Chapter 5

Real Symmetric Generalized Problems

5.1 Introduction

The FORTRAN codes in this Chapter address the question of computing distinct eigenvalues and corresponding eigenvectors of a real symmetric generalized eigenvalue problem. Given two real symmetric matrices A and B, where B is positive definite and its Cholesky factors are available, these codes compute real scalars λ and corresponding real-valued vectors $x \neq 0$ such that

$$Ax = \lambda Bx. \tag{5.1.1}$$

Given a real symmetric positive definite matrix B, the Cholesky decomposition of B has the form

$$B = LL^T, \tag{5.1.2}$$

where L is a lower triangular matrix. Real symmetric matrices and Cholesky factorizations are discussed in detail in Stewart [24]. See Section 2.1 for a brief summary of the properties of real symmetric matrices which we use.

Theoretically, this type of real symmetric generalized problem is equivalent to the following real symmetric problem:

$$L^{-1}AL^{-T}y = \lambda y, \quad y = L^T x. \tag{5.1.3}$$

Therefore, we could solve this type of generalized problem by applying the real symmetric Lanczos procedure given in Chapter 2 directly to the composite matrix $C \equiv L^{-1}AL^{-T}$ given in Eqn(5.1.3). However, we prefer to work directly with the generalized problem. In this setting the role of the B-matrix in the single-vector Lanczos computations is clearly displayed.

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 generalized problem. The documentation for these codes is contained in Section 2.2. As in the real symmetric case, the AB-multiplicity of a given computed 'good' Lanczos eigenvalue can be obtained only with additional computation, and the modifications required to do this additional computation are not included in the enclosed versions of these codes.

We use the following 'generalized' Lanczos recursion. For $i = 1, 2, \ldots m$ and a randomly-generated starting vector v_1 with $\|v_1\|_B = 1$, generate Lanczos vectors v_i using the following recursion.

$$\beta_{i+1} B v_{i+1} = A v_i - \alpha_i B v_i - \beta_i B v_{i-1} \tag{5.1.4}$$

where

$$\begin{aligned} \alpha_i &\equiv v_i^T (A v_i - \beta_i B v_{i-1}) \\ \beta_{i+1} &\equiv \|L^{-1}(A v_i - \alpha_i B v_i - \beta_i B v_{i-1})\| \end{aligned} \tag{5.1.5}$$

By construction, the B-norm of each Lanczos vector is one. That is, for all i, $\|v_i\|_B \equiv \left(v_i^T B v_i\right)^{1/2} = 1$.

The B-norm is used because it is the 'natural' norm for real symmetric generalized problems when the B-matrix is positive definite. Given any two distinct eigenvalues λ and μ of Eqn(5.1.1), and corresponding eigenvectors x and y, we have that $x^T B y = 0$. That is, the eigenvectors are orthogonal w.r.t. the B-norm, and the eigenvectors form a complete set of vectors. The positive definiteness of B is essential. The closer B is to being singular or indefinite, the less stable these computations will be. The generalized Lanczos recursion in Eqns (5.1.4) and (5.1.5) generates a family of real symmetric tridiagonal matrices (T-matrices) whose sizes are specified by the user.

LGVAL, the main program for the real symmetric generalized computations, calls the subroutine BISEC to compute eigenvalues of the specified tridiagonal T-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 generalized problem. The accuracy of these 'good' T-eigenvalues as eigenvalues of the generalized problem is then estimated using error estimates computed by the 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 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 LGVEC, for computing eigenvectors of the real symmetric generalized problem using a factorization of B, is used to compute the desired eigenvectors.

All of the computations are done in double precision arithmetic. Once the Lanczos matrices have been computed, the remaining computations use the same subroutines which 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 USPECA which defines and initializes the A-matrix and a subroutine USPECB which defines and initializes the factors of the B-matrix. A subroutine AMATV which computes matrix-vector multiplies Ax for the A-matrix, and a subroutine BSOLV which solves the system of equations $Bz = v$ must also be supplied. These subroutines must be constructed in such a way as to take advantage of the sparsity (and/or structure) of the two user-supplied matrices A and B and such that they are accurate.

The optional preprocessing programs PERMUT, LORDER, LFACT, and LTEST listed in Chapter 4 can also be used with the codes in this chapter. PERMUT calls the SPARSPAK Library [9] to attempt to identify a reordering or permutation P of the given matrix B for which the sparseness of B is preserved under the factorization of the permuted matrix. LORDER takes a given matrix C and permutation P and computes the sparse format for the permuted matrix, 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]. Obviously, if the B-matrix is permuted then the A-matrix must be subjected to the same permutation. These codes assume that the Cholesky factor supplied in the subroutine USPECB corresponds to the permuted B-matrix and that the AMATV subroutine supplied corresponds to the corresponding permuted A-matrix. Thus, the Lanczos codes compute the eigenvalues and eigenvectors of the permuted problem. The permutation (if any) is then unwrapped in the eigenvector program LGVEC.

5.2 LGVAL: Main Program, Eigenvalue Computations

```
C-----LGVAL  (EIGENVALUES, GENERALIZED SYMMETRIC PROBLEM)---------------
C  Authors:  Jane Cullum and Ralph A. Willoughby (Deceased)
C            Los Alamos National Laboratory
C            Los Alamos, New Mexico 87544
C
C            E-mail:  cullumj@lanl.gov
C
C  These codes are copyrighted by the authors.  These codes
C  and modifications of them or portions of them are NOT to be
C  incorporated into any commercial codes or used for any other
C  commercial purposes such as consulting for other companies,
C  without legal agreements with the authors of these Codes.
C  If these Codes or portions of them are used in other scientific or
C  engineering research works the names of the authors of these codes
C  and appropriate references to their written work are to be
C  incorporated in the derivative works.
C
C  This header is not to be removed from these codes.
C
C       REFERENCE: Cullum and Willoughby, Chapters 1,2,3,4
C       Lanczos Algorithms for Large Symmetric Eigenvalue Computations
C       VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in
C       Applied Mathematics, 2002. SIAM Publications,
C       Philadelphia, PA. USA
C
C
C     CONTAINS MAIN PROGRAM FOR COMPUTING DISTINCT EIGENVALUES OF
C     A*X = EVAL*B*X WHERE A AND B ARE REAL SYMMETRIC MATRICES,
C     B IS POSITIVE DEFINITE, AND THE CHOLESKY FACTORS OF B
C     ARE AVAILABLE FOR USE IN THE PROCEDURE.  PROCEDURE USES
C     GENERALIZATION OF LANCZOS TRIDIAGONALIZATION WITHOUT ANY
C     REORTHGONALIZATION.
C
C     PFORT VERIFIER IDENTIFIED THE FOLLOWING NONPORTABLE
C     CONSTRUCTIONS
C
C     1.  DATA/MACHEP/ STATEMENT
C     2.  ALL READ(5,*) STATEMENTS (FREE FORMAT)
C     3.  FORMAT(20A4) USED WITH EXPLANATORY HEADER EXPLAN.
C     4.  HEXADECIMAL FORMAT (4Z20) USED IN ALPHA/BETA FILES 1 AND 2.
C
C-----------------------------------------------------------------------
C
      DOUBLE PRECISION  ALPHA(5000),BETA(5001)
      DOUBLE PRECISION  V1(5000),V2(5000),VS(5000)
      DOUBLE PRECISION  LB(20),UB(20)
      DOUBLE PRECISION  BTOL,GAPTOL,TTOL,MACHEP,EPSM,RELTOL
      DOUBLE PRECISION  SCALE1,SCALE2,SCALE3,SCALE4,BISTOL,CONTOL,MULTOL
      DOUBLE PRECISION  ONE,ZERO,TEMP,TKMAX,BETAM,BKMIN,T0,T1
      REAL  G(5000),EXPLAN(20)
      INTEGER  MP(5000),NMEV(20)
```

```
      INTEGER  SVSEED,RHSEED,SVSOLD
      INTEGER IABS
      REAL  ABS
      DOUBLE PRECISION  DABS, DSQRT, DFLOAT
      EXTERNAL LSOLV, AMATV
C
C-----------------------------------------------------------------------
      DATA MACHEP/Z3410000000000000/
      EPSM = 2.0D0*MACHEP
C-----------------------------------------------------------------------
C
C     ARRAYS MUST BE DIMENSIONED AS FOLLOWS:
C     DIMENSION OF V2 ASSUMES THAT NO MORE THAN KMAX/2 EIGENVALUES
C     OF THE LANCZOS T-MATRICES ARE BEING COMPUTED IN ANY ONE OF THE
C     SUB-INTERVALS BEING CONSIDERED.  V2 CONTAINS THE UPPER AND LOWER
C     BOUNDS FOR EACH T-EIGENVALUE BEING COMPUTED BY BISEC IN ANY ONE
C     GIVEN INTERVAL.
C
C     1.  ALPHA: >= KMAX,   BETA: >= (KMAX+1)
C     2.  V1:  >= MAX(N,KMAX+1)
C     3.  V2,VS:   >= MAX(N,KMAX)
C     4.  G:  >= MAX(N,2*KMAX)
C     5.  MP:    >= KMAX
C     6.  LB,UB:  >= NUMBER OF SUBINTERVALS SUPPLIED TO BISEC.
C     7.  NMEV:  >= NUMBER OF T-MATRICES ALLOWED.
C     8.  EXPLAN:  DIMENSION IS 20.
C
C
C     IMPORTANT TOLERANCES OR SCALES THAT ARE USED REPEATEDLY
C     THROUGHOUT THE PROGRAM ARE THE FOLLOWING:
C     SCALED MACHINE EPSILON:  TTOL = TKMAX*EPSM WHERE
C     EPSM = 2*MACHINE EPSILON AND
C     TKMAX = MAX(|ALPHA(J)|,BETA(J), J = 1,MEV)
C     BISEC CONVERGENCE TOLERANCE:  BISTOL = DSQRT(1000+MEV)*TTOL
C     BISEC T-MULTIPLICITY TOLERANCE:  MULTOL = (1000+MEV)*TTOL
C     LANCZOS CONVERGENCE TOLERANCE:   CONTOL = BETA(MEV+1)*1.D-10
C-----------------------------------------------------------------------
C     OUTPUT HEADER
      WRITE(6,10)
   10 FORMAT(/' LANCZOS EIGENVALUE PROCEDURE FOR REAL SYMMETRIC GENERALI
     1ZED PROBLEMS,'/' A*X = EVAL*B*X, B POSITIVE DEFINITE WITH CHOLESKY
     1 FACTORS AVAILABLE'/)
C
C     SET PROGRAM PARAMETERS
C     SCALEK ARE USED IN TOLERANCES NEEDED IN SUBROUTINES LUMP,
C     ISOEV AND PRTEST.  USER MUST NOT MODIFY THEM.
      SCALE1 = 5.0D2
      SCALE2 = 5.0D0
      SCALE3 = 5.0D0
      SCALE4 = 1.0D4
      ONE  = 1.0D0
      ZERO = 0.0D0
      BTOL = 1.0D-8
C     BTOL = EPSM
      GAPTOL = 1.0D-8
```

```
      ICONV = 0                                                         LGV01010
      MOLD = 0                                                          LGV01020
      MOLD1 = 1                                                         LGV01030
      ICT = 0                                                           LGV01040
      MMB = 0                                                           LGV01050
      IPROJ = 0                                                         LGV01060
C-----------------------------------------------------------------------LGV01070
C     READ USER-SPECIFIED PARAMETERS FROM INPUT FILE 5 (FREE FORMAT)    LGV01080
C                                                                       LGV01090
C     READ USER-PROVIDED HEADER FOR RUN                                 LGV01100
      READ(5,20) EXPLAN                                                 LGV01110
      WRITE(6,20) EXPLAN                                                LGV01120
      READ(5,20) EXPLAN                                                 LGV01130
      WRITE(6,20) EXPLAN                                                LGV01140
   20 FORMAT(20A4)                                                      LGV01150
C                                                                       LGV01160
C     READ ORDER OF MATRICES (N) , MAXIMUM ORDER OF T-MATRIX (KMAX),    LGV01170
C     NUMBER OF T-MATRICES ALLOWED (NMEVS), AND MATRIX IDENTIFICATION   LGV01180
C     NUMBERS (MATNOA AND MATNOB)                                       LGV01190
      READ(5,20) EXPLAN                                                 LGV01200
      READ(5,*) N,KMAX,NMEVS,MATNOA,MATNOB                              LGV01210
C                                                                       LGV01220
C     READ SEEDS FOR LANCZS AND INVERR SUBROUTINES (SVSEED AND RHSEED)  LGV01230
C     READ MAXIMUM NUMBER OF ITERATIONS ALLOWED FOR EACH INVERSE        LGV01240
C     ITERATION (MXINIT) AND MAXIMUM NUMBER OF STURM SEQUENCES          LGV01250
C     ALLOWED (MXSTUR)                                                  LGV01260
      READ(5,20) EXPLAN                                                 LGV01270
      READ(5,*) SVSEED,RHSEED,MXINIT,MXSTUR                             LGV01280
C                                                                       LGV01290
C     ISTART = (0,1):  ISTART = 0 MEANS ALPHA/BETA FILE IS NOT          LGV01300
C     AVAILABLE.  ISTART = 1 MEANS ALPHA/BETA FILE IS AVAILABLE ON      LGV01310
C     FILE 2.                                                           LGV01320
C     ISTOP = (0,1):  ISTOP = 0 MEANS PROCEDURE GENERATES ALPHA/BETA    LGV01330
C     FILE AND THEN TERMINATES.  ISTOP = 1 MEANS PROCEDURE GENERATES    LGV01340
C     ALPHAS/BETAS IF NEEDED AND THEN COMPUTES EIGENVALUES AND ERROR    LGV01350
C     ESTIMATES AND THEN TERMINATES.                                    LGV01360
      READ(5,20) EXPLAN                                                 LGV01370
      READ(5,*) ISTART,ISTOP                                            LGV01380
C                                                                       LGV01390
C     IHIS = (0,1):  IHIS = 0 MEANS ALPHA/BETA FILE IS NOT WRITTEN      LGV01400
C     TO FILE 1.  IHIS = 1 MEANS ALPHA/BETA FILE IS WRITTEN TO FILE 1.  LGV01410
C     IDIST = (0,1):  IDIST = 0 MEANS DISTINCT EIGENVALUES OF           LGV01420
C     ARE NOT WRITTEN TO FILE 11.  IDIST = 1 MEANS DISTINCT             LGV01430
C     EIGENVALUES ARE WRITTEN TO FILE 11.                               LGV01440
C     IWRITE = (0,1):  IWRITE = 0 MEANS NO INTERMEDIATE OUTPUT          LGV01450
C     FROM THE COMPUTATIONS IS WRITTEN TO FILE 6.  IWRITE = 1 MEANS     LGV01460
C     EIGENVALUES AND ERROR ESTIMATES ARE WRITTEN TO FILE 6             LGV01470
C     AS THEY ARE COMPUTED.                                             LGV01480
      READ(5,20) EXPLAN                                                 LGV01490
      READ(5,*) IHIS,IDIST,IWRITE                                       LGV01500
C                                                                       LGV01510
C     READ IN THE RELATIVE TOLERANCE (RELTOL) FOR USE IN THE            LGV01520
C     SPURIOUS, T-MULTIPLICITY, AND PRTESTS.                            LGV01530
      READ(5,20) EXPLAN                                                 LGV01540
      READ(5,*) RELTOL                                                  LGV01550
```

```
C
C     READ IN THE SIZES OF THE T-MATRICES TO BE CONSIDERED.
      READ(5,20) EXPLAN
      READ(5,*) (NMEV(J), J=1,NMEVS)
C
C     READ IN THE NUMBER OF SUBINTERVALS TO BE CONSIDERED.
      READ(5,20) EXPLAN
      READ(5,*) NINT
C
C     READ IN THE LEFT-END POINTS OF THE SUBINTERVALS TO BE CONSIDERED.
C     THESE MUST BE IN ALGEBRAICALLY-INCREASING ORDER
      READ(5,20) EXPLAN
      READ(5,*) (LB(J), J=1,NINT)
C
C     READ IN THE RIGHT-END POINTS OF THE SUBINTERVALS TO BE CONSIDERED.
C     THESE MUST BE IN ALGEBRAICALLY-INCREASING ORDER
      READ(5,20) EXPLAN
      READ(5,*) (UB(J), J=1,NINT)
C
C-----------------------------------------------------------------------
C     INITIALIZE THE ARRAYS FOR THE USER-SPECIFIED MATRICES
C     AND PASS THE STORAGE LOCATIONS OF THESE ARRAYS TO THE
C     MATRIX-VECTOR MULTIPLY SUBROUTINE AMATV AND THE SOLVE
C     SUBROUTINE LSOLV.
C
      CALL USPECA(N,MATNOA)
      CALL USPECB(N,MATNOB)
C
C-----------------------------------------------------------------------
C
C     MASK UNDERFLOW AND OVERFLOW
      CALL MASK
C
C-----------------------------------------------------------------------
C
C     WRITE TO FILE 6, A SUMMARY OF THE PARAMETERS FOR THIS RUN
C
      WRITE(6,30) MATNOA,MATNOB,N,KMAX
   30 FORMAT(/3X,'A-MATRIX ID',3X,'B-MATRIX ID',4X,'ORDER OF A',4X,
     1'MAX ORDER OF T'/I14,I14,I14,I18/)
C
      WRITE(6,40) ISTART,ISTOP
   40 FORMAT(/2X,'ISTART',3X,'ISTOP'/2I8/)
C
      WRITE(6,50) IHIS,IDIST,IWRITE
   50 FORMAT(/4X,'IHIS',3X,'IDIST',2X,'IWRITE'/3I8/)
C
      WRITE(6,60) SVSEED,RHSEED
   60 FORMAT(/' SEEDS FOR RANDOM NUMBER GENERATOR'//
     1 4X,'LANCZS SEED',4X,'INVERR SEED'/2I15/)
C
      WRITE(6,70) (NMEV(J), J=1,NMEVS)
   70 FORMAT(/' SIZES OF T-MATRICES TO BE CONSIDERED'/(6I12))
C
      WRITE(6,80) RELTOL,GAPTOL,BTOL
```

```
   80 FORMAT(/' RELATIVE TOLERANCE USED TO COMBINE COMPUTED T-EIGENVALUE
     1S'/E15.3/' RELATIVE GAP TOLERANCES USED IN INVERSE ITERATION'/
     1E15.3/' RELATIVE TOLERANCE FOR CHECK ON SIZE OF BETAS'/E15.3/)
C
      WRITE(6,90) (J,LB(J),UB(J), J=1,NINT)
   90 FORMAT(/' BISEC WILL BE USED ON THE FOLLOWING INTERVALS'/
     1 (I6,2E20.6))
C
      IF (ISTART.EQ.0) GO TO 140
C
C     READ IN ALPHA BETA HISTORY
C
      READ(2,100)MOLD,NOLD,SVSOLD,MATAO,MATBO
  100 FORMAT(2I6,I12,2I8)
C
      IF (KMAX.LT.MOLD) KMAX = MOLD
      KMAX1 = KMAX + 1
C
C     CHECK THAT ORDER N, MATRIX IDS (MATNOA AND MATNOB), AND RANDOM
C     SEED (SVSEED) AGREE WITH THOSE IN THE HISTORY FILE.  IF NOT
C     PROCEDURE STOPS.
C
      ITEMP = (NOLD-N)**2 + (MATNOA-MATAO)**2 + (SVSEED-SVSOLD)**2
     1 + (MATNOB-MATBO)**2
C
      IF (ITEMP.EQ.0) GO TO 120
C
      WRITE(6,110)
  110 FORMAT(' PROGRAM TERMINATES'/    ' READ FROM FILE 2 CORRESPONDS TO
     1 DIFFERENT MATRIX THAN MATRIX SPECIFIED'/)
      GO TO 640
C
  120 CONTINUE
      MOLD1 = MOLD+1
C
      READ(2,130)(ALPHA(J), J=1,MOLD)
      READ(2,130)(BETA(J), J=1,MOLD1)
  130 FORMAT(4Z20)
C
      IF (KMAX.EQ.MOLD) GO TO 160
C
C     SAVE V1 = B*V(KMAX), VS = B*V(KMAX+1), V2 = V(KMAX+1)
      READ(2,130) (V1(J), J=1,N)
      READ(2,130) (VS(J), J=1,N)
      READ(2,130) (V2(J), J=1,N)
C
  140 CONTINUE
      IIX = SVSEED
C
C-----------------------------------------------------------------------
C
      CALL LANCZS(LSOLV,AMATV,ALPHA,BETA,V1,V2,VS,G,KMAX,MOLD1,N,IIX)
C
C-----------------------------------------------------------------------
C
```

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

```
     1' BECAUSE THE SIZE REQUESTED',I6,' IS GREATER THAN THE MAXIMUM SIZ
     1E ALLOWED',I6/)
      GO TO 540
C
  200 MP1 = MEV + 1
      BETAM = BETA(MP1)
C
      IF (IB.GE.0) GO TO 210
C
      T0 = BTOL
C
C-----------------------------------------------------------------------
C
      CALL TNORM(ALPHA,BETA,T0,T1,MEV,IBMEV)
C
C-----------------------------------------------------------------------
C
      TEMP = T0/TKMAX
      IBMEV = IABS(IBMEV)
      IF (TEMP.GE.BTOL) GO TO 210
      IBMEV = -IBMEV
      GO TO 600
C
  210 CONTINUE
      IC = MXSTUR-ICT
C
C-----------------------------------------------------------------------
C     BISEC LOOP. THE SUBROUTINE BISEC INCORPORATES DIRECTLY THE
C     T-MULTIPLICITY AND SPURIOUS TESTS. T-EIGENVALUES WILL BE
C     CALCULATED BY BISEC SEQUENTIALLY ON INTERVALS
C     (LB(J),UB(J)), J = 1,NINT).
C
C     ON RETURN FROM BISEC
C     NDIS = NUMBER OF DISTINCT EIGENVALUES OF T(1,MEV) ON UNION
C            OF THE (LB,UB) INTERVALS
C     VS = DISTINCT T-EIGENVALUES IN ALGEBRAICALLY INCREASING ORDER
C     MP = MULTIPLICITIES OF THE T-EIGENVALUES IN VS
C     MP(I) = (0,1,MI), MI>1, I=1,NDIS  MEANS:
C        (0)  VS(I) IS SPURIOUS
C        (1)  VS(I) IS T-SIMPLE AND GOOD
C        (MI) VS(I) IS MULTIPLE AND IS THEREFORE NOT ONLY GOOD BUT
C             ALSO A CONVERGED GOOD T-EIGENVALUE.
C
C
      CALL BISEC(ALPHA,BETA,V1,V2,VS,LB,UB,EPSM,TTOL,MP,NINT,
     1 MEV,NDIS,IC,IWRITE)
C
C-----------------------------------------------------------------------
C
      IF (NDIS.EQ.0) GO TO 620
C
C     COMPUTE THE TOTAL NUMBER OF STURM SEQUENCES USED TO DATE
C     COMPUTE THE BISEC CONVERGENCE AND T-MULTIPLICITY TOLERANCES USED.
C     COMPUTE THE CONVERGENCE TOLERANCE FOR EIGENVALUES OF A.
      ICT = ICT + IC
```

```
      TEMP = DFLOAT(MEV+1000)
      MULTOL = TEMP*TTOL
      TEMP = DSQRT(TEMP)
      BISTOL = TTOL*TEMP
      CONTOL = BETAM*1.D-10
C
C-----------------------------------------------------------------------
C    SUBROUTINE LUMP 'COMBINES' T-EIGENVALUES THAT ARE 'TOO CLOSE'.
C    NOTE HOWEVER THAT CLOSE SPURIOUS T-EIGENVALUES ARE NOT AVERAGED
C    WITH GOOD ONES. HOWEVER, THEY MAY BE USED TO INCREASE THE
C    MULTIPLICITY OF A GOOD T-EIGENVALUE.
C
      LOOP = NDIS
      CALL LUMP(VS,RELTOL,MULTOL,SCALE2,MP,LOOP)
C
C-----------------------------------------------------------------------
C
      IF(NDIS.EQ.LOOP) GO TO 230
C
      WRITE(6,220) NDIS, MEV, LOOP
  220 FORMAT(/I6,' DISTINCT T-EIGENVALUES WERE COMPUTED IN BISEC AT MEV
     1',I6/ 2X,' LUMP SUBROUTINE REDUCES NUMBER OF DISTINCT EIGENVALUES
     1O',I6)
C
  230 CONTINUE
      NDIS = LOOP
      BETA(MP1) = BETAM
C
C-----------------------------------------------------------------------
C     THE SUBROUTINE ISOEV LABELS THOSE SIMPLE EIGENVALUES OF T(1,MEV)
C     WITH VERY SMALL GAPS BETWEEN NEIGHBORING EIGENVALUES OF T(1,MEV)
C     TO AVOID COMPUTING ERROR ESTIMATES FOR ANY SIMPLE GOOD
C     T-EIGENVALUE THAT IS TOO CLOSE TO A SPURIOUS EIGENVALUE.
C     ON RETURN FROM ISOEV, G CONTAINS CODED MINIMAL GAPS
C     BETWEEN THE DISTINCT EIGENVALUES OF T(1,MEV). (G IS REAL).
C     G(I) < 0 MEANS MINGAP IS DUE TO LEFT GAP G(I) > 0 MEANS DUE TO
C     RIGHT GAP. MP(I) = -1 MEANS THAT THE GOOD T-EIGENVALUE IS SIMPLE
C     AND HAS A VERY SMALL MINGAP IN T(1,MEV) DUE TO A SPURIOUS
C     EIGENVALUE.  NG = NUMBER OF GOOD T-EIGENVALUES.
C     NISO = NUMBER OF ISOLATED GOOD T-EIGENVALUES.
C
      CALL ISOEV(VS,GAPTOL,MULTOL,SCALE1,G,MP,NDIS,NG,NISO)
C
C-----------------------------------------------------------------------
C
      WRITE(6,240)NG,NISO,NDIS
  240 FORMAT(/I6,' GOOD T-EIGENVALUES HAVE BEEN COMPUTED'/
     1 I6,' OF THESE ARE T-ISOLATED'/
     2 I6,' = NUMBER OF DISTINCT T-EIGENVALUES COMPUTED'/)
C
C     DO WE WRITE DISTINCT EIGENVALUES OF T-MATRIX TO FILE 11?
      IF (IDIST.EQ.0) GO TO 280
C
      WRITE(11,250) NDIS,NISO,MEV,N,SVSEED,MATNOA,MATNOB
  250 FORMAT(/4I6,I12,I8,'=ND,NIS,MEV,N,SEED,MNA,MNB'/)
```

```
C
      WRITE(11,260) (MP(I),VS(I),G(I), I=1,NDIS)
  260 FORMAT(2(I3,E25.16,E12.3))
C
      WRITE(11,270) NDIS, (MP(I), I=1,NDIS)
  270 FORMAT(/I6,' = NDIS, T-MULTIPLICITIES (0 MEANS  SPURIOUS)'/(20I4))
C
  280 CONTINUE
C
      IF (NISO.NE.0) GO TO 310
C
      WRITE(4,290) MEV
  290 FORMAT(/' AT MEV = ',I6,' THERE ARE NO ISOLATED T-EIGENVALUES'/
     1' SO NO ERROR ESTIMATES WERE COMPUTED/')
C
      WRITE(6,300)
  300 FORMAT(/' ALL COMPUTED GOOD T-EIGENVALUES ARE MULTIPLE'/
     1 ' THEREFORE ALL SUCH EIGENVALUES ARE ASSUMED TO HAVE CONVERGED')
C
      ICONV = 1
      GO TO 350
C
  310 CONTINUE
C
C-----------------------------------------------------------------------
C     SUBROUTINE INVERR COMPUTES ERROR ESTIMATES FOR ISOLATED GOOD
C     T-EIGENVALUES USING INVERSE ITERATION ON T(1,MEV). ON RETURN
C     G(J) = MINIMUM GAP IN T(1,MEV) FOR EACH VS(J), J=1,NDIS
C     G(MEV+I) = BETAM*|U(MEV)| = ERROR ESTIMATE FOR ISOLATED GOOD
C              T-EIGENVALUES, WHERE I = 1, NISO AND  BETAM = BETA(MEV+1)
C              U(MEV) IS MEVTH COMPONENT OF THE UNIT EIGENVECTOR OF T
C              CORRESPONDING TO THE ITH ISOLATED GOOD T-EIGENVALUE.
C     A NEGATIVE ERROR ESTIMATE MEANS THAT FOR THAT PARTICULAR
C     EIGENVALUE THE INVERSE ITERATION DID NOT CONVERGE IN <= MXINIT
C     STEPS AND THAT THE CORRESPONDING ERROR ESTIMATE IS QUESTIONABLE.
C
C     V2 CONTAINS THE ISOLATED GOOD T-EIGENVALUES
C     V1 CONTAINS THE MINGAPS TO THE NEAREST DISTINCT  EIGENVALUE
C        OF T(1,MEV) FOR EACH ISOLATED GOOD T-EIGENVALUE IN V2.
C     VS CONTAINS THE NDIS DISTINCT EIGENVALUES OF T(1,MEV)
C     MP CONTAINS THE CORRESPONDING CODED T-MULTIPLICITIES
C
      IT = MXINIT
      CALL INVERR(ALPHA,BETA,V1,V2,VS,EPSM,G,MP,MEV,MMB,NDIS,NISO,N,
     1 RHSEED,IT,IWRITE)
C
C-----------------------------------------------------------------------
C
C     SIMPLE CHECK FOR CONVERGENCE. CHECKS TO SEE IF ALL OF THE ERROR
C     ESTIMATES ARE SMALLER THAN CONTOL = BETAM*1.D-10.
C     IF THIS TEST IS SATISFIED, THEN CONVERGENCE FLAG, ICONV IS SET
C     TO 1.  TYPICALLY ERROR ESTIMATES ARE VERY CONSERVATIVE.
C
      WRITE(6,320) CONTOL
  320 FORMAT(/' CONVERGENCE IS TESTED USING THE CONVERGENCE TOLERANCE',
```

```
     1E13.4/)
C
      II = MEV +1
      IF = MEV+NISO
      DO 330 I = II,IF
      IF (ABS(G(I)).GT.CONTOL) GO TO 350
  330 CONTINUE
      ICONV = 1
      MMB = NMEVS
C
      WRITE(6,340) CONTOL
  340 FORMAT(' ALL COMPUTED ERROR ESTIMATES WERE LESS THAN',E15.4/
     1 ' THEREFORE PROCEDURE TERMINATES'/)
C
  350 CONTINUE
C
C     IF CONVERGENCE IS INDICATED, THAT IS ICONV = 1 ,THEN
C     THE SUBROUTINE PRTEST IS CALLED TO CHECK FOR ANY CONVERGED
C     T-EIGENVALUES THAT HAVE BEEN MISLABELLED AS SPURIOUS BECAUSE
C     THE PROJECTION OF THEIR EIGENVECTOR(S) ON THE STARTING
C     VECTOR WERE TOO SMALL.
C     NUMERICAL TESTS INDICATE THAT SUCH EIGENVALUES ARE RARE.
C     IF FOR SOME REASON MANY OF THESE HIDDEN EIGENVALUES APPEAR
C     ON SOME RUN, YOU CAN BE CERTAIN THAT SOMETHING IS FOULED UP.
C
      IF (ICONV.EQ.0) GO TO 480
C
C-----------------------------------------------------------------------
C
      CALL PRTEST (ALPHA,BETA,VS,TKMAX,EPSM,RELTOL,SCALE3,SCALE4,
     1 MP,NDIS,MEV,IPROJ)
C
C-----------------------------------------------------------------------
C
      IF(IPROJ.EQ.0) GO TO 470
C
      IF(IDIST.EQ.1)  WRITE(11,360) IPROJ
  360 FORMAT(' SUBROUTINE PRTEST WANTS TO RELABEL',I6,' SPURIOUS T-EIGEN
     1VALUES'/' WE ACCEPT RELABELLING ONLY IF LAST COMPONENT OF T-EIGENV
     1ECTOR IS L.T. 1.D-10'/)
C
      IIX = RHSEED
C
C-----------------------------------------------------------------------
C
      CALL GENRAN(IIX,G,MEV)
C
C-----------------------------------------------------------------------
C
      ITEN = -10
      NISOM = NISO + MEV
      IWRITO = IWRITE
      IWRITE = 0
C
      DO 390 J = 1,NDIS
```

```
      IF(MP(J).NE.ITEN) GO TO 390                                       LGV05410
      T0 = VS(J)                                                        LGV05420
C                                                                       LGV05430
C-----------------------------------------------------------------------LGV05440
C                                                                       LGV05450
      IT = MXINIT                                                       LGV05460
      CALL INVERM(ALPHA,BETA,V1,V2,T0,TEMP,T1,EPSM,G,MEV,IT,IWRITE)     LGV05470
C                                                                       LGV05480
C-----------------------------------------------------------------------LGV05490
C                                                                       LGV05500
      IF(TEMP.LE.1.D-10) GO TO 380                                      LGV05510
C     ERROR ESTIMATE WAS NOT SMALL REJECT RELABELLING OF THIS EIGENVALUELGV05520
      IF(IDIST.EQ.1)  WRITE(11,370) J,T0,TEMP                           LGV05530
  370 FORMAT(/' LAST COMPONENT FOR',I6,'TH EIGENVALUE',E20.12/' IS TOO LLGV05540
     1ARGE = ',E15.6,' SO DO NOT ACCEPT PRTEST RELABELLING'/)           LGV05550
      MP(J) = 0                                                         LGV05560
      IPROJ = IPROJ - 1                                                 LGV05570
      GO TO 390                                                         LGV05580
C     RELABELLING ACCEPTED                                              LGV05590
  380 NISOM = NISOM + 1                                                 LGV05600
      G(NISOM) = BETAM*TEMP                                             LGV05610
  390 CONTINUE                                                          LGV05620
      IWRITE = IWRITO                                                   LGV05630
C                                                                       LGV05640
      IF(IPROJ.EQ.0) GO TO 430                                          LGV05650
      WRITE(6,400) IPROJ                                                LGV05660
  400 FORMAT(/I6,' T-EIGENVALUES WERE RECLASSIFIED AS GOOD.'/           LGV05670
     1' THESE ARE IDENTIFIED IN FILE 3 BY A T-MULTIPLICITY OF -10'/' USELGV05680
     2R SHOULD INSPECT EACH TO MAKE SURE NEIGHBORS HAVE CONVERGED'/)    LGV05690
C                                                                       LGV05700
      IF(IDIST.EQ.1)  WRITE(11,410) IPROJ                               LGV05710
  410 FORMAT(/I6,' T-EIGENVALUES WERE RELABELLED AS GOOD'/              LGV05720
     1' BELOW IS CORRECTED T-MULTIPLICITY PATTERN'/)                    LGV05730
C                                                                       LGV05740
      WRITE(6,420) NDIS, (MP(I), I=1,NDIS)                              LGV05750
      IF(IDIST.EQ.1)  WRITE(11,420) NDIS, (MP(I), I=1,NDIS)             LGV05760
  420 FORMAT(/I6,' = NDIS, T-MULTIPLICITIES (0 MEANS  SPURIOUS)'/       LGV05770
     1 6X, ' (-10) MEANS SPURIOUS T-EIGENVALUE RELABELLED AS GOOD'/(20I4LGV05780
     1))                                                                LGV05790
C                                                                       LGV05800
C     RECALCULATE MINGAPS FOR DISTINCT T(1,MEV) EIGENVALUES.            LGV05810
  430 NM1 = NDIS - 1                                                    LGV05820
      G(NDIS) = VS(NM1)-VS(NDIS)                                        LGV05830
      G(1) = VS(2)-VS(1)                                                LGV05840
C                                                                       LGV05850
      DO 440 J = 2,NM1                                                  LGV05860
      T0 = VS(J)-VS(J-1)                                                LGV05870
      T1 = VS(J+1)-VS(J)                                                LGV05880
      G(J) = T1                                                         LGV05890
      IF (T0.LT.T1) G(J) = -T0                                          LGV05900
  440 CONTINUE                                                          LGV05910
      IF(IPROJ.EQ.0) GO TO 470                                          LGV05920
C     WRITE TO FILE 4 ERROR ESTIMATES FOR THOSE T-EIGENVALUES RELABELLEDLGV05930
      NGOOD = 0                                                         LGV05940
      DO 450 J = 1,NDIS                                                 LGV05950
```

```
      IF(MP(J).EQ.0) GO TO 450
      NGOOD = NGOOD + 1
      IF(MP(J).NE.ITEN) GO TO 450
      T0 = VS(J)
      NISO = NISO + 1
      NISOM = MEV + NISO
      WRITE(4,460) NGOOD,T0,G(NISOM),G(J)
  450 CONTINUE
  460 FORMAT(I10,E25.16,2E14.3)
C
  470 CONTINUE
C
C     WRITE THE GOOD T-EIGENVALUES TO FILE 3.  FIRST TRANSFER THEM
C     TO V2 AND THEIR T-MULTIPLICITIES TO THE CORRESPONDING POSITIONS
C     IN MP AND COMPUTE THE AB-MINGAPS, THE MINIMAL GAPS BETWEEN THE
C     GOOD T-EIGENVALUES.  THESE GAPS WILL BE PUT IN THE ARRAY G.
C     SINCE G CURRENTLY CONTAINS THE MINIMAL GAPS BETWEEN THE DISTINCT
C     EIGENVALUES OF THE T-MATRIX, THESE GAPS WILL FIRST BE
C     TRANSFERRED TO V1.  NOTE THAT V1<0 MEANS THAT THAT MINIMAL GAP
C     IN THE T-MATRIX IS DUE TO A SPURIOUS T-EIGENVALUE.
C     ALL THIS INFORMATION IS PRINTED TO FILE 3
C
  480 CONTINUE
C
      NG = 0
      DO 490 I = 1,NDIS
      IF (MP(I).EQ.0) GO TO 490
      NG = NG+1
      MP(NG) = MP(I)
      V2(NG) = VS(I)
      TEMP = G(I)
      TEMP = DABS(TEMP)
      J = I+1
      IF (G(I).LT.ZERO) J = I-1
      IF (MP(J).EQ.0) TEMP = -TEMP
      V1(NG) = TEMP
  490 CONTINUE
C
      WRITE(6,500)MEV
  500 FORMAT(//' T-EIGENVALUE CALCULATION AT MEV = ',I6,'    IS COMPLETE
     1')
C
C     NG = NUMBER OF COMPUTED DISTINCT GOOD T-EIGENVALUES.  NEXT
C     GENERATE GAPS BETWEEN GOOD T-EIGENVALUES (ABMINGAPS) AND PUT THEM
C     IN G.  G(J) < 0 MEANS THE ABMINGAP IS DUE TO THE LEFT-HAND GAP.
C
      NGM1 = NG - 1
      G(NG) = V2(NGM1)-V2(NG)
      G(1) = V2(2)-V2(1)
C
      DO 510 J = 2,NGM1
      T0 = V2(J)-V2(J-1)
      T1 = V2(J+1)-V2(J)
      G(J) = T1
      IF (T0.LT.T1) G(J) = -T0
```

```
  510 CONTINUE
C
C     WRITE GOOD T-EIGENVALUES OUT TO FILE 3.
C
      WRITE(3,520)NG,NDIS,MEV,N,SVSEED,MATNOA,MATNOB,MULTOL,IB,BTOL
  520 FORMAT(4I6,I12,2I8,'=NG,ND,MEV,N,SEED,MNA,MNB'/
     1 E20.12,I6,E13.4,' = MUTOL,INDEX MINIMAL BETA,BTOL'/
     1' EV NO',1X,'TMULT',10X,'GOOD EIGENVALUE',7X,'TMINGAP',6X,'ABMINGA
     1P')
C
      WRITE(3,530)(I,MP(I),V2(I),V1(I),G(I), I=1,NG)
  530 FORMAT(2I6,E25.16,2E14.3)
C
C     IF CONVERGENCE FLAG ICONV.NE.1 AND NUMBER OF T-MATRICES
C     CONSIDERED TO DATE IS LESS THAN NUMBER ALLOWED, INCREMENT MEV.
C     AND LOOP BACK TO 210 TO REPEAT COMPUTATIONS.  RESTORE BETA(MEV+1).
C
      BETA(MP1) = BETAM
C
      IF (MMB.LT.NMEVS.AND.ICONV.NE.1) GO TO 180
C
C     END OF LOOP ON DIFFERENT SIZE T-MATRICES ALLOWED.
C
  540 CONTINUE
C
      IF(ISTOP.EQ.0)  WRITE(6,550)
  550 FORMAT(/' T-MATRICES (ALPHA AND BETA) ARE NOW AVAILABLE, TERMINATE
     1')
      IF (IHIS.EQ.1.AND.KMAX.NE.MOLD) WRITE(1,560)
  560 FORMAT(/' ABOVE ARE THE FOLLOWING VECTORS '/
     1 '  ALPHA(I), I = 1,KMAX'/
     2 '  BETA(I), I = 1,KMAX+1'/
     3 ' FINAL THREE VECTORS USED IN LANCZS SUBROUTINE'/
     3 ' V1 = B*V(KMAX), VS = B*V(KMAX+1), V2 = V(KMAX+1)'/
     4 ' ALL VECTORS IN THIS FILE HAVE HEX FORMAT 4Z20'/
     5 ' ----- END OF FILE 1 NEW ALPHA, BETA HISTORY---------------'///)
C
      IF (ISTOP.EQ.0) GO TO 640
C
      WRITE(3,570)
  570 FORMAT(/' ABOVE ARE COMPUTED GOOD T-EIGENVALUES'/
     1 ' NG = NUMBER OF GOOD T-EIGENVALUES COMPUTED'/
     2 ' ND = NUMBER OF COMPUTED DISTINCT EIGENVALUES OF T(1,MEV)'/
     3 ' N = ORDER OF A AND B-MATRIX,  MNA, MNB = MATRIX IDENTS'/
     4 ' MULTOL = T-MULTIPLICITY TOLERANCE FOR T-EIGENVALUES IN BISEC'/
     4 ' TMULT IS THE T-MULTIPLICITY OF GOOD T-EIGENVALUE'/
     5 ' TMULT = -1 MEANS SPURIOUS T-EIGENVALUE TOO CLOSE'/
     6 ' DO NOT COMPUTE ERROR ESTIMATES FOR SUCH EIGENVALUES'/
     7 ' ABMINGAP = MINIMAL GAP BETWEEN THE COMPUTED EIGENVALUES'/
     8 ' ABMINGAP .LT. 0. MEANS MINIMAL GAP IS DUE TO LEFT-HAND GAP'/
     9 ' TMINGAP= MINIMAL GAP W.R.T.  DISTINCT EIGENVALUES IN T(1,MEV)'/
     1 ' TMINGAP .LT. 0. MEANS MINGAP IS DUE TO SPURIOUS T-EIGENVALUE'/
     2 ' ----- END OF FILE 3 GOODEIGENVALUES-----------------------'///)
C
      IF (IDIST.EQ.1) WRITE(11,580)
```

```
  580 FORMAT(/' ABOVE ARE THE DISTINCT EIGENVALUES OF T(1,MEV).'/       LGV07060
     2 ' THE FORMAT IS      T-MULTIPLICITY    T-EIGENVALUE   TMINGAP'/  LGV07070
     3 '        THIS FORMAT IS REPEATED TWICE ON EACH LINE.'/           LGV07080
     4 ' T-MULTIPLICITY = -1 MEANS THAT THE SUBROUTINE ISOEV HAS TAGGED'LGV07090
     5/'   THIS SIMPLE T-EIGENVALUE AS HAVING A VERY CLOSE SPURIOUS'/   LGV07100
     6 '   T-EIGENVALUE SO THAT NO ERROR ESTIMATE WILL BE COMPUTED'/    LGV07110
     7 '   FOR THAT EIGENVALUE IN SUBROUTINE INVERR.'/                  LGV07120
     8 ' TMINGAP .LT. 0, TMINGAP IS DUE TO LEFT GAP .GT. 0, RIGHT GAP.'/LGV07130
     9 ' EACH OF THE DISTINCT T-EIGENVALUE TABLES IS FOLLOWED'/         LGV07140
     9 ' BY THE T-MULTIPLICITY PATTERN.'/                               LGV07150
     1 ' NDIS = NUMBER OF COMPUTED DISTINCT EIGENVALUES OF T(1,MEV).'/  LGV07160
     2 ' NG = NUMBER OF GOOD T-EIGENVALUES. '/                           LGV07170
     3 ' NISO = NUMBER OF ISOLATED GOOD T-EIGENVALUES. '/               LGV07180
     4 ' NISO ALSO IS THE COUNT OF +1 ENTRIES IN T-MULTIPLICITY PATTERN.LGV07190
     5 '/' ----- END OF FILE 11 DISTINCT T-EIGENVALUES--------------'///LGV07200
     6 )                                                                LGV07210
C                                                                       LGV07220
      IF(NISO.NE.0)  WRITE(4,590)                                       LGV07230
  590 FORMAT(/' ABOVE ARE THE ERROR ESTIMATES OBTAINED FOR THE ISOLATED LGV07240
     1GOOD T-EIGENVALUES'/                                              LGV07250
     1' OBTAINED VIA INVERSE ITERATION IN THE SUBROUTINE INVERR.'/      LGV07260
     1' ALL OTHER GOOD T-EIGENVALUES HAVE CONVERGED.'/                  LGV07270
     2' ERROR ESTIMATE = BETAM*ABS(UM)'/                                LGV07280
     2' WHERE BETAM = BETA(MEV+1) AND UM = U(MEV).'/                    LGV07290
     3' U = UNIT EIGENVECTOR OF T WHERE T*U = EV*U AND EV = ISOLATED GOOLGV07300
     3D T-EIGENVALUE.'/                                                 LGV07310
     4' TMINGAP = GAP TO NEAREST DISTINCT EIGENVALUE OF T(1,MEV).'/     LGV07320
     5' TMINGAP .LT. 0. MEANS MINGAP IS DUE TO A LEFT NEIGHBOR.'/       LGV07330
     6' ERROR ESTIMATE L.T. 0 MEANS INVERSE ITERATION DID NOT CONVERGE'/LGV07340
     7' ------ END OF FILE 4 ERRINV -------------------------------'//) LGV07350
      GO TO 640                                                         LGV07360
C                                                                       LGV07370
  600 CONTINUE                                                          LGV07380
C                                                                       LGV07390
      IBB = IABS(IBMEV)                                                 LGV07400
      IF (IBMEV.LT.0) WRITE(6,610) MEV,IBB,BETA(IBB)                    LGV07410
  610 FORMAT(/' PROGRAM TERMINATES BECAUSE MEV REQUESTED = ',I6,' IS .GTLGV07420
     1',I6/' AT WHICH AN ABNORMALLY SMALL BETA = ' , E13.4,' OCCURRED'/)LGV07430
      GO TO 640                                                         LGV07440
C                                                                       LGV07450
  620 IF (NDIS.EQ.0.AND.ISTOP.GT.0) WRITE(6,630)                        LGV07460
  630 FORMAT(/' INTERVALS SPECIFIED FOR BISECT DID NOT CONTAIN ANY T-EIGLGV07470
     1ENVALUES'/' PROGRAM TERMINATES')                                  LGV07480
C                                                                       LGV07490
  640 CONTINUE                                                          LGV07500
C                                                                       LGV07510
      STOP                                                              LGV07520
C-----END OF MAIN PROGRAM FOR LANCZOS EIGENVALUE COMPUTATIONS-----------LGV07530
      END                                                               LGV07540
```

5.3 LGVEC: Main Program, Eigenvector Computations

```
C-----LGVEC (EIGENVECTORS OF A*X = EVAL*B*X)----------------------------LGV00010
C  Authors:  Jane Cullum and Ralph A. Willoughby (Deceased)             LGV00020
C            Los Alamos National Laboratory                             LGV00030
C            Los Alamos, New Mexico 87544                               LGV00040
C                                                                       LGV00050
C            E-mail:  cullumj@lanl.gov                                  LGV00060
C                                                                       LGV00070
C  These codes are copyrighted by the authors.  These codes             LGV00080
C  and modifications of them or portions of them are NOT to be          LGV00090
C  incorporated into any commercial codes or used for any other         LGV00100
C  commercial purposes such as consulting for other companies,          LGV00110
C  without legal agreements with the authors of these Codes.            LGV00120
C  If these Codes or portions of them are used in other scientific or   LGV00130
C  engineering research works the names of the authors of these codes   LGV00140
C  and appropriate references to their written work are to be           LGV00150
C  incorporated in the derivative works.                                LGV00160
C                                                                       LGV00170
C  This header is not to be removed from these codes.                   LGV00180
C                                                                       LGV00190
C         REFERENCE: Cullum and Willoughby, Chapters 1,2,3,4            LGV00191
C         Lanczos Algorithms for Large Symmetric Eigenvalue ComputationsLGV00192
C         VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in   LGV00193
C         Applied Mathematics, 2002. SIAM Publications,                 LGV00194
C         Philadelphia, PA. USA                                         LGV00195
C                                                                       LGV00196
C                                                                       LGV00200
C     CONTAINS MAIN PROGRAM FOR COMPUTING AN EIGENVECTOR CORRESPONDING  LGV00210
C     TO EACH OF A SET OF EIGENVALUES WHICH HAVE BEEN COMPUTED          LGV00220
C     ACCURATELY BY THE CORRESPONDING LANCZOS EIGENVALUE PROGRAM        LGV00230
C     (LGVAL) FOR THE SYMMETRIC, GENERALIZED PROBLEM A*X = EVAL*B*X.    LGV00240
C     LGVAL AND LGVEC ASSUME THAT B IS POSITIVE DEFINITE AND THAT THE   LGV00250
C     CHOLESKY FACTORS OF B (OR OF A PERMUTATION OF B) ARE AVAILABLE    LGV00260
C     FOR USE IN THE LANCZOS PROCEDURES.  IF B HAS BEEN PERMUTED,       LGV00270
C     THEN THESE PROCEDURES ASSUME THAT THE DATA PRESENTED FOR THE      LGV00280
C     A-MATRIX HAS BEEN SUBJECTED TO THE SAME PERMUTATION.  THAT        LGV00290
C     PERMUTATION WILL THEN BE USED AFTER THE RITZ VECTORS FOR THE      LGV00300
C     PERMUTED VERSION OF THE ORIGINAL PROBLEM HAVE BEEN COMPUTED,      LGV00310
C     TO OBTAIN THE ASSOCIATED RITZ VECTORS FOR THE ORIGINAL PROBLEM.   LGV00320
C     NOTE THAT THIS PROGRAM COULD BE MODIFIED TO COMPUTE ADDITIONAL    LGV00330
C     EIGENVECTORS FOR ANY COMPUTED EIGENVALUE WHICH IS A MULTIPLE      LGV00340
C     EIGENVALUE OF THE GIVEN A-MATRIX.  THE AMOUNT OF ADDITIONAL       LGV00350
C     COMPUTATION REQUIRED WOULD DEPEND UPON THE PARTICULAR             LGV00360
C     A-MATRIX AND B-MATRIX USED AND UPON WHAT PART OF THE              LGV00370
C     SPECTRUM OF EIGENVALUES IS BEING CONSIDERED.                      LGV00380
C                                                                       LGV00390
C     THE LANCZOS EIGENVECTOR COMPUTATIONS ASSUME THAT EACH             LGV00400
C     EIGENVALUE THAT IS BEING CONSIDERED HAS CONVERGED AS AN           LGV00410
C     EIGENVALUE OF THE ASSOCIATED LANCZOS TRIDIAGONAL MATRICES.        LGV00420
C                                                                       LGV00430
C     PFORT VERIFIER IDENTIFIED THE FOLLOWING NONPORTABLE               LGV00440
C     CONSTRUCTIONS                                                     LGV00450
```

```
C
C     1.  DATA/MACHEP/ STATEMENT
C     2.  ALL READ(5,*) STATEMENTS (FREE FORMAT)
C     3.  FORMAT(20A4) USED WITH THE EXPLANATORY HEADER, EXPLAN
C     4.  HEXADECIMAL FORMAT (4Z20) USED IN ALPHA/BETA FILES 1 AND 2.
C
C     IMPORTANT NOTE:  PROGRAM ALLOWS ENLARGEMENT OF THE ALPHA, BETA
C     ARRAYS.  IN PARTICULAR, IF ANY ONE OF THE EIGENVALUES SUPPLIED
C     IS T-SIMPLE AND NOT CLOSE TO A SPURIOUS EIGENVALUE, THE PROGRAM
C     REQUIRES THAT KMAX BE AT LEAST 11*MEV/8 + 12.  IF KMAX IS NOT
C     THIS LARGE, THEN THE PROGRAM RESETS KMAX TO THIS SIZE
C     AND EXTENDS THE ALPHA, BETA HISTORY IF REQUIRED.
C     THUS, THE DIMENSIONS OF THE ALPHA AND BETA ARRAYS MUST BE
C     LARGE ENOUGH TO ALLOW FOR THIS POSSIBILITY.
C     REMEMBER THAT THE BETA ARRAY, BETA(J), IS SUCH THAT
C     J = 1,..., KMAX+1.  SO IF THE KMAX USED BY THE PROGRAM
C     IS TO BE 3000, THEN BETA MUST BE OF LENGTH AT LEAST 3001.
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION  ALPHA(5000),BETA(5001)
      DOUBLE PRECISION  V1(5000),V2(5000),VS(5000)
      DOUBLE PRECISION  RITVEC(30000),TVEC(30000),GOODEV(50),EVNEW(50)
      DOUBLE PRECISION  EVAL,EVALN,TOLN,TTOL,ERTOL,ALFA,BATA
      DOUBLE PRECISION  MULTOL,SCALE0,STUTOL,BTOL,LB,UB
      DOUBLE PRECISION  ONE,ZERO,MACHEP,EPSM,TEMP,SUM,ERRMIN,BKMIN
      DOUBLE PRECISION  RELTOL,ERROR,TERROR,TLAST(50)
      REAL  G(10000),AMINGP(50),TMINGP(50),EXPLAN(20)
      REAL  TERR(50),ERR(50),ERRDGP(50),RNORM(50),TBETA(50)
      INTEGER  MP(50),M1(50),M2(50),MA(50),ML(50),MINT(50),MFIN(50)
      INTEGER  SVSEED,SVSOLD,RHSEED,IDELTA(50)
      INTEGER  MBOUND,NTVCON,SVTVEC,TVSTOP,LVCONT,ERCONT,TFLAG
      DOUBLE PRECISION  FINPRO
      DOUBLE PRECISION  DABS, DMAX1, DSQRT, DFLOAT
      REAL ABS
      INTEGER  IABS
      EXTERNAL LSOLV, AMATV
C-----------------------------------------------------------------------
      DATA MACHEP/Z3410000000000000/
      EPSM = 2.D0*MACHEP
C-----------------------------------------------------------------------
C
C     ARRAYS MUST BE DIMENSIONED AS FOLLOWS:
C     1.  ALPHA:  >= KMAXN,  BETA: >= (KMAXN+1) WHERE KMAXN, THE
C                 LARGEST SIZE T-MATRIX CONSIDERED BY THE PROGRAM,
C                 IS THE LARGER OF THE SIZE OF THE ALPHA, BETA HISTORY
C                 PROVIDED ON FILE 2 (IF ANY ) AND THE SIZE WHICH THE
C                 PROGRAM SPECIFIES INTERNALLY, THIS LATTER IS ALWAYS
C                 < = 11*MEV / 8  +  12, WHERE MEV IS THE SIZE
C                 T-MATRIX THAT WAS USED IN THE CORRESPONDING EIGENVALUE
C                 COMPUTATIONS.
C     2.  V1:  >= MAX(N,KMAX)
C     3.  V2,VS:   >= N
C     4.  G:  >= MAX(N,KMAX)
C     5.  RITVEC:  >= N*NGOOD, WHERE NGOOD IS NUMBER OF EIGENVALUES
C                   SUPPLIED TO THIS PROGRAM.
```

```
C     6.  TVEC:  >= CUMULATIVE LENGTH OF ALL THE T-EIGENVECTORS
C         NEEDED TO GENERATE THE DESIRED RITZ VECTORS.  AN EDUCATED
C         GUESS AT AN APPROPRIATE LENGTH CAN BE OBTAINED BY RUNNING THE
C         PROGRAM WITH THE FLAG MBOUND = 1 AND MULTIPLYING THE
C         RESULTING SIZE BY 5/4.
C     7.  GOODEV, AMINGP, TMINGP, TERR, ERR, ERRGDP, RNORM, TBETA,
C         TLAST, EVNEW, MP, MA, M1, M2, MINT, MFIN AND IDELTA ALL MUST
C         BE >= NGOOD.
C
C-----------------------------------------------------------------------
C     OUTPUT HEADER
      WRITE(6,10)
   10 FORMAT(/' LANCZOS EIGENVECTOR PROCEDURE FOR REAL SYMMETRIC MATRICE
     1S'/)
C
C     SET PROGRAM PARAMETERS
C     USER MUST NOT MODIFY SCALE0
      SCALE0 = 5.0D0
      ZERO = 0.0D0
      ONE = 1.0D0
      MPMIN = -1000
C     SET CONVERGENCE CRITERION FOR T-EIGENVECTORS.
      ERTOL = 1.D-10
C
C     READ USER-SPECIFIED PARAMETER FROM INPUT FILE 5 (FREE FORMAT)
C
C     READ USER-PROVIDED HEADER FOR RUN
      READ(5,20) EXPLAN
      WRITE(6,20) EXPLAN
   20 FORMAT(20A4)
C
C     READ IN THE MAXIMUM PERMISSIBLE DIMENSIONS FOR THE TVEC ARRAY
C     (MDIMTV), FOR THE RITVEC ARRAY (MDIMRV), AND FOR THE BETA
C     ARRAY (MBETA).
      READ(5,20) EXPLAN
      READ(5,*) MDIMTV, MDIMRV, MBETA
C
C     READ IN RELATIVE TOLERANCE (RELTOL) USED IN DETERMINING
C     APPROPRIATE SIZES FOR THE T-MATRICES TO BE USED IN THE RITZ
C     VECTOR COMPUTATIONS.
      READ(5,20) EXPLAN
      READ(5,*) RELTOL
C
C     SET FLAGS TO 0 OR 1:
C     MBOUND = 1:  PROGRAM TERMINATES AFTER COMPUTING 1ST GUESSES
C                  ON APPROPRIATE T-SIZES FOR USE IN THE RITZ VECTOR
C                  COMPUTATIONS
C     NTVCON = 0:  PROGRAM TERMINATES IF THE TVEC ARRAY IS NOT
C                  LARGE ENOUGH TO HOLD ALL THE T-EIGENVECTORS REQUIRED.
C     SVTVEC = 0:  THE T-EIGENVECTORS ARE NOT WRITTEN TO FILE 11
C                  UNLESS TVSTOP = 1
C     SVTVEC = 1:  WRITE THE T-EIGENVECTORS TO FILE 11.
C     TVSTOP = 1:  PROGRAM TERMINATES AFTER COMPUTING THE
C                  T-EIGENVECTORS
C     LVCONT = 0:  PROGRAM TERMINATES IF THE NUMBER OF T-EIGENVECTORS
```

```
C                COMPUTED IS NOT EQUAL TO THE NUMBER OF RITZ          LGV01560
C                VECTORS REQUESTED.                                   LGV01570
C     ERCONT = 0:  MEANS FOR ANY GIVEN EIGENVALUE, A RITZ VECTOR      LGV01580
C                WILL NOT BE COMPUTED FOR THAT EIGENVALUE UNLESS      LGV01590
C                A T-EIGENVECTOR HAS BEEN IDENTIFIED WITH A LAST      LGV01600
C                COMPONENT WHICH SATISFIES THE SPECIFIED              LGV01610
C                CONVERGENCE CRITERION.                               LGV01620
C     ERCONT = 1:  MEANS FOR ANY GIVEN EIGENVALUE, A RITZ VECTOR      LGV01630
C                WILL BE COMPUTED.  IF A T-EIGENVECTOR CANNOT         LGV01640
C                BE IDENTIFIED WHICH SATISFIES THE LAST               LGV01650
C                COMPONENT CRITERION, THEN THE PROGRAM WILL           LGV01660
C                USE THE T-VECTOR THAT CAME CLOSEST TO                LGV01670
C                SATISFYING THE CRITERION                             LGV01680
C     IWRITE = 1:  EXTENDED OUTPUT OF INTERMEDIATE COMPUTATIONS       LGV01690
C                IS WRITTEN TO FILE 6                                 LGV01700
C     IREAD = 0:   ALPHA/BETA FILE IS REGENERATED.                    LGV01710
C     IREAD = 1:   ALPHA/BETA FILE USED IN EIGENVALUE COMPUTATIONS    LGV01720
C                IS READ IN AND EXTENDED IF NECESSARY.  IN BOTH       LGV01730
C                CASES IREAD = 0 OR 1, THE LANCZOS VECTORS ARE        LGV01740
C                ALWAYS REGENERATED FOR THE RITZ VECTOR               LGV01750
C                COMPUTATIONS                                         LGV01760
C                                                                     LGV01770
      READ(5,20) EXPLAN                                               LGV01780
      READ(5,*) MBOUND,NTVCON,SVTVEC,IREAD                            LGV01790
C                                                                     LGV01800
      READ(5,20) EXPLAN                                               LGV01810
      READ(5,*) TVSTOP,LVCONT,ERCONT,IWRITE                           LGV01820
      IF (TVSTOP.EQ.1) SVTVEC = 1                                     LGV01830
C                                                                     LGV01840
C     READ IN SEED (RHSEED) FOR GENERATING RANDOM STARTING VECTOR     LGV01850
C     FOR INVERSE ITERATION ON THE T-MATRICES.                        LGV01860
      READ(5,20) EXPLAN                                               LGV01870
      READ(5,*) RHSEED                                                LGV01880
C                                                                     LGV01890
C     READ IN MATNOA, MATNOB = MATRIX/RUN IDENTIFICATION NUMBERS,     LGV01900
C     N = ORDER OF A-MATRIX AND B-MATRIX AND FLAG, JPERM.             LGV01910
C     JPERM = (0,1):  1 MEANS PERMUTED A AND B ARE BEING USED, 0      LGV01920
C     MEANS A AND B HAVE NOT BEEN PERMUTED.                           LGV01930
      READ(5,20) EXPLAN                                               LGV01940
      READ(5,*) MATNOA,MATNOB,N,JPERM                                 LGV01950
C                                                                     LGV01960
C-----------------------------------------------------------------------LGV01970
C     INITIALIZE THE ARRAYS FOR THE USER-SPECIFIED MATRICES           LGV01980
C     AND PASS THE STORAGE LOCATIONS OF THESE ARRAYS TO THE           LGV01990
C     MATRIX-VECTOR MULTIPLY SUBROUTINE AMATV AND THE SOLVE           LGV02000
C     SUBROUTINE LSOLV.                                               LGV02010
      CALL USPECA(N,MATNOA)                                           LGV02020
      CALL USPECB(N,MATNOB)                                           LGV02030
C                                                                     LGV02040
C-----------------------------------------------------------------------LGV02050
C                                                                     LGV02060
C     MASK UNDERFLOW AND OVERFLOW                                     LGV02070
      CALL MASK                                                       LGV02080
C                                                                     LGV02090
C-----------------------------------------------------------------------LGV02100
```

```
C     WRITE RUN PARAMETERS OUT TO FILE 6                                LGV02110
C                                                                       LGV02120
      WRITE(6,30) MATNOA,MATNOB,N,JPERM                                 LGV02130
   30 FORMAT(/4X,'A-MATRIX ID',4X,'B-MATRIX ID',4X,'SIZES OF MATRICES', LGV02140
     14X,'JPERM'/I15,I15,I21,I9)                                        LGV02150
C                                                                       LGV02160
      WRITE(6,40) MBOUND,NTVCON,SVTVEC,IREAD                            LGV02170
   40 FORMAT(/3X,'MBOUND',3X,'NTVCON',3X,'SVTVEC',3X,'IREAD'/3I9,I8)    LGV02180
C                                                                       LGV02190
      WRITE(6,50) TVSTOP,LVCONT,ERCONT,IWRITE                           LGV02200
   50 FORMAT(/3X,'TVSTOP',3X,'LVCONT',3X,'ERCONT',3X,'IWRITE'/4I9)      LGV02210
C                                                                       LGV02220
      WRITE(6,60) MDIMTV,MDIMRV,MBETA                                   LGV02230
   60 FORMAT(/3X,'MDIMTV',3X,'MDIMRV',3X,'MBETA'/2I9,I8)                LGV02240
C                                                                       LGV02250
      WRITE(6,70) RELTOL,RHSEED                                         LGV02260
   70 FORMAT(/7X,'RELTOL',3X,'RHSEED'/E13.4,I9)                         LGV02270
C                                                                       LGV02280
C                                                                       LGV02290
C     FROM FILE 3 READ IN THE NUMBER OF EIGENVALUES (NGOOD) FOR WHICH   LGV02300
C     EIGENVECTORS ARE REQUESTED, THE ORDER (MEV) OF THE LANCZOS        LGV02310
C     TRIDIAGONAL MATRIX USED IN COMPUTING THESE EIGENVALUES, THE       LGV02320
C     ORDER (NOLD) OF THE USER-SPECIFIED MATRIX USED IN THE EIGENVALUE  LGV02330
C     COMPUTATIONS, THE SEED (SVSEED) USED FOR GENERATING THE STARTING  LGV02340
C     VECTOR THAT WAS USED IN THOSE LANCZOS EIGENVALUE COMPUTATIONS,    LGV02350
C     AND THE MATRIX/RUN IDENTIFICATION NUMBERS (MATA, MATB) USED IN    LGV02360
C     THOSE COMPUTATIONS.  ALSO READ IN THE NUMBER (NDIS) OF DISTINCT   LGV02370
C     EIGENVALUES OF T(1,MEV) THAT WERE COMPUTED BUT THIS VALUE IS      LGV02380
C     NOT USED IN THE EIGENVECTOR COMPUTATIONS.                         LGV02390
C                                                                       LGV02400
      READ(3,80) NGOOD,NDIS,MEV,NOLD,SVSEED,MATA,MATB                   LGV02410
   80 FORMAT(4I6,I12,2I8)                                               LGV02420
C                                                                       LGV02430
C     READ IN THE T-MULTIPLICITY TOLERANCE USED IN THE BISEC SUBROUTINE LGV02440
C     DURING THE COMPUTATION OF THE GIVEN EIGENVALUES.                  LGV02450
C     ALSO READ IN THE FLAG IB.  IF IB < 0, THEN SOME BETA(I) IN THE    LGV02460
C     T-MATRIX FILE PROVIDED ON FILE 2 FAILED THE ORTHOGONALITY         LGV02470
C     TEST IN THE TNORM SUBROUTINE.  USER SHOULD NOTE THAT THIS VECTOR  LGV02480
C     PROGRAM PROCEEDS INDEPENDENTLY OF THE SIZE OF THE BETA USED.      LGV02490
C                                                                       LGV02500
      READ(3,90) MULTOL,IB,BTOL                                         LGV02510
   90 FORMAT(E20.12,I6,E13.4)                                           LGV02520
C                                                                       LGV02530
      TEMP = DFLOAT(MEV+1000)                                           LGV02540
      TTOL = MULTOL/TEMP                                                LGV02550
      WRITE(6,100) MULTOL,TTOL                                          LGV02560
  100 FORMAT(/' T-MULTIPLICITY TOLERANCE USED IN THE EIGENVALUE COMPUTATLGV02570
     1IONS WAS',E13.4/' SCALED MACHINE EPSILON IS',E13.4)               LGV02580
C                                                                       LGV02590
C     CONTINUE WRITE TO FILE 6 OF THE PARAMETERS FOR THIS RUN           LGV02600
C                                                                       LGV02610
      WRITE(6,110)NGOOD,NDIS,MEV,NOLD,MATA,MATB,SVSEED,MULTOL,IB,BTOL   LGV02620
  110 FORMAT(/' EIGENVALUES SUPPLIED ARE READ IN FROM FILE 3'/' FILE 3  LGV02630
     1HEADER IS'/4X,'NG',2X,'NDIS',3X,'MEV',2X,'NOLD',2X,'MATNOA',2X,   LGV02640
     1'MATNOB'/(4I6,2I8)/4X,'SVSEED',6X,'MULTOL',6X,'IB',9X,'BTOL'/     LGV02650
```

```
     1I12,E12.3,I8,E13.4/)
C
C     IS THE ARRAY RITVEC LONG ENOUGH TO HOLD ALL OF THE DESIRED
C     RITZ VECTORS (APPROXIMATE EIGENVECTORS)?
      NMAX = NGOOD*N
      IF(MBOUND.NE.0) GO TO 120
      IF(TVSTOP.NE.1.AND.NMAX.GT.MDIMRV) GO TO 1350
C
C     CHECK THAT THE ORDER N AND THE MATRIX IDENTIFICATION NUMBERS
C     MATNOA AND MATNOB SPECIFIED BY THE USER AGREE WITH THOSE READ
C     IN FROM FILE 3.
  120 ITEMP = (NOLD-N)**2 + (MATA-MATNOA)**2 + (MATB-MATNOB)**2
      IF (ITEMP.NE.0) GO TO 1370
C
C     READ IN FROM FILE 3, THE T-MULTIPLICITIES OF THE EIGENVALUES
C     WHOSE EIGENVECTORS ARE TO BE COMPUTED, THE VALUES OF THESE
C     EIGENVALUES AND THEIR MINIMAL GAPS AS EIGENVALUES OF THE
C     USER-SPECIFIED MATRIX AND AS EIGENVALUES OF THE T-MATRIX.
C
      READ(3,20) EXPLAN
      READ(3,130) (MP(J),GOODEV(J),TMINGP(J),AMINGP(J), J=1,NGOOD)
  130 FORMAT(6X,I6,E25.16,2E14.3)
C
      WRITE(6,140) (J,GOODEV(J),MP(J),TMINGP(J),AMINGP(J), J=1,NGOOD)
  140 FORMAT(/' EIGENVALUES READ IN, T-MULTIPLICITIES, T-GAPS AND A-GAPS
     1 '/4X,' J ',5X,'GOOD EIGENVALUE',5X,'MULT',4X,'  TMINGAP ',4X,
     1'  ABMINGAP '/(I6,E25.16,I4,2E15.4))
C
C     READ IN ERROR ESTIMATES
      WRITE(6,150) MEV,SVSEED
  150 FORMAT(/' THESE EIGENVALUES WERE COMPUTED USING A T-MATRIX OF
     1ORDER ',I5/' AND SEED FOR RANDOM NUMBER GENERATOR =',I12)
C     CHECK WHETHER OR NOT THERE ARE ANY T-ISOLATED EIGENVALUES IN
C     THE EIGENVALUES PROVIDED
      DO 160 J=1,NGOOD
      IF(MP(J).EQ.1) GO TO 170
  160 CONTINUE
      GO TO 190
  170 READ(4,20) EXPLAN
      READ(4,20) EXPLAN
      READ(4,20) EXPLAN
      READ(4,180) NISO
  180 FORMAT(18X,I6)
      READ(4,20) EXPLAN
      READ(4,20) EXPLAN
      READ(4,20) EXPLAN
  190 DO 220 J=1,NGOOD
      ERR(J) = 0.D0
      IF(MP(J).NE.1) GO TO 220
      READ(4,200) EVAL, ERR(J)
  200 FORMAT(10X,E25.16,E14.3)
      IF(DABS(EVAL - GOODEV(J)).LT.1.D-10) GO TO 220
      WRITE(6,210) EVAL,GOODEV(J)
  210 FORMAT(' PROBLEM WITH READ IN OF ERROR ESTIMATES'/' EIGENVALUE REA
     1D IN',E20.12,' DOES NOT MATCH GOODEV(J) ='/E20.12)
```

```
      GO TO 1590                                                        LGV03210
C                                                                       LGV03220
  220 CONTINUE                                                          LGV03230
C                                                                       LGV03240
      WRITE(6,230) (J,GOODEV(J),ERR(J), J=1,NGOOD)                      LGV03250
  230 FORMAT(' ERROR ESTMATES ='/4X,' J',5X,'EIGENVALUE',10X,' ESTIMATE LGV03260
     1'/(I6,E20.12,E14.3))                                              LGV03270
C                                                                       LGV03280
      IF(IREAD.EQ.0)  GO TO 330                                         LGV03290
C                                                                       LGV03300
C     READ IN THE SIZE OF THE T-MATRIX PROVIDED ON FILE 2.  READ IN     LGV03310
C     THE ORDER OF THE USER-SPECIFIED MATRIX , THE SEED FOR THE         LGV03320
C     RANDOM NUMBER GENERATOR, AND THE MATRIX/TEST IDENTIFICATION       LGV03330
C     NUMBERS THAT WERE USED IN THE LANCZOS EIGENVALUE COMPUTATIONS.    LGV03340
C     THESE ARE USED IN A CONSISTENCY CHECK                             LGV03350
C     IF FLAG IREAD = 0 REGENERATE ALPHA, BETA                          LGV03360
C                                                                       LGV03370
      READ(2,240) KMAX,NOLD,SVSOLD,MATA,MATB                            LGV03380
  240 FORMAT(2I6,I12,2I8)                                               LGV03390
C                                                                       LGV03400
      WRITE(6,250) KMAX,NOLD,SVSOLD,MATA,MATB                           LGV03410
  250 FORMAT(/' READ IN THE T-MATRICES STORED ON FILE 2'/' FILE 2 HEADERLGV03420
     1 IS'/2X,'KMAX',2X,'NOLD',6X,'SVSOLD',2X,'MATNOA',2X,'MATNOB'/     LGV03430
     1 2I6,I12,2I8/)                                                    LGV03440
C                                                                       LGV03450
C     CHECK THAT THE ORDER, THE MATRIX/TEST IDENTIFICATION NUMBERS      LGV03460
C     AND THE SEED FOR THE RANDOM NUMBER GENERATOR USED IN THE          LGV03470
C     LANCZOS COMPUTATIONS THAT GENERATED THE HISTORY FILE              LGV03480
C     BEING USED AGREE WITH WHAT THE USER HAS SPECIFIED.                LGV03490
      IF (NOLD.NE.N.OR.MATA.NE.MATNOA.OR.MATNOB.NE.MATB.OR.SVSOLD.NE.   LGV03500
     1 SVSEED) GO TO 1390                                               LGV03510
C                                                                       LGV03520
      KMAX1 = KMAX + 1                                                  LGV03530
C                                                                       LGV03540
C     READ IN THE T-MATRICES FROM FILE 2.  THESE ARE USED TO GENERATE   LGV03550
C     THE T-EIGENVECTORS THAT WILL BE USED IN THE RITZ VECTOR           LGV03560
C     COMPUTATIONS.  HISTORY MUST BE STORED IN MACHINE FORMAT           LGV03570
C     ((4Z20) FOR IBM/3081).                                            LGV03580
C                                                                       LGV03590
      READ(2,260) (ALPHA(J), J=1,KMAX)                                  LGV03600
      READ(2,260) (BETA(J), J=1,KMAX1)                                  LGV03610
  260 FORMAT(4Z20)                                                      LGV03620
C                                                                       LGV03630
      READ(2,260) (V1(J), J=1,N)                                        LGV03640
      READ(2,260) (VS(J), J=1,N)                                        LGV03650
      READ(2,260) (V2(J), J=1,N)                                        LGV03660
C                                                                       LGV03670
C     KMAX MAY BE ENLARGED IF THE SIZE AT WHICH THE EIGENVALUE          LGV03680
C     COMPUTATIONS WERE PERFORMED IS ESSENTIALLY KMAX AND               LGV03690
C     THERE IS AT LEAST ONE EIGENVALUE THAT IS T-SIMPLE AND             LGV03700
C     T-ISOLATED, IN THE SENSE THAT IF ITS NEAREST NEIGHBOR IS TOO      LGV03710
C     CLOSE THAT NEIGHBOR IS A 'GOOD' T-EIGENVALUE.                     LGV03720
      DO 270 J = 1,NGOOD                                                LGV03730
      IF(MP(J).EQ.1) GO TO 290                                          LGV03740
  270 CONTINUE                                                          LGV03750
```

```
      WRITE(6,280)                                                      LGV03760
  280 FORMAT(/' ALL EIGENVALUES USED ARE T-MULTIPLE OR CLOSE TO SPURIOUSLGV03770
     1 T-EIGENVALUES'/' SO DO NOT CHANGE KMAX')                         LGV03780
      IF(KMAX.LT.MEV) GO TO 1410                                        LGV03790
      GO TO 310                                                         LGV03800
C                                                                       LGV03810
  290 KMAXN= 11*MEV/8 + 12                                              LGV03820
      IF(MBETA.LE.KMAXN) GO TO 1570                                     LGV03830
      IF(KMAX.GE.KMAXN )  GO TO 310                                     LGV03840
      WRITE(6,300) KMAX, KMAXN                                          LGV03850
  300 FORMAT(' ENLARGE KMAX FROM ',I6,' TO ',I6)                        LGV03860
      MOLD1 = KMAX + 1                                                  LGV03870
      KMAX = KMAXN                                                      LGV03880
      GO TO 380                                                         LGV03890
C                                                                       LGV03900
  310 WRITE(6,320) KMAX                                                 LGV03910
  320 FORMAT(/' T-MATRICES HAVE BEEN READ IN FROM FILE 2'/' THE LARGEST LGV03920
     1SIZE T-MATRIX ALLOWED IS',I6/)                                    LGV03930
C                                                                       LGV03940
      IF(IREAD.EQ.1) GO TO 400                                          LGV03950
C                                                                       LGV03960
C     REGENERATE THE ALPHA AND BETA                                     LGV03970
C                                                                       LGV03980
  330 MOLD1 = 1                                                         LGV03990
C                                                                       LGV04000
      DO 340 J = 1,NGOOD                                                LGV04010
      IF(MP(J).EQ.1) GO TO 360                                          LGV04020
  340 CONTINUE                                                          LGV04030
      KMAX = MEV + 12                                                   LGV04040
      WRITE(6,350) KMAX                                                 LGV04050
  350 FORMAT(/' ALL EIGENVALUES FOR WHICH EIGENVECTORS ARE TO BE COMPUTELGV04060
     1D ARE EITHER T-MULTIPLE OR CLOSE TO'/' A SPURIOUS T-EIGENVALUE. THLGV04070
     1EREFORE SET KMAX = MEV + 12 = ',I7)                               LGV04080
      GO TO 380                                                         LGV04090
C                                                                       LGV04100
  360 KMAXN = 11*MEV/8 + 12                                             LGV04110
      IF(MBETA.LE.KMAXN) GO TO 1570                                     LGV04120
      WRITE(6,370) KMAXN                                                LGV04130
  370 FORMAT(' SET KMAX EQUAL TO ',I6)                                  LGV04140
      KMAX = KMAXN                                                      LGV04150
C                                                                       LGV04160
  380 WRITE(6,390) MOLD1,KMAX                                           LGV04170
  390 FORMAT(/' LANCZS SUBROUTINE GENERATES ALPHA(J), BETA(J+1), J =',  LGV04180
     1 I6,' TO ', I6/)                                                  LGV04190
C                                                                       LGV04200
C-----------------------------------------------------------------------LGV04210
C                                                                       LGV04220
      IIX = SVSEED                                                      LGV04230
      CALL LANCZS(LSOLV,AMATV,ALPHA,BETA,V1,V2,VS,G,KMAX,MOLD1,N,IIX)   LGV04240
C                                                                       LGV04250
C-----------------------------------------------------------------------LGV04260
C                                                                       LGV04270
  400 CONTINUE                                                          LGV04280
C                                                                       LGV04290
C     THE SUBROUTINE STURMI DETERMINES THE SMALLEST SIZE T-MATRIX FOR   LGV04300
```

```
C     WHICH THE EIGENVALUE IN QUESTION IS A T-EIGENVALUE (TO WITHIN A
C     GIVEN TOLERANCE) AND IF POSSIBLE THE SMALLEST SIZE T-MATRIX
C     FOR WHICH IT IS A DOUBLE T-EIGENVALUE (TO WITHIN THE SAME
C     TOLERANCE).  THE SIZE T-MATRIX USED IN THE RITZ VECTOR
C     COMPUTATIONS IS THEN DETERMINED BY LOOPING ON SIZE OF THE
C     T-EIGENVECTORS, STARTING WITH A T-SIZE DETERMINED BY STURMI.
C
C
      STUTOL = SCALE0*MULTOL
      IF(IWRITE.EQ.1) WRITE(6,410)
  410 FORMAT(' FROM STURMI')
      DO 450 J = 1,NGOOD
      EVAL = GOODEV(J)
C     COMPUTE THE TOLERANCES USED BY STURMI TO DETERMINE AN INTERVAL
C     CONTAINING THE EIGENVALUE EVAL.
      TEMP = DABS(EVAL)*RELTOL
      TOLN = DMAX1(TEMP,STUTOL)
C
C-----------------------------------------------------------------------
C
      CALL STURMI(ALPHA,BETA,EVAL,TOLN,EPSM,KMAX,MK1,MK2,IC,IWRITE)
C
C-----------------------------------------------------------------------
C
C     STORE THE COMPUTED ORDERS OF T-MATRICES FOR LATER PRINTOUT
      M1(J) = MK1
      M2(J) = MK2
      ML(J) = (MK1 + 3*MK2)/4
      IF(MK2.EQ.KMAX)  ML(J) = KMAX
C
      IF(IC.GT.0) GO TO 430
C     IC = 0 MEANS THERE WAS NO EIGENVALUE IN THE DESIGNATED INTERVAL
C     BY T-SIZE KMAX.  THIS MEANS THAT THE EIGENVALUE PROVIDED HAS
C     NOT YET CONVERGED SO ITS EIGENVECTOR IS NOT COMPUTED.
      WRITE(6,420) J,GOODEV(J),MK1,MK2
  420 FORMAT(I6,'TH EIGENVALUE',E20.12,' HAS NOT CONVERGED '/
     1' SO DO NOT COMPUTE ANY T-EIGENVECTOR OR RITZ VECTOR FOR IT'
     1/' MK1 AND MK2 FOR THIS EIGENVALUE WERE',2I6)
      MP(J) = MPMIN
      MA(J) = -2*KMAX
      GO TO 450
C     COMPUTE AN APPROPRIATE SIZE T-MATRIX FOR THE GIVEN EIGENVALUE.
  430 IF(M2(J).EQ.KMAX) GO TO 440
C     M1 AND M2 WERE BOTH DETERMINED
      MA(J) = (3*M1(J) + M2(J))/4  + 1
      GO TO 450
C     M2 NOT DETERMINED
  440 MA(J) = (5*M1(J))/4  + 1
C
  450 CONTINUE
C
      IF (IWRITE.EQ.1) WRITE(6,460) (MA(JJ), JJ=1,NGOOD)
  460 FORMAT(/' 1ST GUESS AT APPROPRIATE SIZE T-MATRICES'/
     1 ' ACTUAL VALUES WILL PROBABLY BE 1/4 AGAIN AS MUCH'/(13I6))
C
```

```
C     PRINT OUT TO FILE 10 1ST GUESSES AT SIZES OF THE T-MATRICES TO
C     BE USED IN THE EIGENVECTOR COMPUTATIONS.
C     PROGRAM LOOPS ON T-SIZE TO DETERMINE APPROPRIATE SIZE T-MATRIX.
      WRITE(10,470) N,KMAX
  470 FORMAT(2I8,' = ORDER OF USER MATRIX AND MAX ORDER OF T(1,MEV)')
C
      WRITE(10,480)
  480 FORMAT(/' 1ST GUESS AT APPROPRIATE SIZE T-MATRICES'/
     1 ' ACTUAL VALUES WILL PROBABLY BE 1/4 AGAIN AS MUCH'/)
C
      WRITE(10,490)
  490 FORMAT(4X,'J',3X,'AB-EIGENVALUE',4X,'M1(J)',1X,'M2(J)',1X,'MA(J)')
C
      WRITE(10,500) (J,GOODEV(J),M1(J),M2(J), MA(J), J=1,NGOOD)
  500 FORMAT(I5,E19.12,3I6)
C
      IF(MBOUND.EQ.1) WRITE(10,510)
  510 FORMAT(/'  EV = AB-EIGENVALUE IS A GOOD EIGENVALUE OF T(1,MEV)'/
     1' M1 = SMALLEST VALUE OF M SUCH THAT T(1,M) HAS AT LEAST'/
     1 '     ONE EIGENVALUE IN THE INTERVAL (EV-TOLN,EV+TOLN)'/
     1 '     TOLN(J) = DMAX1(EV(J)*RELTOL, SCALE0*MULTOL)'/
     1 ' M2 = SMALLEST M (IF ANY) SUCH THAT IN THE ABOVE INTERVAL'/
     1 '      T(1,M) HAS AT LEAST TWO EIGENVALUES '/
     1 ' IABS(MA(J)) = APPROPRIATE SIZE T-MATRIX FOR EV(J)'/
     1 ' INITIAL VALUE OF MA(J) IS CHOSEN HEURISTICALLY'/
     1 ' PROGRAM LOOPS ON SIZE OF T-MATRIX TO GET BETTER SIZE'/
     1 ' END OF SIZES OF T-MATRICES FILE 10'///)
C
C
C     TERMINATE AFTER COMPUTING 1ST GUESSES AT  SIZES OF THE
C     T-MATRICES REQUIRED FOR THE GIVEN EIGENVALUES?
      IF(MBOUND.EQ.1) GO TO 1430
C
C
C     IS THERE ROOM FOR ALL OF THE REQUESTED T-EIGENVECTORS?
      MTOL = 0
      DO 520 J = 1,NGOOD
      IF(MP(J).EQ.MPMIN) GO TO 520
      MTOL = MTOL + IABS(MA(J))
  520 CONTINUE
      MTOL = (5*MTOL)/4
      IF(MTOL.GT.MDIMTV.AND.NTVCON.EQ.0) GO TO 1450
C
C-----------------------------------------------------------------------
C     GENERATE A RANDOM VECTOR TO BE USED REPEATEDLY BY
C     SUBROUTINE INVERM
C
      IIL = RHSEED
      CALL GENRAN(IIL,G,KMAX)
C
C----------------------------------------------------------------------
C
C     LOOP ON GIVEN EIGENVALUES TO COMPUTE THE CORRESPONDING
C     T-EIGENVECTOR.
C
```

```
      MTOL = 0                                                          LGV05410
      NTVEC = 0                                                         LGV05420
      ILBIS = 0                                                         LGV05430
      DO 710 J = 1,NGOOD                                                LGV05440
      ICOUNT = 0                                                        LGV05450
      ERRMIN = 10.D0                                                    LGV05460
      MABEST = MPMIN                                                    LGV05470
      IF(MP(J).EQ.MPMIN) GO TO 710                                      LGV05480
      TFLAG = 0                                                         LGV05490
      EVAL = GOODEV(J)                                                  LGV05500
      TEMP = DABS(EVAL)*RELTOL                                          LGV05510
      UB = EVAL + DMAX1(STUTOL,TEMP)                                    LGV05520
      LB = EVAL - DMAX1(STUTOL,TEMP)                                    LGV05530
  530 KMAXU = IABS(MA(J))                                               LGV05540
C                                                                       LGV05550
C     SELECT A SUITABLE INCREMENT FOR THE ORDERS OF THE T-MATRICES      LGV05560
C     TO BE CONSIDERED IN DETERMINING APPROPRIATE SIZES FOR THE RITZ    LGV05570
C     VECTOR COMPUTATIONS.                                              LGV05580
      IF(ICOUNT.GT.0) GO TO 550                                         LGV05590
C     SELECT IDELTA(J) BASED UPON THE T-MULTIPLICITY OBTAINED           LGV05600
      IF(M2(J).EQ.KMAX) GO TO 540                                       LGV05610
C     M2 DETERMINED                                                     LGV05620
      IDELTA(J) = ((3*M1(J) + 5*M2(J))/8  +  1 - IABS(MA(J)))/10  +  1  LGV05630
      GO TO 550                                                         LGV05640
C     M2 NOT DETERMINED                                                 LGV05650
  540 MAMAX = MIN0((11*MEV)/8 + 12, (13*M1(J))/8 + 1)                   LGV05660
      IDELTA(J) = (MAMAX - IABS(MA(J)))/10  +  1                        LGV05670
  550 ICOUNT = ICOUNT + 1                                               LGV05680
C                                                                       LGV05690
C-----------------------------------------------------------------------LGV05700
C     TO MIMIMIZE THE EFFECT OF THE ONE-SIDED ACCEPTANCE TEST FOR       LGV05710
C     EIGENVALUES IN THE BISEC SUBROUTINE, RECOMPUTE THE GIVEN          LGV05720
C     EIGENVALUE AT THE SPECIFIED KMAXU                                 LGV05730
C                                                                       LGV05740
      CALL LBISEC(ALPHA,BETA,EPSM,EVAL,EVALN,LB,UB,TTOL,KMAXU,NEVT)     LGV05750
C                                                                       LGV05760
C-----------------------------------------------------------------------LGV05770
C                                                                       LGV05780
C     CHECK WHETHER OR NOT GIVEN T-MATRIX HAS AN EIGENVALUE IN THE      LGV05790
C     SPECIFIED INTERVAL AND IF SO WHAT ITS T-MULTIPLICITY IS.          LGV05800
C                                                                       LGV05810
      IF(NEVT.EQ.1) GO TO 590                                           LGV05820
      IF(NEVT.NE.0) GO TO 570                                           LGV05830
      ILBIS = 1                                                         LGV05840
      WRITE(6,560) EVAL,KMAXU                                           LGV05850
  560 FORMAT(/' PROBLEM ENCOUNTERED IN RECOMPUTATION OF USER-SUPPLIED EILGV05860
     1GENVALUE',E20.12/' THE SIZE T-MATRIX SPECIFIED',I6,' DOES NOT     LGV05870
     1HAVE AN EIGENVALUE IN THE INTERVAL SPECIFIED'/' THEREFORE NO EIGENLGV05880
     1VECTOR WILL BE COMPUTED FOR THIS PARTICULAR EIGENVALUE'/)         LGV05890
      GO TO 610                                                         LGV05900
C                                                                       LGV05910
  570 IF(NEVT.GT.1)  WRITE(6,580) EVAL,KMAXU                            LGV05920
  580 FORMAT(/' PROBLEM ENCOUNTERED IN RECOMPUTATION OF USER-SUPPLIED   LGV05930
     1EIGENVALUE',E20.12/' FOR THE SIZE T-MATRIX SPECIFIED =',I6,' THE  LGV05940
     1GIVEN EIGENVALUE IS T-MULTIPLE IN THE INTERVAL SPECIFIED'/' SOMETHLGV05950
```

```
     1ING IS WRONG, THEREFORE NO EIGENVECTOR WILL BE COMPUTED FOR THIS E
     1IGENVALUE'/)
C
      MP(J) = MPMIN
      MA(J) = -2*KMAX
      GO TO 710
C
  590 CONTINUE
      ILBIS = 0
C
      EVNEW(J) = EVALN
      EVAL = EVALN
      MTOL = MTOL+KMAXU
C
C     IS THERE ROOM IN TVEC ARRAY FOR THE NEXT T-EIGENVECTOR?
C     IF NOT, SKIP TO RITZ VECTOR COMPUTATIONS.
      IF (MTOL.GT.MDIMTV) GO TO 720
C
      IT = 3
      KINT = MTOL - KMAXU +1
C
C     RECORD THE BEGINNING AND END OF THE T-EIGENVECTOR BEING COMPUTED
      MINT(J) = KINT
      MFIN(J) = MTOL
C
C-----------------------------------------------------------------------
C     SUBROUTINE INVERM DOES INVERSE ITERATION, I.E. SOLVES
C     (T(1,KMAXU) - EVAL)*U = RHS  FOR EACH EIGENVALUE TO OBTAIN THE
C     DESIRED T-EIGENVECTOR.
C
      IF(IWRITE.EQ.1)  WRITE(6,600) J
  600 FORMAT(/I6,'TH EIGENVALUE')
C
      CALL INVERM(ALPHA,BETA,V1,TVEC(KINT),EVAL,ERROR,TERROR,EPSM,
     1 G,KMAXU,IT,IWRITE)
C
C-----------------------------------------------------------------------
C
      TERR(J) = TERROR
      TLAST(J) = ERROR
      KMAXU1 = KMAXU + 1
      TBETA(J) = BETA(KMAXU1)*ERROR
C
C     AFTER COMPUTING EACH OF THE T-EIGENVECTORS,
C     CHECK THE SIZE OF THE ERROR ESTIMATE, ERROR.
C     IF THIS ESTIMATE IS NOT AS SMALL AS DESIRED AND
C     |MA(J)| < ML(J), ATTEMPT TO INCREASE THE SIZE OF |MA(J)|
C     AND REPEAT THE T-EIGENVECTOR COMPUTATIONS.
C
      IF(ERROR.LT.ERTOL.OR.TFLAG.EQ.1)  GO TO 700
C
      IF(ERROR.GE.ERRMIN) GO TO 610
C     LAST COMPONENT IS LESS THAN MINIMAL TO DATE
      ERRMIN = ERROR
      MABEST = MA(J)
```

```
  610 CONTINUE                                                          LGV06510
C                                                                       LGV06520
      IF(MA(J).GT.0)  ITEST = MA(J) + IDELTA(J)                         LGV06530
      IF(MA(J).LT.0)  ITEST = -(IABS(MA(J)) + IDELTA(J))                LGV06540
      IF(IABS(ITEST).LE.ML(J).AND.ICOUNT.LE.10) GO TO 630               LGV06550
C     NEW MA(J) IS GREATER THAN MAXIMUM ALLOWED.                        LGV06560
      IF(ERCONT.EQ.0.OR.MABEST.EQ.MPMIN) GO TO 650                      LGV06570
      TFLAG = 1                                                         LGV06580
      MA(J) = MABEST                                                    LGV06590
      IF(ILBIS.EQ.0)  MTOL = MTOL - KMAXU                               LGV06600
      WRITE(6,620) MA(J)                                                LGV06610
  620 FORMAT(' 10 ORDERS WERE CONSIDERED.  NONE SATISFIED THE ERROR TESTLGV06620
     1'/' THEREFORE USE THE BEST ORDER OBTAINED FOR THE EIGENVECTORS'   LGV06630
     1,I6)                                                              LGV06640
      GO TO 530                                                         LGV06650
C                                                                       LGV06660
  630 MA(J) = ITEST                                                     LGV06670
C                                                                       LGV06680
      MT = IABS(MA(J))                                                  LGV06690
      IF(IWRITE.EQ.1) WRITE(6,640)  MT                                  LGV06700
  640 FORMAT(/' CHANGE SIZE OF T-MATRIX TO ',I6,' RECOMPUTE T-EIGENVECTOLGV06710
     1R')                                                               LGV06720
C                                                                       LGV06730
      IF(ILBIS.EQ.0)  MTOL = MTOL - KMAXU                               LGV06740
C                                                                       LGV06750
      GO TO 530                                                         LGV06760
C                                                                       LGV06770
C     APPROPRIATE SIZE T-MATRIX WAS NOT OBTAINED                        LGV06780
  650 CONTINUE                                                          LGV06790
      WRITE(10,660) J,EVAL,MP(J)                                        LGV06800
  660 FORMAT(/' ON 10 INCREMENTS NOT ABLE TO IDENTIFY APPROPRIATE SIZE  LGV06810
     1T-MATRIX FOR'/                                                    LGV06820
     1' EIGENVALUE(',I4,') = ',E20.12,' T-MULTIPLICITY =',I4/)          LGV06830
      IF(M2(J).EQ.KMAX) WRITE(10,670)                                   LGV06840
      IF(M2(J).LT.KMAX) WRITE(10,680)                                   LGV06850
  670 FORMAT(/' ORDERS TESTED RANGED FROM 5*M1(J)/4 TO APPROXIMATELY    LGV06860
     1 '/'  MIN(11*MEV/8,13*M1(J)/8)'/)                                 LGV06870
  680 FORMAT(/' ORDERS TESTED RANGED FROM (3*M1(J)+M2(J))/4 TO APPROXIMALGV06880
     1TELY'/'  (3*M1(J) + 5*M2(J))/8.'/)                                LGV06890
      WRITE(10,690)                                                     LGV06900
  690 FORMAT(' ALLOWING LARGER ORDERS FOR THE T-MATRICES MAY RESULT IN  LGV06910
     1 SUCCESS'/' BUT PROBABLY WILL NOT.  PROBLEM IS PROBABLY DUE TO'   LGV06920
     1 /' LACK OF CONVERGENCE OF GIVEN EIGENVALUE, CHECK THE ERROR ESTIMLGV06930
     1ATE'/)                                                            LGV06940
      MP(J) = MPMIN                                                     LGV06950
      IF(ILBIS.EQ.0)  MTOL = MTOL - KMAXU                               LGV06960
      GO TO 710                                                         LGV06970
  700 NTVEC = NTVEC + 1                                                 LGV06980
C                                                                       LGV06990
  710 CONTINUE                                                          LGV07000
      NGOODC = NGOOD                                                    LGV07010
      GO TO 740                                                         LGV07020
C                                                                       LGV07030
C     COME HERE IF THERE IS NOT ENOUGH ROOM FOR ALL OF T-EIGENVECTORS   LGV07040
  720 NGOODC = J-1                                                      LGV07050
```

```
      WRITE(6,730)  J, MTOL, MDIMTV                                     LGV07060
  730 FORMAT(/'  NOT ENOUGH ROOM IN TVEC FOR ',I4,'TH T-VECTOR'/'  T-DIMLGV07070
     1ENSION REQUESTED = ',I6,' BUT TVEC HAS DIMENSION = ',I6/)         LGV07080
      IF(NGOODC.EQ.0)  GO TO 1470                                       LGV07090
      MTOL = MTOL-KMAXU                                                 LGV07100
C                                                                       LGV07110
  740 CONTINUE                                                          LGV07120
C                                                                       LGV07130
C     THE LOOP ON T-EIGENVECTOR COMPUTATIONS IS COMPLETE.               LGV07140
C     WRITE OUT THE SIZE T-MATRICES THAT WILL BE USED FOR               LGV07150
C     THE RITZ VECTOR COMPUTATIONS.                                     LGV07160
C                                                                       LGV07170
      WRITE(10,750)                                                     LGV07180
  750 FORMAT(/' SIZES OF T-MATRICES THAT WILL BE USED IN THE RITZ COMPUTLGV07190
     1ATIONS'/5X,'J',16X,'GOODEV(J)',1X,'MA(J)')                        LGV07200
C                                                                       LGV07210
      WRITE(10,760)   (J,GOODEV(J),MA(J), J=1,NGOOD)                    LGV07220
  760 FORMAT(I6,E25.14,I6)                                              LGV07230
      WRITE(10,510)                                                     LGV07240
C                                                                       LGV07250
      WRITE(6,770) MTOL                                                 LGV07260
  770 FORMAT(/' THE CUMULATIVE LENGTH OF THE T-EIGENVECTORS IS',I18)    LGV07270
C                                                                       LGV07280
      WRITE(6,780) NTVEC,NGOOD                                          LGV07290
  780 FORMAT(/I6,' T-EIGENVECTORS OUT OF',I6,' REQUESTED WERE COMPUTED')LGV07300
C                                                                       LGV07310
C     SAVE THE T-EIGENVECTORS ON FILE 11?                               LGV07320
      IF(TVSTOP.NE.1.AND.SVTVEC.EQ.0) GO TO 840                         LGV07330
C                                                                       LGV07340
      WRITE(11,790) NTVEC,MTOL,MATNOA,MATNOB,SVSEED                     LGV07350
  790 FORMAT(I6,3I8,I12,' = NTVEC,MTOL,MATNOA,MATNOB,SVSEED')           LGV07360
C                                                                       LGV07370
      DO 820 J=1,NGOODC                                                 LGV07380
C     IF MP(J) = MPMIN THEN NO SUITABLE T-EIGENVECTOR IS AVAILABLE      LGV07390
C     FOR THAT EIGENVALUE.                                              LGV07400
      IF(MP(J).EQ.MPMIN) WRITE(11,800) J,MA(J),GOODEV(J),MP(J)          LGV07410
  800 FORMAT(2I6,E20.12,I6/' TH EIGVAL,T-SIZE,EVALUE,FLAG,NO EIGVEC')   LGV07420
      IF(MP(J).NE.MPMIN) WRITE(11,810) J,MA(J),GOODEV(J),MP(J)          LGV07430
  810 FORMAT(I6,I6,E20.12,I6/' T-EIGVEC,SIZE T,EVALUE OF A,MP(J)')      LGV07440
      IF(MP(J).EQ.MPMIN) GO TO 820                                      LGV07450
      KI = MINT(J)                                                      LGV07460
      KF = MFIN(J)                                                      LGV07470
C                                                                       LGV07480
      WRITE(11,260) (TVEC(K), K=KI,KF)                                  LGV07490
C                                                                       LGV07500
  820 CONTINUE                                                          LGV07510
C                                                                       LGV07520
      IF(TVSTOP.NE.1) GO TO 840                                         LGV07530
C                                                                       LGV07540
      WRITE(6,830) TVSTOP, NTVEC,NGOOD                                  LGV07550
  830 FORMAT(/' USER SET TVSTOP = ',I1/                                 LGV07560
     1' THEREFORE PROGRAM TERMINATES AFTER T-EIGENVECTOR COMPUTATIONS'/ LGV07570
     1' T-EIGENVECTORS THAT WERE COMPUTED ARE SAVED ON FILE 11'/        LGV07580
     1I8,' T-EIGENVECTORS WERE COMPUTED OUT OF',I7,' REQUESTED'/)       LGV07590
C                                                                       LGV07600
```

```
      GO TO 1590                                                        LGV07610
C                                                                       LGV07620
  840 CONTINUE                                                          LGV07630
C     IF NOT ABLE TO COMPUTE ALL THE REQUESTED T-EIGENVECTORS,          LGV07640
C     CONTINUE WITH THE LANCZOS VECTOR COMPUTATIONS ANYWAY?             LGV07650
      IF(NTVEC.NE.NGOOD.AND.LVCONT.EQ.0) GO TO 1490                     LGV07660
C                                                                       LGV07670
C     COMPUTE THE MAXIMUM SIZE OF THE T-MATRIX USED FOR THOSE           LGV07680
C     EIGENVALUES WITH GOOD ERROR ESTIMATES.                            LGV07690
C                                                                       LGV07700
      KMAXU = 0                                                         LGV07710
      DO 850 J = 1,NGOODC                                               LGV07720
      MT = IABS(MA(J))                                                  LGV07730
      IF(MT.LT.KMAXU.OR.MP(J).EQ.MPMIN) GO TO 850                       LGV07740
      KMAXU = MT                                                        LGV07750
  850 CONTINUE                                                          LGV07760
C                                                                       LGV07770
      IF(KMAXU.EQ.0) GO TO 1530                                         LGV07780
C                                                                       LGV07790
      WRITE(6,860) KMAXU                                                LGV07800
  860 FORMAT(/I6,' = LARGEST SIZE T-MATRIX TO BE USED IN THE RITZ VECTORLGV07810
     1 COMPUTATIONS')                                                   LGV07820
C                                                                       LGV07830
C     COUNT THE NUMBER OF RITZ VECTORS NOT BEING COMPUTED               LGV07840
      MREJEC = 0                                                        LGV07850
      DO 870 J=1,NGOODC                                                 LGV07860
  870 IF(MP(J).EQ.MPMIN)  MREJEC = MREJEC + 1                           LGV07870
      MREJET = MREJEC + (NGOOD-NGOODC)                                  LGV07880
      IF(MREJET.NE.0) WRITE(6,880) MREJET                               LGV07890
  880 FORMAT(/' RITZ VECTORS ARE NOT COMPUTED FOR',I6,' OF THE EIGENVALULGV07900
     1ES'/)                                                             LGV07910
      NACT = NGOODC - MREJEC                                            LGV07920
      WRITE(6,890) NGOOD,NTVEC,NACT                                     LGV07930
  890 FORMAT(/I6,' RITZ VECTORS WERE REQUESTED'/I6,' T-EIGENVECTORS WERELGV07940
     1 COMPUTED'/I6,' RITZ VECTORS WILL BE COMPUTED'/)                  LGV07950
C     CHECK IF THERE ARE ANY RITZ VECTORS TO COMPUTE                    LGV07960
      IF(MREJEC.EQ.NGOODC) GO TO 1510                                   LGV07970
C                                                                       LGV07980
C     CONTINUE WITH THE LANCZOS VECTOR COMPUTATIONS?                    LGV07990
      IF(LVCONT.EQ.0.AND.MREJEC.NE.0) GO TO 1490                        LGV08000
C                                                                       LGV08010
C     NOW COMPUTE THE RITZ VECTORS.  REGENERATE THE                     LGV08020
C     LANCZOS VECTORS.                                                  LGV08030
C                                                                       LGV08040
      DO 900 I = 1,NMAX                                                 LGV08050
  900 RITVEC(I) = ZERO                                                  LGV08060
C                                                                       LGV08070
C-----------------------------------------------------------------------LGV08080
C     REGENERATE THE STARTING VECTOR. THIS MUST BE GENERATED AND        LGV08090
C     NORMALIZED PRECISELY THE WAY IT WAS DONE IN THE EIGENVALUE        LGV08100
C     COMPUTATIONS, OTHERWISE THERE WILL BE A MISMATCH BETWEEN          LGV08110
C     THE T-EIGENVECTORS THAT HAVE BEEN COMPUTED FROM THE T-MATRICES    LGV08120
C     READ IN FROM FILE 2 AND THE LANCZOS VECTORS THAT ARE              LGV08130
C     BEING REGENERATED.                                                LGV08140
C                                                                       LGV08150
```

```
      IIL = SVSEED
      CALL GENRAN(IIL,G,N)
C
C-----------------------------------------------------------------------
C
      DO 910 K = 1,N
  910 V2(K) = G(K)
C
C-----------------------------------------------------------------------
C     COMPUTE L-TRANSPOSE*V2 AND ITS NORM
      ISOLV = 2
      CALL LSOLV(V2,VS,ISOLV)
      SUM = FINPRO(N,VS(1),1,VS(1),1)
C-----------------------------------------------------------------------
C
C     NORMALIZE STARTING VECTORS:  (V2-TRANSPOSE*B*V2) = 1
      SUM = ONE/DSQRT(SUM)
      DO 920 K = 1,N
      VS(K) = SUM*VS(K)
  920 V2(K) = SUM*V2(K)
C
C-----------------------------------------------------------------------
C     INITIALIZE V1 = B*V2 = L*VS
      ISOLV = 1
      CALL LSOLV(VS,V1,ISOLV)
C-----------------------------------------------------------------------
C
      DO 930 K = 1,N
      VS(K) = V1(K)
  930 V1(K) = ZERO
C
      IVEC = 1
      BATA = ZERO
C
      GO TO 1000
C
C     VS = B*V(I),  V1 = B*V(I-1),  V2 = V(I)
  940 CONTINUE
      SUM = BATA
C
C-----------------------------------------------------------------------
C     COMPUTE V1 = A*V2 - SUM*V1
      CALL MATVEC(V2,V1,SUM)
C     COMPUTE ALFA
      ALFA = FINPRO(N,V1(1),1,V2(1),1)
C-----------------------------------------------------------------------
C
      DO 950 K = 1,N
  950 V1(K) = V1(K)-ALFA*VS(K)
C
C     SET V1 = B*V(IVEC) AND VS = (NEW BATA)*B*V(IVEC+1)
      DO 960 K = 1,N
      TEMP = V1(K)
      V1(K) = VS(K)
  960 VS(K) = TEMP
```

```
C
C-----------------------------------------------------------------------
C     COMPUTE V2 = (L-INVERSE)*VS
      ISOLV = 3
      CALL LSOLV(VS,V2,ISOLV)
C     COMPUTE NEXT BATA
      SUM = FINPRO(N,V2(1),1,V2(1),1)
C-----------------------------------------------------------------------
C
      BATA = DSQRT(SUM)
      TEMP = BETA(IVEC)
      TEMP = DABS(BATA - TEMP)/TEMP
      IF (TEMP.LT.1.0D-10)GO TO 980
C
C     THE BETA BEING REGENERATED DO NOT MATCH THE BETA IN FILE 2.
C     SOMETHING IS WRONG IN THE LANCZOS VECTOR GENERATION.
C     PROGRAM TERMINATES FOR USER TO CORRECT THE PROBLEM
C     WHICH MUST BE IN THE STARTING VECTOR GENERATION OR IN
C     THE SUBROUTINES AMATV AND LSOLV SUPPLIED.
C     THESE SUBROUTINES MUST BE THE SAME ONES USED IN THE
C     EIGENVALUE COMPUTATIONS OR A MISMATCH WILL ENSUE.
C
      WRITE(6,970) IVEC,BATA,BETA(IVEC),TEMP
  970 FORMAT(/2X,'IVEC',16X,'BATA',10X,'BETA(IVEC)',14X,'RELDIF'/I6,
     13E20.12/' IN LANCZOS VECTOR REGENERATION THE ENTRIES OF THE TRIDIA
     1GONAL MATRICES BEING'/' GENERATED ARE NOT THE SAME AS THOSE IN THE
     1 MATRIX SUPPLIED ON FILE 2.'/' THEREFORE SOMETHING IS BEING INITIA
     1LIZED OR COMPUTED DIFFERENTLY FROM THE WAY'/' IT WAS COMPUTED IN T
     1HE EIGENVALUE COMPUTATIONS'/' THE PROGRAM TERMINATES FOR THE USER
     1TO DETERMINE WHAT THE PROBLEM IS'/)
      GO TO 1590
  980 CONTINUE
C
C-----------------------------------------------------------------------
      ISOLV = 4
      CALL LSOLV(V2,V2,ISOLV)
C-----------------------------------------------------------------------
C
      SUM = ONE/BATA
      DO 990 K = 1,N
      V2(K) = SUM*V2(K)
  990 VS(K) = SUM*VS(K)
C
 1000 CONTINUE
C
      LFIN = 0
      DO 1020 J = 1,NGOODC
      LL = LFIN
      LFIN = LFIN + N
C
      IF(IABS(MA(J)).LT.IVEC.OR.MP(J).EQ.MPMIN) GO TO 1020
      II = IVEC + MINT(J) - 1
      TEMP = TVEC(II)
C     II IS THE (IVEC)TH COMPONENT OF THE T-EIGENVECTOR CONTAINED
C     IN TVEC(MINT(J)).
```

```
C
      DO 1010 K = 1,N
      LL = LL + 1
 1010 RITVEC(LL) = TEMP*V2(K) + RITVEC(LL)
C
 1020 CONTINUE
C
      IVEC = IVEC + 1
      IF (IVEC.LE.KMAXU) GO TO 940
C
C     RITZVECTOR GENERATION IS COMPLETE. B-NORMALIZE EACH RITZVECTOR.
C     NOTE THAT IF CERTAIN RITZ VECTORS WERE NOT COMPUTED THEN THAT
C     PORTION OF THE RITVEC ARRAY WAS NOT UTILIZED.
C
      LFIN = 0
      DO 1090 J = 1,NGOODC
C
      KK = LFIN
      LFIN = LFIN + N
      IF(MP(J).EQ.MPMIN) GO TO 1090
C
      DO 1030 K = 1,N
      KK = KK + 1
 1030 V2(K) = RITVEC(KK)
C
C-----------------------------------------------------------------------
      ISOLV = 2
      CALL LSOLV(V2,VS,ISOLV)
      SUM = FINPRO(N,VS(1),1,VS(1),1)
C-----------------------------------------------------------------------
C
      SUM = DSQRT(SUM)
      RNORM(J) = SUM
      TEMP = DABS(ONE-SUM)
      SUM = ONE/SUM
C
      DO 1040 K = 1,N
      VS(K) = SUM*VS(K)
      V2(K) = SUM*V2(K)
 1040 CONTINUE
C
C-----------------------------------------------------------------------
      ISOLV = 1
      CALL LSOLV(VS,V1,ISOLV)
C-----------------------------------------------------------------------
C
C     V1 = B*V2
      EVAL = EVNEW(J)
C
C     COMPUTE ERROR IN RITZ VECTOR CONSIDERED AS A EIGENVECTOR OF A.
C     V1 = A*RITVEC - EVAL*B*RITVEC
C
C-----------------------------------------------------------------------
      CALL AMATV(V2,V1,EVAL)
      SUM = FINPRO(N,V1(1),1,V1(1),1)
```

```
C-----------------------------------------------------------------------
C
      SUM = DSQRT(SUM)
      ERR(J) = SUM
      GAP = ABS(AMINGP(J))
      ERRDGP(J) = SUM/GAP
C
C
      IF (JPERM.EQ.0) GO TO 1050
C
C-----------------------------------------------------------------------
C     ON RETURN V2 = P(TRANSPOSE)*V2
      IPERM = 2
      CALL LPERM(V2,V1,IPERM)
C-----------------------------------------------------------------------
C
 1050 CONTINUE
      KK = LFIN - N
      DO 1060 K = 1,N
      KK = KK + 1
 1060 RITVEC(KK) = V2(K)
C
      IF (IWRITE.NE.0) WRITE(6,1070) J,GOODEV(J)
 1070 FORMAT(/I5,' TH EIGENVALUE CONSIDERED = ',E20.12/)
C
      IF (IWRITE.NE.0) WRITE(6,1080) TERR(J),TBETA(J),TEMP
 1080 FORMAT(' NORM OF ERROR IN T-EIGENVECTOR = ',E14.3/
     1 ' BETA(MA(J)+1)*U(MA(J)) = ',E14.3/
     1 ' ABS(NORM(RITVEC) - 1.0)  = ',E14.3/)
C
 1090 CONTINUE
C
C     RITZVECTORS ARE NORMALIZED AND ERROR ESTIMATES ARE IN ERR ARRAY
C     AND IN ERRDGP ARRAY. STORE EVERYTHING
C
C
      WRITE(9,1100)
 1100 FORMAT(2X,'AB-EIGENVALUE',2X,'MA(J)',2X,'AB-MINGAP',5X,'ABERROR',1
     1X, 'ABERROR/GAP',6X,'TERROR')
C
      WRITE(13,1110)
 1110 FORMAT(12X,'AB-EIGENVALUE',5X,'RITZNORM',5X,'ABMINGAP',5X,
     1 'TBETA(J)',5X,'TLAST(J)')
C
      DO 1140 J=1,NGOODC
C
      IF(MP(J).EQ.MPMIN) GO TO 1140
C
      WRITE(9,1120)EVNEW(J),MA(J),AMINGP(J),ERR(J),ERRDGP(J),TERR(J)
 1120 FORMAT(E15.8,I6,4E12.4)
C
      WRITE(13,1130) EVNEW(J),RNORM(J),AMINGP(J),TBETA(J),TLAST(J)
 1130 FORMAT(E25.14,4E13.5)
C
 1140 CONTINUE
```

```
C
      IF(MREJEC.EQ.0) GO TO 1220
      WRITE(9,1150)
 1150 FORMAT(/' RITZ VECTORS WERE NOT COMPUTED FOR THE FOLLOWING EIGENVA
     1LUES'/' EITHER BECAUSE THEY HAD NOT CONVERGED OR BECAUSE THE ERROR
     1 ESTIMATE'/'  WAS NOT AS SMALL AS DESIRED'/)
C
      DO 1210 J = 1,NGOODC
      IF(MP(J).NE.MPMIN) GO TO 1210
C     WRITE OUT MESSAGE FOR EACH EIGENVALUE FOR WHICH NO EIGENVECTOR
C     WAS COMPUTED.
C
      WRITE(9,1160)
 1160 FORMAT(2X,'AB-EIGENVALUE',3X,'MA(J)',5X,'AMINGP(J)',6X,'TLAST(J)',
     13X,'MP(J)')
      WRITE(9,1170) GOODEV(J),MA(J),AMINGP(J),TBETA(J),MP(J)
 1170 FORMAT(E15.8,I8,2E14.4,I8)
C
      WRITE(13,1180)
 1180 FORMAT(/' RITZ VECTORS WERE NOT COMPUTED FOR THE FOLLOWING EIGENVA
     1LUES'/' BECAUSE THEY HAD NOT CONVERGED'/)
C
      WRITE(13,1190)
 1190 FORMAT(2X,'AB-EIGENVALUE',3X,'MA(J)',3X,'M1(J)',3X,'M2(J)',3X,'MP(
     1J)'/)
      WRITE(13,1200) GOODEV(J),MA(J),M1(J),M2(J),MP(J)
 1200 FORMAT(E15.8,4I8)
C
 1210 CONTINUE
 1220 CONTINUE
C
      WRITE(9,1230)
 1230 FORMAT(/' ABOVE ARE ERROR ESTIMATES FOR THE AB AND T EIGENVECTORS'
     1 /' ASSOCIATED WITH THE AB-EIGENVALUES LISTED IN COLUMN 1'/
     1 ' ABERROR = NORM(A*X - EV*B*X)  TERROR = NORM(T*Y - EV*Y)
     1 '/' WHERE T = T(1,MA(J))    X = RITZ VECTOR = V*Y  V = SUCCESSIVE
     1 '/' LANCZOS VECTORS. ABMINGAP = GAP TO NEAREST AB-EIGENVALUE'//)
C
      WRITE(13,1240)
 1240 FORMAT(/' ABOVE ARE ERROR ESTIMATES ASSOCIATED WITH THE AB-EIGVALS
     1 '/' RITZNORM = NORM(COMPUTED RITZ VECTOR)'/
     1 ' TBETA(J) = BETA(MA(J)+1)*Y(MA(J)),  T*Y = EVAL*Y'/
     1 ' TLAST(J) = Y(MA(J))'/
     1 ' ABMINGAP = GAP TO NEAREST AB-EIGENVALUE'/)
C
C     NUMBER OF RITZ VECTORS COMPUTED
      NCOMPU = NGOODC - MREJEC
      WRITE(12,1250) N,NCOMPU,NGOODC,MATNOA,MATNOB
 1250 FORMAT(3I6,2I8,' SIZE A, NO.RITZVECS, NO.EVALS,MATNOA,MATNOB')
C
      LFIN = 0
      DO 1310 J = 1,NGOODC
      LINT = LFIN + 1
      LFIN = LFIN + N
C
```

```
      IF(MP(J).EQ.MPMIN) GO TO 1290                                     LGV10910
C     RITZ VECTOR WAS COMPUTED                                          LGV10920
      WRITE(12,1260) J, GOODEV(J), MP(J)                                LGV10930
 1260 FORMAT(I6,4X,E20.12,I6,' J, AB-EIGENVAL, MP(J)')                  LGV10940
C                                                                       LGV10950
      WRITE(12,1270) ERR(J),ERRDGP(J)                                   LGV10960
 1270 FORMAT(2E15.5,'= NORM(A*Z-EVAL*B*Z), NORM(A*Z-EVAL*B*Z)/ABMINGAP')LGV10970
C                                                                       LGV10980
      WRITE(12,1280) (RITVEC(LL), LL=LINT,LFIN)                         LGV10990
 1280 FORMAT(4E20.12)                                                   LGV11000
      GO TO 1310                                                        LGV11010
C     NO RITZ VECTOR WAS COMPUTED FOR THIS EIGENVALUE                   LGV11020
 1290 WRITE(12,1300) J,GOODEV(J),MP(J)                                  LGV11030
 1300 FORMAT(I6,4X,E20.12,I6,' J,AB-EIGVALUE,NO RITZ VECTOR COMPUTED')  LGV11040
C                                                                       LGV11050
 1310 CONTINUE                                                          LGV11060
C                                                                       LGV11070
C     DID ANY T-MATRICES INCLUDE OFF-DIAGONAL ENTRIES SMALLER THAN      LGV11080
C     DESIRED, AS SPECIFIED BY BTOL?                                    LGV11090
C                                                                       LGV11100
      IF(IB.GT.0) GO TO 1340                                            LGV11110
C                                                                       LGV11120
      WRITE(6,1320) KMAXU                                               LGV11130
 1320 FORMAT(/' FOR LARGEST T-MATRIX CONSIDERED',I7,' CHECK THE SIZE OF LGV11140
     1BETAS')                                                           LGV11150
C                                                                       LGV11160
C-----------------------------------------------------------------------LGV11170
C                                                                       LGV11180
      CALL TNORM(ALPHA,BETA,BKMIN,TEMP,KMAXU,IBMT)                      LGV11190
C                                                                       LGV11200
C-----------------------------------------------------------------------LGV11210
C                                                                       LGV11220
      IF(IBMT.LT.0) WRITE (6,1330)                                      LGV11230
 1330 FORMAT(/' WARNING THE T-MATRICES FOR ONE OR MORE OF THE EIGENVALUELGV11240
     1S CONSIDERED'/' HAD AN OFF-DIAGONAL ENTRY THAT WAS SMALLER THAN THLGV11250
     1E BETA TOLERANCE THAT WAS SPECIFIED'/)                            LGV11260
 1340 CONTINUE                                                          LGV11270
C                                                                       LGV11280
       GO TO 1590                                                       LGV11290
C                                                                       LGV11300
 1350 WRITE(6,1360) NGOOD,NMAX,MDIMRV                                   LGV11310
 1360 FORMAT(/I4,' RITZ VECTORS WERE REQUESTED BUT THE REQUIRED DIMENSIOLGV11320
     1N',I6/' IS LARGER THAN THE USER-SPECIFIED DIMENSION OF RITVEC',I6 LGV11330
     1/' THEREFORE, THE EIGENVECTOR PROCEDURE TERMINATES FOR THE USER TOLGV11340
     1 INTERVENE')                                                      LGV11350
C                                                                       LGV11360
      GO TO 1590                                                        LGV11370
C                                                                       LGV11380
 1370 WRITE(6,1380) NOLD,N,MATA,MATNOA,MATB,MATNOB                      LGV11390
 1380 FORMAT(/' PARAMETERS READ FROM FILE 3 DO NOT AGREE WITH USER-SPECILGV11400
     1FIED VALUES'/' NOLD,N,MATA,MATNOA,MATB,MATNOB = '/2I6,4I12/       LGV11410
     1' THEREFORE PROGRAM TERMINATES FOR USER TO RESOLVE DIFFERENCES'/) LGV11420
C                                                                       LGV11430
      GO TO 1590                                                        LGV11440
C                                                                       LGV11450
```

```
 1390 WRITE(6,1400)                                                    LGV11460
 1400 FORMAT(/' PARAMETERS IN ALPHA,BETA FILE READ IN DO NOT AGREE WITH LGV11470
     1THOSE'/' SPECIFIED BY THE USER.  THEREFORE PROGRAM TERMINATES FOR'LGV11480
     1/' USER TO RESOLVE DIFFERENCES'/)                                LGV11490
C                                                                      LGV11500
      GO TO 1590                                                       LGV11510
C                                                                      LGV11520
 1410 WRITE(6,1420) KMAX,MEV                                           LGV11530
 1420 FORMAT(/' ALPHA,BETA HEADER HAS KMAX = ',I6/                     LGV11540
     1' BUT EIGENVALUES WERE COMPUTED AT MEV = ',I6,' PROGRAM STOPS'/)  LGV11550
C                                                                      LGV11560
      GO TO 1590                                                       LGV11570
C                                                                      LGV11580
 1430 WRITE(6,1440)                                                    LGV11590
 1440 FORMAT(/' PROGRAM COMPUTED 1ST GUESSES AT T-MATRIX SIZES AND READ LGV11600
     1THEM TO FILE 10'/' THEN TERMINATED AS REQUESTED.')               LGV11610
      GO TO 1590                                                       LGV11620
C                                                                      LGV11630
 1450 WRITE(6,1460) MTOL, MDIMTV                                       LGV11640
 1460 FORMAT(/' PROGRAM TERMINATES BECAUSE THE TVEC DIMENSION ANTICIPATELGV11650
     1D',I7/'  IS LARGER THAN THE TVEC DIMENSION',I7,' SPECIFIED BY THE LGV11660
     1USER.'/'  USER MAY RESET THE TVEC DIMENSION AND RESTART THE PROGRALGV11670
     1M')                                                              LGV11680
      GO TO 1590                                                       LGV11690
C                                                                      LGV11700
 1470 WRITE(6,1480)                                                    LGV11710
 1480 FORMAT(/' PROGRAM TERMINATES BECAUSE NO SUITABLE T-EIGENVECTORS WELGV11720
     1RE IDENTIFIED'/'  FOR ANY OF THE EIGENVALUES SUPPLIED.  PROBLEM COLGV11730
     1ULD BE CAUSED'/'  BY TOO SMALL A TVEC DIMENSION OR SIMPLY THAT SUILGV11740
     1TABLE T-VECTORS COULD'/'  NOT BE IDENTIFIED.  USER SHOULD CHECK OULGV11750
     1TPUT'/)                                                          LGV11760
      GO TO 1590                                                       LGV11770
C                                                                      LGV11780
 1490 WRITE(6,1500) LVCONT,NTVEC,NGOOD                                 LGV11790
 1500 FORMAT(/' LVCONT FLAG =',I2,' AND NUMBER ',I5,' OF T-EIGENVECTORS LGV11800
     1 COMPUTED N.E.'/' NUMBER',I5,' REQUESTED SO PROGRAM TERMINATES'/) LGV11810
      GO TO 1590                                                       LGV11820
C                                                                      LGV11830
 1510 WRITE(6,1520)                                                    LGV11840
 1520 FORMAT(/' PROGRAM TERMINATES WITHOUT COMPUTING RITZ VECTORS'/     LGV11850
     1' BECAUSE ALL T-EIGENVECTORS WERE REJECTED AS NOT SUITABLE FOR THELGV11860
     1 RITZ VECTOR'/' COMPUTATIONS.  PROBABLE CAUSE IS LACK OF CONVERGENLGV11870
     1CE OF THE EIGENVALUES SUPPLIED'/)                                LGV11880
       GO TO 1590                                                      LGV11890
C                                                                      LGV11900
 1530 WRITE(6,1540)                                                    LGV11910
 1540 FORMAT(/' PROGRAM INDICATES THAT IT IS NOT POSSIBLE TO COMPUTE ANYLGV11920
     1 OF THE'/' REQUESTED EIGENVECTORS. THEREFORE PROGRAM TERMINATES') LGV11930
      DO 1550 J=1,NGOODC                                               LGV11940
 1550 WRITE(6,1560)  J,GOODEV(J),MP(J)                                 LGV11950
 1560 FORMAT(/4X,' J',9X,'AB-EIGENVALUE',4X,'MP(J)'/I6,E20.12,I9)       LGV11960
      GO TO 1590                                                       LGV11970
C                                                                      LGV11980
 1570 WRITE(6,1580) MBETA,KMAXN                                        LGV11990
 1580 FORMAT(/' PROGRAM TERMINATES BECAUSE THE STORAGE ALLOTTED FOR THE LGV12000
```

```
     1BETA ARRAY',I8/' IS NOT SUFFICIENT FOR THE ENLARGED KMAX =',I8,' TLGV12010
     1HAT THE PROGRAM WANTS'/' USER CAN ENLARGE THE ALPHA AND BETA ARRAYLGV12020
     1S AND RERUN THE PROGRAM'/)                                        LGV12030
C                                                                       LGV12040
 1590 CONTINUE                                                          LGV12050
C                                                                       LGV12060
      STOP                                                              LGV12070
C-----END OF MAIN PROGRAM FOR LANCZOS EIGENVECTOR COMPUTATIONS----------LGV12080
      END                                                               LGV12090
```

5.4 LGMULT: LANCZS and Sample Matrix-Vector Multiply Subnroutines

```
C-----LGMULT-----------------------------------------------------------LGM00010
C  Authors:  Jane Cullum and Ralph A. Willoughby (Deceased)            LGM00020
C           Los Alamos National Laboratory                             LGM00030
C           Los Alamos, New Mexico 87544                               LGM00040
C                                                                      LGM00050
C           E-mail:  cullumj@lanl.gov                                  LGM00060
C                                                                      LGM00070
C  These codes are copyrighted by the authors.  These codes            LGM00080
C  and modifications of them or portions of them are NOT to be         LGM00090
C  incorporated into any commercial codes or used for any other        LGM00100
C  commercial purposes such as consulting for other companies,         LGM00110
C  without legal agreements with the authors of these Codes.           LGM00120
C  If these Codes or portions of them are used in other scientific or  LGM00130
C  engineering research works the names of the authors of these codes  LGM00140
C  and appropriate references to their written work are to be          LGM00150
C  incorporated in the derivative works.                               LGM00160
C                                                                      LGM00170
C  This header is not to be removed from these codes.                  LGM00180
C                                                                      LGM00190
C        REFERENCE: Cullum and Willoughby, Chapters 1,2,3,4            LGM00191
C        Lanczos Algorithms for Large Symmetric Eigenvalue ComputationsLGM00192
C        VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in   LGM00193
C        Applied Mathematics, 2002. SIAM Publications,                 LGM00194
C        Philadelphia, PA. USA                                         LGM00195
C                                                                      LGM00196
C                                                                      LGM00200
C     CONTAINS SUBROUTINES  LANCZS, USPECA, USPECB, AMATV, AND LSOLV.  LGM00210
C     TO BE USED WITH THE LANCZOS CODES FOR THE GENERALIZED EIGENVALUE LGM00220
C     PROBLEM, A*X = EVAL*B*X, WHERE A AND B ARE REAL SYMMETRIC, AND   LGM00230
C     B IS POSITIVE DEFINITE WITH ITS CHOLESKY FACTORS AVAILABLE.      LGM00240
C                                                                      LGM00250
C     NONPORTABLE CONSTRUCTIONS:                                       LGM00260
C     1.  THE ENTRY MECHANISM USED TO PASS THE STORAGE                 LGM00270
C         LOCATIONS OF THE USER-SPECIFIED MATRICES FROM THE            LGM00280
C         SUBROUTINES USPECA AND USPECB TO THE MATRIX-VECTOR           LGM00290
C         SUBROUTINE, AMATV AND TO THE SOLVE SUBROUTINE, LSOLV.        LGM00300
C     2.  IN SAMPLE USPECA AND USPECB:  FREE FORMAT (8,*); FORMAT      LGM00310
C         (20A4), AND FORMAT (4Z20).                                   LGM00320
C                                                                      LGM00330
C-----LANCZS-COMPUTE LANCZOS TRIDIAGONAL MATRICES-----------------------LGM00340
C                                                                      LGM00350
      SUBROUTINE LANCZS(LSOLV,MATVEC,ALPHA,BETA,V1,V2,VS,G,KMAX,MOLD1,N,LGM00360
     1 IIX)                                                            LGM00370
C                                                                      LGM00380
C-----------------------------------------------------------------------LGM00390
      DOUBLE PRECISION  ALPHA(1), BETA(1), V1(1), V2(1), VS(1)         LGM00400
      DOUBLE PRECISION SUM, ONE, ZERO, TEMP                            LGM00410
      REAL  G(1)                                                       LGM00420
      DOUBLE PRECISION  FINPRO,DSQRT                                   LGM00430
      EXTERNAL MATVEC, LSOLV                                           LGM00440
```

```
C-----------------------------------------------------------------------
C     ALPHA, BETA, AND LANCZOS VECTOR GENERATION
C     ALPHA BETA GENERATION STARTS WITH IVEC = 1, BETA(1) = ZERO,
C     V2 =  RANDOM VECTOR WITH UNIT B-NORM, VS = B*V2, AND V1 = 0.;
C     OR STARTS WITH AN EXISTING ALPHA/BETA FILE AND THE MOST
C     RECENTLY GENERATED V2, VS, AND V1.
C
      ZERO = 0.0D0
      ONE  = 1.0D0
      IF (MOLD1.GT.1) GO TO 40
      BETA(1) = ZERO
      IIL = IIX
C
C-----------------------------------------------------------------------
      CALL GENRAN(IIL,G,N)
C-----------------------------------------------------------------------
C
      DO 10 K = 1,N
   10 V2(K) = G(K)
C
C-----------------------------------------------------------------------
C     COMPUTE L-TRANSPOSE*V2 AND ITS NORM
      ISOLV = 2
      CALL LSOLV(V2,VS,ISOLV)
      SUM = FINPRO(N,VS(1),1,VS(1),1)
C-----------------------------------------------------------------------
C
C     NORMALIZE STARTING VECTORS:  (V2-TRANSPOSE*B*V2) = 1
      SUM = ONE/DSQRT(SUM)
      DO 20 K = 1,N
      VS(K) = SUM*VS(K)
   20 V2(K) = SUM*V2(K)
C
C-----------------------------------------------------------------------
C     INITIALIZE V1 = B*V2 = L*VS
      ISOLV = 1
      CALL LSOLV(VS,V1,ISOLV)
C-----------------------------------------------------------------------
C
      DO 30 K = 1,N
      VS(K) = V1(K)
   30 V1(K) = 0.D0
   40 CONTINUE
C
C     INITIALIZATIONS ARE:  VS = B*V(I),  V1 = B*V(I-1),  V2 = V(I)
C
      DO 80 IVEC = MOLD1,KMAX
      SUM = BETA(IVEC)
C
C-----------------------------------------------------------------------
C     COMPUTE V1 = A*V2 - SUM*V1
      CALL MATVEC(V2,V1,SUM)
C     COMPUTE ALPHA(I)
      SUM = FINPRO(N,V1(1),1,V2(1),1)
C-----------------------------------------------------------------------
```

```
C
      ALPHA(IVEC) = SUM
      DO 50 K = 1,N
   50 V1(K) = V1(K)-SUM*VS(K)
C
C     SET V1 = B*V(IVEC) AND VS = BETA(IVEC+1)*B*V(IVEC+1)
      DO 60 K = 1,N
      TEMP = V1(K)
      V1(K) = VS(K)
   60 VS(K) = TEMP
C
C-----------------------------------------------------------------------
C     COMPUTE V2 = (L-INVERSE)*VS
      ISOLV = 3
      CALL LSOLV(VS,V2,ISOLV)
C     COMPUTE BETA(IVEC+1)
      SUM = FINPRO(N,V2(1),1,V2(1),1)
C-----------------------------------------------------------------------
C
      IN = IVEC+1
      BETA(IN) = DSQRT(SUM)
C
C-----------------------------------------------------------------------
      ISOLV = 4
      CALL LSOLV(V2,V2,ISOLV)
C-----------------------------------------------------------------------
C
      SUM = ONE/BETA(IN)
      DO 70 K = 1,N
      V2(K) = SUM*V2(K)
   70 VS(K) = SUM*VS(K)
C
   80 CONTINUE
C
      RETURN
C-----END LANCZS--------------------------------------------------------
      END
C
C-----USPEC (GENERAL SYMMETRIC SPARSE MATRICES)-------------------------
C
C     SUBROUTINE USPECA(N,MATNOA)
      SUBROUTINE GUSPEC(N,MATNOA)
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION  ASD(10000),AD(5010)
      INTEGER  IROW(10000),ICOL(5010)
C-----------------------------------------------------------------------
C    USPEC DIMENSIONS AND INITIALIZES THE ARRAYS NEEDED TO DEFINE
C    THE USER-SPECIFIED A-MATRIX AND THEN PASSES THE STORAGE LOCATIONS
C    OF THESE ARRAYS TO THE MULTIPLY SUBROUTINE AMATV.
C
C    MATRIX IS STORED IN FOLLOWING SPARSE MATRIX FORMAT:
C    N = ORDER OF A-MATRIX,
C    NZS = NUMBER OF NONZERO SUBDIAGONAL ENTRIES,
C    NZL = INDEX OF LAST COLUMN CONTAINING NONZERO SUBDIAGONAL ENTRIES,
```

```
C    ICOL(J), J=1,NZL IS THE NUMBER OF NONZERO SUBDIAGONAL ELEMENTS     LGM01550
C             IN COLUMN J.                                              LGM01560
C    IROW(K), K = 1,NZS IS THE CORRESPONDING ROW INDEX FOR ASD(K).      LGM01570
C    AD(I), I=1,N CONTAINS DIAGONAL ENTRIES (INCLUDING ANY 0            LGM01580
C           DIAGONAL ENTRIES).                                          LGM01590
C    ASD(K), K=1,NZS CONTAINS NONZERO SUBDIAGONAL ENTRIES, BY COLUMN    LGM01600
C    FOR J > NZL THERE ARE NO NONZERO SUBDIAGONAL ELEMENTS IN COLUMN J. LGM01610
C    ICOL(J) = 0 IS ALLOWED                                             LGM01620
C                                                                       LGM01630
C-----------------------------------------------------------------------LGM01640
C     ARRAYS THAT DEFINE THE A-MATRIX ARE READ IN FROM FILE 8.  NOTE    LGM01650
C     THAT IF THE B-MATRIX IS PERMUTED, THEN LANCZOS PROGRAM ASSUMES    LGM01660
C     THAT THE DATA ON FILE 8 CORRESPONDS TO THE CORRESPONDING          LGM01670
C     PERMUTED A-MATRIX.  LANCZOS PROCEDURE WORKS DIRECTLY WITH THE     LGM01680
C     PERMUTED MATRICES.  EIGENVECTOR CODE, LGVEC, THEN PERMUTES THE    LGM01690
C     COMPUTED EIGENVECTORS TO GET THOSE CORRESPONDING TO THE ORIGINAL  LGM01700
C     MATRICES.                                                         LGM01710
C                                                                       LGM01720
      READ(8,10) NZS,NOLD,NZL,MATOLD                                    LGM01730
   10 FORMAT(I10,2I6,I8)                                                LGM01740
C                                                                       LGM01750
      WRITE(6,20) NZS,NOLD,NZL,MATOLD                                   LGM01760
   20 FORMAT(I10,2I6,I8,' = NZS,NOLD,NZL,MATOLD'/)                      LGM01770
C                                                                       LGM01780
C     TEST OF PARAMETER CORRECTNESS                                     LGM01790
      ITEMP = (NOLD-N)**2 + (MATNOA-MATOLD)**2                          LGM01800
C                                                                       LGM01810
      IF(ITEMP.EQ.0) GO TO 40                                           LGM01820
C                                                                       LGM01830
      WRITE(6,30)                                                       LGM01840
   30 FORMAT(' PROGRAM TERMINATES BECAUSE EITHER ORDERS OF OR LABELS FORLGM01850
     1 MATRIX DISAGREE')                                                LGM01860
      GO TO 70                                                          LGM01870
C                                                                       LGM01880
   40 CONTINUE                                                          LGM01890
C                                                                       LGM01900
C     NUMBER OF NONZERO SUBDIAGONAL ENTRIES IN EACH COLUMN IS READ      LGM01910
C     THEN THE CORRESPONDING ROW INDEX FOR EACH SUCH ENTRY IS READ      LGM01920
      READ(8,50) (ICOL(K), K=1,NZL)                                     LGM01930
      READ(8,50) (IROW(K), K=1,NZS)                                     LGM01940
   50 FORMAT(13I6)                                                      LGM01950
C                                                                       LGM01960
C     DIAGONAL IS READ FIRST, THEN NONZERO BELOW DIAGONAL ENTRIES       LGM01970
      READ(8,60) (AD(K), K=1,N)                                         LGM01980
      READ(8,60) (ASD(K), K=1,NZS)                                      LGM01990
   60 FORMAT(4E19.10)                                                   LGM02000
C                                                                       LGM02010
C-----------------------------------------------------------------------LGM02020
C     PASS STORAGE LOCATIONS OF ARRAYS THAT DEFINE THE A-MATRIX TO      LGM02030
C     THE MATRIX-VECTOR MULTIPLY SUBROUTINE AMATV                       LGM02040
      CALL AMATVE(ASD,AD,ICOL,IROW,N,NZL)                               LGM02050
C-----------------------------------------------------------------------LGM02060
C                                                                       LGM02070
      RETURN                                                            LGM02080
   70 STOP                                                              LGM02090
```

```
C-----END OF USPECA-----------------------------------------------------
      END
C
C-----USPECB FOR CHOLESKY FACTORS OF GENERAL SPARSE SYMMETRIC MATRIX----
C
C     SUBROUTINE USPECB(N,MATNOB)
      SUBROUTINE CUSPEC(N,MATNOB)
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION  BD(2200),BSD(10000)
      INTEGER  KCOL(2200),KROW(10000),IPR(2200),IPT(2200)
C-----------------------------------------------------------------------
C     DIMENSIONS ARRAYS NEEDED TO DEFINE CHOLESKY FACTOR OF B-MATRIX,
C     READS CHOLESKY FACTOR FROM FILE 7, AND THEN PASSES STORAGE
C     LOCATIONS OF THESE ARRAYS TO THE MATRIX SOLVE SUBROUTINE LSOLV
C
C     THE LANCZOS PROCEDURE LGVAL WILL USE THE CHOLESKY FACTORS ON
C     FILE 7.  THESE FACTORS MAY CORRESPOND TO A PERMUTED VERSION OF
C     THE GIVEN B-MATRIX IN WHICH CASE THIS PERMUTATION WILL BE STORED
C     IN IPR. THE ITH ROW OF THE PERMUTED B WILL CORRESPOND TO THE
C     JTH ROW OF B WHERE J = IPR(I) AND I = IPT(J).  IF B IS
C     PERMUTED, THE LANCZOS PROCEDURE ASSUMES THAT THE USER-PROVIDED
C     A-MATRIX IS IN FACT, THE CORRESPONDING PERMUTED VERSION OF THE
C     ORIGINAL A-MATRIX.
C
C     THE CHOLESKY FACTOR IS STORED IN THE FOLLOWING SPARSE FORMAT:
C     N = ORDER OF THE B-MATRIX.
C     NZT = NUMBER OF NONZERO SUBDIAGONAL ENTRIES IN THE CHOLESKY
C           FACTOR, L.
C     KCOL(J), J=1,N IS THE NUMBER OF NONZERO SUBDIAGONAL ELEMENTS IN
C              COLUMN J OF L.
C     KROW(K), K=1,NZT IS THE ROW INDEX FOR CORRESPONDING ENTRY BSD(K).
C     BD(J), J = 1,N CONTAINS THE DIAGONAL ENTRIES OF L.
C     BSD(K), K =1,NZT CONTAINS THE NONZERO SUBDIAGONAL ENTRIES OF L
C             BY COLUMN.
C-----------------------------------------------------------------------
C
C     READ CHOLESKY FACTOR FROM FILE 7.  MUST BE STORED
C     IN SPARSE MATRIX FORMAT.
      READ(7,10) NZT,NOLD,NZL,MATOLD,JPERM
   10 FORMAT(I10,2I6,I8,I6)
C
      WRITE(6,20) NZT,NZL,N,NOLD,MATOLD,JPERM
   20 FORMAT(' HEADER, CHOLESKY FACTOR FILE'/
     1 3X,'NZT',3X,'NZL',5X,'N',2X,'NOLD',2X,'MATOLD',1X,'JPERM'/
     1 4I6,I8,I6/)
C
      IF (N.NE.NOLD.OR.MATNOB.NE.MATOLD) GO TO 70
C
      READ(7,30) (KCOL(K), K = 1,NZL)
      READ(7,30) (KROW(K), K = 1,NZT)
   30 FORMAT(13I6)
      READ(7,40) (BD(K), K = 1,N)
      READ(7,40) (BSD(K), K = 1,NZT)
   40 FORMAT(4Z20)
```

```
C  20 FORMAT(3E25.16)
C
      IF(JPERM.EQ.0)  GO TO 60
C
      READ(7,30) (IPR(K), K = 1,N)
      DO 50 K = 1,N
      J = IPR(K)
   50 IPT(J) = K
C
C-----------------------------------------------------------------------
      CALL LPERME(IPR,IPT,N)
C-----------------------------------------------------------------------
C
   60 CONTINUE
C
C-----------------------------------------------------------------------
C     PASS STORAGE LOCATIONS OF FACTORS TO SUBROUTINE LSOLV
      CALL LSOLVE(BSD,BD,KCOL,KROW,N,NZT,NZL)
C-----------------------------------------------------------------------
C
      GO TO 90
C
   70 CONTINUE
C     DEFAULT EXIT
      WRITE(6,80) MATNOB,MATOLD
   80 FORMAT(' TERMINATE.  PARAMETERS IN CHOLESKY FACTOR FILE'/
     1' DO NOT AGREE WITH THOSE SPECIFIED BY THE USER'/
     1' MATNOB = ',I8,' MATOLD = ',I8/)
      STOP
C
   90 CONTINUE
C-----END OF USPECB-----------------------------------------------------
      RETURN
      END
C
C-----MATRIX-VECTOR MULTIPLY FOR REAL SPARSE SYMMETRIC MATRICES---------
C
C     SUBROUTINE AMATV(W,U,SUM)
      SUBROUTINE GCMATV(W,U,SUM)
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION   U(1),W(1),ASD(1),AD(1),SUM
      INTEGER  IROW(1),ICOL(1)
C-----------------------------------------------------------------------
C    SPARSE MATRIX-VECTOR MULTIPLY FOR LANCZS  U = A*W - SUM*U
C    SEE USPECA SUBROUTINE FOR DESCRIPTION OF THE ARRAYS
C    THAT DEFINE THE A-MATRIX
C-----------------------------------------------------------------------
      GO TO 3
C     STORAGE LOCATIONS OF ARRAYS ARE PASSED TO AMATV FROM USPECA
      ENTRY AMATVE(ASD,AD,ICOL,IROW,N,NZL)
      GO TO 4
C-----------------------------------------------------------------------
C
C     COMPUTE THE DIAGONAL TERMS
```

```
    3 DO 10 I = 1,N                                                     LGM03200
   10 U(I) = AD(I)*W(I)-SUM*U(I)                                        LGM03210
C                                                                       LGM03220
C     COMPUTE BY COLUMN                                                 LGM03230
      LLAST = 0                                                         LGM03240
      DO 30 J = 1,NZL                                                   LGM03250
C                                                                       LGM03260
      IF (ICOL(J).EQ.0) GO TO 30                                        LGM03270
      LFIRST = LLAST + 1                                                LGM03280
      LLAST = LLAST + ICOL(J)                                           LGM03290
C                                                                       LGM03300
      DO 20 L = LFIRST,LLAST                                            LGM03310
      I = IROW(L)                                                       LGM03320
C                                                                       LGM03330
      U(I) = U(I) + ASD(L)*W(J)                                         LGM03340
      U(J) = U(J) + ASD(L)*W(I)                                         LGM03350
C                                                                       LGM03360
   20 CONTINUE                                                          LGM03370
C                                                                       LGM03380
   30 CONTINUE                                                          LGM03390
C                                                                       LGM03400
    4 RETURN                                                            LGM03410
C                                                                       LGM03420
C-----END OF AMATV----------------------------------------------------- LGM03430
      END                                                               LGM03440
C                                                                       LGM03450
C-----LSOLV-GENERAL SPARSE, POSITIVE DEFINITE B-MATRIX------------------LGM03460
C     (USES THE CHOLESKY FACTORS OF B, B = L*(L-TRANSPOSE))             LGM03470
C                                                                       LGM03480
      SUBROUTINE TLSOLV(W,U,ISOLV)                                      LGM03490
C     SUBROUTINE LSOLV(W,U,ISOLV)                                       LGM03500
C                                                                       LGM03510
C-----------------------------------------------------------------------LGM03520
      DOUBLE PRECISION  U(1),W(1),BD(1),BSD(1), TEMP                    LGM03530
      INTEGER  KCOL(1),KROW(1)                                          LGM03540
C-----------------------------------------------------------------------LGM03550
C     SUBROUTINE HAS 4 BRANCHES:  ISOLV = (1,2,3,4)  CALCULATES         LGM03560
C     ISOLV = 1   U = L*W                                               LGM03570
C     ISOLV = 2   U = L'*W                                              LGM03580
C     ISOLV = 3   SOLVE FOR U IN  L*U = W                               LGM03590
C     ISOLV = 4   SOLVE FOR U IN  L'*U = W                              LGM03600
C-----------------------------------------------------------------------LGM03610
      GO TO 3                                                           LGM03620
      ENTRY LSOLVE(BSD,BD,KCOL,KROW,N,NZT,NZL)                          LGM03630
      GO TO 4                                                           LGM03640
C-----------------------------------------------------------------------LGM03650
    3 GO TO (10,50,80,120), ISOLV                                       LGM03660
C                                                                       LGM03670
C     ISOLV = 1,  U=L*W                                                 LGM03680
   10 CONTINUE                                                          LGM03690
      KL = 0                                                            LGM03700
      DO 20 K = 1,N                                                     LGM03710
   20 U(K) = W(K)*BD(K)                                                 LGM03720
      DO 40 K = 1,N                                                     LGM03730
      TEMP = W(K)                                                       LGM03740
```

```
      IF (KCOL(K).EQ.0.OR.K.EQ.N) GO TO 40                              LGM03750
      KF = KL + 1                                                       LGM03760
      KL = KL + KCOL(K)                                                 LGM03770
      DO 30 KK = KF,KL                                                  LGM03780
      KR = KROW(KK)                                                     LGM03790
   30 U(KR) = U(KR) + TEMP*BSD(KK)                                      LGM03800
   40 CONTINUE                                                          LGM03810
      GO TO 150                                                         LGM03820
C                                                                       LGM03830
C     ISOLV = 2,  U = (L-TRANSPOSE)*W                                   LGM03840
   50 CONTINUE                                                          LGM03850
      KL = 0                                                            LGM03860
      DO 70 J = 1,N                                                     LGM03870
      TEMP = W(J)*BD(J)                                                 LGM03880
      IF (KCOL(J).EQ.0.OR.J.EQ.N) GO TO 70                              LGM03890
      KF = KL + 1                                                       LGM03900
      KL  = KL + KCOL(J)                                                LGM03910
      DO 60 K = KF,KL                                                   LGM03920
      IK = KROW(K)                                                      LGM03930
   60 TEMP = BSD(K)*W(IK) + TEMP                                        LGM03940
   70 U(J) = TEMP                                                       LGM03950
      GO TO 150                                                         LGM03960
C                                                                       LGM03970
C     ISOLV = 3,  U = (L-INVERSE)*W                                     LGM03980
   80 CONTINUE                                                          LGM03990
      DO 90 K = 1,N                                                     LGM04000
   90 U(K) = W(K)                                                       LGM04010
      KL = 0                                                            LGM04020
      DO 110 K = 1,N                                                    LGM04030
      TEMP = U(K)/BD(K)                                                 LGM04040
      U(K) = TEMP                                                       LGM04050
      IF (KCOL(K).EQ.0.OR.K.EQ.N) GO TO 110                             LGM04060
      KF = KL + 1                                                       LGM04070
      KL = KL + KCOL(K)                                                 LGM04080
      DO 100 KK = KF,KL                                                 LGM04090
      KR = KROW(KK)                                                     LGM04100
  100 U(KR) = U(KR) - TEMP*BSD(KK)                                      LGM04110
  110 CONTINUE                                                          LGM04120
      GO TO 150                                                         LGM04130
C                                                                       LGM04140
C     ISOLV = 4,  U = (L-TRANSPOSE)-INVERSE*W                           LGM04150
  120 CONTINUE                                                          LGM04160
      NP1 = N+1                                                         LGM04170
      KF = NZT + 1                                                      LGM04180
      DO 140 K = 1,N                                                    LGM04190
      L = NP1 - K                                                       LGM04200
      TEMP = W(L)                                                       LGM04210
      IF (KCOL(L).EQ.0.OR.L.EQ.N) GO TO 140                             LGM04220
      KL = KF - 1                                                       LGM04230
      KF = KF - KCOL(L)                                                 LGM04240
      DO 130 LL = KF,KL                                                 LGM04250
      LR = KROW(LL)                                                     LGM04260
  130 TEMP = TEMP - BSD(LL)*U(LR)                                       LGM04270
  140 U(L) = TEMP/BD(L)                                                 LGM04280
      GO TO 150                                                         LGM04290
```

```
  150 CONTINUE
C
    4 RETURN
C
C-----END OF LSOLV--------------------------------------------------
      END
C
C-----START OF USPEC FOR DIAGONAL TEST A-MATRIX------------------------
C
      SUBROUTINE USPECA(N,MATNO)
C     SUBROUTINE DUSPEC(N,MATNO)
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION  D(1000), SPACE, SHIFT
      DOUBLE PRECISION  DABS, DFLOAT
      REAL  EXPLAN(20)
C-----------------------------------------------------------------------
C
      READ(8,10) EXPLAN
   10 FORMAT(20A4)
      READ(8,*) NOLD,NUNIF,SPACE,D(1),SHIFT
      NNUNIF = NOLD - NUNIF
      WRITE(6,20) NOLD,SPACE,NNUNIF,D(1),SHIFT
   20 FORMAT(/' DIAGONAL TEST A-MATRIX, SIZE = ',I4/' MOST ENTRIES ARE '
     1,E10.3,' UNITS APART.',I3,' ENTRIES'/' ARE IRREGULARLY SPACED. FIR
     1ST ENTRY IS ',E10.3,' SHIFT = ',E10.3/)
C
      IF(N.NE.NOLD) GO TO 90
C     COMPUTE THE UNIFORM PORTION OF THE SPECTRUM
      DO 30 J=2,NUNIF
   30 D(J) = D(1) - DFLOAT(J-1)*SPACE
      NUNIF1=NUNIF + 1
      READ(8,10)  EXPLAN
      DO 40 J=NUNIF1,N
   40 READ(8,*) D(J)
      NB = NUNIF - 2
C
      IF(SHIFT.EQ.0.) GO TO 60
      DO 50 J=1,N
   50 D(J) = D(J) + SHIFT
C
C     PRINT OUT A-MATRIX
   60 WRITE(6,70) (D(I), I=1,10 )
      WRITE(6,80) (D(I), I = NB,N)
   70 FORMAT(/' GENERALIZED LANCZOS TEST, 1ST 10 ENTRIES OF DIAGONAL A-M
     1ATRIX = '/(3E22.14))
   80 FORMAT(/' MIDDLE UNIFORM PORTION OF MATRIX IS NOT PRINTED OUT'/
     1' END OF UNIFORM PLUS NONUNIFORM SECTION = '/(3E25.16))
C
C     DIAGONAL GENERATION COMPLETE
C
C-----------------------------------------------------------------------
C    CALL ENTRY TO MATRIX-VECTOR MULTIPLY SUBROUTINE  TO PASS
C    STORAGE LOCATION OF D-ARRAY AND ORDER OF A-MATRIX.
       CALL MVDIAE(D,N)
```

```
C-----------------------------------------------------------------------LGM04850
C                                                                       LGM04860
      RETURN                                                            LGM04870
   90 WRITE(6,100) NOLD,N                                               LGM04880
  100 FORMAT(' PROGRAM TERMINATES BECAUSE NOLD = ',I5,'DOES NOT EQUAL N LGM04890
     1 =',I5)                                                           LGM04900
C-----END OF USPECA SUBROUTINE FOR DIAGONAL TEST MATRICES---------------LGM04910
      STOP                                                              LGM04920
      END                                                               LGM04930
C                                                                       LGM04940
C-----USPECB--DIAGONAL TEST B-MATRIX------------------------------------LGM04950
C                                                                       LGM04960
      SUBROUTINE USPECB(N,MATNO)                                        LGM04970
C     SUBROUTINE USPECD(N,MATNO)                                        LGM04980
C                                                                       LGM04990
C-----------------------------------------------------------------------LGM05000
      DOUBLE PRECISION  D(1000), DS(1000), SPACE, SHIFT                 LGM05010
      DOUBLE PRECISION  DFLOAT, DSQRT                                   LGM05020
      REAL  EXPLAN(20)                                                  LGM05030
C-----------------------------------------------------------------------LGM05040
C                                                                       LGM05050
      READ(7,10) EXPLAN                                                 LGM05060
   10 FORMAT(20A4)                                                      LGM05070
      READ(7,*) NOLD,NUNIF,SPACE,D(1),SHIFT                             LGM05080
      NNUNIF = NOLD - NUNIF                                             LGM05090
      WRITE(6,20) NOLD,SPACE,NNUNIF,D(1),SHIFT                          LGM05100
   20 FORMAT(/' DIAGONAL TEST B-MATRIX, SIZE = ',I4/' MOST ENTRIES ARE 'LGM05110
     1,E10.3,' UNITS APART.',I3,' ENTRIES'/' ARE IRREGULARLY SPACED. FIRLGM05120
     1ST ENTRY IS ',E10.3,' SHIFT = ',E10.3/)                           LGM05130
C                                                                       LGM05140
      IF(N.NE.NOLD) GO TO 100                                           LGM05150
C     COMPUTE THE UNIFORM PORTION OF THE SPECTRUM                       LGM05160
      DO 30 J=2,NUNIF                                                   LGM05170
   30 D(J) = D(1) - DFLOAT(J-1)*SPACE                                   LGM05180
      NUNIF1=NUNIF + 1                                                  LGM05190
      READ(7,10)  EXPLAN                                                LGM05200
      DO 40 J=NUNIF1,N                                                  LGM05210
   40 READ(7,*) D(J)                                                    LGM05220
      NB = NUNIF - 2                                                    LGM05230
C                                                                       LGM05240
      IF(SHIFT.EQ.0.) GO TO 60                                          LGM05250
      DO 50 J=1,N                                                       LGM05260
   50 D(J) = D(J) + SHIFT                                               LGM05270
C                                                                       LGM05280
C     PRINT OUT B-MATRIX                                                LGM05290
   60 WRITE(6,70) (D(I), I=1,10 )                                       LGM05300
      WRITE(6,80) (D(I), I = NB,N)                                      LGM05310
   70 FORMAT(/' GENERALIZED LANCZOS TEST, 1ST 10 ENTRIES OF DIAGONAL B-MLGM05320
     1ATRIX = '/(3E22.14))                                              LGM05330
   80 FORMAT(/' MIDDLE UNIFORM PORTION OF MATRIX IS NOT PRINTED OUT'/   LGM05340
     1' END OF UNIFORM PLUS NONUNIFORM SECTION = '/(3E25.16))           LGM05350
C                                                                       LGM05360
C     DIAGONAL GENERATION COMPLETE                                      LGM05370
C                                                                       LGM05380
      DO 90 K = 1,N                                                     LGM05390
```

```
   90 DS(K) = DSQRT(D(K))
C
C-----------------------------------------------------------------------
C     PASS STORAGE LOCATION OF THE L-FACTOR (THE DS-ARRAY) AND ORDER OF
C     B-MATRIX TO LSOLV SUBROUTINE.
        CALL DSOLVE(DS,N)
C-----------------------------------------------------------------------
C
      RETURN
  100 WRITE(6,110) NOLD,N
  110 FORMAT(' PROGRAM TERMINATES BECAUSE NOLD = ',I5,'DOES NOT EQUAL N
     1 =',I5)
C-----END OF USPECB SUBROUTINE FOR DIAGONAL TEST MATRICES---------------
      STOP
      END
C
C-----MATRIX-VECTOR MULTIPLY FOR DIAGONAL TEST MATRICES-----------------
C
      SUBROUTINE AMATV(W,U,SUM)
C     SUBROUTINE DCMATV(W,U,SUM)
C
C     AMATV COMPUTES  U = (DIAGONAL MATRIX) * W - SUM * U
C-----------------------------------------------------------------------
      DOUBLE PRECISION  W(1),U(1),D(1),SUM
C-----------------------------------------------------------------------
      GO TO 3
      ENTRY MVDIAE(D,N)
      GO TO 4
C-----------------------------------------------------------------------
C
    3 DO 10 I=1,N
   10 U(I)= D(I)*W(I) - SUM*U(I)
C
    4 RETURN
C
C-----END OF DIAGONAL TEST MATRIX  MULTIPLY-----------------------------
      END
C
C-----LSOLV FOR DIAGONAL MATRIX-----------------------------------------
C
      SUBROUTINE LSOLV(W,U,ISOLV)
C     SUBROUTINE DSOLV(W,U,ISOLV)
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION U(1), W(1),  DS(1)
C-----------------------------------------------------------------------
      GO TO 3
      ENTRY DSOLVE(DS,N)
      GO TO 4
C-----------------------------------------------------------------------
    3 GO TO (10,30,50,70), ISOLV
C
C     ISOLV = 1
   10 CONTINUE
      DO 20 K = 1,N
```

```
   20 U(K) = DS(K)*W(K)                                                 LGM05950
      GO TO 90                                                          LGM05960
C                                                                       LGM05970
C     ISOLV = 2                                                         LGM05980
   30 CONTINUE                                                          LGM05990
      DO 40 K = 1,N                                                     LGM06000
   40 U(K) = DS(K)*W(K)                                                 LGM06010
      GO TO 90                                                          LGM06020
C                                                                       LGM06030
C     ISOLV = 3                                                         LGM06040
   50 CONTINUE                                                          LGM06050
      DO 60 K = 1,N                                                     LGM06060
   60 U(K) = W(K)/DS(K)                                                 LGM06070
      GO TO 90                                                          LGM06080
C                                                                       LGM06090
C     ISOLV = 4                                                         LGM06100
   70 CONTINUE                                                          LGM06110
      DO 80 K = 1,N                                                     LGM06120
   80 U(K) = W(K)/DS(K)                                                 LGM06130
C                                                                       LGM06140
   90 CONTINUE                                                          LGM06150
C                                                                       LGM06160
    4 RETURN                                                            LGM06170
C                                                                       LGM06180
C-----END OF DSOLV------------------------------------------------------LGM06190
      END                                                               LGM06200
```

5.5 LGVAL: LGVEC: File Definitions, Sample Input Files

Below is a listing of the input/output files which are accessed by the Lanczos eigenvalue program LGVAL for real symmetric generalized problems where one of the two matrices is positive definite. Included also is a sample of the input file which LGVAL requires on file 5. The parameters in this file are supplied in free format. LGVAL computes eigenvalues of the matrix eigenvalue problem $Ax = \lambda Bx$ on user-specified intervals. It is assumed that A and B are real symmetric matrices and that B is positive definite. The program uses Cholesky Factor L of $B = LL^T$.

```
Sample Specification of Input/Output Files for LGVAL
-----------------------------------------------------------------
 LGVAL EXEC LANCZOS EIGENVALUE CALCULATION AX = EV*BX CASE
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 LGVAL     INPUT     A (RECFM F LRECL 80 BLOCK 80
FILEDEF  7 DISK &1        LDATA     A (RECFM F LRECL 80 BLOCK 80
FILEDEF  8 DISK &1        ADATA     A (RECFM F LRECL 80 BLOCK 80
FILEDEF 11 DISK &1        DISTINCT  A (RECFM F LRECL 80 BLOCK 80
LOAD  LGVAL   LESUB  LGMULT
-----------------------------------------------------------------
```

```
Sample Input File for LGVAL
--------------------------------------------------------------------
 LGVAL INPUT LANCZOS EIGENVALUE COMPUTATION, NO REORTHOGONALIZATION
 AX = EV*BX GENERALIZED EIGENVALUE PROBLEM
LINE 1      N    KMAX    NMEVS     MATNOA    MATNOB
         100     300        1        100       100
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) )
           300
LINE 7    NINT     (NUMBER OF SUB-INTERVALS FOR BISEC)
             1
LINE 8    LB(1)    LB(2)   LB(3)   LB(4)  (INTERVAL LOWER BOUNDS)
            1.5
LINE 9    UB(1)    UB(2)   UB(3)   UB(4)  (INTERVAL UPPER BOUNDS)
          2100.
--------------------------------------------------------------------
```

Below is a listing of the input/output files which are accessed by the Lanczos eigenvector program for real symmetric generalized problems, LGVEC. Also included below is a sample of the input file which LGVEC requires on file 5. The parameters in this file are supplied in free format. LGVEC computes eigenvectors for each of a user-specified subset of the eigenvalues computed by the companion program LGVAL.

```
Sample Specifications for the Input/Output Files for LGVEC
-----------------------------------------------------------------------
 LGVEC EXEC TO RUN LANCZOS EIGENVECTOR PROGRAM, REAL SYMMETRIC MATRICES
FI 06 TERM
FILEDEF  2 DISK &1       HISTORY   A (RECFM F LRECL 80 BLOCK 80
FILEDEF  3 DISK &1       GOODEV    A (RECFM F LRECL 80 BLOCK 80
FILEDEF  4 DISK &1       ERRINV    A (RECFM F LRECL 80 BLOCK 80
FILEDEF  5 DISK LGVEC    INPUT     A (RECFM F LRECL 80 BLOCK 80
FILEDEF  7 DISK &1       LDATA     A (RECFM F LRECL 80 BLOCK 80
FILEDEF  8 DISK &1       ADATA     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  LGVEC  LESUB  LGMULT
-----------------------------------------------------------------------
```

```
Sample Input File for LGVEC
-----------------------------------------------------------------------
 LGVEC EIGENVECTOR COMPUTATIONS AX = EV*BX NO REORTHOGONALIZATION
LINE 1  MDIMTV    MDIMRV  MBETA (MAX.DIMENSIONS, TVEC, RITVEC AND BETA
         10000     10000   2000
LINE 2       RELTOL
        .0000000001
LINE 3  MBOUND   NTVCON SVTVEC IREAD (FLAGS
             0        1      0     1
LINE 4  TVSTOP   LVCONT ERCONT  IWRITE (FLAGS
             0        1      1       1
LINE 5    RHSEED  (RANDOM GENERATOR SEED FOR STARTING VECTOR IN INVERM)
        45329517
LINE 6  MATNOA     MATNOB      N    JPERM
           100        100    100        0
------------------------------------------------------------------------
```