Chapter 2

Real Symmetric Matrices

2.1 Introduction

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

$$Ax = \lambda x. \tag{2.1.1}$$

Definition 1 *The real $n \times n$ matrix $A \equiv (a_{ij})$, $1 \leq i,j \leq n$, is a real symmetric matrix if and only if for every i,j, $a_{ij} = a_{ji}$.*

Real symmetric matrices are discussed in detail in Stewart [24]. Properties which we use are:

1. Real symmetric matrices have complete eigensystems. That is, the dimension of the eigenspace corresponding to any given eigenvalue of the given matrix A is the same as the multiplicity of that eigenvalue as a root of the characteristic polynomial of A.
2. For any two distinct eigenvalues of A, λ and μ , and corresponding eigenvectors x and y, $x^T y = 0$. Thus, eigenvectors corresponding to different eigenvalues are orthogonal, and we can construct an eigenvector basis which is orthonormal. Vectors are orthonormal if they are orthogonal and each has a Euclidean norm of 1. (The Euclidean norm of a vector is just the square root of the sum of the squares of its components.)
3. Small perturbations in the matrix cause only small perturbations in the eigenvalues. Of the classes of matrices which we consider, the class of real symmetric matrices is the most well-behaved and thus the 'easiest'.

The Lanczos codes contained in this chapter correspond to the most straight-forward implementation of the Lanczos recursion included in this book. These codes can be used to compute either a very few or very many of the distinct eigenvalues of the given real symmetric matrix. As the documentation in the next section indicates, the A-multiplicity of a given computed 'good' Lanczos eigenvalue can be obtained only with additional computation, and the modifications required to do this additional computation are not included in these versions of the codes. This implementation uses the basic Lanczos recursion given in Eqns (1.2.1) and (1.2.2) in Section 1.2 of Chapter 1 to generate a family of real symmetric, tridiagonal matrices (T-matrices) whose sizes are specified by the user. There is no reorthogonalization of the Lanczos vectors at any stage in any of the computations.

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

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

All computations are in double precision real arithmetic. The user must supply a subroutine USPEC which defines and initializes the user-specified matrix A and a subroutine CMATV which computes matrix-vector multiplies Ax for any given vector x. These subroutines must be constructed in such a way as to take advantage of the sparsity (and/or structure) of the user-supplied A-matrix and such that these computations are done accurately. More details about these real symmetric single-vector Lanczos procedures are given in Chapter 4 of Volume 1 of this book.

2.2 Documentation for the Codes in Chapters 2, 3, 4, 5

```
C-----LEVALHED----------------------------------------------------------
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-----DOCUMENTATION FOR SINGLE-VECTOR-----------------------------------
C
C     LANCZOS EIGENVALUE/EIGENVECTOR PROGRAMS FOR
C     (1) REAL SYMMETRIC MATRICES
C     (2) HERMITIAN MATRICES
C     (3) FACTORED INVERSES OF REAL SYMMETRIC MATRICES
C     (4) REAL SYMMETRIC, GENERALIZED PROBLEMS WHERE ONE OF THE
C         MATRICES IS POSITIVE DEFINITE AND ITS CHOLESKY FACTORS ARE
C         AVAILABLE
C
C
C
C-----------------------------------------------------------------------
C
C     REAL SYMMETRIC MATRICES:
C
C     GIVEN A REAL SYMMETRIC MATRIX A OF ORDER N THE THREE SETS OF
C     FORTRAN FILES LABELLED LEVAL, LESUB, AND LEMULT CAN BE USED TO
C     COMPUTE DISTINCT EIGENVALUES OF THE USER-SPECIFIED MATRIX
C     IN USER-SPECIFIED INTERVALS.
C
C     CORRESPONDING EIGENVECTORS FOR SELECTED, COMPUTED EIGENVALUES CAN
C     BE COMPUTED USING THE SETS OF FILES LABELLED LEVEC, LESUB, AND
C     LEMULT.
C
```

```
C
C     HERMITIAN MATRICES:
C
C     GIVEN A HERMITIAN MATRIX A OF ORDER N THE THREE SETS OF
C     FORTRAN FILES LABELLED HLEVAL, LESUB, AND HLEMULT CAN BE USED
C     TO COMPUTE DISTINCT EIGENVALUES IN USER-SPECIFIED INTERVALS.
C
C     CORRESPONDING EIGENVECTORS FOR SELECTED, COMPUTED EIGENVALUES
C     CAN BE COMPUTED USING THE SETS OF PROGRAMS LABELLED HLEVEC,
C     LESUB, AND HLEMULT.
C
C
C     FACTORED INVERSES OF REAL SYMMETRIC MATRICES:
C
C     GIVEN A REAL SYMMETRIC MATRIX A, THE LANCZOS RECURSION IS
C     APPLIED TO THE INVERSE OF A, USING A FACTORIZATION
C     OF A.  THE SETS OF FILES LIVAL, LESUB, AND LIMULT
C     CAN BE USED TO COMPUTE THE DISTINCT EIGENVALUES OF THE
C     INVERSE OF THE A-MATRIX AND OF A IN  USER-SPECIFIED
C     INTERVALS.  THE PROGRAMS ACTUALLY ALLOW ONE TO WORK WITH
C     ANY MATRIX B = PCP' WHERE  C = S0*A + SHIFT*I, WHERE
C     S0 AND SHIFT ARE SCALARS CHOSEN BY THE USER AND P IS A
C     PERMUTATION MATRIX CHOSEN SUCH THAT THE FACTORIZATION
C     OF THE B-MATRIX RETAINS SPARSITY.  IN THE
C     SAMPLE LIMULT SUBROUTINES PROVIDED, S0 AND SHIFT MUST BE
C     CHOSEN SO THAT THE RESULTING B-MATRIX IS POSITIVE DEFINITE,
C     AND THE CHOLESKY FACTORS ARE USED TO SOLVE B*U = V.
C     HOWEVER, THE USER CAN EASILY REPLACE THE SAMPLE USPEC AND
C     BSOLV SUBROUTINES PROVIDED BY SUBROUTINES THAT ALLOW THE
C     GENERAL FACTORIZATION L*D*(L-TRANSPOSE).  THESE LANCZOS
C     PROGRAMS APPLY THE LANCZOS RECURSION TO B-INVERSE, USING
C     THE FACTORIZATION PROVIDED.  OPTIONAL PREPROCESSING PROGRAMS
C     PERMUT, LORDER, LFACT, AND LTEST ARE PROVIDED FOR SET-UP PURPOSES.
C     PERMUT USES THE SPARSPAK PACKAGE OF A. GEORGE, J. LIU AND
C     E. NG TO OBTAIN A REORDERING OF THE GIVEN MATRIX THAT
C     PRESERVES SPARSENESS ON SUBEQUENT FACTORIZATION.  LORDER
C     CAN BE USED TO REORDER A GIVEN MATRIX, USING A GIVEN
C     PERMUTATION.  LFACT CAN BE USED TO COMPUTE THE CHOLESKY
C     FACTORS OF A GIVEN POSITIVE DEFINITE B-MATRIX.  LTEST CAN
C     BE USED TO ESTIMATE THE NUMERICAL CONDITION OF THE
C     B-MATRIX.
C
C     CORRESPONDING EIGENVECTORS FOR SELECTED, COMPUTED
C     EIGENVALUES CAN BE COMPUTED USING THE SETS OF FILES
C     LABELLED LIVEC, LESUB, AND LIMULT.
C
C     GENERALIZED REAL SYMMETRIC PROBLEMS:
C
C     GIVEN 2 REAL SYMMETRIC MATRICES A AND B WHERE IN ADDITION B IS
C     POSITIVE DEFINITE AND ITS CHOLESKY FACTORS ARE AVAILABLE,
C     THE SETS OF FILES LGVAL, LGMULT, AND LESUB CAN BE USED
C     TO COMPUTE THE DISTINCT EIGENVALUES OF THE GENERALIZED
C     PROBLEM A*X = EVAL*B*X.
C
C     CORRESPONDING EIGENVECTORS CAN BE COMPUTED USING THE PROGRAMS
```

```
C     LGVEC, LGMULT, AND LESUB.  NOTE THAT THE PREPROCESSING PROGRAMS
C     AVAILABLE FOR USE IN CASE (3) (PERMUT, LORDER, LFACT, AND LTEST)
C     CAN ALSO BE USED IN THIS CASE TO OBTAIN A SUITABLE PERMUTATION,
C     AND A FACTORIZATION OF THE RESULTING B-MATRIX.  THE A-MATRIX
C     CAN THEN BE PERMUTED USING LORDER.
C
C
C     THESE PROGRAMS ALL USE LANCZOS TRIDIAGONALIZATION WITHOUT
C     REORTHOGONALIZATION TO GENERATE REAL SYMMETRIC TRIDIAGONAL
C     MATRICES, T(1,MEV), OF ORDER MEV.  SUBSETS OF THE EIGENVALUES OF
C     THESE T-MATRICES, LABELLED AS THE 'GOOD EIGENVALUES', YIELD
C     APPROXIMATIONS TO THE DESIRED EIGENVALUES.  CORRESPONDING
C     RITZ VECTORS ARE APPROXIMATIONS TO THE DESIRED EIGENVECTORS.
C     NOTE THAT FOR CASE (4) THE GENERALIZED LANCZOS RECURSION
C     B*V(I+1)*BETA(I+1) = A*V(I) - B*V(I)*ALPHA(I) - B*V(I-1)*BETA(I)
C     IS USED, ALONG WITH THE B-NORM.
C
C     THE IDEAS USED IN THESE PROGRAMS ARE DISCUSSED IN THE FOLLOWING
C     REFERENCES.
C
C     1.  JANE CULLUM AND RALPH A. WILLOUGHBY, LANCZOS ALGORITHMS
C         FOR LARGE SYMMETRIC MATRICES, VOLUME ?, PROGRESS IN
C         SCIENTIFIC COMPUTING, EDITORS, G. GOLUB, H.O. KREISS,
C         S. ARBARBANEL, AND R. GLOWINSKI,  BIRKHAUSER BOSTON INC.,
C         CAMBRIDGE, MASSACHUSETTS, 1983.
C
C     2.  JANE CULLUM AND RALPH A. WILLOUGHBY, COMPUTING EIGENVECTORS
C         (AND EIGENVALUES) OF LARGE, SYMMETRIC MATRICES USING
C         LANCZOS TRIDIAGONALIZATION, LECTURE NOTES IN MATHEMATICS,
C         773, NUMERICAL ANALYSIS PROCEEDINGS, DUNDEE 1979, EDITED BY
C         G. A. WATSON, SPRINGER-VERLAG, (1980), BERLIN, PP.46-63.
C
C     3. IBID, LANCZOS AND THE COMPUTATION IN SPECIFIED INTERVALS OF
C        THE SPECTRUM OF LARGE SPARSE, REAL SYMMETRIC MATRICES, SPARSE
C        MATRIX PROCEEDINGS 1978, ED. I.S. DUFF AND G. W. STEWART,
C        SIAM, PHILADELPHIA, PP.220-255, 1979.
C
C     4. IBID, COMPUTING EIGENVALUES OF VERY LARGE SYMMETRIC MATRICES-
C        AN IMPLEMENTATION OF A LANCZOS ALGORITHM WITHOUT
C        REORTHOGONALIZATION, J. COMPUT. PHYS. 44(1981), 329-358.
C
C
C-----PORTABILITY------------------------------------------------------
C
C
C     PROGRAMS WERE TESTED FOR PORTABILITY USING THE PFORT VERIFIER.
C     FOR DETAILS OF THE VERIFIER SEE FOR EXAMPLE, B. G. RYDER AND
C     A. D. HALL, "THE PFORT VERIFIER", COMPUTING SCIENCE TECHNICAL
C     REPORT 12, BELL LABORATORIES, MURRAY HILL, NEW JERSEY 07974,
C     (REVISED), JANUARY 1981.
C
C     WITH THE EXCEPTION OF THE PROGRAMS FOR HERMITIAN MATRICES WHICH
C     ARE NOT PORTABLE BECAUSE OF THEIR USE OF COMPLEX*16 VARIABLES,
C     THE OTHER PROGRAMS INCLUDED ARE PORTABLE EXCEPT FOR A FEW
C     CONSTRUCTIONS WHICH, IF NECESSARY, WILL HAVE TO BE MODIFIED
```

```
C     BY THE USER FOR THE PARTICULAR COMPUTER BEING USED.              LEV01620
C                                                                      LEV01630
C     NONPORTABLE CONSTRUCTIONS:                                       LEV01640
C                                                                      LEV01650
C     REAL SYMMETRIC MATRICES:                                         LEV01660
C     IN LEVAL AND IN LEVEC                                            LEV01670
C         1.  DATA/MACHEP STATEMENT                                    LEV01680
C         2.  ALL READ(5,*) STATEMENTS (FREE FORMAT)                   LEV01690
C         3.  FORMAT(20A4) USED FOR THE EXPLANATORY HEADER ARRAY, EXPLANLEV01700
C         4.  FORMAT(4Z20) USED TO READ AND WRITE ALPHA/BETA FILES.    LEV01710
C     IN LEMULT                                                        LEV01720
C         1.  IN CMATV AND  USPEC THE ENTRY THAT PASSES THE STORAGE    LEV01730
C             LOCATIONS OF THE ARRAYS DEFINING THE USER-SPECIFIED      LEV01740
C             MATRIX.                                                  LEV01750
C         2.  IN THE SAMPLE USPEC PROVIDED:  FREE FORMAT (8,*),        LEV01760
C             THE FORMAT (20A4), AND DATA/MACHEP STATEMENT.            LEV01770
C                                                                      LEV01780
C     HERMITIAN MATRICES:                                              LEV01790
C     IN HLEVAL AND IN HLEVEC                                          LEV01800
C         1.  DATA/MACHEP STATEMENT                                    LEV01810
C         2.  ALL READ(5,*) STATEMENTS (FREE FORMAT)                   LEV01820
C         3.  FORMAT(20A4) USED FOR THE EXPLANATORY HEADER ARRAY, EXPLANLEV01830
C         4.  COMPLEX*16 VARIABLES AND FUNCTIONS SUCH AS DCMPLX.       LEV01840
C         5.  FORMAT (4Z20) USED TO READ AND WRITE ALPHA/BETA FILES.   LEV01850
C    IN HLEMULT                                                        LEV01860
C         1.  IN CMATV AND  USPEC THE ENTRY THAT PASSES THE STORAGE    LEV01870
C             LOCATIONS OF THE ARRAYS DEFINING THE USER-SPECIFIED      LEV01880
C             MATRIX.                                                  LEV01890
C         2.  COMPLEX*16 VARIABLES AND FUNCTIONS SUCH AS DCMPLX.       LEV01900
C         3.  IN THE SAMPLE USPEC PROVIDED:  FREE FORMAT (8,*),        LEV01910
C             THE FORMAT (20A4), AND DATA/MACHEP STATEMENT.            LEV01920
C                                                                      LEV01930
C    FACTORED INVERSES OF REAL SYMMETRIC MATRICES:                     LEV01940
C    IN LIVAL AND IN LIVEC                                             LEV01950
C         1.  DATA/MACHEP STATEMENT                                    LEV01960
C         2.  ALL READ(5,*) STATEMENTS (FREE FORMAT)                   LEV01970
C         3.  FORMAT(20A4) USED FOR THE EXPLANATORY HEADER ARRAY, EXPLANLEV01980
C         4.  FORMAT(4Z20) USED TO READ AND WRITE ALPHA/BETA FILES.    LEV01990
C    IN LIMULT                                                         LEV02000
C         1.  IN USPEC AND BSOLV, THE ENTRIES THAT PASS                LEV02010
C             THE STORAGE LOCATIONS OF THE ARRAYS DEFINING THE         LEV02020
C             USER-SPECIFIED MATRIX.                                   LEV02030
C         2.  IN THE SAMPLE USPEC SUBROUTINES PROVIDED:                LEV02040
C             FORMATS (20A4) AND (4Z20), FREE FORMAT (8,*), AND        LEV02050
C             DATA/MACHEP STATEMENTS.                                  LEV02060
C                                                                      LEV02070
C                                                                      LEV02080
C    GENERALIZED SYMMETRIC PROBLEM, B-MATRIX POSITIVE                  LEV02090
C    DEFINITE AND CHOLESKY FACTORS AVAILABLE:                          LEV02100
C    IN LGVAL AND IN LGVEC                                             LEV02110
C         1.  DATA/MACHEP STATEMENT                                    LEV02120
C         2.  ALL READ(5,*) STATEMENTS (FREE FORMAT)                   LEV02130
C         3.  FORMAT(20A4) USED FOR THE EXPLANATORY HEADER ARRAY, EXPLANLEV02140
C         4.  FORMAT(4Z20) USED TO READ AND WRITE ALPHA/BETA FILES.    LEV02150
C    IN LGMULT                                                         LEV02160
```

```
C        1.  IN USPECA, USPECB, AMATV AND LSOLV THE ENTRIES             LEV02170
C            THAT PASS THE STORAGE LOCATIONS OF THE ARRAYS DEFINING    LEV02180
C            THE USER-SPECIFIED MATRICES.                              LEV02190
C        2.  IN THE SAMPLE USPECA AND USPECB SUBROUTINES PROVIDED:     LEV02200
C            FORMATS (20A4) AND (4Z20), FREE FORMAT (8,*), AND         LEV02210
C            DATA/MACHEP STATEMENTS.                                   LEV02220
C                                                                      LEV02230
C     ALL 4 CASES USE THE FORTRAN FILE LESUB:                          LEV02240
C     IN LESUB ALL STATEMENTS ARE PORTABLE EXCEPT FOR:                 LEV02250
C         (1)  THE ENTRY IN SUBROUTINE LPERM THAT PASSES THE           LEV02260
C              PERMUTATION FROM THE USPEC SUBROUTINE TO LPERM.         LEV02270
C              (THIS IS USED ONLY IN CASES (3) AND (4)).               LEV02280
C         (2)  THE COMPLEX*16 VARIABLES AND FUNCTIONS USED IN          LEV02290
C              SUBROUTINE CINPRD. (THIS IS USED ONLY IN CASE (2)).     LEV02300
C                                                                      LEV02310
C     IN THE COMMENTS BELOW:                                           LEV02320
C                                                                      LEV02330
C     COMPLEX*16 = COMPLEX VARIABLE, 16 BYTES OF STORAGE               LEV02340
C     REAL*8 = REAL VARIABLE, 8 BYTES OF STORAGE                       LEV02350
C     REAL*4 = REAL VARIABLE, 4 BYTES OF STORAGE                       LEV02360
C     INTEGER*4 = INTEGER VARIABLE, 4 BYTES                            LEV02370
C                                                                      LEV02380
C                                                                      LEV02390
C-----MATRIX SPECIFICATION----------------------------------------------LEV02400
C                                                                      LEV02410
C                                                                      LEV02420
C     IN CASES (1) AND (2), SUBROUTINE USPEC IS USED TO SPECIFY THE    LEV02430
C     USER-SUPPLIED A-MATRIX.  SIMILARLY, IN CASE (4) SUBROUTINES      LEV02440
C     USPECA AND USPECB DEFINE THE USER-SUPPLIED A-MATRIX AND B-MATRIX. LEV02450
C     IN CASE (3) ((4)), SUBROUTINE USPECB DEFINES THE FACTORIZATION   LEV02460
C     OF THE MATRIX (B-MATRIX) USED BY THE LANCZOS PROCEDURE.          LEV02470
C     (IN CASE (3) THE A-MATRIX IS NOT USED DIRECTLY.)                 LEV02480
C                                                                      LEV02490
C     IN CASES (1) AND (2), SUBROUTINE CMATV IS A CORRESPONDING        LEV02500
C     MATRIX-VECTOR MULTIPLY SUBROUTINE WHICH SHOULD BE DESIGNED       LEV02510
C     TO TAKE ADVANTAGE OF ANY SPECIAL PROPERTIES OF THE GIVEN         LEV02520
C     MATRIX.  IN CASE (4) THIS SUBROUTINE IS NEEDED FOR THE           LEV02530
C     A-MATRIX AND THUS IS CALLED AMATV.  IN CASES (3) AND (4)         LEV02540
C     SUBROUTINES THAT CAN SOLVE B*U = V, USING A SPARSE               LEV02550
C     FACTORIZATION OF B ARE NEEDED.  THESE SUBROUTINES ARE            LEV02560
C     CALLED RESPECTIVELY, BSOLV AND LSOLV.  IN ALL CASES,             LEV02570
C     ANY MATRIX-VECTOR MULTIPLY AND SOLVE SUBROUTINES USED            LEV02580
C     MUST BE DESIGNED TO COMPUTE RAPIDLY AND ACCURATELY.              LEV02590
C                                                                      LEV02600
C     IN ALL CASES:                                                    LEV02610
C     SUBROUTINE USPEC(A OR B) HAS THE CALLING SEQUENCE                LEV02620
C                                                                      LEV02630
C            CALL USPEC(N,MATNO)                                       LEV02640
C                                                                      LEV02650
C     WHERE N IS THE ORDER OF THE USER-SUPPLIED MATRIX A, AND          LEV02660
C     MATNO IS A <= 8 DIGIT INTEGER USED AS A MATRIX AND               LEV02670
C     TEST IDENTIFICATION NUMBER.  IN ALL CASES THIS (THESE)           LEV02680
C     SUBROUTINE(S) DEFINES (DIMENSIONS) THE ARRAYS REQUIRED           LEV02690
C     TO SPECIFY THE MATRIX (MATRICES IN CASE (4)) THAT WILL BE        LEV02700
C     USED BY THE LANCZS SUBROUTINE.  IN CASES (1) AND (2)             LEV02710
```

```
C     THIS IS THE A-MATRIX; IN CASE (3) THIS IS THE FACTORIZATION       LEV02720
C     OF A SCALED, SHIFTED AND PERMUTED VERSION OF THE                  LEV02730
C     USER-SPECIFIED A-MATRIX.  IN CASE (4) THE A-MATRIX                LEV02740
C     IS SPECIFIED AS WELL AS THE FACTORIZATION OF THE                  LEV02750
C     B-MATRIX.  THIS SUBROUTINE ALSO INITIALIZES THE ARRAYS            LEV02760
C     AND ANY OTHER PARAMETERS NEEDED TO DEFINE THE MATRIX              LEV02770
C     (MATRICES).  THE STORAGE LOCATIONS OF THESE PARAMETERS            LEV02780
C     AND ARRAYS ARE THEN PASSED TO THE MATRIX-VECTOR MULTIPLY          LEV02790
C     SUBROUTINE CMATV IN CASES (1) AND (2), TO THE SUBROUTINE          LEV02800
C     BSOLV IN CASE (3), AND TO THE SUBROUTINES AMATV                   LEV02810
C     AND LSOLV IN CASE (4) VIA ENTRY CALLS.  IN CASES (3) AND (4)      LEV02820
C     WHENEVER A MATRIX HAS BEEN PERMUTED, THERE IS ALSO AN             LEV02830
C     ENTRY INTO THE SUBROUTINE LPERM TO PASS THE LOCATIONS OF          LEV02840
C     THE PERMUTATIONS IPR AND IPRT USED.  SAMPLE USPECS, CMATV,        LEV02850
C     AMATV, BSOLV AND LSOLV SUBROUTINES ARE INCLUDED                   LEV02860
C     IN THE RELEVANT FILES.  THESE SAMPLE PROGRAMS ASSUME THAT         LEV02870
C     THE USER-SUPPLIED A-MATRIX IS STORED ON FILE 8 IN CASES (1),      LEV02880
C     (2), AND (4), AND THAT THE FACTORIZATION OF THE B-MATRIX          LEV02890
C     IS ON FILE 7 IN CASES (3) AND (4).  THE USER SHOULD SEE           LEV02900
C     THE INDIVIDUAL SAMPLE SUBROUTINES FOR MORE DETAILS.               LEV02910
C                                                                       LEV02920
C     IN CASES (1) AND (2):                                             LEV02930
C     SUBROUTINE CMATV HAS THE CALLING SEQUENCE                         LEV02940
C                                                                       LEV02950
C            CALL CMATV(W,U,SUM)                                        LEV02960
C                                                                       LEV02970
C     IN THE REAL SYMMETRIC CASE, U AND W ARE REAL*8 VECTORS            LEV02980
C     AND SUM IS A REAL*8 SCALAR.  IN THE HERMITIAN CASE, U             LEV02990
C     AND W ARE COMPLEX*16 VECTORS AND SUM IS A REAL*8 SCALAR.          LEV03000
C     CMATV CALCULATES U = A*W - SUM*U FOR THE USER-SPECIFIED           LEV03010
C     MATRIX A. ONE OF THE SAMPLE CMATV SUBROUTINES INCLUDED            LEV03020
C     COMPUTES MATRIX-VECTOR MULTIPLIES FOR AN ARBITRARY SPARSE,        LEV03030
C     SYMMETRIC MATRIX STORED IN THE SPARSE FORMAT SPECIFIED IN THE     LEV03040
C     CORRESPONDING SAMPLE USPEC SUBROUTINE.  FOR CASES (1) AND         LEV03050
C     (2) CMATV IS THE SUBROUTINE USED BY THE LANCZS SUBROUTINE         LEV03060
C     THAT GENERATES THE T-MATRICES.  IN CASE (4) SUBROUTINE            LEV03070
C     AMATV HAS THE SAME CALLING SEQUENCE AS CMATV IN CASE (1).         LEV03080
C                                                                       LEV03090
C     IN CASES (3) AND (4):                                             LEV03100
C     ALPHA/BETA HISTORY IS GENERATED USING SPARSE MATRIX INVERSION.    LEV03110
C     IN CASE (3), AT EACH ITERATION OF THE LANCZOS RECURSION           LEV03120
C     GIVEN A FACTORIZATION OF THE MATRIX BEING USED, THE               LEV03130
C     SUBROUTINE BSOLV FOR A GIVEN V, COMPUTES U SUCH THAT B*U = V.     LEV03140
C     THE CALLING SEQUENCE OF BSOLV IS                                  LEV03150
C                                                                       LEV03160
C                   CALL BSOLV(V,U,IBSOLV)                              LEV03170
C                                                                       LEV03180
C     WHEN IBSOLV = 2,  U = (B-INVERSE)*V IS RETURNED.  IN CASE (4),    LEV03190
C     AT EACH ITERATION OF THE GENERALIZED LANCZOS RECURSION BOTH THE   LEV03200
C     SUBROUTINE AMATV AND THE SUBROUTINE LSOLV ARE USED.  THE          LEV03210
C     CALLING SEQUENCE OF LSOLV IS                                      LEV03220
C                                                                       LEV03230
C                   CALL LSOLV(V,U,ISOLV)                               LEV03240
C                                                                       LEV03250
C     WHERE U AND V ARE REAL*8 VECTORS.  LSOLV PERFORMS 4 FUNCTIONS.    LEV03260
```

```
C     LET L DENOTE THE CHOLESKY FACTOR OF THE B-MATRIX USED IN LANCZS.
C     WHEN ISOLV = 1, LSOLV COMPUTES U = L*V.  WHEN ISOLV = 2,
C     LSOLV COMPUTES U = (L-TRANSPOSE)*V.  WHEN ISOLV = 3, LSOLV
C     COMPUTES U = (L-INVERSE)*V.  WHEN ISOLV = 4, LSOLV
C     COMPUTES  U = ((L-TRANSPOSE)-INVERSE)*V.
C
C     SAMPLE PROGRAMS ASSUME THAT THE A-MATRIX (CASES (1),(2),(4))
C     IS ON FILE 8 AND STORED IN THE FOLLOWING SPARSE FORMAT:
C     ICOL(K), K = 1,NZL, NUMBER OF SUBDIAGONAL NONZEROS IN COLUMN K.
C     IROW(K), K = 1,NZS, ROW INDEX OF ASD(K).
C     AD(K), K=1,N, CONTAINS THE DIAGONAL ELEMENTS OF THE A-MATRIX.
C     ASD(K), K=1,NZS  CONTAINS THE SUBDIAGONAL ELEMENTS OF A BY COLUMN.
C     NZS = NUMBER OF NONZERO ELEMENTS BELOW THE DIAGONAL OF A
C     NZL = INDEX OF LAST COLUMN WITH NONZERO SUBDIAGONAL ENTRIES
C     N = ORDER OF THE A-MATRIX.
C
C     NOTE THAT THE OPTIONAL PREPROCESSING PROGRAMS PERMUT AND
C     LORDER ASSUME THAT THE GIVEN MATRIX IS ON FILE 8.  CASES (3)
C     AND (4) ASSUME THAT THE SPARSE FACTORIZATION OF B IS STORED ON
C     FILE 7.  THE SAMPLE BSOLV SUBROUTINE SUPPLIED ASSUMES
C     THAT THE B-MATRIX IS POSITIVE DEFINITE AND THAT ITS CHOLESKY
C     FACTOR IS PROVIDED ON FILE 7, STORED IN SPARSE FORMAT IN
C     ARRAYS BD AND BSD.  THE USER CAN EASILY REPLACE THIS SAMPLE
C     BSOLV SUBROUTINE AND THE CORRESPONDING SAMPLE USPEC
C     SUBROUTINE BY SUBROUTINES THAT DEFINE AND USE A GENERAL
C     FACTORIZATION L*D*(L-TRANSPOSE).
C
C     THE SAMPLE USPEC, CMATV (CASES (1) AND (2)), AMATV (CASE (4)),
C     BSOLV (CASE (3)), AND LSOLV (CASE(4)) MUST BE MODIFIED BY
C     THE USER TO ACCOMODATE THE USER-SPECIFIED MATRIX OR MATRICES.
C
C
C-----MACHEP-------------------------------------------------------------
C
C
C     MACHEP IS A MACHINE DEPENDENT PARAMETER SPECIFYING THE RELATIVE
C     PRECISION OF THE FLOATING POINT ARITHMETIC USED.
C     MACHEP = 2.2 * 10**-16 FOR DOUBLE PRECISION ARITHMETIC ON
C     IBM 370-3081.
C
C     THE USER WILL HAVE TO RESET THIS PARAMETER TO
C     THE CORRESPONDING VALUE FOR THE MACHINE BEING USED.  NOTE THAT
C     IF A MACHINE WITH A MACHINE EPSILON THAT IS MUCH LARGER THAN THE
C     VALUE GIVEN HERE IS BEING USED, THEN THERE COULD BE
C     PROBLEMS WITH THE TOLERANCES.
C
C
C-----SUBROUTINES AND FUNCTIONS USER MUST SUPPLY-------------------------
C
C
C     GENRAN,  FINPRO,  MASK,  USPEC,  AND
C     CASES (1) AND (2),  CMATV:  CASE (3), BSOLV:
C     CASE (4),  AMATV AND LSOLV.
C
C     GENRAN = COMPUTES K PSEUDO-RANDOM NUMBERS AND STORES THEM IN
```

```
C              THE REAL*4 ARRAY, G.  THIS SUBROUTINE IS USED TO         LEV03820
C              GENERATE A STARTING VECTOR FOR THE LANCZOS PROCEDURE     LEV03830
C              IN THE SUBROUTINE LANCZS AND A STARTING RIGHT-HAND SIDE  LEV03840
C              FOR INVERSE ITERATION IN THE SUBROUTINE INVERR.          LEV03850
C                                                                       LEV03860
C              TESTS REPORTED IN THE REFERENCES USED EITHER GGL1 OR     LEV03870
C              GGL2 FROM THE IBM LIBRARY  SLMATH.                       LEV03880
C              THE EXISTING CALLING SEQUENCE IS:                        LEV03890
C                                                                       LEV03900
C                      CALL GENRAN(IIX,G,K).                            LEV03910
C                                                                       LEV03920
C              WHERE IIX =INTEGER SEED, G = REAL*4 ARRAY WHOSE          LEV03930
C              DIMENSION MUST BE >= K.  K RANDOM NUMBERS ARE GENERATED  LEV03940
C              AND PLACED IN G.                                         LEV03950
C                                                                       LEV03960
C     FINPRO = DOUBLE PRECISION FUNCTION WHICH COMPUTES THE INNER       LEV03970
C              PRODUCT OF 2 DOUBLE PRECISION VECTORS OF DIMENSION  N.   LEV03980
C              TESTS REPORTED IN THE REFERENCES USED THE HARWELL        LEV03990
C              LIBRARY SUBROUTINE FM02AD.                               LEV04000
C              EXISTING CALLING SEQUENCE IS                             LEV04010
C                                                                       LEV04020
C                      CALL FINPRO(N,V,J,W,K).                          LEV04030
C                                                                       LEV04040
C              COMPUTES THE INNER PRODUCT OF DIMENSION N OF THE VECTORS LEV04050
C              V AND W.  SUCCESSIVE COMPONENTS OF V AND OF W ARE STORED LEV04060
C              AT LOCATIONS THAT ARE ,RESPECTIVELY, J AND K UNITS APART.LEV04070
C                                                                       LEV04080
C     MASK = MASKS OVERFLOW AND UNDERFLOW.                              LEV04090
C            USER MUST SUPPLY OR COMMENT OUT CALL.                      LEV04100
C                                                                       LEV04110
C    USPEC = DIMENSIONS AND INITIALIZES ARRAYS NEEDED TO SPECIFY        LEV04120
C            MATRIX THAT WILL BE USED BY LANCZS SUBROUTINE.             LEV04130
C            IN CASE (4) A-MATRIX AND B-MATRIX MUST BOTH BE SPECIFIED.  LEV04140
C                                                                       LEV04150
C    CMATV = MATRIX-VECTOR MULTIPLY FOR USER-SUPPLIED MATRIX.           LEV04160
C            CASES (1) AND (2).  SEE MATRIX SPECIFICATION SECTION.      LEV04170
C                                                                       LEV04180
C    AMATV = MATRIX-VECTOR MULTIPLY FOR USER-SUPPLIED A-MATRIX.         LEV04190
C            CASES (4) ONLY.  SEE MATRIX SPECIFICATION SECTION.         LEV04200
C                                                                       LEV04210
C    BSOLV = GIVEN A VECTOR V COMPUTES U SUCH THAT B*U = V,             LEV04220
C            USING THE FACTORIZATION OF B.  USED IN CASE (3) ONLY.      LEV04230
C            SEE MATRIX SPECIFICATION SECTION.                          LEV04240
C                                                                       LEV04250
C    LSOLV = PERFORMS 4 FUNCTIONS.  GIVEN A VECTOR V COMPUTES           LEV04260
C            U = L*V, U = (L-TRANSPOSE)*V, U = (L-INVERSE)*V OR         LEV04270
C            U = (L-TRANSPOSE)-INVERSE*V, USING THE CHOLESKY            LEV04280
C            FACTORS OF B.  USED ONLY IN CASE (4).  SEE MATRIX          LEV04290
C            SPECIFICATION SECTION.                                     LEV04300
C                                                                       LEV04310
C                                                                       LEV04320
C-----------------------------------------------------------------------LEV04330
C                                                                       LEV04340
C     COMMENTS FOR EIGENVALUE COMPUTATIONS                              LEV04350
C                                                                       LEV04360
```

```
C-----------------------------------------------------------------------
C
C
C-----PARAMETER CONTROLS FOR EIGENVALUE PROGRAMS------------------------
C
C
C     PARAMETER CONTROLS ARE INTRODUCED TO ALLOW SEGMENTATION OF THE
C     EIGENVALUE COMPUTATIONS AND TO ALLOW VARIOUS COMBINATIONS OF
C     READ/WRITES.
C
C     THE FLAG ISTART CONTROLS THE T-MATRIX (ALPHA/BETA HISTORY)
C     GENERATION.
C
C     ISTART = (0,1)  MEANS
C
C              (0) THERE IS NO EXISTING ALPHA/BETA HISTORY AND ONE
C                  MUST BE GENERATED.
C
C              (1) THERE IS AN EXISTING ALPHA/BETA HISTORY AND IT IS
C                  TO BE READ IN FROM FILE 2 AND EXTENDED IF NECESSARY.
C
C     THE FLAG ISTOP CAN BE USED IN CONJUNCTION WITH THE FLAG ISTART TO
C     ALLOW SEGMENTATION OF THE EIGENVALUE COMPUTATIONS.
C
C     ISTOP  = (0,1)  MEANS
C
C              (0) PROGRAM COMPUTES ONLY THE REQUESTED ALPHAS/BETAS,
C                  STORES THEM AND THE LAST 2 LANCZOS VECTORS GENERATED
C                  IN FILE 1 AND THEN TERMINATES.  IN CASE (4) THERE
C                  ARE ACTUALLY 3 VECTORS TO BE SAVED.
C
C              (1) PROGRAM COMPUTES REQUESTED ALPHAS/BETAS AND THEN
C                  USES THE BISEC SUBROUTINE TO CALCULATE EIGENVALUES
C                  OF THE TRIDIAGONAL MATRICES GENERATED FOR THE ORDERS
C                  SPECIFIED BY THE USER AND ON THE USER-SPECIFIED
C                  INTERVALS.  PROGRAM THEN USES THE SUBROUTINE INVERR
C                  TO COMPUTE ERROR ESTIMATES FOR THE ISOLATED GOOD
C                  T-EIGENVALUES WHICH ARE USED TO CHECK THE
C                  CONVERGENCE OF THESE T-EIGENVALUES.
C
C     CONTROL PARAMETERS FOR WRITES
C
C     IHIS  = (0,1)  MEANS
C
C             (0) IF ISTOP .GT. 0 THEN ALPHA/BETAS ARE NOT SAVED ON
C                 FILE 1.
C
C             (1) PROGRAM WRITES ALPHAS/BETAS AND LAST 2 LANCZOS
C                 VECTORS TO FILE 1 SO THAT THE T-MATRIX GENERATION
C                 MAY BE REUSED OR CONTINUED LATER IF NECESSARY.
C                 TYPICALLY ONE WOULD ALWAYS DO THIS ON ANY RUN WHERE
C                 A HISTORY FILE IS BEING GENERATED.  HISTORY MUST BE
C                 SAVED IN MACHINE FORMAT ((4Z20) FOR IBM 3081) SO
C                 THAT NO ERRORS ARE INTRODUCED BY FORMAT CONVERSIONS.
C
```

```
C     IDIST  = (0,1)  MEANS                                             LEV04920
C                                                                       LEV04930
C              (0) DISTINCT EIGENVALUES OF T-MATRICES ARE NOT SAVED.    LEV04940
C                                                                       LEV04950
C              (1) PROGRAM WRITES COMPUTED DISTINCT EIGENVALUES OF      LEV04960
C                  T-MATRICES ALONG WITH THEIR T-MULTIPLICITIES         LEV04970
C                  TO FILE 11.                                          LEV04980
C                                                                       LEV04990
C     IWRITE = (0,1)   MEANS                                            LEV05000
C                                                                       LEV05010
C              (0) NO EXTENDED OUTPUT FROM SUBROUTINES BISEC AND INVERR LEV05020
C                  IS SENT TO FILE 6.                                   LEV05030
C                                                                       LEV05040
C              (1) INDIVIDUAL COMPUTED T-EIGENVALUES AND CORRESPONDING  LEV05050
C                  ERROR ESTIMATES FROM THE SUBROUTINES BISEC AND INVERRLEV05060
C                  ARE PRINTED OUT TO FILE 6 AS THEY ARE COMPUTED.      LEV05070
C                                                                       LEV05080
C     THE PROGRAM ALWAYS MAKES A SEPARATE LIST OF THE COMPUTED GOOD     LEV05090
C     T-EIGENVALUES ALONG WITH THEIR MINIMAL GAPS AND WRITES THEM OUT   LEV05100
C     TO FILE 3.  CORRESPONDING ERROR ESTIMATES FOR ANY ISOLATED        LEV05110
C     GOOD T-EIGENVALUES ARE ALWAYS WRITTEN TO FILE 4.                  LEV05120
C                                                                       LEV05130
C                                                                       LEV05140
C-----INPUT/OUTPUT FILES FOR EIGENVALUE PROGRAMS------------------------LEV05150
C                                                                       LEV05160
C     ANY INPUT DATA OTHER THAN THE ALPHA/BETA HISTORY SHOULD BE STORED LEV05170
C     ON FILE 5. SEE SAMPLE INPUT/OUTPUT FROM TYPICAL RUN.              LEV05180
C     THE READ STATEMENTS IN THE GIVEN FORTRAN PROGRAM ASSUME THAT      LEV05190
C     THE DATA STORED ON FILE 5 IS IN FREE FORMAT.  USER SHOULD NOTE    LEV05200
C     THAT 'FREE FORMAT' IS NOT CLASSIFIED AS PORTABLE BY PFORT SO THAT LEV05210
C     THE USER MAY HAVE TO MODIFY THE READ STATEMENTS FROM FILE 5 TO    LEV05220
C     CONFORM TO WHAT IS PERMISSIBLE ON THE MACHINE BEING USED.         LEV05230
C                                                                       LEV05240
C     FILE 6 WAS USED AS THE INTERACTIVE TERMINAL OUTPUT FILE.          LEV05250
C     THIS FILE PROVIDES A RUNNING ACCOUNT OF THE PROGRESS OF THE       LEV05260
C     COMPUTATIONS.  THE AMOUNT OF INFORMATION PRINTED OUT IS           LEV05270
C     CONTROLLED BY THE PARAMETER IWRITE.                               LEV05280
C                                                                       LEV05290
C DESCRIPTION OF OTHER I/O FILES                                        LEV05300
C                                                                       LEV05310
C FILE (K)     CONTAINS:                                                LEV05320
C                                                                       LEV05330
C      (1)     OUTPUT FILE:                                             LEV05340
C              HISTORY FILE OF NEWLY-GENERATED T-MATRIX (ALPHA AND      LEV05350
C              BETA VECTORS) AND LAST 2 LANCZOS VECTORS USED            LEV05360
C              IN THE T-MATRIX GENERATION.  NOTE THAT IN CASE (4)       LEV05370
C              THREE 'LANCZOS' VECTORS ARE WRITTEN TO FILE 1.           LEV05380
C              IF IHIS = 0 AND ISTOP = 1, FILE 1 IS NOT WRITTEN.        LEV05390
C                                                                       LEV05400
C      (2)     INPUT FILE:                                              LEV05410
C              SAME AS FILE 1 EXCEPT THAT IT CONTAINS A                 LEV05420
C              PREVIOUSLY-GENERATED T-MATRIX (IF ANY).  IF ISTART = 1,  LEV05430
C              PROGRAM ASSUMES THAT THERE IS A HISTORY FILE OF ALPHAS   LEV05440
C              AND BETAS ON FILE 2.  THESE ALPHAS AND BETAS ARE         LEV05450
C              READ IN ALONG WITH THE LAST 2 LANCZOS VECTORS THAT       LEV05460
```

```
C            WERE GENERATED.  IN CASE (4) THREE 'LANCZOS' VECTORS
C            ARE READ IN FROM FILE 2.
C
C     (3)    OUTPUT FILE:
C            COMPUTED GOOD EIGENVALUES OF THE T-MATRICES USED. ALSO
C            CONTAINS T-MULTIPLICITIES OF THESE EIGENVALUES AS
C            EIGENVALUES OF THE T-MATRIX, AND THEIR GAPS AS
C            EIGENVALUES IN THE A-MATRIX AND IN THE T-MATRIX.
C            FILE 3 IS ALWAYS WRITTEN.  IN CASE (3) THIS OUTPUT
C            CONTAINS THE EIGENVALUES OF THE B-INVERSE MATRIX
C            SINCE IN THIS CASE THE T-MATRICES CORRESPOND TO
C            THE B-INVERSE MATRIX AND NOT TO THE A-MATRIX.  IN
C            THIS CASE, 3 SETS OF GAPS ARE GIVEN, THOSE IN
C            THE T-MATRIX, IN THE B-INVERSE MATRIX AND THOSE
C            FOR THE CORRESPONDING EIGENVALUES IN THE A-MATRIX.
C
C     (4)    OUTPUT FILE:
C            ERROR ESTIMATES FOR THE ISOLATED GOOD T-EIGENVALUES
C            WHICH ARE OBTAINED USING THE SUBROUTINE INVERR. THESE
C            ESITMATES USE THE LAST COMPONENTS OF THE ASSOCIATED
C            T-EIGENVECTORS WHICH ARE COMPUTED USING INVERSE
C            ITERATION.  FILE 4 IS ALWAYS WRITTEN.
C
C
C     (7)    INPUT FILE:
C            USED ONLY IN CASES (3) AND (4), FACTORED INVERSES
C            OF REAL SYMMETRIC MATRICES AND GENERALIZED EIGENVALUE
C            PROBLEM.  CONTAINS THE REQUIRED FACTORIZATION OF THE
C            B-MATRIX.
C
C     (8)    INPUT FILE:
C            SAMPLE USPEC SUBROUTINE ASSUMES THAT THE ARRAYS
C            REQUIRED TO SPECIFY THE USER'S-MATRIX ARE STORED ON
C            FILE 8.  USERS MUST MAKE WHATEVER DEFINITIONS ARE
C            APPROPRIATE FOR THEIR MATRICES.  NOTE THAT IN CASE
C            (3) THE LANCZS SUBROUTINE DOES NOT USE THE MATRIX
C            ON FILE 8 IN THE T-MATRIX GENERATION, RATHER IT
C            USES THE FACTORIZATION OF AN ASSOCIATED
C            B-MATRIX WHICH IS STORED ON FILE 7.  IN CASE (4),
C            THE INFORMATION STORED ON BOTH FILES 7 AND 8 IS USED.
C
C     (9)    INPUT AND OUTPUT FILE:
C            CAN BE USED TO STORE THE TRUE EIGENVALUES OF THE
C            GIVEN PROBLEM,  WHEN THE PROCEDURE
C            IS BEING EXERCISED ON A TEST MATRIX.
C
C    (11)    OUTPUT FILE:
C            COMPUTED DISTINCT EIGENVALUES OF T-MATRICES USED.
C            ALSO CONTAINS THEIR T-MULTIPLICITIES AND T-GAPS TO
C            NEAREST DISTINCT EIGENVALUES, AND THE T-MULTIPLICITY
C            PATTERN OF THE GOOD AND THE SPURIOUS T-EIGENVALUES.
C            FILE 11 IS WRITTEN ONLY IF IDIST = 1.
C
C
C-----PARAMETERS SET BY THE EIGENVALUE PROGRAMS------------------------
```

```
C                                                                       LEV06020
C                                                                       LEV06030
C     THESE PARAMETERS ARE SET INTERNALLY IN THE PROGRAM                LEV06040
C                                                                       LEV06050
C     SCALEK     K = 1,2,3,4                                            LEV06060
C                                                                       LEV06070
C                THE SCALING FACTORS SCALEK HAVE BEEN INTRODUCED IN AN  LEV06080
C                ATTEMPT TO MAKE THE TOLERANCES USED IN THE             LEV06090
C                T-MULTIPLICITY, SPURIOUS, ISOLATION AND PRTESTS ADJUST LEV06100
C                TO THE SCALE OF THE GIVEN MATRIX.  THESE FACTORS MUST  LEV06110
C                NOT BE MODIFIED.  THESE TOLERANCES OCCUR IN LUMP,      LEV06120
C                ISOEV, AND PRTEST.                                     LEV06130
C                                                                       LEV06140
C     NOTE:    THE USER SHOULD NOTE THAT IF THE MATRIX BEING            LEV06150
C     PROCESSED IS VERY STIFF, THAT IS THE RATIO OF THE LARGEST         LEV06160
C     EIGENVALUE IN MAGNITUDE TO THE SMALLEST IN MAGNITUDE IS VERY      LEV06170
C     LARGE, THEN THE TOLERANCES BEING USED IN BISEC, LUMP, ISOEV       LEV06180
C     AND PRTEST MAY NOT TREAT THE SMALL END (SMALL IN MAGNITUDE)       LEV06190
C     VERY WELL.  IN SOME SUCH CASES A USER-INTRODUCED REDUCTION        LEV06200
C     IN THE SIZE OF TKMAX AND THE SUBSEQUENT RECOMPUTATION OF          LEV06210
C     THE T-MATRIX EIGENVALUES IN (ONLY) THE LOWER END OF THE           LEV06220
C     SPECTRUM WITH THIS TKMAX MAY RESULT IN IMPROVED COMPUTATIONS      LEV06230
C     AT THE LOW END.                                                   LEV06240
C                                                                       LEV06250
C     THE LUMP, ISOEV, AND PRTEST TOLERANCES THAT WERE USED             LEV06260
C     MOST IN  THE TESTING OF THESE ALGORITHMS WERE NOT                 LEV06270
C     SCALE INVARIANT BUT SEEMED TO WORK WELL ON MATRICES THAT          LEV06280
C     HAD EIGENVALUES WITH MAGNITUDES BOTH GREATER THAN AND LESS        LEV06290
C     THAN 1.  THESE TOLERANCES ARE ALSO INCLUDED IN THESE THREE        LEV06300
C     SUBROUTINES BUT AS COMMENTED OUT STATEMENTS.  THEY CAN BE         LEV06310
C     REVIVED BY COMMENTING OUT THE CORRESPONDING TOLERANCES            LEV06320
C     SPECIFIED IN THE STATEMENT ABOVE EACH OF THESE.                   LEV06330
C                                                                       LEV06340
C     IMPORTANT TOLERANCES OR SCALES THAT ARE USED REPEATEDLY           LEV06350
C     THROUGHOUT THE LANCZOS EIGENVALUE PROGRAMS ARE THE FOLLOWING:     LEV06360
C     SCALED MACHINE EPSILON:  TTOL = TKMAX*EPSM WHERE                  LEV06370
C     EPSM = 2*MACHINE EPSILON AND                                      LEV06380
C     TKMAX = MAX(|ALPHA(J)|,BETA(J), J = 1,MEV)                        LEV06390
C     BISEC CONVERGENCE TOLERANCE:  BISTOL = DSQRT(1000+MEV)*TTOL       LEV06400
C     BISEC T-MULTIPLICITY TOLERANCE:  MULTOL = (1000+MEV)*TTOL         LEV06410
C     LANCZOS CONVERGENCE TOLERANCE:   CONTOL = BETA(MEV+1)*1.D-10      LEV06420
C                                                                       LEV06430
C                                                                       LEV06440
C     BTOL = RELATIVE TOLERANCE USED TO ESTIMATE ANY LOSS OF LOCAL      LEV06450
C            ORTHOGONALITY OF THE LANCZOS VECTORS AFTER THE T-MATRIX    LEV06460
C            HAS BEEN GENERATED.  THE LANCZOS PROCEDURE WORKS WELL      LEV06470
C            ONLY IF LOCAL ORTHOGONALITY BETWEEN SUCCESSIVE LANCZOS     LEV06480
C            VECTORS IS MAINTAINED.  THE TNORM SUBROUTINE TESTS         LEV06490
C            WHETHER OR NOT                                             LEV06500
C                                                                       LEV06510
C                 MINIMUM  |BETA(I)|/||A|| > BTOL.                      LEV06520
C                 I=2,KMAX                                              LEV06530
C                                                                       LEV06540
C            IF THIS TEST IS VIOLATED BY SOME BETA AND A T-MATRIX THAT  LEV06550
C            WOULD INCLUDE SUCH A BETA IS REQUESTED, THEN THE LANCZOS   LEV06560
```

```
C           PROCEDURE WILL TERMINATE FOR THE USER TO DECIDE WHAT TO
C           DO.  THE USER CAN OVER-RIDE THIS TEST BY SIMPLY DECREASING
C           THE SIZE OF BTOL, BUT THEN CONVERGENCE IS NOT AS CERTAIN.
C           THE PROGRAM SETS BTOL = 1.D-8 WHICH IS A VERY CONSERVATIVE
C           CHOICE. THE || A || IS ESTIMATED BY USING AN ESTIMATE
C           OF THE NORM OF THE T-MATRIX, T(1,KMAX).
C
C     GAPTOL = RELATIVE TOLERANCE USED IN THE SUBROUTINE ISOEV
C              TO DETERMINE WHICH OF THE GOOD T-EIGENVALUES NEED
C              ERROR ESTIMATES.  THE PROGRAM SETS GAPTOL = 1.D-8.
C              IF FOR A GIVEN 'GOOD' T-EIGENVALUE THE COMPUTED GAP
C              IS TOO SMALL AND IS DUE TO A 'SPURIOUS' T-EIGENVALUE
C              THEN THE 'GOOD' T-EIGENVALUE IS ASSUMED TO HAVE CONVERGED
C              AND NO ERROR ESTIMATES ARE COMPUTED.
C
C
C-----USER-SPECIFIED PARAMETERS FOR EIGENVALUE PROGRAMS-----------------
C
C
C     RELTOL = RELATIVE TOLERANCE USED IN 'COMBINING' COMPUTED
C              EIGENVALUES OF T(1,MEV) PRIOR TO COMPUTING ERROR
C              ESTIMATES.
C
C     THE LUMPING OF T-EIGENVALUES OCCURS IN SUBROUTINE LUMP.
C     LUMPING IS NECESSARY BECAUSE IT IS IMPOSSIBLE TO ACCURATELY
C     PREDICT THE ACCURACY OF THE BISEC SUBROUTINE.  LUMP 'COMBINES'
C     T-EIGENVALUES THAT HAVE SLIPPED BY THE TOLERANCE THAT WAS USED
C     IN THE T-MULTIPLICITY TESTS.  IN PARTICULAR IF FOR SOME J,
C
C   |EVALUE(J)-EVALUE(J-1)| < DMAX1(RELTOL*|EVALUE(J)|,SCALE2*MULTOL)
C
C     THEN THESE T-EIGENVALUES ARE 'COMBINED'.  MULTOL IS THE TOLERANCE
C     THAT WAS USED IN THE T-MULTIPLICITY TEST IN BISEC.  SEE THE HEADER
C     ON THE LUMP SUBROUTINE FOR MORE DETAILS.
C
C     RELTOL IS SET TO 1.D-10.
C
C     MXINIT = MAXIMUM NUMBER OF INVERSE ITERATIONS ALLOWED IN
C              SUBROUTINE INVERR FOR EACH ISOLATED GOOD T-EIGENVALUE.
C              TYPICALLY ONLY ONE ITERATION IS REQUIRED.
C
C     SEEDS FOR RANDOM NUMBER GENERATORS = INTEGER*4 SCALARS.
C
C               (1) SVSEED = SEED FOR STARTING VECTOR USED IN
C                   T-MATRIX GENERATION IN LANCZS SUBROUTINE
C
C               (2) RHSEED = SEED FOR RIGHT-HAND SIDE USED IN
C                   INVERSE ITERATION COMPUTATIONS IN INVERR.
C
C     BISEC DATA
C
C     (1) NINT  =  NUMBER OF SUBINTERVALS ON WHICH EIGENVALUES ARE
C                  TO BE COMPUTED.
C
C     (2) LB(J) = (J = 1,NINT)  =  LEFT END POINTS OF THESE INTERVALS.
```

```
C               MUST BE PROVIDED IN INCREASING ORDER.  THAT IS,       LEV07120
C               LB(J) < LB(J+1) FOR J = 1,NINT.                       LEV07130
C                                                                     LEV07140
C     (3) UB(J) = (J = 1,NINT) =  RIGHT END POINTS OF THESE INTERVALS.  LEV07150
C               MUST BE PROVIDED IN INCREASING ORDER.  THAT IS,       LEV07160
C               UB(J) < UB(J+1) FOR J = 1,NINT.                       LEV07170
C                                                                     LEV07180
C     (4) MXSTUR  =  MAXIMUM NUMBER OF STURM ITERATIONS ALLOWED FOR    LEV07190
C                    ENTIRE SET OF EIGENVALUE CALCULATIONS OVER ALL    LEV07200
C                    SPECIFIED SIZE T-MATRICES.  PROGRAM WILL          LEV07210
C                    TERMINATE IF THIS LIMIT IS EXCEEDED.              LEV07220
C                                                                     LEV07230
C     T-MATRICES                                                      LEV07240
C                                                                     LEV07250
C     SIZES OF T-MATRICES                                             LEV07260
C                                                                     LEV07270
C             (1) KMAX= MAXIMUM ORDER FOR T-MATRIX THAT USER IS WILLING LEV07280
C                       TO CONSIDER.                                  LEV07290
C                                                                     LEV07300
C             (2) NMEVS = MAXIMUM NUMBER OF T-MATRICES THAT WILL BE    LEV07310
C                         CONSIDERED.                                 LEV07320
C                                                                     LEV07330
C             (3) NMEV(J)  (J=1,NMEVS)  = SIZES OF T-MATRIX TO BE      LEV07340
C                                         CONSIDERED SEQUENTIALLY.    LEV07350
C                                                                     LEV07360
C     T-MATRIX-GENERATION                                             LEV07370
C                                                                     LEV07380
C     USER SHOULD NOTE THAT THIS PROGRAM FIRST COMPUTES A T-MATRIX     LEV07390
C     OF ORDER KMAX AND THEN CYCLES THROUGH THE T-MATRICES SPECIFIED   LEV07400
C     A PRIORI BY THE USER, USING THE SUBROUTINE BISEC TO COMPUTE THE  LEV07410
C     EIGENVALUES OF THE T-MATRICES ON THE INTERVALS SPECIFIED BY      LEV07420
C     THE USER.                                                       LEV07430
C                                                                     LEV07440
C     IDEALLY, ONE WOULD COMPUTE THE EIGENVALUE APPROXIMATIONS AT A    LEV07450
C     REASONABLE SIZE T-MATRIX, LOOK AT THE ACCURACY OF THE COMPUTED   LEV07460
C     RESULTS AND USE THAT TO DETERMINE AN APPROPRIATE                LEV07470
C     INCREMENT FOR THE SIZE OF THE T-MATRIX BASED UPON WHAT          LEV07480
C     HAS ALREADY CONVERGED AND UPON THE SIZES OF THE ERROR ESTIMATES  LEV07490
C     ON THOSE EIGENVALUES THAT ARE DESIRED BUT THAT HAVE NOT YET      LEV07500
C     CONVERGED. HOWEVER, IN THE INTERESTS OF GENERALITY AND          LEV07510
C     SIMPLICITY WE DID NOT DO THAT HERE.                             LEV07520
C                                                                     LEV07530
C                                                                     LEV07540
C-----CONVERGENCE TESTS FOR THE EIGENVALUE PROGRAMS-------------------- LEV07550
C                                                                     LEV07560
C                                                                     LEV07570
C     THE CONVERGENCE TEST INCORPORATED IN THIS PROGRAM IS            LEV07580
C     BASED UPON THE ASSUMPTION THAT THOSE T-EIGENVALUES AND THEIR     LEV07590
C     ASSOCIATED T-EIGENVECTORS WHICH CORRESPOND TO THE               LEV07600
C     EIGENVALUES AND RITZVECTORS WHICH WE WISH TO COMPUTE            LEV07610
C     CONVERGE AS THE T-SIZE IS INCREASED.                            LEV07620
C                                                                     LEV07630
C     AS CURRENTLY PROGRAMMED, CONVERGENCE IS CHECKED BY EXAMINING     LEV07640
C     THE SIZES OF ALL OF THE COMPUTED ERROR ESTIMATES ON ALL OF THE   LEV07650
C     INTERVALS SPECIFIED BY THE USER.  IDEALLY CONVERGENCE SHOULD     LEV07660
```

```
C     BE CHECKED ONLY ON THOSE EIGENVALUES OF INTEREST AND              LEV07670
C     ONCE THE EIGENVALUES ON SUB-INTERVALS OF THESE INTERVALS HAVE     LEV07680
C     CONVERGED, ANY SUBSEQUENT EIGENVALUE COMPUTATIONS SHOULD BE       LEV07690
C     MADE ONLY ON THE UNCONVERGED PORTIONS.  OBVIOUSLY, IT WOULD BE    LEV07700
C     DIFFICULT TO INCORPORATE CODE TO DO THE ABOVE WITHOUT KNOWING     LEV07710
C     A PRIORI PRECISELY WHAT THE USER IS TRYING TO COMPUTE.            LEV07720
C     THEREFORE, WE DID NOT ATTEMPT TO DO THIS.  IF ONE WISHES TO       LEV07730
C     MAKE SUCH A MODIFICATION THEN ONE MUST ALSO MODIFY THE PROGRAM    LEV07740
C     SO THAT IT CREATES AN OVERALL LIST OF THE CONVERGED 'GOOD'        LEV07750
C     T-EIGENVALUES AS THEY ARE COMPUTED, SINCE CONVERGED 'GOOD'        LEV07760
C     T-EIGENVALUES WHICH WERE COMPUTED AT A PARTICULAR VALUE OF MEV    LEV07770
C     WOULD NO LONGER BE RECOMPUTED AT LARGER VALUES OF MEV.            LEV07780
C                                                                       LEV07790
C     IF ONLY A FEW EIGENVALUES ARE TO BE COMPUTED THEN SUCH CHANGES    LEV07800
C     WOULD NOT MAKE MUCH DIFFERENCE IN THE RUNNING TIME.               LEV07810
C                                                                       LEV07820
C                                                                       LEV07830
C-----ARRAYS REQUIRED BY THE EIGENVALUE PROGRAMS------------------------LEV07840
C                                                                       LEV07850
C                                                                       LEV07860
C     ALL 4 CASES                                                       LEV07870
C                                                                       LEV07880
C     ALPHA(J) = REAL*8 ARRAY. ITS DIMENSION MUST BE AT LEAST KMAX,     LEV07890
C                THE LENGTH OF THE LARGEST T-MATRIX ALLOWED.  THIS      LEV07900
C                ARRAY CONTAINS THE DIAGONAL ENTRIES OF THE T-MATRICES. LEV07910
C                                                                       LEV07920
C     BETA(J) = REAL*8 ARRAY.  ITS DIMENSION MUST BE AT LEAST KMAX+1.   LEV07930
C               THIS ARRAY CONTAINS THE SUBDIAGONAL ENTRIES OF THE      LEV07940
C               T-MATRICES.                                             LEV07950
C                                                                       LEV07960
C               THE ALPHA AND BETA VECTORS ARE NOT ALTERED              LEV07970
C               DURING THE CALCULATIONS.                                LEV07980
C                                                                       LEV07990
C     V1(J),V2(J),VS(J) = REAL*8 ARRAYS IN REAL SYMMETRIC CASES.        LEV08000
C                         V1 AND V2 ARE COMPLEX*16 IN HERMITIAN CASE.   LEV08010
C                         IN CASES (1) AND (2) VS MUST BE OF            LEV08020
C                         DIMENSION AT LEAST KMAX.  IN CASES (3) AND    LEV08030
C                         (4) VS MUST BE OF DIMENSION AT LEAST          LEV08040
C                         MAX(N,KMAX).  IN REAL SYMMETRIC CASES         LEV08050
C                         V1 MUST BE OF DIMENSION AT LEAST              LEV08060
C                         MAX(KMAX+1,N) AND V2 MUST BE OF DIMENSION     LEV08070
C                         MAX(KMAX,N).  IN HERMITIAN CASES V1           LEV08080
C                         MUST BE OF DIMENSION MAX(N,(KMAX+1)/2)        LEV08090
C                         AND V2 OF DIMENSION AT LEAST MAX(N,KMAX/2).   LEV08100
C                         IN ALL CASES HOWEVER, THE ABOVE DIMENSIONS    LEV08110
C                         FOR V2 ARE VALID ONLY IF NO MORE              LEV08120
C                         THAN KMAX/2 EIGENVALUES OF THE GIVEN          LEV08130
C                         T-MATRICES ARE TO BE COMPUTED IN ANY GIVEN    LEV08140
C                         SUBINTERVAL. V2 IS USED IN THE SUBROUTINE     LEV08150
C                         BISEC TO HOLD THE UPPER AND LOWER             LEV08160
C                         ENDPOINTS OF THE SUBINTERVALS GENERATED       LEV08170
C                         DURING THE BISECTIONS.  THEREFORE, ITS        LEV08180
C                         REAL*8 DIMENSION MUST ALWAYS BE AT LEAST      LEV08190
C                         2*Q WHERE Q IS THE MAXIMUM NUMBER OF          LEV08200
C                         EIGENVALUES OF THE SPECIFIED T-MATRIX IN ANY  LEV08210
```

```
C                        ONE OF THE SPECIFIED INTERVALS.              LEV08220
C                        NOTE THAT IN THE HERMITIAN CASE, V1 AND V2   LEV08230
C                        ARE DEFINED AS COMPLEX*16 IN THE MAIN PROGRAM LEV08240
C                        AND IN THE LANCZS SUBROUTINE BUT ARE         LEV08250
C                        REDEFINED AS REAL*8 IN OTHER SUBROUTINES.    LEV08260
C                                                                     LEV08270
C     LB(J),UB(J) = REAL*8 ARRAYS. EACH MUST BE OF DIMENSION AT LEAST LEV08280
C                  NINT, THE NUMBER OF SUBINTERVALS TO BE CONSIDERED. LEV08290
C                  LB CONTAINS THE LEFT-END POINTS OF THE INTERVALS   LEV08300
C                  ON WHICH EIGENVALUES ARE TO BE COMPUTED.  UB       LEV08310
C                  CONTAINS THE RIGHT-END POINTS.                     LEV08320
C                                                                     LEV08330
C     EXPLAN(J) = REAL*4 ARRAY.  ITS DIMENSION IS 20.  THIS ARRAY IS  LEV08340
C                USED TO ALLOW EXPLANATORY COMMENTS IN THE INPUT FILES.LEV08350
C                                                                     LEV08360
C     G(J) = REAL*4 ARRAY.  ITS DIMENSION MUST BE >= MAX(2*KMAX,N)    LEV08370
C           IT IS USED FOR HOLDING THE RANDOM VECTORS GENERATED,      LEV08380
C           HOLDING THE COMPUTED ERROR ESTIMATES AND THE COMPUTED     LEV08390
C           MINIMAL GAPS FOR THE GOOD T-EIGENVALUES.                  LEV08400
C                                                                     LEV08410
C     MP(J) = INTEGER*4 ARRAY.  ITS DIMENSION MUST BE AT LEAST KMAX,  LEV08420
C             THE MAXIMUM SIZE OF THE T-MATRICES ALLOWED.  IT CONTAINS LEV08430
C             THE T-MULTIPLICITIES OF THE COMPUTED EIGENVALUES. NOTE  LEV08440
C             THAT 'SPURIOUS' T-EIGENVALUES ARE DENOTED BY A          LEV08450
C             T-MULTIPLICITY OF 0.  T-EIGENVALUES THAT THE SUBROUTINE LEV08460
C             PRTEST HAS IDENTIFIED AS 'GOOD' BUT HIDDEN ARE IDENTIFIED LEV08470
C             BY A T-MULTIPLICITY OF -10.                             LEV08480
C                                                                     LEV08490
C     NMEV(J) = INTEGER*4 ARRAY.  ITS DIMENSION MUST BE AT LEAST THE  LEV08500
C               NUMBER OF T-MATRICES ALLOWED.  IT CONTAINS THE ORDERS LEV08510
C               OF THE T-MATRICES TO BE CONSIDERED.                   LEV08520
C                                                                     LEV08530
C                                                                     LEV08540
C     FOR CASE (3) ONLY:                                              LEV08550
C     GR(J),GC(J) = REAL*8 ARRAYS. USED ONLY IN THE HERMITIAN CASE.   LEV08560
C                  GR AND GC MUST EACH BE OF DIMENSION AT LEAST N.    LEV08570
C                  BOTH ARE USED IN THE RANDOM VECTOR GENERATION.     LEV08580
C                  GR IS ALSO USED TO STORE MINIMAL GAPS BETWEEN      LEV08590
C                  'GOOD' T-EIGENVALUES.                              LEV08600
C                                                                     LEV08610
C     FOR CASES (3) AND (4) FOR THE PERMUTATION:                      LEV08620
C                                                                     LEV08630
C     IPR(J), IPT(J) = INTEGER*4 ARRAYS.  EACH OF DIMENSION AT LEAST N. LEV08640
C                     USED TO STORE THE REORDERING OF THE GIVEN MATRIX LEV08650
C                     OR MATRICES.                                    LEV08660
C                                                                     LEV08670
C                                                                     LEV08680
C     OTHER ARRAYS                                                    LEV08690
C                                                                     LEV08700
C     THE USER MUST SPECIFY IN THE SUBROUTINE USPEC (A OR B) WHATEVER LEV08710
C     ARRAYS ARE REQUIRED TO DEFINE THE MATRIX OR MATRICES BEING USED. LEV08720
C     ALSO IN CASES (3) AND (4) ONLY, WHEN WORKING WITH INVERSES      LEV08730
C     OF SPARSE SYMMETRIC MATRICES, IN THE OPTIONAL, PREPROCESSING    LEV08740
C     PROGRAMS PERMUT, LFACT, LORDER, AND LTEST IT IS NECESSARY TO    LEV08750
C     SPECIFY ADDITIONAL ARRAYS JUST FOR THESE COMPUTATIONS.  THE USER LEV08760
```

```
C     IS REFERRED TO THOSE PROGRAMS FOR DETAILS.
C
C
C-----SUBROUTINES INCLUDED----------------------------------------------
C
C
C     LANCZS = COMPUTES THE ALPHA/BETA HISTORY. IN CASES (1) AND (2)
C              REAL SYMMETRIC AND HERMITIAN MATRICES, USES SUBROUTINES
C              FINPRO, GENRAN AND CMATV.  IN CASE (3), INVERSES OF
C              REAL SYMMETRIC MATRICES, USES SUBROUTINES FINPRO,
C              GENRAN AND BSOLV.  IN CASE (4), GENERALIZED EIGENVALUE
C              PROBLEM, USES SUBROUTINES FINPRO, GENRAN, AMATV AND
C              LSOLV.
C
C     BISEC = COMPUTES EIGENVALUES OF THE SPECIFIED T-MATRIX
C             USING STURM SEQUENCING, ON SEQUENCE OF INTERVALS
C             SPECIFIED BY THE USER.  EACH SUBINTERVAL IS TREATED
C             AS OPEN ON THE LEFT AND CLOSED ON THE RIGHT.
C             EIGENVALUES ARE COMPUTED WITH SIMULTANEOUS DETERMINATION
C             OF THE T-MULTIPLICITIES AND OF SPURIOUS T-EIGENVALUES.
C
C     INVERR = USES INVERSE ITERATION ON T-MATRICES TO COMPUTE ERROR
C              ESTIMATES ON COMPUTED GOOD T-EIGENVALUES. (USES GENRAN)
C
C     LUMP = 'COMBINES' EIGENVALUES OF T-MATRIX USING THE RELATIVE
C             TOLERANCE RELTOL.
C
C     ISOEV = CALCULATES GAPS BETWEEN DISTINCT EIGENVALUES OF T-MATRIX
C             AND THEN USES THESE GAPS TO LABEL THOSE 'GOOD'
C             T-EIGENVALUES FOR WHICH ERROR ESTIMATES ARE NOT COMPUTED.
C
C     TNORM = COMPUTES THE SCALE TKMAX USED IN DETERMINING THE
C             TOLERANCES FOR THE SPURIOUS, T-MULTIPLICITY AND PRTESTS.
C             IT ALSO CHECKS FOR LOCAL ORTHOGONALITY OF THE LANCZOS
C             VECTORS BY TESTING THE RELATIVE SIZE OF THE BETAS USING
C             THE RELATIVE TOLERANCE BTOL.
C
C     PRTEST = LOOKS FOR GOOD T-EIGENVALUES THAT HAVE BEEN MISLABELLED
C              BY THE SPURIOUS TEST BECAUSE THEY HAD 'TOO SMALL' A
C              PROJECTION ON THE STARTING LANCZOS VECTOR.
C              (LESS THAN SINGLE PRECISION)
C              TESTS INDICATE THAT SUCH EIGENVALUES ARE RARE.
C              PRTEST SHOULD BE CALLED ONLY AFTER CONVERGENCE
C              HAS BEEN ESTABLISHED.
C
C     INVERM = USED TO COMPUTE ERROR ESTIMATES FOR ANY T-EIGENVALUES
C              WHICH PRTEST INDICATES MAY HAVE BEEN MISLABELLED.
C              SUCH T-EIGENVALUES ARE RELABELLED ONLY IF THEIR ERROR
C              ESTIMATES ARE SUFFICIENTLY SMALL.  PRIMARY USE OF
C              INVERM IS IN THE CORRESPONDING EIGENVECTOR COMPUTATIONS.
C
C     CASES (3) AND (4) ONLY, FACTORED INVERSES:
C
C     FOR OPTIONAL, PRELIMINARY PROCESSING:
C     PERMUT  (PROGRAM CALLS SPARSPAK PACKAGE) :
```

```
C     USES THE NONZERO STRUCTURE OF A GIVEN MATRIX A.                   LEV09320
C     CAN BE USED TO OBTAIN A REORDERING OF A THAT PRESERVES            LEV09330
C     SPARSITY UNDER FACTORIZATION.  PERMUT CALLS                       LEV09340
C     THE SPARSPAK PROGRAMS, (A. GEORGE, J. LIU, E.NG,                  LEV09350
C     U. WATERLOO).  PERMUT ALSO TAKES THE USER-SPECIFIED MATRIX,       LEV09360
C     APPLIES THE SCALE S0 AND THE SHIFT TO IT, AND THEN WRITES         LEV09370
C     OUT THE CORRESPONDING SPARSE MATRIX DATA FILE FOR THE             LEV09380
C     RESULTING MATRIX C = S0*A + SHIFT*I.  SEE THE PERMUT FORTRAN      LEV09390
C     CODE FOR DETAILS.                                                 LEV09400
C                                                                       LEV09410
C     LORDER (STAND-ALONE PROGRAM):                                     LEV09420
C     GIVEN A MATRIX C IN SPARSE FORMAT AND A PERMUTATION P,            LEV09430
C     COMPUTES THE REORDERED MATRIX B = P*C*P' AND WRITES IT            LEV09440
C     TO FILE 9 IN SPARSE FORMAT.  SEE THE LORDER FORTRAN CODE          LEV09450
C     FOR DETAILS.                                                      LEV09460
C                                                                       LEV09470
C     LFACT (STAND-ALONE PROGRAM) :                                     LEV09480
C     GIVEN A POSITIVE DEFINITE MATRIX B IN SPARSE FORMAT               LEV09490
C     COMPUTES THE SPARSE CHOLESKY FACTOR L OF B AND WRITES IT          LEV09500
C     TO FILE 7 IN SPARSE FORMAT.  THUS, B = L*L'.                      LEV09510
C     SEE THE LFACT FORTRAN CODE FOR DETAILS.                           LEV09520
C                                                                       LEV09530
C     LTEST (CALLS 3 USER-SUPPLIED PROGRAMS CMATV, CMATS, AND BSOLV):   LEV09540
C     GIVEN THE FACTORIZATION OF A MATRIX B, LTEST COMPUTES             LEV09550
C     THE SOLUTION OF THE EQUATION B*U = B*V1 FOR A SPECIFIC RANDOMLY-  LEV09560
C     GENERATED V1, WITH AND WITHOUT ITERATIVE REFINEMENT, TO           LEV09570
C     OBTAIN A ROUGH CHECK ON THE NUMERICAL CONDITION OF THE MATRIX B.  LEV09580
C     THIS PROGRAM USES 3 SUBROUTINES CMATV, CMATS, AND BLSOLV.         LEV09590
C     SEE THE LTEST FORTRAN PROGRAM FOR DETAILS.                        LEV09600
C                                                                       LEV09610
C                                                                       LEV09620
C-----OTHER PROGRAMS PROVIDED-------------------------------------------LEV09630
C                                                                       LEV09640
C     LECOMPAC (STAND ALONE PROGRAM):                                   LEV09650
C                  TRANSLATES A REAL SYMMETRIC MATRIX PROVIDED IN THE   LEV09660
C                  FORMAT  I, J, A(I,J) INTO THE SPARSE MATRIX          LEV09670
C                  FORMAT USED IN THE SAMPLE USPEC, CMATV, BSOLV AND    LEV09680
C                  LSOLV SUBROUTINES PROVIDED. IT ASSUMES THAT THE      LEV09690
C                  MATRIX ENTRIES ARE GIVEN EITHER COLUMN BY COLUMN OR  LEV09700
C                  ROW BY ROW.  THE DATA SET CREATED IS WRITTEN TO      LEV09710
C                  FILE 8.                                              LEV09720
C                                                                       LEV09730
C                                                                       LEV09740
C-----COMMENTS ON THE STORAGE REQUIRED FOR EIGENVALUE PROGRAMS----------LEV09750
C                                                                       LEV09760
C                                                                       LEV09770
C     CASE (1), REAL SYMMETRIC MATRICES:                                LEV09780
C                                                                       LEV09790
C     THE ARRAYS IN THE REAL SYMMETRIC EIGENVALUE PROGRAM REQUIRE       LEV09800
C     APPROXIMATELY THE EQUIVALENT OF ONE REAL*8 ARRAY OF DIMENSION     LEV09810
C                                                                       LEV09820
C         3.5*KMAX + 2*MAX(KMAX,N) + .5* MAX(2*KMAX,N)                  LEV09830
C                                                                       LEV09840
C     PLUS WHATEVER IS NEEDED TO GENERATE A*X FOR THE GIVEN MATRIX A.   LEV09850
C     THE ARRAYS ALPHA, BETA, VS AND MP CONSUME 3.5*KMAX*8 BYTES.       LEV09860
```

```
C     THE ARRAYS V1 AND V2 CONSUME 2*MAXIMUM(KMAX,N)*8 BYTES, WITH THE
C     QUALIFICATION STATED ABOVE WHERE V2 IS DEFINED.  THE G-ARRAY
C     CONSUMES .5*MAX(2*KMAX,N)*8 BYTES.
C
C     CASE (2), HERMITIAN MATRICES:
C
C     THE ARRAYS IN THE HERMITIAN EIGENVALUE PROGRAMS REQUIRE
C     THE EQUIVALENT OF ONE REAL*8 ARRAY OF DIMENSION
C
C         3.5*KMAX + 4*MAX(KMAX/2,N) + .5*MAX(2*KMAX,N) + 2*N
C
C     PLUS WHATEVER IS NEEDED TO GENERATE A*X FOR THE GIVEN MATRIX A.
C     THE ARRAYS ALPHA, BETA, VS, AND MP CONSUME 3.5*KMAX*8 BYTES.
C     THE ARRAYS V1 AND V2 CONSUME 4*MAXIMUM(KMAX/2,N)*8 BYTES, WITH THE
C     QUALIFICATION STATED ABOVE WHERE V2 IS DEFINED.  THE G-ARRAY
C     CONSUMES .5*MAX(2*KMAX,N)*8 BYTES.  GR REQUIRES
C     AND GC REQUIRE 2*N*8BYTES.
C
C
C     CASE (3), INVERSES OF REAL SYMMETRIC MATRICES:
C
C     THE ARRAYS IN THE EIGENVALUE PROGRAMS DESIGNED FOR
C     CASE (3), INVERSES OF REAL SYMMETRIC MATRICES USING
C     REORDERING AND FACTORIZATION, REQUIRE
C     THE EQUIVALENT OF ONE REAL*8 ARRAY OF DIMENSION
C
C         3*KMAX + 3*MAX(KMAX,N) + .5*MAX(2*KMAX,N)
C
C     PLUS WHATEVER IS NEEDED TO GENERATE B(INVERSE)*X  FOR THE
C     SCALED, SHIFTED AND PERMUTED VERSION OF A WHICH WE DENOTE
C     BY B.  THE ARRAYS ALPHA, BETA, MP, AND MP2 CONSUME 3*KMAX*8
C     BYTES.  THE ARRAYS V1, V2, AND VS CONSUME 3*MAX(KMAX,N)*8 BYTES,
C     WITH THE QUALIFICATION STATED ABOVE WHERE V2 IS DEFINED.
C     THE G ARRAY CONSUMES .5*MAX(2*KMAX,N)*8 BYTES. THESE NUMBERS
C     DO NOT INCLUDE THE STORAGE REQUIRED BY THE PREPROCESSING PROGRAMS
C     PERMUT, LORDER, LFACT, AND LTEST.
C
C
C     A SYMMETRIC, SPARSE MATRIX OF ORDER N WITH NZS NONZERO ELEMENTS
C     BELOW THE MAIN DIAGONAL WOULD REQUIRE THE EQUIVALENT OF ONE
C     REAL*8 ARRAY OF DIMENSION 1.5*(NZS + N) IF THE POINTERS USED
C     ARE INTEGER*4.
C
C     SOME OF THE ARRAY STORAGE IS NOT ESSENTIAL AND COULD BE
C     ELIMINATED IF STORAGE IS A PROBLEM.
C     THE FOLLOWING COMMENTS APPLY DIRECTLY ONLY TO CASE (1),
C     THE PROGRAMS FOR REAL SYMMETRIC MATRICES, HOWEVER, SIMILAR
C     STATEMENTS COULD BE MADE ABOUT THE OTHER CASES.
C
C     CASE (1), REAL SYMMETRIC PROGRAMS:
C     THE G ARRAY COULD BE REMOVED IF THE USER IS WILLING TO
C
C            (1) REGENERATE THE RANDOM STARTING VECTOR IN INVERR
C                FOR EACH ERROR ESTIMATE
C            (2) WRITE OUT THE ERROR ESTIMATES AND VARIOUS GAPS AS
```

```
C             THEY ARE GENERATED RATHER THAN STORING THEM IN G FOR   LEV10420
C             LATER PRINTOUT                                         LEV10430
C         (3) CHECK CONVERGENCE WITHIN INVERR                        LEV10440
C                                                                    LEV10450
C     CLEARLY THE INDEX VECTOR MP COULD BE AN INTEGER*2 ARRAY AS COULD  LEV10460
C     THE POINTERS USED TO DEFINE THE USER'S MATRIX.                 LEV10470
C                                                                    LEV10480
C     THE USER SHOULD NOTE THAT WITH AN EIGENVALUE SUBROUTINE THAT   LEV10490
C     USES BISECTION (LIKE BISEC) IF MORE THAN 25% OF THE            LEV10500
C     EIGENVALUES ARE TO BE COMPUTED, THEN IT MAY BE MORE            LEV10510
C     ECONOMICAL TO USE THE EISPACK SUBROUTINE IMTQL1.               LEV10520
C     (SEE MATRIX EIGENSYTEM ROUTINES-EISPACK GUIDE (2ND EDITION)    LEV10530
C     B.T. SMITH ET AL, SPRINGER-VERLAG, NEW YORK, 1976, P213.).     LEV10540
C     HOWEVER, IF THE SUBROUTINE IMTQL1 IS TO BE USED IN PLACE       LEV10550
C     OF BISEC, THEN NONTRIVIAL CHANGES IN THE LANCZOS CODE MUST BE  LEV10560
C     MADE. FOR DETAILS OF ONE SUCH IMPLEMENTATION SEE               LEV10570
C     IBM RESEARCH REPORT 8298, COMPUTING                            LEV10580
C     EIGENVALUES OF LARGE SYMMETRIC MATRICES - AN IMPLEMENTATION OF A  LEV10590
C     LANCZOS ALGORITHM WITH NO REORTHOGONALIZATION.  PART II. COMPUTER LEV10600
C     PROGRAMS., DECEMBER 1980, WHICH CONTAINS A GENERAL             LEV10610
C     LANCZOS CODE WHICH INCLUDES AN IMTQL1 OPTION OR                LEV10620
C     PREFERABLY CONTACT THE AUTHORS.                                LEV10630
C                                                                    LEV10640
C     THE BISEC SUBROUTINE WHICH IS INCLUDED IS A MODIFIED FORM OF   LEV10650
C     THE BISECT SUBROUTINE IN EISPACK. BISEC ASSUMES THAT THE       LEV10660
C     VECTOR V2 IS LONG ENOUGH TO HOLD BOTH THE UPPER AND THE        LEV10670
C     LOWER BOUNDS ON THE BISECTION INTERVALS USED TO COMPUTE        LEV10680
C     THE EIGENVALUES OF THE T-MATRICES.  THEREFORE, IF THE          LEV10690
C     LENGTH OF V2 IS ONLY KMAX, BISEC CAN COMPUTE ONLY AT MOST       LEV10700
C     KMAX/2 EIGENVALUES OF THE GIVEN T-MATRIX IN ANY GIVEN          LEV10710
C     SUBINTERVAL.                                                   LEV10720
C                                                                    LEV10730
C     AS PROGRAMMED BISEC USES THE ARRAYS ALPHA,BETA,V1,V2,VS AND MP.   LEV10740
C     HOWEVER, V1 IS USED ONLY TO STORE BETA(J)**2 SO THAT THEY DO NOT  LEV10750
C     HAVE TO BE REGENERATED ON EACH STURM.  IF THE USER IS WILLING TO  LEV10760
C     COMPUTE THE BETA(J)**2 AS NEEDED, THEN V1 COULD BE ELIMINATED  LEV10770
C     FROM BISEC. BISEC STORAGE THEN BECOMES A REAL*8 ARRAY OF DIMENSIONLEV10780
C     4.25*KMAX IF WE ALSO REDUCE MP TO INTEGER*2.  FURTHERMORE,     LEV10790
C     IF ONE KNEW THAT ONLY Q*MEV EIGENVALUES OF T(1,MEV) WERE TO BE LEV10800
C     COMPUTED AT EACH STAGE FOR SOME Q<.5 THEN FURTHER REDUCTIONS IN   LEV10810
C     STORAGE COULD BE MADE IN BISEC.                                LEV10820
C                                                                    LEV10830
C     AS PROGRAMMED INVERR USES ALPHA, BETA,V1,V2,VS,G AND MP.       LEV10840
C     VS CONTAINS THE COMPUTED EIGENVALUES OF T(1,MEV). MP GIVES     LEV10850
C     THEIR T-MULTIPLICITIES AND FLAGS WHICH EIGENVALUES ARE TO HAVE LEV10860
C     ERROR ESTIMATES COMPUTED.  V2 IS USED FOR THE SOLUTION         LEV10870
C     VECTOR IN THE INVERSE ITERATION AND V1 FOR THE FACTORIZATION.  LEV10880
C     G CONTAINS THE RANDOMLY-GENERATED STARTING VECTOR FOR THE      LEV10890
C     INVERSE ITERATION. THE BASIC STORAGE FOR INVERR IS THEREFORE   LEV10900
C     A REAL*8 ARRAY OF DIMENSION 4*KMAX PLUS THE STORAGE NEEDED FOR LEV10910
C     THE COMPUTED T-EIGENVALUES AND THEIR T-MULTIPLICITIES.         LEV10920
C                                                                    LEV10930
C     VS COULD BE USED TO STORE ONLY THOSE COMPUTED EIGENVALUES OF   LEV10940
C     T(1,MEV) THAT ARE OF INTEREST. IN THAT CASE THE DIMENSIONS OF VS  LEV10950
C     AND OF MP NEED NOT BE ANY LONGER THAN THE NUMBER OF SUCH       LEV10960
```

```
C     EIGENVALUES. AS PROGRAMMED, ALL THE COMPUTED DISTINCT EIGENVALUES
C     OF T(1,MEV) ARE STORED IN VS.  THEREFORE TO TAKE ADVANTAGE OF
C     SUCH A REDUCTION IN STORAGE THE USER WOULD HAVE TO MODIFY THAT
C     PART OF THE PROGRAM AND ALSO COMMENT OUT THE CALL TO THE
C     SUBROUTINE PRTEST.
C
C
C-----------------------------------------------------------------------
C
C     COMMENTS FOR EIGENVECTOR COMPUTATIONS
C
C-----------------------------------------------------------------------
C
C
C     THE EIGENVALUES WHOSE EIGENVECTORS ARE TO BE COMPUTED MUST
C     HAVE BEEN COMPUTED USING THE CORRESPONDING LANCZOS EIGENVALUE
C     PROGRAMS BECAUSE THE EIGENVECTOR PROGRAMS WILL USE THE SAME
C     FAMILY OF LANCZOS TRIDIAGONAL MATRICES THAT WAS USED IN THE
C     CORRESPONDING EIGENVALUE COMPUTATIONS.
C
C     THESE PROGRAMS ASSUME THAT THE EIGENVALUES SUPPLIED TO IT
C     HAVE BEEN COMPUTED ACCURATELY, AS MEASURED BY THE
C     ERROR ESTIMATES COMPUTED IN THE CORRESPONDING LANCZOS
C     EIGENVALUE COMPUTATIONS, ALTHOUGH THESE ESTIMATES ARE
C     TYPICALLY CONSERVATIVE.  IN CASES (1), (2) AND (4), THE
C     EIGENVALUES OF INTEREST ARE STORED IN THE ARRAY GOODEV(J),
C     J=1,NGOOD.  IN CASE (3) THE PROGRAM WORKS WITH THE
C     EIGENVALUES OF B(INVERSE) WHICH ARE STORED IN THE ARRAY
C     GOODBI(J), J=1,NGOOD.  THE CORRESPONDING EIGENVALUES
C     OF A ARE STORED IN GOODA(J), J=1,NGOOD.
C
C     FOR EACH GOODEV(J), THE SUBROUTINE STURMI COMPUTES THE
C     SMALLEST SIZE LANCZOS TRIDIAGONAL MATRIX, T(1,M1(J)), FOR
C     WHICH  GOODEV(J) IS AN EIGENVALUE TO WITHIN A SPECIFIED
C     TOLERANCE.  IT ALSO ATTEMPTS TO COMPUTE THE SIZE, M2(J),
C     BY WHICH THE GIVEN EIGENVALUE BECOMES A DOUBLE EIGENVALUE
C     TO WITHIN THE GIVEN TOLERANCE.  THESE VALUES ARE USED
C     TO DETERMINE 1ST GUESSES AT SIZES FOR THE T-EIGENVECTORS
C     THAT WILL BE USED IN THE RITZ VECTOR COMPUTATIONS.
C     SUBROUTINE INVERM SUCCESSIVELY COMPUTES CORRESPONDING
C     EIGENVECTORS OF ENLARGED T-MATRICES UNTIL A SUITABLE
C     SIZE T-MATRIX IS DETERMINED FOR EACH J.  UP TO 10 SUCH
C     EIGENVECTOR COMPUTATIONS ARE ALLOWED FOR EACH EIGENVALUE.
C
C     AFTER APPROPRIATE T-EIGENVECTORS HAVE BEEN COMPUTED,
C     RITZ VECTOR CORRESPONDING TO THESE T-EIGENVECTORS ARE THEN
C     COMPUTED AND TAKEN AS APPROXIMATE EIGENVECTORS OF A FOR THE
C     GIVEN EIGENVALUES, GOODEV(J), J = 1, ..., NGOOD.
C
C     THIS IMPLEMENTATION FIRST COMPUTES ALL OF THE RELEVANT
C     EIGENVECTORS OF THE SYMMETRIC TRIDIAGONAL MATRICES
C     IN THE VECTOR, TVEC.
C
C     THEN, AS EACH OF THE LANCZOS VECTORS IS REGENERATED, ALL
C     OF THE RITZ VECTORS CORRESPONDING TO THESE
```

```
C     T-EIGENVECTORS ARE UPDATED USING THE CURRENTLY-GENERATED          LEV11520
C     LANCZOS VECTOR.  LANCZOS VECTORS ARE GENERATED (NOTE              LEV11530
C     THAT THEY ARE NOT BEING KEPT), UNTIL ENOUGH HAVE                  LEV11540
C     BEEN GENERATED TO MAP THE LONGEST T-EIGENVECTOR INTO ITS          LEV11550
C     CORRESPONDING RITZ VECTOR.  THE ARRAY RITVEC CONTAINS THE         LEV11560
C     SUCCESSIVE RITZ VECTORS WHICH ARE THE APPROXIMATE                 LEV11570
C     EIGENVECTORS OF A.                                                LEV11580
C                                                                       LEV11590
C                                                                       LEV11600
C-----PARAMETER CONTROLS FOR EIGENVECTOR PROGRAMS-----------------------LEV11610
C                                                                       LEV11620
C                                                                       LEV11630
C     IN CASES (3) AND (4) WHERE A SPARSE FACTORIZATION OF A            LEV11640
C     SPECIFIED MATRIX IS USED, THE USER SPECIFIES USING THE FLAG       LEV11650
C     JPERM WHETHER OR NOT THE FACTORIZATION SUPPLIED CORRESPONDS       LEV11660
C     TO THE ORIGINAL MATRIX OR TO A PERMUTATION OF THE ORIGINAL        LEV11670
C     MATRIX.                                                           LEV11680
C                                                                       LEV11690
C     JPERM = (0,1) MEANS                                               LEV11700
C              0    NO PERMUTATION                                      LEV11710
C              1    MATRIX HAS BEEN PERMUTED.  NOTE THAT IN             LEV11720
C                   CASE (4) THE PROGRAM WILL ASSUME THAT THE           LEV11730
C                   DATA SUPPLIED FOR THE A-MATRIX CORRESPONDS TO THE   LEV11740
C                   CORRESPONDING PERMUTATION OF THE ORIGINAL A-MATRIX. LEV11750
C                   IN BOTH CASES THE LANCZS CODES WILL WORK WITH THE   LEV11760
C                   PERMUTED MATRICES AND THE PERMUTATION WILL BE       LEV11770
C                   UNDONE ONLY IN THE EIGENVECTOR PROGRAM AFTER        LEV11780
C                   THE RITZ VECTORS FOR THE PERMUTED PROBLEM HAVE      LEV11790
C                   BEEN COMPUTED.                                      LEV11800
C                                                                       LEV11810
C     OTHER PARAMETER CONTROLS ARE INTRODUCED TO ALLOW SEGMENTATION     LEV11820
C     OF THE EIGENVECTOR COMPUTATIONS AND TO ALLOW VARIOUS COMBINATIONS LEV11830
C     OF READ/WRITES.                                                   LEV11840
C                                                                       LEV11850
C     THE FLAG MBOUND ALLOWS THE USER TO DETERMINE A FIRST GUESS ON THE LEV11860
C     STORAGE THAT WILL BE REQUIRED BY THE T-EIGENVECTORS FOR THE       LEV11870
C     EIGENVALUES WHOSE EIGENVECTORS ARE TO BE COMPUTED.                LEV11880
C     THIS CAN BE USED TO ESTIMATE THE REQUIRED SIZE OF THE TVEC ARRAY. LEV11890
C                                                                       LEV11900
C     MBOUND = (0,1) MEANS                                              LEV11910
C                                                                       LEV11920
C              (0)  PROGRAM COMPUTES FIRST GUESSES AT THE SIZES         LEV11930
C                   OF THE T-MATRICES REQUIRED BY EACH OF THE           LEV11940
C                   EIGENVALUES SUPPLIED AND THEN CONTINUES WITH        LEV11950
C                   THE CORRESPONDING T-EIGENVECTOR COMPUTATIONS.       LEV11960
C                                                                       LEV11970
C              (1)  PROGRAM COMPUTES FIRST GUESSES AT THE SIZES         LEV11980
C                   OF THE T-MATRICES REQUIRED BY EACH OF THE           LEV11990
C                   EIGENVALUES SUPPLIED, STORES THESE IN FILE 10       LEV12000
C                   AND THEN TERMINATES.  THE USER CAN USE THESE        LEV12010
C                   SIZES TO ESTIMATE THE SIZE TVEC ARRAY NEEDED        LEV12020
C                   FOR THE DESIRED T-EIGENVECTOR COMPUTATIONS.         LEV12030
C                                                                       LEV12040
C     THE FLAGS NTVCON, TVSTOP, LVCONT, AND ERCONT CONTROL THE STOPPING LEV12050
C     CRITERIA FOR INTERMEDIATE POINTS IN THE LANCZOS PROCEDURE.  THEY  LEV12060
```

```
C     TERMINATE THE PROCEDURE IF VARIOUS QUANTITIES COULD NOT BE        LEV12070
C     COMPUTED AS DESIRED.                                              LEV12080
C                                                                       LEV12090
C     NTVCON = (0,1) MEANS                                              LEV12100
C                                                                       LEV12110
C              (0)  IF THE ESTIMATED STORAGE FOR THE T-EIGENVECTORS     LEV12120
C                   EXCEEDS THE USER-SPECIFIED DIMENSION OF THE         LEV12130
C                   TVEC ARRAY PROGRAM DOES NOT CONTINUE WITH THE       LEV12140
C                   T-EIGENVECTOR COMPUTATIONS.  TERMINATION OCCURS.    LEV12150
C                                                                       LEV12160
C              (1)  CONTINUE WITH THE T-EIGENVECTOR COMPUTATIONS        LEV12170
C                   EVEN IF THE ESTIMATED STORAGE FOR TVEC EXCEEDS      LEV12180
C                   THE USER-SPECIFIED DIMENSION OF THE TVEC ARRAY.     LEV12190
C                   IN THIS SITUATION THE PROGRAM COMPUTES AS MANY      LEV12200
C                   T-EIGENVECTORS AS IT HAS ROOM FOR, IN THE SAME      LEV12210
C                   ORDER IN WHICH THE EIGENVALUES ARE PROVIDED.        LEV12220
C                                                                       LEV12230
C      SVTVEC = (0,1) MEANS                                             LEV12240
C                                                                       LEV12250
C              (0)  DO NOT STORE THE COMPUTED T-EIGENVECTORS ON         LEV12260
C                   FILE 11 UNLESS ALSO HAVE THE FLAG TVSTOP = 1,       LEV12270
C                   IN WHICH CASE THE T-EIGENVECTORS ARE ALWAYS         LEV12280
C                   WRITTEN TO FILE 11.                                 LEV12290
C                                                                       LEV12300
C              (1)  STORE THE COMPUTED T-EIGENVECTORS ON FILE 11.       LEV12310
C                                                                       LEV12320
C      TVSTOP = (0,1) MEANS                                             LEV12330
C                                                                       LEV12340
C              (0)  ATTEMPT TO CONTINUE ON TO THE COMPUTATION           LEV12350
C                   OF THE RITZVECTORS AFTER COMPLETING THE             LEV12360
C                   COMPUTATION OF THE T-EIGENVECTORS.                  LEV12370
C                                                                       LEV12380
C              (1)  TERMINATE AFTER COMPUTING THE                       LEV12390
C                   T-EIGENVECTORS AND STORING THEM ON FILE 11.         LEV12400
C                                                                       LEV12410
C      LVCONT = (0,1) MEANS                                             LEV12420
C                                                                       LEV12430
C              (0)  IF SOME OF THE T-EIGENVECTORS THAT WERE             LEV12440
C                   REQUESTED WERE NOT COMPUTED, EXIT                   LEV12450
C                   FROM THE PROGRAM WITHOUT COMPUTING THE              LEV12460
C                   CORRESPONDING RITZ VECTORS.                         LEV12470
C                                                                       LEV12480
C              (1)  CONTINUE ON TO THE RITZ VECTOR COMPUTATIONS         LEV12490
C                   EVEN IF NOT ALL OF THE T-EIGENVECTORS               LEV12500
C                   REQUESTED WERE COMPUTED.                            LEV12510
C                                                                       LEV12520
C      ERCONT = (0,1) MEANS                                             LEV12530
C                                                                       LEV12540
C              (0)  PROCEDURE WILL NOT COMPUTE A RITZ VECTOR FOR        LEV12550
C                   ANY EIGENVALUE FOR WHICH NO T-EIGENVECTOR WHICH     LEV12560
C                   SATISFIES THE ERROR ESTIMATE TEST (ERTOL) HAS       LEV12570
C                   BEEN IDENTIFIED.                                    LEV12580
C                                                                       LEV12590
C              (1)  A RITZ VECTOR WILL BE COMPUTED FOR EVERY            LEV12600
C                   EIGENVALUE FOR WHICH A T-EIGENVECTOR HAS BEEN       LEV12610
```

```
C                    COMPUTED REGARDLESS OF WHETHER OR NOT THAT
C                    T-EIGENVECTOR SATISFIED THE ERROR ESTIMATE TEST.
C
C
C-----INPUT/OUTPUT FILES FOR THE EIGENVECTOR COMPUTATIONS---------------
C
C
C     ANY INPUT DATA OTHER THAN THE T-MATRIX HISTORY FILE AND THE
C     PREVIOUSLY COMPUTED EIGENVALUES AND CORRESPONDING ERROR
C     ESTIMATES SHOULD BE STORED ON FILE 5 IN FREE FORMAT.
C     SEE SAMPLE INPUT/OUTPUT FOR TYPICAL INPUT FILE.
C
C     FILE 6 WAS USED AS THE INTERACTIVE TERMINAL OUTPUT FILE.
C     THIS FILE PROVIDES A RUNNING ACCOUNT OF THE PROGRESS OF THE
C     COMPUTATIONS.  ADDITIONAL PRINTOUT IS GENERATED WHEN
C     THE FLAG IWRITE = 1.
C
C
C DESCRIPTION OF OTHER I/O FILES
C
C FILE (K)      CONTAINS:
C
C      (2)      INPUT FILE:
C               PREVIOUSLY-GENERATED T-MATRICES (ALPHA/BETA ARRAYS)
C               AND THE FINAL TWO LANCZOS VECTORS USED ON THAT
C               COMPUTATION.  THIS PROGRAM ALLOWS ENLARGEMENT
C               OF ANY T-MATRICES PROVIDED ON FILE 2.  NOTE THAT
C               IN CASE (4), THREE 'LANCZOS' VECTORS ARE ON FILE 2.
C
C      (3)      INPUT FILE:
C               THE GOOD EIGENVALUES OF THE T-MATRIX  T(1,MEV)
C               FOR WHICH RITZ VECTORS ARE REQUESTED.
C               FILE 3 ALSO CONTAINS THE T-MULTIPLICITIES OF THESE
C               EIGENVALUES AND THEIR COMPUTED GAPS IN THE
C               T-MATRICES AND IN THE USER-SUPPLIED MATRIX. IN
C               CASE (3) FILE 3 CONTAINS THE EIGENVALUES OF THE
C               B-INVERSE MATRIX AND 3 SETS OF CORRESPONDING GAPS.
C               THIS FILE IS CREATED IN THE LANCZOS EIGENVALUE
C               COMPUTATIONS.
C
C      (4)      INPUT FILE:
C               ERROR ESTIMATES FOR THE ISOLATED GOOD T-EIGENVALUES
C               ON FILE 3.  THIS FILE IS CREATED DURING THE LANCZOS
C               EIGENVALUE COMPUTATIONS.
C
C      (7)      INPUT FILE:
C               IN CASE (3) ((4)),
C               CONTAINS SPARSE MATRIX REPRESENTATION OF FACTORIZATION
C               OF MATRIX (B-MATRIX) USED BY LANCZS SUBROUTINE.
C
C      (8)      INPUT FILE:
C               IN CASES (1),(2) AND (4), USPEC SUBROUTINE ASSUMES
C               THAT USER-SUPPLIED A-MATRIX IS ON FILE 8.  IN CASE (3)
C               A-MATRIX CAN BE STORED ON FILE 8, BUT IT IS NOT
C               USED BY THE LANCZOS PROGRAMS.
```

```
C                                                                       LEV13170
C     (9)     OUTPUT FILE:                                              LEV13180
C             ERROR ESTIMATES FOR THE COMPUTED RITZ VECTORS CONSIDERED LEV13190
C             AS EIGENVECTORS OF THE MATRIX USED BY THE LANCZS         LEV13200
C             SUBROUTINE.  THESE ESTIMATES ARE OF THE FORM             LEV13210
C                   AERROR = || A*RITVEC - EVAL*RITVEC ||              LEV13220
C             WHERE A DENOTES THE MATRIX USED BY LANCZS, EVAL DENOTES  LEV13230
C             THE EIGENVALUE BEING CONSIDERED AND RITVEC DENOTES       LEV13240
C             THE COMPUTED RITZ VECTOR.                                LEV13250
C                                                                       LEV13260
C     (10)    OUTPUT FILE:                                              LEV13270
C             GUESSES AT APPROPRIATE SIZE T-MATRICES FOR THE           LEV13280
C             T-EIGENVECTORS FOR EACH SUPPLIED EIGENVALUE, GOODEV(J).  LEV13290
C                                                                       LEV13300
C     (11)    OUTPUT FILE:                                              LEV13310
C             COMPUTED T-EIGENVECTORS CORRESPONDING TO EIGENVALUES     LEV13320
C             IN THE GOODEV ARRAY.  NOTE THAT IT IS POSSIBLE IN        LEV13330
C             CERTAIN SITUATIONS THAT FOR SOME EIGENVALUES IN THE      LEV13340
C             GOODEV ARRAY A T-EIGENVECTOR WILL NOT BE COMPUTED.       LEV13350
C                                                                       LEV13360
C     (12)    OUTPUT FILE:                                              LEV13370
C             CONTAINS COMPUTED RITZ VECTORS CORRESPONDING TO          LEV13380
C             THE T-EIGENVECTORS ON FILE 11. NOTE THAT IN              LEV13390
C             SOME SITUATIONS THAT FOR SOME EIGENVALUES IN             LEV13400
C             THE GOODEV ARRAY FOR WHICH T-EIGENVECTORS HAVE           LEV13410
C             BEEN COMPUTED NO RITZ VECTOR WILL HAVE BEEN              LEV13420
C             COMPUTED.                                                LEV13430
C                                                                       LEV13440
C     (13)    OUTPUT FILE:                                              LEV13450
C             ADDITIONAL INFORMATION ABOUT THE BOUNDS AND ERROR         LEV13460
C             ESTIMATES OBTAINED.                                      LEV13470
C                                                                       LEV13480
C                                                                       LEV13490
C-----SEEDS FOR EIGENVECTOR PROGRAMS------------------------------------LEV13500
C                                                                       LEV13510
C     SEEDS FOR RANDOM NUMBER GENERATOR GENRAN                          LEV13520
C               (1) SVSEED = INTEGER*4 SCALAR USED IN THE SUBROUTINE   LEV13530
C                            GENRAN TO GENERATE THE STARTING VECTOR FORLEV13540
C                            THE REGENERATION OF THE LANCZOS VECTORS.  LEV13550
C                                                                       LEV13560
C               (2) RHSEED = INTEGER*4 SCALAR USED IN THE SUBROUTINE   LEV13570
C                            GENRAN TO GENERATE A RANDOM VECTOR FOR    LEV13580
C                            USE IN SUBROUTINE INVERM.                 LEV13590
C                                                                       LEV13600
C     USER SHOULD NOTE THAT SVSEED MUST BE THE SAME SEED THAT           LEV13610
C     WAS USED TO GENERATE THE T-MATRICES THAT WERE USED TO             LEV13620
C     COMPUTE THE EIGENVALUES WHOSE EIGENVECTORS ARE TO BE COMPUTED.    LEV13630
C     SVSEED IS READ IN FROM FILE 3.                                    LEV13640
C                                                                       LEV13650
C                                                                       LEV13660
C-----USER-SPECIFIED PARAMETERS FOR THE EIGENVECTOR PROGRAMS------------LEV13670
C                                                                       LEV13680
C                                                                       LEV13690
C     NGOOD   = NUMBER OF EIGENVALUES READ INTO THE GOODEV ARRAY        LEV13700
C               READ FROM FILE 3.                                       LEV13710
```

```
C                                                                      LEV13720
C     N       = SIZE OF THE USER-SUPPLIED MATRIX.                      LEV13730
C                                                                      LEV13740
C     MEV     = SIZE OF THE T-MATRIX THAT WAS USED TO COMPUTE          LEV13750
C               THE EIGENVALUES WHOSE EIGENVECTORS ARE REQUESTED.      LEV13760
C               MEV IS READ IN FROM FILE 3.                            LEV13770
C                                                                      LEV13780
C     KMAX    =  SIZE OF THE T-MATRIX PROVIDED ON FILE 2.              LEV13790
C                                                                      LEV13800
C     MDIMTV  = MAXIMUM CUMULATIVE SIZE OF THE TVEC ARRAY ALLOWED      LEV13810
C               FOR ALL OF THE T-EIGENVECTORS REQUIRED.  MDIMTV        LEV13820
C               MUST NOT EXCEED THE USER-SPECIFIED DIMENSION OF        LEV13830
C               THE TVEC ARRAY.  PROGRAM CAN BE RUN WITH THE FLAG      LEV13840
C               MBOUND = 1 TO DETERMINE AN EDUCATED GUESS ON AN        LEV13850
C               APPROPRIATE DIMENSION FOR THE TVEC ARRAY.              LEV13860
C                                                                      LEV13870
C     MDIMRV = MAXIMUM CUMULATIVE SIZE OF THE RITVEC ARRAY ALLOWED     LEV13880
C              FOR ALL OF THE RITZ VECTORS TO BE COMPUTED. MDIMRV      LEV13890
C              MUST NOT EXCEED THE USER-SPECIFIED DIMENSION OF         LEV13900
C              THE RITVEC ARRAY.  MUST BE SELECTED SO THAT             LEV13910
C              THERE IS ENOUGH ROOM FOR A RITZ VECTOR FOR EVERY        LEV13920
C              GOODEV(J) READ INTO PROGRAM.  (>= NGOOD*N)              LEV13930
C                                                                      LEV13940
C                                                                      LEV13950
C-----ARRAYS REQUIRED BY THE EIGENVECTOR PROGRAMS--------------------- LEV13960
C                                                                      LEV13970
C                                                                      LEV13980
C     ALL 4 CASES                                                      LEV13990
C                                                                      LEV14000
C     ALPHA(J) = REAL*8 ARRAY.  ITS DIMENSION MUST BE AT LEAST         LEV14010
C                KMAXN, THE LARGEST SIZE T-MATRIX CONSIDERED BY        LEV14020
C                THE PROGRAM.  NOTE THAT KMAXN IS THE LARGER OF        LEV14030
C                THE SIZE OF THE ALPHA, BETA HISTORY PROVIDED          LEV14040
C                ON FILE 2 (IF ANY ) AND THE SIZE WHICH THE PROGRAM    LEV14050
C                SPECIFIES INTERNALLY, THIS LATTER IS ALWAYS           LEV14060
C                < = 11*MEV / 8  +  12, WHERE MEV IS THE SIZE          LEV14070
C                T-MATRIX THAT WAS USED IN THE CORRESPONDING EIGENVALUE LEV14080
C                COMPUTATIONS.  ALPHA CONTAINS THE DIAGONAL ENTRIES    LEV14090
C                OF THE LANCZOS T-MATRICES.  ALPHA IS NOT DESTROYED    LEV14100
C                IN THE COMPUTATIONS.                                  LEV14110
C                                                                      LEV14120
C     BETA(J) = REAL*8 ARRAY.  ITS DIMENSION MUST BE AT LEAST 1        LEV14130
C               MORE THAN THAT OF ALPHA.  DIMENSION COMMENTS ABOVE     LEV14140
C               ABOUT ALPHA APPLY ALSO TO THE BETA ARRAY.  BETA        LEV14150
C               CONTAINS THE SUBDIAGONAL ENTRIES OF THE T-MATRICES.    LEV14160
C               BETA IS NOT DESTROYED IN THE COMPUTATIONS.             LEV14170
C                                                                      LEV14180
C     RITVEC(J) = REAL*8 ARRAY IN REAL SYMMETRIC AND INVERSE OF        LEV14190
C                 REAL SYMMETRIC CASES.  COMPLEX*16 IN CASE (2),       LEV14200
C                 HERMITIAN MATRICES.  IN EACH CASE ITS DIMENSION >=   LEV14210
C                 NGOOD*N WHERE N IS THE ORDER OF THE USER-SUPPLIED    LEV14220
C                 MATRIX AND NGOOD IS THE NUMBER OF EIGENVALUES WHOSE  LEV14230
C                 EIGENVECTORS ARE TO BE COMPUTED.  IT CONTAINS THE    LEV14240
C                 COMPUTED APPROXIMATE EIGENVECTORS OF A.  THESE       LEV14250
C                 COMPUTED RITZ VECTORS ARE STORED ON FILE 12.         LEV14260
```

```
C                                                                       LEV14270
C     TVEC(J)   = REAL*8 ARRAY.  ITS DIMENSION MUST BE AT LEAST         LEV14280
C                 MTOL = |MA(1)| + |MA(2)| + ... + |MA(NGOOD)|          LEV14290
C                 WHERE NGOOD IS THE NUMBER OF EIGENVALUES BEING        LEV14300
C                 CONSIDERED AND |MA(J)| IS THE SIZE OF THE             LEV14310
C                 T-MATRIX BEING USED FOR THE RITZ VECTOR               LEV14320
C                 COMPUTATION FOR GOODEV(J).  THESE SIZES               LEV14330
C                 ARE COMPUTED BY THE PROGRAM.  AN ESTIMATE OF          LEV14340
C                 MTOL CAN BE OBTAINED BY SETTING MBOUND = 1,           LEV14350
C                 RUNNING THE PROGRAM, AND MULTIPLYING THE RESULTING    LEV14360
C                 TOTAL OF THE T-SIZES SPECIFIED BY 5/4.  THE ARRAY     LEV14370
C                 TVEC CONTAINS THE COMPUTED T-EIGENVECTORS.  IF THE    LEV14380
C                 FLAG SVTVEC = 1 OR THE FLAG TVSTOP = 1, THEN          LEV14390
C                 THESE VECTORS ARE SAVED ON FILE 11.                   LEV14400
C                                                                       LEV14410
C     V1(J)     = REAL*8 ARRAY IN REAL SYMMETRIC AND INVERSE OF         LEV14420
C                 REAL SYMMETRIC CASES.  COMPLEX*16 IN CASE (2),        LEV14430
C                 HERMITIAN MATRICES.  IN THE REAL CASES ITS            LEV14440
C                 DIMENSION MUST BE THE MAXIMUM OF KMAX AND N.          LEV14450
C                 IN THE HERMITIAN CASE ITS DIMENSION MUST BE           LEV14460
C                 THE MAXIMUM OF KMAX/2 AND N WHERE KMAX IS THE         LEV14470
C                 LARGEST SIZE T-MATRIX THAT IS TO BE CONSIDERED        LEV14480
C                 IN THE T-EIGENVECTOR COMPUTATIONS.  V1 IS USED        LEV14490
C                 IN THE SUBROUTINE INVERM AND IN THE REGENERATION      LEV14500
C                 OF THE LANCZOS VECTORS.                               LEV14510
C                                                                       LEV14520
C     V2(J)     = REAL*8 ARRAY IN THE REAL SYMMETRIC AND INVERSE        LEV14530
C                 OF REAL SYMMETRIC CASES.  COMPLEX*16 IN CASE (2),     LEV14540
C                 HERMITIAN MATRICES.  IN CASES (1),(3) AND (4), ITS    LEV14550
C                 DIMENSION MUST BE > = MAX(KMAX,N); IN CASE (2)        LEV14560
C                 > = MAX(KMAX/2,N).  IT IS USED IN THE REGENERATION    LEV14570
C                 OF THE LANCZOS VECTORS AND IN SUBROUTINE INVERM.      LEV14580
C                                                                       LEV14590
C     GOODEV(J),  = REAL*8 ARRAYS EACH OF DIMENSION AT LEAST NGOOD.     LEV14600
C     EVNEW(J)      CONTAIN THE EIGENVALUES FOR WHICH EIGENVECTORS      LEV14610
C                   ARE REQUESTED.  EIGENVALUES IN GOODEV ARE READ      LEV14620
C                   IN FROM FILE 3.  IN CASE (3) GOODEV IS REPLACED     LEV14630
C                   BY GOODA AND GOODBI ARRAYS, SEE BELOW.              LEV14640
C                                                                       LEV14650
C     AMINGP(J), = REAL*4 ARRAYS OF DIMENSION AT LEAST NGOOD.           LEV14660
C     TMINGP(J)    CONTAIN, RESPECTIVELY, THE MINIMAL GAPS FOR          LEV14670
C                  CORRESPONDING EIGENVALUES IN GOODEV ARRAY IN         LEV14680
C                  A-MATRIX AND IN T-MATRIX.                            LEV14690
C                                                                       LEV14700
C     TERR(J), ERR(J),     = REAL*4 ARRAYS (EXCEPT TLAST WHICH IS       LEV14710
C     ERRDGP(J), TLAST(J)    REAL*8).  EACH MUST BE OF DIMENSION        LEV14720
C     RNORM(J), TBETA(J)     AT LEAST NGOOD.  USED TO STORE QUANTITIES  LEV14730
C                            GENERATED DURING THE COMPUTATIONS FOR      LEV14740
C                            LATER PRINTOUT.                            LEV14750
C                                                                       LEV14760
C     G(J)     = REAL*4 ARRAY WHOSE DIMENSION MUST BE AT LEAST          LEV14770
C                MAX(KMAX,N).  USED IN SUBROUTINE GENRAN TO HOLD        LEV14780
C                RANDOM NUMBERS NEEDED FOR THE LANCZOS VECTOR           LEV14790
C                REGENERATION AND FOR THE INVERSE ITERATION             LEV14800
C                COMPUTATIONS IN THE SUBROUTINE INVERM.                 LEV14810
```

```
C
C     MP(J) = INTEGER*4 ARRAY WHOSE DIMENSION IS AT LEAST NGOOD.
C             INITIALLY CONTAINS THE MULTIPLICITY OF THE EIGENVALUE
C             GOODEV(J) AS AN EIGENVALUE OF THE T-MATRIX T(1,MEV).
C             USED TO FLAG EIGENVALUES FOR WHICH NO T-EIGENVECTOR
C             OR NO RITZ VECTOR IS TO BE COMPUTED.
C
C     MA(J)   = INTEGER*4 ARRAYS EACH OF WHOSE DIMENSIONS
C               IS AT LEAST NGOOD.  USED IN DETERMINING
C               AN APPROPRIATE T-MATRIX FOR EACH EIGENVALUE
C               IN GOODEV ARRAY.
C
C     MINT(J),MFIN(J) = INTEGER*4 ARRAYS WHOSE DIMENSIONS MUST BE AT
C                       LEAST NGOOD.  USED TO POINT TO THE BEGINNINGS
C                       AND THE ENDS OF THE COMPUTED EIGENVECTOR
C                       OF THE T-MATRIX, T(1,|MA(J)|).
C
C     IDELTA(J)  = INTEGER*4 ARRAY WHOSE DIMENSION MUST BE AT
C                  LEAST NGOOD.  CONTAINS INCREMENTS USED IN LOOPS
C                  ON APPROPRIATE SIZE T-MATRIX FOR THE T-EIGENVECTOR
C                  COMPUTATIONS.
C
C
C     CASE (2) ONLY, HERMITIAN MATRICES:
C
C     GR(J),GC(J)     = REAL*8 ARRAYS USED ONLY IN CASE (2),
C                       HERMITIAN MATRICES.  EACH MUST BE AT
C                       LEAST MAX(N,KMAX).  USED TO HOLD
C                       STARTING VECTORS FOR LANCZS
C                       COMPUTATIONS AND FOR INVERM SUBROUTINES.
C
C     CASES (3) AND (4) ONLY, FACTORED INVERSES OF REAL SYMMETRIC
C     MATRICES AND GENERALIZED EIGENVALUE PROBLEMS:
C
C     VS(J) =  REAL*8 ARRAY WHOSE DIMENSION MUST BE AT LEAST N.
C              USED IN REGENERATION OF THE LANCZOS VECTORS.
C
C     IPR(J), IPT(J) = INTEGER*4 ARRAYS.  EACH MUST BE OF DIMENSION
C                      AT LEAST N, THE ORDER OF A. USED TO STORE
C                      THE REORDERING OF THE GIVEN MATRIX.
C
C     CASE (3) ONLY, INVERSES OF REAL SYMMETRIC MATRICES:
C
C     GOODA(J), GOODBI(J) = REAL*8 ARRAYS.  EACH MUST BE OF DIMENSION
C                           AT LEAST NGOOD, THE NUMBER OF EIGENVALUES
C                           BEING CONSIDERED.  GOODA CONTAINS THE
C                           EIGENVALUES OF A AND GOODBI CONTAINS THE
C                           EIGENVALUES OF B(INVERSE).  THE PROGRAM
C                           WORKS DIRECTLY WITH THE GOODBI ARRAY.
C
C
C-----SUBROUTINES INCLUDED FOR THE EIGENVECTOR COMPUTATIONS-------------
C
C
C     STURMI = FOR EACH GIVEN EIGENVALUE GOODEV(J) DETERMINES
```

```
C            THE SMALLEST SIZE T-MATRIX FOR WHICH GOODEV(J) IS        LEV15370
C            AN EIGENVALUE (TO WITHIN A GIVEN TOLERANCE) AND IF       LEV15380
C            POSSIBLE THE SMALLEST SIZE T-MATRIX FOR WHICH            LEV15390
C            IT IS A DOUBLE EIGENVALUE (TO WITHIN THE SAME            LEV15400
C            TOLERANCE).  THE SIZE T-MATRIX USED IN THE               LEV15410
C            EIGENVECTOR COMPUTATIONS IS THEN DETERMINED BY           LEV15420
C            STARTING WITH AN INITIAL GUESS BASED UPON THE            LEV15430
C            INFORMATION FROM STURMI, AND LOOPING ON THE SIZE         LEV15440
C            OF THE T-EIGENVECTOR COMPUTATIONS.                       LEV15450
C                                                                     LEV15460
C     LBISEC = RECOMPUTES THE VALUE OF THE GIVEN EIGENVALUE AT THE    LEV15470
C            T-SIZE SPECIFIED FOR THE T-EIGENVECTOR COMPUTATION.      LEV15480
C            LBISEC IS A SIMPLIFICATION OF THE BISEC SUBROUTINE       LEV15490
C            USED IN THE LANCZOS EIGENVALUE COMPUTATIONS.             LEV15500
C                                                                     LEV15510
C     INVERM = FOR THE T-SIZES CONSIDERED BY THE PROGRAM COMPUTES     LEV15520
C            THE CORRESPONDING EIGENVECTORS OF THESE T-MATRICES       LEV15530
C            CORRESPONDING TO THE USER-SUPPLIED EIGENVALUES IN        LEV15540
C            THE GOODEV ARRAY.                                        LEV15550
C                                                                     LEV15560
C     THE LANCZS, TNORM , AND CINPRD (CASE (2) ONLY) SUBROUTINES      LEV15570
C     ARE USED HERE AS WELL AS IN THE EIGENVALUE COMPUTATIONS.        LEV15580
C                                                                     LEV15590
C     IN CASES (3) AND (4) ONLY AND THEN ONLY IF THE ORIGINAL MATRIX  LEV15600
C     (MATRICES) HAS (HAVE) BEEN PERMUTED:                            LEV15610
C                                                                     LEV15620
C     LPERM  = (USED IN CASE (3) AND (4) ONLY). GIVEN A B-MATRIX AND  LEV15630
C            A PERMUTATION P DEFINED IN THE VECTORS IPR AND IPT,      LEV15640
C            AND A VECTOR X COMPUTE EITHER (P-TRANSPOSE)*X OR PX.     LEV15650
C                                                                     LEV15660
C---------------------------------------------------------------------LEV15670
```

2.3 LEVAL: Main Program, Eigenvalue Computations

```
C-----LEVAL  (EIGENVALUES OF REAL SYMMETRIC MATRICES)-------------------LEV00010
C  Authors:  Jane Cullum and Ralph A. Willoughby (Deceased)             LEV00020
C            Los Alamos National Laboratory                             LEV00030
C            Los Alamos, New Mexico 87544                               LEV00040
C                                                                       LEV00050
C            E-mail:  cullumj@lanl.gov                                  LEV00060
C                                                                       LEV00070
C  These codes are copyrighted by the authors.  These codes             LEV00080
C  and modifications of them or portions of them are NOT to be          LEV00090
C  incorporated into any commercial codes or used for any other         LEV00100
C  commercial purposes such as consulting for other companies,          LEV00110
C  without legal agreements with the authors of these Codes.            LEV00120
C  If these Codes or portions of them are used in other scientific or   LEV00130
C  engineering research works the names of the authors of these codes   LEV00140
C  and appropriate references to their written work are to be           LEV00150
C  incorporated in the derivative works.                                LEV00160
C                                                                       LEV00170
C  This header is not to be removed from these codes.                   LEV00180
C                                                                       LEV00190
C         REFERENCE: Cullum and Willoughby, Chapters 1,2,3,4            LEV00191
C         Lanczos Algorithms for Large Symmetric Eigenvalue ComputationsLEV00192
C         VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in   LEV00193
C         Applied Mathematics, 2002. SIAM Publications,                 LEV00194
C         Philadelphia, PA. USA                                         LEV00195
C                                                                       LEV00196
C                                                                       LEV00200
C      CONTAINS MAIN PROGRAM FOR COMPUTING DISTINCT EIGENVALUES OF      LEV00210
C      A REAL SYMMETRIC MATRIX USING LANCZOS TRIDIAGONALIZATION         LEV00220
C      WITHOUT REORTHOGONALIZATION.                                     LEV00230
C                                                                       LEV00240
C      PFORT VERIFIER IDENTIFIED THE FOLLOWING NONPORTABLE              LEV00250
C      CONSTRUCTIONS                                                    LEV00260
C                                                                       LEV00270
C      1.  DATA/MACHEP/ STATEMENT                                       LEV00280
C      2.  ALL READ(5,*) STATEMENTS (FREE FORMAT)                       LEV00290
C      3.  FORMAT(20A4) USED WITH EXPLANATORY HEADER EXPLAN.            LEV00300
C      4.  HEXADECIMAL FORMAT (4Z20) USED IN ALPHA/BETA FILES 1 AND 2.  LEV00310
C                                                                       LEV00320
C-----------------------------------------------------------------------LEV00330
C                                                                       LEV00340
      DOUBLE PRECISION  ALPHA(5000),BETA(5001)                          LEV00350
      DOUBLE PRECISION  V1(5001),V2(5000),VS(5000)                      LEV00360
      DOUBLE PRECISION  LB(20),UB(20)                                   LEV00370
      DOUBLE PRECISION  BTOL,GAPTOL,TTOL,MACHEP,EPSM,RELTOL             LEV00380
      DOUBLE PRECISION  SCALE1,SCALE2,SCALE3,SCALE4,BISTOL,CONTOL,MULTOLLEV00390
      DOUBLE PRECISION  ONE,ZERO,TEMP,TKMAX,BETAM,BKMIN,T0,T1           LEV00400
      REAL  G(10000),EXPLAN(20)                                         LEV00410
      INTEGER  MP(5000),NMEV(20)                                        LEV00420
      INTEGER  SVSEED,RHSEED,SVSOLD                                     LEV00430
      INTEGER IABS                                                      LEV00440
      REAL  ABS                                                         LEV00450
      DOUBLE PRECISION  DABS, DSQRT, DFLOAT                             LEV00460
```

```
      EXTERNAL CMATV                                                    LEV00470
C                                                                       LEV00480
C-----------------------------------------------------------------------LEV00490
      DATA MACHEP/Z3410000000000000/                                    LEV00500
      EPSM = 2.0D0*MACHEP                                               LEV00510
C-----------------------------------------------------------------------LEV00520
C                                                                       LEV00530
C     ARRAYS MUST BE DIMENSIONED AS FOLLOWS:                            LEV00540
C     DIMENSION OF V2 ASSUMES THAT NO MORE THAN KMAX/2 EIGENVALUES      LEV00550
C     OF THE T-MATRICES ARE BEING COMPUTED IN ANY ONE OF THE            LEV00560
C     SUB-INTERVALS BEING CONSIDERED.  V2 CONTAINS THE UPPER AND LOWER  LEV00570
C     BOUNDS FOR EACH T-EIGENVALUE BEING COMPUTED BY BISEC IN ANY ONE   LEV00580
C     GIVEN INTERVAL.                                                   LEV00590
C                                                                       LEV00600
C     1.  ALPHA: >= KMAX,   BETA: >= (KMAX+1)                           LEV00610
C     2.  V1:  >= MAX(N,KMAX+1)                                         LEV00620
C     3.  V2:   >= MAX(N,KMAX)                                          LEV00630
C     4.  VS:  >= KMAX                                                  LEV00640
C     5.  G:  >= MAX(N,2*KMAX)                                          LEV00650
C     6.  MP:    >= KMAX                                                LEV00660
C     7.  LB,UB:  >= NUMBER OF SUBINTERVALS SUPPLIED TO BISEC.          LEV00670
C     8.  NMEV:  >= NUMBER OF T-MATRICES ALLOWED.                       LEV00680
C     9.  EXPLAN:  DIMENSION IS 20.                                     LEV00690
C                                                                       LEV00700
C                                                                       LEV00710
C     IMPORTANT TOLERANCES OR SCALES THAT ARE USED REPEATEDLY           LEV00720
C     THROUGHOUT THIS PROGRAM ARE THE FOLLOWING:                        LEV00730
C     SCALED MACHINE EPSILON:  TTOL = TKMAX*EPSM WHERE                  LEV00740
C     EPSM = 2*MACHINE EPSILON AND                                      LEV00750
C     TKMAX = MAX(|ALPHA(J)|,BETA(J), J = 1,MEV)                        LEV00760
C     BISEC CONVERGENCE TOLERANCE:  BISTOL = DSQRT(1000+MEV)*TTOL       LEV00770
C     BISEC T-MULTIPLICITY TOLERANCE:  MULTOL = (1000+MEV)*TTOL         LEV00780
C     LANCZOS CONVERGENCE TOLERANCE:   CONTOL = BETA(MEV+1)*1.D-10      LEV00790
C-----------------------------------------------------------------------LEV00800
C     OUTPUT HEADER                                                     LEV00810
      WRITE(6,10)                                                       LEV00820
   10 FORMAT(/' LANCZOS PROCEDURE FOR REAL SYMMETRIC MATRICES'/)        LEV00830
C                                                                       LEV00840
C     SET PROGRAM PARAMETERS                                            LEV00850
C     SCALEK ARE USED IN TOLERANCES NEEDED IN SUBROUTINES LUMP,         LEV00860
C     ISOEV AND PRTEST.  USER MUST NOT MODIFY THESE SCALES.             LEV00870
      SCALE1 = 5.0D2                                                    LEV00880
      SCALE2 = 5.0D0                                                    LEV00890
      SCALE3 = 5.0D0                                                    LEV00900
      SCALE4 = 1.0D4                                                    LEV00910
      ONE  = 1.0D0                                                      LEV00920
      ZERO = 0.0D0                                                      LEV00930
      BTOL = 1.0D-8                                                     LEV00940
C     BTOL = EPSM                                                       LEV00950
      GAPTOL = 1.0D-8                                                   LEV00960
      ICONV = 0                                                         LEV00970
      MOLD = 0                                                          LEV00980
      MOLD1 = 1                                                         LEV00990
      ICT = 0                                                           LEV01000
      MMB = 0                                                           LEV01010
```

```
      IPROJ = 0                                                         LEV01020
C-----------------------------------------------------------------------LEV01030
C     READ USER-SPECIFIED PARAMETERS FROM INPUT FILE 5 (FREE FORMAT)    LEV01040
C                                                                       LEV01050
C     READ USER-PROVIDED HEADER FOR RUN                                 LEV01060
      READ(5,20) EXPLAN                                                 LEV01070
      WRITE(6,20) EXPLAN                                                LEV01080
      READ(5,20) EXPLAN                                                 LEV01090
      WRITE(6,20) EXPLAN                                                LEV01100
   20 FORMAT(20A4)                                                      LEV01110
C                                                                       LEV01120
C     READ ORDER OF MATRICES (N) , MAXIMUM ORDER OF T-MATRIX (KMAX),     LEV01130
C     NUMBER OF T-MATRICES ALLOWED (NMEVS), AND MATRIX IDENTIFICATION   LEV01140
C     NUMBERS (MATNO)                                                   LEV01150
      READ(5,20) EXPLAN                                                 LEV01160
      READ(5,*) N,KMAX,NMEVS,MATNO                                      LEV01170
C                                                                       LEV01180
C     READ SEEDS FOR LANCZS AND INVERR SUBROUTINES (SVSEED AND RHSEED)  LEV01190
C     READ MAXIMUM NUMBER OF ITERATIONS ALLOWED FOR EACH INVERSE        LEV01200
C     ITERATION (MXINIT) AND MAXIMUM NUMBER OF STURM SEQUENCES          LEV01210
C     ALLOWED (MXSTUR)                                                  LEV01220
      READ(5,20) EXPLAN                                                 LEV01230
      READ(5,*) SVSEED,RHSEED,MXINIT,MXSTUR                             LEV01240
C                                                                       LEV01250
C     ISTART = (0,1):  ISTART = 0 MEANS ALPHA/BETA FILE IS NOT          LEV01260
C     AVAILABLE.  ISTART = 1 MEANS ALPHA/BETA FILE IS AVAILABLE ON      LEV01270
C     FILE 2.                                                           LEV01280
C     ISTOP = (0,1):  ISTOP = 0 MEANS PROCEDURE GENERATES ALPHA/BETA    LEV01290
C     FILE AND THEN TERMINATES.  ISTOP = 1 MEANS PROCEDURE GENERATES    LEV01300
C     ALPHAS/BETAS IF NEEDED AND THEN COMPUTES EIGENVALUES AND ERROR    LEV01310
C     ESTIMATES AND THEN TERMINATES.                                    LEV01320
      READ(5,20) EXPLAN                                                 LEV01330
      READ(5,*) ISTART,ISTOP                                            LEV01340
C                                                                       LEV01350
C     IHIS = (0,1):  IHIS = 0 MEANS ALPHA/BETA FILE IS NOT WRITTEN      LEV01360
C     TO FILE 1.  IHIS = 1 MEANS ALPHA/BETA FILE IS WRITTEN TO FILE 1.  LEV01370
C     IDIST = (0,1):  IDIST = 0 MEANS DISTINCT T-EIGENVALUES            LEV01380
C     ARE NOT WRITTEN TO FILE 11.  IDIST = 1 MEANS DISTINCT             LEV01390
C     T-EIGENVALUES ARE WRITTEN TO FILE 11.                             LEV01400
C     IWRITE = (0,1):  IWRITE = 0 MEANS NO INTERMEDIATE OUTPUT          LEV01410
C     FROM THE COMPUTATIONS IS WRITTEN TO FILE 6.  IWRITE = 1 MEANS     LEV01420
C     T-EIGENVALUES AND ERROR ESTIMATES ARE WRITTEN TO FILE 6           LEV01430
C     AS THEY ARE COMPUTED.                                             LEV01440
      READ(5,20) EXPLAN                                                 LEV01450
      READ(5,*) IHIS,IDIST,IWRITE                                       LEV01460
C                                                                       LEV01470
C     READ IN THE RELATIVE TOLERANCE (RELTOL) FOR USE IN THE            LEV01480
C     SPURIOUS, T-MULTIPLICITY, AND PRTESTS.                            LEV01490
      READ(5,20) EXPLAN                                                 LEV01500
      READ(5,*) RELTOL                                                  LEV01510
C                                                                       LEV01520
C     READ IN THE SIZES OF THE T-MATRICES TO BE CONSIDERED.             LEV01530
      READ(5,20) EXPLAN                                                 LEV01540
      READ(5,*) (NMEV(J), J=1,NMEVS)                                    LEV01550
C                                                                       LEV01560
```

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

```
C                                                                       LEV02120
      IF (ISTART.EQ.0) GO TO 140                                        LEV02130
C                                                                       LEV02140
C     READ IN ALPHA BETA HISTORY                                        LEV02150
C                                                                       LEV02160
      READ(2,100)MOLD,NOLD,SVSOLD,MATOLD                                LEV02170
  100 FORMAT(2I6,I12,I8)                                                LEV02180
C                                                                       LEV02190
      IF (KMAX.LT.MOLD) KMAX = MOLD                                     LEV02200
      KMAX1 = KMAX + 1                                                  LEV02210
C                                                                       LEV02220
C     CHECK THAT ORDER N, MATRIX ID MATNO, AND RANDOM SEED SVSEED       LEV02230
C     AGREE WITH THOSE IN THE HISTORY FILE.  IF NOT PROCEDURE STOPS.    LEV02240
C                                                                       LEV02250
      ITEMP = (NOLD-N)**2+(MATNO-MATOLD)**2+(SVSEED-SVSOLD)**2          LEV02260
C                                                                       LEV02270
      IF (ITEMP.EQ.0) GO TO 120                                         LEV02280
C                                                                       LEV02290
      WRITE(6,110)                                                      LEV02300
  110 FORMAT(' PROGRAM TERMINATES'/    ' READ FROM FILE 2 CORRESPONDS TOLEV02310
     1 DIFFERENT MATRIX THAN MATRIX SPECIFIED'/)                        LEV02320
      GO TO 640                                                         LEV02330
C                                                                       LEV02340
  120 CONTINUE                                                          LEV02350
      MOLD1 = MOLD+1                                                    LEV02360
C                                                                       LEV02370
      READ(2,130)(ALPHA(J), J=1,MOLD)                                   LEV02380
      READ(2,130)(BETA(J), J=1,MOLD1)                                   LEV02390
  130 FORMAT(4Z20)                                                      LEV02400
C                                                                       LEV02410
      IF (KMAX.EQ.MOLD) GO TO 160                                       LEV02420
C                                                                       LEV02430
      READ(2,130)(V1(J), J=1,N)                                         LEV02440
      READ(2,130)(V2(J), J=1,N)                                         LEV02450
C                                                                       LEV02460
  140 CONTINUE                                                          LEV02470
      IIX = SVSEED                                                      LEV02480
C                                                                       LEV02490
C-----------------------------------------------------------------------LEV02500
C                                                                       LEV02510
      CALL LANCZS(CMATV,ALPHA,BETA,V1,V2,G,KMAX,MOLD1,N,IIX)            LEV02520
C                                                                       LEV02530
C-----------------------------------------------------------------------LEV02540
C                                                                       LEV02550
      KMAX1 = KMAX + 1                                                  LEV02560
C                                                                       LEV02570
      IF (IHIS.EQ.0.AND.ISTOP.GT.0) GO TO 160                           LEV02580
C                                                                       LEV02590
      WRITE(1,150) KMAX,N,SVSEED,MATNO                                  LEV02600
  150 FORMAT(2I6,I12,I8,' = KMAX,N,SVSEED,MATNO')                       LEV02610
C                                                                       LEV02620
      WRITE(1,130)(ALPHA(I), I=1,KMAX)                                  LEV02630
      WRITE(1,130)(BETA(I), I=1,KMAX1)                                  LEV02640
C                                                                       LEV02650
      WRITE(1,130)(V1(I), I=1,N)                                        LEV02660
```

```
      WRITE(1,130)(V2(I), I=1,N)
C
      IF (ISTOP.EQ.0) GO TO 540
C
  160 CONTINUE
      BKMIN = BTOL
C
      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                                                                       LEV03220
      CALL TNORM(ALPHA,BETA,T0,T1,MEV,IBMEV)                            LEV03230
C                                                                       LEV03240
C-----------------------------------------------------------------------LEV03250
C                                                                       LEV03260
      TEMP = T0/TKMAX                                                   LEV03270
      IBMEV = IABS(IBMEV)                                               LEV03280
      IF (TEMP.GE.BTOL) GO TO 210                                       LEV03290
      IBMEV = -IBMEV                                                    LEV03300
      GO TO 600                                                         LEV03310
C                                                                       LEV03320
  210 CONTINUE                                                          LEV03330
      IC = MXSTUR-ICT                                                   LEV03340
C                                                                       LEV03350
C-----------------------------------------------------------------------LEV03360
C     BISEC LOOP. THE SUBROUTINE BISEC INCORPORATES DIRECTLY THE        LEV03370
C     T-MULTIPLICITY AND SPURIOUS TESTS. T-EIGENVALUES WILL BE          LEV03380
C     CALCULATED BY BISEC SEQUENTIALLY ON INTERVALS                     LEV03390
C     (LB(J),UB(J)), J = 1,NINT).                                       LEV03400
C                                                                       LEV03410
C     ON RETURN FROM BISEC                                              LEV03420
C     NDIS = NUMBER OF DISTINCT EIGENVALUES OF T(1,MEV) ON UNION        LEV03430
C            OF THE (LB,UB) INTERVALS                                   LEV03440
C     VS = DISTINCT T-EIGENVALUES IN ALGEBRAICALLY INCREASING ORDER     LEV03450
C     MP = MULTIPLICITIES OF THE T-EIGENVALUES IN VS                    LEV03460
C     MP(I) = (0,1,MI), MI>1, I=1,NDIS  MEANS:                          LEV03470
C        (0)  VS(I) IS SPURIOUS                                         LEV03480
C        (1)  VS(I) IS T-SIMPLE AND GOOD                                LEV03490
C        (MI) VS(I) IS MULTIPLE AND IS THEREFORE NOT ONLY GOOD BUT      LEV03500
C             ALSO A CONVERGED GOOD T-EIGENVALUE.                       LEV03510
C                                                                       LEV03520
C                                                                       LEV03530
      CALL BISEC(ALPHA,BETA,V1,V2,VS,LB,UB,EPSM,TTOL,MP,NINT,           LEV03540
     1 MEV,NDIS,IC,IWRITE)                                              LEV03550
C                                                                       LEV03560
C-----------------------------------------------------------------------LEV03570
C                                                                       LEV03580
      IF (NDIS.EQ.0) GO TO 620                                          LEV03590
C                                                                       LEV03600
C     COMPUTE THE TOTAL NUMBER OF STURM SEQUENCES USED TO DATE          LEV03610
C     COMPUTE THE BISEC CONVERGENCE AND T-MULTIPLICITY TOLERANCES USED. LEV03620
C     COMPUTE THE CONVERGENCE TOLERANCE FOR EIGENVALUES OF A.           LEV03630
      ICT = ICT + IC                                                    LEV03640
      TEMP = DFLOAT(MEV+1000)                                           LEV03650
      MULTOL = TEMP*TTOL                                                LEV03660
      TEMP = DSQRT(TEMP)                                                LEV03670
      BISTOL = TTOL*TEMP                                                LEV03680
      CONTOL = BETAM*1.D-10                                             LEV03690
C                                                                       LEV03700
C-----------------------------------------------------------------------LEV03710
C    SUBROUTINE LUMP 'COMBINES' T-EIGENVALUES THAT ARE 'TOO CLOSE'.     LEV03720
C    NOTE HOWEVER THAT CLOSE SPURIOUS T-EIGENVALUES ARE NOT AVERAGED    LEV03730
C    WITH GOOD ONES. HOWEVER, THEY MAY BE USED TO INCREASE THE          LEV03740
C    MULTIPLICITY OF A GOOD T-EIGENVALUE.                               LEV03750
C                                                                       LEV03760
```

```
      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,MATNO
  250 FORMAT(/4I6,I12,I8,' = NDIS,NISO,MEV,N,SVSEED,MATNO'/)
C
      WRITE(11,260) (MP(I),VS(I),G(I), I=1,NDIS)
  260 FORMAT(2(I3,E25.16,E12.3))
C
      WRITE(11,270) NDIS, (MP(I), I=1,NDIS)
  270 FORMAT(/I6,' = NDIS, T-MULTIPLICITIES (0 MEANS  SPURIOUS)'/(20I4))
C
  280 CONTINUE
C
      IF (NISO.NE.0) GO TO 310
C
      WRITE(4,290) MEV
```

```
  290 FORMAT(/' AT MEV = ',I6,' THERE ARE NO ISOLATED T-EIGENVALUES'/
     1' SO NO ERROR ESTIMATES WERE COMPUTED/')
C
      WRITE(6,300)
  300 FORMAT(/' ALL COMPUTED GOOD T-EIGENVALUES ARE MULTIPLE'/
     1 ' THEREFORE ALL SUCH EIGENVALUES ARE ASSUMED TO HAVE CONVERGED')
C
      ICONV = 1
      GO TO 350
C
  310 CONTINUE
C
C-----------------------------------------------------------------------
C     SUBROUTINE INVERR COMPUTES ERROR ESTIMATES FOR ISOLATED GOOD
C     T-EIGENVALUES USING INVERSE ITERATION ON T(1,MEV). ON RETURN
C     G(J) = MINIMUM GAP IN T(1,MEV) FOR EACH VS(J), J=1,NDIS
C     G(MEV+I) = BETAM*|U(MEV)| = ERROR ESTIMATE FOR ISOLATED GOOD
C              T-EIGENVALUES, WHERE I = 1, NISO AND  BETAM = BETA(MEV+1)
C              U(MEV) IS MEVTH COMPONENT OF THE UNIT EIGENVECTOR OF T
C              CORRESPONDING TO THE ITH ISOLATED GOOD T-EIGENVALUE.
C     A NEGATIVE ERROR ESTIMATE MEANS THAT FOR THAT PARTICULAR
C     EIGENVALUE THE INVERSE ITERATION DID NOT CONVERGE IN <= MXINIT
C     STEPS AND THAT THE CORRESPONDING ERROR ESTIMATE IS QUESTIONABLE.
C
C     V2 CONTAINS THE ISOLATED GOOD T-EIGENVALUES
C     V1 CONTAINS THE MINGAPS TO THE NEAREST DISTINCT  EIGENVALUE
C        OF T(1,MEV) FOR EACH ISOLATED GOOD 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'/)                               LEV04870
C                                                                       LEV04880
  350 CONTINUE                                                          LEV04890
C                                                                       LEV04900
C     IF CONVERGENCE IS INDICATED, THAT IS ICONV = 1 ,THEN              LEV04910
C     THE SUBROUTINE PRTEST IS CALLED TO CHECK FOR ANY CONVERGED        LEV04920
C     T-EIGENVALUES THAT HAVE BEEN MISLABELLED AS SPURIOUS BECAUSE      LEV04930
C     THE PROJECTION OF THEIR EIGENVECTOR(S) ON THE STARTING            LEV04940
C     VECTOR WERE TOO SMALL.                                            LEV04950
C     NUMERICAL TESTS INDICATE THAT SUCH EIGENVALUES ARE RARE.          LEV04960
C     IF FOR SOME REASON MANY OF THESE HIDDEN EIGENVALUES APPEAR        LEV04970
C     ON SOME RUN, YOU CAN BE CERTAIN THAT SOMETHING IS FOULED UP.      LEV04980
C                                                                       LEV04990
      IF (ICONV.EQ.0) GO TO 480                                         LEV05000
C                                                                       LEV05010
C-----------------------------------------------------------------------LEV05020
C                                                                       LEV05030
      CALL PRTEST (ALPHA,BETA,VS,TKMAX,EPSM,RELTOL,SCALE3,SCALE4,       LEV05040
     1 MP,NDIS,MEV,IPROJ)                                               LEV05050
C                                                                       LEV05060
C-----------------------------------------------------------------------LEV05070
C                                                                       LEV05080
      IF(IPROJ.EQ.0) GO TO 470                                          LEV05090
C                                                                       LEV05100
      IF(IDIST.EQ.1)  WRITE(11,360) IPROJ                               LEV05110
  360 FORMAT(' SUBROUTINE PRTEST WANTS TO RELABEL',I6,' SPURIOUS T-EIGENLEV05120
     1VALUES'/' WE ACCEPT RELABELLING ONLY IF LAST COMPONENT OF T-EIGENVLEV05130
     1ECTOR IS L.T. 1.D-10'/)                                           LEV05140
C                                                                       LEV05150
      IIX = RHSEED                                                      LEV05160
C                                                                       LEV05170
C-----------------------------------------------------------------------LEV05180
C                                                                       LEV05190
      CALL GENRAN(IIX,G,MEV)                                            LEV05200
C                                                                       LEV05210
C-----------------------------------------------------------------------LEV05220
C                                                                       LEV05230
      ITEN = -10                                                        LEV05240
      NISOM = NISO + MEV                                                LEV05250
      IWRITO = IWRITE                                                   LEV05260
      IWRITE = 0                                                        LEV05270
C                                                                       LEV05280
      DO 390 J = 1,NDIS                                                 LEV05290
      IF(MP(J).NE.ITEN) GO TO 390                                       LEV05300
      T0 = VS(J)                                                        LEV05310
C                                                                       LEV05320
C-----------------------------------------------------------------------LEV05330
C                                                                       LEV05340
      IT = MXINIT                                                       LEV05350
      CALL INVERM(ALPHA,BETA,V1,V2,T0,TEMP,T1,EPSM,G,MEV,IT,IWRITE)     LEV05360
C                                                                       LEV05370
C-----------------------------------------------------------------------LEV05380
C                                                                       LEV05390
      IF(TEMP.LE.1.D-10) GO TO 380                                      LEV05400
C     ERROR ESTIMATE WAS NOT SMALL REJECT RELABELLING OF THIS EIGENVALUELEV05410
```

```
      IF(IDIST.EQ.1)  WRITE(11,370) J,T0,TEMP                           LEV05420
  370 FORMAT(/' LAST COMPONENT FOR',I6,'TH EIGENVALUE',E20.12/' IS TOO LLEV05430
     1ARGE = ',E15.6,' SO DO NOT ACCEPT PRTEST RELABELLING'/)           LEV05440
      MP(J) = 0                                                         LEV05450
      IPROJ = IPROJ - 1                                                 LEV05460
      GO TO 390                                                         LEV05470
C     RELABELLING ACCEPTED                                              LEV05480
  380 NISOM = NISOM + 1                                                 LEV05490
      G(NISOM) = BETAM*TEMP                                             LEV05500
  390 CONTINUE                                                          LEV05510
      IWRITE = IWRITO                                                   LEV05520
C                                                                       LEV05530
      IF(IPROJ.EQ.0) GO TO 430                                          LEV05540
      WRITE(6,400) IPROJ                                                LEV05550
  400 FORMAT(/I6,' T-EIGENVALUES WERE RECLASSIFIED AS GOOD.'/           LEV05560
     1' THESE ARE IDENTIFIED IN FILE 3 BY A T-MULTIPLICITY OF -10'/' USELEV05570
     2R SHOULD INSPECT EACH TO MAKE SURE NEIGHBORS HAVE CONVERGED'/)    LEV05580
C                                                                       LEV05590
      IF(IDIST.EQ.1)  WRITE(11,410) IPROJ                               LEV05600
  410 FORMAT(/I6,' T-EIGENVALUES WERE RELABELLED AS GOOD'/              LEV05610
     1' BELOW IS CORRECTED T-MULTIPLICITY PATTERN'/)                    LEV05620
C                                                                       LEV05630
      WRITE(6,420) NDIS, (MP(I), I=1,NDIS)                              LEV05640
      IF(IDIST.EQ.1)  WRITE(11,420) NDIS, (MP(I), I=1,NDIS)             LEV05650
  420 FORMAT(/I6,' = NDIS, T-MULTIPLICITIES (0 MEANS  SPURIOUS)'/       LEV05660
     1 6X, ' (-10) MEANS SPURIOUS T-EIGENVALUE RELABELLED AS GOOD'/(20I4LEV05670
     1))                                                                LEV05680
C                                                                       LEV05690
C     RECALCULATE MINGAPS FOR DISTINCT T(1,MEV) EIGENVALUES.            LEV05700
  430 NM1 = NDIS - 1                                                    LEV05710
      G(NDIS) = VS(NM1)-VS(NDIS)                                        LEV05720
      G(1) = VS(2)-VS(1)                                                LEV05730
C                                                                       LEV05740
      DO 440 J = 2,NM1                                                  LEV05750
      T0 = VS(J)-VS(J-1)                                                LEV05760
      T1 = VS(J+1)-VS(J)                                                LEV05770
      G(J) = T1                                                         LEV05780
      IF (T0.LT.T1) G(J) = -T0                                          LEV05790
  440 CONTINUE                                                          LEV05800
      IF(IPROJ.EQ.0) GO TO 470                                          LEV05810
C     WRITE TO FILE 4 ERROR ESTIMATES FOR THOSE T-EIGENVALUES RELABELLEDLEV05820
      NGOOD = 0                                                         LEV05830
      DO 450 J = 1,NDIS                                                 LEV05840
      IF(MP(J).EQ.0) GO TO 450                                          LEV05850
      NGOOD = NGOOD + 1                                                 LEV05860
      IF(MP(J).NE.ITEN) GO TO 450                                       LEV05870
      T0 = VS(J)                                                        LEV05880
      NISO = NISO + 1                                                   LEV05890
      NISOM = MEV + NISO                                                LEV05900
      WRITE(4,460) NGOOD,T0,G(NISOM),G(J)                               LEV05910
  450 CONTINUE                                                          LEV05920
  460 FORMAT(I10,E25.16,2E14.3)                                         LEV05930
C                                                                       LEV05940
  470 CONTINUE                                                          LEV05950
C                                                                       LEV05960
```

```
C     WRITE THE GOOD T-EIGENVALUES TO FILE 3.  FIRST TRANSFER THEM      LEV05970
C     TO V2 AND THEIR T-MULTIPLICITIES TO THE CORRESPONDING POSITIONS   LEV05980
C     IN MP AND COMPUTE THE A-MINGAPS, THE MINIMAL GAPS BETWEEN THE     LEV05990
C     GOOD T-EIGENVALUES.  THESE GAPS WILL BE PUT IN THE ARRAY G.       LEV06000
C     SINCE G CURRENTLY CONTAINS THE MINIMAL GAPS BETWEEN THE DISTINCT  LEV06010
C     EIGENVALUES OF THE T-MATRIX, THESE GAPS WILL FIRST BE             LEV06020
C     TRANSFERRED TO V1.  NOTE THAT V1<0 MEANS THAT THAT MINIMAL GAP    LEV06030
C     IN THE T-MATRIX IS DUE TO A SPURIOUS T-EIGENVALUE.                LEV06040
C     ALL THIS INFORMATION IS PRINTED TO FILE 3                         LEV06050
C                                                                       LEV06060
  480 CONTINUE                                                          LEV06070
C                                                                       LEV06080
      NG = 0                                                            LEV06090
      DO 490 I = 1,NDIS                                                 LEV06100
      IF (MP(I).EQ.0) GO TO 490                                         LEV06110
      NG = NG+1                                                         LEV06120
      MP(NG) = MP(I)                                                    LEV06130
      V2(NG) = VS(I)                                                    LEV06140
      TEMP = G(I)                                                       LEV06150
      TEMP = DABS(TEMP)                                                 LEV06160
      J = I+1                                                           LEV06170
      IF (G(I).LT.ZERO) J = I-1                                         LEV06180
      IF (MP(J).EQ.0) TEMP = -TEMP                                      LEV06190
      V1(NG) = TEMP                                                     LEV06200
  490 CONTINUE                                                          LEV06210
C                                                                       LEV06220
      WRITE(6,500)MEV                                                   LEV06230
  500 FORMAT(//' T-EIGENVALUE CALCULATION AT MEV = ',I6,'    IS COMPLETELEV06240
     1')                                                                LEV06250
C                                                                       LEV06260
C     NG = NUMBER OF COMPUTED DISTINCT GOOD T-EIGENVALUES.  NEXT        LEV06270
C     GENERATE GAPS BETWEEN GOOD T-EIGENVALUES (AMINGAPS) AND PUT THEM  LEV06280
C     IN G.  G(J) < 0 MEANS THE AMINGAP IS DUE TO THE LEFT-HAND GAP.    LEV06290
C                                                                       LEV06300
      NGM1 = NG - 1                                                     LEV06310
      G(NG) = V2(NGM1)-V2(NG)                                           LEV06320
      G(1) = V2(2)-V2(1)                                                LEV06330
C                                                                       LEV06340
      DO 510 J = 2,NGM1                                                 LEV06350
      T0 = V2(J)-V2(J-1)                                                LEV06360
      T1 = V2(J+1)-V2(J)                                                LEV06370
      G(J) = T1                                                         LEV06380
      IF (T0.LT.T1) G(J) = -T0                                          LEV06390
  510 CONTINUE                                                          LEV06400
C                                                                       LEV06410
C     WRITE GOOD T-EIGENVALUES OUT TO FILE 3.                           LEV06420
C                                                                       LEV06430
      WRITE(3,520)NG,NDIS,MEV,N,SVSEED,MATNO,MULTOL,IB,BTOL             LEV06440
  520 FORMAT(4I6,I12,I8,' = NG,NDIS,MEV,N,SVEED,MATNO'/                 LEV06450
     1 E20.12,I6,E13.4,' = MUTOL,INDEX MINIMAL BETA,BTOL'/              LEV06460
     1' EV NO',1X,'TMULT',10X,'GOOD EIGENVALUE',7X,'TMINGAP',7X,'AMINGAPLEV06470
     1')                                                                LEV06480
C                                                                       LEV06490
      WRITE(3,530)(I,MP(I),V2(I),V1(I),G(I), I=1,NG)                    LEV06500
  530 FORMAT(2I6,E25.16,2E14.3)                                         LEV06510
```

```
C                                                                       LEV06520
C     IF CONVERGENCE FLAG ICONV.NE.1 AND NUMBER OF T-MATRICES           LEV06530
C     CONSIDERED TO DATE IS LESS THAN NUMBER ALLOWED, INCREMENT MEV.    LEV06540
C     AND LOOP BACK TO 210 TO REPEAT COMPUTATIONS.  RESTORE BETA(MEV+1).LEV06550
C                                                                       LEV06560
      BETA(MP1) = BETAM                                                 LEV06570
C                                                                       LEV06580
      IF (MMB.LT.NMEVS.AND.ICONV.NE.1) GO TO 180                        LEV06590
C                                                                       LEV06600
C     END OF LOOP ON DIFFERENT SIZE T-MATRICES ALLOWED.                 LEV06610
C                                                                       LEV06620
  540 CONTINUE                                                          LEV06630
C                                                                       LEV06640
      IF(ISTOP.EQ.0)  WRITE(6,550)                                      LEV06650
  550 FORMAT(/' T-MATRICES (ALPHA AND BETA) ARE NOW AVAILABLE, TERMINATELEV06660
     1')                                                                LEV06670
      IF (IHIS.EQ.1.AND.KMAX.NE.MOLD) WRITE(1,560)                      LEV06680
  560 FORMAT(/' ABOVE ARE THE FOLLOWING VECTORS '/                      LEV06690
     1 '  ALPHA(I), I = 1,KMAX'/                                        LEV06700
     2 '  BETA(I), I = 1,KMAX+1'/                                       LEV06710
     3 ' FINAL TWO LANCZOS VECTORS OF ORDER N FOR I = KMAX,KMAX+1'/     LEV06720
     4 ' ALL VECTORS IN THIS FILE HAVE HEX FORMAT 4Z20 '/               LEV06730
     5 ' ----- END OF FILE 1 NEW ALPHA, BETA HISTORY---------------'///)LEV06740
C                                                                       LEV06750
      IF (ISTOP.EQ.0) GO TO 640                                         LEV06760
C                                                                       LEV06770
      WRITE(3,570)                                                      LEV06780
  570 FORMAT(/' ABOVE ARE COMPUTED GOOD T-EIGENVALUES'/                 LEV06790
     1 ' NG = NUMBER OF GOOD T-EIGENVALUES COMPUTED'/                   LEV06800
     2 ' NDIS = NUMBER OF COMPUTED DISTINCT EIGENVALUES OF T(1,MEV)'/   LEV06810
     3 ' N = ORDER OF A,  MATNO = MATRIX IDENT'/                        LEV06820
     4 ' MULTOL = T-MULTIPLICITY TOLERANCE FOR T-EIGENVALUES IN BISEC'/ LEV06830
     4 ' TMULT IS THE T-MULTIPLICITY OF GOOD T-EIGENVALUE'/             LEV06840
     5 ' TMULT = -1 MEANS SPURIOUS T-EIGENVALUE TOO CLOSE'/             LEV06850
     6 ' DO NOT COMPUTE ERROR ESTIMATES FOR SUCH EIGENVALUES'/          LEV06860
     7 ' AMINGAP = MINIMAL GAP BETWEEN THE COMPUTED A-EIGENVALUES'/     LEV06870
     8 ' AMINGAP .LT. 0. MEANS MINIMAL GAP IS DUE TO LEFT-HAND GAP'/    LEV06880
     9 ' TMINGAP= MINIMAL GAP W.R.T.  DISTINCT EIGENVALUES IN T(1,MEV)'/LEV06890
     1 ' TMINGAP .LT. 0. MEANS MINGAP IS DUE TO SPURIOUS T-EIGENVALUE'/ LEV06900
     2 ' ----- END OF FILE 3 GOODEIGENVALUES----------------------'///)LEV06910
C                                                                       LEV06920
      IF (IDIST.EQ.1) WRITE(11,580)                                     LEV06930
  580 FORMAT(/' ABOVE ARE THE DISTINCT EIGENVALUES OF T(1,MEV).'/       LEV06940
     2 ' THE FORMAT IS      T-MULTIPLICITY    T-EIGENVALUE   TMINGAP'/  LEV06950
     3 '         THIS FORMAT IS REPEATED TWICE ON EACH LINE.'/          LEV06960
     4 ' T-MULTIPLICITY = -1 MEANS THAT THE SUBROUTINE ISOEV HAS TAGGED'LEV06970
     5/'   THIS SIMPLE T-EIGENVALUE AS HAVING A VERY CLOSE SPURIOUS'/   LEV06980
     6 '   T-EIGENVALUE SO THAT NO ERROR ESTIMATE WILL BE COMPUTED'/    LEV06990
     7 '   FOR THAT EIGENVALUE IN SUBROUTINE INVERR.'/                  LEV07000
     8 ' TMINGAP .LT. 0, TMINGAP IS DUE TO LEFT GAP .GT. 0, RIGHT GAP.'/LEV07010
     9 ' EACH OF THE DISTINCT T-EIGENVALUE TABLES IS FOLLOWED'/         LEV07020
     9 ' BY THE T-MULTIPLICITY PATTERN.'/                               LEV07030
     1 ' NDIS = NUMBER OF COMPUTED DISTINCT EIGENVALUES OF T(1,MEV).'/  LEV07040
     2 ' NG = NUMBER OF GOOD T-EIGENVALUES. '/                          LEV07050
     3 ' NISO = NUMBER OF ISOLATED GOOD T-EIGENVALUES. '/               LEV07060
```

```
     4 ' NISO ALSO IS THE COUNT OF +1 ENTRIES IN T-MULTIPLICITY PATTERN.LEV07070
     5 '/' ----- END OF FILE 11 DISTINCT T-EIGENVALUES--------------'///LEV07080
     6 )                                                                LEV07090
C                                                                       LEV07100
      IF(NISO.NE.0)  WRITE(4,590)                                       LEV07110
  590 FORMAT(/' ABOVE ARE THE ERROR ESTIMATES OBTAINED FOR THE ISOLATED LEV07120
     1GOOD T-EIGENVALUES'/                                              LEV07130
     1' OBTAINED VIA INVERSE ITERATION IN THE SUBROUTINE INVERR.'/      LEV07140
     1' ALL OTHER GOOD T-EIGENVALUES HAVE CONVERGED.'/                  LEV07150
     2' ERROR ESTIMATE = BETAM*ABS(UM)'/                                LEV07160
     2' WHERE BETAM = BETA(MEV+1) AND UM = U(MEV).'/                    LEV07170
     3' U = UNIT EIGENVECTOR OF T WHERE T*U = EV*U AND EV = ISOLATED GOOLEV07180
     3D T-EIGENVALUE.'/                                                 LEV07190
     4' TMINGAP = GAP TO NEAREST DISTINCT EIGENVALUE OF T(1,MEV).'/     LEV07200
     5' TMINGAP .LT. 0. MEANS MINGAP IS DUE TO A LEFT NEIGHBOR.'/       LEV07210
     6' ERROR ESTIMATE L.T. 0 MEANS INVERSE ITERATION DID NOT CONVERGE'/LEV07220
     7' ------ END OF FILE 4 ERRINV -------------------------------'//) LEV07230
      GO TO 640                                                         LEV07240
C                                                                       LEV07250
  600 CONTINUE                                                          LEV07260
C                                                                       LEV07270
      IBB = IABS(IBMEV)                                                 LEV07280
      IF (IBMEV.LT.0) WRITE(6,610) MEV,IBB,BETA(IBB)                    LEV07290
  610 FORMAT(/' PROGRAM TERMINATES BECAUSE MEV REQUESTED = ',I6,' IS .GTLEV07300
     1',I6/' AT WHICH AN ABNORMALLY SMALL BETA = ' , E13.4,' OCCURRED'/)LEV07310
      GO TO 640                                                         LEV07320
C                                                                       LEV07330
  620 IF (NDIS.EQ.0.AND.ISTOP.GT.0) WRITE(6,630)                        LEV07340
  630 FORMAT(/' INTERVALS SPECIFIED FOR BISECT DID NOT CONTAIN ANY T-EIGLEV07350
     1ENVALUES'/' PROGRAM TERMINATES')                                  LEV07360
C                                                                       LEV07370
  640 CONTINUE                                                          LEV07380
C                                                                       LEV07390
      STOP                                                              LEV07400
C-----END OF MAIN PROGRAM FOR LANCZOS EIGENVALUE COMPUTATIONS-----------LEV07410
      END                                                               LEV07420
```

2.4 LEVEC: Main Program, Eigenvector Computations

```
C-----LEVEC (EIGENVECTORS OF REAL SYMMETRIC MATRICES)------------------LEV00010
C  Authors:  Jane Cullum and Ralph A. Willoughby (Deceased)             LEV00020
C            Los Alamos National Laboratory                             LEV00030
C            Los Alamos, New Mexico 87544                               LEV00040
C                                                                       LEV00050
C            E-mail:  cullumj@lanl.gov                                  LEV00060
C                                                                       LEV00070
C  These codes are copyrighted by the authors.  These codes             LEV00080
C  and modifications of them or portions of them are NOT to be          LEV00090
C  incorporated into any commercial codes or used for any other         LEV00100
C  commercial purposes such as consulting for other companies,          LEV00110
C  without legal agreements with the authors of these Codes.            LEV00120
C  If these Codes or portions of them are used in other scientific or   LEV00130
C  engineering research works the names of the authors of these codes   LEV00140
C  and appropriate references to their written work are to be           LEV00150
C  incorporated in the derivative works.                                LEV00160
C                                                                       LEV00170
C  This header is not to be removed from these codes.                   LEV00180
C                                                                       LEV00190
C         REFERENCE: Cullum and Willoughby, Chapters 1,2,3,4            LEV00191
C         Lanczos Algorithms for Large Symmetric Eigenvalue ComputationsLEV00192
C         VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in   LEV00193
C         Applied Mathematics, 2002. SIAM Publications,                 LEV00194
C         Philadelphia, PA. USA                                         LEV00195
C                                                                       LEV00196
C                                                                       LEV00197
C                                                                       LEV00200
C      CONTAINS MAIN PROGRAM FOR COMPUTING AN EIGENVECTOR CORRESPONDING  LEV00210
C      TO EACH OF A SET OF EIGENVALUES WHICH HAVE BEEN COMPUTED         LEV00220
C      ACCURATELY BY THE CORRESPONDING LANCZOS EIGENVALUE PROGRAM       LEV00230
C      (LEVAL) FOR REAL SYMMETRIC MATRICES.  THIS PROGRAM COULD BE      LEV00240
C      MODIFIED TO COMPUTE ADDITIONAL EIGENVECTORS FOR ANY EIGENVALUE   LEV00250
C      WHICH IS A MULTIPLE EIGENVALUE OF THE GIVEN A-MATRIX.  THE       LEV00260
C      AMOUNT OF ADDITIONAL COMPUTATION REQUIRED WOULD DEPEND UPON      LEV00270
C      THE GIVEN A-MATRIX AND UPON WHAT PART OF THE SPECTRUM OF         LEV00280
C      A IS INVOLVED.                                                   LEV00290
C                                                                       LEV00300
C      THE LANCZOS EIGENVECTOR COMPUTATIONS ASSUME THAT EACH            LEV00310
C      EIGENVALUE THAT IS BEING CONSIDERED HAS CONVERGED AS AN          LEV00320
C      EIGENVALUE OF THE LANCZOS TRIDIAGONAL MATRICES.                  LEV00330
C                                                                       LEV00340
C      PFORT VERIFIER IDENTIFIED THE FOLLOWING NONPORTABLE              LEV00350
C      CONSTRUCTIONS                                                    LEV00360
C                                                                       LEV00370
C      1.  DATA/MACHEP/ STATEMENT                                       LEV00380
C      2.  ALL READ(5,*) STATEMENTS (FREE FORMAT)                       LEV00390
C      3.  FORMAT(20A4) USED WITH THE EXPLANATORY HEADER, EXPLAN        LEV00400
C      4.  HEXADECIMAL FORMAT (4Z20) USED IN ALPHA/BETA FILES 1 AND 2.  LEV00410
C                                                                       LEV00420
C      IMPORTANT NOTE:  THIS PROGRAM ALLOWS ENLARGEMENT OF THE ALPHA,   LEV00430
C      BETA ARRAYS.  IN PARTICULAR, IF ANY ONE OF THE EIGENVALUES       LEV00440
C      SUPPLIED IS T-SIMPLE AND NOT CLOSE TO A SPURIOUS EIGENVALUE,     LEV00450
```

```
C     THE PROGRAM REQUIRES THAT KMAX BE AT LEAST 11*MEV/8 + 12.  IF
C     KMAX IS NOT THIS LARGE, THEN THE PROGRAM RESETS KMAX TO THIS
C     SIZE 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)
      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
C-----------------------------------------------------------------------
      EXTERNAL CMATV
      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:   >= 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, ERRDGP, 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
C                  VECTORS REQUESTED.
C     ERCONT = 0:  MEANS FOR ANY GIVEN EIGENVALUE, A RITZ VECTOR
C                  WILL NOT BE COMPUTED FOR THAT EIGENVALUE UNLESS
C                  A T-EIGENVECTOR HAS BEEN IDENTIFIED WITH A LAST
C                  COMPONENT WHICH SATISFIES THE SPECIFIED
C                  CONVERGENCE CRITERION.
C     ERCONT = 1:  MEANS FOR ANY GIVEN EIGENVALUE, A RITZ VECTOR
C                  WILL BE COMPUTED.  IF A T-EIGENVECTOR CANNOT
```

```
C                BE IDENTIFIED WHICH SATISFIES THE LAST
C                COMPONENT CRITERION, THEN THE PROGRAM WILL
C                USE THE T-VECTOR THAT CAME CLOSEST TO
C                SATISFYING THE CRITERION.
C     IWRITE = 1:  EXTENDED OUTPUT OF INTERMEDIATE COMPUTATIONS
C                IS WRITTEN TO FILE 6
C     IREAD = 0:   ALPHA/BETA FILE IS REGENERATED.
C     IREAD = 1:   ALPHA/BETA FILE USED IN EIGENVALUE COMPUTATIONS
C                IS READ IN AND EXTENDED IF NECESSARY.  IN BOTH
C                CASES IREAD = 0 OR 1, THE LANCZOS VECTORS ARE
C                ALWAYS REGENERATED FOR THE RITZ VECTOR
C                COMPUTATIONS
C
      READ(5,20) EXPLAN
      READ(5,*) MBOUND,NTVCON,SVTVEC,IREAD
C
      READ(5,20) EXPLAN
      READ(5,*) TVSTOP,LVCONT,ERCONT,IWRITE
      IF (TVSTOP.EQ.1) SVTVEC = 1
C
C     READ IN SEED (RHSEED) FOR GENERATING RANDOM STARTING VECTOR
C     FOR INVERSE ITERATION ON THE T-MATRICES.
      READ(5,20) EXPLAN
      READ(5,*) RHSEED
C
C     READ IN MATNO = MATRIX/RUN IDENTIFICATION NUMBER AND
C     N = ORDER OF A-MATRIX
      READ(5,20) EXPLAN
      READ(5,*) MATNO,N
C
C-----------------------------------------------------------------------
C     INITIALIZE THE ARRAYS FOR THE USER-SPECIFIED MATRIX
C     AND PASS THE STORAGE LOCATIONS OF THESE ARRAYS TO THE
C     MATRIX-VECTOR MULTIPLY SUBROUTINE CMATV.
C
      CALL USPEC(N,MATNO)
C
C-----------------------------------------------------------------------
C     MASK UNDERFLOW AND OVERFLOW
      CALL MASK
C
C-----------------------------------------------------------------------
C     WRITE RUN PARAMETERS OUT TO FILE 6
C
      WRITE(6,30) MATNO,N
   30 FORMAT(/' MATRIX IDENTIFICATION NO. = ',I10,' ORDER OF A = ',I5)
C
      WRITE(6,40) MBOUND,NTVCON,SVTVEC,IREAD
   40 FORMAT(/3X,'MBOUND',3X,'NTVCON',3X,'SVTVEC',3X,'IREAD'/3I9,I8)
C
      WRITE(6,50) TVSTOP,LVCONT,ERCONT,IWRITE
   50 FORMAT(/3X,'TVSTOP',3X,'LVCONT',3X,'ERCONT',3X,'IWRITE'/4I9)
C
      WRITE(6,60) MDIMTV,MDIMRV,MBETA
   60 FORMAT(/3X,'MDIMTV',3X,'MDIMRV',3X,'MBETA'/2I9,I8)
```

```
C
      WRITE(6,70) RELTOL,RHSEED
   70 FORMAT(/7X,'RELTOL',3X,'RHSEED'/E13.4,I9)
C
C
C     FROM FILE 3 READ IN THE NUMBER OF EIGENVALUES (NGOOD) FOR WHICH
C     EIGENVECTORS ARE REQUESTED, THE ORDER (MEV) OF THE LANCZOS
C     TRIDIAGONAL MATRIX USED IN COMPUTING THESE EIGENVALUES, THE
C     ORDER (NOLD) OF THE USER-SPECIFIED MATRIX USED IN THE EIGENVALUE
C     COMPUTATIONS, THE SEED (SVSEED) USED FOR GENERATING THE STARTING
C     VECTOR THAT WAS USED IN THOSE LANCZOS EIGENVALUE COMPUTATIONS,
C     AND THE MATRIX/RUN IDENTIFICATION NUMBER (MATOLD) USED IN THOSE
C     COMPUTATIONS.  ALSO READ IN THE NUMBER (NDIS) OF DISTINCT
C     EIGENVALUES OF T(1,MEV) THAT WERE COMPUTED BUT THIS VALUE IS
C     NOT USED IN THE EIGENVECTOR COMPUTATIONS.
C
      READ(3,80) NGOOD,NDIS,MEV,NOLD,SVSEED,MATOLD
   80 FORMAT(4I6,I12,I8)
C
C     READ IN THE T-MULTIPLICITY TOLERANCE USED IN THE BISEC SUBROUTINE
C     DURING THE COMPUTATION OF THE GIVEN EIGENVALUES.
C     ALSO READ IN THE FLAG IB.  IF IB < 0, THEN SOME BETA(I) IN THE
C     T-MATRIX FILE PROVIDED ON FILE 2 FAILED THE ORTHOGONALITY
C     TEST IN THE TNORM SUBROUTINE.  USER SHOULD NOTE THAT THIS VECTOR
C     PROGRAM PROCEEDS INDEPENDENTLY OF THE SIZE OF THE BETA USED.
C
      READ(3,90) MULTOL,IB,BTOL
   90 FORMAT(E20.12,I6,E13.4)
C
      TEMP = DFLOAT(MEV+1000)
      TTOL = MULTOL/TEMP
      WRITE(6,100) MULTOL,TTOL
  100 FORMAT(/' T-MULTIPLICITY TOLERANCE USED IN THE EIGENVALUE COMPUTAT
     1IONS WAS',E13.4/' SCALED MACHINE EPSILON IS',E13.4)
C
C     CONTINUE WRITE TO FILE 6 OF THE PARAMETERS FOR THIS RUN
C
      WRITE(6,110)NGOOD,NDIS,MEV,NOLD,MATOLD,SVSEED,MULTOL,IB,BTOL
  110 FORMAT(/' EIGENVALUES SUPPLIED ARE READ IN FROM FILE 3'/' FILE 3
     1HEADER IS'/4X,'NG',2X,'NDIS',3X,'MEV',2X,'NOLD',2X,'MATOLD',4X,
     1'SVSEED',6X,'MULTOL',6X,'IB',9X,'BTOL'/4I6,I8,I10,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 1310
C
C     CHECK THAT THE ORDER N AND THE MATRIX IDENTIFICATION NUMBER
C     MATNO SPECIFIED BY THE USER AGREE WITH THOSE READ IN FROM
C     FILE 3.
  120 ITEMP = (NOLD-N)**2+(MATOLD-MATNO)**2
      IF (ITEMP.NE.0) GO TO 1330
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'  AMINGAP '/(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 1550
C
  220 CONTINUE
C
      WRITE(6,230) (J,GOODEV(J),ERR(J), J=1,NGOOD)
  230 FORMAT(' ERROR ESTMATES ='/4X,' J',5X,'EIGENVALUE',10X,' ESTIMATE
     1'/(I6,E20.12,E14.3))
C
      IF(IREAD.EQ.0)  GO TO 330
C
C     READ IN THE SIZE OF THE T-MATRIX PROVIDED ON FILE 2.  READ IN
C     THE ORDER OF THE USER-SPECIFIED MATRIX , THE SEED FOR THE
C     RANDOM NUMBER GENERATOR, AND THE MATRIX/TEST IDENTIFICATION
C     NUMBER THAT WERE USED IN THE LANCZOS EIGENVALUE COMPUTATIONS.
C     THESE ARE USED IN A CONSISTENCY CHECK
```

```
C     IF FLAG IREAD = 0 REGENERATE ALPHA, BETA
C
      READ(2,240) KMAX,NOLD,SVSOLD,MATOLD
  240 FORMAT(2I6,I12,I8)
C
      WRITE(6,250) KMAX,NOLD,SVSOLD,MATOLD
  250 FORMAT(/' READ IN THE T-MATRICES STORED ON FILE 2'/' FILE 2 HEADER
     1 IS'/2X,'KMAX',2X,'NOLD',6X,'SVSOLD',2X,'MATOLD'/2I6,I12,I8/)
C
C     CHECK THAT THE ORDER, THE MATRIX/TEST IDENTIFICATION NUMBER
C     AND THE SEED FOR THE RANDOM NUMBER GENERATOR USED IN THE
C     LANCZOS COMPUTATIONS THAT GENERATED THE ALPHA,BETA FILE
C     BEING USED AGREE WITH WHAT THE USER HAS SPECIFIED.
      IF (NOLD.NE.N.OR.MATOLD.NE.MATNO.OR.SVSOLD.NE.SVSEED) GO TO 1350
C
      KMAX1 = KMAX + 1
C
C     READ IN THE T-MATRICES FROM FILE 2.  THESE ARE USED TO GENERATE
C     THE T-EIGENVECTORS THAT WILL BE USED IN THE RITZ VECTOR
C     COMPUTATIONS.  HISTORY MUST BE STORED IN MACHINE FORMAT
C     ((4Z20) FOR IBM/3081)
C
      READ(2,260) (ALPHA(J), J=1,KMAX)
      READ(2,260) (BETA(J), J=1,KMAX1)
  260 FORMAT(4Z20)
C
      READ(2,260) (V1(J), J=1,N)
      READ(2,260) (V2(J), J=1,N)
C
C     KMAX MAY BE ENLARGED IF THE SIZE AT WHICH THE EIGENVALUE
C     COMPUTATIONS WERE PERFORMED IS ESSENTIALLY KMAX AND
C     THERE IS AT LEAST ONE EIGENVALUE THAT IS T-SIMPLE AND
C     T-ISOLATED, IN THE SENSE THAT IF ITS NEAREST NEIGHBOR IS TOO
C     CLOSE THAT NEIGHBOR IS A 'GOOD' T-EIGENVALUE.
      DO 270 J = 1,NGOOD
      IF(MP(J).EQ.1) GO TO 290
  270 CONTINUE
      WRITE(6,280)
  280 FORMAT(/' ALL EIGENVALUES USED ARE T-MULTIPLE OR CLOSE TO SPURIOUS
     1 T-EIGENVALUES'/' SO KMAX IS NOT INCREASED')
      IF(KMAX.LT.MEV) GO TO 1370
      GO TO 310
C
  290 KMAXN= 11*MEV/8 + 12
      IF(MBETA.LE.KMAXN) GO TO 1530
      IF(KMAX.GE.KMAXN )  GO TO 310
      WRITE(6,300) KMAX, KMAXN
  300 FORMAT(' ENLARGE KMAX FROM ',I6,' TO ',I6)
      MOLD1 = KMAX + 1
      KMAX = KMAXN
      GO TO 380
C
  310 WRITE(6,320) KMAX
  320 FORMAT(/' T-MATRICES HAVE BEEN READ IN FROM FILE 2'/' THE LARGEST
     1SIZE T-MATRIX ALLOWED IS',I6/)
```

```
C                                                                       LEV03760
      IF(IREAD.EQ.1) GO TO 400                                          LEV03770
C                                                                       LEV03780
C     REGENERATE THE ALPHA AND BETA                                     LEV03790
C                                                                       LEV03800
  330 MOLD1 = 1                                                         LEV03810
C                                                                       LEV03820
      DO 340 J = 1,NGOOD                                                LEV03830
      IF(MP(J).EQ.1) GO TO 360                                          LEV03840
  340 CONTINUE                                                          LEV03850
      KMAX = MEV + 12                                                   LEV03860
      WRITE(6,350) KMAX                                                 LEV03870
  350 FORMAT(/' ALL EIGENVALUES FOR WHICH EIGENVECTORS ARE TO BE COMPUTELEV03880
     1D ARE EITHER T-MULTIPLE OR CLOSE TO'/' A SPURIOUS T-EIGENVALUE. THLEV03890
     1EREFORE SET KMAX = MEV + 12 = ',I7)                               LEV03900
      GO TO 380                                                         LEV03910
C                                                                       LEV03920
  360 KMAXN = 11*MEV/8 + 12                                             LEV03930
      IF(MBETA.LE.KMAXN) GO TO 1530                                     LEV03940
      WRITE(6,370) KMAXN                                                LEV03950
  370 FORMAT(' SET KMAX EQUAL TO ',I6)                                  LEV03960
      KMAX = KMAXN                                                      LEV03970
C                                                                       LEV03980
  380 WRITE(6,390) MOLD1,KMAX                                           LEV03990
  390 FORMAT(/' LANCZS SUBROUTINE GENERATES ALPHA(J), BETA(J+1), J =',  LEV04000
     1 I6,' TO ', I6/)                                                  LEV04010
C                                                                       LEV04020
C-----------------------------------------------------------------------LEV04030
C                                                                       LEV04040
      IIX = SVSEED                                                      LEV04050
      CALL LANCZS(CMATV,ALPHA,BETA,V1,V2,G,KMAX,MOLD1,N,IIX)            LEV04060
C                                                                       LEV04070
C-----------------------------------------------------------------------LEV04080
C                                                                       LEV04090
  400 CONTINUE                                                          LEV04100
C                                                                       LEV04110
C     THE SUBROUTINE STURMI DETERMINES THE SMALLEST SIZE T-MATRIX FOR   LEV04120
C     WHICH THE EIGENVALUE IN QUESTION IS A T-EIGENVALUE (TO WITHIN A   LEV04130
C     GIVEN TOLERANCE) AND IF POSSIBLE THE SMALLEST SIZE T-MATRIX       LEV04140
C     FOR WHICH IT IS A DOUBLE T-EIGENVALUE (TO WITHIN THE SAME         LEV04150
C     TOLERANCE).  THE SIZE T-MATRIX USED IN THE RITZ VECTOR            LEV04160
C     COMPUTATIONS IS THEN DETERMINED BY LOOPING ON SIZE OF THE         LEV04170
C     T-EIGENVECTORS STARTING WITH A T-SIZE DETERMINED FROM THE         LEV04180
C     OUTPUT FROM STURMI.                                               LEV04190
C                                                                       LEV04200
C                                                                       LEV04210
      STUTOL = SCALE0*MULTOL                                            LEV04220
      IF(IWRITE.EQ.1) WRITE(6,410)                                      LEV04230
  410 FORMAT(' FROM STURMI')                                            LEV04240
      DO 450 J = 1,NGOOD                                                LEV04250
      EVAL = GOODEV(J)                                                  LEV04260
C     COMPUTE THE TOLERANCES USED BY STURMI TO DETERMINE AN INTERVAL    LEV04270
C     CONTAINING THE EIGENVALUE EVAL.                                   LEV04280
      TEMP = DABS(EVAL)*RELTOL                                          LEV04290
      TOLN = DMAX1(TEMP,STUTOL)                                         LEV04300
```

```
C                                                                       LEV04310
C-----------------------------------------------------------------------LEV04320
C                                                                       LEV04330
      CALL STURMI(ALPHA,BETA,EVAL,TOLN,EPSM,KMAX,MK1,MK2,IC,IWRITE)     LEV04340
C                                                                       LEV04350
C-----------------------------------------------------------------------LEV04360
C                                                                       LEV04370
C     STORE THE COMPUTED ORDERS OF T-MATRICES FOR LATER PRINTOUT        LEV04380
      M1(J) = MK1                                                       LEV04390
      M2(J) = MK2                                                       LEV04400
      ML(J) = (MK1 + 3*MK2)/4                                           LEV04410
      IF(MK2.EQ.KMAX)  ML(J) = KMAX                                     LEV04420
C                                                                       LEV04430
      IF(IC.GT.0) GO TO 430                                             LEV04440
C     IC = 0 MEANS THERE WAS NO EIGENVALUE IN THE DESIGNATED INTERVAL   LEV04450
C     BY T-SIZE KMAX.  THIS MEANS THAT THE EIGENVALUE PROVIDED HAS      LEV04460
C     NOT YET CONVERGED SO ITS EIGENVECTOR WILL NOT BE COMPUTED.        LEV04470
      WRITE(6,420) J,GOODEV(J),MK1,MK2                                  LEV04480
  420 FORMAT(I6,'TH EIGENVALUE',E20.12,' HAS NOT CONVERGED '/           LEV04490
     1' SO DO NOT COMPUTE ANY T-EIGENVECTOR OR RITZ VECTOR FOR IT'      LEV04500
     1/' MK1 AND MK2 FOR THIS EIGENVALUE WERE',2I6)                     LEV04510
      MP(J) = MPMIN                                                     LEV04520
      MA(J) = -2*KMAX                                                   LEV04530
      GO TO 450                                                         LEV04540
C     COMPUTE AN APPROPRIATE SIZE T-MATRIX FOR THE GIVEN EIGENVALUE.    LEV04550
  430 IF(M2(J).EQ.KMAX) GO TO 440                                       LEV04560
C     M1 AND M2 WERE BOTH DETERMINED                                    LEV04570
      MA(J) = (3*M1(J) + M2(J))/4  + 1                                  LEV04580
      GO TO 450                                                         LEV04590
C     M2 NOT DETERMINED                                                 LEV04600
  440 MA(J) = (5*M1(J))/4  + 1                                          LEV04610
C                                                                       LEV04620
  450 CONTINUE                                                          LEV04630
C                                                                       LEV04640
      IF (IWRITE.EQ.1) WRITE(6,460) (MA(JJ), JJ=1,NGOOD)                LEV04650
  460 FORMAT(/' 1ST GUESS AT APPROPRIATE SIZE T-MATRICES'/              LEV04660
     1 ' ACTUAL VALUES WILL PROBABLY BE 1/4 AGAIN AS MUCH'/(13I6))      LEV04670
C                                                                       LEV04680
C     PRINT OUT TO FILE 10 1ST GUESSES AT SIZES OF THE T-MATRICES TO    LEV04690
C     BE USED IN THE EIGENVECTOR COMPUTATIONS.                          LEV04700
C     PROGRAM LOOPS ON T-SIZE TO DETERMINE APPROPRIATE SIZE T-MATRIX.   LEV04710
      WRITE(10,470) N,KMAX                                              LEV04720
  470 FORMAT(2I8,' = ORDER OF USER MATRIX AND MAX ORDER OF T(1,MEV)')   LEV04730
C                                                                       LEV04740
      WRITE(10,480)                                                     LEV04750
  480 FORMAT(/' 1ST GUESS AT APPROPRIATE SIZE T-MATRICES'/              LEV04760
     1 ' ACTUAL VALUES WILL PROBABLY BE 1/4 AGAIN AS MUCH'/)            LEV04770
C                                                                       LEV04780
      WRITE(10,490)                                                     LEV04790
  490 FORMAT(4X,'J',4X,'A-EIGENVALUE',4X,'M1(J)',1X,'M2(J)',1X,'MA(J)') LEV04800
C                                                                       LEV04810
      WRITE(10,500) (J,GOODEV(J),M1(J),M2(J), MA(J), J=1,NGOOD)         LEV04820
  500 FORMAT(I5,E19.12,3I6)                                             LEV04830
C                                                                       LEV04840
      IF(MBOUND.EQ.1) WRITE(10,510)                                     LEV04850
```

```
  510 FORMAT(/'  EV = GOODEV(J) IS A GOOD EIGENVALUE OF T(1,MEV)'/
     1' M1 = SMALLEST VALUE OF M SUCH THAT T(1,M) HAS AT LEAST'/
     1 '     ONE EIGENVALUE IN THE INTERVAL (EV-TOLN,EV+TOLN)'/
     1 '     TOLN(J) = DMAX1(GOODEV(J)*RELTOL, SCALE0*MULTOL)'/
     1 ' M2 = SMALLEST M (IF ANY) SUCH THAT IN THE ABOVE INTERVAL'/
     1 '      T(1,M) HAS AT LEAST TWO EIGENVALUES '/
     1 ' IABS(MA(J)) = APPROPRIATE SIZE T-MATRIX FOR GOODEV(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     TERMINATE AFTER COMPUTING 1ST GUESSES AT SIZES OF THE
C     T-MATRICES REQUIRED FOR THE GIVEN EIGENVALUES?
      IF(MBOUND.EQ.1) GO TO 1390
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 1410
C
C-----------------------------------------------------------------------
C     GENERATE A RANDOM VECTOR TO BE USED REPEATEDLY BY
C     SUBROUTINE INVERM
C
      CALL GENRAN(RHSEED,G,KMAX)
C
C----------------------------------------------------------------------
C
C     LOOP ON GIVEN EIGENVALUES TO COMPUTE THE CORRESPONDING
C     T-EIGENVECTOR.
C
      MTOL = 0
      NTVEC = 0
      ILBIS = 0
      DO 710 J = 1,NGOOD
      ICOUNT = 0
      ERRMIN = 10.D0
      MABEST = MPMIN
      IF(MP(J).EQ.MPMIN) GO TO 710
      TFLAG = 0
      EVAL = GOODEV(J)
      TEMP = DABS(EVAL)*RELTOL
      UB = EVAL + DMAX1(STUTOL,TEMP)
      LB = EVAL - DMAX1(STUTOL,TEMP)
  530 KMAXU = IABS(MA(J))
C
C     SELECT A SUITABLE INCREMENT FOR THE ORDERS OF THE T-MATRICES
C     TO BE CONSIDERED IN DETERMINING APPROPRIATE SIZES FOR THE RITZ
C     VECTOR COMPUTATIONS.
      IF(ICOUNT.GT.0) GO TO 550
C     SELECT IDELTA(J) BASED UPON THE T-MULTIPLICITY OBTAINED
```

```
      IF(M2(J).EQ.KMAX) GO TO 540
C     M2 DETERMINED
      IDELTA(J) = ((3*M1(J) + 5*M2(J))/8  +  1 - IABS(MA(J)))/10  +  1
      GO TO 550
C     M2 NOT DETERMINED
  540 MAMAX = MIN0((11*MEV)/8 + 12, (13*M1(J))/8 + 1)
      IDELTA(J) = (MAMAX - IABS(MA(J)))/10  +  1
  550 ICOUNT = ICOUNT + 1
C
C-----------------------------------------------------------------------
C     TO MIMIMIZE THE EFFECT OF THE ONE-SIDED ACCEPTANCE TEST FOR
C     EIGENVALUES IN THE BISEC SUBROUTINE, RECOMPUTE THE GIVEN
C     EIGENVALUE AT THE SPECIFIED KMAXU
C
      CALL LBISEC(ALPHA,BETA,EPSM,EVAL,EVALN,LB,UB,TTOL,KMAXU,NEVT)
C
C-----------------------------------------------------------------------
C
C     CHECK WHETHER OR NOT GIVEN T-MATRIX HAS AN EIGENVALUE IN THE
C     SPECIFIED INTERVAL AND IF SO WHAT ITS T-MULTIPLICITY IS.
C
      IF(NEVT.EQ.1) GO TO 590
      IF(NEVT.NE.0) GO TO 570
      ILBIS = 1
      WRITE(6,560) EVAL,KMAXU
  560 FORMAT(/' PROBLEM ENCOUNTERED IN RECOMPUTATION OF USER-SUPPLIED EI
     1GENVALUE',E20.12/' THE SIZE T-MATRIX SPECIFIED',I6,' DOES NOT
     1HAVE AN EIGENVALUE IN THE INTERVAL SPECIFIED'/' THEREFORE NO EIGEN
     1VECTOR WILL BE COMPUTED FOR THIS PARTICULAR EIGENVALUE'/)
      GO TO 610
C
  570 IF(NEVT.GT.1)  WRITE(6,580) EVAL,KMAXU
  580 FORMAT(/' PROBLEM ENCOUNTERED IN RECOMPUTATION OF USER-SUPPLIED
     1EIGENVALUE',E20.12/' FOR THE SIZE T-MATRIX SPECIFIED =',I6,' THE
     1GIVEN EIGENVALUE IS T-MULTIPLE IN THE INTERVAL SPECIFIED'/' SOMETH
     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                                                                       LEV05960
C     RECORD THE BEGINNING AND END OF THE T-EIGENVECTOR BEING COMPUTED  LEV05970
      MINT(J) = KINT                                                    LEV05980
      MFIN(J) = MTOL                                                    LEV05990
C                                                                       LEV06000
C-----------------------------------------------------------------------LEV06010
C     SUBROUTINE INVERM DOES INVERSE ITERATION, I.E. SOLVES             LEV06020
C     (T(1,KMAXU) - EVAL)*U = RHS  FOR EACH EIGENVALUE TO OBTAIN THE    LEV06030
C     DESIRED T-EIGENVECTOR.                                            LEV06040
C                                                                       LEV06050
      IF(IWRITE.EQ.1)  WRITE(6,600) J                                   LEV06060
  600 FORMAT(/I6,'TH EIGENVALUE')                                       LEV06070
C                                                                       LEV06080
      CALL INVERM(ALPHA,BETA,V1,TVEC(KINT),EVAL,ERROR,TERROR,EPSM,      LEV06090
     1 G,KMAXU,IT,IWRITE)                                               LEV06100
C                                                                       LEV06110
C-----------------------------------------------------------------------LEV06120
C                                                                       LEV06130
      TERR(J) = TERROR                                                  LEV06140
      TLAST(J) = ERROR                                                  LEV06150
      KMAXU1 = KMAXU + 1                                                LEV06160
      TBETA(J) = BETA(KMAXU1)*ERROR                                     LEV06170
C                                                                       LEV06180
C     AFTER EACH OF THE T-EIGENVECTORS IS COMPUTED, THE                 LEV06190
C     SIZE OF THE ERROR ESTIMATE, ERROR IS CHECKED.                     LEV06200
C     IF THIS ESTIMATE IS NOT AS SMALL AS DESIRED AND                   LEV06210
C     |MA(J)| < ML(J), PROGRAM ATTEMPTS TO INCREASE THE SIZE OF |MA(J)| LEV06220
C     AND REPEAT THE T-EIGENVECTOR COMPUTATIONS.                        LEV06230
C                                                                       LEV06240
      IF(ERROR.LT.ERTOL.OR.TFLAG.EQ.1)  GO TO 700                       LEV06250
C                                                                       LEV06260
      IF(ERROR.GE.ERRMIN) GO TO 610                                     LEV06270
C     LAST COMPONENT IS LESS THAN MINIMAL TO DATE                       LEV06280
      ERRMIN = ERROR                                                    LEV06290
      MABEST = MA(J)                                                    LEV06300
  610 CONTINUE                                                          LEV06310
C                                                                       LEV06320
      IF(MA(J).GT.0)  ITEST = MA(J) + IDELTA(J)                         LEV06330
      IF(MA(J).LT.0)  ITEST = -(IABS(MA(J)) + IDELTA(J))                LEV06340
      IF(IABS(ITEST).LE.ML(J).AND.ICOUNT.LE.10) GO TO 630               LEV06350
C     NEW MA(J) IS GREATER THAN MAXIMUM ALLOWED.                        LEV06360
      IF(ERCONT.EQ.0.OR.MABEST.EQ.MPMIN) GO TO 650                      LEV06370
      TFLAG = 1                                                         LEV06380
      MA(J) = MABEST                                                    LEV06390
      IF(ILBIS.EQ.0)  MTOL = MTOL - KMAXU                               LEV06400
      WRITE(6,620) MA(J)                                                LEV06410
  620 FORMAT(' 10 ORDERS WERE CONSIDERED.  NONE SATISFIED THE ERROR TESTLEV06420
     1'/' THEREFORE USE THE BEST ORDER OBTAINED FOR THE EIGENVECTORS'   LEV06430
     1,I6)                                                              LEV06440
      GO TO 530                                                         LEV06450
C                                                                       LEV06460
  630 MA(J) = ITEST                                                     LEV06470
C                                                                       LEV06480
      MT = IABS(MA(J))                                                  LEV06490
      IF(IWRITE.EQ.1) WRITE(6,640)  MT                                  LEV06500
```

```
  640 FORMAT(/' CHANGE SIZE OF T-MATRIX TO ',I6,' RECOMPUTE T-EIGENVECTOLEV06510
     1R')                                                               LEV06520
C                                                                       LEV06530
      IF(ILBIS.EQ.0)  MTOL = MTOL - KMAXU                               LEV06540
C                                                                       LEV06550
      GO TO 530                                                         LEV06560
C                                                                       LEV06570
C     APPROPRIATE SIZE T-MATRIX WAS NOT OBTAINED                        LEV06580
  650 CONTINUE                                                          LEV06590
      WRITE(10,660) J,EVAL,MP(J)                                        LEV06600
  660 FORMAT(/' ON 10 INCREMENTS NOT ABLE TO IDENTIFY APPROPRIATE SIZE  LEV06610
     1T-MATRIX FOR'/                                                    LEV06620
     1' EIGENVALUE(',I4,') = ',E20.12,' T-MULTIPLICITY =',I4/)          LEV06630
      IF(M2(J).EQ.KMAX) WRITE(10,670)                                   LEV06640
      IF(M2(J).LT.KMAX) WRITE(10,680)                                   LEV06650
  670 FORMAT(/' ORDERS TESTED RANGED FROM 5*M1(J)/4 TO APPROXIMATELY    LEV06660
     1 '/'  MIN(11*MEV/8,13*M1(J)/8)'/)                                 LEV06670
  680 FORMAT(/' ORDERS TESTED RANGED FROM (3*M1(J)+M2(J))/4 TO APPROXIMALEV06680
     1TELY'/'  (3*M1(J) + 5*M2(J))/8.'/)                                LEV06690
      WRITE(10,690)                                                     LEV06700
  690 FORMAT(' ALLOWING LARGER ORDERS FOR THE T-MATRICES MAY RESULT IN  LEV06710
     1 SUCCESS'/' BUT PROBABLY WILL NOT.  PROBLEM IS PROBABLY DUE TO'   LEV06720
     1 /' LACK OF CONVERGENCE OF GIVEN EIGENVALUE, CHECK THE ERROR ESTIMLEV06730
     1ATE'/)                                                            LEV06740
      MP(J) = MPMIN                                                     LEV06750
      IF(ILBIS.EQ.0)  MTOL = MTOL - KMAXU                               LEV06760
      GO TO 710                                                         LEV06770
  700 NTVEC = NTVEC + 1                                                 LEV06780
C                                                                       LEV06790
  710 CONTINUE                                                          LEV06800
      NGOODC = NGOOD                                                    LEV06810
      GO TO 740                                                         LEV06820
C                                                                       LEV06830
C     COME HERE IF THERE IS NOT ENOUGH ROOM FOR ALL OF T-EIGENVECTORS   LEV06840
  720 NGOODC = J-1                                                      LEV06850
      WRITE(6,730)  J, MTOL, MDIMTV                                     LEV06860
  730 FORMAT(/'  NOT ENOUGH ROOM IN TVEC FOR ',I4,'TH T-VECTOR'/'  T-DIMLEV06870
     1ENSION REQUESTED = ',I6,' BUT TVEC HAS DIMENSION = ',I6/)         LEV06880
      IF(NGOODC.EQ.0)  GO TO 1430                                       LEV06890
      MTOL = MTOL-KMAXU                                                 LEV06900
C                                                                       LEV06910
  740 CONTINUE                                                          LEV06920
C                                                                       LEV06930
C     THE LOOP ON T-EIGENVECTOR COMPUTATIONS IS COMPLETE.               LEV06940
C     WRITE OUT THE SIZE T-MATRICES THAT WILL BE USED FOR               LEV06950
C     THE RITZ VECTOR COMPUTATIONS.                                     LEV06960
C                                                                       LEV06970
      WRITE(10,750)                                                     LEV06980
  750 FORMAT(/' SIZES OF T-MATRICES THAT WILL BE USED IN THE RITZ COMPUTLEV06990
     1ATIONS'/5X,'J',16X,'GOODEV(J)',1X,'MA(J)')                        LEV07000
C                                                                       LEV07010
      WRITE(10,760)   (J,GOODEV(J),MA(J), J=1,NGOOD)                    LEV07020
  760 FORMAT(I6,E25.14,I6)                                              LEV07030
      WRITE(10,510)                                                     LEV07040
C                                                                       LEV07050
```

```
      WRITE(6,770) MTOL                                                 LEV07060
  770 FORMAT(/' THE CUMULATIVE LENGTH OF THE T-EIGENVECTORS IS',I18)    LEV07070
C                                                                       LEV07080
      WRITE(6,780) NTVEC,NGOOD                                          LEV07090
  780 FORMAT(/I6,' T-EIGENVECTORS OUT OF',I6,' REQUESTED WERE COMPUTED')LEV07100
C                                                                       LEV07110
C     SAVE THE T-EIGENVECTORS ON FILE 11?                               LEV07120
      IF(TVSTOP.NE.1.AND.SVTVEC.EQ.0) GO TO 840                         LEV07130
C                                                                       LEV07140
      WRITE(11,790) NTVEC,MTOL,MATNO,SVSEED                             LEV07150
  790 FORMAT(I6,3I12,' = NTVEC,MTOL,MATNO,SVSEED')                      LEV07160
C                                                                       LEV07170
      DO 820 J=1,NGOODC                                                 LEV07180
C     IF MP(J) = MPMIN THEN NO SUITABLE T-EIGENVECTOR IS AVAILABLE      LEV07190
C     FOR THAT EIGENVALUE.                                              LEV07200
      IF(MP(J).EQ.MPMIN) WRITE(11,800) J,MA(J),GOODEV(J),MP(J)          LEV07210
  800 FORMAT(2I6,E20.12,I6/' TH EIGVAL,T-SIZE,EVALUE,FLAG,NO EIGVEC')   LEV07220
      IF(MP(J).NE.MPMIN) WRITE(11,810) J,MA(J),GOODEV(J),MP(J)          LEV07230
  810 FORMAT(I6,I6,E20.12,I6/' T-EIGVEC,SIZE T,EVALUE OF A,MP(J)')      LEV07240
      IF(MP(J).EQ.MPMIN) GO TO 820                                      LEV07250
      KI = MINT(J)                                                      LEV07260
      KF = MFIN(J)                                                      LEV07270
C                                                                       LEV07280
      WRITE(11,260) (TVEC(K), K=KI,KF)                                  LEV07290
C                                                                       LEV07300
  820 CONTINUE                                                          LEV07310
C                                                                       LEV07320
      IF(TVSTOP.NE.1) GO TO 840                                         LEV07330
C                                                                       LEV07340
      WRITE(6,830) TVSTOP, NTVEC,NGOOD                                  LEV07350
  830 FORMAT(/' USER SET TVSTOP = ',I1/                                 LEV07360
     1' THEREFORE PROGRAM TERMINATES AFTER T-EIGENVECTOR COMPUTATIONS'/ LEV07370
     1' T-EIGENVECTORS THAT WERE COMPUTED ARE SAVED ON FILE 11'/        LEV07380
     1I8,' T-EIGENVECTORS WERE COMPUTED OUT OF',I7,' REQUESTED'/)       LEV07390
C                                                                       LEV07400
      GO TO 1550                                                        LEV07410
C                                                                       LEV07420
  840 CONTINUE                                                          LEV07430
C     IF NOT ABLE TO COMPUTE ALL THE REQUESTED T-EIGENVECTORS,          LEV07440
C     CONTINUE WITH THE LANCZOS VECTOR COMPUTATIONS ANYWAY?             LEV07450
      IF(NTVEC.NE.NGOOD.AND.LVCONT.EQ.0) GO TO 1450                     LEV07460
C                                                                       LEV07470
C     COMPUTE THE MAXIMUM SIZE OF THE T-MATRIX USED FOR THOSE           LEV07480
C     EIGENVALUES WITH GOOD ERROR ESTIMATES.                            LEV07490
C                                                                       LEV07500
      KMAXU = 0                                                         LEV07510
      DO 850 J = 1,NGOODC                                               LEV07520
      MT = IABS(MA(J))                                                  LEV07530
      IF(MT.LT.KMAXU.OR.MP(J).EQ.MPMIN) GO TO 850                       LEV07540
      KMAXU = MT                                                        LEV07550
  850 CONTINUE                                                          LEV07560
C                                                                       LEV07570
      IF(KMAXU.EQ.0) GO TO 1490                                         LEV07580
C                                                                       LEV07590
      WRITE(6,860) KMAXU                                                LEV07600
```

```
  860 FORMAT(/I6,' = LARGEST SIZE T-MATRIX TO BE USED IN THE RITZ VECTORLEV07610
     1 COMPUTATIONS')                                                   LEV07620
C                                                                       LEV07630
C     COUNT THE NUMBER OF RITZ VECTORS NOT BEING COMPUTED               LEV07640
      MREJEC = 0                                                        LEV07650
      DO 870 J=1,NGOODC                                                 LEV07660
  870 IF(MP(J).EQ.MPMIN)  MREJEC = MREJEC + 1                           LEV07670
      MREJET = MREJEC + (NGOOD-NGOODC)                                  LEV07680
      IF(MREJET.NE.0) WRITE(6,880) MREJET                               LEV07690
  880 FORMAT(/' RITZ VECTORS ARE NOT COMPUTED FOR',I6,' OF THE EIGENVALULEV07700
     1ES'/)                                                             LEV07710
      NACT = NGOODC - MREJEC                                            LEV07720
      WRITE(6,890) NGOOD,NTVEC,NACT                                     LEV07730
  890 FORMAT(/I6,' RITZ VECTORS WERE REQUESTED'/I6,' T-EIGENVECTORS WERELEV07740
     1 COMPUTED'/I6,' RITZ VECTORS WILL BE COMPUTED'/)                  LEV07750
C     CHECK IF THERE ARE ANY RITZ VECTORS TO COMPUTE                    LEV07760
      IF(MREJEC.EQ.NGOODC) GO TO 1470                                   LEV07770
C                                                                       LEV07780
C     CONTINUE WITH THE LANCZOS VECTOR COMPUTATIONS?                    LEV07790
      IF(LVCONT.EQ.0.AND.MREJEC.NE.0) GO TO 1450                        LEV07800
C                                                                       LEV07810
C     NOW COMPUTE THE RITZ VECTORS.  REGENERATE THE                     LEV07820
C     LANCZOS VECTORS.                                                  LEV07830
C                                                                       LEV07840
      DO 900 I = 1,NMAX                                                 LEV07850
  900 RITVEC(I) = ZERO                                                  LEV07860
C                                                                       LEV07870
C-----------------------------------------------------------------------LEV07880
C     REGENERATE THE STARTING VECTOR. THIS MUST BE GENERATED AND        LEV07890
C     NORMALIZED PRECISELY THE WAY IT WAS DONE IN THE EIGENVALUE        LEV07900
C     COMPUTATIONS, OTHERWISE THERE WILL BE A MISMATCH BETWEEN          LEV07910
C     THE T-EIGENVECTORS THAT HAVE BEEN COMPUTED FROM THE T-MATRICES    LEV07920
C     READ IN FROM FILE 2 AND THE LANCZOS VECTORS THAT ARE              LEV07930
C     BEING REGENERATED.                                                LEV07940
C                                                                       LEV07950
      IIL = SVSEED                                                      LEV07960
      CALL GENRAN(IIL,G,N)                                              LEV07970
C                                                                       LEV07980
C-----------------------------------------------------------------------LEV07990
C                                                                       LEV08000
      DO 910 J = 1,N                                                    LEV08010
  910 V2(J) = G(J)                                                      LEV08020
C                                                                       LEV08030
      SUM = FINPRO(N,V2(1),1,V2(1),1)                                   LEV08040
      SUM = ONE/DSQRT(SUM)                                              LEV08050
C                                                                       LEV08060
      DO 920 J = 1,N                                                    LEV08070
      V1(J) = ZERO                                                      LEV08080
  920 V2(J) = V2(J)*SUM                                                 LEV08090
C                                                                       LEV08100
C     LOOP FOR GENERATING RITZ VECTORS  (IVEC = 1,KMAXU)                LEV08110
      IVEC = 1                                                          LEV08120
      BATA = ZERO                                                       LEV08130
C                                                                       LEV08140
      GO TO 980                                                         LEV08150
```

```
C                                                                       LEV08160
  930 CONTINUE                                                          LEV08170
C                                                                       LEV08180
C     COMPUTE V1 = A*V2 - BATA*V1                                       LEV08190
C                                                                       LEV08200
C-----------------------------------------------------------------------LEV08210
C                                                                       LEV08220
      CALL CMATV(V2,V1,BATA)                                            LEV08230
C                                                                       LEV08240
C-----------------------------------------------------------------------LEV08250
C                                                                       LEV08260
      ALFA = FINPRO(N,V1(1),1,V2(1),1)                                  LEV08270
C                                                                       LEV08280
      DO 940 J = 1,N                                                    LEV08290
  940 V1(J) = V1(J)-ALFA*V2(J)                                          LEV08300
C                                                                       LEV08310
      BATA = FINPRO(N,V1(1),1,V1(1),1)                                  LEV08320
      BATA = DSQRT(BATA)                                                LEV08330
      SUM = ONE/BATA                                                    LEV08340
C                                                                       LEV08350
      TEMP = BETA(IVEC)                                                 LEV08360
      TEMP = DABS(BATA - TEMP)/TEMP                                     LEV08370
      IF (TEMP.LT.1.0D-10)GO TO 960                                     LEV08380
C                                                                       LEV08390
C     THE BETA BEING REGENERATED DO NOT MATCH THE BETA IN FILE 2.       LEV08400
C     SOMETHING IS WRONG IN THE LANCZOS VECTOR GENERATION.              LEV08410
C     PROGRAM TERMINATES FOR USER TO CORRECT THE PROBLEM                LEV08420
C     WHICH MUST BE IN THE STARTING VECTOR GENERATION OR IN             LEV08430
C     THE MATRIX-VECTOR MULTIPLY SUBROUTINE CMATV SUPPLIED.             LEV08440
C     THIS SUBROUTINE MUST BE THE SAME ONE USED IN THE                  LEV08450
C     EIGENVALUE COMPUTATIONS OR A MISMATCH WILL ENSUE.                 LEV08460
C                                                                       LEV08470
      WRITE(6,950) IVEC,BATA,BETA(IVEC),TEMP                            LEV08480
  950 FORMAT(/2X,'IVEC',16X,'BATA',10X,'BETA(IVEC)',14X,'RELDIF'/I6,    LEV08490
     13E20.12/' IN LANCZOS VECTOR REGENERATION THE ENTRIES OF THE TRIDIALEV08500
     1GONAL MATRICES BEING'/' GENERATED ARE NOT THE SAME AS THOSE IN THELEV08510
     1 MATRIX SUPPLIED ON FILE 2.'/' THEREFORE SOMETHING IS BEING INITIALEV08520
     1LIZED OR COMPUTED DIFFERENTLY FROM THE WAY'/' IT WAS COMPUTED IN TLEV08530
     1HE EIGENVALUE COMPUTATIONS'/' THE PROGRAM TERMINATES FOR THE USER LEV08540
     1TO DETERMINE WHAT THE PROBLEM IS'/)                               LEV08550
      GO TO 1550                                                        LEV08560
C                                                                       LEV08570
C                                                                       LEV08580
  960 CONTINUE                                                          LEV08590
      DO 970 J = 1,N                                                    LEV08600
      TEMP = SUM*V1(J)                                                  LEV08610
      V1(J) = V2(J)                                                     LEV08620
  970 V2(J) = TEMP                                                      LEV08630
C                                                                       LEV08640
  980 CONTINUE                                                          LEV08650
C                                                                       LEV08660
      LFIN = 0                                                          LEV08670
      DO 1000 J = 1,NGOODC                                              LEV08680
      LL = LFIN                                                         LEV08690
      LFIN = LFIN + N                                                   LEV08700
```

```
C
      IF(IABS(MA(J)).LT.IVEC.OR.MP(J).EQ.MPMIN) GO TO 1000
      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 990 K = 1,N
      LL = LL + 1
  990 RITVEC(LL) = TEMP*V2(K) + RITVEC(LL)
C
 1000 CONTINUE
C
      IVEC = IVEC + 1
      IF (IVEC.LE.KMAXU) GO TO 930
C
C
C     RITZVECTOR GENERATION IS COMPLETE. 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 1050 J = 1,NGOODC
C
      KK = LFIN
      LFIN = LFIN + N
      IF(MP(J).EQ.MPMIN) GO TO 1050
C
      DO 1010 K = 1,N
      KK = KK + 1
 1010 V2(K) = RITVEC(KK)
C
      SUM = FINPRO(N,V2(1),1,V2(1),1)
      SUM = DSQRT(SUM)
      RNORM(J) = SUM
      TEMP = DABS(ONE-SUM)
      SUM = ONE/SUM
C
      KK = LFIN - N
      DO 1020 K = 1,N
      KK = KK + 1
      V2(K) = SUM*V2(K)
 1020 RITVEC(KK) = V2(K)
C
      IF (IWRITE.NE.0) WRITE(6,1030) J,GOODEV(J)
 1030 FORMAT(/I5,' TH EIGENVALUE CONSIDERED = ',E20.12/)
C
      IF (IWRITE.NE.0) WRITE(6,1040) TERR(J),TBETA(J),TEMP
 1040 FORMAT(' NORM OF ERROR IN T-EIGENVECTOR = ',E14.3/
     1 ' BETA(MA(J)+1)*U(MA(J)) = ',E14.3/
     1 ' ABS(NORM(RITVEC) - 1.0)  = ',E14.3/)
C
      LINT = LFIN - N + 1
      EVAL = EVNEW(J)
C
```

```
C-----------------------------------------------------------------------LEV09260
C                                                                       LEV09270
      CALL CMATV(RITVEC(LINT),V2,EVAL)                                  LEV09280
C                                                                       LEV09290
C-----------------------------------------------------------------------LEV09300
C                                                                       LEV09310
C     COMPUTE ERROR IN RITZ VECTOR CONSIDERED AS A EIGENVECTOR OF A.    LEV09320
C     V2 = A*RITVEC - EVAL*RITVEC                                       LEV09330
C                                                                       LEV09340
      SUM = FINPRO(N,V2(1),1,V2(1),1)                                   LEV09350
      SUM = DSQRT(SUM)                                                  LEV09360
      ERR(J) = SUM                                                      LEV09370
      GAP = ABS(AMINGP(J))                                              LEV09380
      ERRDGP(J) = SUM/GAP                                               LEV09390
C                                                                       LEV09400
 1050 CONTINUE                                                          LEV09410
C                                                                       LEV09420
C                                                                       LEV09430
C     RITZVECTORS ARE NORMALIZED AND ERROR ESTIMATES ARE IN ERR ARRAY   LEV09440
C     AND IN ERRDGP ARRAY. STORE EVERYTHING                             LEV09450
C                                                                       LEV09460
C                                                                       LEV09470
      WRITE(9,1060)                                                     LEV09480
 1060 FORMAT(3X,'A-EIGENVALUE',2X,'MA(J)',3X,'A-MINGAP',6X,'AERROR',2X, LEV09490
     1 'AERROR/GAP',6X,'TERROR')                                        LEV09500
C                                                                       LEV09510
      WRITE(13,1070)                                                    LEV09520
 1070 FORMAT(16X,'GOODEV(J)',5X,'RITZNORM',6X,'AMINGAP',5X,             LEV09530
     1 'TBETA(J)',5X,'TLAST(J)')                                        LEV09540
C                                                                       LEV09550
      DO 1100 J=1,NGOODC                                                LEV09560
C                                                                       LEV09570
      IF(MP(J).EQ.MPMIN) GO TO 1100                                     LEV09580
C                                                                       LEV09590
      WRITE(9,1080)EVNEW(J),MA(J),AMINGP(J),ERR(J),ERRDGP(J),TERR(J)    LEV09600
 1080 FORMAT(E15.8,I6,4E12.4)                                           LEV09610
C                                                                       LEV09620
      WRITE(13,1090) EVNEW(J),RNORM(J),AMINGP(J),TBETA(J),TLAST(J)      LEV09630
 1090 FORMAT(E25.14,4E13.5)                                             LEV09640
C                                                                       LEV09650
 1100 CONTINUE                                                          LEV09660
C                                                                       LEV09670
      IF(MREJEC.EQ.0) GO TO 1180                                        LEV09680
      WRITE(9,1110)                                                     LEV09690
 1110 FORMAT(/' RITZ VECTORS WERE NOT COMPUTED FOR THE FOLLOWING EIGENVALEV09700
     1LUES'/' EITHER BECAUSE THEY HAD NOT CONVERGED OR BECAUSE THE ERRORLEV09710
     1 ESTIMATE'/'  WAS NOT AS SMALL AS DESIRED'/)                      LEV09720
C                                                                       LEV09730
      DO 1170 J = 1,NGOODC                                              LEV09740
      IF(MP(J).NE.MPMIN) GO TO 1170                                     LEV09750
C     WRITE OUT MESSAGE FOR EACH EIGENVALUE FOR WHICH NO EIGENVECTOR    LEV09760
C     WAS COMPUTED.                                                     LEV09770
C                                                                       LEV09780
      WRITE(9,1120)                                                     LEV09790
 1120 FORMAT(6X,'GOODEV(J)',3X,'MA(J)',5X,'AMINGP(J)',6X,'TLAST(J)',3X, LEV09800
```

```
     1'MP(J)')
      WRITE(9,1130) GOODEV(J),MA(J),AMINGP(J),TBETA(J),MP(J)
 1130 FORMAT(E15.8,I8,2E14.4,I8)
C
      WRITE(13,1140)
 1140 FORMAT(/' RITZ VECTORS WERE NOT COMPUTED FOR THE FOLLOWING EIGENVA
     1LUES'/' BECAUSE THEY HAD NOT CONVERGED'/)
C
      WRITE(13,1150)
 1150 FORMAT(6X,'GOODEV(J)',3X,'MA(J)',3X,'M1(J)',3X,'M2(J)',3X,'MP(J)'
     1/)
      WRITE(13,1160) GOODEV(J),MA(J),M1(J),M2(J),MP(J)
 1160 FORMAT(E15.8,4I8)
C
 1170 CONTINUE
 1180 CONTINUE
C
      WRITE(9,1190)
 1190 FORMAT(/' ABOVE ARE ERROR ESTIMATES FOR THE A AND T EIGENVECTORS'/
     1 ' ASSOCIATED WITH THE GOODEV LISTED IN COLUMN 1'/
     1 ' AERROR = NORM(A*X - EV*X)  TERROR = NORM(T*Y - EV*Y) '/
     1 ' WHERE T = T(1,MA(J))    X = RITZ VECTOR = V*Y  V = SUCCESSIVE'/
     1 ' LANCZOS VECTORS. AMINGAP = GAP TO NEAREST A-EIGENVALUE'//)
C
      WRITE(13,1200)
 1200 FORMAT(/' ABOVE ARE ERROR ESTIMATES ASSOCIATED WITH THE GOODEV'/
     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 ' AMINGAP = GAP TO NEAREST A-EIGENVALUE'/)
C
C     NUMBER OF RITZ VECTORS COMPUTED
      NCOMPU = NGOODC - MREJEC
      WRITE(12,1210) N,NCOMPU,NGOODC,MATNO
 1210 FORMAT(3I6,I12,' SIZE A, NO.RITZVECS, NO.EVALUES,MATNO')
C
      LFIN = 0
      DO 1270 J = 1,NGOODC
      LINT = LFIN + 1
      LFIN = LFIN + N
C
      IF(MP(J).EQ.MPMIN) GO TO 1250
C     RITZ VECTOR WAS COMPUTED
      WRITE(12,1220) J, GOODEV(J), MP(J)
 1220 FORMAT(I6,4X,E20.12,I6,' J, EIGENVAL, MP(J)')
C
      WRITE(12,1230) ERR(J),ERRDGP(J)
 1230 FORMAT(2E15.5,' = NORM(A*Z-EVAL*Z) AND  NORM(A*Z-EVAL*Z)/MINGAP')
C
      WRITE(12,1240) (RITVEC(LL), LL=LINT,LFIN)
 1240 FORMAT(4E20.12)
      GO TO 1270
C     NO RITZ VECTOR WAS COMPUTED FOR THIS EIGENVALUE
 1250 WRITE(12,1260) J,GOODEV(J),MP(J)
 1260 FORMAT(I6,4X,E20.12,I6,' J,EIGVALUE,NO RITZ VECTOR COMPUTED')
```

```
C                                                                       LEV10360
 1270 CONTINUE                                                          LEV10370
C                                                                       LEV10380
C     DID ANY T-MATRICES INCLUDE OFF-DIAGONAL ENTRIES SMALLER THAN      LEV10390
C     DESIRED, AS SPECIFIED BY BTOL?                                    LEV10400
C                                                                       LEV10410
      IF(IB.GT.0) GO TO 1300                                            LEV10420
C                                                                       LEV10430
      WRITE(6,1280) KMAXU                                               LEV10440
 1280 FORMAT(/' FOR LARGEST T-MATRIX CONSIDERED',I7,' CHECK THE SIZE OF LEV10450
     1BETAS')                                                           LEV10460
C                                                                       LEV10470
C-----------------------------------------------------------------------LEV10480
C                                                                       LEV10490
      CALL TNORM(ALPHA,BETA,BKMIN,TEMP,KMAXU,IBMT)                      LEV10500
C                                                                       LEV10510
C-----------------------------------------------------------------------LEV10520
C                                                                       LEV10530
      IF(IBMT.LT.0) WRITE (6,1290)                                      LEV10540
 1290 FORMAT(/' WARNING THE T-MATRICES FOR ONE OR MORE OF THE EIGENVALUELEV10550
     1S CONSIDERED'/' HAD AN OFF-DIAGONAL ENTRY THAT WAS SMALLER THAN THLEV10560
     1E BETA TOLERANCE THAT WAS SPECIFIED'/)                            LEV10570
 1300 CONTINUE                                                          LEV10580
C                                                                       LEV10590
       GO TO 1550                                                       LEV10600
C                                                                       LEV10610
 1310 WRITE(6,1320) NGOOD,NMAX,MDIMRV                                   LEV10620
 1320 FORMAT(/I4,' RITZ VECTORS WERE REQUESTED BUT THE REQUIRED DIMENSIOLEV10630
     1N',I6/' IS LARGER THAN THE USER-SPECIFIED DIMENSION OF RITVEC',I6 LEV10640
     1/' THEREFORE, THE EIGENVECTOR PROCEDURE TERMINATES FOR THE USER TOLEV10650
     1 INTERVENE')                                                      LEV10660
C                                                                       LEV10670
      GO TO 1550                                                        LEV10680
C                                                                       LEV10690
 1330 WRITE(6,1340) NOLD,N,MATOLD,MATNO                                 LEV10700
 1340 FORMAT(//' PARAMETERS READ FROM FILE 3 DO NOT AGREE WITH THOSE SPELEV10710
     1CIFIED BY THE USER'/' N,NOLD,MATOLD,MATNO = ',2I6,2I12/' PROGRAM TLEV10720
     1ERMINATES FOR USER TO RESOLVE PROBLEM'/)                          LEV10730
C                                                                       LEV10740
      GO TO 1550                                                        LEV10750
C                                                                       LEV10760
 1350 WRITE(6,1360)                                                     LEV10770
 1360 FORMAT(//' PARAMETERS IN THE ALPHA,BETA FILE HEADER DO NOT AGREE WLEV10780
     1ITH PARAMTERS'/' SPECIFIED BY THE USER.  THEREFORE THE PROGRAM TERLEV10790
     1MINATES FOR THE USER'/' TO RESOLVE THE PROBLEM'/)                 LEV10800
C                                                                       LEV10810
      GO TO 1550                                                        LEV10820
C                                                                       LEV10830
 1370 WRITE(6,1380) KMAX,MEV                                            LEV10840
 1380 FORMAT(/' ALPHA,BETA FILE HEADER GIVES KMAX =',I6/                LEV10850
     1' BUT EIGENVALUES WERE COMPUTED AT MEV = ',I6,' PROGRAM STOPS'/)  LEV10860
C                                                                       LEV10870
      GO TO 1550                                                        LEV10880
C                                                                       LEV10890
 1390 WRITE(6,1400)                                                     LEV10900
```

```
 1400 FORMAT(//' PROGRAM COMPUTED 1ST GUESSES AT T-MATRIX SIZES AND READLEV10910
     1 THEM TO'/' FILE 10, THEN TERMINATED AS REQUESTED.')              LEV10920
      GO TO 1550                                                        LEV10930
C                                                                       LEV10940
 1410 WRITE(6,1420) MTOL, MDIMTV                                        LEV10950
 1420 FORMAT(/' PROGRAM TERMINATES BECAUSE THE TVEC DIMENSION ANTICIPATELEV10960
     1D',I7/'  IS LARGER THAN THE TVEC DIMENSION',I7,' SPECIFIED BY THE LEV10970
     1USER.'/'  USER MAY RESET THE TVEC DIMENSION AND RESTART THE PROGRALEV10980
     1M')                                                               LEV10990
      GO TO 1550                                                        LEV11000
C                                                                       LEV11010
 1430 WRITE(6,1440)                                                     LEV11020
 1440 FORMAT(/' PROGRAM TERMINATES BECAUSE NO SUITABLE T-EIGENVECTORS WELEV11030
     1RE IDENTIFIED'/'  FOR ANY OF THE EIGENVALUES SUPPLIED.  PROBLEM COLEV11040
     1ULD BE CAUSED'/'  BY TOO SMALL A TVEC DIMENSION OR SIMPLY THAT SUILEV11050
     1TABLE T-VECTORS COULD'/'  NOT BE IDENTIFIED.  USER SHOULD CHECK OULEV11060
     1TPUT'/)                                                           LEV11070
      GO TO 1550                                                        LEV11080
C                                                                       LEV11090
 1450 WRITE(6,1460) LVCONT,NTVEC,NGOOD                                  LEV11100
 1460 FORMAT(/' LVCONT FLAG =',I2,' AND NUMBER ',I5,' OF T-EIGENVECTORS LEV11110
     1 COMPUTED N.E.'/' NUMBER',I5,' REQUESTED SO PROGRAM TERMINATES'/) LEV11120
      GO TO 1550                                                        LEV11130
C                                                                       LEV11140
 1470 WRITE(6,1480)                                                     LEV11150
 1480 FORMAT(//' PROGRAM TERMINATES WITHOUT COMPUTING RITZ VECTORS'/    LEV11160
     1' BECAUSE ALL T-EIGENVECTORS WERE REJECTED AS NOT SUITABLE FOR THELEV11170
     1 RITZ VECTOR'/' COMPUTATIONS.  PROBABLE CAUSE IS LACK OF CONVERGENLEV11180
     1CE OF THE EIGENVALUES SUPPLIED'/)                                 LEV11190
       GO TO 1550                                                       LEV11200
C                                                                       LEV11210
 1490 WRITE(6,1500)                                                     LEV11220
 1500 FORMAT(/' PROGRAM INDICATES THAT IT IS NOT POSSIBLE TO COMPUTE ANYLEV11230
     1 OF THE'/' REQUESTED EIGENVECTORS. THEREFORE PROGRAM TERMINATES') LEV11240
      DO 1510 J=1,NGOODC                                                LEV11250
 1510 WRITE(6,1520)  J,GOODEV(J),MP(J)                                  LEV11260
 1520 FORMAT(/4X,' J',11X,'GOODEV(J)',4X,'MP(J)'/I6,E20.12,I9)          LEV11270
      GO TO 1550                                                        LEV11280
C                                                                       LEV11290
 1530 WRITE(6,1540) MBETA,KMAXN                                         LEV11300
 1540 FORMAT(/' PROGRAM TERMINATES BECAUSE THE STORAGE ALLOTTED FOR THE LEV11310
     1BETA ARRAY',I8/' IS NOT SUFFICIENT FOR THE ENLARGED KMAX =',I8,' TLEV11320
     1HAT THE PROGRAM WANTS'/' USER CAN ENLARGE THE DIMENSIONS OF THE ALLEV11330
     1PHA AND BETA ARRAYS'/' AND RERUN THE PROGRAM'/)                   LEV11340
C                                                                       LEV11350
 1550 CONTINUE                                                          LEV11360
C                                                                       LEV11370
      STOP                                                              LEV11380
C-----END OF MAIN PROGRAM FOR LANCZOS EIGENVECTORS--------------------- LEV11390
      END                                                               LEV11400
```

2.5 LEMULT: LANCZS and Sample Matrix-Vector Multiply Subroutines

```
C-----LEMULT-----------------------------------------------------------
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 SUBROUTINES  LANCZS, USPEC, AND CMATV
C     TO BE USED WITH THE REAL SYMMETRIC VERSION OF THE LANCZOS
C     EIGENVALUE/EIGENVECTOR PROCEDURES.
C     ALSO CONTAINS SUBROUTINES FOR POISSON TEST MATRICES THAT ALLOW
C     COMPUTATION OF TRUE ERRORS IN COMPUTED EIGENVALUES AND
C     IN CORRESPONDING EIGENVECTORS.
C
C     NONPORTABLE CONSTRUCTIONS:
C     1.  THE ENTRY MECHANISM USED TO PASS THE STORAGE
C         LOCATIONS OF THE USER-SPECIFIED MATRIX FROM THE
C         SUBROUTINE USPEC TO THE MATRIX-VECTOR SUBROUTINE
C         CMATV.
C     2.  IN THE SAMPLE USPEC AND CMATV FOR DIAGONAL TEST MATRICES:
C         FREE FORMAT (8,*) AND THE FORMAT (20A4).
C     3.  IN THE POISSON SUBROUTINES PROVIDED, THE DATA MACHEP
C         DEFINITION AND MANY OF THE INDICES FOR ARRAYS ARE NOT
C         IN A PORTABLE CONSTRUCTION.  THESE PROGRAMS SHOULD BE
C         REMOVED FROM THE LEMULT FILE IF THE USER IS NOT USING THEM.
C
C-----LANCZS-COMPUTE THE LANCZOS TRIDIAGONAL MATRICES-------------------
C
      SUBROUTINE LANCZS(MATVEC,ALPHA,BETA,V1,V2,G,KMAX,MOLD1,N,IIX)
C
C----------------------------------------------------------------------
```

```
      DOUBLE PRECISION  ALPHA(1),BETA(1),V1(1),V2(1),SUM,TEMP,ONE,ZERO
      REAL  G(1)
      DOUBLE PRECISION FINPRO, DSQRT
      EXTERNAL MATVEC
C-----------------------------------------------------------------------
C
      ZERO = 0.D0
      ONE = 1.D0
C
      IF(MOLD1.GT.1)GO TO 30
C
C     ALPHA/BETA GENERATION STARTS AT I = 1
C     MOLD1 = 1 SET V1 = 0. AND V2 = RANDOM UNIT VECTOR
      BETA(1) = ZERO
      IIL=IIX
C
C-----------------------------------------------------------------------
      CALL GENRAN(IIL,G,N)
C-----------------------------------------------------------------------
C
      DO 10 I = 1,N
   10 V2(I) = G(I)
C
C-----------------------------------------------------------------------
      SUM = FINPRO(N,V2(1),1,V2(1),1)
C-----------------------------------------------------------------------
C
      SUM = ONE/DSQRT(SUM)
      DO 20 I = 1,N
      V1(I) = ZERO
   20 V2(I) = V2(I)*SUM
C
C     ALPHA BETA GENERATION LOOP
   30 CONTINUE
C
      DO 60 I=MOLD1,KMAX
      SUM = BETA(I)
C     MATVEC(V2,V1,SUM) CALCULATES  V1 = A*V2 - SUM*V1
C
C-----------------------------------------------------------------------
      CALL MATVEC(V2,V1,SUM)
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      SUM = FINPRO(N,V1(1),1,V2(1),1)
C-----------------------------------------------------------------------
C
      ALPHA(I) = SUM
      DO 40 J=1,N
   40 V1(J) = V1(J)-SUM*V2(J)
C
C-----------------------------------------------------------------------
      SUM = FINPRO(N,V1(1),1,V1(1),1)
C-----------------------------------------------------------------------
C
```

```
      IN = I+1
      BETA(IN) = DSQRT(SUM)
      SUM = ONE/BETA(IN)
      DO 50 J=1,N
      TEMP = SUM*V1(J)
      V1(J) = V2(J)
   50 V2(J) = TEMP
   60 CONTINUE
C
C     END ALPHA, BETA GENERATION LOOP
C
      RETURN
C-----END OF LANCZS---------------------------------------------------
      END
C
C-----USPEC (GENERAL SYMMETRIC SPARSE MATRICES)-------------------------
C
C     SUBROUTINE USPEC(N,MATNO)
      SUBROUTINE GUSPEC(N,MATNO)
C
C----------------------------------------------------------------------
      DOUBLE PRECISION  A(10000),AD(5010)
      INTEGER  IROW(10000),ICOL(5010)
C----------------------------------------------------------------------
C    USPEC DIMENSIONS AND INITIALIZES THE ARRAYS NEEDED TO DEFINE
C    THE USER-SPECIFIED MATRIX AND THEN PASSES THE STORAGE LOCATIONS
C    OF THESE ARRAYS TO THE MULTIPLY SUBROUTINE CMATV.
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
C             IN COLUMN J.
C    IROW(K), K = 1,NZS IS THE CORRESPONDING ROW INDEX FOR A(K).
C    AD(I), I=1,N CONTAINS DIAGONAL ENTRIES (INCLUDING ANY 0
C           DIAGONAL ENTRIES).
C    A(K), K=1,NZS CONTAINS NONZERO SUBDIAGONAL ENTRIES, BY COLUMN
C    FOR J > NZL THERE ARE NO NONZERO SUBDIAGONAL ELEMENTS IN COLUMN J.
C    ICOL(J) = 0 IS ALLOWED
C
C----------------------------------------------------------------------
C     ARRAYS THAT DEFINE THE MATRIX ARE READ IN FROM FILE 8
C
      READ(8,10) NZS,NOLD,NZL,MATOLD
   10 FORMAT(I10,2I6,I8)
C
      WRITE(6,20) NZS,NOLD,NZL,MATOLD
   20 FORMAT(I10,2I6,I8,' = NZS,NOLD,NZL,MATOLD'/)
C
C     TEST OF PARAMETER CORRECTNESS
      ITEMP = (NOLD-N)**2 + (MATNO-MATOLD)**2
C
      IF(ITEMP.EQ.0) GO TO 40
C
```

```
      WRITE(6,30)
   30 FORMAT(' PROGRAM TERMINATES BECAUSE EITHER ORDERS OF OR LABELS FOR
     1 MATRIX DISAGREE')
      GO TO 70
C
   40 CONTINUE
C
C     NUMBER OF NONZERO SUBDIAGONAL ENTRIES IN EACH COLUMN IS READ
C     THEN THE CORRESPONDING ROW INDEX FOR EACH SUCH ENTRY IS READ
      READ(8,50) (ICOL(K), K=1,NZL)
      READ(8,50) (IROW(K), K=1,NZS)
   50 FORMAT(13I6)
C
C     DIAGONAL IS READ FIRST, THEN NONZERO BELOW DIAGONAL ENTRIES
      READ(8,60) (AD(K), K=1,N)
      READ(8,60) (A(K), K=1,NZS)
   60 FORMAT(4E19.10)
C
C-----------------------------------------------------------------------
C     PASS STORAGE LOCATIONS OF ARRAYS THAT DEFINE THE MATRIX TO
C     THE MATRIX-VECTOR MULTIPLY SUBROUTINE CMATV
C
      CALL CMATVE(A,AD,ICOL,IROW,N,NZL)
C
C-----------------------------------------------------------------------
C
      RETURN
   70 STOP
C-----END OF USPEC------------------------------------------------------
      END
C
C-----MATRIX-VECTOR MULTIPLY FOR REAL SPARSE SYMMETRIC MATRICES---------
C
C     SUBROUTINE CMATV(W,U,SUM)
      SUBROUTINE GCMATV(W,U,SUM)
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION   U(1),W(1),A(1),AD(1),SUM
      INTEGER  IROW(1),ICOL(1)
C-----------------------------------------------------------------------
C    SPARSE MATRIX-VECTOR MULTIPLY FOR LANCZS  U = A*W - SUM*U
C    SEE USPEC SUBROUTINE FOR DESCRIPTION OF THE ARRAYS THAT DEFINE
C    THE MATRIX
C-----------------------------------------------------------------------
C
      GO TO 3
C     STORAGE LOCATIONS OF ARRAYS ARE PASSED TO CMATV FROM USPEC
      ENTRY CMATVE(A,AD,ICOL,IROW,N,NZL)
      GO TO 4
C-----------------------------------------------------------------------
C     COMPUTE THE DIAGONAL TERMS
    3 DO 10 I = 1,N
   10 U(I) = AD(I)*W(I)-SUM*U(I)
C
C     COMPUTE BY COLUMN
```

```
      LLAST = 0
      DO 30 J = 1,NZL
C
      IF (ICOL(J).EQ.0) GO TO 30
      LFIRST = LLAST + 1
      LLAST = LLAST + ICOL(J)
C
      DO 20 L = LFIRST,LLAST
      I = IROW(L)
C
      U(I) = U(I) + A(L)*W(J)
      U(J) = U(J) + A(L)*W(I)
C
   20 CONTINUE
C
   30 CONTINUE
C
    4 RETURN
C
C-----END OF CMATV--------------------------------------------------------
      END
C
C-----MATRIX-VECTOR MULTIPLY FOR DIAGONAL TEST MATRICES------------------
C
C     SUBROUTINE CMATV(W,U,SUM)
      SUBROUTINE DCMATV(W,U,SUM)
C
C     CMATV COMPUTES  U = (DIAGONAL MATRIX) * W - SUM * U
C-----------------------------------------------------------------------
      DOUBLE PRECISION  W(1),U(1),SUM
      DOUBLE PRECISION  D(1)
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)
    4 RETURN
C
C-----END OF DIAGONAL TEST MATRIX  MULTIPLY-----------------------------
      END
C
C
C-----START OF USPEC FOR DIAGONAL TEST MATRIX---------------------------
C
C     SUBROUTINE USPEC(N,MATNO)
      SUBROUTINE DUSPEC(N,MATNO)
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION  D(1000), SHIFT, SPACE
      DOUBLE PRECISION  DABS, DFLOAT
      REAL  EXPLAN(20)
C-----------------------------------------------------------------------
```

```
C                                                                       LEM02650
      READ(8,10) EXPLAN                                                 LEM02660
   10 FORMAT(20A4)                                                      LEM02670
      READ(8,*) NOLD,NUNIF,SPACE,D(1),SHIFT                             LEM02680
      NNUNIF = NOLD - NUNIF                                             LEM02690
      WRITE(6,20) NOLD,SPACE,NNUNIF,D(1),SHIFT                          LEM02700
   20 FORMAT(/' DIAGONAL TEST MATRIX, SIZE = ',I4/' MOST ENTRIES ARE ', LEM02710
     1E10.3,' UNITS APART.',I3,' ENTRIES'/' ARE IRREGULARLY SPACED. FIRSLEM02720
     1T ENTRY IS ',E10.3,' SHIFT = ',E10.3/)                            LEM02730
C                                                                       LEM02740
      IF(N.NE.NOLD) GO TO 90                                            LEM02750
C     COMPUTE THE UNIFORM PORTION OF THE SPECTRUM                       LEM02760
      DO 30 J=2,NUNIF                                                   LEM02770
   30 D(J) = D(1) - DFLOAT(J-1)*SPACE                                   LEM02780
      NUNIF1=NUNIF + 1                                                  LEM02790
      READ(8,10)  EXPLAN                                                LEM02800
      DO 40 J=NUNIF1,N                                                  LEM02810
   40 READ(8,*) D(J)                                                    LEM02820
      NB = NUNIF - 2                                                    LEM02830
C                                                                       LEM02840
      IF(SHIFT.EQ.0.) GO TO 60                                          LEM02850
      DO 50 J=1,N                                                       LEM02860
   50 D(J) = D(J) + SHIFT                                               LEM02870
C                                                                       LEM02880
C     PRINT OUT THE EIGENVALUES OF INTEREST                             LEM02890
   60 WRITE(6,70) (D(I), I=1,10 )                                       LEM02900
      WRITE(6,80) (D(I), I = NB,N)                                      LEM02910
   70 FORMAT(/' REAL SYMMETRIC LANCZOS TEST, 1ST 10 ENTRIES OF DIAGONAL LEM02920
     1TEST MATRIX = '/(3E22.14))                                        LEM02930
   80 FORMAT(/' MIDDLE UNIFORM PORTION OF MATRIX IS NOT PRINTED OUT'/   LEM02940
     1' END OF UNIFORM PLUS NONUNIFORM SECTION = '/(3E25.16))           LEM02950
C                                                                       LEM02960
C     DIAGONAL GENERATION COMPLETE                                      LEM02970
C                                                                       LEM02980
C    CALL ENTRY TO MATRIX-VECTOR MULTIPLY SUBROUTINE  TO PASS           LEM02990
C    STORAGE LOCATION OF D-ARRAY AND ORDER OF A-MATRIX.                 LEM03000
       CALL MVDIAE(D,N)                                                 LEM03010
C                                                                       LEM03020
      RETURN                                                            LEM03030
   90 WRITE(6,100) NOLD,N                                               LEM03040
  100 FORMAT(' PROGRAM TERMINATES BECAUSE NOLD = ',I5,'DOES NOT EQUAL N LEM03050
     1 =',I5)                                                           LEM03060
C-----END OF USPEC SUBROUTINE FOR DIAGONAL TEST MATRICES----------------LEM03070
      STOP                                                              LEM03080
      END                                                               LEM03090
C                                                                       LEM03100
C-----POISSON TEST MATRICES---------------------------------------------LEM03110
C                                                                       LEM03120
C     CONTAINS SUBROUTINES USPEC, CMATV, EXEVG, EXERR AND  EXVEC        LEM03130
C                                                                       LEM03140
C-----START OF USPEC----------------------------------------------------LEM03150
C                                                                       LEM03160
      SUBROUTINE USPEC(N,MATNO)                                         LEM03170
C     SUBROUTINE PUSPEC(N,MATNO)                                        LEM03180
C                                                                       LEM03190
```

```
C-----------------------------------------------------------------------
      DOUBLE PRECISION  C0,C1,C2,HALF,ONE
      REAL  EXPLAN(20)
C--------------------------------------------------------------------
      HALF = 0.5D0
      ONE  = 1.0D0
C
C     READ USER-SPECIFIED PARAMETERS FROM INPUT FILE 8 (FREE FORMAT)
C
      READ(8,10) EXPLAN
      WRITE(6,10) EXPLAN
   10 FORMAT(20A4)
C
      READ(8,10) EXPLAN
      READ(8,*) KX,KY,C0
      N = KX*KY
      C1 = HALF-C0
      C2 = ONE
C
      WRITE(6,20) N,KX,KY,C2,C0,C1
   20 FORMAT(/5X,'N',4X,'KX',4X,'KY',7X,'DIAGONAL',3X,'X-CODIAGONAL',
     1 3X,'Y-CODIAGONAL'/3I6,3E15.8/)
C
C-----------------------------------------------------------------------
      CALL PMATVE(C0,C1,C2,KX,KY)
      CALL EXEVE(C0,C1,C2,KX,KY)
      CALL EXERRP(C0,C1,C2,KX,KY)
C     CALL EXVECP(C0,C1,C2,KX,KY)
C-----------------------------------------------------------------------
C
      RETURN
C-----END OF USPEC------------------------------------------------------
      END
C
C-----START OF CMATV----------------------------------------------------
C     CALCULATE U = A*W - SUM*U FOR REAL POISSON MATRICES
C
      SUBROUTINE CMATV(W,U,SUM)
C     SUBROUTINE PMATV(W,U,SUM)
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION  U(1),W(1)
      DOUBLE PRECISION  C0,C1,C2,CC0,CC1,SUM
C-----------------------------------------------------------------------
      GO TO 3
      ENTRY  PMATVE(CC0,CC1,C2,KX,KY)
C-----------------------------------------------------------------------
C
      C0 = -CC0
      C1 = -CC1
      GO TO 4
C
    3 N = KX*KY
      KX1 = KX-1
      KY1 = KY-1
```

```
C                                                                       LEM03750
      KK = 1                                                            LEM03760
      U(KK) = (C2*W(KK)+C0*W(KK+1)+C1*W(KK+KX)) - SUM*U(KK)             LEM03770
      KK = KX                                                           LEM03780
      U(KK) = (C2*W(KK)+C0*W(KK-1)+C1*W(KK+KX)) - SUM*U(KK)             LEM03790
      KK = N - KX + 1                                                   LEM03800
      U(KK) = (C2*W(KK)+C0*W(KK+1)+C1*W(KK-KX)) - SUM*U(KK)             LEM03810
      KK = N                                                            LEM03820
      U(KK) = (C2*W(KK)+C0*W(KK-1)+C1*W(KK-KX)) - SUM*U(KK)             LEM03830
C                                                                       LEM03840
      DO 10 J = 2,KX1                                                   LEM03850
      KK = J                                                            LEM03860
      U(KK) = (C2*W(KK)+C0*W(KK-1)+C0*W(KK+1)+C1*W(KK+KX)) - SUM*U(KK)  LEM03870
      KK = J+N-KX                                                       LEM03880
      U(KK) = (C2*W(KK)+C0*W(KK-1)+C0*W(KK+1)+C1*W(KK-KX))-SUM*U(KK)    LEM03890
   10 CONTINUE                                                          LEM03900
C                                                                       LEM03910
      DO 30 J = 2,KY1                                                   LEM03920
      KK = (J-1)*KX + 1                                                 LEM03930
      U(KK) = (C2*W(KK)+C0*W(KK+1)+C1*W(KK-KX)+C1*W(KK+KX)) - SUM*U(KK) LEM03940
      DO 20 I = 2,KX1                                                   LEM03950
      KK = KK + 1                                                       LEM03960
      U(KK) = (C2*W(KK)+C0*W(KK-1)+C0*W(KK+1)+C1*W(KK-KX)               LEM03970
     1 +C1*W(KK+KX)) - SUM*U(KK)                                        LEM03980
   20 CONTINUE                                                          LEM03990
      KK = KK + 1                                                       LEM04000
      U(KK) = (C2*W(KK)+C0*W(KK-1)+C1*W(KK-KX)+C1*W(KK+KX)) - SUM*U(KK) LEM04010
   30 CONTINUE                                                          LEM04020
C                                                                       LEM04030
    4 RETURN                                                            LEM04040
C                                                                       LEM04050
C-----END OF CMATV------------------------------ ----------------------LEM04060
      END                                                               LEM04070
C                                                                       LEM04080
C-----START OF EXEVG---------------------------------------------------LEM04090
C                                                                       LEM04100
C     COMPUTES TRUE EIGENVALUES OF POISSON MATRIX, GAPS BETWEEN         LEM04110
C     TRUE EIGENVALUES, AND MULTIPLICITIES OF TRUE EIGENVALUES          LEM04120
C     AND STORE THESE VALUES, RESPECTIVELY, IN U, G, AND MP.            LEM04130
C     THESE QUANTITIES ARE WRITTEN OUT TO FILE 9                        LEM04140
C                                                                       LEM04150
      SUBROUTINE EXEVG(U,G,MP)                                          LEM04160
C                                                                       LEM04170
C----------------------------------------------------------------------LEM04180
      DOUBLE PRECISION  U(*)                                            LEM04190
      DOUBLE PRECISION MACHEP,EPSM,C0,C1,C2,T0,T1,PIK,PIL,ONE,TWO       LEM04200
      DOUBLE PRECISION ATOLN,EE                                         LEM04210
      REAL G(1)                                                         LEM04220
      INTEGER MP(1)                                                     LEM04230
C----------------------------------------------------------------------LEM04240
      DATA MACHEP/Z3410000000000000/                                    LEM04250
      EPSM = 2.0D0*MACHEP                                               LEM04260
C----------------------------------------------------------------------LEM04270
      GO TO 3                                                           LEM04280
      ENTRY EXEVE(C0,C1,C2,KX,KY)                                       LEM04290
```

```
      GO TO 4                                                           LEM04300
C-----------------------------------------------------------------------LEM04310
    3 N = KX*KY                                                         LEM04320
      ONE  = 1.0D0                                                      LEM04330
      TWO  = 2.0D0                                                      LEM04340
      T0 = DARCOS(-ONE)                                                 LEM04350
      T1 = DFLOAT(KX+1)                                                 LEM04360
      PIK = T0/T1                                                       LEM04370
      T1 = DFLOAT(KY+1)                                                 LEM04380
      PIL = T0/T1                                                       LEM04390
C     GENERATE TRUE EIGENVALUES                                         LEM04400
      KP = 0                                                            LEM04410
      DO 20 J = 1,KY                                                    LEM04420
      T1 = PIL*DFLOAT(J)                                                LEM04430
      T0 = C2 - TWO*C1*DCOS(T1)                                         LEM04440
      DO 10 I = 1,KX                                                    LEM04450
      KP = KP+1                                                         LEM04460
      T1 = PIK*DFLOAT(I)                                                LEM04470
   10 U(KP) = T0 - TWO*C0*DCOS(T1)                                      LEM04480
   20 CONTINUE                                                          LEM04490
C                                                                       LEM04500
C     ORDER U VECTOR BY INCREASING ALGEBRAIC SIZE                       LEM04510
      DO 40 K = 2,N                                                     LEM04520
      KM1 = K-1                                                         LEM04530
      DO 30 L = 1,KM1                                                   LEM04540
      JJ = K-L                                                          LEM04550
      IF (U(JJ+1).GE.U(JJ)) GO TO 40                                    LEM04560
      T0 = U(JJ)                                                        LEM04570
      U(JJ) = U(JJ+1)                                                   LEM04580
   30 U(JJ+1) = T0                                                      LEM04590
   40 CONTINUE                                                          LEM04600
      ATOLN = DMAX1(DABS(U(1)),DABS(U(N)))*EPSM                         LEM04610
C                                                                       LEM04620
      WRITE(9,50)                                                       LEM04630
   50 FORMAT(' TRUE EIGENVALUES FOR POISSON'/)                          LEM04640
C                                                                       LEM04650
      WRITE(9,60)N,KX,KY,C2,C0,C1,ATOLN                                 LEM04660
      WRITE(6,60) N,KX,KY,C2,C0,C1,ATOLN                                LEM04670
   60 FORMAT(1X,'A-SIZE',2X,'X-DIM',2X,'Y-DIM'/3I7/                     LEM04680
     1 5X,'A-DIAGONAL',3X,'X-CODIAGONAL',3X,'Y-CODIAGONAL',10X,'ATOLN'/ LEM04690
     2 4E15.8)                                                          LEM04700
C                                                                       LEM04710
C     DETERMINE MULTIPLICITIES FOR TRUE EIGENVALUES                     LEM04720
      I = 1                                                             LEM04730
      IDEX = 1                                                          LEM04740
      J = 1                                                             LEM04750
      NEXACT = 0                                                        LEM04760
   70 J = J+1                                                           LEM04770
      IF (J.GT.N) GO TO 80                                              LEM04780
      EE = DABS(U(J)-U(I))                                              LEM04790
      IF (EE.GT.ATOLN) GO TO 80                                         LEM04800
      IDEX = IDEX+1                                                     LEM04810
      GO TO 70                                                          LEM04820
   80 NEXACT = NEXACT+1                                                 LEM04830
      U(NEXACT) = U(I)                                                  LEM04840
```

```
      MP(NEXACT) = IDEX
C     MP(K) = MULTIPLICITY OF KTH EIGENVALUE CLUSTER FOR A
      IDEX = 1
      I = J
      IF (I.GT.N) GO TO 90
      GO TO 70
   90 CONTINUE
C
C     MULTIPLICITIES HAVE BEEN DETERMINED
C     NEXACT = NUMBER OF DISTINCT A-EIGENVALUES
C
      WRITE(9,100)NEXACT
      WRITE(6,100)NEXACT
  100 FORMAT(I6,' = NUMBER OF TRUE A-EIGENVALUES WHICH ARE DISTINCT'/)
C
C     MINGAP CALCULATION FOR DISTINCT A-EIGENVALUES
      NM1 = NEXACT - 1
      G(NEXACT) = U(NM1)-U(NEXACT)
      G(1) = U(2)-U(1)
C
      DO 110 J = 2,NM1
      T0 = U(J)-U(J-1)
      T1 = U(J+1)-U(J)
      G(J) = T1
      IF (T0.LT.T1) G(J) = -T0
  110 CONTINUE
C
C     NEXACT DISTINCT A-EIGENVALUES ARE IN U IN ASCENDING ORDER
C     MP = MULTIPLICITIES OF THE DISTINCT EIGENVALUES OF A
C     G = TRUE MINIMUM GAP IN A FOR EACH OF THESE EIGENVALUES
C     G < 0 INDICATES THE LEFT-HAND GAP WAS MINIMAL.
C     OUTPUT MULTIPLICITIES, DISTINCT EVS, AND MINGAPS TO FILE 9
C
      WRITE(9,120)
  120 FORMAT(5X,'I',1X,'AMULT',5X,'TRUE A-EIGENVALUE(I)',
     1 3X,'A-MINGAP(I)')
C
      WRITE(9,130)(J,MP(J),U(J),G(J), J=1,NEXACT)
  130 FORMAT(2I6,E25.16,E14.3)
C
      WRITE(9,140)
  140 FORMAT(' NEXACT DISTINCT A-EIGENVALUES ARE IN ASCENDING ORDER'/
     1 ' AMULT = MULTIPLICITIES OF THE DISTINCT EIGENVALUES OF A.'/
     2 ' A-MINGAP(I) = TRUE MINIMUM GAP IN A FOR EACH EIGENVALUE.'/
     3 ' A-MINGAP(I) LT 0 INDICATES THE LEFT-HAND GAP WAS MINIMAL.'//)
C
C     WE ORDER U VECTOR BY INCREASING SIZE OF THE GAPS
C
      DO 150 K = 1,NEXACT
  150 MP(K) = K
C
      DO 170 K = 2,NEXACT
      KM1 = K-1
C
      DO 160 L = 1,KM1
```

```
      JJ = K - L                                                        LEM05400
      IF (ABS(G(JJ+1)).GE.ABS(G(JJ))) GO TO 170                         LEM05410
      EE = U(JJ)                                                        LEM05420
      U(JJ) = U(JJ+1)                                                   LEM05430
      U(JJ+1) = EE                                                      LEM05440
      GG = G(JJ)                                                        LEM05450
      G(JJ) = G(JJ+1)                                                   LEM05460
      G(JJ+1) = GG                                                      LEM05470
      IEE = MP(JJ)                                                      LEM05480
      MP(JJ) = MP(JJ+1)                                                 LEM05490
  160 MP(JJ+1) = IEE                                                    LEM05500
C                                                                       LEM05510
  170 CONTINUE                                                          LEM05520
C                                                                       LEM05530
      WRITE(9,180)                                                      LEM05540
  180 FORMAT(5X,'K',6X,'A-MINGAP',5X,'TRUE A-EIGENVALUE(I)',2X,'A-EVNO')LEM05550
C                                                                       LEM05560
      WRITE(9,190)(J,G(J),U(J),MP(J), J=1,NEXACT)                       LEM05570
  190 FORMAT(I6,E14.3,E25.16,I8)                                        LEM05580
C                                                                       LEM05590
      WRITE(9,200)                                                      LEM05600
  200 FORMAT(' NEXACT DISTINCT A-EIGENVALUES. GAPS IN ASCENDING ORDER'/ LEM05610
     2 ' A-MINGAP(I) = TRUE MINIMUM GAP IN A FOR EACH EIGENVALUE.'/     LEM05620
     3 ' A-MINGAP(I) LT 0 INDICATES THE LEFT-HAND GAP WAS MINIMAL.'/    LEM05630
     3 ' A-MATRIX IS BLOCK TRIDIAGONAL AND EACH DIAGONAL BLOCK IS OF ORDLEM05640
     3ER NX.'/                                                          LEM05650
     4 ' NX = NUMBER OF POINTS ON EACH X-LINE. THERE ARE NY DIAGONAL BLOLEM05660
     4CKS.'/                                                            LEM05670
     5 ' NY = NUMBER OF POINTS ON EACH Y-LINE.'/                        LEM05680
     5 ' A-DIAGONAL   = A(K,K)'/                                        LEM05690
     6 ' X-CODIAGONAL = A(I,I+1)'/                                      LEM05700
     7 ' Y-CODIAGONAL = A(I,I+NX)'/                                     LEM05710
     8 ' ----- END OF FILE 9 TRUEEV----------------------------'//)    LEM05720
C                                                                       LEM05730
    4 RETURN                                                            LEM05740
C                                                                       LEM05750
C-----END OF EXEVG-------------------------------------------------------LEM05760
      END                                                               LEM05770
C                                                                       LEM05780
C-----START OF EXERR-----------------------------------------------------LEM05790
C                                                                       LEM05800
C     FOR GIVEN COMPUTED EIGENVALUES, V(I), I=1,2,...,NG                LEM05810
C     COMPUTES THE CLOSEST TRUE EIGENVALUES AND THE ERROR IN THE        LEM05820
C     COMPUTED EIGENVALUES, AND STORES THESE RESPECTIVELY               LEM05830
C     IN U(I) AND IN G(MEV+I).  THESE QUANTITIES ARE WRITTEN            LEM05840
C     TO FILE 10.                                                       LEM05850
C                                                                       LEM05860
      SUBROUTINE EXERR(V,U,G,MP,MEV,NG,NEXACT,IWRITE)                   LEM05870
C                                                                       LEM05880
C-----------------------------------------------------------------------LEM05890
      DOUBLE PRECISION U(1),V(1)                                        LEM05900
      DOUBLE PRECISION EV,EE,T0,T1,C0,C1,C2,PIK,PIL                     LEM05910
      DOUBLE PRECISION ATOLN,EPSM,MACHEP,ZERO,ONE,TWO                   LEM05920
      REAL G(1)                                                         LEM05930
      INTEGER MP(1)                                                     LEM05940
```

```
C-----------------------------------------------------------------------
      DATA MACHEP/Z3410000000000000/
      EPSM = 2.0D0*MACHEP
C-----------------------------------------------------------------------
C     ON ENTRY V CONTAINS NG GOOD EIGENVALUES OF T(1,MEV)
C     MP CONTAINS THE MULTIPLICITIES OF THESE EIGENVALUES.
C     U(I) = GAP TO NEAREST DISTINCT TMEV I=1,NG
C     U < 0. MEANS GAP IS DUE TO SPURIOUS EV
C
C     ON EXIT G(MEV+I) = ERROR FOR V(I) < 0. IF MULT EV > 1
C     K = MP(I) MEANS |V(I) - U(K)| = MIN
C     MP < 0 MEANS MORE THAN ONE I USES SAME K
C
C     T0 = C2 - 2*C1*COS(PIL*J)
C     U(KP) = T0 - 2*C0*COS(PIK*I)
C     KP = (J-1)*KX + I
C     C2 = ONE
C     C0 = X-CODIAGONAL = INPUT
C     C1 = Y-CODIAGONAL = HALF - C0
C-----------------------------------------------------------------------
      GO TO 3
      ENTRY EXERRP(C0,C1,C2,KX,KY)
      GO TO 4
C-----------------------------------------------------------------------
    3 N = KX*KY
      ZERO = 0.0D0
      ONE  = 1.0D0
      TWO  = 2.0D0
C
C     SET G(I) = GAP FROM GOOD T(MEV) TO NEAREST DISTINCT TMEV I=1,NG
      DO 10 I = 1,NG
      G(I) = U(I)
   10 CONTINUE
C
C     REGENERATE A-EIGENVALUES
      T0 = DARCOS(-ONE)
      T1 = DFLOAT(KX+1)
      PIK = T0/T1
      T1 = DFLOAT(KY+1)
      PIL = T0/T1
      KP = 0
C
      DO 30 J = 1,KY
      T1 = PIL*DFLOAT(J)
      T0 = C2 - TWO*C1*DCOS(T1)
      DO 20 I = 1,KX
      KP = KP+1
      T1 = PIK*DFLOAT(I)
   20  U(KP) = T0 - TWO*C0*DCOS(T1)
   30 CONTINUE
C
C     ORDER U VECTOR BY INCREASING ALGEBRAIC SIZE
      DO 50 K = 2,N
      KM1 = K-1
      DO 40 L = 1,KM1
```

```
      JJ = K-L                                                          LEM06500
      IF (U(JJ+1).GE.U(JJ)) GO TO 50                                    LEM06510
      T0 = U(JJ)                                                        LEM06520
      U(JJ) = U(JJ+1)                                                   LEM06530
   40 U(JJ+1) = T0                                                      LEM06540
   50 CONTINUE                                                          LEM06550
C                                                                       LEM06560
      ATOLN = DMAX1(DABS(U(1)),DABS(U(N)))*EPSM                         LEM06570
C                                                                       LEM06580
C     DETERMINE MULTIPLICITIES FOR TRUE EIGENVALUES                     LEM06590
      I = 1                                                             LEM06600
      J = 1                                                             LEM06610
      NEXACT = 0                                                        LEM06620
   60 J = J+1                                                           LEM06630
      IF (J.GT.N) GO TO 70                                              LEM06640
      EE = DABS(U(J)-U(I))                                              LEM06650
      IF (EE.GT.ATOLN) GO TO 70                                         LEM06660
      IDEX = IDEX+1                                                     LEM06670
      GO TO 60                                                          LEM06680
   70 NEXACT = NEXACT+1                                                 LEM06690
      U(NEXACT) = U(I)                                                  LEM06700
      I = J                                                             LEM06710
      IF (I.GT.N) GO TO 80                                              LEM06720
      GO TO 60                                                          LEM06730
   80 CONTINUE                                                          LEM06740
C                                                                       LEM06750
C     NEXACT = NUMBER OF DISTINCT A-EIGENVALUES                         LEM06760
C     U CONTAINS TRUE DISTINCT A-EV ORDERED BY INCREASING SIZE          LEM06770
C                                                                       LEM06780
      IF ( IWRITE.EQ.1) WRITE(6,90)MEV,NG,NEXACT                        LEM06790
   90 FORMAT(/3I6,' = MEV, NG, NEXACT, POISZ CASE'/                     LEM06800
     1 ' TRUE ERRORS FOR GOOD EIGENVALUES'/)                            LEM06810
C                                                                       LEM06820
C      WRITE(6,61) (K,U(K), K=1,NEXACT)                                 LEM06830
C  61 FORMAT(4(I5,E15.8))                                               LEM06840
C                                                                       LEM06850
C     CALCULATION OF THE TRUE ERRORS.                                   LEM06860
      KL = 1                                                            LEM06870
      DO 110 ITEV = 1,NG                                                LEM06880
      EV = V(ITEV)                                                      LEM06890
      K = KL                                                            LEM06900
      T1 = DABS(EV - U(KL))                                             LEM06910
C                                                                       LEM06920
      DO 100 KP = KL,NEXACT                                             LEM06930
      T0 = DABS(EV - U(KP))                                             LEM06940
      IF (T0.GE.T1) GO TO 100                                           LEM06950
      K = KP                                                            LEM06960
      T1 = T0                                                           LEM06970
  100 CONTINUE                                                          LEM06980
C                                                                       LEM06990
      IF (K.EQ.KL.AND.ITEV.GT.1) T1 = -T1                               LEM07000
      KL = K                                                            LEM07010
      MP(ITEV) = K                                                      LEM07020
      G(MEV+ITEV) = T1                                                  LEM07030
  110 CONTINUE                                                          LEM07040
```

```
C                                                                       LEM07050
C     TRUE ERRORS HAVE BEEN COMPUTED OUTPUT THEM TO FILE 10             LEM07060
C     FORM HEADER FOR ERREXACT FILE 10                                  LEM07070
      WRITE(10,120)N,KX,KY,C2,C0,C1                                     LEM07080
  120 FORMAT(' POISSONZ TRUE ERROR FOR GOOD EIGENVALUES'/               LEM07090
     1 5X,'N',4X,'NX',4X,'NY'/3I6//                                     LEM07100
     2 5X,'A-DIAGONAL',3X,'X-CODIAGONAL',3X,'Y-CODIAGONAL'/3E15.8//)    LEM07110
C                                                                       LEM07120
      WRITE(10,130)MEV,NG,NEXACT                                        LEM07130
  130 FORMAT(/3I6,' = MEV,NG,NEXACT'/1X,'T-EV NO',1X,'A-EV NO',         LEM07140
     1 10X,'GOOD EIGENVALUE',5X,'TRUEERROR',7X,'TMINGAP')               LEM07150
C                                                                       LEM07160
      WRITE(10,140)(I,MP(I),V(I),G(MEV+I),G(I), I=1,NG)                 LEM07170
  140 FORMAT(2I8,E25.16,2E14.3)                                         LEM07180
C                                                                       LEM07190
      WRITE(10,150)                                                     LEM07200
  150 FORMAT(' ABOVE ARE THE TRUE ERRORS FOR POISSON GOODEV'/           LEM07210
     1 ' IF A-EV NO LT 0 THEN GOODEV HAS MULTIPLICITY GT 1'/            LEM07220
     1 ' IF TRUE ERROR LT 0 THEN MORE THAN ONE GOODEV APPROXIMATES'/    LEM07230
     1 ' THE SAME TRUE POISSON EIGENVALUE'/                             LEM07240
     1 ' IF TMINGAP LT 0 THE MINGAP IS DUE TO SPURIOUS EIGENVALUE'//)   LEM07250
C                                                                       LEM07260
    4 RETURN                                                            LEM07270
C                                                                       LEM07280
C-----END OF EXERR-------------------------------------------------------LEM07290
      END                                                               LEM07300
C                                                                       LEM07310
C-----START OF EXVEC-----------------------------------------------------LEM07320
C                                                                       LEM07330
C     (JVEC = 1): FOR A GIVEN RITZ VECTOR V AND EIGENVALUE X1, COMPUTES LEM07340
C     THE CLOSEST EIGENVALUE Y1 AND CORRESPONDING TRUE EIGENVECTOR U,   LEM07350
C     AND THEN CALCULATES THE NORM OF THE DIFFERENCE BETWEEN            LEM07360
C     V AND U AND THE MAXIMAL DIFFERENCE BETWEEN THE COMPONENTS.        LEM07370
C     THESE QUANTITIES ARE WRITTEN TO FILE 6.                           LEM07380
C                                                                       LEM07390
C     (JVEC = 2):  COMPUTES THE PROJECTION OF EACH                      LEM07400
C     OF THE TRUE EIGENVECTORS ON THE LANCZOS STARTING VECTOR           LEM07410
C     USED BY THE LANCZS SUBROUTINE AND WRITES THEM TO FILE 12.         LEM07420
C                                                                       LEM07430
      SUBROUTINE EXVEC(U,V,X1,Y1,G,MP,IIX,JVEC,ICOUNT)                  LEM07440
C                                                                       LEM07450
C-----------------------------------------------------------------------LEM07460
      DOUBLE PRECISION U(*),V(1)                                        LEM07470
      DOUBLE PRECISION WI(110),WJ(110),WII(110)                         LEM07480
      DOUBLE PRECISION X1,Y1,EV,EE,WS,PIK,PIL,SUM,PROJ,TEMP,S           LEM07490
      DOUBLE PRECISION ATOLN,EPSM,MACHEP,ZERO,HALF,ONE,TWO              LEM07500
      DOUBLE PRECISION C0,C1,C2,T0,T1,T2                                LEM07510
      REAL G(1),GG                                                      LEM07520
      INTEGER MP(1)                                                     LEM07530
      DOUBLE PRECISION FINPRO                                           LEM07540
C-----------------------------------------------------------------------LEM07550
C     THIS PROGRAM CALCULATES THE TRUE EIGENVALUES AND EIGENVECTORS     LEM07560
C     OF THE POISSON MATRIX A OF ORDER  N = KX*KY                       LEM07570
C     A CONSISTS OF KY TRIDIAGONAL BLOCKS OF ORDER KX                   LEM07580
C     KX = X-DIMENSION     KY = Y-DIMENSION.                            LEM07590
```

```
C
C     IIX = SEED FOR RANDOM NUMBER GENERATOR USED TO CALCULATE
C           STARTING LANCZOS VECTOR IN LANCZS
C     V = RANDOM UNIT STARTING VECTOR FOR LANCZS
C     A*U = EV*U   ||U|| = ONE
C
C     C2 = DIAGONAL OF KX BY KX MATRIX
C    -C0 = CO-DIAGONAL OF THE KX BY KX MATRIX.
C    -C1 = Y-CODIAGONAL.
C
C     NOTE THAT THE VECTORS WI,WJ,WII ARE DIMENSIONED INTERNALLY
C     THEY ARE USED JUST TO KEEP FROM REGENERATING INFORMATION.
C     WI,WII = REAL*8 ARRAYS OF DIMENSION AT LEAST KX
C     WJ     = REAL*8 ARRAY  OF DIMENSION AT LEAST KY.
C
C     NOTATION USED IN PROGRAM
C
C     PIK = ARCOS(-1)/(KX+1)    PIL = ARCOS(-1)/(KY+1)
C     WI(I) = PIK*I        WJ(J) = PIL*J
C
C     U(K) IS A-EV ORDERED BY INCREASING SIZE, K = 1,N
C     LATER U IS USED TO STORE THE TRUE EIGENVECTOR
C     T0 = C2 - 2*C1*COS(PIL*J)    EV(I,J) = T0 - 2*C0*COS(PIK*I)
C     I = 1,KX    J = 1,KY    KP = (J-1)*KX + I
C
C     U(KV) = SIN(PIK*I*IK)*SIN(PIL*J*JK)
C     IK = 1,KX    JK = 1,KY    KV = (JK-1)*KX + IK
C     U IS UNSCALED EIGENVECTOR FOR EV(I,J) = Y1
C     WS = 1/||U||: ||U|| = .5*DSQRT(T2*T3)  T2 = KX+1  T3 = KY+1
C
C    JVEC  = (1,2) FLAGS COMPUTATIONS TO BE PERFORMED.
C
C          = (1) MEANS GIVEN X1 FIND Y1 AND KVEC SUCH THAT
C                       Y1 = EV(KVEC) AND |X1-Y1| = MIN
C                ALSO GIVEN UNIT RITZ VECTOR ASSOCIATED WITH X1
C                     CALCULATE UNIT EIGENVECTOR U, A*U = Y1*U
C                     T2 = ||V-U||  T1 = MAX(|V(K)-U(K)|, K= 1,N)
C                     MAX OCCURS FIRST AT K = KK
C
C          = (2) MEANS CALCULATION OF THE PROJECTION OF THE STARTING
C                     LANCZOS VECTOR ON EACH EIGENVECTOR OF A.
C
C-----------------------------------------------------------------------
      DATA MACHEP/Z3410000000000000/
      EPSM = 2.0D0*MACHEP
C----------------------------------------------------------------------
      GO TO 3
      ENTRY EXVECP(C0,C1,C2,KX,KY)
      GO TO 4
C-----------------------------------------------------------------------
C     SPECIFY PARAMETERS
    3 N = KX*KY
      ZERO = 0.0D0
      HALF = 0.5D0
      ONE  = 1.0D0
```

```
      TWO  = 2.0D0
      T0 = DARCOS(-ONE)
      T1 = DFLOAT(KX+1)
      PIK = T0/T1
      T2 = DFLOAT(KY+1)
      PIL = T0/T2
      WS = TWO/DSQRT(T1*T2)
C
C     GENERATE WI WJ VECTORS
      KP = 0
      DO 20 J = 1,KY
      T1 = PIL*DFLOAT(J)
      WJ(J) = T1
      T0 = C2 - TWO*C1*DCOS(T1)
      DO 10 I = 1,KX
      KP = KP+1
      T1 = PIK*DFLOAT(I)
      WI(I) = T1
   10 U(KP) = T0 - TWO*C0*DCOS(T1)
   20 CONTINUE
C     U(KP) = EV(I,J) = C2 - 2*C1*COS(PIL*J) - 2*C0*COS(PIK*I)
C
C     INITIALIZE MP VECTOR
      DO 30 K = 1,N
   30 MP(K) = K
C
C     WE ORDER U VECTOR BY INCREASING SIZE OF THE EVS
      DO 50 K = 2,N
      KM1 = K-1
C
      DO 40 L = 1,KM1
      JJ = K - L
      IF (U(JJ+1).GE.U(JJ)) GO TO 50
      EE = U(JJ)
      U(JJ) = U(JJ+1)
      U(JJ+1) = EE
      IEE = MP(JJ)
      MP(JJ) = MP(JJ+1)
   40 MP(JJ+1) = IEE
C
   50 CONTINUE
C
      ATOLN = DMAX1(DABS(U(1)),DABS(U(N)))*EPSM
C
      IF (ICOUNT.EQ.1) WRITE(6,60) N,KX,KY,JVEC,C2,C0,C1,ATOLN
   60 FORMAT(/' TRUE ERRORS FOR CONVERGED GOODEV'/
     1 4I6,' = N KX KY JVEC'//
     1 4E12.5,' = C2 C0 C1 ATOLN'//)
C
C     KP = MP(K) MEANS EIGENVALUE U(K) CORRESPONDS TO EIGENVECTOR W(KP)
C     COMPUTE TOLERANCE USED IN COMPUTING TRUE MULTIPLICITIES
C
      IF (JVEC.EQ.1) GO TO 180
C
C     JVEC = 2 SO CALCULATE PROJECTIONS AND WRITE IN FILE 12
```

```
C
      WRITE(12,70)
   70 FORMAT(' PROJECTIONS OF LANCZOS STARTING VECTOR ON A-EIGENVECS')
C
      WRITE(12,80)N,KX,KY,IIX,C2,C0,C1,ATOLN
   80 FORMAT(1X,'A-SIZE',2X,'X-DIM',2X,'Y-DIM',6X,'SVSEED'/3I7,I12/
     1 5X,'A-DIAGONAL',3X,'X-CODIAGONAL',3X,'Y-CODIAGONAL',5X,'ATOLN'/
     2 3E15.8,E10.3)
C
      WRITE(12,90)
   90 FORMAT(5X,'PROJECTION',8X,'TRUE A-EIGENVALUE',1X,'EV NO'
     1,2X,'VEC NO')
C
C     GENERATE SAME RANDOM UNIT VECTOR USED IN THE LANCZS RECURSIONS.
      IIL=IIX
C
C-----------------------------------------------------------------------
      CALL GENRAN(IIL,G,N)
C-----------------------------------------------------------------------
C
      DO 100 I = 1,N
  100 V(I) = G(I)
C
C-----------------------------------------------------------------------
      SUM = FINPRO(N,V(1),1,V(1),1)
C-----------------------------------------------------------------------
C
      SUM = 1.D0/DSQRT(SUM)
C
      DO 110 I = 1,N
  110 V(I) = V(I)*SUM
C
C     DETERMINE UNIT EIGENVECTOR W ASSOCIATED WITH EACH EV(I,J) = Y1
C     AND CALCULATE THE PROJECTION G(K) OF U ON THE STARTING VECTOR V
C     A*U = EV*U       WS = 1/||WU||: WU = UNSCALED EIGENVECTOR
C
      DO 160 K =1,N
C     DETERMINE I J FROM K: MP(K) = KP = (J-1)*KX+I
      KP = MP(K)
      I = MOD(KP,KX)
      IF (I.EQ.0) I = KX
      T1 = WI(I)
      J = 1 + (KP-1)/KX
      T0 = WJ(J)
      T0 = WJ(J)
C
      Y1 = C2 - TWO*C1*DCOS(WJ(J)) - TWO*C0*DCOS(WI(I))
C     Y1 = EV(I,J)
C
      DO 120 II = 1,KX
      T2 = T1*DFLOAT(II)
  120 WII(II) = WS*DSIN(T2)
C
      KV = 0
      DO 140 JJ = 1,KY
```

```
      T2 = T0*DFLOAT(JJ)                                                LEM09250
      T2 = DSIN(T2)                                                     LEM09260
C                                                                       LEM09270
      DO 130 II = 1,KX                                                  LEM09280
      KV = KV + 1                                                       LEM09290
  130 U(KV) = T2*WII(II)                                                LEM09300
C                                                                       LEM09310
  140 CONTINUE                                                          LEM09320
C                                                                       LEM09330
C     U IS UNIT EIGENVECTOR OF A ASSOCIATED WITH EV(I,J) = Y1           LEM09340
C     G(K) IS THE PROJECTION OF U ON V  FOR Y1                          LEM09350
C                                                                       LEM09360
C-----------------------------------------------------------------------LEM09370
      PROJ = FINPRO(N,U(1),1,V(1),1)                                    LEM09380
C-----------------------------------------------------------------------LEM09390
C                                                                       LEM09400
      TEMP = DABS(PROJ)                                                 LEM09410
      G(K) = TEMP                                                       LEM09420
C                                                                       LEM09430
C     DESIRED PROJECTION HAS BEEN COMPUTED OUTPUT IT TO FILE 12.        LEM09440
      WRITE(12,150) G(K),Y1,K,MP(K)                                     LEM09450
  150 FORMAT(E15.8,E25.16,I6,I8)                                        LEM09460
C                                                                       LEM09470
  160 CONTINUE                                                          LEM09480
C                                                                       LEM09490
      WRITE(12,170)                                                     LEM09500
  170 FORMAT(' ----- END OF FILE 12 PROJECT -----------------------'//) LEM09510
C                                                                       LEM09520
      GO TO 310                                                         LEM09530
C                                                                       LEM09540
C     JVEC = 1                                                          LEM09550
C                                                                       LEM09560
C     X1 IS AN INPUT PARAMETER. WE CALCULATE TRUE                       LEM09570
C     A-EIGENVALUE WHICH IS CLOSEST TO X1, LABEL IT Y1 AND CALCULATE    LEM09580
C     UNIT EIGENVECTOR OF A ASSOCIATED WITH Y1. A*U = Y1*U, ||U|| = 1.  LEM09590
C     Y1 = EV(I,J). EIGENVALUES OF A ARE ORDERED BY INCREASING SIZE.    LEM09600
C     V = RITZ VECTOR ASSOCIATED WITH GOODEV X1                         LEM09610
C                                                                       LEM09620
  180 CONTINUE                                                          LEM09630
      KX1 = 0                                                           LEM09640
      IF (X1.LE.U(1)) KX1 = 1                                           LEM09650
      IF (X1.GE.U(N)) KX1 = N                                           LEM09660
      NM1 = N-1                                                         LEM09670
      IF (KX1.NE.0) GO TO 200                                           LEM09680
C                                                                       LEM09690
      DO 190 KVEC = 2,N                                                 LEM09700
      IF (X1.GE.U(KVEC)) GO TO 190                                      LEM09710
C     U(KVEC-1).LE.X1.LT.U(KVEC)                                        LEM09720
      T1 = X1 - U(KVEC-1)                                               LEM09730
      T2 = U(KVEC) - X1                                                 LEM09740
      KX1 = KVEC - 1                                                    LEM09750
      IF (T1.GT.T2) KX1 = KVEC                                          LEM09760
      GO TO 200                                                         LEM09770
  190 CONTINUE                                                          LEM09780
C                                                                       LEM09790
```

```
  200 Y1 = U(KX1)
C
      IF (KX1.EQ.1) EE = U(2) - U(1)
      IF (KX1.EQ.N) EE = U(N) - U(NM1)
      IF (KX1.EQ.1.OR.KX1.EQ.N) GO TO 210
      EE = DMIN1(U(KX1+1)-U(KX1),U(KX1)-U(KX1-1))
  210 CONTINUE
C
      T0 = DABS(ONE - X1/Y1)
C
      WRITE(6,220) N,KX1,ICOUNT,Y1,X1,T0,EE
  220 FORMAT(3I8,' = N, A-EV NUMBER,RITZ NUMBER'//
     1 18X,' TRUEEV',19X,'GOODEV',4X,'RELERROR',4X,'A-MINGAP'/
     1 2E25.16,2E12.3/)
C
      IF (EE.GT.ATOLN) GO TO 240
C
      WRITE(6,230)
  230 FORMAT(' Y1 IS A MULTIPLE EIGENVALUE OF A SO WE EXIT'/)
C
      GO TO 310
C
C     Y1 IS TOEPLITZ EIGENVALUE CLOSEST TO X1.
C     CALCULATION OF EIGENVECTOR ASSOCIATED WITH EIGENVALUE Y1
C     A*U = Y1*U
C     DETERMINE I J FROM K: MP(K) = KP = (J-1)*KX+I
  240 CONTINUE
      K = KX1
      KP = MP(K)
      I = MOD(KP,KX)
      IF (I.EQ.0) I = KX
      T1 = WI(I)
      J = 1 + (KP-1)/KX
      T2 = WJ(J)
C
      DO 250 II = 1,KX
      T0 = T1*DFLOAT(II)
  250 WII(II) = WS*DSIN(T0)
C
      KV = 0
      DO 270 JJ = 1,KY
      T0 = T2*DFLOAT(JJ)
      T0 = DSIN(T0)
C
      DO 260 II = 1,KX
      KV = KV + 1
  260 U(KV) = T0*WII(II)
C
  270 CONTINUE
C
C     U IS UNIT TRUE EIGENVECTOR OF A ASSOCIATED WITH Y1
C     V IS UNIT RITZVECTOR OF A ASSOCIATED WITH X1
C
      KK = 0
      S = ONE
```

```
      T1 = ZERO
C
      DO 280 K = 1,N
      IF (DABS(U(K)).LE.T1) GO TO 280
      T1 = DABS(U(K))
      KK = K
  280 CONTINUE
      IF (U(KK)*V(KK).LT.ZERO) S = - ONE
C
      KK = 0
      T1 = ZERO
      T2 = ZERO
      DO 290 K = 1,N
      TEMP = DABS(S*U(K) - V(K))
      T2 = T2 + TEMP**2
      IF (TEMP.LE.T1) GO TO 290
      KK = K
      T1 = TEMP
  290 CONTINUE
C
      T2 = DSQRT(T2)
      WRITE(6,300) KK,T1,T2
  300 FORMAT(' EIGENVECTOR ERROR. MAX ERROR AT COMPONENT = ',I6/
     1 ' MAX DABS(TRUEVEC(K)-RITZVEC(K)) = ',E12.5/
     1 ' NORM(TRUEVEC-RITZVEC)  = ',E12.5/)
C
  310 CONTINUE
C
    4 RETURN
C
C-----END OF EXVEC-------------------------------------------------------
      END
```

2.6 LESUB: Other Subroutines used by the Codes in Chapters 2, 3, 4, 5

```
C-----------------------------------------------------------------
C
C     LESUB
C
C----------------------------------------------------------------------
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            (1) REAL SYMMETRIC
C            (2) HERMITIAN MATRICES
C            (3) FACTORED INVERSES OF REAL SYMMETRIC MATRICES AND
C            (4) REAL SYMMETRIC GENERALIZED, A*X = EVAL*B*X WHERE
C                B IS POSITIVE DEFINITE, CHOLESKY FACTOR AVAILABLE
C
C     ACCORDING TO PFORT THESE SUBROUTINES ARE PORTABLE EXCEPT FOR:
C     (1) THE COMPLEX*16 VARIABLES AND THE CORRESPONDING FUNCTIONS
C         FOR COMPLEX VARIABLES, DCMPLX, DREAL AND DCONJG USED IN
C         THE SUBROUTINE CINPRD (USED ONLY IN CASE (2), HERMITIAN)
C     (2) THE ENTRY IN THE SUBROUTINE LPERM USED TO PASS THE
C         PERMUTATION FROM THE UPSEC SUBROUTINE TO LPERM. (USED
C         ONLY IN CASES (3) AND (4), INVERSE AND GENERALIZED).
C
C     SUBROUTINES    BISEC, INVERR, TNORM, LUMP, ISOEV, PRTEST, AND
C                    INVERM ARE USED WITH THE LANCZOS EIGENVALUE
C                    PROGRAMS LEVAL, HLEVAL, LIVAL AND LGVAL. STURMI,
C                    INVERM, LBISEC, AND TNORM ARE USED WITH THE
C                    EIGENVECTOR PROGRAMS LEVEC, HLEVEC, LIVEC AND
C                    LGVEC.  LPERM IS USED WITH LIVEC AND LGVEC.
```

```
C                  IN THE HERMITIAN CASE, THE SUBROUTINE CINPRD
C                  IS ALSO USED.
C
C-----COMPUTE T-EIGENVALUES BY BISECTION--------------------------------
C
      SUBROUTINE BISEC(ALPHA,BETA,BETA2,VB,VS,LBD,UBD,EPS,TTOL,MP,
     1 NINT,MEV,NDIS,IC,IWRITE)
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION  ALPHA(1),BETA(1),BETA2(1),VB(1),VS(1)
      DOUBLE PRECISION  LBD(1),UBD(1),EPS,EPT,EP0,EP1,TEMP,TTOL
      DOUBLE PRECISION  ZERO,ONE,HALF,YU,YV,LB,UB,XL,XU,X1,X0,XS,BETAM
      INTEGER  MP(1),IDEF(100)
      DOUBLE PRECISION  DABS, DSQRT, DMAX1, DMIN1, DFLOAT
C-----------------------------------------------------------------------
C     COMPUTES EIGENVALUES OF T(1,MEV) BY LOOPING INTERNALLY ON THE
C     USER-SPECIFIED INTERVALS,  (LB(J),UB(J)), J = 1,NINT.  INTERVALS
C     ARE TREATED AS OPEN ON THE LEFT AND CLOSED ON THE RIGHT.
C     THE BISEC SUBROUTINE SIMULTANEOUSLY LABELS SPURIOUS T-EIGENVALUES
C     AND DETERMINES THE T-MULTIPLICITIES OF EACH GOOD T-EIGENVALUE.
C     SPURIOUS T-EIGENVALUES ARE LABELLED BY A T-MULTIPLICITY = 0.
C     ANY T-EIGENVALUE WITH A T-MULTIPLICITY >= 1 IS 'GOOD'.
C
C     IF IWRITE = 0 THEN MOST OF THE WRITES TO FILE 6 ARE NOT
C     ACTIVATED.
C
C     NOTE THAT PROGRAM ASSUMES THAT NO MORE THAN MMAX/2 EIGENVALUES
C     OF T(1,MEV) ARE TO BE COMPUTED IN ANY ONE OF THE SUBINTERVALS
C     CONSIDERED, WHERE MMAX = DIMENSION OF VB SPECIFIED BY THE USER
C     IN THE MAIN PROGRAM LEVAL.
C
C     ON ENTRY
C     BETA2(J) IS SET = BETA(J)*BETA(J).  THE STORAGE FOR BETA2 COULD
C     BE ELIMINATED BY RECOMPUTING THE BETA(J)**2 FOR EACH STURM
C     SEQUENCE.
C
C     EPS = 2*MACHEP =  4.4 * 10**-16 ON IBM 3081.
C     TTOL = EPS*TKMAX WHERE
C     TKMAX = MAX(|ALPHA(K)|,BETA(K), K=1,KMAX)
C
C     ON EXIT
C     NDIS = TOTAL NUMBER OF COMPUTED DISTINCT EIGENVALUES OF
C            T(1,MEV) ON THE UNION OF THE (LB,UB) INTERVALS.
C     VS = COMPUTED DISTINCT EIGENVALUES OF T(1,MEV) IN ALGEBRAICALLY-
C          INCREASING ORDER
C     MP = CORRESPONDING T-MULTIPLICITIES OF THESE EIGENVALUES
C     MP(I) = (0,1,MI), MI>1, I=1,NDIS  MEANS:
C        (0)  V(I) IS SPURIOUS
C        (1)  V(I) IS ISOLATED AND GOOD
C        (MI) V(I) IS MULTIPLE AND HENCE A CONVERGED GOOD T-EIGENVALUE.
C     IC = TOTAL NUMBER OF STURMS USED
C
C     DEFAULTS
C     ISKIP = 0 INITIALLY. IF DEFAULT OCCURS ON J-TH SUB-INTERVAL, SET
C             ISKIP=ISKIP+1 AND IDEF(ISKIP) = J
```

```
C             DEFAULTS OCCUR IF THERE ARE NO T-EIGENVALUES  IN THE
C             SUBINTERVAL SPECIFIED OR IF THE NUMBER
C             OF STURMS SEQUENCES REQUIRED EXCEEDS MXSTUR.
C             WHEN A DEFAULT OCCURS THE PROGRAM
C             SKIPS THE INTERVAL INVOLVED AND GOES ON TO THE NEXT
C             INTERVAL.
C
C-----------------------------------------------------------------------
C     SPECIFY PARAMETERS
      ZERO = 0.0D0
      ONE  = 1.0D0
      HALF = 0.5D0
      MXSTUR = IC
      NDIS = 0
      IC = 0
      ISKIP = 0
      MP1 = MEV+1
C     SAVE THEN SET BETA(MEV+1) = 0. GENERATE BETA**2
      BETAM = BETA(MP1)
      BETA(MP1) = ZERO
C
      DO 10 I = 1,MP1
   10 BETA2(I) = BETA(I)*BETA(I)
C
C     EP0 IS USED IN T-MULTIPLICITY AND SPURIOUS TESTS
C     EP1 AND EPS ARE USED IN THE BISEC CONVERGENCE TEST
C
      TEMP = DFLOAT(MEV+1000)
      EP0  = TEMP*TTOL
      EP1  = DSQRT(TEMP)*TTOL
C
      WRITE(6,20)MEV,NINT
   20 FORMAT(/' BISEC CALCULATION'/' ORDER OF T IS',I6/
     1' NUMBER OF INTERVALS IS',I6/)
C
      WRITE(6,30) EP0,EP1
   30 FORMAT(/' MULTOL, TOLERANCE USED IN T-MULTIPLICITY AND SPURIOUS TE
     1STS = ',E10.3/' BISTOL, TOLERANCE USED IN BISEC CONVERGENCE TEST =
     1 ',E10.3/)
C
C     LOOP ON THE NINT INTERVALS  (LB(J),UB(J)), J=1,NINT
      DO 430 JIND = 1,NINT
      LB = LBD(JIND)
      UB = UBD(JIND)
C
      WRITE(6,40)JIND,LB,UB
   40 FORMAT(//1X,'BISEC INTERVAL NO',2X,'LOWER BOUND',2X,'UPPER BOUND'/
     1I18,2E13.5/)
C
C     INITIALIZATION AND PARAMETER SPECIFICATION
C     ICT IS TOTAL STURM COUNT ON (LB,UB)
C
      NA = 0
      MD = 0
      NG = 0
```

```
      ICT = 0                                                           LES01540
C                                                                       LES01550
C     START OF T-EIGENVALUE CALCULATIONS                                LES01560
      X1 = UB                                                           LES01570
      ISTURM = 1                                                        LES01580
      GO TO 330                                                         LES01590
C     FORWARD STURM CALCULATION TO DETERMINE NA = NO. T-EIGENVALUES > UBLES01600
   50 NA = NEV                                                          LES01610
C                                                                       LES01620
      X1 = LB                                                           LES01630
      ISTURM = 2                                                        LES01640
      GO TO 330                                                         LES01650
C     FORWARD STURM CALC TO DETERMINE MT = NO. T-EIGENVALUES ON (LB,UB) LES01660
   60 CONTINUE                                                          LES01670
      MT=NEV                                                            LES01680
      ICT = ICT +2                                                      LES01690
C                                                                       LES01700
      WRITE(6,70)MT,NA                                                  LES01710
   70 FORMAT(/2I6,' = NO. TMEV ON (LB,UB) AND NO. .GT. UB'/)            LES01720
C                                                                       LES01730
C     DEFAULT TEST: IS ESTIMATED NUMBER OF STURMS > MXSTUR?             LES01740
      IEST = 30*MT                                                      LES01750
      IF (IEST.LT.MXSTUR) GO TO 90                                      LES01760
C                                                                       LES01770
      WRITE(6,80)                                                       LES01780
   80 FORMAT(//' ESTIMATED NUMBER OF STURMS REQUIRED EXCEEDS USER LIMIT'LES01790
     1/' SKIP THIS SUBINTERVAL')                                        LES01800
      GO TO 110                                                         LES01810
C                                                                       LES01820
   90 CONTINUE                                                          LES01830
C                                                                       LES01840
      IF (MT.GE.1) GO TO 120                                            LES01850
C                                                                       LES01860
      WRITE(6,100)                                                      LES01870
  100 FORMAT(//' THERE ARE NO T-EIGENVALUES ON THIS INTERVAL)'/)        LES01880
C                                                                       LES01890
  110 ISKIP = ISKIP+1                                                   LES01900
      IDEF(ISKIP) = JIND                                                LES01910
      GO TO 430                                                         LES01920
C                                                                       LES01930
C     REGULAR CASE.                                                     LES01940
  120 CONTINUE                                                          LES01950
C                                                                       LES01960
      IF (IWRITE.NE.0) WRITE(6,130)                                     LES01970
  130 FORMAT(/' DISTINCT T-EIGENVALUES COMPUTED USING BISEC'/           LES01980
     1 13X,'T-EIGENVALUE',2X,'TMULT',3X,'MD',4X,'NG')                   LES01990
C                                                                       LES02000
C     SET UP INITIAL UPPER AND LOWER BOUNDS FOR T-EIGENVALUES           LES02010
      DO 140 I=1,MT                                                     LES02020
      VB(I) = LB                                                        LES02030
      MTI = MT + I                                                      LES02040
  140 VB(MTI) = UB                                                      LES02050
C                                                                       LES02060
C     CALCULATE T-EIGENVALUES FROM LB UP TO UB  K = MT,...,1            LES02070
C     MAIN LOOP FOR FINDING KTH T-EIGENVALUE                            LES02080
```

```
C
      K = MT
  150 CONTINUE
      IC0 = 0
      XL = VB(K)
      MTK = MT+K
      XU = VB(MTK)
C
      ISTURM = 3
      X1 = XU
      IC0 = IC0 + 1
      GO TO 330
C     FORWARD STURM CALCULATION AT XU
  160 NU=NEV
C
C     BISECTION LOOP FOR KTH T-EIGENVALUE. TEST  X1=MIDPOINT OF (XL,XU)
      ISTURM = 4
  170 CONTINUE
      X1 = (XL+XU)*HALF
      XS = DABS(XL)+DABS(XU)
      X0 = XU-XL
      EPT = EPS*XS+EP1
C
C     EPT IS CONVERGENCE TOLERANCE FOR KTH T-EIGENVALUE
C
      IF (X0.LE.EPT) GO TO 230
C
C     T-EIGENVALUE HAS NOT YET CONVERGED
C
      IC0 = IC0 + 1
      GO TO 330
C     FORWARD STURM CALCULATION AT CURRENT T-EIGENVALUE APPROXIMATION.
  180 CONTINUE
C
C     UPDATE T-EIGENVALUE INTERVAL (XL,XU)
C
      IF (NEV.LT.K) GO TO 190
C
C     NUMBER OF T-EIGENVALUES NEV = K
      XL = X1
      GO TO 170
  190 CONTINUE
C     NUMBER OF T-EIGENVALUES NEV<K
      XU = X1
      NU = NEV
C
C     UPDATE OF T-EIGENVALUE BOUNDS
C
      IF (NEV.EQ.0) GO TO 210
C
      DO 200 I = 1,NEV
  200 VB(I) = DMAX1(X1,VB(I))
C
  210 NEV1 = NEV+1
C
```

```
      DO 220 II = NEV1,K
      I = MT+II
  220 VB(I) = DMIN1(X1,VB(I))
C
      GO TO 170
C
C     END (XL,XU) BISECTION LOOP FOR KTH T-EIGENVALUE ON (LB,UB)
C     TEST FOR T-MULTIPLICITY AND IF SIMPLE THEN TEST FOR SPURIOUSNESS
C
  230 CONTINUE
      NDIS = NDIS+1
      MD = MD+1
      VS(NDIS) = X1
C
      JSTURM = 1
      X1 = XL-EP0
      GO TO 370
C     BACKWARD STURM CALCULATION
  240 KL = KEV
      JL = JEV
C
      JSTURM = 2
      IC0 = IC0 + 2
      X1 = XU+EP0
      GO TO 370
C     BACKWARD STURM CALCULATION
  250 JU = JEV
      KU = KEV
C
C     FOR T(1,MEV)
C     NU - KU = NO. T-EIGENVALUES ON (XU, XU + EP0)
C     KL - KU = NO. T-EIGENVALUES ON (XL - EP0, XU + EP0)
C
C     FOR T(2,MEV)
C     JL -JU = NO. T-EIGENVALUES ON (XL - EP0, XU + EP0)
C
C     IS THIS A SIMPLE T-EIGENVALUE?
C
      IF (KL-KU-1.EQ.0) GO TO 290
C
C     VS(NDIS) = KTH-T-EIGENVALUE OF (LB,UB) IS MULTIPLE AND HENCE GOOD
      IF (KU.EQ.NU) GO TO 280
C     CONTINUE TO CHECK FOR T-MULTIPLICITY
  260 CONTINUE
      ISTURM = 5
      X1 = X1+EP0
      IC0 = IC0 + 1
      GO TO 330
C     FORWARD STURM CALCULATION
  270 KNE = KU-NEV
      KU = NEV
      IF (KNE.NE.0) GO TO 260
C     SPECIFY T-MULTIPLICITY = MP(NDIS)
  280 MPEV = KL-KU
      KNEW = KU
```

```
      GO TO 300
C     END MULTIPLE CASE
C
C     T-EIGENVALUE IS SIMPLE   CHECK IF IT IS SPURIOUS
  290 CONTINUE
      MPEV = 1
      IF (JU.LT.JL) MPEV=0
      KNEW = K-1
C
C     X1 >= XU+EP0
C     SPURIOUS TEST AND T-SIMPLE CASE COMPLETED
C     START OF NEXT T-EIGENVALUE COMPUTATION
C
  300 K = KNEW
      MP(NDIS) = MPEV
      IF (MPEV.GE.1) NG = NG + 1
C
      IF (IWRITE.NE.0) WRITE(6,310) VS(NDIS),MPEV,MD,NG
  310 FORMAT(E25.16,3I6)
C
C     UPDATE STURM COUNT. IC0 = STURM COUNT FOR KTH T-EIGENVALUE
      ICT = ICT + IC0
C
C     EXIT TEST FOR K DO LOOP
C
      IF (K.LE.0) GO TO 410
C
C     UPDATE LOWER BOUNDS
      DO 320 I=1,KNEW
  320 VB(I) = DMAX1(X1,VB(I))
C
      GO TO 150
C     END OF BISECTION LOOP FOR KTH T-EIGENVALUE
C
C     FORWARD STURM CALCULATION
  330 NEV = -NA
      YU  = ONE
C
      DO 360 I = 1,MEV
      IF (YU.NE.ZERO) GO TO 340
      YV = BETA(I)/EPS
      GO TO 350
  340 YV = BETA2(I)/YU
  350 YU = X1 - ALPHA(I) - YV
      IF (YU.GE.ZERO) GO TO 360
      NEV = NEV + 1
  360 CONTINUE
C     NEV = NUMBER OF T-EIGENVALUES ON (X1,UB)
C
      GO TO (50,60,160,180,270), ISTURM
C
C     BACKWARD STURM CALCULATION FOR T(1,MEV) AND T(2,MEV)
  370 KEV = -NA
      YU  = ONE
C
```

```
      DO 400 II = 1,MEV
      I = MP1-II
      IF (YU.NE.ZERO) GO TO 380
      YV = BETA(I+1)/EPS
      GO TO 390
  380 YV = BETA2(I+1)/YU
  390 YU = X1-ALPHA(I)-YV
      JEV = 0
      IF (YU.GE.ZERO) GO TO 400
      KEV = KEV+1
      JEV = 1
  400 CONTINUE
      JEV = KEV-JEV
C
      GO TO (240,250), JSTURM
C
C     KEV = -NA + (NUMBER OF T(1,MEV) EIGENVALUES) > X1
C     JEV = -NA + (NUMBER OF T(2,MEV) EIGENVALUES) > X1
C     SET PARAMETERS FOR NEXT INTERVAL
  410 CONTINUE
      IC = ICT+IC
      MXSTUR = MXSTUR-ICT
C
      WRITE(6,420) JIND,NG,MD
  420 FORMAT(/' T-EIGENVALUE CALCULATION ON INTERVAL',I6,'  IS COMPLETE'
     1 /3X,'NO. GOOD',3X,'NO. DISTINCT'/I10,I13)
C
  430 CONTINUE
C
C     END LOOP ON THE SUBINTERVALS (LB(J),UB(J)), J=1,NINT
C     ISKIP OUTPUT
C
      IF (ISKIP.GT.0) WRITE(6,440)ISKIP
  440 FORMAT(' BISEC DEFAULTED ON',I3,3X,'INTERVALS'/
     1 ' DEFAULTS OCCUR IF AN INTERVAL HAS NO T-EIGENVALUES'/
     2 ' OR THE STURM ESTIMATE EXCEEDS THE USER-SPECIFIED LIMIT'/)
C
      IF (ISKIP.GT.0) WRITE(6,450)(IDEF(I), I=1,ISKIP)
  450 FORMAT(' BISEC DEFAULTED ON INTERVALS'/(10I8))
C
C     RESET BETA AT I = MP1
      BETA(MP1) = BETAM
C-----END OF BISEC------------------------------------------------------
      RETURN
      END
C
C-----INVERSE ITERATION ON T(1,MEV)-------------------------------------
C
      SUBROUTINE INVERR(ALPHA,BETA,V1,V2,VS,EPS,G,MP,MEV,MMB,NDIS,NISO,
     1 N,IKL,IT,IWRITE)
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION ALPHA(1),BETA(*),V1(1),V2(*),VS(*)
      DOUBLE PRECISION X1,U,Z,EST,TEMP,T0,T1,RATIO,SUM,XU,NORM,TSUM
      DOUBLE PRECISION  BETAM,EPS,EPS3,EPS4,ZERO,ONE
```

```
      REAL G(1)                                                         LES04290
      INTEGER  MP(1)                                                    LES04300
      DOUBLE PRECISION FINPRO                                           LES04310
      REAL  ABS                                                         LES04320
      DOUBLE PRECISION  DABS, DMIN1, DSQRT, DFLOAT                      LES04330
C-----------------------------------------------------------------------LES04340
C     COMPUTES ERROR ESTIMATES FOR COMPUTED ISOLATED GOOD T-EIGENVALUES LES04350
C     IN VS AND WRITES THESE T-EIGENVALUES AND ESTIMATES TO FILE 4.     LES04360
C     BY DEFINITION A GOOD T-EIGENVALUE IS ISOLATED IF ITS              LES04370
C     CLOSEST T-NEIGHBOR IS ALSO GOOD, OR ITS CLOSEST NEIGHBOR IS       LES04380
C     SPURIOUS, BUT THAT NEIGHBOR IS FAR ENOUGH AWAY.  SO               LES04390
C     IN PARTICULAR, WE COMPUTE ESTIMATES FOR GOOD T-EIGENVALUES        LES04400
C     THAT ARE IN CLUSTERS OF GOOD T-EIGENVALUES.                       LES04410
C                                                                       LES04420
C     USES INVERSE ITERATION ON T(1,MEV) SOLVING THE EQUATION           LES04430
C     (T - X1*I)V2 = RIGHT-HAND SIDE (RANDOMLY-GENERATED)               LES04440
C     FOR EACH SUCH GOOD T-EIGENVALUE X1.                               LES04450
C                                                                       LES04460
C     PROGRAM REFACTORS T-X1*I ON EACH ITERATION OF INVERSE ITERATION.  LES04470
C     TYPICALLY ONLY ONE ITERATION IS NEEDED PER EIGENVALUE X1.         LES04480
C                                                                       LES04490
C     POSSIBLE STORAGE COMPRESSION                                      LES04500
C     G STORAGE COULD BE ELIMINATED BY REGENERATING THE RANDOM          LES04510
C     RIGHT-HAND SIDE ON EACH ITERATION AND PRINTING OUT THE            LES04520
C     ERROR ESTIMATES AS THEY ARE GENERATED.                            LES04530
C                                                                       LES04540
C     ON ENTRY AND EXIT                                                 LES04550
C     MEV = ORDER OF T                                                  LES04560
C     ALPHA, BETA CONTAIN THE NONZERO ENTRIES OF THE T-MATRIX           LES04570
C     VS = COMPUTED DISTINCT EIGENVALUES OF T(1,MEV)                    LES04580
C     MP = T-MULTIPLICITY OF EACH T-EIGENVALUE IN VS. MP(I) = -1 MEANS  LES04590
C          VS(I) IS A GOOD T-EIGENVALUE BUT THAT IT IS SITTING CLOSE TO LES04600
C          A SPURIOUS T-EIGENVALUE.  MP(I) = 0 MEANS VS(I) IS SPURIOUS. LES04610
C          ESTIMATES ARE COMPUTED ONLY FOR THOSE T-EIGENVALUES          LES04620
C          WITH MP(I) = 1. FLAGGING WAS DONE IN SUBROUTINE ISOEV        LES04630
C          PRIOR TO ENTERING INVERR.                                    LES04640
C     NISO = NUMBER OF ISOLATED GOOD T-EIGENVALUES CONTAINED IN VS      LES04650
C     NDIS =  NUMBER OF DISTINCT T-EIGENVALUES IN VS                    LES04660
C     IKL = SEED FOR RANDOM NUMBER GENERATOR                            LES04670
C     EPS = 2. * MACHINE EPSILON                                        LES04680
C                                                                       LES04690
C     IN PROGRAM:                                                       LES04700
C     ITER = MAXIMUM NUMBER OF INVERSE ITERATION STEPS ALLOWED FOR EACH LES04710
C            X1.  ITER = IT ON ENTRY.                                   LES04720
C     G = ARRAY OF DIMENSION AT LEAST MEV + NISO.  USED TO STORE        LES04730
C         RANDOMLY-GENERATED RIGHT-HAND SIDE.  THIS IS NOT              LES04740
C         REGENERATED FOR EACH X1. G IS ALSO USED TO STORE ERROR        LES04750
C         ESTIMATES AS THEY ARE COMPUTED FOR LATER PRINTOUT.            LES04760
C     V1,V2 = WORK SPACES USED IN THE FACTORIZATION OF T(1,MEV).        LES04770
C     AT THE END OF THE INVERSE ITERATION COMPUTATION FOR X1, V2        LES04780
C     CONTAINS THE UNIT EIGENVECTOR OF T(1,MEV) CORRESPONDING TO X1.    LES04790
C     V1 AND V2 MUST BE OF DIMENSION AT LEAST MEV.                      LES04800
C                                                                       LES04810
C     ON EXIT                                                           LES04820
C     G(J) = MINIMUM GAP IN T(1,MEV) FOR EACH VS(J), J=1,NDIS           LES04830
```

```
C     G(MEV+I) = BETAM*|V2(MEV)| = ERROR ESTIMATE FOR ISOLATED GOOD     LES04840
C              T-EIGENVALUES, WHERE I = 1,NISO  AND  BETAM = BETA(MEV+1)LES04850
C              V2(MEV) IS LAST COMPONENT OF THE UNIT EIGENVECTOR OF     LES04860
C              T(1,MEV) CORRESPONDING TO ITH ISOLATED GOOD T-EIGENVALUE.LES04870
C                                                                       LES04880
C     IF FOR SOME X1 IT.GT.ITER THEN THE ERROR ESTIMATE IN G IS MARKED  LES04890
C     WITH A - SIGN.                                                    LES04900
C                                                                       LES04910
C     V2 = ISOLATED GOOD T-EIGENVALUES                                  LES04920
C     V1 = MINIMAL T-GAPS FOR THE T-EIGENVALUES IN V2.                  LES04930
C     THESE ARE CONSTRUCTED FOR WRITE-OUT PURPOSES ONLY AND NOT         LES04940
C     NEEDED ELSEWHERE IN THE PROGRAM.                                  LES04950
C-----------------------------------------------------------------------LES04960
C                                                                       LES04970
C     LABEL OUTPUT FILE  4                                              LES04980
      IF (MMB.EQ.1) WRITE(4,10)                                         LES04990
   10 FORMAT(' INVERSE ITERATION ERROR ESTIMATES'/)                     LES05000
C                                                                       LES05010
C     FILE 6 (TERMINAL) OUTPUT OF ERROR ESTIMATES                       LES05020
      IF (IWRITE.NE.0.AND.NISO.NE.0) WRITE(6,20)                        LES05030
   20 FORMAT(/' INVERSE ITERATION ERROR ESTIMATES'/'  JISO',' JDIST',8X LES05040
     1,'GOOD T-EIGENVALUE',4X,'BETAM*UM',5X,'TMINGAP')                  LES05050
C                                                                       LES05060
C     INITIALIZATION AND PARAMETER SPECIFICATION                        LES05070
      ZERO = 0.0D0                                                      LES05080
      ONE = 1.0D0                                                       LES05090
      NG = 0                                                            LES05100
      NISO = 0                                                          LES05110
      ITER = IT                                                         LES05120
      MP1 = MEV+1                                                       LES05130
      MM1 = MEV-1                                                       LES05140
      BETAM = BETA(MP1)                                                 LES05150
      BETA(MP1) = ZERO                                                  LES05160
C                                                                       LES05170
C     CALCULATE SCALE AND TOLERANCES                                    LES05180
      TSUM = DABS(ALPHA(1))                                             LES05190
      DO 30 I = 2,MEV                                                   LES05200
   30 TSUM = TSUM + DABS(ALPHA(I)) + BETA(I)                            LES05210
C                                                                       LES05220
      EPS3 = EPS*TSUM                                                   LES05230
      EPS4 = DFLOAT(MEV)*EPS3                                           LES05240
C                                                                       LES05250
C     GENERATE SCALED RANDOM RIGHT-HAND SIDE                            LES05260
      ILL = IKL                                                         LES05270
C                                                                       LES05280
C-----------------------------------------------------------------------LES05290
      CALL GENRAN(ILL,G,MEV)                                            LES05300
C-----------------------------------------------------------------------LES05310
C                                                                       LES05320
      GSUM = ZERO                                                       LES05330
      DO 40 I = 1,MEV                                                   LES05340
   40 GSUM = GSUM+ABS(G(I))                                             LES05350
      GSUM = EPS4/GSUM                                                  LES05360
C                                                                       LES05370
      DO 50 I = 1,MEV                                                   LES05380
```

```
   50 G(I) = GSUM*G(I)
C
C     LOOP ON ISOLATED GOOD T-EIGENVALUES IN VS (MP(I) = 1) TO
C     CALCULATE CORRESPONDING UNIT EIGENVECTOR OF T(1,MEV)
C
      DO 180 JEV = 1,NDIS
C
      IF (MP(JEV).EQ.0) GO TO 180
      NG = NG + 1
      IF (MP(JEV).NE.1) GO TO 180
C
      IT = 1
      NISO = NISO + 1
      X1 = VS(JEV)
C
C     INITIALIZE RIGHT HAND SIDE FOR INVERSE ITERATION
      DO 60 I = 1,MEV
   60 V2(I) = G(I)
C
C     TRIANGULAR FACTORIZATION WITH NEAREST NEIGHBOR PIVOT
C     STRATEGY. INTERCHANGES ARE LABELLED BY SETTING BETA < 0.
C
   70 CONTINUE
      U = ALPHA(1)-X1
      Z = BETA(2)
C
      DO 90 I = 2,MEV
      IF (BETA(I).GT.DABS(U)) GO TO 80
C     NO INTERCHANGE
      V1(I-1) = Z/U
      V2(I-1) = V2(I-1)/U
      V2(I) = V2(I)-BETA(I)*V2(I-1)
      RATIO = BETA(I)/U
      U = ALPHA(I)-X1-Z*RATIO
      Z = BETA(I+1)
      GO TO 90
   80 CONTINUE
C     INTERCHANGE CASE
      RATIO = U/BETA(I)
      BETA(I) = -BETA(I)
      V1(I-1) = ALPHA(I)-X1
      U = Z-RATIO*V1(I-1)
      Z = -RATIO*BETA(I+1)
      TEMP = V2(I-1)
      V2(I-1) = V2(I)
      V2(I) = TEMP-RATIO*V2(I)
   90 CONTINUE
      IF (U.EQ.ZERO) U = EPS3
C
C     SMALLNESS TEST AND DEFAULT VALUE FOR LAST COMPONENT
C     PIVOT(I-1) = |BETA(I)| FOR INTERCHANGE CASE
C     (I-1,I+1) ELEMENT IN RIGHT FACTOR = BETA(I+1)
C     END OF FACTORIZATION AND FORWARD SUBSTITUTION
C
C     BACK SUBSTITUTION
```

```
      V2(MEV) = V2(MEV)/U
      DO 110 II = 1,MM1
      I = MEV-II
      IF (BETA(I+1).LT.ZERO) GO TO 100
C     NO INTERCHANGE
      V2(I) = V2(I)-V1(I)*V2(I+1)
      GO TO 110
C     INTERCHANGE CASE
  100 BETA(I+1) = -BETA(I+1)
      V2(I) = (V2(I)-V1(I)*V2(I+1)-BETA(I+2)*V2(I+2))/BETA(I+1)
  110 CONTINUE
C
C     TESTS FOR CONVERGENCE OF INVERSE ITERATION
C     IF SUM |V2| COMPS. LE. 1 AND IT. LE. ITER DO ANOTHER INVIT STEP
C
      NORM = DABS(V2(MEV))
      DO 120 II = 1,MM1
      I = MEV-II
  120 NORM = NORM+DABS(V2(I))
C
      IF (NORM.GE.ONE) GO TO 140
      IT = IT+1
      IF (IT.GT.ITER) GO TO 140
      XU = EPS4/NORM
C
      DO 130 I = 1,MEV
  130 V2(I) = V2(I)*XU
C
      GO TO 70
C     ANOTHER INVERSE ITERATION STEP
C
C     INVERSE ITERATION FINISHED
C     NORMALIZE COMPUTED T-EIGENVECTOR : V2 = V2/||V2||
  140 CONTINUE
      SUM = FINPRO(MEV,V2(1),1,V2(1),1)
      SUM = ONE/DSQRT(SUM)
C
      DO 150 II = 1,MEV
  150 V2(II) = SUM*V2(II)
C
C     SAVE ERROR ESTIMATE FOR LATER OUTPUT
      EST = BETAM*DABS(V2(MEV))
      IF (IT.GT.ITER) EST = -EST
      MEVPNI = MEV + NISO
      G(MEVPNI) = EST
      IF (IWRITE.EQ.0) GO TO 180
C
C     FILE 6 (TERMINAL) OUTPUT OF ERROR ESTIMATES.
      IF (JEV.EQ.1) GAP = VS(2) - VS(1)
      IF (JEV.EQ.MEV) GAP = VS(MEV) - VS(MEV-1)
      IF (JEV.EQ.MEV.OR.JEV.EQ.1) GO TO 160
      TEMP = DMIN1(VS(JEV+1)-VS(JEV),VS(JEV)-VS(JEV-1))
      GAP = TEMP
  160 CONTINUE
C
```

```
      WRITE(6,170) NISO,JEV,X1,EST,GAP
  170 FORMAT(2I6,E25.16,2E12.3)
C
  180 CONTINUE
C
C     END ERROR ESTIMATE LOOP ON ISOLATED GOOD T-EIGENVALUES.
C     GENERATE DISTINCT MINGAPS FOR T(1,MEV).  THIS IS USEFUL AS AN
C     INDICATOR OF THE GOODNESS OF THE INVERSE ITERATION ESTIMATES.
C     TRANSFER ISOLATED GOOD T-EIGENVALUES AND CORRESPONDING TMINGAPS
C     TO V2 AND V1 FOR OUTPUT PURPOSES ONLY.
C
      NM1 = NDIS - 1
      G(NDIS) = VS(NM1)-VS(NDIS)
      G(1) = VS(2)-VS(1)
C
      DO 190 J = 2,NM1
      T0 = VS(J)-VS(J-1)
      T1 = VS(J+1)-VS(J)
      G(J) = T1
      IF (T0.LT.T1) G(J)=-T0
  190 CONTINUE
      ISO = 0
      DO 200 J = 1,NDIS
      IF (MP(J).NE.1) GO TO 200
      ISO = ISO+1
      V1(ISO) = G(J)
      V2(ISO) = VS(J)
  200 CONTINUE
C
      IF(NISO.EQ.0) GO TO 250
C
C     ERROR ESTIMATES ARE WRITTEN TO FILE 4
      WRITE(4,210)MEV,NDIS,NG,NISO,N,IKL,ITER,BETAM
  210 FORMAT(1X,'TSIZE',2X,'NDIS',1X,'NGOOD',2X,'NISO',1X,'ASIZE'/5I6/
     1 4X,'RHSEED',2X,'MXINIT',5X,'BETAM'/I10,I8,E10.3/
     2 2X,'GOODEVNO',8X,'GOOD T-EIGENVALUE',6X,'BETAM*UM',7X,'TMINGAP')
C
      ISPUR = 0
      I = 0
      DO 240 J = 1,NDIS
      IF(MP(J).NE.0) GO TO 220
      ISPUR = ISPUR + 1
      GO TO 240
  220 IF(MP(J).NE.1) GO TO 240
      I = I + 1
      MEVI = MEV + I
      IGOOD = J - ISPUR
      WRITE(4,230) IGOOD,V2(I),G(MEVI),V1(I)
  230 FORMAT(I10,E25.16,2E14.3)
  240 CONTINUE
      GO TO 270
C
  250 WRITE(4,260)
  260 FORMAT(/' THERE ARE NO ISOLATED T-EIGENVALUES SO NO ERROR ESTIMATE
     1S WERE COMPUTED')
```

```
C                                                                       LES07040
C     RESTORE BETA(MEV+1) = BETAM                                       LES07050
  270 BETA(MP1) = BETAM                                                 LES07060
C-----END OF INVERR-----------------------------------------------------LES07070
      RETURN                                                            LES07080
      END                                                               LES07090
C                                                                       LES07100
C-----START OF TNORM----------------------------------------------------LES07110
C                                                                       LES07120
      SUBROUTINE TNORM(ALPHA,BETA,BMIN,TMAX,MEV,IB)                     LES07130
C                                                                       LES07140
C-----------------------------------------------------------------------LES07150
      DOUBLE PRECISION  ALPHA(1),BETA(*)                                LES07160
      DOUBLE PRECISION  TMAX,BMIN,BMAX,BSIZE,BTOL                       LES07170
      DOUBLE PRECISION  DABS, DMAX1                                     LES07180
C-----------------------------------------------------------------------LES07190
C     COMPUTE SCALING FACTOR USED IN THE T-MULTIPLICITY, SPURIOUS AND   LES07200
C     PRTESTS.   CHECK RELATIVE SIZE OF THE BETA(K), K=1,MEV            LES07210
C     AS A TEST ON THE LOCAL ORTHOGONALITY OF THE LANCZOS VECTORS.      LES07220
C                                                                       LES07230
C             TMAX = MAX (|ALPHA(I)|, BETA(I),  I=1,MEV)                LES07240
C             BMIN = MIN (BETA(I) I=2,MEV)                              LES07250
C             BSIZE = BMIN/TMAX                                         LES07260
C             |IB| = INDEX OF MINIMAL(BETA)                             LES07270
C             IB < 0 IF BMIN/TMAX < BTOL                                LES07280
C-----------------------------------------------------------------------LES07290
C     SPECIFY PARAMETERS                                                LES07300
      IB = 2                                                            LES07310
      BTOL = BMIN                                                       LES07320
      BMIN = BETA(2)                                                    LES07330
      BMAX = BETA(2)                                                    LES07340
      TMAX = DABS(ALPHA(1))                                             LES07350
C                                                                       LES07360
      DO 20 I = 2,MEV                                                   LES07370
      IF (BETA(I).GE.BMIN) GO TO 10                                     LES07380
      IB = I                                                            LES07390
      BMIN = BETA(I)                                                    LES07400
   10 TMAX = DMAX1(TMAX,DABS(ALPHA(I)))                                 LES07410
      BMAX = DMAX1(BETA(I),BMAX)                                        LES07420
   20 CONTINUE                                                          LES07430
      TMAX = DMAX1(BMAX,TMAX)                                           LES07440
C                                                                       LES07450
C     TEST OF LOCAL ORTHOGONALITY USING SCALED BETAS                    LES07460
      BSIZE = BMIN/TMAX                                                 LES07470
      IF (BSIZE.GE.BTOL) GO TO 40                                       LES07480
C                                                                       LES07490
C     DEFAULT.  BSIZE IS SMALLER THAN TOLERANCE BTOL SPECIFIED IN MAIN  LES07500
C     PROGRAM.  PROGRAM TERMINATES FOR USER TO DECIDE WHAT TO DO        LES07510
C     BECAUSE LOCAL ORTHOGONALITY OF THE LANCZOS VECTORS COULD BE       LES07520
C     LOST.                                                             LES07530
C                                                                       LES07540
      IB = -IB                                                          LES07550
      WRITE(6,30) MEV                                                   LES07560
   30 FORMAT(/' BETA TEST INDICATES POSSIBLE LOSS OF LOCAL ORTHOGONALITYLES07570
     1OVER 1ST',I6,' LANCZOS VECTORS'/)                                 LES07580
```

```
C                                                                       LES07590
   40 CONTINUE                                                          LES07600
C                                                                       LES07610
      WRITE(6,50) IB                                                    LES07620
   50 FORMAT(/' MINIMUM BETA RATIO OCCURS AT',I6,' TH BETA'/)           LES07630
C                                                                       LES07640
      WRITE(6,60) MEV,BMIN,TMAX,BSIZE                                   LES07650
   60 FORMAT(/1X,'TSIZE',6X,'MIN BETA',5X,'TKMAX',6X,'MIN RATIO'/       LES07660
     1 I6,E14.3,E10.3,E15.3/)                                           LES07670
C                                                                       LES07680
C-----END OF TNORM------------------------------------------------------LES07690
      RETURN                                                            LES07700
      END                                                               LES07710
C                                                                       LES07720
C                                                                       LES07730
C-----START OF LUMP-----------------------------------------------------LES07740
C                                                                       LES07750
      SUBROUTINE LUMP(V1,RELTOL,MULTOL,SCALE2,LINDEX,LOOP)              LES07760
C                                                                       LES07770
C-----------------------------------------------------------------------LES07780
      DOUBLE PRECISION  V1(1),SUM,RELTOL,MULTOL,THOLD,ZERO,SCALE2       LES07790
      INTEGER  LINDEX(1)                                                LES07800
      DOUBLE PRECISION  DABS, DFLOAT, DMAX1                             LES07810
C-----------------------------------------------------------------------LES07820
C     LINDEX(J) = T-MULTIPLICITY OF JTH DISTINCT T-EIGENVALUE           LES07830
C     LOOP = NUMBER OF DISTINCT T-EIGENVALUES                           LES07840
C     LUMP 'COMBINES' COMPUTED 'GOOD' T-EIGENVALUES THAT ARE            LES07850
C     'TOO CLOSE'.                                                      LES07860
C     VALUE OF RELTOL IS 1.D-10.                                        LES07870
C                                                                       LES07880
C     IF IN A SET OF T-EIGENVALUES TO BE COMBINED THERE IS AN EIGENVALUELES07890
C     WITH LINDEX=1, THEN THE VALUE OF THE COMBINED EIGENVALUES IS SET  LES07900
C     EQUAL TO THE VALUE OF THAT EIGENVALUE.  NOTE THAT IF A SPURIOUS   LES07910
C     T-EIGENVALUE IS TO BE 'COMBINED' WITH A GOOD T-EIGENVALUE, THEN   LES07920
C     THIS IS DONE ONLY BY INCREASING THE INDEX, LINDEX, FOR THAT       LES07930
C     T-EIGENVALUE.  NUMERICAL VALUES OF SPURIOUS EIGENVALUES ARE NEVER LES07940
C     COMBINE WITH THOSE OF GOOD T-EIGENVALUES.                         LES07950
C-----------------------------------------------------------------------LES07960
      ZERO = 0.0D0                                                      LES07970
      NLOOP = 0                                                         LES07980
      J = 0                                                             LES07990
      ICOUNT = 1                                                        LES08000
      JI = 1                                                            LES08010
      THOLD = DMAX1(RELTOL*DABS(V1(1)),SCALE2*MULTOL)                   LES08020
C     THOLD = DMAX1(RELTOL*DABS(V1(1)),RELTOL)                          LES08030
C                                                                       LES08040
   10 J = J+1                                                           LES08050
      IF (J.EQ.LOOP) GO TO 20                                           LES08060
      SUM = DABS(V1(J)-V1(J+1))                                         LES08070
      IF (SUM.LT.THOLD) GO TO 60                                        LES08080
   20 JF = JI + ICOUNT - 1                                              LES08090
      INDSUM = 0                                                        LES08100
      ISPUR = 0                                                         LES08110
C                                                                       LES08120
      DO 30 KK = JI,JF                                                  LES08130
```

```
      IF (LINDEX(KK).NE.0) GO TO 30
      ISPUR = ISPUR + 1
      INDSUM = INDSUM + 1
   30 INDSUM = INDSUM + LINDEX(KK)
C
C     IF (JF-JI.GE.1) WRITE(6,40) (V1(KKK), KKK=JI,JF)
   40 FORMAT(/' LUMP LUMPS THE T-EIGENVALUES'/(4E20.13))
C
C     COMPUTE THE 'COMBINED' T-EIGENVALUE AND THE RESULTING
C     T-MULTIPLICITY
      K = JI - 1
   50 K = K+1
      IF (K.GT.JF) GO TO 70
      IF (LINDEX(K) .NE.1) GO TO 50
      NLOOP = NLOOP + 1
      V1(NLOOP) = V1(K)
      GO TO 100
   60 ICOUNT = ICOUNT + 1
      GO TO 10
C
C     ALL INDICES WERE 0 OR >1
   70 NLOOP = NLOOP + 1
      IDIF = INDSUM - ISPUR
      IF (IDIF.EQ.0) GO TO 90
C
      SUM = ZERO
      DO 80 KK = JI,JF
   80 SUM = SUM + V1(KK) * DFLOAT(LINDEX(KK))
C
      V1(NLOOP) = SUM/DFLOAT(IDIF)
      GO TO 100
   90 V1(NLOOP) = V1(JI)
  100 LINDEX(NLOOP) = INDSUM
      IDIF = INDSUM - ISPUR
      IF (IDIF.EQ.0.AND.ISPUR.EQ.1) LINDEX(NLOOP) = 0
      IF (J.EQ.LOOP) GO TO 110
      ICOUNT = 1
      JI= J+1
      THOLD = DMAX1(RELTOL*DABS(V1(JI)),SCALE2*MULTOL)
C     THOLD = DMAX1(RELTOL*DABS(V1(JI)),RELTOL)
      IF (JI.LT.LOOP) GO TO 10
      NLOOP = NLOOP + 1
      V1(NLOOP)= V1(JI)
      LINDEX(NLOOP) = LINDEX(JI)
  110 CONTINUE
C
C     ON RETURN V1 CONTAINS THE DISTINCT T-EIGENVALUES
C     LINDEX CONTAINS THE CORRESPONDING T-MULTIPLICITIES
C
      LOOP = NLOOP
      RETURN
C-----END OF LUMP-----------------------------------------------------
      END
C
C
```

```
C-----START OF ISOEV-------------------------------------------------
C
      SUBROUTINE ISOEV(VS,GAPTOL,MULTOL,SCALE1,G,MP,NDIS,NG,NISO)
C
C-------------------------------------------------------------------
      DOUBLE PRECISION  VS(*),T0,T1,MULTOL,GAPTOL,SCALE1,TEMP
      REAL   G(1),GAP
      INTEGER  MP(1)
      REAL  ABS
      DOUBLE PRECISION  DABS, DMAX1
C-------------------------------------------------------------------
C     GENERATE DISTINCT TMINGAPS AND USE THEM TO LABEL THE ISOLATED
C     GOOD T-EIGENVALUES THAT ARE VERY CLOSE TO SPURIOUS ONES.
C     ERROR ESTIMATES WILL NOT BE COMPUTED FOR THESE T-EIGENVALUES.
C
C     ON ENTRY AND EXIT
C     VS CONTAINS THE COMPUTED DISTINCT EIGENVALUES OF T(1,MEV)
C     MP CONTAINS THE CORRESPONDING T-MULTIPLICITIES
C     NDIS = NUMBER OF DISTINCT EIGENVALUES
C     GAPTOL = RELATIVE GAP TOLERANCE SET IN MAIN
C
C     ON EXIT
C     G CONTAINS THE TMINGAPS.
C     G(I) < 0 MEANS MINGAP IS DUE TO LEFT GAP
C     MP(I) IS NOT CHANGED EXCEPT THAT  MP(I)=-1, IF MP(I)=1,
C     TMINGAP WAS TOO SMALL AND DUE TO A SPURIOUS T-EIGENVALUE.
C
C     IF MP(I)=-1 THAT SIMPLE GOOD T-EIGENVALUE WILL BE SKIPPED
C     IN THE SUBSEQUENT ERROR ESTIMATE COMPUTATIONS IN INVERR
C     THAT IS, WE COMPUTE ERROR ESTIMATES ONLY FOR THOSE GOOD
C     T-EIGENVALUES WITH MP(I)=1.
C-------------------------------------------------------------------
C     CALCULATE MINGAPS FOR DISTINCT T(1,MEV) EIGENVALUES.
      NM1 = NDIS - 1
      G(NDIS) = VS(NM1)-VS(NDIS)
      G(1) = VS(2)-VS(1)
C
      DO 10 J = 2,NM1
      T0 = VS(J)-VS(J-1)
      T1 = VS(J+1)-VS(J)
      G(J) = T1
      IF (T0.LT.T1) G(J) = -T0
   10 CONTINUE
C
C     SET MP(I)=-1 FOR SIMPLE GOOD T-EIGENVALUES WHOSE MINGAPS  ARE
C     'TOO SMALL' AND DUE TO SPURIOUS T-EIGENVALUES.
C
      NISO = 0
      NG = 0
      DO 20 J = 1,NDIS
      IF (MP(J).EQ.0) GO TO 20
      NG = NG+1
      IF (MP(J).NE.1) GO TO 20
C     VS(J) IS NEXT SIMPLE GOOD T-EIGENVALUE
      NISO = NISO + 1
```

```
      I = J+1
      IF (G(J).LT.0.0) I = J-1
      IF (MP(I).NE.0) GO TO 20
      GAP = ABS(G(J))
      T0 = DMAX1(SCALE1*MULTOL,GAPTOL*DABS(VS(J)))
C     T0 = DMAX1(GAPTOL,GAPTOL*DABS(VS(J)))
      TEMP = T0
      IF (GAP.GT.TEMP) GO TO 20
      MP(J) = -MP(J)
      NISO = NISO-1
   20 CONTINUE
C
C-----END OF ISOEV-----------------------------------------------------
      RETURN
      END
C
C-----START OF PRTEST--------------------------------------------------
C
      SUBROUTINE PRTEST(ALPHA,BETA,TEIG,TKMAX,EPSM,RELTOL,SCALE3,SCALE4,
     1 TMULT,NDIST,MEV,IPROJ)
C
C----------------------------------------------------------------------
      DOUBLE PRECISION  ALPHA(1), BETA(1),TEIG(*),SIGMA(10)
      DOUBLE PRECISION  EPSM,RELTOL,PRTOL,TKMAX,LRATIO,URATIO
      DOUBLE PRECISION  EPS,EPS1,BETAM,LBD,UBD,SIG,YU,YV,LRATS,URATS
      DOUBLE PRECISION ZERO,ONE,TEN,BISTOL,SCALE3,SCALE4,AEV,TEMP
      INTEGER  TMULT(*),ISIGMA(10)
      DOUBLE PRECISION  DABS, DMAX1, DSQRT, DFLOAT
C----------------------------------------------------------------------
C     AFTER CONVERGENCE HAS BEEN ESTABLISHED, SUBROUTINE PRTEST
C     TESTS COMPUTED EIGENVALUES OF T(1,MEV) THAT HAVE BEEN LABELLED
C     SPURIOUS TO DETERMINE IF ANY EIGENVALUES OF A HAVE BEEN
C     MISSED BY LANCZOS PROCEDURE.  AN EIGENVALUE WITH A VERY SMALL
C     PROJECTION ON THE STARTING VECTOR (< SINGLE PRECISION)
C     CAN BE MISSED BECAUSE IT IS ALSO AN EIGENVALUE OF T(2,MEV) TO
C     WITHIN THE SQUARE OF THIS ORIGINAL PROJECTION.
C     OUR EXPERIENCE IS THAT SUCH SMALL PROJECTIONS OCCUR ONLY
C     VERY INFREQUENTLY.
C
C     THIS SUBROUTINE IS CALLED ONLY AFTER CONVERGENCE HAS BEEN
C     ESTABLISHED. ONCE CONVERGENCE HAS BEEN OBSERVED ON THE
C     OTHER EIGENVALUES THEN ONE CAN EXPECT TO ALSO HAVE CONVERGENCE
C     ON ANY SUCH HIDDEN EIGENVALUES.(IF THERE ARE ANY).  THIS
C     PROCEDURE CONSIDERS ONLY SPURIOUS T-EIGENVALUES AND ONLY THOSE
C     SPURIOUS T-EIGENVALUES THAT ARE ISOLATED FROM GOOD T-EIGENVALUES.
C     FOR EACH SUCH T-EIGENVALUE IT DOES 2 STURM SEQUENCES
C     AND A FEW SCALAR MULTIPLICATIONS.  UPON RETURN TO MAIN
C     PROGRAM ERROR ESTIMATES WILL BE COMPUTED FOR ANY EIGENVALUES
C     THAT HAVE BEEN LABELLED AS 'HIDDEN'.  SUCH T-EIGENVALUES
C     WILL BE RELABELLED AS 'GOOD' ONLY IF THESE ERROR ESTIMATES
C     ARE SUFFICIENTLY SMALL.
C----------------------------------------------------------------------
      ZERO = 0.0D0
      ONE  = 1.0D0
      TEN  = 10.0D0
```

```
      PRTOL = 1.D-6
      TEMP = DFLOAT(MEV+1000)
      TEMP = DSQRT(TEMP)
      BISTOL = TKMAX*EPSM*TEMP
      NSIGMA = 4
      SIGMA(1) = TEN*TKMAX
C
      DO 10 J = 2,NSIGMA
   10 SIGMA(J) = TEN*SIGMA(J-1)
C
      IFIN = 0
      MF = 1
      ML = MEV
      BETAM = BETA(MF)
      BETA(MF) = ZERO
      IPROJ = 0
      J = 1
C
      IF (TMULT(1).NE.0) GO TO 110
C
      AEV = DABS(TEIG(1))
      TEMP = PRTOL*AEV
      EPS1 = DMAX1(TEMP,SCALE4*BISTOL)
C     EPS1 = DMAX1(TEMP,PRTOL)
      TEMP = RELTOL*AEV
      EPS  = DMAX1(TEMP,SCALE3*BISTOL)
C     EPS  = DMAX1(TEMP,RELTOL)
C
      IF (TEIG(2)-TEIG(1).LT.EPS1.AND.TMULT(2).NE.0) GO TO 110
C
   20 LBD = TEIG(J) - EPS
      UBD = TEIG(J) + EPS
      MEVL = 0
      IL = 0
      YU = ONE
C
      DO 50 I=MF,ML
      IF (YU.NE.ZERO) GO TO 30
      YV = BETA(I)/EPSM
      GO TO 40
   30 YV = BETA(I)*BETA(I)/YU
   40 YU = ALPHA(I)-LBD-YV
      IF (YU.GE.ZERO) GO TO 50
C     MEVL INCREMENTED
      MEVL = MEVL + 1
      IL = I
   50 CONTINUE
C
      LRATIO = YU
      MEV1L = MEVL
      IF (IL.EQ.ML) MEV1L=MEVL-1
C
C     MEVL = NUMBER OF EVS OF T(1,MEV) WHICH ARE < LBD
C     MEV1L = NUMBER OF EVS OF T(1,MEV-1) WHICH ARE < LBD
C     LRATIO = DET(T(1,MEV)-LBD)/DET(T(1,MEV-1)-LBD):
```

```
C
      MEVU = 0
      IL = 0
      YU = ONE
C
      DO 80 I=MF,ML
      IF (YU.NE.ZERO) GO TO 60
      YV = BETA(I)/EPSM
      GO TO 70
   60 YV = BETA(I)*BETA(I)/YU
   70 YU = ALPHA(I)-UBD-YV
      IF (YU.GE.ZERO) GO TO 80
C     MEVU INCREMENTED
      MEVU = MEVU + 1
      IL = I
   80 CONTINUE
C
      URATIO = YU
      MEV1U = MEVU
      IF (IL.EQ.ML) MEV1U=MEVU-1
C
C     MEVU = NUMBER OF EVS OF T(MEV) WHICH ARE < UBD
C     MEV1U = NUMBER OF EVS OF T(MEV-1) WHICH ARE < UBD
C     URATIO = DET(TM-UBD)/DET(T(M-1)-UBD): TM=T(MF,ML)
C
      NEV1 = MEV1U-MEV1L
C
      DO 90 K=1,NSIGMA
      SIG = SIGMA(K)
      LRATS = LRATIO-SIG
      URATS = URATIO-SIG
C     NOTE THE INCREMENT IS ON NUMBER OF EVALUES OF T(M-1)
      MEVLS = MEV1L
      IF (LRATS.LT.0.) MEVLS=MEV1L+1
      MEVUS = MEV1U
      IF (URATS.LT.0.) MEVUS=MEV1U+1
      ISIGMA(K) = MEVUS - MEVLS
   90 CONTINUE
C
      ICOUNT = 0
      DO 100 K=1,NSIGMA
  100 IF (ISIGMA(K).EQ.1) ICOUNT=ICOUNT + 1
C
      IF (ICOUNT.LT.2.OR.NEV1.EQ.0) GO TO 110
      TMULT(J) = -10
      IPROJ=IPROJ+1
C
  110 J=J+1
C
      IF (J.GE.NDIST) GO TO 120
      IF (TMULT(J).NE.0) GO TO 110
C
      AEV = DABS(TEIG(J))
      TEMP = PRTOL*AEV
      EPS1 = DMAX1(TEMP,SCALE4*BISTOL)
```

```
C     EPS1 = DMAX1(TEMP,PRTOL)
      TEMP = RELTOL*AEV
      EPS  = DMAX1(TEMP,SCALE3*BISTOL)
C     EPS  = DMAX1(TEMP,RELTOL)
C
      IF (TEIG(J)-TEIG(J-1).LT.EPS1.AND.TMULT(J-1).NE.0) GO TO 110
      IF (TEIG(J+1)-TEIG(J).LT.EPS1.AND.TMULT(J+1).NE.0) GO TO 110
C
      GO TO 20
C
  120 IF (IFIN.EQ.1) GO TO 130
      IF (TMULT(NDIST).NE.0) GO TO 130
C
      AEV = DABS(TEIG(NDIST))
      TEMP = PRTOL*AEV
      EPS1 = DMAX1(TEMP,SCALE4*BISTOL)
C     EPS1 = DMAX1(TEMP,PRTOL)
      TEMP = RELTOL*AEV
      EPS  = DMAX1(TEMP,SCALE3*BISTOL)
C     EPS  = DMAX1(TEMP,RELTOL)
C
      NDIST1=NDIST -1
      TEMP = TEIG(NDIST)-TEIG(NDIST1)
      IF (TEMP.LT.EPS1.AND.TMULT(NDIST1).NE.0) GO TO 130
      IFIN = 1
C
      GO TO 20
C
  130 BETA(MF) = BETAM
C
C-----END OF PRTEST-----------------------------------------------------
      RETURN
      END
C
C------START OF STURMI--------------------------------------------------
C
      SUBROUTINE STURMI(ALPHA,BETA,X1,TOLN,EPSM,MMAX,MK1,MK2,IC,IWRITE)
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION  ALPHA(1),BETA(1)
      DOUBLE PRECISION  EPSM,X1,TOLN,EVL,EVU,BETA2
      DOUBLE PRECISION  U1,U2,V1,V2,ZERO,ONE
      INTEGER I,IC,ICD,IC0,IC1,IC2,MK1,MK2,MMAX
C-----------------------------------------------------------------------
C
C     FOR ANY EIGENVALUE OF A THAT HAS CONVERGED AS AN EIGENVALUE
C     OF THE T-MATRICES THIS SUBROUTINE CALCULATES
C     THE SMALLEST SIZE OF THE T-MATRIX, T(1,MK1) DEFINED
C     BY THE ALPHA AND BETA ARRAYS SUCH THAT MK1.LE.MMAX
C     AND THE INTERVAL (X1-TOLN,X1+TOLN) CONTAINS AT LEAST ONE
C     EIGENVALUE OF T(1,MK1). IT ALSO CALCULATES MK2 <= MMAX
C     AS THE SMALLEST SIZE T-MATRIX (IF ANY) SUCH THAT THIS INTERVAL
C     CONTAINS AT LEAST TWO EIGENVALUES OF T(1,MK2).
C     IF NO T-MATRIX OF ORDER < MMAX SATISFIES THIS REQUIREMENT
C     THEN MK2 IS SET EQUAL TO MMAX.  THE EIGENVECTOR PROGRAM
```

```
C     USES THESE VALUES TO DETERMINE AN APPROPRIATE 1ST GUESS AT
C     AN APPROPRIATE SIZE T-MATRIX FOR THE EIGENVALUE X1.
C
C     ON EXIT IC = NUMBER OF EIGENVALUES OF T(1,MK2) IN THIS INTERVAL
C
C     STURMI REGENERATES THE QUANTITIES BETA(I)**2 EACH TIME IT IS
C     CALLED, OBVIOUSLY FOR THE PRICE OF ANOTHER VECTOR OF LENGTH
C     MMAX THIS GENERATION COULD BE DONE ONCE IN THE MAIN
C     PROGRAM BEFORE THE LOOP ON THE CALLS TO SUBROUTINE STURMI.
C
C     IF ANY OF THE EIGENVALUES BEING CONSIDERED WERE MULTIPLE
C     AS EIGENVALUES OF THE USER-SPECIFIED MATRIX, THEN
C     THIS SUBROUTINE COULD BE MODIFIED TO COMPUTE ADDITIONAL
C     SIZES MKJ, J = 3, ...  WHICH COULD THEN BE USED IN THE
C     MAIN LANCZOS EIGENVECTOR PROGRAM TO COMPUTE ADDITIONAL
C     EIGENVECTORS CORRESPONDING TO THESE MULTIPLE EIGENVALUES.
C     THE MAIN PROGRAM PROVIDED DOES NOT INCLUDE THIS OPTION.
C
C-----------------------------------------------------------------------
C     INITIALIZATION OF PARAMETERS
      MK1 = 0
      MK2 = 0
      ZERO = 0.0D0
      ONE  = 1.0D0
      BETA(1) = ZERO
      EVL = X1-TOLN
      EVU = X1+TOLN
      U1 = ONE
      U2 = ONE
      IC0 = 0
      IC1 = 0
      IC2 = 0
C
C     MAIN LOOP FOR CALCULATING THE SIZES MK1,MK2
      DO 60 I = 1,MMAX
      BETA2 = BETA(I)*BETA(I)
      IF (U1.NE.ZERO) GO TO 10
      V1 = BETA(I)/EPSM
      GO TO 20
   10 V1 = BETA2/U1
   20 U1 = EVL - ALPHA(I) - V1
      IF (U1.LT.ZERO) IC1 = IC1+1
      IF (U2.NE.ZERO) GO TO 30
      V2 = BETA(I)/EPSM
      GO TO 40
   30 V2 = BETA2/U2
   40 U2 = EVU - ALPHA(I) - V2
      IF (U2.LT.ZERO) IC2 = IC2+1
C     TEST FOR CHANGE IN NUMBER OF T-EIGENVALUES ON (EVL,EVU)
      ICD = IC1-IC2
      IC = ICD-IC0
      IF (IC.GE.1) GO TO 50
      GO TO 60
   50 CONTINUE
      IF (IC0.EQ.0) MK1 = I
```

```
      IC0 = IC0+1
      IF (IC0.GT.1) GO TO 70
   60 CONTINUE
C
      I = I-1
      IF (IC0.EQ.0) MK1 = MMAX
   70 MK2 = I
      IC = ICD
C
      IF (IWRITE.EQ.1) WRITE(6,80) X1,MK1,MK2,IC
   80 FORMAT(' EVAL =',E20.12,' MK1 =',I6,' MK2 =',I6,' IC =',I3/)
C
      RETURN
C-----END OF STURMI----------------------------------------------------
      END
C
C
C-----START OF INVERM--------------------------------------------------
C
      SUBROUTINE INVERM(ALPHA,BETA,V1,V2,X1,ERROR,ERRORV,EPS,G,MEV,IT,
     1 IWRITE)
C
C----------------------------------------------------------------------
      DOUBLE PRECISION  ALPHA(1),BETA(*),V1(1),V2(*)
      DOUBLE PRECISION  X1,U,Z,TEMP,RATIO,SUM,XU,NORM,TSUM,BETAM
      DOUBLE PRECISION  EPS,EPS3,EPS4,ERROR,ERRORV,ZERO,ONE
      REAL  G(1)
      DOUBLE PRECISION  DABS, DSQRT, DFLOAT
      DOUBLE PRECISION FINPRO
      REAL  ABS
C----------------------------------------------------------------------
C
C     COMPUTES T-EIGENVECTORS FOR ISOLATED GOOD T-EIGENVALUES X1
C     USING INVERSE ITERATION ON T(1,MEV(X1)) SOLVING EQUATION
C     (T - X1*I)V2 = RIGHT-HAND SIDE (RANDOMLY-GENERATED) .
C     PROGRAM REFACTORS T- X1*I ON EACH ITERATION OF INVERSE ITERATION.
C     TYPICALLY ONLY ONE ITERATION IS NEEDED PER T-EIGENVALUE X1.
C
C     IF IWRITE = 1 THEN THERE ARE EXTENDED WRITES TO FILE 6 (TERMINAL)
C
C     ON ENTRY G CONTAINS A REAL*4 RANDOM VECTOR WHICH WAS GENERATED
C     IN MAIN PROGRAM.
C
C     ON ENTRY AND EXIT
C     MEV = ORDER OF T
C     ALPHA, BETA CONTAIN THE DIAGONAL AND OFFDIAGONAL ENTRIES OF T.
C     EPS = 2. * MACHINE EPSILON
C
C     IN PROGRAM:
C     ITER = MAXIMUM NUMBER STEPS ALLOWED FOR INVERSE ITERATION
C     ITER = IT ON ENTRY.
C     V1,V2 = WORK SPACES USED IN THE FACTORIZATION OF T(1,MEV).
C     V1 AND V2 MUST BE OF DIMENSION AT LEAST MEV.
C
C     ON EXIT
```

```
C     V2 = THE UNIT EIGENVECTOR OF T(1,MEV) CORRESPONDING TO X1.
C     ERROR = |V2(MEV)| = ERROR ESTIMATE FOR CORRESPONDING
C             RITZ VECTOR FOR X1.
C
C     ERRORV = || T*V2 - X1*V2 || = ERROR ESTIMATE ON T-EIGENVECTOR.
C     IF IT.GT.ITER THEN ERRORV = -ERRORV
C     IT = NUMBER OF ITERATIONS ACTUALLY REQUIRED
C-----------------------------------------------------------------------
C     INITIALIZATION AND PARAMETER SPECIFICATION
      ONE  = 1.0D0
      ZERO = 0.0D0
      ITER = IT
      MP1 = MEV+1
      MM1 = MEV-1
      BETAM = BETA(MP1)
      BETA(MP1) = ZERO
C
C     CALCULATE SCALE AND TOLERANCES
      TSUM = DABS(ALPHA(1))
      DO 10 I = 2,MEV
   10 TSUM = TSUM + DABS(ALPHA(I)) + BETA(I)
C
      EPS3 = EPS*TSUM
      EPS4 = DFLOAT(MEV)*EPS3
C
C     GENERATE SCALED RANDOM RIGHT-HAND SIDE
      GSUM = ZERO
      DO 20 I = 1,MEV
   20 GSUM = GSUM+ABS(G(I))
      GSUM = EPS4/GSUM
C
C     INITIALIZE RIGHT HAND SIDE FOR INVERSE ITERATION
      DO 30 I = 1,MEV
   30 V2(I) = GSUM*G(I)
      IT = 1
C
C     CALCULATE UNIT EIGENVECTOR OF T(1,MEV) FOR ISOLATED GOOD
C     T-EIGENVALUE X1.
C
C     TRIANGULAR FACTORIZATION WITH NEAREST NEIGHBOR PIVOT
C     STRATEGY. INTERCHANGES ARE LABELLED BY SETTING BETA < 0.
C
   40 CONTINUE
      U = ALPHA(1)-X1
      Z = BETA(2)
C
      DO 60 I=2,MEV
      IF (BETA(I).GT.DABS(U)) GO TO 50
C     NO PIVOT INTERCHANGE
      V1(I-1) = Z/U
      V2(I-1) = V2(I-1)/U
      V2(I) = V2(I)-BETA(I)*V2(I-1)
      RATIO = BETA(I)/U
      U = ALPHA(I)-X1-Z*RATIO
      Z = BETA(I+1)
```

```
      GO TO 60                                                          LES13090
C     PIVOT INTERCHANGE                                                 LES13100
   50 CONTINUE                                                          LES13110
      RATIO = U/BETA(I)                                                 LES13120
      BETA(I) = -BETA(I)                                                LES13130
      V1(I-1) = ALPHA(I)-X1                                             LES13140
      U = Z-RATIO*V1(I-1)                                               LES13150
      Z = -RATIO*BETA(I+1)                                              LES13160
      TEMP = V2(I-1)                                                    LES13170
      V2(I-1) = V2(I)                                                   LES13180
      V2(I) = TEMP-RATIO*V2(I)                                          LES13190
   60 CONTINUE                                                          LES13200
C                                                                       LES13210
      IF (U.EQ.ZERO) U=EPS3                                             LES13220
C                                                                       LES13230
C     SMALLNESS TEST AND DEFAULT VALUE FOR LAST COMPONENT               LES13240
C     PIVOT(I-1) = |BETA(I)| FOR INTERCHANGE CASE                       LES13250
C     (I-1,I+1) ELEMENT IN RIGHT FACTOR = BETA(I+1)                     LES13260
C     END OF FACTORIZATION AND FORWARD SUBSTITUTION                     LES13270
C                                                                       LES13280
C     BACK SUBSTITUTION                                                 LES13290
      V2(MEV) = V2(MEV)/U                                               LES13300
      DO 80 II = 1,MM1                                                  LES13310
      I = MEV-II                                                        LES13320
      IF (BETA(I+1).LT.ZERO) GO TO 70                                   LES13330
C     NO PIVOT INTERCHANGE                                              LES13340
      V2(I) = V2(I)-V1(I)*V2(I+1)                                       LES13350
      GO TO 80                                                          LES13360
C     PIVOT INTERCHANGE                                                 LES13370
   70 BETA(I+1) = -BETA(I+1)                                            LES13380
      V2(I) = (V2(I)-V1(I)*V2(I+1)-BETA(I+2)*V2(I+2))/BETA(I+1)         LES13390
   80 CONTINUE                                                          LES13400
C                                                                       LES13410
C                                                                       LES13420
C     TESTS FOR CONVERGENCE OF INVERSE ITERATION                        LES13430
C     IF SUM |V2| COMPS. LE. 1 AND IT. LE. ITER DO ANOTHER INVIT STEP   LES13440
C                                                                       LES13450
      NORM = DABS(V2(MEV))                                              LES13460
      DO 90 II = 1,MM1                                                  LES13470
      I = MEV-II                                                        LES13480
   90 NORM = NORM+DABS(V2(I))                                           LES13490
C                                                                       LES13500
C     IS DESIRED GROWTH IN VECTOR ACHIEVED ?                            LES13510
C     IF NOT, DO ANOTHER INVERSE ITERATION STEP UNLESS NUMBER ALLOWED ISLES13520
C     EXCEEDED.                                                         LES13530
      IF (NORM.GE.ONE) GO TO 110                                        LES13540
C                                                                       LES13550
      IT=IT+1                                                           LES13560
      IF (IT.GT.ITER) GO TO 110                                         LES13570
C                                                                       LES13580
      XU = EPS4/NORM                                                    LES13590
      DO 100 I=1,MEV                                                    LES13600
  100 V2(I) = V2(I)*XU                                                  LES13610
C                                                                       LES13620
      GO TO 40                                                          LES13630
```

```
C                                                                       LES13640
C     NORMALIZE COMPUTED T-EIGENVECTOR : V2 = V2/||V2||                 LES13650
C                                                                       LES13660
  110 CONTINUE                                                          LES13670
C                                                                       LES13680
      SUM = FINPRO(MEV,V2(1),1,V2(1),1)                                 LES13690
      SUM = ONE/DSQRT(SUM)                                              LES13700
      DO 120 II = 1,MEV                                                 LES13710
  120 V2(II) = SUM*V2(II)                                               LES13720
C                                                                       LES13730
C     SAVE ERROR ESTIMATE FOR LATER OUTPUT                              LES13740
      ERROR = DABS(V2(MEV))                                             LES13750
C                                                                       LES13760
C     GENERATE ERRORV = ||T*V2 - X1*V2||.                               LES13770
      V1(MEV) = ALPHA(MEV)*V2(MEV)+BETA(MEV)*V2(MEV-1)-X1*V2(MEV)       LES13780
      DO 130 J = 2,MM1                                                  LES13790
      JM = MP1 - J                                                      LES13800
      V1(JM) = ALPHA(JM)*V2(JM) + BETA(JM)*V2(JM-1) + BETA(JM+1)*V2(JM+1LES13810
     1) - X1*V2(JM)                                                     LES13820
  130 CONTINUE                                                          LES13830
C                                                                       LES13840
      V1(1) = ALPHA(1)*V2(1) + BETA(2)*V2(2) - X1*V2(1)                 LES13850
      ERRORV = FINPRO(MEV,V1(1),1,V1(1),1)                              LES13860
      ERRORV = DSQRT(ERRORV)                                            LES13870
      IF (IT.GT.ITER) ERRORV = -ERRORV                                  LES13880
      IF (IWRITE.EQ.0) GO TO 150                                        LES13890
C                                                                       LES13900
C     FILE 6 (TERMINAL) OUTPUT OF ERROR ESTIMATES.                      LES13910
      WRITE(6,140) MEV,X1,ERROR,ERRORV                                  LES13920
  140 FORMAT(2X,'TSIZE',15X,'EIGENVALUE',11X,'U(M)',9X,'ERRORV'/        LES13930
     1 I6,E25.16,2E15.5)                                                LES13940
C                                                                       LES13950
C     RESTORE BETA(MEV+1) = BETAM                                       LES13960
  150 CONTINUE                                                          LES13970
      BETA(MP1) = BETAM                                                 LES13980
C-----END OF INVERM---------------------------------------------------LES13990
      RETURN                                                            LES14000
      END                                                               LES14010
C                                                                       LES14020
C-----START OF LBISEC--------------------------------------------------LES14030
C                                                                       LES14040
      SUBROUTINE LBISEC(ALPHA,BETA,EPSM,EVAL,EVALN,LB,UB,TTOL,M,NEVT)   LES14050
C                                                                       LES14060
C---------------------------------------------------------------------LES14070
      DOUBLE PRECISION  ALPHA(1),BETA(1),X0,X1,XL,XU,YU,YV,LB,UB        LES14080
      DOUBLE PRECISION  EPSM,EP1,EVAL,EVALN,EVD,EPT                     LES14090
      DOUBLE PRECISION  ZERO,ONE,HALF,TTOL,TEMP                         LES14100
      DOUBLE PRECISION  DABS,DSQRT,DFLOAT                               LES14110
C---------------------------------------------------------------------LES14120
C     SPECIFY PARAMETERS                                                LES14130
      ZERO = 0.0D0                                                      LES14140
      HALF = 0.5D0                                                      LES14150
      ONE  = 1.0D0                                                      LES14160
      XL = LB                                                           LES14170
      XU = UB                                                           LES14180
```

```
C
C     EP1 = DSQRT(1000+M)*TTOL     TTOL = EPSM*TKMAX
C     TKMAX = MAX(|ALPHA(K)|,BETA(K), K= 1,KMAX)
C
      TEMP = DFLOAT(1000+M)
      EP1 = DSQRT(TEMP)*TTOL
C
      NA = 0
      X1 = XU
      JSTURM = 1
      GO TO 60
C     FORWARD STURM CALCULATION
   10 NA = NEV
      X1 = XL
      JSTURM = 2
      GO TO 60
C     FORWARD STURM CALCULATION
   20 NEVT = NEV
C
C     WRITE(6,30) M,EVAL,NEVT,EP1
   30 FORMAT(/3X,'TSIZE',23X,'EV',9X/I8,E25.16/
     1 I6,' = NUMBER OF T(1,M) EIGENVALUES ON TEST INTERVAL'/
     1 E12.3,' =  CONVERGENCE TOLERANCE'/)
C
      IF (NEVT.NE.1) GO TO 120
C
C     BISECTION LOOP
      JSTURM = 3
   40 X1 = HALF*(XL+XU)
      X0 = XU-XL
      EPT = EPSM*(DABS(XL) + DABS(XU)) + EP1
C     CONVERGENCE TEST
      IF (X0.LE.EP1) GO TO 100
      GO TO 60
C     FORWARD STURM CALCULATION
   50 CONTINUE
      IF(NEV.EQ.0) XU = X1
      IF(NEV.EQ.1) XL = X1
      GO TO 40
C     NEV = NUMBER OF T-EIGENVALUES OF T(1,M) ON (X1,XU)
C     THERE IS EXACTLY ONE T-EIGENVALUE OF T(1,M) ON (XL,XU)
C
C     FORWARD STURM CALCULATION
   60 NEV = -NA
      YU = ONE
      DO 90 I = 1,M
      IF (YU.NE.ZERO) GO TO 70
      YV = BETA(I)/EPSM
      GO TO 80
   70 YV = BETA(I)*BETA(I)/YU
   80 YU = X1 - ALPHA(I) - YV
      IF (YU.GE.ZERO) GO TO 90
      NEV = NEV+1
   90 CONTINUE
      GO TO (10,20,50), JSTURM
```

```
C
  100 CONTINUE
C
      EVALN = X1
      EVD = DABS(EVALN-EVAL)
C     WRITE(6,110) EVALN,EVAL,EVD
  110 FORMAT(/20X,'EVALN',21X,'EVAL',6X,'CHANGE'/2E25.16,E12.3/)
C
  120 CONTINUE
      RETURN
C-----END OF LBISEC-----------------------------------------------------
      END
C-----START OF COMPLEX INNER PRODUCT------------------------------------
C
C     COMPLEX INNER PRODUCT
C
      SUBROUTINE CINPRD(V2,V1,SUM,N)
C-----------------------------------------------------------------------
      DOUBLE PRECISION  ZERO,SUM
      COMPLEX*16 V2(1),V1(1),SUMC
C-----------------------------------------------------------------------
C
C     NOTE THAT THE ORDER MATTERS HERE
C     COMPUTES THE INNER PRODUCT OF THE CONJUGATE OF V2 WITH V1.
      ZERO = 0.D0
      SUMC = DCMPLX(ZERO,ZERO)
      DO 10 J=1,N
   10 SUMC = SUMC + DCONJG(V2(J))*V1(J)
      SUM = DREAL(SUMC)
C
      RETURN
C-----END OF COMPLEX INNER PRODUCT SUBROUTINE---------------------------
      END
C
C-----LPERM-PERMUTES VECTORS--------------------------------------------
C
      SUBROUTINE LPERM(W,U,IPERM)
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION  U(1),W(1)
      INTEGER  IPR(1),IPT(1)
C-----------------------------------------------------------------------
C     SUBROUTINE HAS 2 BRANCHES:  IPERM = 1,  CALCULATES
C     U = P*W
C     LET J = IPR(K) THEN U(K) = W(J), K = 1,N.
C     IPERM = 2, CALCULATES
C     U = P'*W  LET J = IPT(K) THEN U(K) = W(J), K=1,N.
C-----------------------------------------------------------------------
      GO TO 3
      ENTRY LPERME(IPR,IPT,N)
      GO TO 4
C-----------------------------------------------------------------------
C
    3 IF(IPERM.EQ.2) GO TO 30
C
```

```
C-----------------------------------------------------------------------
C     IPERM = 1
      DO 10 K = 1,N
      J = IPR(K)
   10 U(K) = W(J)
      GO TO 60
C----------------------------------------------------------------------
C     IPERM = 2
   30 DO 40 K = 1,N
      J = IPT(K)
   40 U(K) = W(J)
C----------------------------------------------------------------------
   60 CONTINUE
      DO 50 K = 1,N
   50 W(K) = U(K)
C
    4 RETURN
C
C-----END OF LPERM-------------------------------------------------
      END
```

2.7 LECOMPAC: Optional Preprocessing Program

```
C-----LECOMPAC-(STAND-ALONE PROGRAM)-----------------------------------
C  AUTHOR:   RALPH A. WILLOUGHBY (Deceased)
C
C
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
C     THIS PROGRAM TRANSLATES A SPARSE SYMMETRIC N X N MATRIX A,
C     GIVEN AS I, J, A(I,J), INTO THE SPARSE MATRIX FORMAT
C     REQUIRED BY THE SAMPLE USPEC AND CMATV PROGRAMS PROVIDED
C     FOR USE WITH THE LANCZOS EIGENVALUE/EIGENVECTOR PROCEDURES.
C     THIS PROGRAM ASSUMES THAT THE MATRIX ENTRIES ARE PROVIDED
C     EITHER COLUMN BY COLUMN OR ROW BY ROW.
C     NOTE THAT THIS PROGRAM DOES NOT DIRECTLY APPLY TO THE
C     HERMITIAN CASE BECAUSE FOR HERMITIAN MATRICES THE DIAGONALS
C     ARE REAL AND THE OFF-DIAGONAL ENTRIES ARE COMPLEX VARIABLES.
C
C     NONPORTABLE STATEMENTS:  PFORT VERIFIER INDICATES THAT THIS
C                              IS PORTABLE.
C
C----------------------------------------------------------------------
      DOUBLE PRECISION A(15000), AD(2000)
      DOUBLE PRECISION ZERO
      INTEGER IROW(15000),ICOL(15000)
C----------------------------------------------------------------------
C    INPUT FILE 7 CONTAINS THE SPARSE SYMMETRIC NXN MATRIX STORED AS:
C
C               NZ,M,N,MATNO
C               I(K)  J(K)  A(K)  K = 1,NZ
C
C     WHERE NZ IS THE TOTAL NUMBER OF NONZEROS IN THE MATRIX A,
C     N IS THE ROW AND COLUMN DIMENSION OF A,
C     AND A(K) ARE THE NONZERO ENTRIES STORED ROW BY ROW OR
C     COLUMN BY COLUMN.  PROGRAM READS THIS IN AS IROW(K) = I(K),
C     ICOL(K) = J(K), AND A(K) = A(K).
C
C     OUTPUT FILE = 8 CONTAINS THE A-MATRIX IN SPARSE FORMAT
C
C             NZS,N,NZL,MATNO
```

```
C              ICOL(K)    K = 1,NZL
C              IROW(K)    K = 1,NZS
C                AD(K)    K = 1,N
C                 A(K)    K = 1,NZS
C
C       WHERE N IS THE ORDER OF THE INPUT MATRIX A,
C       NZ EQUALS THE NUMBER OF NONZERO ELEMENTS IN A WHICH ARE ON
C       OR BELOW THE MAIN DIAGONAL.  NZL EQUALS THE NUMBER OF THE
C       LAST COLUMN HAVING NONZEROES BELOW THE DIAGONAL IN A.
C       NZS EQUALS THE NUMBER OF NONZERO ELEMENTS BELOW THE MAIN
C       DIAGONAL.   AD(K), K=1,N, CONTAINS THE DIAGONAL ELEMENTS OF A.
C       A(K), K=1,NZS, CONTAINS THE KTH NONZERO SUB-DIAGONAL ELEMENT
C       OF THE INPUT MATRIX.  A IS STORED COLUMN BY COLUMN.
C       IROW(K), K=1,NZS, CONTAINS THE ROW INDEX OF THE  NONZERO
C       STRICTLY LOWER TRIANGULAR ELEMENT A(K).
C       ICOL(K), K=1,NZL, EQUALS THE NUMBER OF STRICTLY LOWER
C       TRIANGULAR NONZEROES IN COLUMN K OF THE INPUT MATRIX.
C
C-----------------------------------------------------------------------
      ZERO = 0.0D0
C
      READ(7,10) NZ,N,MATNO,IIROW
   10 FORMAT(2I6,I8,I4)
C
      WRITE(6,20) NZ,N,MATNO,IIROW
   20 FORMAT(I10,I6,I10,' = NO. NONZERO AIJ J.GE.I, ORDER OF A, MATNO'/
     1 I6,' = IIROW IF IIROW=0 ORDERING IS BY COLS IIROW=1 BY ROWS'/)
C
      DO 30 K = 1,N
   30 AD(K) = ZERO
C
      IF (IIROW.EQ.0) READ(7,40) (IROW(K),ICOL(K),A(K), K=1,NZ)
C
      IF (IIROW.EQ.1) READ(7,40) (ICOL(K),IROW(K),A(K), K=1,NZ)
   40 FORMAT(2I5,E14.7)
C
      LCOUNT = 0
      K = 1
C
C     START OF A NEW COLUMN
   50 CONTINUE
      J = ICOL(K)
      ICOL(J) = 0
   60 CONTINUE
C
      IF (J.NE.IROW(K)) GO TO 70
C
C     DIAGONAL CASE
      AD(J) = A(K)
      GO TO 80
C
C     SUB-DIAGONAL NONZERO
   70 CONTINUE
      NZL = J
      LCOUNT = LCOUNT + 1
```

```
      A(LCOUNT) = A(K)
      IROW(LCOUNT) = IROW(K)
      ICOL(J) = ICOL(J) + 1
C
   80 CONTINUE
      K = K+1
C
      IF(K.GT.NZ) GO TO 90
C
      IF(ICOL(K).GT.J) GO TO 50
C
      GO TO 60
C
   90 CONTINUE
      NZS = LCOUNT
C
      WRITE(8,100) NZS,N,NZL,MATNO
      WRITE(6,100) NZS,N,NZL,MATNO
  100 FORMAT(I10,2I6,I8,' = NZS N NZL MATNO')
C
      WRITE(8,110) (ICOL(I), I=1,NZL)
      WRITE(8,110) (IROW(K), K=1,NZS)
  110 FORMAT(13I6)
C
      WRITE(8,120) (AD(K), K=1,N)
      WRITE(8,120) (A(K), K=1,NZS)
  120 FORMAT(4E19.10)
C
C-----END LECOMPAC-----------------------------------------------------
      STOP
      END
```

2.8 LEVAL: LEVEC: File Definitions, Sample Input Files

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

```
Sample Specifications of the Input/Output Files for LEVAL
-----------------------------------------------------------------
 LEVAL EXEC LANCZOS EIGENVALUE CALCULATION REAL SYMMETRIC MATRICES
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 LEVAL     INPUT     A (RECFM F LRECL 80 BLOCK 80
FILEDEF  8 DISK &1        INPUT     A (RECFM F LRECL 80 BLOCK 80
FILEDEF 11 DISK &1        DISTINCT  A (RECFM F LRECL 80 BLOCK 80
LOAD   LEVAL   LESUB   LEMULT
-----------------------------------------------------------------
```

```
Sample Input File for LEVAL
-----------------------------------------------------------
 LANCZOS EIGENVALUE COMPUTATIONS, NO REORTHOGONALIZATION
 TEST MATRIX
LINE 1     N    KMAX    NMEVS     MATNO
         143     429        1    706830
LINE 2   SVSEED    RHSEED      MXINIT    MXSTUR
        7892713    147935           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))
           190
LINE 7    NINT     (NUMBER OF SUB-INTERVALS FOR BISEC)
             1
LINE 8    LB(1)    LB(2)   LB(3)  (INTERVAL LOWER BOUNDS)
            0.0
LINE 9    UB(1)    UB(2)   UB(3)  (INTERVAL UPPER BOUNDS)
          1.001
-----------------------------------------------------------
```

Below is a listing of the input/output filew which are accessed by the real symmetric Lanczos eigenvector program, LEVEC. Included also is a sample of the input file which LEVEC requires on file 5. The parameters are supplied in free format. LEVEC computes eigenvectors for each of a user-specified subset of the eigenvalues computed by the companion program LEVAL. Eigenvector approximation are computed only for eigenvalue approximations which have 'converged'.

```
Sample Specifications of the Input/Output Files for LEVEC
----------------------------------------------------------------------
 LEVEC 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 LEVEC    INPUT     A (RECFM F LRECL 80 BLOCK 80
FILEDEF  8 DISK &1       INPUT     A (RECFM F LRECL 80 BLOCK 80
FILEDEF  9 DISK &1       ERREST    A (RECFM F LRECL 80 BLOCK 80
FILEDEF 10 DISK &1       BOUNDS    A (RECFM F LRECL 80 BLOCK 80
FILEDEF 11 DISK &1       TEIGVECS  A (RECFM F LRECL 80 BLOCK 80
FILEDEF 12 DISK &1       RITZVECS  A (RECFM F LRECL 80 BLOCK 80
FILEDEF 13 DISK &1       PAIGE     A (RECFM F LRECL 80 BLOCK 80
LOAD  LEVEC  LESUB  LEMULT
----------------------------------------------------------------------
```

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