,IROW,N,NZT,NZL)                           LTE01230
C-----------------------------------------------------------------------LTE01240
C                                                                       LTE01250
C     SOLVE B*X = B*V1 WITH AND WITHOUT ITERATIVE REFINEMENT, COMPARE   LTE01260
C     ERRORS IN SOLVING AS A ROUGH CHECK ON THE CONDITION OF THE        LTE01270
C     MATRIX B.                                                         LTE01280
C                                                                       LTE01290
      IIX = SVSEED                                                      LTE01300
C                                                                       LTE01310
C-----------------------------------------------------------------------LTE01320
C     COMPUTES RANDOM VECTOR FOR USE IN RIGHT-HAND SIDE                 LTE01330
      CALL GENRAN(IIX,G,N)                                              LTE01340
C-----------------------------------------------------------------------LTE01350
C                                                                       LTE01360
      DO 120 K = 1,N                                                    LTE01370
  120 V1(K) = G(K)                                                      LTE01380
C                                                                       LTE01390
C-----------------------------------------------------------------------LTE01400
      SUM = FINPRO(N,V1(1),1,V1(1),1)                                   LTE01410
C-----------------------------------------------------------------------LTE01420
      SUM = ONE/DSQRT(SUM)                                              LTE01430
C                                                                       LTE01440
      DO 130 K = 1,N                                                    LTE01450
  130 V1(K) = V1(K)*SUM                                                 LTE01460
C                                                                       LTE01470
      SUM = ZERO                                                        LTE01480
C                                                                       LTE01490
C-----------------------------------------------------------------------LTE01500
C     COMPUTE V2 = RHS = B*V1  C = S0*A + SHIFT*I  B = P*C*P'           LTE01510
C     VS = B(INVERSE)*V2                                                LTE01520
      CALL CMATV(V1,V2,SUM)                                             LTE01530
      CALL BSOLV(VS,V2)                                                 LTE01540
C-----------------------------------------------------------------------LTE01550
C                                                                       LTE01560
      SUM = ZERO                                                        LTE01570
      ERROR0 = ZERO                                                     LTE01580
      DO 140 K = 1,N                                                    LTE01590
      TEMP = DABS(V1(K) - VS(K))                                        LTE01600
      SUM = SUM + TEMP*TEMP                                             LTE01610
  140 ERROR0 = DMAX1(ERROR0,TEMP)                                       LTE01620
      ENORM0 = DSQRT(SUM)                                               LTE01630
C                                                                       LTE01640
      WRITE(6,150) ENORM0,ERROR0                                        LTE01650
```

```
  150 FORMAT(6X,'ENORM0',6X,'ERROR0'/2E12.4/                          LTE01660
     1 ' ENORM0  = NORM (V1 - VS),     VS = BI*(B*V1)'/               LTE01670
     1 ' ERROR0 = MAX DABS(V1(K) - VS(K)),  K = 1,N'/)                LTE01680
C                                                                     LTE01690
      SUM = ONE                                                       LTE01700
C                                                                     LTE01710
C-----------------------------------------------------------------    LTE01720
C     CALCULATE RESIDUAL IN EXTENDED PRECISION V2 = B*VS - V2         LTE01730
C     THEN DO ITERATIVE REFINEMENT                                    LTE01740
      CALL CMATS(VS,V2,V0,SUM)                                        LTE01750
      CALL BSOLV(V2,V2)                                               LTE01760
C-----------------------------------------------------------------    LTE01770
C                                                                     LTE01780
      DO 160 K = 1,N                                                  LTE01790
  160 VS(K) = VS(K) - V2(K)                                           LTE01800
C                                                                     LTE01810
      SUM = ZERO                                                      LTE01820
      ERROR1 = ZERO                                                   LTE01830
      DO 170 K = 1,N                                                  LTE01840
      TEMP = DABS(V1(K) - VS(K))                                      LTE01850
      SUM = SUM + TEMP*TEMP                                           LTE01860
  170 ERROR1 = DMAX1(ERROR1,TEMP)                                     LTE01870
      ENORM1 = DSQRT(SUM)                                             LTE01880
C                                                                     LTE01890
      WRITE(6,180) ENORM1,ERROR1                                      LTE01900
  180 FORMAT(6X,'ENORM1',6X,'ERROR1'/2E12.4/                          LTE01910
     1 ' ERROR AFTER ITERATIVE REFINEMENT'/                           LTE01920
     1 ' ENORM1  = NORM (V1 - VS),     VS = BI*(B*V1)'/               LTE01930
     1 ' ERROR1 = MAX DABS(V1(K) - VS(K)),  K = 1,N'/)                LTE01940
C                                                                     LTE01950
      STOP                                                            LTE01960
C-----END OF LTEST-----------------------------------------------     LTE01970
      END                                                             LTE01980
C                                                                     LTE01990
C----CMATS-------------------------------------------------------------LTE02000
C                                                                     LTE02010
C     REAL, SYMMETRIC, SPARSE MATRIX-VECTOR MULTIPLY USING EXTENDED   LTE02020
C     PRECISION.  CALCULATES U = B*W - SUM*U FOR USE IN ITERATIVE      LTE02030
C     REFINEMENT.  MATRIX B STORED IN SPARSE FORMAT.                  LTE02040
C                                                                     LTE02050
      SUBROUTINE CMATS(W,U,Z,SUM)                                     LTE02060
C                                                                     LTE02070
C-----------------------------------------------------------------------LTE02080
      DOUBLE PRECISION  U(1),W(1),BSD(1),BD(1),SUM                    LTE02090
      REAL*16 Z(1),T0,T1,T2,S0                                        LTE02100
      INTEGER  IROW(1),ICOL(1)                                        LTE02110
C-----------------------------------------------------------------------LTE02120
      S0 = SUM                                                        LTE02130
C                                                                     LTE02140
      DO 10 I = 1,N                                                   LTE02150
      T0 = BD(I)                                                      LTE02160
      T1 = W(I)                                                       LTE02170
      T2 = U(I)                                                       LTE02180
   10 Z(I) = T0*T1-S0*T2                                              LTE02190
C                                                                     LTE02200
```

```
      LLAST = 0                                                         LTE02210
C                                                                       LTE02220
      DO 30 J = 1,NZL                                                   LTE02230
C                                                                       LTE02240
      IF (ICOL(J).EQ.0) GO TO 30                                        LTE02250
C                                                                       LTE02260
      LFIRST = LLAST + 1                                                LTE02270
      LLAST = LLAST + ICOL(J)                                           LTE02280
C                                                                       LTE02290
      DO 20 L = LFIRST,LLAST                                            LTE02300
      I = IROW(L)                                                       LTE02310
      T0 = BSD(L)                                                       LTE02320
      T1 = W(J)                                                         LTE02330
      T2 = W(I)                                                         LTE02340
C                                                                       LTE02350
      Z(I) = Z(I) + T0*T1                                               LTE02360
      Z(J) = Z(J) + T0*T2                                               LTE02370
C                                                                       LTE02380
   20 CONTINUE                                                          LTE02390
C                                                                       LTE02400
   30 CONTINUE                                                          LTE02410
C                                                                       LTE02420
      DO 40 I =1,N                                                      LTE02430
   40 U(I) = Z(I)                                                       LTE02440
C                                                                       LTE02450
      RETURN                                                            LTE02460
C                                                                       LTE02470
C-----------------------------------------------------------------------LTE02480
      ENTRY CMATSE(BSD,BD,ICOL,IROW,N,NZL)                              LTE02490
C-----------------------------------------------------------------------LTE02500
C                                                                       LTE02510
      RETURN                                                            LTE02520
C-----END OF CMATS----------------------------------------------------LTE02530
      END                                                               LTE02540
C                                                                       LTE02550
C-----CMATV-----------------------------------------------------------LTE02560
C                                                                       LTE02570
C     SYMMETRIC, SPARSE MATRIX-VECTOR MULTIPLY, B MATRIX STORED         LTE02580
C     IN SPARSE FORMAT.  CMATV CALCULATES U = B*W - SUM*U               LTE02590
C                                                                       LTE02600
      SUBROUTINE CMATV(W,U,SUM)                                         LTE02610
C                                                                       LTE02620
C-----------------------------------------------------------------------LTE02630
      DOUBLE PRECISION  U(1),W(1),BSD(1),BD(1),SUM                      LTE02640
      INTEGER  KROW(1),KCOL(1)                                          LTE02650
C-----------------------------------------------------------------------LTE02660
C                                                                       LTE02670
      DO 10 I = 1,N                                                     LTE02680
   10 U(I) = BD(I)*W(I) - SUM*U(I)                                      LTE02690
C                                                                       LTE02700
      LLAST = 0                                                         LTE02710
C                                                                       LTE02720
      DO 30 J = 1,NZL                                                   LTE02730
C                                                                       LTE02740
      IF (KCOL(J).EQ.0) GO TO 30                                        LTE02750
```

```
C                                                                       LTE02760
      LFIRST = LLAST + 1                                                LTE02770
      LLAST = LLAST + KCOL(J)                                           LTE02780
C                                                                       LTE02790
      DO 20 L = LFIRST,LLAST                                            LTE02800
      I = KROW(L)                                                       LTE02810
C                                                                       LTE02820
      U(I) = U(I) + BSD(L)*W(J)                                         LTE02830
      U(J) = U(J) + BSD(L)*W(I)                                         LTE02840
C                                                                       LTE02850
   20 CONTINUE                                                          LTE02860
C                                                                       LTE02870
   30 CONTINUE                                                          LTE02880
C                                                                       LTE02890
      RETURN                                                            LTE02900
C                                                                       LTE02910
C-----------------------------------------------------------------------LTE02920
      ENTRY CMATVE(BSD,BD,KCOL,KROW,N,NZL)                              LTE02930
C-----------------------------------------------------------------------LTE02940
C                                                                       LTE02950
      RETURN                                                            LTE02960
C-----END OF CMATV------------------------------------------------------LTE02970
      END                                                               LTE02980
C                                                                       LTE02990
C-----BSOLV-------------------------------------------------------------  LTE03000
C                                                                       LTE03010
C     SOLVES B*U = V WHERE  B = L*L'.                                   LTE03020
C     FIRST SOLVES L*U = V FOR U, THEN SOLVES L'*U = U FOR U            LTE03030
C                                                                       LTE03040
      SUBROUTINE BSOLV(U,V)                                             LTE03050
C                                                                       LTE03060
C-----------------------------------------------------------------------LTE03070
      DOUBLE PRECISION   AD(1),ASD(1),U(1),V(1),TEMP                    LTE03080
      INTEGER  ICOL(1),IROW(1)                                          LTE03090
C-----------------------------------------------------------------------LTE03100
      KL = 0                                                            LTE03110
      DO 10 K = 1,N                                                     LTE03120
   10 U(K) = V(K)                                                       LTE03130
      DO 30 K = 1,N                                                     LTE03140
      TEMP = U(K)/AD(K)                                                 LTE03150
      U(K) = TEMP                                                       LTE03160
      IF (ICOL(K).EQ.0.OR.K.EQ.N) GO TO 30                              LTE03170
      KF = KL + 1                                                       LTE03180
      KL = KL + ICOL(K)                                                 LTE03190
      DO 20 KK = KF,KL                                                  LTE03200
      KR = IROW(KK)                                                     LTE03210
   20 U(KR) = U(KR) - TEMP*ASD(KK)                                      LTE03220
   30 CONTINUE                                                          LTE03230
C                                                                       LTE03240
      NP1 = N+1                                                         LTE03250
      KF = NZT + 1                                                      LTE03260
      DO 50 K = 1,N                                                     LTE03270
      L = NP1 - K                                                       LTE03280
      TEMP = U(L)                                                       LTE03290
      IF (ICOL(L).EQ.0.OR.L.EQ.N) GO TO 50                              LTE03300
```

```
      KL = KF - 1                                                       LTE03310
      KF = KF - ICOL(L)                                                 LTE03320
      DO 40 LL = KF,KL                                                  LTE03330
      LR = IROW(LL)                                                     LTE03340
   40 TEMP = TEMP - ASD(LL)*U(LR)                                       LTE03350
   50 U(L) = TEMP/AD(L)                                                 LTE03360
C                                                                       LTE03370
      RETURN                                                            LTE03380
C                                                                       LTE03390
C-----------------------------------------------------------------------LTE03400
      ENTRY BSOLVE(ASD,AD,ICOL,IROW,N,NZT,NZL)                          LTE03410
C-----------------------------------------------------------------------LTE03420
C                                                                       LTE03430
C-----END OF BSOLV------------------------------------------------------LTE03440
      RETURN                                                            LTE03450
      END                                                               LTE03460
```

4.6 LIVAL: LIVEC: File Definitions, Sample Input Files

Below is a listing of the input/output files which are accessed by the real symmetric Lanczos eigenvalue program, LIVAL. Included also is a sample of the input file which LIVAL requires on file 5. The parameters in this file are supplied in free format. LIVAL computes eigenvalues of real symmetric matrices B^{-1} on user-specified intervals where $B = PCP^T$ with $C = (SCALE) * A + (SHIFT) * I$ where $SCALE$ and $SHIFT$ are scalars. The sample codes assume that C is positive definite and has a reasonable condition number. The permutation matrix P Is used to preserve the sparseness of the given matrix in the Cholesky factorization, $B = LL^T$. The user could replace the BSOLVE subroutine provided here by another more general factorization subroutine.

```
Sample Specification of the Input/Output Files for LIVAL
-----------------------------------------------------------------
 LIVAL EXEC LANCZOS EIGENVALUE CALCULATION USING FACTORIZATION
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 LIVAL     INPUT     A (RECFM F LRECL 80 BLOCK 80
FILEDEF  7 DISK &1        LDATA     A (RECFM F LRECL 80 BLOCK 80
FILEDEF 11 DISK &1        DISTINCT  A (RECFM F LRECL 80 BLOCK 80
LOAD   LIVAL   LESUB   LIMULT
-----------------------------------------------------------------
```

```
Sample Input File for LIVAL
-------------------------------------------------------------------
 LIVAL EIGENVALUE COMPUTATION, NO REORTHOGONALIZATION
 USING INVERSE OF REAL SYMMETRIC MATRIX VIA FACTORIZATION
LINE 1     N    KMAX    NMEVS     MATNO     SO    SHIFT
         528    2640        2    721830    1.0       0.
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) )
           100     125
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)
           100.0
-------------------------------------------------------------------
```

Below is a listing of the input/output files which are accessed by the real symmetric Lanczos eigenvector program, LIVEC. Included also is a sample of the input file which LIVEC requires on file 5. The parameters in this file are supplied in free format. LIVEC computes eigenvectors for each of a user-specified subset of the eigenvalues computed by the companion program LIVAL. The matrix used in the eigenvector computation is a scaled, shifted and inverted version of a given matrix. Inversion is accomplished via matrix factorization.

```
Sample Specifications of the Input/Output Files for LIVEC
-----------------------------------------------------------------
 LIVEC EXEC, EIGENVECTORS FOR INVERSE OF REAL SYMMETRIC MATRIX
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 LIVEC    INPUT     A (RECFM F LRECL 80 BLOCK 80
FILEDEF  7 DISK &1       LDATA     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  LIVEC  LESUB  LIMULT
-----------------------------------------------------------------
```

```
Sample Input File for LIVEC
-----------------------------------------------------------------------
 LIVEC INPUT LANCZOS EIGENVECTOR COMPUTATIONS, NO REORTHOGONALIZATION
LINE 1  MATNO     N    S0   SHIFT  JPERM (ID,SIZE,SCALE,SHIFT,PERMUT?
           20  2161   -1.0     0.01     0
LINE 2  MDIMTV    MDIMRV  MBETA (MAX.DIMENSIONS,TVEC,RITVEC AND BETA
         10000     10000   2000
LINE 3       RELTOL
        .0000000001
LINE 4  MBOUND   NTVCON SVTVEC IREAD (FLAGS
             0        1      0     1
LINE 5  TVSTOP   LVCONT ERCONT  IWRITE (FLAGS
             0        1      1       1
LINE 6    RHSEED  (RANDOM GENERATOR SEED FOR STARTING VECTOR IN INVERM)
        45329517
-----------------------------------------------------------------------
```