Chapter 7

Nondefective Complex Symmetric Matrices

7.1 Introduction

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

$$Ax = \lambda x. \tag{7.1.1}$$

Definition 3 . *A complex nxn matrix* $A \equiv (a_{ij})$, $1 \leq i, j \leq n$, *is complex symmetric if and only if for every* i *and* j, $a_{ij} = a_{ji}$. *A complex symmetric matrix is nondefective if and only if it has a complete set of eigenvectors.*

It is straight-forward to show from Definition 3 that if $A = B + iC$, where A and B are real matrices and $i = \sqrt{-1}$, is a complex symmetric matrix then B and C are real symmetric matrices. It is also easy to prove that if λ and μ are two distinct eigenvalues of A and x and y are corresponding eigenvectors of A, then the Euclidean inner product applied to the complex vectors x and y satisfies

$$x^T y = 0. \tag{7.1.2}$$

In Eqn(7.1.2) the superscript T denotes transpose. Thus, although the eigenvectors of a complex symmetric matrix are not orthogonal with respect to the complex norm, $\|x\|_C^2 = \sum_{i=1}^n \overline{x(i)} x(i)$, they are real orthogonal in the sense specified in Eqn(7.1.2). Therefore, when we consider generalizing the Lanczos recursion to the complex symmetric case we are led to consider an 'inner product' which is a mixture of real and complex quantities. In fact the Euclidean inner product, which of course is not an inner product for complex vectors, is the natural 'inner product' to use in the complex symmetric case.

Complex symmetric matrices are not 'easy' like real symmetric matrices. They bear little resemblance to real symmetric matrices. Complex symmetric matrices need not have complete sets of eigenvectors. Even if a complete set of eigenvectors exists, eigenvectors corresponding to different eigenvalues are only real orthogonal in the sense of Eqn(7.1.2). If a small perturbation is applied to a complex symmetric matrix, then large perturbations in the eigenvalues may result. See Wilkinson [25] for a discussion of the properties of complex symmetric matrices.

The Lanczos recursion as presented in Eqns(1.2.1), (1.2.2) is only applicable to real symmetric matrices so we ask the question: How do we construct a complex symmetric version of the basic Lanczos recursion which will give us the desired eigenvalues? We have used what has been suggested elsewhere, Moro and Freed [16]. In particular, we use the recursion in Eqn(1.2.1) with the formulas for the scalars α_i and β_{i+1} given in Eqn(1.2.2), except that the quantities involved are now complex-valued, but the real Euclidean inner product is used. See Section 6.3 in Chapter 6 in Volume 1.

There are some fundamental differences between the amount of computation required by the complex symmetric codes versus that required by the real symmetric codes. First, all of the complex symmetric computations are done in double precision complex arithmetic. All the vectors used are complex vectors. Each of the Lanczos matrices generated is a complex symmetric tridiagonal matrix. Unfortunately, there is no simple analog of the bisection procedure used in the real symmetric case which would allow us to compute the eigenvalues of a given complex symmetric tridiagonal matrix on only some small portion of the spectrum. We are therefore forced to do a complete eigenvalue computation on each complex symmetric tridiagonal matrix which we consider. Actually in the complex symmetric case we are forced to do two complete eigenvalue computations for each Lanczos tridiagonal matrix which we consider. Two are required because the identification test for categorizing the eigenvalues of the Lanczos T-matrices into 'good' and 'spurious' ones uses the eigenvalues of the corresponding tridiagonal matrix obtained from the Lanczos T-matrix by crossing out the first row and column of that matrix. This is the same identification test as that used in the procedures for real symmetric problems. However, in the real symmetric cases this test is directly incorporated into the BISEC subroutine which is used to compute the eigenvalues of the Lanczos matrices, and the resulting cost of this test is negligible for those types of problems.

These codes can be used to compute either a very few or very many of the distinct eigenvalues of a nondefective, complex symmetric matrix. As the documentation in the next section indicates, the A-multiplicity of a given computed eigenvalue can be obtained only with additional computation, and the modifications required to do this additional computation are not included in these versions of the codes.

The Lanczos recursions used generate a family of complex symmetric, tridiagonal matrices. A real orthogonal analog of the EISPACK [23, 8] subroutine IMTQL1 which we call CMTQL1 was developed to compute the eigenvalues of the complex symmetric, tridiagonal Lanczos matrices generated. There is no reorthogonalization of the Lanczos vectors at any stage in any of the computations.

CSLEVAL, the main program for the complex symmetric eigenvalue computations, calls the subroutines COMPEV and CMTQL1 to compute the eigenvalues of the Lanczos T-matrices specified by the user. The eigenvalues of the related complex symmetric tridiagonal matrices obtained by deleting the first row and first column from the given Lanczos T-matrix are also computed. COMPEV then determines the T-multiplicities of the T-eigenvalues 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 a complex version of the subroutine INVERR. Error estimates are computed only for isolated 'good' T-eigenvalues. All other 'good' T-eigenvalues are assumed to have converged. Convergence is then checked. If convergence has not yet occurred and a larger Lanczos matrix has been specified by the user, the program will continue on to the larger T-matrix, repeating the above procedure on this larger matrix.

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

As stated earlier, all computations are in double precision complex 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.

The user should note that the complex symmetric computations are considerably more expensive than the corresponding real symmetric ones. Two complete T-matrix eigenvalue computations must be done for each T-size. Moreover, the accuracy of these computations is noticeably less than that achievable in the real symmetric case. This is to be expected from the perturbation analysis for the complex symmetric case. Therefore we reduced the anticipated accuracy of the computed eigenvalues and used larger tolerances in our multiplicity and spuriousness tests. These larger tolerances decrease the resolution capabilities of these codes. However, these tolerances are realistic. Moreover, these complex symmetric codes cannot be expected to handle stiff problems effectively. More details about these complex symmetric, single-vector Lanczos procedures are included in Chapter 6 of Volume 1.

7.2 Documentation for the Codes in Chapter 7

```
C-----CSLEVALD-----------------------------------------------------
C
C     DOCUMENTATION FOR SINGLE-VECTOR
C     LANCZOS EIGENVALUE/EIGENVECTOR PROGRAMS FOR
C     NONDEFECTIVE COMPLEX SYMMETRIC MATRICES
C
C-------------------------------------------------------------------------
C         REFERENCE:  Cullum and Willoughby, Chapter 6,
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-------------------------------------------------------------------------
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
C  are used in other scientific or engineering research works
C  the names of the authors of these codes and appropriate
C  references to their written work are to be incorporated in the
C  derivative works.
C
C  This header is not to be removed from these codes.
C
C     GIVEN A NONDEFECTIVE COMPLEX SYMMETRIC MATRIX A OF ORDER N
C     THE THREE SETS OF FORTRAN FILES LABELLED CSLEVAL, CSLESUB,
C     AND CSLEMULT CAN BE USED TO COMPUTE DISTINCT EIGENVALUES OF
C     A.  NOTE THAT THESE PROGRAMS DIFFER FROM THE REAL SYMMETRIC
C     AND HERMITIAN PROGRAMS IN THAT IT IS NOT POSSIBLE TO
C     COMPUTE THE EIGENVALUES OF THE LANCZOS TRIDIAGONAL MATRICES
C     ONLY IN SPECIFIED INTERVALS.  THUS, ON ANY GIVEN
C     ITERATION ALL OF THE EIGENVALUES OF THESE TRIDIAGONAL MATRICES
C     MUST BE COMPUTED.  IN FACT TWO COMPLETE TRIDIAGONAL EIGENVALUE
C     COMPUTATIONS ARE USED.
C
C     CORRESPONDING EIGENVECTORS FOR SELECTED, COMPUTED EIGENVALUES CAN
C     BE COMPUTED USING THE CORRESPONDING SETS OF FILES LABELLED
C     CSLEVEC, CSLESUB AND CSLEMULT.
C
C     THESE PROGRAMS ALL USE A GENERALIZATION OF LANCZOS
C     TRIDIAGONALIZATION TO COMPLEX SYMMETRIC MATRICES TO
```

```
C     GENERATE COMPLEX SYMMETRIC TRIDIAGONAL MATRICES, T(1,MEV)
C     OF ORDER MEV.  NO REORTHOGONALIZATION IS USED.  SUBSETS OF
C     THE EIGENVALUES OF THESE T-MATRICES, LABELLED AS THE
C     'GOOD EIGENVALUES', YIELD APPROXIMATIONS TO THE DESIRED
C     EIGENVALUES OF A.  CORRESPONDING RITZ VECTORS ARE APPROXIMATIONS
C     TO THE DESIRED EIGENVECTORS OF A.  NOTE THAT IN THE DISCUSSION
C     T(1,MEV) DENOTES THE LANCZOS MATRIX OF ORDER MEV AND T(2,MEV)
C     DENOTES THE MATRIX OF SIZE MEV-1 OBTAINED FROM T(1,MEV) BY
C     DELETING THE FIRST ROW AND COLUMN OF T(1,MEV).
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,  PROGRESS IN
C         SCIENTIFIC COMPUTING, EDITORS, G. GOLUB, H.O. KREISS,
C         S. ARBARBANEL, AND R. GLOWINSKI,  BIRKHAUSER BOSTON INC.,
C         CAMBRIDGE, MASSACHUSETTS, 1985.
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     5. IBID, A LANCZOS ALGORITHM FOR NONDEFECTIVE COMPLEX SYMMETRIC
C        MATRICES, IBM RESEARCH REPORT, 1984.
C
C-----PORTABILITY-----------------------------------------------------
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     PORTABILITY:
C     THESE PROGRAMS ARE NOT PORTABLE DUE TO THE USE OF COMPLEX*16
C     VARIABLES AND CORRESPONDING COMPLEX FUNCTIONS.  IN ADDITION, THE
C     PFORT VERIFIER IDENTIFIED THE FOLLOWING NONPORTABLE
C     CONSTRUCTIONS.
C     IN CSLEVAL AND IN CSLEVEC
C         1.  DATA/MACHEP STATEMENT
C         2.  ALL READ(5,*) STATEMENTS (FREE FORMAT)
C         3.  FORMAT(20A4) USED FOR THE EXPLANATORY HEADER ARRAY, EXPLAN
C         4.  HEXADECIMAL FORMAT (4Z20) FOR ALPHA/BETA FILES 1 AND 2.
```

```
C     IN CSLEMULT
C         1.  IN CMATV AND  USPEC THE ENTRY THAT PASSES THE STORAGE
C             LOCATIONS OF THE ARRAYS DEFINING THE USER-SPECIFIED
C             MATRIX.
C         2.  IN SAMPLE USPEC PROVIDED : FREE FORMAT (8,*), THE
C             FORMAT (20A4), AND THE DATA/MACHEP STATEMENT.
C
C     IN THE COMMENTS BELOW :
C     REAL*16 = COMPLEX VARIABLE, 16 BYTES OF STORAGE
C     REAL*8 = REAL VARIABLE, 8 BYTES OF STORAGE
C     REAL*4 = REAL VARIABLE, 4 BYTES OF STORAGE
C     INTEGER*4 = INTEGER VARIABLE, 4 BYTES OF STORAGE
C
C-----A-MATRIX SPECIFICATION--------------------------------------------
C
C     SUBROUTINE USPEC IS USED TO SPECIFY THE USER-SUPPLIED MATRIX.
C     SUBROUTINE CMATV IS A CORRESPONDING MATRIX-VECTOR MULTIPLY
C     SUBROUTINE WHICH SHOULD BE DESIGNED TO TAKE ADVANTAGE OF
C     ANY SPECIAL PROPERTIES OF THE USER-SUPPLIED MATRIX.  THE
C     MATRIX-VECTOR MULTIPLIES REQUIRED BY THE LANCZOS PROCEDURES
C     MUST BE COMPUTED RAPIDLY AND ACCURATELY.
C
C     SUBROUTINE USPEC HAS THE CALLING SEQUENCE
C
C            CALL USPEC(N,MATNO)
C
C     WHERE N IS THE ORDER OF THE USER-SUPPLIED MATRIX A AND
C     MATNO IS A <= 8 DIGIT INTEGER USED AS A MATRIX AND
C     TEST IDENTIFICATION NUMBER.  THIS SUBROUTINE DEFINES (DIMENSIONS)
C     THE ARRAYS REQUIRED TO SPECIFY THE USER-SUPPLIED MATRIX AND
C     INITIALIZES THESE ARRAYS AND ANY OTHER PARAMETERS NEEDED TO
C     DEFINE THE MATRIX.  THE STORAGE LOCATIONS OF THESE PARAMETERS
C     AND ARRAYS ARE THEN PASSED TO THE MATRIX-VECTOR MULTIPLY
C     SUBROUTINE CMATV VIA AN ENTRY.  A SAMPLE USPEC SUBROUTINE
C     IS INCLUDED.  THIS SAMPLE SUBROUTINE ASSUMES THAT THE MATRIX
C     IS STORED ON FILE 8 IN A TYPICAL SPARSE MATRIX FORMAT.
C     SEE THE HEADER ON THE SUBROUTINE USPEC FOR DETAILS ON THIS
C     PARTICULAR STORAGE FORMAT.
C
C     SUBROUTINE CMATV HAS THE CALLING SEQUENCE
C
C            CALL CMATV(W,U,SUM)
C
C     IN THE COMPLEX SYMMETRIC CASE, U AND W ARE
C     COMPLEX*16 VECTORS AND SUM IS A COMPLEX*16
C     SCALAR.   CMATV CALCULATES U = A*W - SUM*U FOR THE
C     USER-SPECIFIED MATRIX A.  THE ARRAY AND PARAMETER INFORMATION
C     NEEDED TO PERFORM THE MATRIX-VECTOR MULTIPLIES IS PASSED TO
C     THE CMATV SUBROUTINE FROM THE USPEC SUBROUTINE VIA THE CMATVE
C     ENTRY IN CMATV.  A SAMPLE CMATV SUBROUTINE IS INCLUDED WHICH
C     COMPUTES MATRIX-VECTOR MULTIPLIES FOR AN ARBITRARY SPARSE,
C     COMPLEX SYMMETRIC MATRIX STORED IN THE SPARSE FORMAT
C     SPECIFIED IN THE SAMPLE USPEC SUBROUTINE.
C
C     CMATV IS CALLED FROM THE SUBROUTINE LANCZS WHICH GENERATES
```

```
C     THE T-MATRICES IN THE ALPHA, BETA ARRAYS.  IT IS ALSO CALLED
C     FROM THE MAIN PROGRAM CSLEVEC FOR THE EIGENVECTOR COMPUTATIONS.
C     CMATV IS DECLARED AS AN EXTERNAL VARIABLE AND IS AN ARGUMENT
C     FOR THE SUBROUTINE LANCZS.
C
C     THE USPEC AND CMATV SUBROUTINES MUST BE MODIFIED BY THE USER
C     TO ACCOMODATE THE USER'S SPECIFIED MATRIX.
C
C     THE MAIN PROGRAMS FOR THE EIGENVALUE AND EIGENVECTOR
C     CALCULATIONS ASSUME THAT INPUT FILE 5 CONTAINS N = ORDER OF
C     THE MATRIX AND MATNO = AN IDENTIFICATION NUMBER OF <= 8 DIGITS
C     FOR THE MATRIX AND THE RUN.
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     GENRAN,  MASK,  USPEC,  AND  CMATV
C
C     GENRAN = COMPUTES K PSEUDO-RANDOM NUMBERS AND STORES THEM IN
C              THE REAL ARRAY, G.  THIS SUBROUTINE IS USED TO
C              GENERATE A STARTING VECTOR FOR THE LANCZOS PROCEDURE
C              IN THE SUBROUTINE LANCZS AND A STARTING RIGHT-HAND SIDE
C              FOR INVERSE ITERATION IN THE SUBROUTINE INVERR.
C
C              TESTS REPORTED IN THE REFERENCES USED EITHER GGL1 OR
C              GGL2 FROM THE IBM LIBRARY  SLMATH.
C              THE EXISTING CALLING SEQUENCE IS:
C
C                      CALL GENRAN(IIX,G,K).
C
C              WHERE IIX =INTEGER SEED, G = REAL*4 ARRAY WHOSE
C              DIMENSION MUST BE >= K.  K RANDOM NUMBERS ARE GENERATED
C              AND PLACED IN G.
C
C     MASK = MASKS OVERFLOW AND UNDERFLOW.
C            USER MUST SUPPLY OR COMMENT OUT CALL.
C
C     USPEC = DIMENSIONS AND INITIALIZES ARRAYS NEEDED TO SPECIFY
C             USER-SUPPLIED MATRIX.  SEE A-MATRIX SPECIFICATION SECTION.
C
```

```
C     CMATV = MATRIX-VECTOR MULTIPLY FOR USER-SUPPLIED MATRIX.
C             SEE A-MATRIX SPECIFICATION SECTION.
C
C
C-----------------------------------------------------------------------
C
C     COMMENTS FOR EIGENVALUE COMPUTATIONS
C
C-----------------------------------------------------------------------
C
C
C-----PARAMETER CONTROLS FOR EIGENVALUE PROGRAMS------------------------
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.
C
C              (1) PROGRAM COMPUTES REQUESTED ALPHAS/BETAS AND THEN
C                  USES THE CMTQL1 SUBROUTINE TO CALCULATE EIGENVALUES
C                  OF THE TRIDIAGONAL MATRICES GENERATED FOR THE ORDERS
C                  SPECIFIED BY THE USER.  PROGRAM THEN USES THE
C                  SUBROUTINE INVERR TO COMPUTE ERROR ESTIMATES FOR
C                  THE ISOLATED GOOD T-EIGENVALUES WHICH ARE USED TO
C                  CHECK THE CONVERGENCE OF THESE GOOD T-EIGENVALUES.
C
C     CONTROL PARAMETERS FOR WRITES
C
C     IHIS  = (0,1)  MEANS
C
C             (0) IF ISTOP .GT. 0 THEN ALPHAS/BETAS ARE NOT SAVED
C                 ON 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
C                 BE SAVED IN MACHINE FORMAT ((4Z20) FOR IBM/3081)
C                 SO THAT NO ERRORS ARE INTRODUCED DUE TO FORMAT
C                 CONVERSIONS.
C
C     IDIST  = (0,1)  MEANS
C
C              (0) DISTINCT EIGENVALUES OF T-MATRICES ARE NOT SAVED.
C
C              (1) PROGRAM WRITES COMPUTED DISTINCT EIGENVALUES OF
C                  T-MATRICES ALONG WITH THEIR T-MULTIPLICITIES
C                  TO FILE 11.
C
C     IWRITE = (0,1)   MEANS
C
C              (0) NO EXTENDED OUTPUT FROM SUBROUTINES COMPEV AND INVERR
C                  IS SENT TO FILE 6.
C
C              (1) INDIVIDUAL COMPUTED EIGENVALUES AND CORRESPONDING
C                  ERROR ESTIMATES FROM THE SUBROUTINES COMPEV AND
C                  INVERR ARE PRINTED OUT TO FILE 6 AS THEY ARE COMPUTED
C
C     SAVTEV = (-1,0,1)   MEANS
C
C              (-1) NO T-EIGENVALUE COMPUTATIONS.  PREVIOUSLY-COMPUTED
C                   EIGENVALUES OF T(1,MEV) AND T(2,MEV) ARE TO
C                   BE READ IN FROM FILE 10.
C
C              (0) COMPUTED EIGENVALUES OF T(1,MEV) AND OF T(2,MEV)
C                  ARE NOT TO BE SAVED ON FILE 10.  THIS IS NOT
C                  RECOMMENDED IF THE T-MATRICES BEING USED ARE VERY
C                  LARGE BECAUSE IN THAT CASE THE TRIDIAGONAL
C                  EIGENVALUE COMPUTATIONS ARE VERY EXPENSIVE.
C
C              (1) COMPUTED EIGENVALUES OF T(1,MEV) AND OF T(2,MEV)
C                  WILL BE SAVED ON FILE 10.  THIS IS RECOMMENDED
C                  BECAUSE ONCE THESE T-EIGENVALUES ARE COMPUTED THE
C                  LATTER PORTION OF THE EIGENVALUE PROGRAM IS EASILY
C                  RESTARTED FROM THE POINT OF THESE EIGENVALUE
C                  COMPUTATIONS.
C
C     THE PROGRAM ALWAYS MAKES A SEPARATE LIST OF THE COMPUTED GOOD
C     T-EIGENVALUES ALONG WITH THEIR MINIMAL GAPS AND WRITES THEM OUT
C     TO FILE 3.  CORRESPONDING ERROR ESTIMATES FOR ANY ISOLATED
C     GOOD T-EIGENVALUES ARE ALWAYS WRITTEN TO FILE 4.
C
C
C-----INPUT/OUTPUT FILES FOR EIGENVALUE PROGRAMS------------------------
C
C     ANY INPUT DATA OTHER THAN THE ALPHA/BETA HISTORY OR PREVIOUSLY-
C     COMPUTED EIGENVALUES OF T(1,MEV) AND T(2,MEV) SHOULD BE STORED
C     ON FILE 5. SEE SAMPLE INPUT/OUTPUT FROM TYPICAL RUN.
C     THE READ STATEMENTS IN THE GIVEN FORTRAN PROGRAM ASSUME THAT
C     THE DATA STORED ON FILE 5 IS IN FREE FORMAT.  USER SHOULD NOTE
```

```
C     THAT 'FREE FORMAT' IS NOT CLASSIFIED AS PORTABLE BY PFORT SO THAT CSL03270
C     THE USER MAY HAVE TO MODIFY THE READ STATEMENTS FROM FILE 5 TO    CSL03280
C     CONFORM TO WHAT IS PERMISSIBLE ON THE MACHINE BEING USED.         CSL03290
C                                                                       CSL03300
C     FILE 6 WAS USED AS THE INTERACTIVE TERMINAL OUTPUT FILE.          CSL03310
C     THIS FILE PROVIDES A RUNNING ACCOUNT OF THE PROGRESS OF THE       CSL03320
C     COMPUTATIONS.  THE AMOUNT OF INFORMATION PRINTED OUT IS           CSL03330
C     CONTROLLED BY THE PARAMETER IWRITE.                               CSL03340
C                                                                       CSL03350
C DESCRIPTION OF OTHER I/O FILES                                        CSL03360
C                                                                       CSL03370
C FILE (K)     CONTAINS:                                                CSL03380
C                                                                       CSL03390
C      (1)     OUTPUT FILE:                                             CSL03400
C              HISTORY FILE OF NEWLY-GENERATED T-MATRIX (ALPHA AND      CSL03410
C              BETA VECTORS) AND LAST 2 LANCZOS VECTORS USED            CSL03420
C              IN THE T-MATRIX GENERATION.                              CSL03430
C              IF IHIS = 0 AND ISTOP = 1, FILE 1 IS NOT WRITTEN.        CSL03440
C                                                                       CSL03450
C      (2)     INPUT FILE:                                              CSL03460
C              SAME AS FILE 1 EXCEPT THAT IT CONTAINS A                 CSL03470
C              PREVIOUSLY-GENERATED T-MATRIX (IF ANY).  IF ISTART = 1,  CSL03480
C              PROGRAM ASSUMES THAT THERE IS A HISTORY FILE OF ALPHAS   CSL03490
C              AND BETAS ON FILE 2.  THESE ALPHAS AND BETAS ARE         CSL03500
C              READ IN ALONG WITH THE LAST TWO LANCZOS VECTORS          CSL03510
C              USED IN THE T-MATRIX GENERATION.                         CSL03520
C                                                                       CSL03530
C      (3)     OUTPUT FILE:                                             CSL03540
C              COMPUTED GOOD EIGENVALUES OF THE T-MATRICES USED. ALSO   CSL03550
C              CONTAINS T-MULTIPLICITIES OF THESE EIGENVALUES AS        CSL03560
C              EIGENVALUES OF THE T-MATRIX, AND THEIR GAPS AS           CSL03570
C              EIGENVALUES IN THE A MATRIX AND IN THE T-MATRIX.         CSL03580
C              FILE 3 IS ALWAYS WRITTEN.                                CSL03590
C                                                                       CSL03600
C      (4)     OUTPUT FILE:                                             CSL03610
C              ERROR ESTIMATES FOR THE ISOLATED GOOD T-EIGENVALUES WHICHCSL03620
C              ARE OBTAINED USING THE SUBROUTINE INVERR. THESE          CSL03630
C              ESITMATES USE THE LAST COMPONENTS OF THE ASSOCIATED      CSL03640
C              T-EIGENVECTORS WHICH ARE COMPUTED USING INVERSE          CSL03650
C              ITERATION.  FILE 4 IS ALWAYS WRITTEN.                    CSL03660
C                                                                       CSL03670
C      (8)     INPUT FILE:                                              CSL03680
C              SAMPLE USPEC SUBROUTINE ASSUMES THAT THE ARRAYS          CSL03690
C              REQUIRED TO SPECIFY THE USER'S-MATRIX ARE STORED ON      CSL03700
C              FILE 8.  USERS MUST MAKE WHATEVER DEFINITIONS ARE        CSL03710
C              APPROPRIATE FOR THEIR MATRICES.                          CSL03720
C                                                                       CSL03730
C     (10)     OUTPUT OR INPUT FILE DEPENDING UPON VALUE OF SAVTEV:     CSL03740
C              COMPUTED EIGENVALUES OF EACH T(1,MEV) FOLLOWED           CSL03750
C              BY THE COMPUTED EIGENVALUES OF THE CORRESPONDING         CSL03760
C              T(2,MEV).                                                CSL03770
C                                                                       CSL03780
C     (11)     OUTPUT FILE:                                             CSL03790
C              COMPUTED DISTINCT EIGENVALUES OF T-MATRICES USED.        CSL03800
C              ALSO CONTAINS THEIR T-MULTIPLICITIES AND T-GAPS TO       CSL03810
```

```
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 PROGRAM-------------------------
C
C     THESE PARAMETERS ARE SET INTERNALLY IN THE PROGRAM
C
C     SCALEK     K = 1,2,3,4
C
C                THE SCALING FACTORS SCALEK HAVE BEEN INTRODUCED IN AN
C                ATTEMPT TO MAKE THE TOLERANCES USED IN THE
C                T-MULTIPLICITY, SPURIOUS, AND ISOLATION TESTS ADJUST
C                TO THE SCALE OF THE GIVEN MATRIX.  THESE FACTORS MUST
C                NOT BE MODIFIED.
C
C     BTOL = RELATIVE TOLERANCE USED TO ESTIMATE ANY LOSS OF LOCAL
C            ORTHOGONALITY OF THE LANCZOS VECTORS AFTER THE T-MATRIX
C            HAS BEEN GENERATED.  THE LANCZOS PROCEDURE WORKS WELL
C            ONLY IF LOCAL ORTHOGONALITY BETWEEN SUCCESSIVE LANCZOS
C            VECTORS IS MAINTAINED.  THE TNORM SUBROUTINE TESTS
C            WHETHER OR NOT
C
C                 MINIMUM  |BETA(I)|/||A|| > BTOL.
C                 I=2,KMAX
C
C            IF THIS TEST IS VIOLATED BY SOME BETA AND A T-MATRIX THAT
C            WOULD INCLUDE SUCH A BETA IS REQUESTED, THEN THE LANCZOS
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
C            AN ESTIMATE 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-7.
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-----USER-SPECIFIED PARAMETERS FOR EIGENVALUE PROGRAMS-----------------
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 CMTQL1 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                                                                       CSL04370
C    |EVALUE(J)-EVALUE(J-1)| < DMAX1(RELTOL*|EVALUE(J)|,SCALE2*MULTOL)  CSL04380
C                                                                       CSL04390
C     THEN THESE T-EIGENVALUES ARE 'COMBINED'.  MULTOL IS THE TOLERANCE CSL04400
C     THAT WAS USED IN THE T-MULTIPLICITY TEST IN COMPEV.  SEE THE      CSL04410
C     HEADER ON THE LUMP SUBROUTINE FOR MORE DETAILS.                   CSL04420
C                                                                       CSL04430
C     THE RECOMMENDED VALUE OF RELTOL (ONLY IN THE COMPLEX SYMMETRIC    CSL04440
C     CASE) IS 1.D-8 BECAUSE THE OBSERVED ACCURACY OF THE               CSL04450
C     COMPUTED EIGENVALUES OF THE T-MATRICES IS SEVERAL DIGITS          CSL04460
C     LESS THAN THAT OBSERVED IN THE REAL SYMMETRIC CASE.               CSL04470
C     THUS, THE OBSERVED RESOLUTION OF THE COMPLEX SYMMETRIC            CSL04480
C     VERSION IS LESS THAN THAT OBTAINABLE IN THE REAL SYMMETRIC CASE.  CSL04490
C                                                                       CSL04500
C     MXINIT = MAXIMUM NUMBER OF INVERSE ITERATIONS ALLOWED IN          CSL04510
C              SUBROUTINE INVERR FOR EACH ISOLATED GOOD T-EIGENVALUE.   CSL04520
C              TYPICALLY ONLY ONE ITERATION IS REQUIRED.                CSL04530
C                                                                       CSL04540
C     SEEDS FOR RANDOM NUMBER GENERATORS = INTEGER*4 SCALARS.           CSL04550
C                                                                       CSL04560
C               (1) SVSEED = SEED FOR STARTING VECTOR USED IN           CSL04570
C                   T-MATRIX GENERATION IN LANCZS SUBROUTINE            CSL04580
C                                                                       CSL04590
C               (2) RHSEED = SEED FOR RIGHT-HAND SIDE USED IN           CSL04600
C                   INVERSE ITERATION COMPUTATIONS IN INVERR.           CSL04610
C                                                                       CSL04620
C                                                                       CSL04630
C     T-MATRICES                                                        CSL04640
C                                                                       CSL04650
C     SIZES OF T-MATRICES                                               CSL04660
C                                                                       CSL04670
C             (1) KMAX= MAXIMUM ORDER FOR T-MATRIX THAT USER IS WILLING CSL04680
C                       TO CONSIDER.                                    CSL04690
C                                                                       CSL04700
C             (2) NMEVS = MAXIMUM NUMBER OF T-MATRICES THAT WILL BE     CSL04710
C                         CONSIDERED.                                   CSL04720
C                                                                       CSL04730
C             (3) NMEV(J)  (J=1,NMEVS)  = SIZES OF T-MATRIX TO BE       CSL04740
C                                         CONSIDERED SEQUENTIALLY.      CSL04750
C                                                                       CSL04760
C     T-MATRIX-GENERATION                                               CSL04770
C                                                                       CSL04780
C     USER SHOULD NOTE THAT THIS PROGRAM FIRST COMPUTES A T-MATRIX      CSL04790
C     OF ORDER KMAX AND THEN CYCLES THROUGH THE T-MATRICES SPECIFIED    CSL04800
C     A PRIORI BY THE USER, USING THE SUBROUTINE CMTQL1 TO COMPUTE THE  CSL04810
C     EIGENVALUES OF THE T-MATRICES.  THE EIGENVALUE COMPUTATION        CSL04820
C     FOR THE COMPLEX SYMMETRIC CASE WILL BE                            CSL04830
C     CONSIDERABLY MORE EXPENSIVE THAN FOR THE REAL SYMMETRIC OR        CSL04840
C     HERMITIAN CASES BECAUSE WE DO NOT HAVE AN ANALOG OF               CSL04850
C     THE BISECTION SUBROUTINE FOR THE COMPLEX SYMMETRIC CASE.          CSL04860
C     THUS, ANY RECYCLING AND SUBSEQUENT ENLARGEMENT OF THE T-MATRIX    CSL04870
C     REQUIRES THE RECOMPUTATION OF ALL OF THE EIGENVALUES OF           CSL04880
C     THE RESULTING T-MATRIX.  WE CANNOT GO IN AND COMPUTE ONLY THOSE   CSL04890
C     T-EIGENVALUES ON SOME SUBINTERVAL OF THE SPECTRUM OF THE          CSL04900
C     T-MATRIX AS WE DID IN THE REAL SYMMETRIC AND HERMITIAN CASES.     CSL04910
```

```
C     OF COURSE, IF THE T-MATRICES BEING CONSIDERED ARE NOT             CSL04920
C     VERY LARGE, THEN THIS IS NOT REALLY A PROBLEM.  HOWEVER, IF THEY  CSL04930
C     ARE VERY LARGE, THEN THE USER SHOULD PROBABLY DO ONE EIGENVALUE   CSL04940
C     COMPUTATION OF A LARGE T-MATRIX RATHER THAN START WITH            CSL04950
C     A SMALLER T-MATRIX AND WORK UP TO A BIG ONE.                      CSL04960
C                                                                       CSL04970
C-----CONVERGENCE TESTS FOR THE EIGENVALUE PROGRAMS---------------------CSL04980
C                                                                       CSL04990
C     THE CONVERGENCE TEST INCORPORATED IN THIS PROGRAM IS              CSL05000
C     BASED UPON THE ASSUMPTION THAT THOSE T-EIGENVALUES AND            CSL05010
C     THEIR ASSOCIATED T-EIGENVECTORS WHICH CORRESPOND TO THE           CSL05020
C     EIGENVALUES AND RITZVECTORS WHICH ARE TO BE COMPUTED              CSL05030
C     CONVERGE AS THE T-SIZE IS INCREASED.                              CSL05040
C                                                                       CSL05050
C-----ARRAYS REQUIRED BY THE EIGENVALUE PROGRAM-------------------------CSL05060
C                                                                       CSL05070
C     ALPHA(J) = COMPLEX*16  ARRAY. ITS DIMENSION MUST BE AT LEAST      CSL05080
C                KMAX, THE LENGTH OF THE LARGEST T-MATRIX ALLOWED.      CSL05090
C                THIS ARRAY CONTAINS THE DIAGONAL ENTRIES OF THE        CSL05100
C                T-MATRICES GENERATED.                                  CSL05110
C                                                                       CSL05120
C     BETA(J) = COMPLEX*16 ARRAY.  ITS DIMENSION MUST BE AT LEAST       CSL05130
C               KMAX+1. THIS ARRAY CONTAINS THE SUBDIAGONAL ENTRIES OF  CSL05140
C               THE T-MATRICES.                                         CSL05150
C                                                                       CSL05160
C               THE ALPHA AND BETA VECTORS ARE NOT ALTERED              CSL05170
C               DURING THE CALCULATIONS.                                CSL05180
C                                                                       CSL05190
C     V1(J),V2(J),VS(J) = COMPLEX*16 ARRAYS.  V1 AND V2                 CSL05200
C                         MUST BE OF DIMENSION AT LEAST MAX(KMAX,N).    CSL05210
C                         VS MUST BE OF DIMENSION AT LEAST KMAX.        CSL05220
C                                                                       CSL05230
C     GR(J),GC(J) = REAL*8 ARRAYS. USED FOR RANDOM VECTOR GENERATION.   CSL05240
C                   EACH MUST BE OF DIMENSION AT LEAST MAX(KMAX,N).     CSL05250
C                                                                       CSL05260
C     EXPLAN(J) = REAL*4 ARRAY.  ITS DIMENSION IS 20.  THIS ARRAY IS    CSL05270
C                 USED TO ALLOW EXPLANATORY COMMENTS IN THE INPUT FILES.CSL05280
C                                                                       CSL05290
C     G(J),GG(J) = REAL*4 ARRAYS.  G MUST BE OF DIMENSION AT LEAST      CSL05300
C                  MAX(N,KMAX).  GG MUST BE OF DIMENSION AT LEAST       CSL05310
C                  KMAX.  G AND GG ARE USED IN RANDOM VECTOR GENERATIONSCSL05320
C                  AND TO STORE GAPS IN T-MATRIX, GAPS IN A-MATRIX,     CSL05330
C                  AND ERROR ESTIMATES.                                 CSL05340
C                                                                       CSL05350
C     MP(J),MP2(J) = INTEGER*4 ARRAYS.  EACH MUST HAVE DIMENSION        CSL05360
C                    AT LEAST KMAX, THE MAXIMUM SIZE OF THE T-MATRICES. CSL05370
C                    MP CONTAINS THE T-MULTIPLICITIES OF THE COMPUTED   CSL05380
C                    T-EIGENVALUES. 'SPURIOUS' T-EIGENVALUES ARE DENOTEDCSL05390
C                    BY A T-MULTIPLICITY OF 0.  NOTE THAT WE DO NOT HAVECSL05400
C                    AN ANALOG OF THE SUBROUTINE PRTEST FOR THE         CSL05410
C                    COMPLEX SYMMETRIC CASE, SO NO RELABELLING OF       CSL05420
C                    MP OCCURS.  MP2 IS USED TO KEEP TRACK OF WHICH     CSL05430
C                    EIGENVALUES OF T(1,MEV) HAVE BEEN USED IN THE      CSL05440
C                    T-MULTIPLICITY TEST AND WHICH EIGENVALUES OF       CSL05450
C                    T(2,MEV) HAVE BEEN USED IN THE SPURIOUS TEST.      CSL05460
```

```
C                                                                       CSL05470
C     NMEV(J) = INTEGER*4 ARRAY.  ITS DIMENSION MUST BE AT LEAST THE    CSL05480
C               NUMBER OF T-MATRICES ALLOWED.  IT CONTAINS THE ORDERS   CSL05490
C               OF THE T-MATRICES TO BE CONSIDERED.                     CSL05500
C                                                                       CSL05510
C     OTHER ARRAYS                                                      CSL05520
C                                                                       CSL05530
C     THE USER MUST SPECIFY IN THE SUBROUTINE USPEC WHATEVER ARRAYS     CSL05540
C     ARE REQUIRED TO DEFINE THE MATRIX BEING USED.                     CSL05550
C                                                                       CSL05560
C                                                                       CSL05570
C-----SUBROUTINES INCLUDED FOR EIGENVALUE COMPUTATIONS------------------CSL05580
C                                                                       CSL05590
C     LANCZS = COMPUTES THE ALPHA/BETA HISTORY. USES SUBROUTINES        CSL05600
C              CINPRD, INPRDC, GENRAN, AND CMATV.                       CSL05610
C                                                                       CSL05620
C     COMPEV = CALLS CMTQL1 TO COMPUTE THE EIGENVALUES OF T(1,MEV)      CSL05630
C              AND OF T(2,MEV), THEN DETERMINES T-MULTIPLE AND          CSL05640
C              SPURIOUS T-EIGENVALUES.                                  CSL05650
C                                                                       CSL05660
C     COMGAP = COMPUTES MINIMAL GAPS BETWEEN T-EIGENVALUES              CSL05670
C              SUPPLIED.                                                CSL05680
C                                                                       CSL05690
C     CMTQL1 = COMPUTES EIGENVALUES OF THE SPECIFIED T-MATRIX USING     CSL05700
C              A REAL ORTHOGONAL ANALOG OF THE QL ALGORITHM IMTQL1      CSL05710
C              IN EISPACK.                                              CSL05720
C                                                                       CSL05730
C     INVERR = USES INVERSE ITERATION ON T-MATRICES TO COMPUTE ERROR    CSL05740
C              ESTIMATES ON COMPUTED T-EIGENVALUES. (USES GENRAN)       CSL05750
C                                                                       CSL05760
C     LUMP = 'COMBINES' EIGENVALUES OF T-MATRIX USING THE RELATIVE      CSL05770
C            TOLERANCE RELTOL.                                          CSL05780
C                                                                       CSL05790
C     ISOEV = CALCULATES GAPS BETWEEN DISTINCT EIGENVALUES OF T-MATRIX  CSL05800
C             AND THEN USES THESE GAPS TO LABEL THOSE 'GOOD'            CSL05810
C             T-EIGENVALUES FOR WHICH ERROR ESTIMATES ARE NOT COMPUTED. CSL05820
C                                                                       CSL05830
C     TNORM = COMPUTES THE SCALE TKMAX USED IN CHECKING                 CSL05840
C             FOR LOCAL ORTHOGONALITY OF THE LANCZOS VECTORS            CSL05850
C             BY TESTING THE RELATIVE SIZE OF THE BETAS USING           CSL05860
C             THE RELATIVE TOLERANCE BTOL.                              CSL05870
C                                                                       CSL05880
C    CINPRD = COMPUTES THE HERMITIAN INNER PRODUCT OF TWO               CSL05890
C             COMPLEX*16 VECTORS, USED IN SUBROUTINE INVERR             CSL05900
C             AND IN THE MAIN PROGRAM.                                  CSL05910
C                                                                       CSL05920
C    INPRDC = COMPUTES THE EUCLIDEAN INNER PRODUCT OF TWO               CSL05930
C             COMPLEX*16 VECTORS. USED IN SUBROUTINE LANCZS.            CSL05940
C                                                                       CSL05950
C                                                                       CSL05960
C-----OTHER PROGRAMS SUPPLIED-------------------------------------------CSL05970
C                                                                       CSL05980
C                                                                       CSL05990
C     LCCOMPAC = PROGRAM TO TRANSLATE A SPARSE, COMPLEX SYMMETRIC       CSL06000
C                MATRIX GIVEN AS I, J, A(I,J), INTO THE SPARSE MATRIX   CSL06010
```

```
C                FORMAT USED IN THE SAMPLE USPEC AND CMATV SUBROUTINES  CSL06020
C                PROVIDED.  PROGRAM ASSUMES THAT THE MATRIX ENTRIES     CSL06030
C                ARE GIVEN EITHER COLUMN BY COLUMN OR ROW BY ROW.       CSL06040
C                                                                       CSL06050
C                                                                       CSL06060
C-----COMMENTS ON THE STORAGE REQUIRED FOR EIGENVALUE COMPUTATIONS------CSL06070
C                                                                       CSL06080
C     THE ARRAYS USED IN THIS EIGENVALUE PROGRAM USE THE EQUIVALENT OF  CSL06090
C     ONE REAL*8 ARRAY OF DIMENSION                                     CSL06100
C                                                                       CSL06110
C        8*KMAX + 4*MAX(KMAX,N)                                         CSL06120
C                                                                       CSL06130
C     PLUS WHATEVER IS NEEDED TO GENERATE A*X FOR THE GIVEN MATRIX A.   CSL06140
C     THE ARRAYS ALPHA, BETA, VS, G, GG, MP, AND MP2 CONSUME            CSL06150
C     8*KMAX*8 BYTES.  THE ARRAYS V1 AND V2 CONSUME                     CSL06160
C     4*MAXIMUM(KMAX,N)*8 BYTES.                                        CSL06170
C                                                                       CSL06180
C                                                                       CSL06190
C-----------------------------------------------------------------------CSL06200
C                                                                       CSL06210
C     COMMENTS FOR EIGENVECTOR COMPUTATIONS                             CSL06220
C                                                                       CSL06230
C-----------------------------------------------------------------------CSL06240
C                                                                       CSL06250
C                                                                       CSL06260
C     THE EIGENVALUES WHOSE EIGENVECTORS ARE TO BE COMPUTED MUST        CSL06270
C     HAVE BEEN COMPUTED USING THE CORRESPONDING LANCZOS EIGENVALUE     CSL06280
C     FILES:  CSLEVAL + CSLESUB + CSLEMULT, FOR COMPLEX SYMMETRIC       CSL06290
C     MATRICES BECAUSE THE EIGENVECTOR PROGRAMS WILL USE THE SAME       CSL06300
C     FAMILY OF LANCZOS TRIDIAGONAL MATRICES AND LANCZOS VECTORS        CSL06310
C     THAT WAS USED IN THE EIGENVALUE COMPUTATIONS.                     CSL06320
C                                                                       CSL06330
C     THESE PROGRAMS ASSUME THAT THE EIGENVALUES SUPPLIED TO IT          CSL06340
C     HAVE BEEN COMPUTED ACCURATELY, AS MEASURED BY THE                 CSL06350
C     ERROR ESTIMATES COMPUTED IN THE CORRESPONDING LANCZOS             CSL06360
C     EIGENVALUE COMPUTATIONS, ALTHOUGH THESE ESTIMATES ARE             CSL06370
C     TYPICALLY CONSERVATIVE.  THE EIGENVALUES OF INTEREST              CSL06380
C     ARE IN THE ARRAY GOODEV(J), J=1,NGOOD.                            CSL06390
C                                                                       CSL06400
C     FOR EACH GOODEV(J), AN INITIAL ESTIMATE IS MADE OF AN             CSL06410
C     APPROPRIATE ORDER, MA(J), J=1,NGOOD, FOR A LANCZOS TRIDIAGONAL    CSL06420
C     FOR THE JTH EIGENVECTOR COMPUTATION.  THEN FOR EACH J,            CSL06430
C     SUBROUTINE INVERM SUCCESSIVELY COMPUTES CORRESPONDING             CSL06440
C     EIGENVECTORS OF ENLARGED T-MATRICES UNTIL A SUITABLE              CSL06450
C     SIZE T-MATRIX IS DETERMINED FOR EACH J.  UP TO 10 SUCH            CSL06460
C     EIGENVECTOR COMPUTATIONS ARE ALLOWED FOR EACH EIGENVALUE.         CSL06470
C                                                                       CSL06480
C     ONCE SUITABLE T-EIGENVECTORS HAVE BEEN OBTAINED THEN THE          CSL06490
C     RITZ VECTOR CORRESPONDING TO THESE T-EIGENVECTORS ARE             CSL06500
C     COMPUTED AND TAKEN AS APPROXIMATE EIGENVECTORS OF A FOR THE       CSL06510
C     GIVEN EIGENVALUES, GOODEV(J), J = 1, ..., NGOOD.                  CSL06520
C                                                                       CSL06530
C     THIS IMPLEMENTATION FIRST COMPUTES ALL OF THE RELEVANT            CSL06540
C     EIGENVECTORS OF THE COMPLEX SYMMETRIC TRIDIAGONAL MATRICES        CSL06550
C     IN THE VECTOR, TVEC.                                              CSL06560
```

```
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
C     LANCZOS VECTOR.  LANCZOS VECTORS ARE GENERATED (NOTE
C     THAT THEY ARE NOT BEING KEPT), UNTIL ENOUGH HAVE
C     BEEN GENERATED TO MAP THE LONGEST T-EIGENVECTOR INTO ITS
C     CORRESPONDING RITZ VECTOR.  THE ARRAY RITVEC CONTAINS THE
C     SUCCESSIVE RITZ VECTORS WHICH ARE THE APPROXIMATE
C     EIGENVECTORS OF A.
C
C
C-----PARAMETER CONTROLS FOR EIGENVECTOR PROGRAMS-----------------------
C
C
C     PARAMETER CONTROLS ARE INTRODUCED TO ALLOW SEGMENTATION OF THE
C     EIGENVECTOR COMPUTATIONS AND TO ALLOW VARIOUS COMBINATIONS OF
C     READ/WRITES.
C
C     THE FLAG MBOUND ALLOWS THE USER TO DETERMINE A FIRST GUESS ON THE
C     STORAGE THAT WILL BE REQUIRED BY THE T-EIGENVECTORS FOR THE
C     EIGENVALUES WHOSE EIGENVECTORS ARE TO BE COMPUTED.
C     THIS CAN BE USED TO ESTIMATE THE REQUIRED SIZE OF THE TVEC ARRAY.
C
C     MBOUND = (0,1) MEANS
C
C              (0)  PROGRAM COMPUTES FIRST GUESSES AT THE SIZES
C                   OF THE T-MATRICES REQUIRED BY EACH OF THE
C                   EIGENVALUES SUPPLIED AND THEN CONTINUES WITH
C                   THE CORRESPONDING T-EIGENVECTOR COMPUTATIONS.
C
C              (1)  PROGRAM COMPUTES FIRST GUESSES AT THE SIZES
C                   OF THE T-MATRICES REQUIRED BY EACH OF THE
C                   EIGENVALUES SUPPLIED, STORES THESE IN FILE 10
C                   AND THEN TERMINATES.  THE USER CAN USE THESE
C                   SIZES TO ESTIMATE THE SIZE TVEC ARRAY NEEDED
C                   FOR THE DESIRED T-EIGENVECTOR COMPUTATIONS.
C
C     THE FLAGS NTVCON, TVSTOP, LVCONT, AND ERCONT CONTROL THE STOPPING
C     CRITIERIA FOR INTERMEDIATE POINTS IN THE LANCZOS PROCEDURE.
C     THEY CAUSE TERMINATION OF THE LANCZOS PROCEDURE IF VARIOUS
C     QUANTITIES CANNOT BE COMPUTED AS DESIRED.
C
C     NTVCON = (0,1) MEANS
C
C              (0)  IF THE ESTIMATED STORAGE FOR THE T-EIGENVECTORS
C                   EXCEEDS THE USER-SPECIFIED DIMENSION OF THE
C                   TVEC ARRAY PROGRAM DOES NOT CONTINUE WITH THE
C                   T-EIGENVECTOR COMPUTATIONS. TERMINATION OCCURS.
C
C              (1)  CONTINUE WITH THE T-EIGENVECTOR COMPUTATIONS
C                   EVEN IF THE ESTIMATED STORAGE FOR TVEC EXCEEDS
C                   THE USER-SPECIFIED DIMENSION OF THE TVEC ARRAY.
C                   IN THIS SITUATION THE PROGRAM COMPUTES AS MANY
C                   T-EIGENVECTORS AS IT HAS ROOM FOR, IN THE SAME
```

```
C                    ORDER IN WHICH THE EIGENVALUES ARE PROVIDED.        CSL07120
C                                                                        CSL07130
C      SVTVEC = (0,1) MEANS                                              CSL07140
C                                                                        CSL07150
C               (0)  DO NOT STORE THE COMPUTED T-EIGENVECTORS ON         CSL07160
C                    FILE 11 UNLESS ALSO HAVE THE FLAG TVSTOP = 1,       CSL07170
C                    IN WHICH CASE THE T-EIGENVECTORS ARE ALWAYS         CSL07180
C                    WRITTEN TO FILE 11.                                 CSL07190
C                                                                        CSL07200
C               (1)  STORE THE COMPUTED T-EIGENVECTORS ON FILE 11.       CSL07210
C                                                                        CSL07220
C      TVSTOP = (0,1) MEANS                                              CSL07230
C                                                                        CSL07240
C               (0)  ATTEMPT TO CONTINUE ON TO THE COMPUTATION           CSL07250
C                    OF THE RITZVECTORS AFTER COMPLETING THE             CSL07260
C                    COMPUTATION OF THE T-EIGENVECTORS.                  CSL07270
C                                                                        CSL07280
C               (1)  TERMINATE AFTER COMPUTING THE                       CSL07290
C                    T-EIGENVECTORS AND STORING THEM ON FILE 11.         CSL07300
C                                                                        CSL07310
C      LVCONT = (0,1) MEANS                                              CSL07320
C                                                                        CSL07330
C               (0)  IF SOME OF THE T-EIGENVECTORS THAT WERE             CSL07340
C                    REQUIRED WERE NOT COMPUTED, EXIT                    CSL07350
C                    FROM THE PROGRAM WITHOUT COMPUTING THE              CSL07360
C                    CORRESPONDING RITZ VECTORS.                         CSL07370
C                                                                        CSL07380
C               (1)  CONTINUE ON TO THE RITZ VECTOR COMPUTATIONS         CSL07390
C                    EVEN IF NOT ALL OF THE T-EIGENVECTORS THAT          CSL07400
C                    WERE REQUESTED WERE COMPUTED.                       CSL07410
C                                                                        CSL07420
C      ERCONT = (0,1) MEANS                                              CSL07430
C                                                                        CSL07440
C               (0)  PROGRAM WILL NOT COMPUTE THE RITZ                   CSL07450
C                    VECTOR FOR ANY EIGENVALUE FOR WHICH NO              CSL07460
C                    T-EIGENVECTOR WHICH SATISFIES THE ERROR ESTIMATE    CSL07470
C                    TEST (ERTOL) HAS BEEN IDENTIFIED.                   CSL07480
C                                                                        CSL07490
C               (1)  A RITZ VECTOR WILL BE COMPUTED FOR EVERY            CSL07500
C                    EIGENVALUE FOR WHICH A T-EIGENVECTOR HAS BEEN       CSL07510
C                    COMPUTED REGARDLESS OF WHETHER OR NOT THAT          CSL07520
C                    T-EIGENVECTOR SATISFIED THE ERROR ESTIMATE TEST.    CSL07530
C                                                                        CSL07540
C                                                                        CSL07550
C-----INPUT/OUTPUT FILES FOR THE EIGENVECTOR COMPUTATIONS---------------CSL07560
C                                                                        CSL07570
C                                                                        CSL07580
C     INPUT DATA OTHER THAN THE T-MATRIX HISTORY FILE AND THE            CSL07590
C     EIGENVALUES AND ERROR ESTIMATES SUPPLIED SHOULD BE STORED ON       CSL07600
C     FILE 5 IN FREE FORMAT.  SEE SAMPLE INPUT/OUTPUT FOR TYPICAL        CSL07610
C     INPUT/OUTPUT FILE.                                                 CSL07620
C                                                                        CSL07630
C     FILE 6 WAS USED AS THE INTERACTIVE TERMINAL OUTPUT FILE.           CSL07640
C     THIS FILE PROVIDES A RUNNING ACCOUNT OF THE PROGRESS OF THE        CSL07650
C     COMPUTATIONS.  ADDITIONAL PRINTOUT IS GENERATED WHEN               CSL07660
```

```
C     THE FLAG IWRITE = 1.                                              CSL07670
C                                                                       CSL07680
C                                                                       CSL07690
C DESCRIPTION OF OTHER I/O FILES                                        CSL07700
C                                                                       CSL07710
C FILE (K)     CONTAINS:                                                CSL07720
C                                                                       CSL07730
C      (2)     INPUT FILE:                                              CSL07740
C              PREVIOUSLY-GENERATED T-MATRICES (ALPHA/BETA ARRAYS)      CSL07750
C              AND THE FINAL TWO LANCZOS VECTORS USED ON THAT           CSL07760
C              COMPUTATION.  THIS PROGRAM ALLOWS ENLARGEMENT            CSL07770
C              OF ANY T-MATRICES PROVIDED ON FILE 2.                    CSL07780
C                                                                       CSL07790
C      (3)     INPUT FILE:                                              CSL07800
C              THE GOOD EIGENVALUES OF THE T-MATRIX  T(1,MEV)           CSL07810
C              FOR WHICH EIGENVECTORS ARE REQUESTED.                    CSL07820
C              FILE 3 ALSO CONTAINS THE T-MULTIPLICITIES OF THESE       CSL07830
C              EIGENVALUES AND THEIR COMPUTED GAPS IN THE               CSL07840
C              T-MATRICES AND IN THE USER-SUPPLIED MATRIX. THIS         CSL07850
C              FILE IS CREATED IN THE LANCZOS EIGENVALUE COMPUTATIONS.  CSL07860
C                                                                       CSL07870
C      (4)     INPUT FILE:                                              CSL07880
C              ERROR ESTIMATES FOR THE ISOLATED GOOD T-EIGENVALUES      CSL07890
C              IN FILE 3.  THIS FILE IS CREATED DURING THE LANCZOS      CSL07900
C              EIGENVALUE COMPUTATIONS.                                 CSL07910
C                                                                       CSL07920
C      (8)     INPUT FILE:                                              CSL07930
C              SAMPLE USPEC SUBROUTINE ASSUMES THAT THE ARRAYS          CSL07940
C              REQUIRED TO SPECIFY THE USER'S-MATRIX ARE STORED ON      CSL07950
C              FILE 8.  USERS MUST MAKE WHATEVER DEFINITIONS ARE        CSL07960
C              APPROPRIATE FOR THEIR MATRICES.                          CSL07970
C                                                                       CSL07980
C      (9)     OUTPUT FILE:                                             CSL07990
C              ERROR ESTIMATES FOR THE COMPUTED RITZ VECTORS CONSIDERED CSL08000
C              AS EIGENVECTORS OF THE ORIGINAL MATRIX. THESE ESTIMATES  CSL08010
C              ARE OF THE FORM                                          CSL08020
C                    AERROR = || A*RITVEC - EVAL*RITVEC ||              CSL08030
C              WHERE A DENOTES THE USER-SUPPLIED MATRIX, EVAL DENOTES   CSL08040
C              THE EIGENVALUE BEING CONSIDERED AND RITVEC DENOTES       CSL08050
C              THE COMPUTED RITZ VECTOR.                                CSL08060
C                                                                       CSL08070
C      (10)    OUTPUT FILE:                                             CSL08080
C              GUESSES AT APPROPRIATE SIZE T-MATRICES FOR THE           CSL08090
C              T-EIGENVECTORS FOR EACH SUPPLIED EIGENVALUE GOODEV(J).   CSL08100
C                                                                       CSL08110
C      (11)    OUTPUT FILE:                                             CSL08120
C              COMPUTED T-EIGENVECTORS CORRESPONDING TO EIGENVALUES     CSL08130
C              IN THE GOODEV ARRAY.  NOTE THAT IT IS POSSIBLE IN        CSL08140
C              CERTAIN SITUATIONS THAT FOR SOME EIGENVALUES IN THE      CSL08150
C              GOODEV ARRAY A T-EIGENVECTOR WILL NOT BE COMPUTED.       CSL08160
C              (WRITTEN ONLY IF FLAG SVTVEC = 1).                       CSL08170
C                                                                       CSL08180
C      (12)    OUTPUT FILE:                                             CSL08190
C              CONTAINS COMPUTED RITZ VECTORS CORRESPONDING TO          CSL08200
C              THE T-EIGENVECTORS ON FILE 11. NOTE THAT IN              CSL08210
```

```
C              SOME SITUATIONS THAT FOR SOME EIGENVALUES IN             CSL08220
C              THE GOODEV ARRAY FOR WHICH T-EIGENVECTORS HAVE           CSL08230
C              BEEN COMPUTED NO RITZ VECTOR WILL HAVE BEEN              CSL08240
C              COMPUTED.                                                CSL08250
C                                                                       CSL08260
C       (13)   OUTPUT FILE:                                             CSL08270
C              ADDITIONAL INFORMATION ABOUT THE BOUNDS AND ERROR        CSL08280
C              ESTIMATES OBTAINED.                                      CSL08290
C                                                                       CSL08300
C                                                                       CSL08310
C-----SEEDS FOR EIGENVECTOR PROGRAMS------------------------------------CSL08320
C                                                                       CSL08330
C     SEEDS FOR RANDOM NUMBER GENERATOR GENRAN                          CSL08340
C                (1) SVSEED = INTEGER*4 SCALAR USED IN THE SUBROUTINE   CSL08350
C                             GENRAN TO GENERATE THE STARTING VECTOR FORCSL08360
C                             THE REGENERATION OF THE LANCZOS VECTORS.  CSL08370
C                                                                       CSL08380
C                (2) RHSEED = INTEGER*4 SCALAR USED IN THE SUBROUTINE   CSL08390
C                             GENRAN TO GENERATE A RANDOM VECTOR FOR    CSL08400
C                             USE IN SUBROUTINE INVERM.                 CSL08410
C                                                                       CSL08420
C     USER SHOULD NOTE THAT SVSEED MUST BE THE SAME SEED THAT           CSL08430
C     WAS USED TO GENERATE THE T-MATRICES THAT WERE USED TO             CSL08440
C     COMPUTE THE EIGENVALUES WHOSE EIGENVECTORS ARE TO BE COMPUTED.    CSL08450
C     SVSEED IS READ IN FROM FILE 3.                                    CSL08460
C                                                                       CSL08470
C                                                                       CSL08480
C-----USER-SPECIFIED PARAMETERS FOR THE EIGENVECTOR PROGRAMS------------CSL08490
C                                                                       CSL08500
C     NGOOD   = NUMBER OF EIGENVALUES READ INTO THE GOODEV ARRAY        CSL08510
C               READ FROM FILE 3.                                       CSL08520
C                                                                       CSL08530
C     N       = SIZE OF THE USER-SUPPLIED MATRIX.                       CSL08540
C                                                                       CSL08550
C     MEV     = SIZE OF THE T-MATRIX THAT WAS USED TO COMPUTE           CSL08560
C               THE EIGENVALUES WHOSE EIGENVECTORS ARE REQUESTED.       CSL08570
C               MEV IS READ IN FROM FILE 3.                             CSL08580
C                                                                       CSL08590
C     KMAX    =  SIZE OF THE T-MATRIX PROVIDED ON FILE 2.               CSL08600
C                                                                       CSL08610
C     MDIMTV  = MAXIMUM CUMULATIVE SIZE OF THE TVEC ARRAY ALLOWED       CSL08620
C               FOR ALL OF THE T-EIGENVECTORS REQUIRED.  MDIMTV         CSL08630
C               MUST NOT EXCEED THE USER-SPECIFIED DIMENSION OF         CSL08640
C               THE TVEC ARRAY.  PROGRAM CAN BE RUN WITH THE FLAG       CSL08650
C               MBOUND = 1 TO DETERMINE AN EDUCATED GUESS ON AN         CSL08660
C               APPROPRIATE DIMENSION FOR THE TVEC ARRAY.               CSL08670
C                                                                       CSL08680
C     MDIMRV = MAXIMUM CUMULATIVE SIZE OF THE RITVEC ARRAY ALLOWED      CSL08690
C              FOR ALL OF THE RITZ VECTORS TO BE COMPUTED. MDIMRV       CSL08700
C              MUST NOT EXCEED THE USER-SPECIFIED DIMENSION OF          CSL08710
C              THE RITVEC ARRAY.  MUST BE SELECTED SO THAT              CSL08720
C              THERE IS ENOUGH ROOM FOR A RITZ VECTOR FOR EVERY         CSL08730
C              GOODEV(J) READ INTO PROGRAM.  (>= NGOOD*N)               CSL08740
C                                                                       CSL08750
C                                                                       CSL08760
```

```
C-----ARRAYS REQUIRED BY THE EIGENVECTOR PROGRAMS-----------------------CSL08770
C                                                                       CSL08780
C                                                                       CSL08790
C     ALPHA(J) = COMPLEX*16 ARRAY WHOSE DIMENSION MUST BE AT LEAST      CSL08800
C                KMAXN, THE LARGEST SIZE T-MATRIX CONSIDERED BY         CSL08810
C                THE PROGRAM.  NOTE THAT KMAXN IS THE LARGER OF         CSL08820
C                THE SIZE OF THE ALPHA, BETA HISTORY PROVIDED           CSL08830
C                ON FILE 2 (IF ANY ) AND THE SIZE WHICH THE PROGRAM     CSL08840
C                SPECIFIES INTERNALLY, THIS LATTER IS ALWAYS            CSL08850
C                < = 11*MEV / 8  +  12, WHERE MEV IS THE SIZE           CSL08860
C                T-MATRIX THAT WAS USED IN THE CORRESPONDING EIGENVALUE CSL08870
C                COMPUTATIONS.  ALPHA CONTAINS THE DIAGONAL ENTRIES     CSL08880
C                OF THE LANCZOS T-MATRICES.  ALPHA IS NOT DESTROYED     CSL08890
C                IN THE COMPUTATIONS.                                   CSL08900
C                                                                       CSL08910
C     BETA(J) = COMPLEX*16 ARRAY WHOSE DIMENSION MUST BE AT LEAST 1     CSL08920
C               MORE THAN THAT OF ALPHA.  DIMENSION COMMENTS ABOVE      CSL08930
C               ABOUT ALPHA APPLY ALSO TO THE BETA ARRAY.  BETA         CSL08940
C               CONTAINS THE SUBDIAGONAL ENTRIES OF THE T-MATRICES.     CSL08950
C               BETA IS NOT DESTROYED IN THE COMPUTATIONS.              CSL08960
C                                                                       CSL08970
C     RITVEC(J) = COMPLEX*16 ARRAY WHOSE DIMENSION MUST BE AT LEAST     CSL08980
C                 NGOOD*N WHERE N IS THE ORDER OF THE USER-SUPPLIED     CSL08990
C                 MATRIX AND NGOOD IS THE NUMBER OF EIGENVALUES         CSL09000
C                 WHOSE EIGENVECTORS ARE TO BE COMPUTED.  IT CONTAINS   CSL09010
C                 THE COMPUTED RITZ VECTORS  (THE APPROXIMATE           CSL09020
C                 EIGENVECTORS OF A).  THESE VECTORS ARE STORED         CSL09030
C                 ON FILE 12.                                           CSL09040
C                                                                       CSL09050
C     TVEC(J)   = COMPLEX*16 ARRAY WHOSE DIMENSION MUST BE AT LEAST     CSL09060
C                 MTOL = |MA(1)| + |MA(2)| + ... + |MA(NGOOD)|          CSL09070
C                 WHERE NGOOD IS THE NUMBER OF EIGENVALUES BEING        CSL09080
C                 CONSIDERED AND |MA(J)| IS THE SIZE OF THE             CSL09090
C                 T-MATRIX BEING USED IN THE RITZ VECTOR COMPUTATIONS   CSL09100
C                 FOR GOODEV(J).  THESE SIZES ARE DETERMINED BY THE     CSL09110
C                 PROGRAM.  AN ESTIMATE OF MTOL CAN BE OBTAINED BY      CSL09120
C                 SETTING MBOUND = 1, RUNNING THE PROGRAM, AND          CSL09130
C                 MULTIPLYING THE RESULTING TOTAL T-SIZES BY 5/4.       CSL09140
C                 THE ARRAY TVEC IS USED TO HOLD THE COMPUTED           CSL09150
C                 T-EIGENVECTORS.  IF THE FLAG SVTVEC = 1 OR THE        CSL09160
C                 FLAG TVSTOP = 1, THESE VECTORS ARE SAVED ON FILE 11.  CSL09170
C                                                                       CSL09180
C     V1(J)     = COMPLEX*16 ARRAY WHOSE DIMENSION MUST BE AT LEAST     CSL09190
C                 MAX(KMAX,N)  WHERE KMAX IS THE                        CSL09200
C                 LARGEST SIZE T-MATRIX THAT CAN BE CONSIDERED          CSL09210
C                 IN THE T-EIGENVECTOR COMPUTATIONS.  V1 IS USED        CSL09220
C                 IN THE SUBROUTINE INVERM AND IN THE REGENERATION      CSL09230
C                 OF THE LANCZOS VECTORS.                               CSL09240
C                                                                       CSL09250
C     V2(J)     = COMPLEX*16 ARRAY WHOSE DIMENSION MUST BE AT LEAST     CSL09260
C                 MAX(KMAX,N). IT IS USED IN THE REGENERATION OF        CSL09270
C                 THE LANCZOS VECTORS AND IN THE SUBROUTINE INVERM.     CSL09280
C                                                                       CSL09290
C     GOODEV(J) = COMPLEX*16 ARRAY OF DIMENSION AT LEAST NGOOD.         CSL09300
C                 CONTAINS THE EIGENVALUES FOR WHICH EIGENVECTORS       CSL09310
```

```
C               ARE REQUESTED. THESE EIGENVALUES ARE READ IN
C               FROM FILE 3.
C
C     GR(J),GC(J)     = REAL*8 ARRAYS WHOSE DIMENSION MUST BE AT
C                       LEAST MAX(N,KMAX).  USED TO HOLD RANDOMLY-
C                       GENERATED STARTING VECTORS FOR LANCZS
C                       COMPUTATIONS AND FOR THE INVERM SUBROUTINE.
C
C     AMINGP(J), = REAL*4 ARRAYS OF DIMENSION AT LEAST NGOOD.
C     TMINGP(J)    CONTAIN, RESPECTIVELY, THE MINIMAL GAPS FOR
C                  CORRESPONDING EIGENVALUES IN GOODEV ARRAY IN
C                  A-MATRIX AND IN T-MATRIX.
C
C     TERR(J), ERR(J),     = REAL*4 ARRAYS (EXCEPT TLAST WHICH IS
C     ERRDGP(J), TLAST(J)    REAL*8) EACH OF WHOSE DIMENSIONS MUST BE
C     RNORM(J), TBETA(J)     AT LEAST NGOOD.  USED TO STORE QUANTITIES
C                            GENERATED DURING THE COMPUTATIONS FOR
C                            LATER PRINTOUT.
C
C     G(J)     = REAL*4 ARRAY WHOSE DIMENSION MUST BE AT LEAST
C                MAX(KMAX,N).  USED IN SUBROUTINE GENRAN TO HOLD
C                RANDOM NUMBERS NEEDED FOR THE LANCZOS VECTORS
C                REGENERATION AND FOR THE INVERSE ITERATION
C                COMPUTATIONS IN THE SUBROUTINE INVERM.
C
C     MP(J) = INTEGER*4 ARRAY WHOSE DIMENSION IS AT LEAST NGOOD.
C             INITIALLY CONTAINS THE T-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     INTERC(J)  = INTEGER*4 ARRAY WHOSE DIMENSION MUST BE AT
C                  LEAST KMAX.  WORK SPACE USED IN INVERM.
C
C-----SUBROUTINES INCLUDED FOR THE EIGENVECTOR COMPUTATIONS-------------
C
C
C     INVERM = FOR THE T-SIZES CONSIDERED BY THE PROGRAM COMPUTES
C              THE CORRESPONDING EIGENVECTORS OF THESE T-MATRICES
```

```
C             CORRESPONDING TO THE USER-SUPPLIED EIGENVALUES IN        CSL09870
C             THE GOODEV ARRAY.                                        CSL09880
C                                                                      CSL09890
C     LANCZS, TNORM , CINPRD, INPRDC, CMATV AND GENRAN ARE USED        CSL09900
C     HERE AS WELL AS IN THE EIGENVALUE COMPUTATIONS.                  CSL09910
C                                                                      CSL09920
C                                                                      CSL09930
C----------------------------------------------------------------------CSL09940
```

7.3 CSLEVAL: Main Program, Eigenvalue Computations

```
C-----CSLEVAL (EIGENVALUES OF COMPLEX SYMMETRIC MATRICES)---------------CSL00010
C  Authors:  Jane Cullum and Ralph A. Willoughby (Deceased)             CSL00020
C            Los Alamos National Laboratory                             CSL00030
C            Los Alamos, New Mexico 87544                               CSL00040
C                                                                       CSL00050
C            E-mail:  cullumj@lanl.gov                                  CSL00060
C                                                                       CSL00070
C  These codes are copyrighted by the authors.  These codes             CSL00080
C  and modifications of them or portions of them are NOT to be          CSL00090
C  incorporated into any commercial codes or used for any other         CSL00100
C  commercial purposes such as consulting for other companies,          CSL00110
C  without legal agreements with the authors of these Codes.            CSL00120
C  If these Codes or portions of them                                   CSL00130
C  are used in other scientific or engineering research works           CSL00140
C  the names of the authors of these codes and appropriate              CSL00150
C  references to their written work are to be incorporated in the       CSL00160
C  derivative works.                                                    CSL00170
C                                                                       CSL00180
C  This header is not to be removed from these codes.                   CSL00190
C                                                                       CSL00200
C         REFERENCE:  Cullum and Willoughby, Chapter 6,                 CSL00201
C         Lanczos Algorithms for Large Symmetric Eigenvalue ComputationsCSL00202
C         VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in   CSL00203
C         Applied Mathematics, 2002. SIAM Publications,                 CSL00204
C         Philadelphia, PA. USA                                         CSL00205
C                                                                       CSL00206
C                                                                       CSL00207
C     CONTAINS MAIN PROGRAM FOR COMPUTING DISTINCT EIGENVALUES OF       CSL00210
C     A NONDEFECTIVE COMPLEX SYMMETRIC MATRIX USING LANCZOS             CSL00220
C     TRIDIAGONALIZATION WITHOUT REORTHOGONALIZATION                    CSL00230
C                                                                       CSL00240
C     PORTABILITY:                                                      CSL00250
C     THESE PROGRAMS ARE NOT PORTABLE DUE TO THE USE OF COMPLEX*16      CSL00260
C     VARIABLES AND CORRESPONDING COMPLEX FUNCTIONS SUCH AS DCMPLX      CSL00270
C     AND CDABS.  FURTHERMORE, OTHER NONPORTABLE CONSTRUCTIONS          CSL00280
C     IDENTIFIED BY THE PFORT VERIFIER ARE THE FOLLOWING:               CSL00290
C                                                                       CSL00300
C     1.  DATA/MACHEP/ STATEMENT THAT DEFINES MACHINE EPSILON           CSL00310
C     2.  ALL READ(5,*) INPUT STATEMENTS IN FREE FORMAT                 CSL00320
C     3.  FORMAT(20A4) USED WITH EXPLANATORY HEADER EXPLAN.             CSL00330
C     4.  HEXADECIMAL FORMAT (4Z20) USED WITH ALPHA/BETA FILES 1 AND 2. CSL00340
C                                                                       CSL00350
C-----------------------------------------------------------------------CSL00360
C                                                                       CSL00370
      COMPLEX*16  ALPHA(3000),BETA(3000),VS(3000)                       CSL00380
      COMPLEX*16  V1(3000),V2(3000),ZEROC,BETAM,Z                       CSL00390
      DOUBLE PRECISION  GR(3000),GC(3000)                               CSL00400
      DOUBLE PRECISION  BTOL,GAPTOL,TTOL,MACHEP,EPSM,RELTOL             CSL00410
      DOUBLE PRECISION  SCALE1,SCALE2,SPUTOL,CONTOL,MULTOL,EVMAX        CSL00420
      DOUBLE PRECISION  ONE,ZERO,TEMP,TKMAX,BKMIN,T0,T1                 CSL00430
      REAL  G(3000),GG(3000),EXPLAN(20),GTEMP                           CSL00440
```

```
      INTEGER  MP(3000),MP2(3000),NMEV(20)                              CSL00450
      INTEGER  SVSEED,RHSEED,SVSOLD,SAVTEV                              CSL00460
      INTEGER IABS                                                      CSL00470
      REAL  ABS                                                         CSL00480
      DOUBLE PRECISION  DABS, DFLOAT                                    CSL00490
      EXTERNAL CMATV                                                    CSL00500
C                                                                       CSL00510
C-----------------------------------------------------------------------CSL00520
      DATA MACHEP/Z3410000000000000/                                    CSL00530
      EPSM = 2.0D0*MACHEP                                               CSL00540
C-----------------------------------------------------------------------CSL00550
C                                                                       CSL00560
C     ARRAYS MUST BE DIMENSIONED AS FOLLOWS:                            CSL00570
C          1. ALPHA AND VS:  >= KMAX.   BETA:  >= (KMAX+1)              CSL00580
C          2. V1, V2, GR, GC:  >= MAX(N,KMAX)                           CSL00590
C          3. G:  >= MAX(N,KMAX).   GG:  >= KMAX.                       CSL00600
C          4. MP, MP2:   >= KMAX                                        CSL00610
C          5. NMEV:  >= NUMBER OF T-MATRICES ALLOWED                    CSL00620
C          6. EXPLAN:  DIMENSION IS 20.                                 CSL00630
C                                                                       CSL00640
C     NOTE:   THE OBSERVED ACHIEVABLE ACCURACY FOR THE COMPLEX          CSL00650
C     SYMMETRIC MATRICES TESTED WAS SIGNIFICANTLY LESS THAN THAT        CSL00660
C     OBTAINED WITH THE REAL SYMMETRIC AND HERMITIAN VERSIONS           CSL00670
C     OF THESE LANCZOS CODES AND IT IS DOUBTFUL THAT THIS CODE          CSL00680
C     CAN HANDLE VERY STIFF COMPLEX SYMMETRIC MATRICES.                 CSL00690
C                                                                       CSL00700
C     IMPORTANT TOLERANCES OR SCALES THAT ARE USED REPEATEDLY           CSL00710
C     THROUGHOUT THE PROGRAM ARE THE FOLLOWING:                         CSL00720
C     SCALED MACHINE EPSILON:  TTOL = EVMAX*EPSM WHERE                  CSL00730
C     EPSM = 2*MACHINE EPSILON AND                                      CSL00740
C     EVMAX = MAX(|LAMBDA(J)|), J =1,MEV OF EIGENVALUES OF T(1,MEV).    CSL00750
C     TOLERANCE: T-MULTIPLICITY TESTS:  MULTOL = 500*(1000+MEV)*TTOL    CSL00760
C     TOLERANCE: SPURIOUS TESTS   SPUTOL = MULTOL                       CSL00770
C     NOTE THAT IN THE MAIN PROGRAM THESE TOLERANCES ARE INITIALIZED    CSL00780
C     TO QUANTITIES THAT ARE NOT A FUNCTION OF THE SIZE OF THE          CSL00790
C     T-EIGENVALUES AND THEN THE SIZES OF THE T-EIGENVALUES ARE         CSL00800
C     INTRODUCED IN THE SUBROUTINE COMPEV.                              CSL00810
C                                                                       CSL00820
C     LANCZOS CONVERGENCE TOLERANCE: CONTOL = CDABS(BETA(MEV+1)*1.D-10  CSL00830
C-----------------------------------------------------------------------CSL00840
C     OUTPUT HEADER                                                     CSL00850
      WRITE(6,10)                                                       CSL00860
   10 FORMAT(/' LANCZOS EIGENVALUE PROCEDURE FOR COMPLEX SYMMETRIC MATRICSL00870
     1CES'/)                                                            CSL00880
C                                                                       CSL00890
C     SET PROGRAM PARAMETERS                                            CSL00900
C     SCALEK ARE USED IN TOLERANCES NEEDED IN SUBROUTINES LUMP          CSL00910
C     AND ISOEV.  USER MUST NOT MODIFY THESE SCALES.                    CSL00920
      SCALE1 = 5.0D2                                                    CSL00930
      SCALE2 = 5.0D0                                                    CSL00940
      ONE  = 1.0D0                                                      CSL00950
      ZERO = 0.0D0                                                      CSL00960
      ZEROC = DCMPLX(ZERO,ZERO)                                         CSL00970
      BTOL = 1.0D-8                                                     CSL00980
C     BTOL = MACHEP                                                     CSL00990
```

```
      GAPTOL = 1.0D-7                                                   CSL01000
      ICONV = 0                                                         CSL01010
      MOLD = 0                                                          CSL01020
      MOLD1 = 1                                                         CSL01030
      MMB = 0                                                           CSL01040
C                                                                       CSL01050
C     READ USER-SPECIFIED PARAMETERS FROM INPUT FILE 5 (FREE FORMAT)    CSL01060
C                                                                       CSL01070
C     READ USER-PROVIDED HEADER FOR RUN                                 CSL01080
      READ(5,20) EXPLAN                                                 CSL01090
      WRITE(6,20) EXPLAN                                                CSL01100
      READ(5,20) EXPLAN                                                 CSL01110
      WRITE(6,20) EXPLAN                                                CSL01120
   20 FORMAT(20A4)                                                      CSL01130
C                                                                       CSL01140
C     READ ORDER OF MATRICES (N) , MAXIMUM ORDER OF T-MATRIX (KMAX),    CSL01150
C     NUMBER OF T-MATRICES ALLOWED (NMEVS), AND MATRIX IDENTIFICATION   CSL01160
C     NUMBERS (MATNO)                                                   CSL01170
      READ(5,20) EXPLAN                                                 CSL01180
      READ(5,*) N,KMAX,NMEVS,MATNO                                      CSL01190
C                                                                       CSL01200
C     READ SEEDS FOR LANCZS AND INVERR SUBROUTINES (SVSEED AND RHSEED)  CSL01210
C     READ MAXIMUM NUMBER OF ITERATIONS ALLOWED FOR EACH INVERSE        CSL01220
C     ITERATION (MXINIT).                                               CSL01230
      READ(5,20) EXPLAN                                                 CSL01240
      READ(5,*) SVSEED,RHSEED,MXINIT                                    CSL01250
C                                                                       CSL01260
C     ISTART = (0,1):  ISTART = 0 MEANS ALPHA/BETA FILE IS NOT          CSL01270
C     AVAILABLE.  ISTART = 1 MEANS ALPHA/BETA FILE IS AVAILABLE ON      CSL01280
C     FILE 2.  COMPLEX SYMMETRIC HISTORIES MUST BE STORED               CSL01290
C     IN HEX FORMAT (4Z20).                                             CSL01300
C     ISTOP = (0,1):  ISTOP = 0 MEANS PROCEDURE GENERATES ALPHA/BETA    CSL01310
C     FILE AND THEN TERMINATES.  ISTOP = 1 MEANS PROCEDURE GENERATES    CSL01320
C     ALPHAS/BETAS IF NEEDED AND THEN COMPUTES EIGENVALUES AND ERROR    CSL01330
C     ESTIMATES AND THEN TERMINATES.                                    CSL01340
      READ(5,20) EXPLAN                                                 CSL01350
      READ(5,*) ISTART,ISTOP                                            CSL01360
C                                                                       CSL01370
C     IHIS = (0,1):  IHIS = 0 MEANS ALPHA/BETA FILE IS NOT WRITTEN      CSL01380
C     TO FILE 1.  IHIS = 1 MEANS ALPHA/BETA FILE IS WRITTEN TO FILE 1.  CSL01390
C     IDIST = (0,1):  IDIST = 0 MEANS DISTINCT T(1,MEV)-EIGENVALUES     CSL01400
C     ARE NOT WRITTEN TO FILE 11.  IDIST = 1 MEANS DISTINCT             CSL01410
C     T(1,MEV)-EIGENVALUES ARE WRITTEN TO FILE 11.                      CSL01420
C     SAVTEV = (-1,0,1):  SAVTEV = - 1 MEANS T(1,MEV) AND T(2,MEV)      CSL01430
C     EIGENVALUES ARE AVAILABLE ON FILE 10 FROM AN EARLIER RUN.         CSL01440
C     IN THIS CASE, ALPHA/BETA FILE FROM THAT RUN MUST BE               CSL01450
C     AVAILABLE ON FILE 2.                                              CSL01460
C     SAVTEV = 0 MEANS WE WILL NOT SAVE THE T(1,MEV) AND T(2,MEV)       CSL01470
C     EIGENVALUES.  SAVTEV = 1 MEANS WE WRITE THE T(1,MEV) AND          CSL01480
C     T(2,MEV) EIGENVALUES TO FILE 10.                                  CSL01490
C     IWRITE = (0,1):  IWRITE = 0 MEANS NO INTERMEDIATE OUTPUT          CSL01500
C     FROM THE COMPUTATIONS IS WRITTEN TO FILE 6.  IWRITE = 1 MEANS     CSL01510
C     EIGENVALUES AND ERROR ESTIMATES ARE WRITTEN TO FILE 6             CSL01520
C     AS THEY ARE COMPUTED.                                             CSL01530
      READ(5,20) EXPLAN                                                 CSL01540
```

```
      READ(5,*) IHIS,IDIST,SAVTEV,IWRITE                                CSL01550
C                                                                       CSL01560
      IF(SAVTEV.GE.0) GO TO 30                                          CSL01570
      NMEVS = 1                                                         CSL01580
      IF(ISTART.EQ.0) GO TO 610                                         CSL01590
C                                                                       CSL01600
   30 CONTINUE                                                          CSL01610
C     READ IN THE RELATIVE TOLERANCE (RELTOL) FOR USE IN THE LUMP       CSL01620
C     SUBROUTINE                                                        CSL01630
      READ(5,20) EXPLAN                                                 CSL01640
      READ(5,*) RELTOL                                                  CSL01650
C                                                                       CSL01660
C     READ IN THE SIZES OF THE T(1,MEV) MATRICES TO BE CONSIDERED.      CSL01670
      READ(5,20) EXPLAN                                                 CSL01680
      READ(5,*) (NMEV(J), J=1,NMEVS)                                    CSL01690
C                                                                       CSL01700
C-----------------------------------------------------------------------CSL01710
C     INITIALIZE THE ARRAYS FOR THE USER-SPECIFIED MATRIX               CSL01720
C     AND PASS THE STORAGE LOCATIONS OF THESE ARRAYS TO THE             CSL01730
C     MATRIX-VECTOR MULTIPLY SUBROUTINE CMATV.                          CSL01740
C                                                                       CSL01750
      CALL USPEC(N,MATNO)                                               CSL01760
C                                                                       CSL01770
C-----------------------------------------------------------------------CSL01780
C     MASK UNDERFLOW AND OVERFLOW                                       CSL01790
C                                                                       CSL01800
      CALL MASK                                                         CSL01810
C                                                                       CSL01820
C-----------------------------------------------------------------------CSL01830
C                                                                       CSL01840
C     WRITE TO FILE 6, A SUMMARY OF THE PARAMETERS FOR THIS RUN         CSL01850
C                                                                       CSL01860
      WRITE(6,40) MATNO,N,KMAX                                          CSL01870
   40 FORMAT(/3X,'MATRIX ID',4X,'ORDER OF A',4X,'MAX ORDER OF T'/       CSL01880
     1 I12,I14,I18/)                                                    CSL01890
C                                                                       CSL01900
      WRITE(6,50) ISTART,ISTOP                                          CSL01910
   50 FORMAT(/2X,'ISTART',3X,'ISTOP'/2I8/)                              CSL01920
C                                                                       CSL01930
      WRITE(6,60) IHIS,IDIST,SAVTEV,IWRITE                              CSL01940
   60 FORMAT(/4X,'IHIS',3X,'IDIST',3X,'SAVTEV',2X,'IWRITE'/2I8,I9,I8/)  CSL01950
C                                                                       CSL01960
      WRITE(6,70) SVSEED,RHSEED                                         CSL01970
   70 FORMAT(/' SEEDS FOR RANDOM NUMBER GENERATOR'//                    CSL01980
     1 4X,'LANCZS SEED',4X,'INVERR SEED'/2I15/)                         CSL01990
C                                                                       CSL02000
      WRITE(6,80) (NMEV(J), J=1,NMEVS)                                  CSL02010
   80 FORMAT(/' SIZES OF T-MATRICES TO BE CONSIDERED'/(6I12))           CSL02020
C                                                                       CSL02030
      WRITE(6,90) RELTOL,GAPTOL,BTOL                                    CSL02040
   90 FORMAT(/' RELATIVE TOLERANCE USED TO COMBINE COMPUTED T-EIGENVALUECSL02050
     1S'/E15.3/' RELATIVE GAP TOLERANCES USED IN INVERSE ITERATION'/    CSL02060
     1E15.3/' RELATIVE TOLERANCE FOR CHECK ON SIZE OF BETAS'/E15.3/)    CSL02070
C                                                                       CSL02080
      IF (ISTART.EQ.0) GO TO 140                                        CSL02090
```

```
C
C     READ IN ALPHA BETA HISTORY
C     HISTORY MUST BE STORED IN MACHINE FORMAT TO PREVENT
C     ERRORS CAUSED BY INPUT/OUTPUT CONVERSIONS.
C
      READ(2,100)MOLD,NOLD,SVSOLD,MATOLD
  100 FORMAT(2I6,I12,I8)
C
      IF (KMAX.LT.MOLD) KMAX = MOLD
      KMAX1 = KMAX + 1
C
C     CHECK THAT ORDER N, MATRIX ID MATNO, AND RANDOM SEED SVSEED
C     AGREE WITH THOSE IN THE HISTORY FILE.  IF NOT PROCEDURE STOPS.
C
      ITEMP = (NOLD-N)**2+(MATNO-MATOLD)**2+(SVSEED-SVSOLD)**2
C
      IF (ITEMP.EQ.0) GO TO 120
C
      WRITE(6,110)
  110 FORMAT(' PROGRAM TERMINATES'/    ' READ FROM FILE 2 CORRESPONDS TO
     1 DIFFERENT MATRIX THAN MATRIX SPECIFIED'/)
      GO TO 650
C
  120 CONTINUE
      MOLD1 = MOLD+1
C
      READ(2,130)(ALPHA(J), J=1,MOLD)
      READ(2,130)(BETA(J), J=1,MOLD1)
  130 FORMAT(4Z20)
C
      IF (KMAX.EQ.MOLD) GO TO 160
C
      READ(2,130)(V1(J), J=1,N)
      READ(2,130)(V2(J), J=1,N)
C
  140 CONTINUE
      IIX = SVSEED
C
C-----------------------------------------------------------------------
C
      CALL LANCZS(CMATV,V1,V2,ALPHA,BETA,GR,GC,G,KMAX,MOLD1,N,IIX)
C
C-----------------------------------------------------------------------
C
      KMAX1 = KMAX + 1
C
      IF (IHIS.EQ.0.AND.ISTOP.GT.0) GO TO 160
C
      WRITE(1,150) KMAX,N,SVSEED,MATNO
  150 FORMAT(2I6,I12,I8,' = KMAX,N,SVSEED,MATNO')
C
      WRITE(1,130)(ALPHA(I), I=1,KMAX)
      WRITE(1,130)(BETA(I), I=1,KMAX1)
C
      WRITE(1,130)(V1(I), I=1,N)
```

```
      WRITE(1,130)(V2(I), I=1,N)                                        CSL02650
C                                                                       CSL02660
      IF (ISTOP.EQ.0) GO TO 520                                         CSL02670
C                                                                       CSL02680
  160 CONTINUE                                                          CSL02690
      BKMIN = BTOL                                                      CSL02700
      WRITE(6,170)                                                      CSL02710
  170 FORMAT(/' T-MATRICES (ALPHA AND BETA) ARE NOW AVAILABLE'/)        CSL02720
C                                                                       CSL02730
C-----------------------------------------------------------------------CSL02740
C     SUBROUTINE TNORM CHECKS MIN|BETA|/(ESTIMATED NORM(A)) > BTOL .    CSL02750
C     IF THIS IS VIOLATED IB IS SET EQUAL TO THE NEGATIVE OF THE INDEX  CSL02760
C     OF THE MINIMAL BETA.  IF(IB < 0) THEN SUBROUTINE TNORM IS         CSL02770
C     CALLED FOR EACH VALUE OF MEV TO DETERMINE WHETHER OR NOT THERE    CSL02780
C     IS A BETA IN THE T-MATRIX SPECIFIED THAT VIOLATES THIS TEST.      CSL02790
C     IF THERE IS SUCH A BETA THE PROGRAM TERMINATES FOR THE USER       CSL02800
C     TO DECIDE WHAT TO DO.  THIS TEST CAN BE OVER-RIDDEN BY            CSL02810
C     SIMPLY MAKING BTOL SMALLER, BUT THEN THERE IS THE POSSIBILITY     CSL02820
C     THAT LOSSES IN THE LOCAL ORTHOGONALITY MAY HURT THE COMPUTATIONS. CSL02830
C                                                                       CSL02840
C     TNORM ALSO COMPUTES TKMAX = MAX(|ALPHA(K)|,|BETA(K)|, K=1,KMAX).  CSL02850
C     HOWEVER, IN THE COMPLEX SYMMETRIC CASE SINCE ALL OF THE           CSL02860
C     EIGENVALUES OF T(1,MEV) ARE COMPUTED, TKMAX IS NOT USED TO SCALE  CSL02870
C     THE T-MULTIPLICITY AND SPURIOUS TOLERANCES.  THE COMPUTED         CSL02880
C     T-EIGENVALUE LARGEST IN MAGNITUDE IS USED INSTEAD.                CSL02890
C                                                                       CSL02900
      CALL TNORM(ALPHA,BETA,BKMIN,TKMAX,KMAX,IB)                        CSL02910
C                                                                       CSL02920
C-----------------------------------------------------------------------CSL02930
C                                                                       CSL02940
C     LOOP ON THE SIZE OF THE T-MATRIX                                  CSL02950
C                                                                       CSL02960
  180 CONTINUE                                                          CSL02970
      MMB = MMB + 1                                                     CSL02980
      MEV = NMEV(MMB)                                                   CSL02990
C     IS MEV TOO LARGE ?                                                CSL03000
      IF(MEV.LE.KMAX) GO TO 200                                         CSL03010
      WRITE(6,190) MMB, MEV, KMAX                                       CSL03020
  190 FORMAT(/' TERMINATE PRIOR TO CONSIDERING THE',I6,'TH T-MATRIX'/   CSL03030
     1' BECAUSE THE SIZE REQUESTED',I6,' IS GREATER THAN THE MAXIMUM SIZCSL03040
     1E ALLOWED',I6/)                                                   CSL03050
      GO TO 520                                                         CSL03060
C                                                                       CSL03070
  200 MP1 = MEV + 1                                                     CSL03080
      BETAM = BETA(MP1)                                                 CSL03090
C                                                                       CSL03100
      IF (IB.GE.0) GO TO 220                                            CSL03110
C                                                                       CSL03120
      T0 = BTOL                                                         CSL03130
C                                                                       CSL03140
C-----------------------------------------------------------------------CSL03150
C                                                                       CSL03160
      CALL TNORM(ALPHA,BETA,T0,T1,MEV,IBMEV)                            CSL03170
C                                                                       CSL03180
C-----------------------------------------------------------------------CSL03190
```

```
C
  210 TEMP = T0/TKMAX
      IBMEV = IABS(IBMEV)
      IF (TEMP.GE.BTOL) GO TO 220
      IBMEV = -IBMEV
      GO TO 590
  220 CONTINUE
C
C-----------------------------------------------------------------------
C     SUBROUTINE COMPEV CALLS SUBROUTINE CMTQL1 TO COMPUTE THE
C     T-EIGENVALUES.  COMPEV THEN APPLIES THE T-MULTIPLICITY AND
C     SPURIOUS TESTS TO THE COMPUTED T-EIGENVALUES.  HERE INITIALIZE
C     THE TOLERANCES USED IN THE T-MULTIPLICITY AND THE SPURIOUS
C     TESTS.  THE MAX(|LAMBDA(T(1,MEV)|) WILL BE INCORPORATED
C     INSIDE THE SUBROUTINE COMPEV.  NOTE THAT THE OBSERVED ACCURACY
C     OF THE COMPUTED T-EIGENVALUES FOR THE COMPLEX SYMMETRIC CASE
C     IS APPROXIMATELY 3 DIGITS LESS THAN THAT ACHIEVED IN THE REAL
C     CASE. THUS, A FACTOR OF 500 HAS BEEN INTRODUCED.  THIS HOWEVER
C     MEANS THAT THIS TEST IS NOT AS SHARP AS IT WAS IN THE
C     REAL SYMMETRIC AND HERMITIAN CASES.  THUS, IT HAS LOWER
C     RESOLUTION AND CAN OCCASIONALLY MAKE A MISTAKE.
C
      MULTOL = 500.D0 * DFLOAT(MEV+1000) * EPSM
      SPUTOL =  MULTOL
C
C     ON RETURN FROM COMPEV
C     NDIS = NUMBER OF DISTINCT EIGENVALUES OF T(1,MEV)
C     VS = DISTINCT T-EIGENVALUES IN INCREASING ORDER OF MAGNITUDE
C     GR(K) = |VS(K)|, K = 1,NDIS,  GR(K).LE.GR(K+1)
C     MP = T-MULTIPLICITIES OF THE T-EIGENVALUES IN VS
C     MP(I) = (0,1,MI), MI>1, I=1,NDIS  MEANS:
C        (0)  VS(I) IS SPURIOUS
C        (1)  VS(I) IS SIMPLE AND GOOD
C        (MI) VS(I) IS T-MULTIPLE AND IS THEREFORE NOT ONLY GOOD BUT
C             ALSO A CONVERGED GOOD T-EIGENVALUE.
C
C
      CALL COMPEV(ALPHA,BETA,V1,V2,VS,GR,MULTOL,SPUTOL,MP,MP2,
     1MEV,NDIS,SAVTEV)
C
C-----------------------------------------------------------------------
C
      IF (NDIS.EQ.0) GO TO 630
C
C     ON EXIT FROM COMPEV MULTOL AND SPUTOL SHOULD BE SCALED
C     BY THE SIZES OF THE T-EIGENVALUES
      EVMAX = GR(NDIS)
      LOOP = NDIS
C
C-----------------------------------------------------------------------
C
      CALL LUMP(VS,V1,GR,RELTOL,SPUTOL,SCALE2,MP,MP2,LOOP)
C
C-----------------------------------------------------------------------
C
```

```
      IF (LOOP.LT.0) GO TO 650
C
      IF (NDIS.EQ.LOOP) GO TO 240
C
      WRITE(6,230) NDIS,LOOP,MEV
  230 FORMAT(/' AFTER LUMP NDIS,LOOP,MEV = ',3I6/)
C
  240 CONTINUE
      NDIS = LOOP
C
C-----------------------------------------------------------------------
C     CALCULATE MINGAPS FOR DISTINCT T(1,MEV) EIGENVALUES.
C     ON EXIT |GG(K)| = MIN(J.NE.K,|VS(K)-VS(J)|), MP2(K)=J INDEX
C     FOR MINIMUM.  GG(K)< 0 MEANS NEAREST NEIGHBOR IS SPURIOUS.
      IGAP = 0
      ITAG = 1
C
      CALL COMGAP(VS,GR,GG,MP,MP2,NDIS,IGAP,ITAG)
C
C-----------------------------------------------------------------------
C
C     SET CONVERGENCE CRITIERION
      TTOL = EPSM * EVMAX
      CONTOL = CDABS(BETAM)*1.D-10
C
  250 CONTINUE
      BETA(MP1) = BETAM
C
C-----------------------------------------------------------------------
C     THE SUBROUTINE ISOEV LABELS THOSE SIMPLE EIGENVALUES OF T(1,MEV)
C     WITH VERY SMALL GAPS BETWEEN NEIGHBORING EIGENVALUES OF T(1,MEV)
C     TO AVOID COMPUTING ERROR ESTIMATES FOR ANY SIMPLE GOOD
C     T-EIGENVALUE THAT IS TOO CLOSE TO A SPURIOUS T-EIGENVALUE.
C     MP(I) = -1 MEANS THAT THE GOOD T-EIGENVALUE IS SIMPLE AND
C     IS TOO CLOSE TO A SPURIOUS T-EIGENVALUE.
C
C     NG = NUMBER OF GOOD T-EIGENVALUES.
C     NISO = NUMBER OF ISOLATED GOOD T-EIGENVALUES.
C     GG = MINIMAL GAPS IN T(1,MEV)
C     GR(K) = |VS(K)|, K=1,NDIS
C
      CALL ISOEV(VS,GR,GG,GAPTOL,SPUTOL,SCALE1,MP,NDIS,NG,NISO)
C
C-----------------------------------------------------------------------
C
      WRITE(6,260)NG,NISO,NDIS
  260 FORMAT(/I6,' GOOD T-EIGENVALUES HAVE BEEN COMPUTED'/
     1 I6,' OF THESE ARE 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 300
C
      WRITE(11,270) NDIS,NISO,MEV,N,SVSEED,MATNO
  270 FORMAT(/4I6,I12,I8,' = NDIS,NISO,MEV,N,SVSEED,MATNO'/)
```

```
C
      WRITE(11,280) (I,MP(I),VS(I),GG(I),MP2(I), I=1,NDIS)
  280 FORMAT(I4,I4,2E20.12,E12.3,I6)
C
      WRITE(11,290) NDIS, (MP(I), I=1,NDIS)
  290 FORMAT(/I6,' = NDIS, T-MULTIPLICITIES (0 MEANS  SPURIOUS)'/(20I4))
C
  300 CONTINUE
C
      IF (NISO.NE.0) GO TO 330
C
      WRITE(4,310) MEV
  310 FORMAT(/' AT MEV = ',I6,' THERE ARE NO ISOLATED T-EIGENVALUES'/
     1' SO NO ERROR ESTIMATES WERE COMPUTED/')
C
      WRITE(6,320)
  320 FORMAT(/' ALL COMPUTED GOOD T-EIGENVALUES ARE T-MULTIPLE'/
     1 ' THEREFORE THESE EIGENVALUES ARE ASSUMED TO HAVE CONVERGED')
C
      ICONV = 1
      GO TO 370
C
  330 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     GG(J) = MINIMUM GAP IN T(1,MEV) FOR EACH VS(J), J=1,NDIS
C     G(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     T-EIGENVALUE THE INVERSE ITERATION DID NOT CONVERGE IN <= MXINIT
C     STEPS AND THAT THE CORRESPONDING ERROR ESTIMATE IS QUESTIONABLE.
C
C     ON EXIT
C     V2 CONTAINS THE ISOLATED GOOD T-EIGENVALUES
C     GR 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
C
      CALL INVERR(ALPHA,BETA,V1,V2,VS,EPSM,GR,GC,G,GG,MP,MP2,MEV,MMB,
     1NDIS,NISO,N,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 = CDABS(BETA(MEV+1)*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(6,340) CONTOL
  340 FORMAT(/' CONVERGENCE IS TESTED USING THE CONVERGENCE TOLERANCE',
     1E13.4/)
C
      DO 350 I = 1,NISO
      IF (ABS(G(I)).GT.CONTOL) GO TO 370
  350 CONTINUE
      ICONV = 1
      MMB = NMEVS
C
      WRITE(6,360) CONTOL
  360 FORMAT(' ALL COMPUTED ERROR ESTIMATES WERE LESS THAN',E15.4/
     1 ' THEREFORE PROCEDURE TERMINATES'/)
C
  370 CONTINUE
C
C     IN REAL SYMMETRIC AND HERMITIAN LANCZOS PROGRAMS
C     AT THIS CORRESPONDING POINT THE SUBROUTINE PRTEST IS CALLED
C     TO IDENTIFY ANY T-EIGENVALUES THAT MAY HAVE BEEN MISLABELLED
C     AS SPURIOUS BECAUSE THEIR PROJECTIONS ON THE STARTING VECTOR
C     WERE TOO SMALL.  THIS CHECK WAS MADE ONLY AFTER CONVERGENCE
C     HAD OCCURRED.  HOWEVER, THE PRTEST SUBROUTINE IS BASED UPON
C     STURM SEQUENCING AND THAT IS NOT VALID FOR COMPLEX SYMMETRIC
C     MATRICES.  PERHAPS THERE IS SOME RECTANGLE ANALOG OF THE
C     PRTEST BUT WE HAVE NOT ATTEMPTED TO IDENTIFY AND INCLUDE
C     SUCH A TEST BECAUSE WE EXPECT, AS IN THE REAL SYMMETRIC AND
C     HERMITIAN CASES THAT HIDDEN EIGENVALUES WILL BE RARE.
C
C     WRITE THE GOOD T-EIGENVALUES TO FILE 3.  FIRST TRANSFER THEM
C     TO V2 AND THEIR T-MULTIPLICITIES TO THE CORRESPONDING POSITIONS
C     IN MP AND COMPUTE THE A-MINGAPS, THE MINIMAL GAPS BETWEEN THE
C     GOOD T-EIGENVALUES.  THESE GAPS WILL BE PUT IN THE ARRAY GG.
C     NOTE THAT AFTER THE SECOND CALL TO COMGAP THE ARRAY GC
C     WILL CONTAIN THE CORRESPONDING MINIMAL GAPS IN THE
C     T-MATRIX, T(1,MEV).
C
  380 CONTINUE
C
      NG = 0
      DO 390 I = 1,NDIS
      IF (MP(I).EQ.0) GO TO 390
      NG = NG+1
      MP(NG) = MP(I)
      V2(NG) = VS(I)
      GC(NG) = GG(I)
  390 CONTINUE
C
      DO 400 I = 1,NG
  400 GR(I) = CDABS(V2(I))
C
C-----------------------------------------------------------------------
C     CALCULATE MINGAPS FOR GOODEV
C     ON EXIT GG(K) = MIN(J.NE.K,|V2(K)-V2(J)|), MP2(K)=J INDEX FOR MIN
C     NG = NUMBER OF COMPUTED DISTINCT GOOD T-EIGENVALUES.
      IGAP = 0
```

```
      ITAG = 0
C
      CALL COMGAP(V2,GR,GG,MP,MP2,NG,IGAP,ITAG)
C
C-----------------------------------------------------------------------
C
C     WRITE GOOD T-EIGENVALUES OUT TO FILE 3.
C
      WRITE(6,410)MEV
  410 FORMAT(//' EIGENVALUE CALCULATION AT MEV = ',I6,'  IS COMPLETE'/)
C
      WRITE(3,420)NG,NDIS,MEV,N,SVSEED,MATNO,MULTOL,SPUTOL,IB,BTOL
  420 FORMAT(4I6,I12,I8,' = NG,NDIS,MEV,N,SVEED,MATNO'/
     1 2E15.5,I6,E13.4,' = MULTOL,SPUTOL,IB,BTOL'/
     1' EVNO',1X,'MULT',13X,'R(GOODEV)',13X,'I(GOODEV)',
     1 3X,'TMINGAP',3X,'AMINGAP',1X,'NEIGH')
C
      WRITE(3,430)(I,MP(I),V2(I),GC(I),GG(I),MP2(I), I=1,NG)
  430 FORMAT(2I5,2E22.14,2E10.3,I6)
C
C     ORDER GOODEV BY INCREASING GAP SIZE
      DO 440 I = 1,NG
      MP(I) = I
      V1(I) = V2(I)
      G(I) = GG(I)
  440 CONTINUE
C
C     WRITE(12,436)
  450 FORMAT(' MINGAPS FOR GOOD T-EIGENVALUES'/
     1 1X,'EVNUM',1X,'NEIGH',15X,'R(EV)',15X,'I(EV)',4X,'MINGAP')
C
C     WRITE(12,439) (K,MP2(K),V2(K),G(K), K = 1,NG)
  460 FORMAT(2I6,2E20.12,E10.3)
C
      DO 480 K = 2,NG
      KM1 = K-1
      DO 470 L = 1,KM1
      KK = K-L
      KP1 = KK+1
      IF (G(KP1).GE.G(KK)) GO TO 480
      Z = V1(KK)
      V1(KK) = V1(KP1)
      V1(KP1) = Z
      GTEMP = G(KK)
      G(KK) = G(KP1)
      G(KP1) = GTEMP
      ITEMP = MP(KK)
      MP(KK) = MP(KP1)
      MP(KP1) = ITEMP
  470 CONTINUE
  480 CONTINUE
C
C     WRITE(12,441)
      WRITE(3,490)
  490 FORMAT(' T-EIGENVALUES ORDERED BY INCREASING MINGAP'/
```

```
     1 1X,'GAPNUM',1X,'EVNUM',15X,'R(EV)',15X,'I(EV)',4X,'MINGAP')      CSL05950
C                                                                       CSL05960
C     WRITE(12,442) (K,MP(K),V1(K),G(K), K = 1,NG)                      CSL05970
      WRITE(3,500) (K,MP(K),V1(K),G(K), K = 1,NG)                       CSL05980
  500 FORMAT(I7,I6,2E20.12,E10.3)                                       CSL05990
C                                                                       CSL06000
  510 CONTINUE                                                          CSL06010
C                                                                       CSL06020
C     IF CONVERGENCE FLAG ICONV.NE.1 AND NUMBER OF T-MATRICES           CSL06030
C     CONSIDERED TO DATE IS LESS THAN NUMBER ALLOWED, INCREMENT MEV.    CSL06040
C     AND LOOP BACK TO 210 TO REPEAT COMPUTATIONS.  RESTORE BETA(MEV+1).CSL06050
C                                                                       CSL06060
      BETA(MP1) = BETAM                                                 CSL06070
C                                                                       CSL06080
      IF (MMB.LT.NMEVS.AND.ICONV.NE.1) GO TO 180                        CSL06090
C                                                                       CSL06100
C     END OF LOOP ON DIFFERENT SIZE T-MATRICES ALLOWED.                 CSL06110
C                                                                       CSL06120
  520 CONTINUE                                                          CSL06130
C                                                                       CSL06140
      IF(ISTOP.EQ.0)  WRITE(6,530)                                      CSL06150
  530 FORMAT(/' T-MATRICES (ALPHA AND BETA) ARE NOW AVAILABLE, TERMINATECSL06160
     1')                                                                CSL06170
      IF (ISTOP.EQ.0.AND.KMAX.NE.MOLD) WRITE(1,540)                     CSL06180
      IF (IHIS.EQ.1.AND.KMAX.NE.MOLD) WRITE(1,540)                      CSL06190
  540 FORMAT(/' ABOVE ARE THE FOLLOWING VECTORS '/                      CSL06200
     1 '  ALPHA(I), I = 1,KMAX'/                                        CSL06210
     2 '  BETA(I), I = 1,KMAX+1'/                                       CSL06220
     3 ' FINAL TWO LANCZOS VECTORS OF ORDER N FOR I = KMAX,KMAX+1'/     CSL06230
     4 ' ALPHA BETA ARE IN HEX FORMAT 4Z20 '/                           CSL06240
     4 ' LANCZOS VECTORS ARE IN HEX FORMAT 4Z20 '/                      CSL06250
     5 ' ----- END OF FILE 1 NEW ALPHA, BETA HISTORY---------------'///)CSL06260
C                                                                       CSL06270
      IF (ISTOP.EQ.0) GO TO 650                                         CSL06280
C                                                                       CSL06290
      WRITE(3,550)                                                      CSL06300
  550 FORMAT(/' ABOVE ARE COMPUTED GOOD T-EIGENVALUES'/                 CSL06310
     1 ' NG = NUMBER OF GOOD T-EIGENVALUES COMPUTED'/                   CSL06320
     2 ' NDIS = NUMBER OF COMPUTED DISTINCT EIGENVALUES OF T(1,MEV)'/   CSL06330
     3 ' N = ORDER OF A,  MATNO = MATRIX IDENT'/                        CSL06340
     4 ' MULTOL = T-MULTIPLICITY TOLERANCE FOR T-EIGENVALUES'/          CSL06350
     4 ' SPUTOL = SPURIOUS TOLERANCE FOR T-EIGENVALUES'/                CSL06360
     4 ' MULT IS THE T-MULTIPLICITY OF GOOD T-EIGENVALUE'/              CSL06370
     5 ' MULT = -1 MEANS SPURIOUS T-EIGENVALUE TOO CLOSE'/              CSL06380
     6 ' DO NOT COMPUTE ERROR ESTIMATES FOR SUCH T-EIGENVALUES'/        CSL06390
     7 ' AMINGAP = MINIMAL GAP BETWEEN THE COMPUTED A-EIGENVALUES'/     CSL06400
     9 ' TMINGAP= MINIMAL GAP W.R.T.  DISTINCT EIGENVALUES IN T(1,MEV)'/CSL06410
     2 ' ----- END OF FILE 3 GOOD T-EIGENVALUES---------------------'///CSL06420
     3 )                                                                CSL06430
C                                                                       CSL06440
      IF (IDIST.NE.0) WRITE(11,560)                                     CSL06450
  560 FORMAT(/' ABOVE ARE THE DISTINCT EIGENVALUES OF T(1,MEV).'/       CSL06460
     2 ' THE FORMAT IS      T-MULTIPLICITY   T-EIGENVALUE   TMINGAP'/  CSL06470
     4 ' T-MULTIPLICITY = -1 MEANS THAT THE SUBROUTINE ISOEV HAS TAGGED'CSL06480
     5 /'   THIS SIMPLE T-EIGENVALUE AS HAVING A VERY CLOSE SPURIOUS'/  CSL06490
```

```
     6 '   T-EIGENVALUE SO THAT NO ERROR ESTIMATE WILL BE COMPUTED'/    CSL06500
     7 '   FOR THAT EIGENVALUE IN SUBROUTINE INVERR.'/                  CSL06510
     9 ' EACH OF THE DISTINCT T-EIGENVALUE TABLES IS FOLLOWED'/         CSL06520
     9 ' BY THE T-MULTIPLICITY PATTERN.'/                               CSL06530
     1 ' NDIS = NUMBER OF COMPUTED DISTINCT EIGENVALUES OF T(1,MEV).'/  CSL06540
     2 ' NG = NUMBER OF GOOD T-EIGENVALUES. '/                          CSL06550
     3 ' NISO = NUMBER OF ISOLATED GOOD T-EIGENVALUES. '/               CSL06560
     4 ' NISO ALSO IS THE COUNT OF +1 ENTRIES IN MULTIPLICITY PATTERN.'/CSL06570
     5 ' -----END OF FILE 11 DISTINCT T-EIGENVALUES----------------'///)CSL06580
C                                                                       CSL06590
      WRITE(4,570)                                                      CSL06600
  570 FORMAT(/' ABOVE ARE THE ERROR ESTIMATES OBTAINED FOR THE ISOLATED CSL06610
     1GOOD T-EIGENVALUES'/                                              CSL06620
     1' OBTAINED VIA INVERSE ITERATION IN THE SUBROUTINE INVERR.'/      CSL06630
     1' ALL OTHER GOOD T-EIGENVALUES HAVE CONVERGED.'/                  CSL06640
     2' ERROR ESTIMATE = CDABS(BETAM*(UM))'/                            CSL06650
     2' WHERE BETAM = BETA(MEV+1) AND UM = U(MEV).'/                    CSL06660
     3' U = UNIT EIGENVECTOR OF T WHERE T*U = EV*U AND EV = ISOLATED GOOCSL06670
     3D T-EIGENVALUE.'/                                                 CSL06680
     4' TMINGAP = GAP TO NEAREST DISTINCT EIGENVALUE OF T(1,MEV).'/     CSL06690
     6' ------ END OF FILE 4 ERRINV -------------------------------'///)CSL06700
C                                                                       CSL06710
      IF(SAVTEV.LT.0) GO TO 650                                         CSL06720
      WRITE(10,580)                                                     CSL06730
  580 FORMAT(//' ABOVE ARE THE T(1,MEV) EIGENVALUES FOLLOWED BY THE'/   CSL06740
     1 ' T(2,MEV) EIGENVALUES FOR MEV = NMEV(J), J = 1,NMEVS'/          CSL06750
     1 ' ------END OF FILE 10 T-T2EVAL----------------------'///)       CSL06760
C                                                                       CSL06770
      GO TO 650                                                         CSL06780
C                                                                       CSL06790
  590 CONTINUE                                                          CSL06800
C                                                                       CSL06810
      IBB = IABS(IBMEV)                                                 CSL06820
      TEMP = CDABS(BETA(IBB))                                           CSL06830
      IF (IBMEV.LT.0) WRITE(6,600) MEV,IBB,TEMP                         CSL06840
  600 FORMAT(/' PROGRAM TERMINATES BECAUSE MEV REQUESTED = ',I6,' IS .GTCSL06850
     1',I6/' AT WHICH AN ABNORMALLY SMALL BETA = ' ,E13.4,' OCCURRED'/) CSL06860
      GO TO 650                                                         CSL06870
C                                                                       CSL06880
  610 WRITE(6,620) SAVTEV,ISTART                                        CSL06890
  620 FORMAT(2I6,' = SAVTEV,ISTART'/' WHEN SAVTEV = -1, WE MUST HAVE ISTCSL06900
     1ART = 1'/)                                                        CSL06910
      GO TO 650                                                         CSL06920
C                                                                       CSL06930
  630 IF (NDIS.EQ.0.AND.ISTOP.GT.0) WRITE(6,640)                        CSL06940
  640 FORMAT(/' INTERVALS SPECIFIED FOR BISECT DID NOT CONTAIN ANY T-EIGCSL06950
     1ENVALUES'/' PROGRAM TERMINATES')                                  CSL06960
C                                                                       CSL06970
  650 CONTINUE                                                          CSL06980
C                                                                       CSL06990
      STOP                                                              CSL07000
C-----END OF MAIN PROGRAM FOR COMPLEX SYMMETRIC EIGENVALUE COMPUTATIONS-CSL07010
      END                                                               CSL07020
```

7.4 CSLEVEC: Main Program, Eigenvector Computations

```
C-----CSLEVEC (EIGENVECTORS OF COMPLEX SYMMETRIC MATRICES)--------------CSL00010
C  Authors:  Jane Cullum and Ralph A. Willoughby (Deceased)             CSL00020
C            Los Alamos National Laboratory                             CSL00030
C            Los Alamos, New Mexico 87544                               CSL00040
C                                                                       CSL00050
C            E-mail:  cullumj@lanl.gov                                  CSL00060
C                                                                       CSL00070
C  These codes are copyrighted by the authors.  These codes             CSL00080
C  and modifications of them or portions of them are NOT to be          CSL00090
C  incorporated into any commercial codes or used for any other         CSL00100
C  commercial purposes such as consulting for other companies,          CSL00110
C  without legal agreements with the authors of these Codes.            CSL00120
C  If these Codes or portions of them                                   CSL00130
C  are used in other scientific or engineering research works           CSL00140
C  the names of the authors of these codes and appropriate              CSL00150
C  references to their written work are to be incorporated in the       CSL00160
C  derivative works.                                                    CSL00170
C                                                                       CSL00180
C  This header is not to be removed from these codes.                   CSL00190
C                                                                       CSL00200
C        REFERENCE:  Cullum and Willoughby, Chapter 6,                  CSL00201
C        Lanczos Algorithms for Large Symmetric Eigenvalue ComputationsCSL00202
C        VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in    CSL00203
C        Applied Mathematics, 2002. SIAM Publications,                  CSL00204
C        Philadelphia, PA. USA                                          CSL00205
C                                                                       CSL00206
C                                                                       CSL00207
C     CONTAINS MAIN PROGRAM FOR COMPUTING AN EIGENVECTOR CORRESPONDING  CSL00210
C     TO EACH OF A SET OF EIGENVALUES THAT HAVE BEEN COMPUTED           CSL00220
C     ACCURATELY BY THE CORRESPONDING LANCZOS EIGENVALUE PROGRAM        CSL00230
C     (CSLEVAL) FOR NONDEFECTIVE COMPLEX SYMMETRIC MATRICES.            CSL00240
C     THIS PROGRAM COULD BE MODIFIED TO COMPUTE ADDITIONAL              CSL00250
C     EIGENVECTORS FOR THOSE EIGENVALUES WHICH ARE MULTIPLE EIGENVALUES CSL00260
C     OF THE GIVEN A-MATRIX.  THE AMOUNT OF ADDITIONAL COMPUTATION      CSL00270
C     REQUIRED WOULD DEPEND UPON THE GIVEN A-MATRIX AND UPON WHAT       CSL00280
C     PART OF THE SPECTRUM OF A IS INVOLVED.                            CSL00290
C                                                                       CSL00300
C     THESE LANCZOS EIGENVECTOR COMPUTATIONS ASSUME THAT EACH           CSL00310
C     EIGENVALUE THAT IS BEING CONSIDERED HAS CONVERGED AS AN           CSL00320
C     EIGENVALUE OF THE CORRESPONDING LANCZOS TRIDIAGONAL MATRICES.     CSL00330
C                                                                       CSL00340
C     PORTABILITY:                                                      CSL00350
C     THIS PROGRAM IS NOT PORTABLE DUE TO THE USE OF THE COMPLEX*16     CSL00360
C     VARIABLES AND CORRESPONDING COMPLEX FUNCTIONS.  MOREOVER, PFORT   CSL00370
C     IDENTIFIED THE FOLLOWING ADDITIONAL NONPORTABLE CONSTRUCTIONS:    CSL00380
C                                                                       CSL00390
C     1.  DATA/MACHEP/ STATEMENT                                        CSL00400
C     2.  ALL READ(5,*) STATEMENTS (FREE FORMAT)                        CSL00410
C     3.  FORMAT(20A4) USED WITH THE EXPLANATORY HEADER, EXPLAN         CSL00420
C     4.  FORMAT (4Z20) USED FOR ALPHA/ BETA FILE 2.                    CSL00430
C                                                                       CSL00440
```

```
C     IMPORTANT NOTE:  PROGRAM ALLOWS ENLARGEMENT OF THE ALPHA,BETA
C     ARRAYS.  IN PARTICULAR, IF ANY ONE OF THE EIGENVALUES SUPPLIED
C     IS T-SIMPLE AND NOT CLOSE TO A SPURIOUS T-EIGENVALUE, THE PROGRAM
C     REQUIRES THAT KMAX BE AT LEAST 11*MEV/8 + 12.  IF KMAX IS NOT
C     THIS LARGE, THEN THE PROGRAM WILL RESET KMAX TO THIS SIZE
C     AND EXTEND 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-----------------------------------------------------------------------
      COMPLEX*16  V1(1600),V2(1600),RITVEC(10000),ZEROC,TEMPC
      COMPLEX*16  ALPHA(1600),BETA(1601),GOODEV(50),TVEC(20000)
      COMPLEX*16  EVAL,ALFA,BATA,SUMC
      DOUBLE PRECISION  GR(1600),GC(1600)
      DOUBLE PRECISION  ERTOL,SUM,TEMP,BKMIN
      DOUBLE PRECISION  MULTOL,SPUTOL,SCALE0,BTOL
      DOUBLE PRECISION  ONE,ZERO,MACHEP,EPSM
      DOUBLE PRECISION  RELTOL,ERROR,ERRMIN,TERROR,TLAST(50)
      REAL  G(1600),AMINGP(50),TMINGP(50),EXPLAN(20)
      REAL  TERR(50),ERR(50),ERRDGP(50),RNORM(50),TBETA(50)
      INTEGER  MP(50),MA(50),ML(50),MINT(50),MFIN(50),IDELTA(50)
      INTEGER  SVSEED,SVSOLD,RHSEED
      INTEGER INTERC(1600)
      INTEGER  MBOUND,NTVCON,SVTVEC,TVSTOP,LVCONT,ERCONT,TFLAG
      DOUBLE PRECISION  DABS, DMAX1, DSQRT
      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, GR, GC:  >= MAX(N,KMAX)
C     5.  RITVEC:  >= N*NGOOD, WHERE NGOOD IS THE NUMBER OF EIGENVALUES
C                   SUPPLIED TO THIS PROGRAM.
C     6.  TVEC:  >= CUMULATIVE LENGTH OF ALL THE T-EIGENVECTORS NEEDED
C                    TO GENERATE THE DESIRED RITZ VECTORS.  AN EDUCATED
C                    GUESS AT AN APPROPRIATE LENGTH CAN BE OBTAINED
C                    BY RUNNING THE PROGRAM WITH THE FLAG MBOUND = 1
C                    AND MULTIPLYING THE RESULTING SIZE BY 5/4.
```

```
C     7.  INTERC:   >= KMAX                                             CSL01000
C     8.  GOODEV, AMINGP, TMINGP, TERR, ERR, ERRDGP, RNORM, TBETA,      CSL01010
C         TLAST, MP, MA, MINT, MFIN, AND IDELTA :    >= NUMBER OF       CSL01020
C         EIGENVALUES SUPPLIED.                                         CSL01030
C                                                                       CSL01040
C     OUTPUT HEADER                                                     CSL01050
      WRITE(6,10)                                                       CSL01060
   10 FORMAT(/' LANCZOS EIGENVECTOR PROCEDURE FOR COMPLEX SYMMETRIC MATRCSL01070
     1ICES'/)                                                           CSL01080
C                                                                       CSL01090
C     SET PROGRAM PARAMETERS                                            CSL01100
C     USER MUST NOT MODIFY SCALE0                                       CSL01110
      SCALE0 = 5.0D0                                                    CSL01120
      ZERO = 0.0D0                                                      CSL01130
      ZEROC = DCMPLX(ZERO,ZERO)                                         CSL01140
      ONE = 1.0D0                                                       CSL01150
      MPMIN = -1000                                                     CSL01160
      MONE = -1                                                         CSL01170
C     CONVERGENCE TOLERANCE FOR T-EIGENVECTORS FOR RITZ COMPUTATIONS    CSL01180
      ERTOL = 1.D-10                                                    CSL01190
C-----------------------------------------------------------------------CSL01200
C     READ USER-SPECIFIED PARAMETERS FROM INPUT FILE 5 (FREE FORMAT)    CSL01210
C                                                                       CSL01220
C     READ USER-PROVIDED HEADER FOR RUN                                 CSL01230
      READ(5,20) EXPLAN                                                 CSL01240
      WRITE(6,20) EXPLAN                                                CSL01250
   20 FORMAT(20A4)                                                      CSL01260
C                                                                       CSL01270
C     READ IN THE MAXIMUM PERMISSIBLE DIMENSIONS FOR THE TVEC ARRAY     CSL01280
C     (MDIMTV), FOR THE RITVEC ARRAY (MDIMRV), AND FOR THE BETA         CSL01290
C     ARRAY (MBETA).                                                    CSL01300
C                                                                       CSL01310
      READ(5,20) EXPLAN                                                 CSL01320
      READ(5,*) MDIMTV, MDIMRV, MBETA                                   CSL01330
C                                                                       CSL01340
C     READ IN RELATIVE TOLERANCE (RELTOL) USED IN DETERMINING           CSL01350
C     APPROPRIATE SIZES FOR THE T-MATRICES USED IN THE RITZ             CSL01360
C     VECTOR COMPUTATIONS                                               CSL01370
C                                                                       CSL01380
      READ(5,20) EXPLAN                                                 CSL01390
      READ(5,*) RELTOL                                                  CSL01400
C                                                                       CSL01410
C     SET FLAGS TO 0 OR 1:                                              CSL01420
C     MBOUND = 1:  PROGRAM TERMINATES AFTER COMPUTING 1ST GUESSES       CSL01430
C                  ON APPROPRIATE T-SIZES FOR USE IN THE RITZ VECTOR    CSL01440
C                  COMPUTATIONS                                         CSL01450
C     NTVCON = 0:  PROGRAM TERMINATES IF THE TVEC ARRAY IS NOT          CSL01460
C                  LARGE ENOUGH TO HOLD ALL THE T-EIGENVECTORS REQUIRED.CSL01470
C     SVTVEC = 0:  THE T-EIGENVECTORS ARE NOT WRITTEN TO FILE 11        CSL01480
C                  UNLESS TVSTOP = 1                                    CSL01490
C     SVTVEC = 1:  WRITE THE T-EIGENVECTORS TO FILE 11.                 CSL01500
C     TVSTOP = 1:  PROGRAM TERMINATES AFTER COMPUTING THE               CSL01510
C                  T-EIGENVECTORS                                       CSL01520
C     LVCONT = 0:  PROGRAM TERMINATES IF THE NUMBER OF T-EIGENVECTORS   CSL01530
C                  COMPUTED IS NOT EQUAL TO THE NUMBER OF RITZ          CSL01540
```

```
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.
C
      READ(5,20) EXPLAN
      READ(5,*) RHSEED
C
C     READ IN MATNO = MATRIX/RUN IDENTIFICATION NUMBER AND
C     N = ORDER OF A-MATRIX
C
      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
C     MASK UNDERFLOW AND OVERFLOW
      CALL MASK
C
C-----------------------------------------------------------------------
C     WRITE RUN PARAMETERS OUT TO FILE 6
C
```

```
      WRITE(6,30) MATNO,N                                               CSL02100
   30 FORMAT(/' MATRIX IDENTIFICATION NO. = ',I10,' ORDER OF A = ',I5)  CSL02110
C                                                                       CSL02120
      WRITE(6,40) MBOUND,NTVCON,SVTVEC,IREAD                            CSL02130
   40 FORMAT(/3X,'MBOUND',3X,'NTVCON',3X,'SVTVEC',3X,'IREAD'/3I9,I8)    CSL02140
C                                                                       CSL02150
      WRITE(6,50) TVSTOP,LVCONT,ERCONT,IWRITE                           CSL02160
   50 FORMAT(/3X,'TVSTOP',3X,'LVCONT',3X,'ERCONT',3X,'IWRITE'/4I9)      CSL02170
C                                                                       CSL02180
      WRITE(6,60) MDIMTV,MDIMRV,MBETA                                   CSL02190
   60 FORMAT(/3X,'MDIMTV',3X,'MDIMRV',3X,'MBETA'/2I9,I8)                CSL02200
C                                                                       CSL02210
      WRITE(6,70) RELTOL,RHSEED                                         CSL02220
   70 FORMAT(/7X,'RELTOL',3X,'RHSEED'/E13.4,I9)                         CSL02230
C                                                                       CSL02240
C                                                                       CSL02250
C     FROM FILE 3 READ IN THE NUMBER OF EIGENVALUES (NGOOD) FOR WHICH   CSL02260
C     EIGENVECTORS ARE REQUESTED, THE ORDER (MEV) OF THE LANCZOS        CSL02270
C     TRIDIAGONAL MATRIX USED IN COMPUTING THESE EIGENVALUES, THE       CSL02280
C     ORDER (NOLD) OF THE USER-SPECIFIED MATRIX USED IN THE EIGENVALUE  CSL02290
C     COMPUTATIONS, THE SEED (SVSEED) USED FOR GENERATING THE STARTING  CSL02300
C     VECTOR THAT WAS USED IN THOSE LANCZOS EIGENVALUE COMPUTATIONS,    CSL02310
C     AND THE MATRIX/RUN IDENTIFICATION NUMBER (MATOLD) USED IN THOSE   CSL02320
C     COMPUTATIONS.  ALSO READ IN THE NUMBER (NDIS) OF DISTINCT         CSL02330
C     EIGENVALUES OF T(1,MEV) THAT WERE COMPUTED BUT THIS VALUE IS      CSL02340
C     NOT USED IN THE EIGENVECTOR COMPUTATIONS.                         CSL02350
C                                                                       CSL02360
      READ(3,80) NGOOD,NDIS,MEV,NOLD,SVSEED,MATOLD                      CSL02370
   80 FORMAT(4I6,I12,I8)                                                CSL02380
C                                                                       CSL02390
C     READ IN THE TOLERANCES USED IN THE T-MULTIPLICITY AND SPURIOUS    CSL02400
C     TESTS DURING THE EIGENVALUE COMPUTATIONS.                         CSL02410
C     ALSO READ IN THE FLAG IB.  IF IB < 0, THEN SOME BETA(I) IN THE    CSL02420
C     T-MATRIX FILE PROVIDED ON FILE 2 FAILED THE ORTHOGONALITY         CSL02430
C     TEST IN THE TNORM SUBROUTINE.  USER SHOULD NOTE THAT THIS         CSL02440
C     PROGRAM PROCEEDS INDEPENDENTLY OF THE SIZES OF THE BETA USED.     CSL02450
C                                                                       CSL02460
      READ(3,90) MULTOL,SPUTOL,IB,BTOL                                  CSL02470
   90 FORMAT(2E15.5,I6,E13.4)                                           CSL02480
C                                                                       CSL02490
      WRITE(6,100) MULTOL,SPUTOL                                        CSL02500
  100 FORMAT(/' MULTIPLICITY TOLERANCE USED IN THE T-EIGENVALUE COMPUTATCSL02510
     1IONS WAS',E13.4/' TOLERANCE USED IN SPURIOUS CHECK',E13.4)        CSL02520
C                                                                       CSL02530
C     CONTINUE WRITE TO FILE 6 OF THE PARAMETERS FOR THIS RUN           CSL02540
C                                                                       CSL02550
      WRITE(6,110)NGOOD,NDIS,MEV,NOLD,MATOLD,SVSEED,MULTOL,SPUTOL,IB,   CSL02560
     1BTOL                                                              CSL02570
  110 FORMAT(/' EIGENVALUES SUPPLIED ARE READ IN FROM FILE 3'/' FILE 3  CSL02580
     1HEADER IS'/4X,'NG',2X,'NDIS',3X,'MEV',2X,'NOLD',2X,'MATOLD',4X,   CSL02590
     1'SVSEED'/4I6,I8,I10/7X,'MULTOL',7X,'SPUTOL',6X,'IB',9X,'BTOL'/    CSL02600
     12E13.4,I8,E13.4)                                                  CSL02610
C                                                                       CSL02620
C     IS THE ARRAY RITVEC LONG ENOUGH TO HOLD ALL OF THE DESIRED        CSL02630
C     RITZ VECTORS (APPROXIMATE EIGENVECTORS)?                          CSL02640
```

```
      NMAX = NGOOD*N
      IF(MBOUND.EQ.1) 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(1,MEV)-MULTIPLICITIES OF THE
C     EIGENVALUES WHOSE EIGENVECTORS ARE TO BE COMPUTED, THE VALUES
C     OF THESE EIGENVALUES AND THEIR MINIMAL GAPS AS EIGENVALUES
C     OF THE 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(5X,I5,2E22.14,2E10.3)
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 ',15X,' EIGENVALUE',14X,'TMULT',4X,'  TMINGAP ',4X,
     1'  AMINGAP '/(I6,2E20.12,I4,2E15.4))
C
C     READ IN ERROR ESTIMATES
      WRITE(6,170) MEV,SVSEED
C     CHECK WHETHER OR NOT THERE ARE ANY T-ISOLATED EIGENVALUES IN
C     THE EIGENVALUES PROVIDED
      DO 150 J=1,NGOOD
      IF(MP(J).EQ.1) GO TO 160
  150 CONTINUE
      GO TO 190
  160 READ(4,20) EXPLAN
      READ(4,20) EXPLAN
      READ(4,20) EXPLAN
  170 FORMAT(/' THESE EIGENVALUES WERE COMPUTED USING A T-MATRIX OF
     1ORDER ',I5/' AND SEED FOR RANDOM NUMBER GENERATOR =',I12)
      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,2E20.12,E14.3)
      IF(CDABS(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',2E20.12,' DOES NOT MATCH GOODEV(J) ='/2E20.12)
      GO TO 1550
C
  220 CONTINUE
C
```

```
      WRITE(6,230) (J,GOODEV(J),ERR(J), J=1,NGOOD)                      CSL03200
  230 FORMAT(' ERROR ESTIMATES ='/4X,' J',15X,'EIGENVALUE',20X,'ESTIMATECSL03210
     1'/(I6,2E20.12,E14.3))                                             CSL03220
C                                                                       CSL03230
C     READ IN THE SIZE OF THE T-MATRIX PROVIDED ON FILE 2.  READ IN     CSL03240
C     THE ORDER OF THE USER-SPECIFIED MATRIX , THE SEED FOR THE         CSL03250
C     RANDOM NUMBER GENERATOR, AND THE MATRIX/TEST IDENTIFICATION       CSL03260
C     NUMBER THAT WERE USED IN THE LANCZOS EIGENVALUE COMPUTATIONS.     CSL03270
C     IF FLAG IREAD = 0, REGENERATE HISTORY FROM SCRATCH                CSL03280
C     HISTORY MUST BE STORED IN MACHINE FORMAT, ((4Z20) FOR             CSL03290
C     IBM/3081)                                                         CSL03300
C                                                                       CSL03310
      IF(IREAD.EQ.0)  GO TO 330                                         CSL03320
C                                                                       CSL03330
      READ(2,240) KMAX,NOLD,SVSOLD,MATOLD                               CSL03340
  240 FORMAT(2I6,I12,I8)                                                CSL03350
C                                                                       CSL03360
      WRITE(6,250) KMAX,NOLD,SVSOLD,MATOLD                              CSL03370
  250 FORMAT(/' READ IN HEADER FOR T-MATRICES'/' FILE 2 HEADER IS'/     CSL03380
     1 2X,'KMAX',2X,'NOLD',6X,'SVSOLD',2X,'MATOLD'/2I6,I12,I8/)         CSL03390
C                                                                       CSL03400
C     CHECK THAT THE ORDER, THE MATRIX/TEST IDENTIFICATION NUMBER       CSL03410
C     AND THE SEED FOR THE RANDOM NUMBER GENERATOR USED IN THE          CSL03420
C     LANCZOS COMPUTATIONS THAT GENERATED THE HISTORY FILE              CSL03430
C     BEING USED AGREE WITH WHAT THE USER HAS SPECIFIED.                CSL03440
      IF (NOLD.NE.N.OR.MATOLD.NE.MATNO.OR.SVSOLD.NE.SVSEED) GO TO 1350  CSL03450
C                                                                       CSL03460
      KMAX1 = KMAX + 1                                                  CSL03470
C                                                                       CSL03480
C     READ IN THE T-MATRICES FROM FILE 2.  THESE ARE USED TO GENERATE   CSL03490
C     THE T-EIGENVECTORS THAT WILL BE USED IN THE RITZ VECTOR           CSL03500
C     COMPUTATIONS.  HISTORY MUST BE IN MACHINE FORMAT.                 CSL03510
C                                                                       CSL03520
      READ(2,260) (ALPHA(J), J=1,KMAX)                                  CSL03530
      READ(2,260) (BETA(J), J=1,KMAX1)                                  CSL03540
  260 FORMAT(4Z20)                                                      CSL03550
C                                                                       CSL03560
      READ(2,260) (V1(J), J=1,N)                                        CSL03570
      READ(2,260) (V2(J), J=1,N)                                        CSL03580
C                                                                       CSL03590
C     KMAX MAY BE ENLARGED IF THE SIZE AT WHICH THE EIGENVALUE          CSL03600
C     COMPUTATIONS WERE PERFORMED IS ESSENTIALLY KMAX AND               CSL03610
C     THERE IS AT LEAST ONE EIGENVALUE THAT IS T-SIMPLE AND             CSL03620
C     T-ISOLATED, IN THE SENSE THAT IF ITS CLOSEST NEIGHBOR IS TOO      CSL03630
C     CLOSE THAT NEIGHBOR IS A 'GOOD' T-EIGENVALUE.                     CSL03640
      DO 270 J = 1,NGOOD                                                CSL03650
      IF(MP(J).EQ.1) GO TO 290                                          CSL03660
  270 CONTINUE                                                          CSL03670
      WRITE(6,280)                                                      CSL03680
  280 FORMAT(/' ALL EIGENVALUES USED ARE T-MULTIPLE OR CLOSE TO SPURIOUSCSL03690
     1 T-EIGENVALUES'/' SO DO NOT CHANGE KMAX')                         CSL03700
      IF(KMAX.LT.MEV) GO TO 1370                                        CSL03710
      GO TO 310                                                         CSL03720
C                                                                       CSL03730
  290 KMAXN= 11*MEV/8 + 12                                              CSL03740
```

```
      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
      IF(IREAD.EQ.1) GO TO 400
C
C     REGENERATE THE ALPHA AND BETA
C
  330 MOLD1 = 1
C
C     SET KMAX
      DO 340 J = 1,NGOOD
      IF(MP(J).EQ.1) GO TO 360
  340 CONTINUE
      KMAX = MEV + 12
      WRITE(6,350) KMAX
  350 FORMAT(/' ALL EIGENVALUES FOR WHICH EIGENVECTORS ARE TO BE COMPUTE
     1D ARE EITHER T-MULTIPLE OR CLOSE TO'/' A SPURIOUS EIGENVALUE. THER
     1EFORE SET KMAX = MEV + 12 = ',I7)
      GO TO 380
C
  360 KMAXN = 11*MEV/8 + 12
      IF(MBETA.LE.KMAXN) GO TO 1530
      WRITE(6,370) KMAXN
  370 FORMAT(' SET KMAX EQUAL TO ',I6)
      KMAX = KMAXN
C
  380 WRITE(6,390) MOLD1,KMAX
  390 FORMAT(/' LANCZS SUBROUTINE GENRATES ALPHA(J), BETA(J+1), J =',
     1 I6,' TO ', I6/)
C
C-----------------------------------------------------------------------
C
      CALL LANCZS(CMATV,V1,V2,ALPHA,BETA,GR,GC,G,KMAX,MOLD1,N,SVSEED)
C
C-----------------------------------------------------------------------
C
  400 CONTINUE
C
C     SIMPLE STURM SEQUENCING IS NOT VALID FOR COMPLEX SYMMETRIC
C     MATRICES.  THUS, THE STRATEGY USED HERE FOR SELECTING
C     APPROPRIATE SIZE T-MATRICES FOR THE EIGENVECTOR COMPUTATIONS
C     MUST BE DIFFERENT FROM THAT USED IN THE REAL SYMMETRIC,
C     HERMITIAN, AND SINGULAR VALUE CASES.  AS IN THOSE CASES,
C     FOR EACH EIGENVALUE, A FIRST GUESS IS SELECTED AND THEN
C     LOOPING ON THE SIZE OF THE T-EIGENVECTOR COMPUTATIONS
C     DETERMINES APPROPRIATE SIZES FOR THE EIGENVECTOR COMPUTATIONS.
```

```
C     FIRST GUESSES AT APPROPRIATE SIZES ARE SPECIFIED BELOW.
C
      DO 430 J = 1,NGOOD
      EVAL = GOODEV(J)
C     COMPUTE A FIRST GUESS ON AN APPROPRIATE SIZE T-MATRIX EACH
C     EIGENVALUE.
      IF(MP(J).GT.1) GO TO 410
C     EIGENVALUE IS T-SIMPLE
      IF(MP(J).EQ.MONE) GO TO 420
C     EIGENVALUE IS T-SIMPLE AND T-ISOLATED
      MA(J) = (8*MEV)/9  +  1
      ML(J) = ((11*MEV)/8 + 12)
      GO TO 430
C     EIGENVALUE IS T-MULTIPLE
  410 MA(J) = (5*MEV)/(4*MP(J))  +  1
      ML(J) = (7*MEV)/(4*MP(J)) + 1
      GO TO 430
C     EIGENVALUE IS T-SIMPLE AND NOT T-ISOLATED
  420 MA(J) = (5*MEV)/8 + 1
      ML(J) = MEV
  430 CONTINUE
C
      IF (IWRITE.EQ.1) WRITE(6,440) (MA(JJ), JJ=1,NGOOD)
  440 FORMAT(/' 1ST GUESS AT APPROPRIATE SIZES FOR T-MATRICES '/
     1' ACTUAL VALUES WILL PROBABLY BE 1/4 AGAIN AS MUCH'/(13I6))
C
      WRITE(10,450) N,KMAX
  450 FORMAT(2I8,' = ORDER OF USER MATRIX AND MAX ORDER OF T(1,MEV)')
C
      WRITE(10,460)
  460 FORMAT(/' 1ST GUESS AT APPROPRIATE SIZES FOR T-MATRICES '/
     1' ACTUAL VALUES WILL PROBABLY BE 1/4 AGAIN AS MUCH'/)
      WRITE(10,470)
  470 FORMAT(5X,'J',8X,'REAL(GOODEV)',8X,'IMAG(GOODEV)',7X,'MA(J)',
     17X,'MP(J)')
C
      WRITE(10,480) (J,GOODEV(J), MA(J), MP(J), J=1,NGOOD)
  480 FORMAT(I6,2E20.12,I12,I12)
C
      IF(MBOUND.EQ.1) WRITE(10,490)
  490 FORMAT(/' GOODEV(J) IS A GOOD EIGENVALUE OF T(1,MEV)'/
     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
C     TERMINATE AFTER COMPUTING 1ST GUESSES ON SIZES OF T-MATRICES
C     REQUIRED FOR THE GIVEN EIGENVALUES?
      IF(MBOUND.EQ.1) GO TO 1390
C
C
C     IS THERE ROOM FOR ALL OF THE REQUESTED T-EIGENVECTORS?
      MTOL = 0
      DO 500 J = 1,NGOOD
```

```
      MTOL = MTOL + IABS(MA(J))                                         CSL04850
  500 CONTINUE                                                          CSL04860
      MTOL = (5*MTOL)/4                                                 CSL04870
      IF(MTOL.GT.MDIMTV.AND.NTVCON.EQ.0) GO TO 1410                     CSL04880
C                                                                       CSL04890
C-----------------------------------------------------------------------CSL04900
C     GENERATE A RANDOM VECTOR TO BE USED REPEATEDLY BY                 CSL04910
C     SUBROUTINE INVERM                                                 CSL04920
C                                                                       CSL04930
      ILL = RHSEED                                                      CSL04940
      CALL GENRAN(ILL,G,KMAX)                                           CSL04950
C                                                                       CSL04960
C-----------------------------------------------------------------------CSL04970
C                                                                       CSL04980
      DO 510 I = 1,KMAX                                                 CSL04990
  510 GR(I) = G(I)                                                      CSL05000
C                                                                       CSL05010
C-----------------------------------------------------------------------CSL05020
C                                                                       CSL05030
      CALL GENRAN(ILL,G,KMAX)                                           CSL05040
C                                                                       CSL05050
C-----------------------------------------------------------------------CSL05060
C                                                                       CSL05070
      DO 520 I = 1,KMAX                                                 CSL05080
  520 GC(I) = G(I)                                                      CSL05090
C                                                                       CSL05100
C     FOR EACH EIGENVALUE LOOP ON T-EIGENVECTOR COMPUTATIONS TO         CSL05110
C     COMPUTE AN APPROPRIATE T-EIGENVECTOR TO USE IN THE RITZ           CSL05120
C     VECTOR COMPUTATIONS.                                              CSL05130
C                                                                       CSL05140
      MTOL = 0                                                          CSL05150
      NTVEC = 0                                                         CSL05160
      DO 690 J = 1,NGOOD                                                CSL05170
      ICOUNT = 0                                                        CSL05180
      TFLAG = 0                                                         CSL05190
      ERRMIN = 10.D0                                                    CSL05200
      MABEST = MPMIN                                                    CSL05210
      IF(MP(J).EQ.MPMIN) GO TO 690                                      CSL05220
      EVAL = GOODEV(J)                                                  CSL05230
  530 KMAXU = IABS(MA(J))                                               CSL05240
C     SELECT A SUITABLE INCREMENT FOR THE ORDERS OF T-MATRICES          CSL05250
C     TO BE CONSIDERED IN DETERMINING APPROPRIATE SIZES FOR THE RITZ    CSL05260
C     VECTOR COMPUTATIONS                                               CSL05270
      IF(ICOUNT.GT.0) GO TO 560                                         CSL05280
C     SELECT IDELTA(J) BASED UPON THE MULTIPLICITY IN T(1,MEV)          CSL05290
      IF(MP(J).GT.1) GO TO 540                                          CSL05300
      IF(MP(J).LT.0) GO TO 550                                          CSL05310
C     MP(J) = 1, INITIAL MA(J) = 8*MEV/9 + 1                            CSL05320
      IDELTA(J) = (ML(J) - IABS(MA(J)))/10  + 1                         CSL05330
      GO TO 560                                                         CSL05340
C     MULTIPLE T-EIGENVALUE: INITIAL MA(J) = 5*MEV/4*MP + 1             CSL05350
  540 IDELTA(J) = (ML(J) - IABS(MA(J)))/10  +  1                        CSL05360
      GO TO 560                                                         CSL05370
C     T-SIMPLE EVALUE, NEAR SPURIOUS ONE, INITIAL MA(J) = 5*MEV/8 + 1   CSL05380
  550 IDELTA(J) = (ML(J) - IABS(MA(J)))/10  +  1                        CSL05390
```

```
  560 ICOUNT = ICOUNT + 1
      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 700
C
      IT = 3
      KINT = MTOL - KMAXU +1
C
C     RECORD THE BEGINNING AND END OF THE T-EIGENVECTOR BEING COMPUTED
      MINT(J) = KINT
      MFIN(J) = MTOL
C
C-----------------------------------------------------------------------
C     SUBROUTINE INVERM DOES INVERSE ITERATION, I.E. SOLVES
C     (T(1,KMAXU) - EVAL)*U = RHS  FOR EACH EIGENVALUE TO OBTAIN THE
C     DESIRED T-EIGENVECTOR.
C
      IF(IWRITE.EQ.1)  WRITE(6,570) J
  570 FORMAT(/I6,'TH EIGENVALUE')
C
      CALL INVERM(ALPHA,BETA,V1,TVEC(KINT),EVAL,ERROR,TERROR,EPSM,
     1    GR,GC,INTERC,KMAXU,IT,IWRITE)
C
C-----------------------------------------------------------------------
C
      TERR(J) = TERROR
      TLAST(J) = ERROR
      KMAXU1 = KMAXU + 1
      TBETA(J) = CDABS(BETA(KMAXU1))*ERROR
C
C     AFTER COMPUTING EACH OF THE T-EIGENVECTORS,
C     CHECK THE SIZE OF THE ERROR ESTIMATE, ERROR.
C     IF THIS ESTIMATE IS NOT AS SMALL AS DESIRED AND
C     |MA(J)| < ML(J), ATTEMPT TO INCREASE THE SIZE OF |MA(J)|
C     AND REPEAT THE T-EIGENVECTOR COMPUTATIONS.
C
      IF(ERROR.LT.ERTOL.OR.TFLAG.EQ.1)  GO TO 680
C
      IF(ERROR.GE.ERRMIN) GO TO 580
C     LAST COMPONENT IS LESS THAN MINIMAL TO DATE
      ERRMIN = ERROR
      MABEST = MA(J)
  580 CONTINUE
C
      IF(MA(J).GT.0)  ITEST = MA(J) + IDELTA(J)
      IF(MA(J).LT.0)  ITEST = -(IABS(MA(J)) + IDELTA(J))
      IF(IABS(ITEST).LE.ML(J).AND.ICOUNT.LE.10) GO TO 600
C     NEW MA(J) IS GREATER THAN MAXIMUM ALLOWED.
      IF(ERCONT.EQ.0.OR.MABEST.EQ.MPMIN) GO TO 620
      TFLAG = 1
      MA(J) = MABEST
      MTOL = MTOL - KMAXU
      WRITE(6,590) MA(J)
```

```
  590 FORMAT(' 10 ORDERS WERE CONSIDERED.  NONE SATISFIED THE ERROR TESTCSL05950
     1'/' THEREFORE USE THE BEST ORDER OBTAINED FOR THE EIGENVECTORS'   CSL05960
     1,I6)                                                              CSL05970
      GO TO 530                                                         CSL05980
C                                                                       CSL05990
  600 MA(J) = ITEST                                                     CSL06000
C                                                                       CSL06010
      MT = IABS(MA(J))                                                  CSL06020
      IF(IWRITE.EQ.1) WRITE(6,610) MT                                   CSL06030
  610 FORMAT(/' CHANGE SIZE OF T-MATRIX TO',I6,' RECOMPUTE T-EIGENVECTORCSL06040
     1')                                                                CSL06050
C                                                                       CSL06060
      MTOL = MTOL - KMAXU                                               CSL06070
C                                                                       CSL06080
      GO TO 530                                                         CSL06090
C                                                                       CSL06100
C       APPROPRIATE SIZE T-MATRIX WAS NOT OBTAINED                      CSL06110
  620 CONTINUE                                                          CSL06120
      WRITE(10,630) J,EVAL,MP(J)                                        CSL06130
  630 FORMAT(/' ON 10 INCREMENTS NOT ABLE TO IDENTIFY APPROPRIATE SIZE  CSL06140
     1T-MATRIX FOR'/                                                    CSL06150
     1' EIGENVALUE(',I4,') =',2E20.12,' T-MULTIPLICITY =',I4/)          CSL06160
      IF(MP(J).GT.1) WRITE(10,640)                                      CSL06170
      IF(MP(J).LT.0)  WRITE(10,650)                                     CSL06180
      IF(MP(J).EQ.1) WRITE(10,660)                                      CSL06190
  640 FORMAT(/' ORDERS TESTED RANGED FROM (5*MEV/4*MP(J)) TO APPROXIMATECSL06200
     1LY'/'  (7*MEV)/(4*MP(J)'/)                                        CSL06210
  650 FORMAT(/' ORDERS TESTED RANGED FROM (5*MEV/8)  TO MEV'/)          CSL06220
  660 FORMAT(/' ORDERS TESTED RANGED FROM 8*MEV/9 TO APPROXIMATELY 11*MECSL06230
     1V/8'/)                                                            CSL06240
      WRITE(10,670)                                                     CSL06250
  670 FORMAT(' ALLOWING LARGER ORDERS FOR THE T-MATRICES MAY RESULT IN  CSL06260
     1 SUCCESS'/' BUT PROBABLY WILL NOT.  PROBLEM IS PROBABLY DUE TO'   CSL06270
     1 /' LACK OF CONVERGENCE OF GIVEN EIGENVALUE, CHECK THE ERROR ESTIMCSL06280
     1ATE')                                                             CSL06290
      MP(J) = MPMIN                                                     CSL06300
      MTOL = MTOL - KMAXU                                               CSL06310
      GO TO 690                                                         CSL06320
  680 NTVEC = NTVEC + 1                                                 CSL06330
C                                                                       CSL06340
  690 CONTINUE                                                          CSL06350
      NGOODC = NGOOD                                                    CSL06360
      GO TO 720                                                         CSL06370
C                                                                       CSL06380
C     COME HERE IF THERE IS NOT ENOUGH ROOM FOR ALL OF T-EIGENVECTORS   CSL06390
  700 NGOODC = J-1                                                      CSL06400
      WRITE(6,710) J,MTOL,MDIMTV                                        CSL06410
  710 FORMAT(/' NOT ENOUGH ROOM IN TVEC ARRAY FOR ',I4,'TH T-EIGENVECTORCSL06420
     1'/' TVEC-DIMENSION REQUESTED = ',I6,' BUT TVEC HAS DIMENSION ',I6/CSL06430
     1)                                                                 CSL06440
      IF(NGOODC.EQ.0) GO TO 1430                                        CSL06450
      MTOL = MTOL-KMAXU                                                 CSL06460
C                                                                       CSL06470
  720 CONTINUE                                                          CSL06480
C                                                                       CSL06490
```

```
C     THE LOOP ON T-EIGENVECTOR COMPUTATIONS IS COMPLETE.
C     WRITE OUT THE SIZE T-MATRICES THAT WILL BE USED FOR
C     THE RITZ VECTOR COMPUTATIONS.
C
      WRITE(10,730)
  730 FORMAT(/' SIZES OF T-MATRICES THAT WILL BE USED IN THE RITZ COMPUT
     1ATIONS'/5X,'J',13X,'REAL(GOODEV)',13X,'IMAG(GOODEV)',1X,'MA(J)')
C
      WRITE(10,740)   (J,GOODEV(J),MA(J), J=1,NGOOD)
  740 FORMAT(I6,2E25.14,I6)
      WRITE(10,490)
C
      WRITE(6,750) MTOL
  750 FORMAT(/' THE CUMULATIVE LENGTH OF THE T-EIGENVECTORS IS',I18)
C
      WRITE(6,760) NTVEC,NGOOD
  760 FORMAT(/I6,' T-EIGENVECTORS OUT OF',I6,' REQUESTED WERE COMPUTED')
C
C     SAVE THE T-EIGENVECTORS ON FILE 11?
      IF(TVSTOP.NE.1.AND.SVTVEC.EQ.0) GO TO 820
C
      WRITE(11,770) NTVEC,MTOL,MATNO,SVSEED
  770 FORMAT(I6,3I12,' = NTVEC,MTOL,MATNO,SVSEED')
C
      DO 800 J=1,NGOODC
C     IF MP(J) = MPMIN THEN NO SUITABLE T-EIGENVECTOR IS AVAILABLE
C     FOR THAT EIGENVALUE.
      IF(MP(J).EQ.MPMIN) WRITE(11,780) J,MA(J),GOODEV(J),MP(J)
  780 FORMAT(2I6,2E20.12,I6/' TH EIGVAL,T-SIZE,EVALUE,FLAG,NO EIGVEC')
      IF(MP(J).NE.MPMIN) WRITE(11,790) J,MA(J),GOODEV(J),MP(J)
  790 FORMAT(I6,I6,2E20.12,I6/' T-EIGVEC,SIZE T,EVALUE OF A,MP(J)')
      IF(MP(J).EQ.MPMIN) GO TO 800
      KI = MINT(J)
      KF = MFIN(J)
C
      WRITE(11,260) (TVEC(K), K=KI,KF)
C
  800 CONTINUE
C
      IF(TVSTOP.NE.1) GO TO 820
C
      WRITE(6,810) TVSTOP, NTVEC,NGOOD
  810 FORMAT(/' USER SET TVSTOP = ',I1/
     1' THEREFORE PROGRAM TERMINATES AFTER T-EIGENVECTOR COMPUTATIONS'/
     1' T-EIGENVECTORS THAT WERE COMPUTED ARE SAVED ON FILE 11'/
     1I8,' T-EIGENVECTORS WERE COMPUTED OUT OF',I7,' REQUESTED'/)
C
      GO TO 1550
C
  820 CONTINUE
C     IF NOT ABLE TO COMPUTE ALL THE REQUESTED T-EIGENVECTORS
C     CONTINUE WITH THE LANCZOS VECTOR COMPUTATIONS ANYWAY?
C
      IF(NTVEC.NE.NGOOD.AND.LVCONT.EQ.0) GO TO 1450
C
```

```
C     COMPUTE THE MAXIMUM SIZE OF THE T-MATRIX USED FOR THOSE          CSL07050
C     EIGENVALUES WITH GOOD ERROR ESTIMATES.                           CSL07060
C                                                                      CSL07070
      KMAXU = 0                                                        CSL07080
      DO 830 J = 1,NGOODC                                              CSL07090
      MT = IABS(MA(J))                                                 CSL07100
      IF(MT.LT.KMAXU.OR.MP(J).EQ.MPMIN) GO TO 830                      CSL07110
      KMAXU = MT                                                       CSL07120
  830 CONTINUE                                                         CSL07130
C                                                                      CSL07140
      IF(KMAXU.EQ.0) GO TO 1490                                        CSL07150
C                                                                      CSL07160
      WRITE(6,840) KMAXU                                               CSL07170
  840 FORMAT(/I6,' = LARGEST SIZE T-MATRIX TO BE USED IN THE RITZ VECTORCSL07180
     1 COMPUTATIONS')                                                  CSL07190
C                                                                      CSL07200
C     COUNT THE NUMBER OF RITZ VECTORS NOT BEING COMPUTED              CSL07210
      MREJEC = 0                                                       CSL07220
      DO 850 J=1,NGOODC                                                CSL07230
  850 IF(MP(J).EQ.MPMIN)  MREJEC = MREJEC + 1                          CSL07240
      MREJET = MREJEC + (NGOOD-NGOODC)                                 CSL07250
      IF(MREJET.NE.0) WRITE(6,860) MREJET                              CSL07260
  860 FORMAT(/' RITZ VECTORS ARE NOT COMPUTED FOR',I6,' OF THE EIGENVALUCSL07270
     1ES'/)                                                            CSL07280
      NACT = NGOODC - MREJEC                                           CSL07290
      WRITE(6,870) NGOOD,NTVEC,NACT                                    CSL07300
  870 FORMAT(/I6,' RITZ VECTORS WERE REQUESTED'/I6,' T-EIGENVECTORS WERECSL07310
     1COMPUTED'/I6,' RITZ VECTORS WILL BE COMPUTED'/)                  CSL07320
C     CHECK IF THERE ARE ANY RITZ VECTORS TO COMPUTE                   CSL07330
      IF(MREJEC.EQ.NGOODC) GO TO 1470                                  CSL07340
C                                                                      CSL07350
C     CONTINUE WITH THE LANCZOS VECTOR COMPUTATIONS?                   CSL07360
      IF(LVCONT.EQ.0.AND.MREJEC.NE.0) GO TO 1450                       CSL07370
C                                                                      CSL07380
C     NOW COMPUTE THE RITZ VECTORS.  REGENERATE THE                    CSL07390
C     LANCZOS VECTORS.                                                 CSL07400
C                                                                      CSL07410
      DO 880 I = 1,NMAX                                                CSL07420
  880 RITVEC(I) = ZEROC                                                CSL07430
C                                                                      CSL07440
C-----------------------------------------------------------------------CSL07450
C     REGENERATE THE STARTING VECTOR. THIS MUST BE GENERATED AND       CSL07460
C     NORMALIZED PRECISELY THE WAY IT WAS DONE IN THE EIGENVALUE       CSL07470
C     COMPUTATIONS, OTHERWISE THERE WILL BE A MISMATCH BETWEEN         CSL07480
C     THE T-EIGENVECTORS THAT HAVE BEEN COMPUTED FROM THE T-MATRICES   CSL07490
C     READ IN FROM FILE 2 AND THE LANCZOS VECTORS THAT ARE             CSL07500
C     BEING REGENERATED.                                               CSL07510
C                                                                      CSL07520
      IIL = SVSEED                                                     CSL07530
      CALL GENRAN(IIL,G,N)                                             CSL07540
C                                                                      CSL07550
C-----------------------------------------------------------------------CSL07560
C                                                                      CSL07570
      DO 890 I = 1,N                                                   CSL07580
  890 GR(I) = G(I)                                                     CSL07590
```

```
C
C-----------------------------------------------------------------------
C
      CALL GENRAN(IIL,G,N)
C
C-----------------------------------------------------------------------
C
      DO 900 I = 1,N
  900 GC(I) = G(I)
C
      DO 910 I = 1,N
  910 V2(I) = DCMPLX(GR(I),GC(I))
C
C-----------------------------------------------------------------------
      CALL INPRDC(V2,V2,SUMC,N)
C-----------------------------------------------------------------------
C
      SUMC = ONE/CDSQRT(SUMC)
      DO 920 I = 1,N
      V1(I) = ZEROC
  920 V2(I) = V2(I)*SUMC
C
C     LOOP FOR GENERATING REQUIRED RITZ VECTORS (IVEC = 1,KMAXU)
C
      IVEC = 1
      BATA = ZEROC
C
      GO TO 980
C
  930 CONTINUE
C
C-----------------------------------------------------------------------
C
C    CMATV(V2,V1,BATA) CALCULATES  V1 = A*V2 - BATA*V1
      CALL CMATV(V2,V1,BATA)
      CALL INPRDC(V2,V1,ALFA,N)
C
C-----------------------------------------------------------------------
C
      DO 940 J=1,N
  940 V1(J) = V1(J)-ALFA*V2(J)
C
C-----------------------------------------------------------------------
      CALL INPRDC(V1,V1,BATA,N)
C-----------------------------------------------------------------------
C
      BATA = CDSQRT(BATA)
      SUMC = ONE/BATA
C
      TEMPC = BETA(IVEC)
      TEMP = CDABS(BATA - TEMPC)/CDABS(TEMPC)
      IF (TEMP.LT.1.0D-10)GO TO 960
C
C     IF THE BETA BEING REGENERATED DO NOT MATCH THE HISTORY FILE
C     THEN SOMETHING IS WRONG IN THE LANCZOS VECTOR GENERATION
```

```
C     AND PROGRAM TERMINATES FOR USER TO CORRECT THE PROBLEM
C     WHICH MUST BE IN THE STARTING VECTOR GENERATION OR IN
C     THE MATRIX-VECTOR MULTIPLY SUBROUTINE CMATV SUPPLIED.
C     THIS PART OF THE COMPUTATIONS MUST BE IDENTICAL TO THE
C     CORRESPONDING PART IN THE EIGENVALUE COMPUTATIONS.
C
      WRITE(6,950) IVEC,BATA,BETA(IVEC),TEMP
  950 FORMAT(/2X,'IVEC',16X,'BATA',10X,'BETA(IVEC)',14X,'RELDIF'/I6,
     13E20.12/' IN LANCZOS VECTOR REGENERATION THE ENTRIES OF THE TRIDIA
     1GONAL MATRICES BEING'/' GENERATED ARE NOT THE SAME AS THOSE IN THE
     1 MATRIX SUPPLIED ON FILE 2.'/' THEREFORE SOMETHING IS BEING INITIA
     1LIZED OR COMPUTED DIFFERENTLY FROM THE WAY'/' IT WAS COMPUTED IN T
     1HE EIGENVALUE COMPUTATIONS'/' THE PROGRAM TERMINATES FOR THE USER
     1TO DETERMINE WHAT THE PROBLEM IS'/)
      GO TO 1550
C
C
  960 CONTINUE
      DO 970 J = 1,N
      TEMPC = SUMC*V1(J)
      V1(J) = V2(J)
  970 V2(J) = TEMPC
C
  980 CONTINUE
C
      LFIN = 0
      DO 1000 J = 1,NGOODC
      LL = LFIN
      LFIN = LFIN + N
C
      IF(IABS(MA(J)).LT.IVEC.OR.MP(J).EQ.MPMIN) GO TO 1000
      II = IVEC + MINT(J) - 1
      TEMPC = 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) = TEMPC*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 THE
C     CORRESPONDING 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                                     CSL08700
C                                                                       CSL08710
      DO 1010 K = 1,N                                                   CSL08720
      KK = KK + 1                                                       CSL08730
 1010 V2(K) = RITVEC(KK)                                                CSL08740
C                                                                       CSL08750
C-----------------------------------------------------------------------CSL08760
      CALL INPRDC(V2,V2,SUMC,N)                                         CSL08770
C-----------------------------------------------------------------------CSL08780
C                                                                       CSL08790
      SUMC = CDSQRT(SUMC)                                               CSL08800
      RNORM(J) = CDABS(SUMC)                                            CSL08810
      TEMP = DABS(ONE-RNORM(J))                                         CSL08820
      SUMC = DCMPLX(ONE,ZERO)/SUMC                                      CSL08830
C                                                                       CSL08840
      KK = LFIN - N                                                     CSL08850
      DO 1020 K = 1,N                                                   CSL08860
      KK = KK + 1                                                       CSL08870
      V2(K) = SUMC*V2(K)                                                CSL08880
 1020 RITVEC(KK) = V2(K)                                                CSL08890
C                                                                       CSL08900
C     COMPUTE THE 'REAL' NORM                                           CSL08910
C                                                                       CSL08920
C-----------------------------------------------------------------------CSL08930
      CALL CINPRD(V2,V2,SUM,N)                                          CSL08940
C-----------------------------------------------------------------------CSL08950
C                                                                       CSL08960
      IF (IWRITE.NE.0) WRITE(6,1030) J,GOODEV(J)                        CSL08970
 1030 FORMAT(/I5,' TH EIGENVALUE CONSIDERED = ',2E20.12/)               CSL08980
C                                                                       CSL08990
      IF (IWRITE.NE.0) WRITE(6,1040) TERR(J),TBETA(J),RNORM(J),SUM      CSL09000
 1040 FORMAT(' NORM OF ERROR IN T-EIGENVECTOR = ',E14.3/                CSL09010
     1 ' CDABS(BETA(MA(J)+1)*U(MA(J))) ',E14.3/                         CSL09020
     1 ' CDABS(EUCLIDEAN-NORM(RITVEC))  = ',E14.3/                      CSL09030
     1 ' HERMITIAN-NORM(RITVEC)**2  = ',E14.3/)                         CSL09040
C                                                                       CSL09050
      LINT = LFIN - N + 1                                               CSL09060
      EVAL = GOODEV(J)                                                  CSL09070
C                                                                       CSL09080
C-----------------------------------------------------------------------CSL09090
C                                                                       CSL09100
      CALL CMATV(RITVEC(LINT),V2,EVAL)                                  CSL09110
C                                                                       CSL09120
C-----------------------------------------------------------------------CSL09130
C                                                                       CSL09140
C     COMPUTE ERROR IN RITZ VECTOR CONSIDERED AS A EIGENVECTOR OF A.    CSL09150
C     V2 = A*RITVEC - EVAL*RITVEC                                       CSL09160
C                                                                       CSL09170
C-----------------------------------------------------------------------CSL09180
      CALL CINPRD(V2,V2,SUM,N)                                          CSL09190
C-----------------------------------------------------------------------CSL09200
C                                                                       CSL09210
      SUM = DSQRT(SUM)                                                  CSL09220
      ERR(J) = SUM                                                      CSL09230
      GAP = ABS(AMINGP(J))                                              CSL09240
```

```
      ERRDGP(J) = SUM/GAP
C
 1050 CONTINUE
C
C
C     RITZVECTORS ARE NORMALIZED AND ERROR ESTIMATES ARE IN ERR ARRAY
C     AND IN ERRDGP ARRAY. STORE EVERYTHING
C
C
      WRITE(9,1060)
 1060 FORMAT(3X,'REAL(GOODEV)',3X,'IMAG(GOODEV)',1X,'MA(J)',7X,'AMINGAP'
     1 ,4X,'AERROR',2X,'AERR/GAP',4X,'TERROR')
C
      WRITE(13,1070)
 1070 FORMAT(8X,'REAL(GOODEV)',8X,'IMAG(GOODEV)',2X,'RITZNORM',3X,'AMING
     1AP',2X,'TBETA(J)',2X,'TLAST(J)')
C
      DO 1100 J=1,NGOODC
C
      IF(MP(J).EQ.MPMIN) GO TO 1100
C
      WRITE(9,1080)GOODEV(J),MA(J),AMINGP(J),ERR(J),ERRDGP(J),TERR(J)
 1080 FORMAT(2E15.8,I6,E14.6,3E10.3)
C
      WRITE(13,1090) GOODEV(J),RNORM(J),AMINGP(J),TBETA(J),TLAST(J)
 1090 FORMAT(2E20.12,4E10.3)
C
 1100 CONTINUE
C
      IF(MREJEC.EQ.0) GO TO 1180
      WRITE(9,1110)
 1110 FORMAT(/' RITZ VECTORS WERE NOT COMPUTED FOR THE FOLLOWING EIGENVA
     1LUES'/' EITHER BECAUSE THEY HAD NOT CONVERGED OR BECAUSE THE ERROR
     1 ESTIMATE'/'  WAS NOT AS SMALL AS DESIRED'/)
C
      WRITE(13,1120)
 1120 FORMAT(/' RITZ VECTORS WERE NOT COMPUTED FOR THE FOLLOWING EIGENVA
     1LUES'/' EITHER BECAUSE THEY HAD NOT CONVERGED OR BECAUSE'/' THE ER
     1ROR ESTIMATE WAS NOT AS SMALL AS DESIRED'/)
C
      DO 1170 J = 1,NGOODC
      IF(MP(J).NE.MPMIN) GO TO 1170
C     WRITE OUT MESSAGE FOR EACH EIGENVALUE FOR WHICH NO EIGENVECTOR
C     WAS COMPUTED.
C
      WRITE(9,1130)
 1130 FORMAT(6X,'GOODEV(J)',3X,'MA(J)',5X,'AMINGP(J)',6X,'TLAST(J)',3X,
     1'MP(J)')
      WRITE(9,1140) GOODEV(J),MA(J),AMINGP(J),TBETA(J),MP(J)
 1140 FORMAT(2E15.8,I8,2E14.4,I8)
C
      WRITE(13,1150)
 1150 FORMAT(6X,'REAL(GOODEV(J))',6X,'IMAG(GOODEV(J))',4X,'MA(J)',3X,
     1'MP(J)')
      WRITE(13,1160) GOODEV(J),MA(J),MP(J)
```

```
 1160 FORMAT(2E15.8,2I8)
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. A MINGAP = GAP TO NEAREST A-EIGENVALUE'//)
C
      WRITE(13,1200)
 1200 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) WHERE'
     1 /' T = T(1,MA(J))    X = RITZ VECTOR = V*Y  V = SUCCESSIVE '/
     1 ' LANCZOS VECTORS. A MINGAP = GAP TO NEAREST A-EIGENVALUE'/
     1 ' AERROR AND TERROR ARE GIVEN IN FILE 9. RNORM = NORM(X)'/
     1 ' BETA(M+1)*ABS(Y(M)) IS AN ESTIMATOR OF NORM(A*X-EV*X)'//)
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,2E20.12,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)
C1240 FORMAT(4Z20)
 1240 FORMAT(2(2E20.12))
      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
 1270 CONTINUE
C
C     DID ANY T-MATRICES INCLUDE OFF-DIAGONAL ENTRIES SMALLER THAN
C     DESIRED, AS SPECIFIED BY BTOL?
C
      IF(IB.GT.0) GO TO 1300
      WRITE(6,1280) KMAXU
 1280 FORMAT(/' FOR LARGEST T-MATRIX CONSIDERED',I7,' CHECK THE SIZE OF
```

```
     1BETAS')
C
C-----------------------------------------------------------------------
C
      CALL TNORM(ALPHA,BETA,BKMIN,TEMP,KMAXU,IBMT)
C
C-----------------------------------------------------------------------
C
      IF(IBMT.LT.0) WRITE (6,1290)
 1290 FORMAT(/' WARNING THE T-MATRICES FOR ONE OR MORE OF THE EIGENVALUE
     1S CONSIDERED'/' HAD AN OFF-DIAGONAL ENTRY THAT WAS SMALLER THAN TH
     1E BETA TOLERANCE THAT WAS SPECIFIED'/)
 1300 CONTINUE
C
       GO TO 1550
C
 1310 WRITE(6,1320) NGOOD,NMAX,MDIMRV
 1320 FORMAT(/I4,' RITZ VECTORS WERE REQUESTED BUT THE REQUIRED DIMENSIO
     1N',I6/' IS LARGER THAN THE USER-SPECIFIED DIMENSION OF RITVEC',I6
     1/' THEREFORE, THE EIGENVECTOR PROCEDURE TERMINATES FOR THE USER TO
     1 INTERVENE')
C
      GO TO 1550
C
 1330 WRITE(6,1340) NOLD,N,MATOLD,MATNO
 1340 FORMAT(/' PARAMETERS READ FROM FILE 3 DO NOT AGREE WITH USER-SPECI
     1FIED'/' PARAMETERS, NOLD,N,MATOLD,MATNO = '/2I6,2I12/
     1' THEREFORE PROGRAM TERMINATES FOR USER TO RESOLVE DIFFERENCES'/)
C
      GO TO 1550
C
 1350 WRITE(6,1360)
 1360 FORMAT(/' PARAMETERS IN ALPHA,BETA FILE READ IN DO NOT AGREE WITH
     1 THOSE'/' SPECIFIED BY THE USER.  THEREFORE, THE PROGRAM TERMINATE
     1S FOR'/' THE USER TO RESOLVE THE DIFFERENCES'/)
C
      GO TO 1550
C
 1370 WRITE(6,1380) KMAX,MEV
 1380 FORMAT(/' IN ALPHA, BETA HISTORY HEADER KMAX =',I6/
     1' BUT EIGENVALUES WERE COMPUTED AT MEV = ',I6,' PROGRAM STOPS'/)
C
      GO TO 1550
C
 1390 WRITE(6,1400)
 1400 FORMAT(/' PROGRAM COMPUTED 1ST GUESSES AT T-MATRIX SIZES'/' READ T
     1HEM TO FILE 10, THEN TERMINATED AS REQUESTED.')
      GO TO 1550
C
 1410 WRITE(6,1420) MTOL, MDIMTV
 1420 FORMAT(/' PROGRAM TERMINATES BECAUSE THE MINIMAL TVEC DIMENSION AN
     1TICIPATED',I7/'  IS LARGER THAN THE TVEC DIMENSION',I7,' SPECIFIED
     1 BY THE USER.'/'  USER MAY RESET THE TVEC DIMENSION AND RESTART TH
     1E PROGRAM')
      GO TO 1550
```

```
C                                                                       CSL10900
 1430 WRITE(6,1440)                                                     CSL10910
 1440 FORMAT(/' PROGRAM TERMINATES BECAUSE NO SUITABLE T-EIGENVECTORS WECSL10920
     1RE IDENTIFIED'/' FOR ANY OF THE EIGENVALUES SUPPLIED.  PROBLEM COUCSL10930
     1LD BE CAUSED BY'/'  TOO SMALL A TVEC DIMENSION OR SIMPLY BE THAT TCSL10940
     1-EIGENVECTORS COULD'/'  NOT BE IDENTIFIED.  USER SHOULD CHECK OUTPCSL10950
     1UT'/)                                                             CSL10960
      GO TO 1550                                                        CSL10970
C                                                                       CSL10980
 1450 WRITE(6,1460) LVCONT,NTVEC,NGOOD                                  CSL10990
 1460 FORMAT(/' LVCONT FLAG =',I2,' AND NUMBER ',I5,' OF T-EIGENVECTORS CSL11000
     1 COMPUTED N.E.'/' NUMBER',I5,' REQUESTED SO PROGRAM TERMINATES'/) CSL11010
      GO TO 1550                                                        CSL11020
 1470 WRITE(6,1480)                                                     CSL11030
 1480 FORMAT(/' PROGRAM TERMINATES WITHOUT COMPUTING ANY RITZ VECTORS'/ CSL11040
     1' BECAUSE ALL T-EIGENVECTORS COMPUTED WERE REJECTED AS NOT SUITABLCSL11050
     1E'/' FOR THE RITZ VECTOR COMPUTATIONS.  PROBABLE CAUSE IS LACK OF CSL11060
     1'/' CONVERGENCE OF THE EIGENVALUES'/)                             CSL11070
       GO TO 1550                                                       CSL11080
C                                                                       CSL11090
 1490 WRITE(6,1500)                                                     CSL11100
 1500 FORMAT(/' PROGRAM INDICATES THAT IT IS NOT POSSIBLE TO COMPUTE ANYCSL11110
     1 OF THE'/' REQUESTED EIGENVECTORS. THEREFORE PROGRAM TERMINATES') CSL11120
      DO 1510 J=1,NGOODC                                                CSL11130
 1510 WRITE(6,1520)  J,GOODEV(J),MP(J)                                  CSL11140
 1520 FORMAT(/4X,' J',11X,'GOODEV(J)',4X,'MP(J)'/I6,2E20.12,I9)         CSL11150
      GO TO 1550                                                        CSL11160
C                                                                       CSL11170
 1530 WRITE(6,1540) MBETA,KMAXN                                         CSL11180
 1540 FORMAT(/' PROGRAM TERMINATES BECAUSE THE STORAGE ALLOTTED FOR THE CSL11190
     1BETA ARRAY',I8/' IS NOT SUFFICIENT FOR THE ENLARGED KMAX =',I8,' TCSL11200
     1HAT THE PROGRAM WANTS'/' USER CAN ENLARGE THE ALPHA AND BETA ARRAYCSL11210
     1S AND RERUN THE PROGRAM'/)                                        CSL11220
C                                                                       CSL11230
 1550 CONTINUE                                                          CSL11240
C                                                                       CSL11250
      STOP                                                              CSL11260
C-----END OF MAIN PROGRAM FOR COMPLEX SYMMETRIC EIGENVECTORS------------CSL11270
      END                                                               CSL11280
```

7.5 CSLEMULT: LANCZS and Sample Matrix-Vector Multiply Subroutines

```
C-----CSLEMULT--(COMPLEX SYMMETRIC MATRICES)----------------------------
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
C  are used in other scientific or engineering research works
C  the names of the authors of these codes and appropriate
C  references to their written work are to be incorporated in the
C  derivative works.
C
C  This header is not to be removed from these codes.
C
C         REFERENCE:  Cullum and Willoughby, Chapter 6,
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
C     CONTAINS SUBROUTINE LANCZS USED IN THE COMPLEX SYMMETRIC
C     VERSION OF THE LANCZOS PROCEDURES PLUS SAMPLE USPEC AND
C     CMATV SUBROUTINES.
C
C     PORTABILITY:
C     THESE PROGRAMS ARE NOT PORTABLE DUE TO THE USE OF COMPLEX*16
C     VARIABLES AND CORRESPONDING FUNCTIONS.  MOREOVER, THE PFORT
C     VERIFIER IDENTIFIED THE FOLLOWING ADDITIONAL NONPORTABLE
C     CONSTRUCTIONS:
C
C     1.  ENTRIES USED TO PASS THE STORAGE LOCATIONS OF THE
C         ARRAYS AND PARAMETERS NEEDED TO SPECIFY THE GIVEN MATRIX
C         FROM THE USPEC SUBROUTINE TO THE MATRIX-VECTOR MULTIPLY
C         SUBROUTINE CMATV.
C     2.  IN THE SAMPLE USPEC SUBROUTINES PROVIDED: THE FREE FORMAT
C         READ(8,*) AND THE FORMAT (20A4).  IN THE SAMPLE CMATV:
C         THE COMPUTATION OF INDICES:  IN THE AUXILIARY SUBROUTINE
C         USED FOR COMPUTING THE KNOWN EIGENVALUES OF TEST CLASS 2
C         MATRICES, THE DATA/MACHEP DEFINITION.
C
C-----LANCZS-COMPUTE THE LANCZOS TRIDIAGONAL MATRICES------------------
C
```

```
      SUBROUTINE LANCZS(MATVEC,V1,V2,ALPHA,BETA,
     1GR,GC,G,KMAX,MOLD1,N,IIX)
C
C-----------------------------------------------------------------------
      COMPLEX*16 V1(1), V2(1), BATA, ZEROC, TEMP, SUMC
      COMPLEX*16  ALPHA(1), BETA(1)
      DOUBLE PRECISION SUM, ONE, ZERO, GR(1), GC(1)
      REAL  G(1)
      EXTERNAL MATVEC
C     COMPLEX*16  CDSQRT, DCMPLX
C-----------------------------------------------------------------------
C
      ZERO = 0.D0
      ONE = 1.D0
      ZEROC = DCMPLX(ZERO,ZERO)
C
      IF(MOLD1.GT.1)GO TO 50
C
C    ALPHA/BETA GENERATION STARTS AT I = 1
C    MOLD1 = 1 SET V1 = 0. AND V2 = RANDOM UNIT VECTOR
      IIL=IIX
C
C-----------------------------------------------------------------------
      CALL GENRAN(IIL,G,N)
C-----------------------------------------------------------------------
C
      DO 10 I = 1,N
   10 GR(I) = G(I)
C
C-----------------------------------------------------------------------
      CALL GENRAN(IIL,G,N)
C-----------------------------------------------------------------------
C
      DO 20 I = 1,N
   20 GC(I) = G(I)
C
      DO 30 I = 1,N
   30 V2(I) = DCMPLX(GR(I),GC(I))
C
C-----------------------------------------------------------------------
      CALL INPRDC(V2,V2,SUMC,N)
C-----------------------------------------------------------------------
C
      SUMC = ONE/CDSQRT(SUMC)
      DO 40 I = 1,N
      V1(I) = ZEROC
   40 V2(I) = V2(I)*SUMC
      BETA(1) = ZEROC
C
C    ALPHA BETA GENERATION LOOP
   50 CONTINUE
C
      DO 80 I=MOLD1,KMAX
      SUMC = BETA(I)
C
```

```
C-----------------------------------------------------------------------CSL00980
C    MATVEC(V2,V1,SUMC) CALCULATES  V1 = A*V2 - SUMC*V1                 CSL00990
      CALL MATVEC(V2,V1,SUMC)                                           CSL01000
      CALL INPRDC(V2,V1,SUMC,N)                                         CSL01010
C-----------------------------------------------------------------------CSL01020
C                                                                       CSL01030
      ALPHA(I) = SUMC                                                   CSL01040
      DO 60 J=1,N                                                       CSL01050
   60 V1(J) = V1(J)-SUMC*V2(J)                                          CSL01060
C                                                                       CSL01070
C-----------------------------------------------------------------------CSL01080
      CALL INPRDC(V1,V1,SUMC,N)                                         CSL01090
C-----------------------------------------------------------------------CSL01100
C                                                                       CSL01110
      IN = I+1                                                          CSL01120
      BATA = CDSQRT(SUMC)                                               CSL01130
      BETA(IN) = BATA                                                   CSL01140
      SUMC = ONE/BATA                                                   CSL01150
      DO 70 J=1,N                                                       CSL01160
      TEMP = SUMC*V1(J)                                                 CSL01170
      V1(J) = V2(J)                                                     CSL01180
   70 V2(J) = TEMP                                                      CSL01190
   80 CONTINUE                                                          CSL01200
C     END ALPHA, BETA GENERATION LOOP                                   CSL01210
C                                                                       CSL01220
C-----END OF LANCZS-----------------------------------------------------CSL01230
C                                                                       CSL01240
      RETURN                                                            CSL01250
      END                                                               CSL01260
C                                                                       CSL01270
C-----USPEC, AND CMATV FOR COMPLEX SYMMETRIC TEST MATRICES 1------------CSL01280
C                                                                       CSL01290
C-----START OF USPEC-(COMPLEX SYMMETRIC TEST MATRICES 1)----------------CSL01300
C                                                                       CSL01310
C     SUBROUTINE CSPEC(N,MATNO)                                         CSL01320
      SUBROUTINE USPEC(N,MATNO)                                         CSL01330
C                                                                       CSL01340
C-----------------------------------------------------------------------CSL01350
      DOUBLE PRECISION  C0,C1,C2,HALF,ONE,SCR,SCI,ANGLE                 CSL01360
      COMPLEX*16 SC,TC,CL0,CL1                                          CSL01370
      REAL EXPLAN(20)                                                   CSL01380
      DOUBLE PRECISION DARCOS                                           CSL01390
C     COMPLEX*16 DCMPLX                                                 CSL01400
C-----------------------------------------------------------------------CSL01410
      HALF = 0.5D0                                                      CSL01420
      ONE  = 1.0D0                                                      CSL01430
C                                                                       CSL01440
C     READ IN PARAMETERS TO DEFINE MATRIX                               CSL01450
C     MATRIX IS COMPLEX DIAGONAL SIMILITARY TRANSFORM OF THE BLOCK      CSL01460
C     TOEPLITZ POISSON MATRICES USED TO TEST REAL SYMMETRIC MATRICES.   CSL01470
C     THE REAL POISSON MATRIX HAS SYMMETRIC TOEPLITZ BLOCKS ALONG THE   CSL01480
C     DIAGONAL.  EACH ONE OF THESE HAS THE PARAMETER C2 ALONG THE       CSL01490
C     DIAGONAL AND -C0 ABOVE AND BELOW THE DIAGONAL.  THE OFF-DIAGONAL  CSL01500
C     BLOCKS ARE DIAGONAL WITH DIAGONAL ENTRIES -C1.  EACH BLOCK IS     CSL01510
C     KX*KX AND THERE ARE KY BLOCKS.  A HERMITIAN VERSION IS OBTAINED   CSL01520
```

```
C     BY APPLYING A DIAGONAL SIMILARITY TRANSFORM TO THE ABOVE          CSL01530
C     MATRIX WHERE THE DIAGONAL MATRIX IS SUCH THAT ITS                 CSL01540
C     DIAGONAL ENTRIES ARE (SC)**(K-1), K=1,...,N-1.                    CSL01550
C     THIS HERMITIAN VERSION IS TURNED INTO A COMPLEX SYMMETRIC ONE     CSL01560
C     IN THE MATRIX VECTOR MULTIPLY BY TREATING THE BELOW DIAGONAL      CSL01570
C     ENTRIES AS BEING EQUAL TO THE ABOVE DIAGONAL ENTRIES RATHER       CSL01580
C     THAN THEIR COMPLEX CONJUGATES.                                    CSL01590
C                                                                       CSL01600
      READ(8,10) EXPLAN                                                 CSL01610
   10 FORMAT(20A4)                                                      CSL01620
      READ(8,*) NOLD,MATOLD                                             CSL01630
      WRITE(6,20) NOLD,MATOLD                                           CSL01640
   20 FORMAT(' ORDER OF MATRIX READ FROM FILE =',I6/' MATRIX NUMBER =', CSL01650
     1I8)                                                               CSL01660
C                                                                       CSL01670
C     TEST OF PARAMETER CORRECTNESS                                     CSL01680
      ITEMP = (NOLD-N)**2 + (MATNO-MATOLD)**2                           CSL01690
C                                                                       CSL01700
      IF(ITEMP.EQ.0) GO TO 40                                           CSL01710
C                                                                       CSL01720
      WRITE(6,30)                                                       CSL01730
   30 FORMAT(' PROGRAM TERMINATES BECAUSE EITHER ORDERS OF OR LABELS FORCSL01740
     1 MATRIX DISAGREE')                                                CSL01750
      GO TO 100                                                         CSL01760
C                                                                       CSL01770
   40 CONTINUE                                                          CSL01780
C                                                                       CSL01790
      READ(8,10) EXPLAN                                                 CSL01800
      READ(8,*)  C0,KX,KY                                               CSL01810
      READ(8,10) EXPLAN                                                 CSL01820
      READ(8,*) SCR                                                     CSL01830
      ANGLE = DARCOS(SCR)                                               CSL01840
      SCI = DSIN(ANGLE)                                                 CSL01850
      SC = DCMPLX(SCR,SCI)                                              CSL01860
      WRITE(6,50) SC                                                    CSL01870
      WRITE(9,50) SC                                                    CSL01880
   50 FORMAT(' GENERATOR OF DIAGONAL TRANSFORMATION ='/2E20.12)         CSL01890
C                                                                       CSL01900
      TC = SC                                                           CSL01910
      DO 60 J=2,KX                                                      CSL01920
   60 TC = SC*TC                                                        CSL01930
      WRITE(6,70) TC                                                    CSL01940
   70 FORMAT(' TC = ',2E20.12)                                          CSL01950
C                                                                       CSL01960
      N = KX*KY                                                         CSL01970
      C2 = ONE                                                          CSL01980
      C1 = HALF-C0                                                      CSL01990
      CL0 = -SC*C0                                                      CSL02000
      CL1 = -TC*C1                                                      CSL02010
C                                                                       CSL02020
      WRITE(6,80) N,KX,KY,C2,C0,C1                                      CSL02030
   80 FORMAT(/5X,'N',4X,'KX',4X,'KY',7X,'DIAGONAL',3X,'X-CODIAGONAL',   CSL02040
     1 3X,'Y-CODIAGONAL'/3I6,3E15.8/)                                   CSL02050
C                                                                       CSL02060
C-----------------------------------------------------------------------CSL02070
```

```
       CALL HMATVE(C2,CL0,CL1,KX,KY)                                    CSL02080
C-----------------------------------------------------------------------CSL02090
C                                                                       CSL02100
   90 CONTINUE                                                          CSL02110
      RETURN                                                            CSL02120
C                                                                       CSL02130
C-----END OF USPEC------------------------------------------------------CSL02140
  100 STOP                                                              CSL02150
      END                                                               CSL02160
C                                                                       CSL02170
C-----START OF CSMATV (FOR TEST MATRICES 1)----------------------------CSL02180
C     CALCULATE U = A*W - SUMC*U FOR COMPLEX SYMMETRIC MATRICES         CSL02190
C     HERE WE HAVE TAKEN A HERMITIAN VERSION OF POISSON MATRICES        CSL02200
C     AND TURNED IT INTO A COMPLEX SYMMETRIC TEST PROBLEM (WHOSE        CSL02210
C     EIGENVALUES WE DO NOT KNOW)                                       CSL02220
C                                                                       CSL02230
C     SUBROUTINE CSMATV(W,U,CSUM)                                       CSL02240
      SUBROUTINE CMATV(W,U,CSUM)                                        CSL02250
C                                                                       CSL02260
C-----------------------------------------------------------------------CSL02270
      DOUBLE PRECISION  C2                                              CSL02280
      COMPLEX*16 U(1),W(1)                                              CSL02290
      COMPLEX*16 CL0,CL1,CR0,CR1,CSUM                                   CSL02300
C-----------------------------------------------------------------------CSL02310
C                                                                       CSL02320
      N = KX*KY                                                         CSL02330
      KX1 = KX-1                                                        CSL02340
      KY1 = KY-1                                                        CSL02350
      CR0 = CL0                                                         CSL02360
      CR1 = CL1                                                         CSL02370
C                                                                       CSL02380
      KK = 1                                                            CSL02390
      U(KK)=(C2*W(KK)+CR0*W(KK+1)+CR1*W(KK+KX)) - CSUM*U(KK)            CSL02400
      KK = KX                                                           CSL02410
      U(KK)=(C2*W(KK)+CL0*W(KK-1)+CR1*W(KK+KX)) - CSUM*U(KK)            CSL02420
      KK = N - KX + 1                                                   CSL02430
      U(KK)=(C2*W(KK)+CR0*W(KK+1)+CL1*W(KK-KX)) - CSUM*U(KK)            CSL02440
      KK = N                                                            CSL02450
      U(KK)=(C2*W(KK)+CL0*W(KK-1)+CL1*W(KK-KX)) - CSUM*U(KK)            CSL02460
C                                                                       CSL02470
      DO 10 J = 2,KX1                                                   CSL02480
      KK = J                                                            CSL02490
      U(KK)=(C2*W(KK)+CL0*W(KK-1)+CR0*W(KK+1)+CR1*W(KK+KX))-CSUM*U(KK)  CSL02500
      KK = J+N-KX                                                       CSL02510
      U(KK)=(C2*W(KK)+CL0*W(KK-1)+CR0*W(KK+1)+CL1*W(KK-KX))-CSUM*U(KK)  CSL02520
   10 CONTINUE                                                          CSL02530
C                                                                       CSL02540
      DO 30 J = 2,KY1                                                   CSL02550
      KK = (J-1)*KX + 1                                                 CSL02560
      U(KK)=(C2*W(KK)+CR0*W(KK+1)+CL1*W(KK-KX)+CR1*W(KK+KX))-CSUM*U(KK) CSL02570
      KK = J*KX                                                         CSL02580
      U(KK)=(C2*W(KK)+CL0*W(KK-1)+CL1*W(KK-KX)+CR1*W(KK+KX))-CSUM*U(KK) CSL02590
      DO 20 I = 2,KX1                                                   CSL02600
      KK = (J-1)*KX + I                                                 CSL02610
      U(KK)=(C2*W(KK)+CL0*W(KK-1)+CR0*W(KK+1)+CL1*W(KK-KX)              CSL02620
```

```
     1 +CR1*W(KK+KX)) - CSUM*U(KK)                                      CSL02630
   20 CONTINUE                                                          CSL02640
   30 CONTINUE                                                          CSL02650
C                                                                       CSL02660
      RETURN                                                            CSL02670
C                                                                       CSL02680
C-----------------------------------------------------------------------CSL02690
      ENTRY HMATVE(C2,CL0,CL1,KX,KY)                                    CSL02700
C-----------------------------------------------------------------------CSL02710
C                                                                       CSL02720
C-----END OF CSMATV-----------------------------------------------------CSL02730
      RETURN                                                            CSL02740
      END                                                               CSL02750
C                                                                       CSL02760
C     BELOW IS USPEC AND CMATV FOR TEST MATRICES 2.  IN THIS CASE       CSL02770
C     THE EIGENVALUES ARE KNOWN AND WE COMPUTE THEM TO CHECK            CSL02780
C     VALUES OBTAINED FROM THE LANCZOS PROGRAMS.                        CSL02790
C                                                                       CSL02800
C     USES 3 SUBROUTINES BELOW, USPEC CMATV EXEVG                       CSL02810
C                                                                       CSL02820
C-----START OF USPEC (TEST MATRICES 2)----------------------------------CSL02830
C                                                                       CSL02840
C     SUBROUTINE USPEC(N,MATNO)                                         CSL02850
      SUBROUTINE CSPEC(N,MATNO)                                         CSL02860
C                                                                       CSL02870
C-----------------------------------------------------------------------CSL02880
      COMPLEX*16 CPAR,CC0,CC1,CC2                                       CSL02890
      DOUBLE PRECISION   C0,C1,C2,HALF,ONE                              CSL02900
      REAL  EXPLAN(20)                                                  CSL02910
C     COMPLEX*16  DCMPLX                                                CSL02920
C-----------------------------------------------------------------------CSL02930
C     IVEC   = (0,-1,1) MEANS                                           CSL02940
C              (0) ONLY SET ENTRY FOR CMATV                             CSL02950
C              (-1) CALCULATE EXACTEV AND MINGAPS AND STOP.             CSL02960
C              (1) CALCULATE EXACTEV AND MINGAPS AND THEN CONTINUE.     CSL02970
C-----------------------------------------------------------------------CSL02980
      HALF = 0.5D0                                                      CSL02990
      ONE  = 1.0D0                                                      CSL03000
      CPAR = DCMPLX(ONE,ONE)                                            CSL03010
C-----------------------------------------------------------------------CSL03020
C     READ USER-SPECIFIED PARAMETERS FROM INPUT FILE 8 (FREE FORMAT)    CSL03030
C                                                                       CSL03040
      READ(8,10) EXPLAN                                                 CSL03050
      WRITE(6,10) EXPLAN                                                CSL03060
   10 FORMAT(20A4)                                                      CSL03070
C                                                                       CSL03080
      READ(8,10) EXPLAN                                                 CSL03090
      READ(8,*) KX,KY,IVEC,C0                                           CSL03100
      N = KX*KY                                                         CSL03110
      C1 = HALF-C0                                                      CSL03120
      C2 = ONE                                                          CSL03130
      CC0 = CPAR*C0                                                     CSL03140
      CC1 = CPAR*C1                                                     CSL03150
      CC2 = CPAR*C2                                                     CSL03160
C                                                                       CSL03170
```

```
      WRITE(6,20) N,KX,KY,C2,C0,C1,CPAR                                 CSL03180
   20 FORMAT(/5X,'N',4X,'KX',4X,'KY',7X,'DIAGONAL',3X,'X-CODIAGONAL',   CSL03190
     1 3X,'Y-CODIAGONAL'/3I6,3E15.8/7X,' COMPLEX SCALAR MULTIPLIER'/    CSL03200
     13X,2E15.4)                                                        CSL03210
C                                                                       CSL03220
C-----------------------------------------------------------------------CSL03230
      CALL CMATVE(CC0,CC1,CC2,KX,KY)                                    CSL03240
C-----------------------------------------------------------------------CSL03250
C                                                                       CSL03260
      IF (IVEC.EQ.0) GO TO 30                                           CSL03270
C                                                                       CSL03280
C-----------------------------------------------------------------------CSL03290
C     COMPUTE TRUE EIGENVALUES FOR CORRESPONDING REAL POISSON MATRIX    CSL03300
      CALL EXEVG(C0,C1,C2,KX,KY)                                        CSL03310
C-----------------------------------------------------------------------CSL03320
C                                                                       CSL03330
      IF (IVEC.LT.0) STOP                                               CSL03340
C                                                                       CSL03350
   30 CONTINUE                                                          CSL03360
C                                                                       CSL03370
C-----END OF USPEC------------------------------------------------------CSL03380
      RETURN                                                            CSL03390
      END                                                               CSL03400
C                                                                       CSL03410
C-----START OF CMATV (USES TEST MATRICES 2)-----------------------------CSL03420
C     CALCULATE U = A*W - SUM*U                                         CSL03430
C                                                                       CSL03440
C     SUBROUTINE CMATV(W,U,CSUM)                                        CSL03450
      SUBROUTINE CSRMAT(W,U,CSUM)                                       CSL03460
C                                                                       CSL03470
C-----------------------------------------------------------------------CSL03480
      COMPLEX*16 U(1),W(1)                                              CSL03490
      COMPLEX*16 CC0,CC1,CC2,CL0,CL1,CR0,CR1,CSUM                       CSL03500
C-----------------------------------------------------------------------CSL03510
C                                                                       CSL03520
      N = KX*KY                                                         CSL03530
      KX1 = KX-1                                                        CSL03540
      KY1 = KY-1                                                        CSL03550
      CR0 = CC0                                                         CSL03560
      CR1 = CC1                                                         CSL03570
      CL0 = CC0                                                         CSL03580
      CL1 = CC1                                                         CSL03590
C                                                                       CSL03600
      KK = 1                                                            CSL03610
      U(KK)=(CC2*W(KK)+CR0*W(KK+1)+CR1*W(KK+KX)) - CSUM*U(KK)           CSL03620
      KK = KX                                                           CSL03630
      U(KK)=(CC2*W(KK)+CL0*W(KK-1)+CR1*W(KK+KX)) - CSUM*U(KK)           CSL03640
      KK = N - KX + 1                                                   CSL03650
      U(KK)=(CC2*W(KK)+CR0*W(KK+1)+CL1*W(KK-KX)) - CSUM*U(KK)           CSL03660
      KK = N                                                            CSL03670
      U(KK)=(CC2*W(KK)+CL0*W(KK-1)+CL1*W(KK-KX)) - CSUM*U(KK)           CSL03680
C                                                                       CSL03690
      DO 10 J = 2,KX1                                                   CSL03700
      KK = J                                                            CSL03710
      U(KK)=(CC2*W(KK)+CL0*W(KK-1)+CR0*W(KK+1)+CR1*W(KK+KX))-CSUM*U(KK) CSL03720
```

```
      KK = J+N-KX                                                       CSL03730
      U(KK)=(CC2*W(KK)+CL0*W(KK-1)+CR0*W(KK+1)+CL1*W(KK-KX))-CSUM*U(KK) CSL03740
   10 CONTINUE                                                          CSL03750
C                                                                       CSL03760
      DO 30 J = 2,KY1                                                   CSL03770
      KK = (J-1)*KX + 1                                                 CSL03780
      U(KK)=(CC2*W(KK)+CR0*W(KK+1)+CL1*W(KK-KX)+CR1*W(KK+KX))-CSUM*U(KK)CSL03790
      DO 20 I = 2,KX1                                                   CSL03800
      KK = KK + 1                                                       CSL03810
      U(KK)=(CC2*W(KK)+CL0*W(KK-1)+CR0*W(KK+1)+CL1*W(KK-KX)             CSL03820
     1 +CR1*W(KK+KX)) - CSUM*U(KK)                                      CSL03830
   20 CONTINUE                                                          CSL03840
      KK = KK + 1                                                       CSL03850
      U(KK)=(CC2*W(KK)+CL0*W(KK-1)+CL1*W(KK-KX)+CR1*W(KK+KX))-CSUM*U(KK)CSL03860
   30 CONTINUE                                                          CSL03870
C                                                                       CSL03880
      RETURN                                                            CSL03890
C                                                                       CSL03900
C-----------------------------------------------------------------------CSL03910
      ENTRY CMATVE(CC0,CC1,CC2,KX,KY)                                   CSL03920
C-----------------------------------------------------------------------CSL03930
C                                                                       CSL03940
C-----END OF CMATV------------------------------------------------------CSL03950
      RETURN                                                            CSL03960
      END                                                               CSL03970
C                                                                       CSL03980
C-----START OF EXEVG (COMPUTES EXACT EIGENVALUES FOR TEST MATRICES 2)---CSL03990
C                                                                       CSL04000
      SUBROUTINE EXEVG(C0,C1,C2,KX,KY)                                  CSL04010
C                                                                       CSL04020
C-----------------------------------------------------------------------CSL04030
      DOUBLE PRECISION U(2000),MACHEP                                   CSL04040
      DOUBLE PRECISION EPSM,C0,C1,C2,T0,T1,PIK,PIL,ONE,TWO,ATOLN,EE     CSL04050
      REAL G(2000)                                                      CSL04060
      INTEGER MP(2000)                                                  CSL04070
      REAL ABS                                                          CSL04080
      DOUBLE PRECISION DABS, DARCOS, DFLOAT, DCOS, DMAX1                CSL04090
C-----------------------------------------------------------------------CSL04100
      DATA MACHEP/Z3410000000000000/                                    CSL04110
      EPSM = 2.0D0*MACHEP                                               CSL04120
C-----------------------------------------------------------------------CSL04130
      N = KX*KY                                                         CSL04140
      ONE  = 1.0D0                                                      CSL04150
      TWO  = 2.0D0                                                      CSL04160
      T0 = DARCOS(-ONE)                                                 CSL04170
      T1 = DFLOAT(KX+1)                                                 CSL04180
      PIK = T0/T1                                                       CSL04190
      T1 = DFLOAT(KY+1)                                                 CSL04200
      PIL = T0/T1                                                       CSL04210
C     GENERATE EXACT EIGENVALUES                                        CSL04220
      KP = 0                                                            CSL04230
      DO 20 J = 1,KY                                                    CSL04240
      T1 = PIL*DFLOAT(J)                                                CSL04250
      T0 = C2 - TWO*C1*DCOS(T1)                                         CSL04260
      DO 10 I = 1,KX                                                    CSL04270
```

```
      KP = KP+1                                                         CSL04280
      T1 = PIK*DFLOAT(I)                                                CSL04290
   10 U(KP) = T0 - TWO*C0*DCOS(T1)                                      CSL04300
   20 CONTINUE                                                          CSL04310
C                                                                       CSL04320
C     ORDER U VECTOR BY INCREASING ALGEBRAIC SIZE                       CSL04330
      DO 40 K = 2,N                                                     CSL04340
      KM1 = K-1                                                         CSL04350
      DO 30 L = 1,KM1                                                   CSL04360
      JJ = K-L                                                          CSL04370
      IF (U(JJ+1).GE.U(JJ)) GO TO 40                                    CSL04380
      T0 = U(JJ)                                                        CSL04390
      U(JJ) = U(JJ+1)                                                   CSL04400
   30 U(JJ+1) = T0                                                      CSL04410
   40 CONTINUE                                                          CSL04420
      ATOLN = DMAX1(DABS(U(1)),DABS(U(N)))*EPSM                         CSL04430
C                                                                       CSL04440
      WRITE(9,50)                                                       CSL04450
   50 FORMAT(' TRUE EIGENVALUES FOR POISSON'/)                          CSL04460
C                                                                       CSL04470
      WRITE(9,60)N,KX,KY,C2,C0,C1,ATOLN                                 CSL04480
      WRITE(6,60) N,KX,KY,C2,C0,C1,ATOLN                                CSL04490
   60 FORMAT(1X,'A-SIZE',2X,'X-DIM',2X,'Y-DIM'/3I7/                     CSL04500
     1 5X,'A-DIAGONAL',3X,'X-CODIAGONAL',3X,'Y-CODIAGONAL',10X,'ATOLN'/ CSL04510
     2 4E15.8)                                                          CSL04520
C                                                                       CSL04530
C     DETERMINE MULTIPLICITIES FOR EXACT EIGENVALUES                    CSL04540
      I = 1                                                             CSL04550
      IDEX = 1                                                          CSL04560
      J = 1                                                             CSL04570
      NEXACT = 0                                                        CSL04580
   70 J = J+1                                                           CSL04590
      IF (J.GT.N) GO TO 80                                              CSL04600
      EE = DABS(U(J)-U(I))                                              CSL04610
      IF (EE.GT.ATOLN) GO TO 80                                         CSL04620
      IDEX = IDEX+1                                                     CSL04630
      GO TO 70                                                          CSL04640
   80 NEXACT = NEXACT+1                                                 CSL04650
      U(NEXACT) = U(I)                                                  CSL04660
      MP(NEXACT) = IDEX                                                 CSL04670
C     MP(K) = MULTIPLICITY OF KTH EIGENVALUE CLUSTER FOR A              CSL04680
      IDEX = 1                                                          CSL04690
      I = J                                                             CSL04700
      IF (I.GT.N) GO TO 90                                              CSL04710
      GO TO 70                                                          CSL04720
   90 CONTINUE                                                          CSL04730
C                                                                       CSL04740
C     MULTIPLICITIES HAVE BEEN DETERMINED                               CSL04750
C     NEXACT = NUMBER OF DISTINCT A-EIGENVALUES                         CSL04760
C                                                                       CSL04770
C                                                                       CSL04780
      WRITE(9,100)NEXACT                                                CSL04790
      WRITE(6,100)NEXACT                                                CSL04800
  100 FORMAT(I6,' = NUMBER OF TRUE A-EIGENVALUES WHICH ARE DISTINCT'/)  CSL04810
C                                                                       CSL04820
```

```
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,N
  150 MP(K) = K
C
      DO 170 K = 2,N
      KM1 = K-1
C
      DO 160 L = 1,KM1
      JJ = K - L
      IF (ABS(G(JJ+1)).GE.ABS(G(JJ))) GO TO 170
      EE = U(JJ)
      U(JJ) = U(JJ+1)
      U(JJ+1) = EE
      GG = G(JJ)
      G(JJ) = G(JJ+1)
      G(JJ+1) = GG
      IEE = MP(JJ)
      MP(JJ) = MP(JJ+1)
  160 MP(JJ+1) = IEE
C
  170 CONTINUE
C
      WRITE(9,180)
```

```
  180 FORMAT(5X,'K',6X,'A-MINGAP',5X,'TRUE A-EIGENVALUE(I)',2X,'A-EVNO')CSL05380
C                                                                       CSL05390
      WRITE(9,190)(J,G(J),U(J),MP(J), J=1,NEXACT)                       CSL05400
  190 FORMAT(I6,E14.3,E25.16,I8)                                        CSL05410
C                                                                       CSL05420
      WRITE(9,200)                                                      CSL05430
  200 FORMAT(' NEXACT DISTINCT A-EIGENVALUES. GAPS IN ASCENDING ORDER'/ CSL05440
     2 ' A-MINGAP(I) = TRUE MINIMUM GAP IN A FOR EACH EIGENVALUE.'/     CSL05450
     3 ' A-MINGAP(I).LT.0 INDICATES THE LEFT-HAND GAP WAS MINIMAL.'/    CSL05460
     3 ' A-MATRIX IS BLOCK TRIDIAGONAL AND EACH DIAGONAL BLOCK IS OF ORDCSL05470
     3ER NX.'/                                                          CSL05480
     4 ' NX = NUMBER OF POINTS ON EACH X-LINE. THERE ARE NY DIAGONAL BLOCSL05490
     4CKS.'/                                                            CSL05500
     5 ' NY = NUMBER OF POINTS ON EACH Y-LINE.'/                        CSL05510
     5 ' A-DIAGONAL   = A(K,K)'/                                        CSL05520
     6 ' X-CODIAGONAL = A(I,I+1)'/                                      CSL05530
     7 ' Y-CODIAGONAL = A(I,I+NX)'/                                     CSL05540
     8 ' ----- END OF FILE 9 EXACTEV----------------------------'//)    CSL05550
C                                                                       CSL05560
C-----END OF EXEVG------------------------------------------------------CSL05570
C                                                                       CSL05580
      RETURN                                                            CSL05590
      END                                                               CSL05600
```

7.6 CSLESUB: Other Subroutines used by the Codes in Chapter 7

```
C-----CSLESUB-(NONDEFECTIVE COMPLEX SYMMETRIC MATRICES)-----------------CSL00010
C  Authors:  Jane Cullum and Ralph A. Willoughby (Deceased)             CSL00020
C            Los Alamos National Laboratory                             CSL00030
C            Los Alamos, New Mexico 87544                               CSL00040
C                                                                       CSL00050
C            E-mail:  cullumj@lanl.gov                                  CSL00060
C                                                                       CSL00070
C  These codes are copyrighted by the authors.  These codes             CSL00080
C  and modifications of them or portions of them are NOT to be          CSL00090
C  incorporated into any commercial codes or used for any other         CSL00100
C  commercial purposes such as consulting for other companies,          CSL00110
C  without legal agreements with the authors of these Codes.            CSL00120
C  If these Codes or portions of them                                   CSL00130
C  are used in other scientific or engineering research works           CSL00140
C  the names of the authors of these codes and appropriate              CSL00150
C  references to their written work are to be incorporated in the      CSL00160
C  derivative works.                                                    CSL00170
C                                                                       CSL00180
C  This header is not to be removed from these codes.                   CSL00190
C                                                                       CSL00200
C         REFERENCE:  Cullum and Willoughby, Chapter 6,                 CSL00201
C         Lanczos Algorithms for Large Symmetric Eigenvalue ComputationsCSL00202
C         VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in   CSL00203
C         Applied Mathematics, 2002. SIAM Publications,                 CSL00204
C         Philadelphia, PA. USA                                         CSL00205
C                                                                       CSL00206
C                                                                       CSL00207
C                                                                       CSL00210
C     NONPORTABLE CONSTRUCTIONS:                                        CSL00220
C     THESE SUBROUTINES ARE NOT PORTABLE DUE TO THE USE OF THE          CSL00230
C     COMPLEX*16 VARIABLES AND THE CORRESPONDING COMPLEX FUNCTIONS,     CSL00240
C     CDABS, DCMPLX, DREAL, DIMAG.  MOREOVER, IN SUBROUTINE             CSL00250
C     COMPEV THE NONPORTABLE FORMATS (4Z20) AND (20A4) ARE USED,        CSL00260
C     AND IN SUBROUTINE CMTQL1 THE MACHINE EPSILON IS INTRODUCED        CSL00270
C     VIA A NONPORTABLE DATA DEFINITION.                                CSL00280
C                                                                       CSL00290
C     CONTAINS SUBROUTINES USED BY THE COMPLEX SYMMETRIC VERSION OF     CSL00300
C     THE LANCZOS EIGENVALUE/EIGENVECTOR CODES.                         CSL00310
C                                                                       CSL00320
C     SUBROUTINES      COMPEV, CMTQL1, INVERR, TNORM, LUMP, ISOEV AND   CSL00330
C                      COMGAP ARE USED WITH THE LANCZOS EIGENVALUE      CSL00340
C                      PROGRAM CSLEVAL.  INVERM IS USED                 CSL00350
C                      IN THE EIGENVECTOR PROGRAM CSLEVEC.  THE INNER   CSL00360
C                      PRODUCT SUBROUTINES CINPRD AND INPRDC ARE USED   CSL00370
C                      BY BOTH PROGRAMS.                                CSL00380
C                                                                       CSL00390
C-----INVERSE ITERATION ON COMPLEX SYMMETRIC T(1,MEV)-------------------CSL00400
C                                                                       CSL00410
      SUBROUTINE INVERR(ALPHA,BETA,V1,V2,VS,EPS,GR,GC,G,GG,MP,INTERC,   CSL00420
     1MEV,MMB,NDIS,NISO,N,IKL,IT,IWRITE)                                CSL00430
```

```
C
C-----------------------------------------------------------------------
      COMPLEX*16  ALPHA(1),BETA(1),V1(1),V2(1),VS(1)
      COMPLEX*16  U,Z,X1,RATIO,BETAM,TEMP,ZEROC
      DOUBLE PRECISION EST,ESTR,ESTC,SUM,XU,NORM,TSUM,GSUM
      DOUBLE PRECISION  EPS,EPS3,EPS4,ZERO,ONE,GR(1),GC(1),GAP
      REAL G(1),GG(1)
      INTEGER  MP(1), INTERC(1)
      REAL  ABS
      DOUBLE PRECISION  DABS, DMIN1, DSQRT, DFLOAT, CDABS, DIMAG, DREAL
C     COMPLEX*16 DCMPLX
C-----------------------------------------------------------------------
C
C     COMPUTES ERROR ESTIMATES FOR COMPUTED ISOLATED GOOD T-EIGENVALUES
C     IN VS AND WRITES THESE EIGENVALUES AND ESTIMATES TO FILE 4.
C     BY DEFINITION A GOOD T-EIGENVALUE IS ISOLATED IF ITS CLOSEST
C     NEIGHBOR IS ALSO GOOD, OR IF ITS CLOSEST NEIGHBOR IS
C     SPURIOUS BUT THAT NEIGHBOR IS FAR ENOUGH AWAY.  SO
C     IN PARTICULAR, WE WILL COMPUTE ESTIMATES FOR ANY GOOD
C     T-EIGENVALUE THAT IS IN A CLUSTER OF GOOD T-EIGENVALUES.
C
C     USES INVERSE ITERATION ON T(1,MEV) SOLVING THE EQUATION
C     (T - X1*I)V2 = RIGHT-HAND SIDE (RANDOMLY-GENERATED)
C     FOR EACH SUCH GOOD T-EIGENVALUE X1.
C
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     ON ENTRY AND EXIT
C     MEV = ORDER OF T :  N = ORDER OF ORIGINAL MATRIX A
C     ALPHA, BETA CONTAIN THE NONZERO ENTRIES OF THE T-MATRIX
C     VS = COMPUTED DISTINCT EIGENVALUES OF T(1,MEV)
C     MP = T-MULTIPLICITY OF EACH T-EIGENVALUE IN VS. MP(I) = -1 MEANS
C          VS(I) IS A GOOD T-EIGENVALUE BUT THAT IT IS SITTING CLOSE TO
C          A SPURIOUS T-EIGENVALUE.  MP(I) = 0 MEANS VS(I) IS SPURIOUS.
C          ESTIMATES ARE COMPUTED ONLY FOR THOSE T-EIGENVALUES
C          WITH MP(I) = 1. FLAGGING WAS DONE IN SUBROUTINE ISOEV
C          PRIOR TO ENTERING INVERR.
C     NISO = NUMBER OF ISOLATED GOOD T-EIGENVALUES CONTAINED IN VS
C     NDIS =  NUMBER OF DISTINCT T-EIGENVALUES IN VS
C     IKL = SEED FOR RANDOM NUMBER GENERATOR
C     EPS = 2. * MACHINE EPSILON
C
C     IN PROGRAM:
C     ITER = MAXIMUM NUMBER OF INVERSE ITERATION STEPS ALLOWED FOR EACH
C            X1.  ITER = IT ON ENTRY.
C     GR,GC = ARRAYS OF DIMENSION AT LEAST MEV + NISO.  USED TO STORE
C         RANDOMLY-GENERATED RIGHT-HAND SIDE.  THIS IS NOT
C         REGENERATED FOR EACH X1. G IS ALSO USED TO STORE ERROR
C         ESTIMATES AS THEY ARE COMPUTED FOR LATER PRINTOUT.
C     V1,V2 = WORK SPACES USED IN THE FACTORIZATION OF T(1,MEV).
C     AT THE END OF THE INVERSE ITERATION COMPUTATION FOR X1, V2
C     CONTAINS THE UNIT EIGENVECTOR OF T(1,MEV) CORRESPONDING TO X1.
C     V1 AND V2 MUST BE OF DIMENSION AT LEAST MEV.
C
```

```
C     ON EXIT
C     GG(J) = MINIMUM GAP IN T(1,MEV) FOR EACH VS(J), J=1,NDIS
C     G(I) = |BETAM|*|V2(MEV)| = ERROR ESTIMATE FOR ISOLATED GOOD
C              T-EIGENVALUES, WHERE I = 1,NISO  AND  BETAM = BETA(MEV+1)
C              T(1,MEV) CORRESPONDING TO ITH ISOLATED GOOD T-EIGENVALUE.
C
C     IF FOR SOME X1 IT.GT.ITER THEN THE ERROR ESTIMATE IN G IS MARKED
C     WITH A - SIGN.
C
C     V2 = ISOLATED GOOD T-EIGENVALUES
C     V1 = MINIMAL T-GAPS FOR THE T-EIGENVALUES IN V2.
C     THESE ARE CONSTRUCTED FOR WRITE-OUT PURPOSES ONLY AND NOT
C     NEEDED ELSEWHERE IN THE PROGRAM.
C-----------------------------------------------------------------------
C
C     LABEL OUTPUT FILE  4
      IF (MMB.EQ.1) WRITE(4,10)
   10 FORMAT(' INVERSE ITERATION ERROR ESTIMATES'/)
C
C     FILE 6 (TERMINAL) OUTPUT OF ERROR ESTIMATES
      IF (IWRITE.NE.0.AND.NISO.NE.0) WRITE(6,20)
   20 FORMAT(/' INVERSE ITERATION ERROR ESTIMATES'/'  JISO',' JDIST',8X
     1,'GOOD T-EIGENVALUE',4X,'BETAM*UM',5X,'TMINGAP')
C
C     INITIALIZATION AND PARAMETER SPECIFICATION
      ZERO = 0.0D0
      ONE = 1.0D0
      ZEROC = DCMPLX(ZERO,ZERO)
      NG = 0
      NISO = 0
      ITER = IT
      MP1 = MEV+1
      MM1 = MEV-1
      BETAM = BETA(MP1)
      BETA(MP1) = ZEROC
C
C     CALCULATE SCALE AND TOLERANCES
      TSUM = CDABS(ALPHA(1))
      DO 30 I = 2,MEV
   30 TSUM = TSUM + CDABS(ALPHA(I)) + CDABS(BETA(I))
C
      EPS3 = EPS*TSUM
      EPS4 = DFLOAT(MEV)*EPS3
C
C     GENERATE SCALED RANDOM RIGHT-HAND SIDE
      ILL = IKL
C
C-----------------------------------------------------------------------
      CALL GENRAN(ILL,G,MEV)
C-----------------------------------------------------------------------
C
      DO 40 I = 1,MEV
   40 GR(I) = G(I)
C
C-----------------------------------------------------------------------
```

```
      CALL GENRAN(ILL,G,MEV)
C-----------------------------------------------------------------------
C
      DO 50 I = 1,MEV
   50 GC(I) = G(I)
C
      GSUM = ZERO
      DO 60 I = 1,MEV
   60 GSUM = GSUM + DABS(GR(I)) + DABS(GC(I))
      GSUM = EPS4/GSUM
C
      DO 70 I = 1,MEV
      GR(I) = GSUM*GR(I)
   70 GC(I) = GSUM*GC(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 200 JEV = 1,NDIS
      IF (MP(JEV).EQ.0) GO TO 200
      NG = NG + 1
      IF (MP(JEV).NE.1) GO TO 200
      IT = 1
      NISO = NISO + 1
      X1 = VS(JEV)
C
C     INITIALIZE RIGHT HAND SIDE FOR INVERSE ITERATION
C     AND THE FLAG ON WHICH ROWS ARE INTERCHANGED
      DO 80 I = 1,MEV
      INTERC(I) = 0
   80 V2(I) = DCMPLX(GR(I),GC(I))
C
C     TRIANGULAR FACTORIZATION WITH NEAREST NEIGHBOR PIVOT
C     STRATEGY. INTERCHANGES ARE LABELLED BY SETTING INTERC = 1.
C
   90 CONTINUE
      U = ALPHA(1)-X1
      Z = BETA(2)
C
      DO 110 I = 2,MEV
      IF (CDABS(BETA(I)).GT.CDABS(U)) GO TO 100
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 110
  100 CONTINUE
C     INTERCHANGE CASE
      RATIO = U/BETA(I)
      INTERC(I) = 1
      V1(I-1) = ALPHA(I)-X1
      U = Z-RATIO*V1(I-1)
```

```
      Z = -RATIO*BETA(I+1)                                              CSL02090
      TEMP = V2(I-1)                                                    CSL02100
      V2(I-1) = V2(I)                                                   CSL02110
      V2(I) = TEMP-RATIO*V2(I)                                          CSL02120
  110 CONTINUE                                                          CSL02130
      IF (CDABS(U).EQ.ZERO) U = DCMPLX(EPS3,EPS3)                       CSL02140
C                                                                       CSL02150
C     SMALLNESS TEST AND DEFAULT VALUE FOR LAST COMPONENT               CSL02160
C     PIVOT(I-1) = BETA(I) FOR INTERCHANGE CASE                         CSL02170
C     (I-1,I+1) ELEMENT IN RIGHT FACTOR = BETA(I+1)                     CSL02180
C     END OF FACTORIZATION AND FORWARD SUBSTITUTION                     CSL02190
C                                                                       CSL02200
C     BACK SUBSTITUTION                                                 CSL02210
      V2(MEV) = V2(MEV)/U                                               CSL02220
      DO 130 II = 1,MM1                                                 CSL02230
      I = MEV-II                                                        CSL02240
      IF (INTERC(I+1).EQ.1) GO TO 120                                   CSL02250
C     NO INTERCHANGE                                                    CSL02260
      V2(I) = V2(I)-V1(I)*V2(I+1)                                       CSL02270
      GO TO 130                                                         CSL02280
C     INTERCHANGE CASE                                                  CSL02290
  120 CONTINUE                                                          CSL02300
      V2(I) = (V2(I)-V1(I)*V2(I+1)-BETA(I+2)*V2(I+2))/BETA(I+1)         CSL02310
  130 CONTINUE                                                          CSL02320
C                                                                       CSL02330
C     TESTS FOR CONVERGENCE OF INVERSE ITERATION                        CSL02340
C     IF SUM |V2| COMPS. LE. 1 AND IT. LE. ITER DO ANOTHER INVIT STEP   CSL02350
C                                                                       CSL02360
      NORM = CDABS(V2(MEV))                                             CSL02370
      DO 140 II = 1,MM1                                                 CSL02380
      I = MEV-II                                                        CSL02390
  140 NORM = NORM + CDABS(V2(I))                                        CSL02400
C                                                                       CSL02410
      IF (NORM.GE.ONE) GO TO 160                                        CSL02420
      IT = IT+1                                                         CSL02430
      IF (IT.GT.ITER) GO TO 160                                         CSL02440
      XU = EPS4/NORM                                                    CSL02450
C                                                                       CSL02460
      DO 150 I = 1,MEV                                                  CSL02470
  150 V2(I) = V2(I)*XU                                                  CSL02480
C                                                                       CSL02490
      GO TO 90                                                          CSL02500
C     ANOTHER INVERSE ITERATION STEP                                    CSL02510
C                                                                       CSL02520
C     INVERSE ITERATION FINISHED                                        CSL02530
C     NORMALIZE COMPUTED T-EIGENVECTOR : V2 = V2/||V2||                 CSL02540
  160 CONTINUE                                                          CSL02550
C                                                                       CSL02560
C-----------------------------------------------------------------------CSL02570
      CALL CINPRD(V2,V2,SUM,MEV)                                        CSL02580
C-----------------------------------------------------------------------CSL02590
C                                                                       CSL02600
      SUM = ONE/DSQRT(SUM)                                              CSL02610
C                                                                       CSL02620
      DO 170 II = 1,MEV                                                 CSL02630
```

```
  170 V2(II) = SUM*V2(II)
C
C     SAVE ERROR ESTIMATE FOR LATER OUTPUT
      EST = CDABS(BETAM)*CDABS(V2(MEV))
      ESTR = DABS(DREAL(V2(MEV)))
      ESTC = DABS(DIMAG(V2(MEV)))
      GSUM = CDABS(BETAM)
      IF (IT.GT.ITER) EST = -EST
      G(NISO) = EST
      IF (IWRITE.EQ.0) GO TO 200
C
C     FILE 6 (TERMINAL) OUTPUT OF ERROR ESTIMATES.
      GAP = GG(JEV)
      WRITE(6,180) NISO,JEV,X1,EST,GAP
  180 FORMAT(2I6,2E20.12,2E12.3)
      WRITE(6,190) JEV, X1, EST,ESTR,ESTC
  190 FORMAT(I6,2E20.12,3E11.3)
C
  200 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
      ISO = 0
      DO 210 J = 1,NDIS
      IF (MP(J).NE.1) GO TO 210
      ISO = ISO+1
      GR(ISO) = GG(J)
      V2(ISO) = VS(J)
  210 CONTINUE
      IF(NISO.EQ.0) GO TO 270
C
C     ERROR ESTIMATES ARE WRITTEN TO FILE 4
      WRITE(4,220)MEV,NDIS,NG,NISO,N,IKL,ITER,GSUM
  220 FORMAT(1X,'TSIZE',2X,'NDIS',1X,'NGOOD',2X,'NISO',1X,'ASIZE'/5I6/
     1 4X,'RHSEED',2X,'MXINIT',5X,'BETAM'/I10,I8,E10.3)
C
      WRITE(4,230)
  230 FORMAT(2X,'GOODEVNO',11X,'R(GOODEV)',11X,'I(GOODEV)',
     1 6X,'BETAM*UM',7X,'TMINGAP')
C
      ISPUR = 0
      I = 0
      DO 260 J = 1,NDIS
      IF(MP(J).NE.0) GO TO 240
      ISPUR = ISPUR + 1
      GO TO 260
  240 IF(MP(J).NE.1) GO TO 260
      I = I + 1
      IGOOD = J - ISPUR
      WRITE(4,250) IGOOD,V2(I),G(I),GR(I)
  250 FORMAT(I10,2E20.12,2E14.3)
```

```
  260 CONTINUE                                                          CSL03190
      GO TO 290                                                         CSL03200
C                                                                       CSL03210
  270 WRITE(4,280)                                                      CSL03220
  280 FORMAT(/' THERE ARE NO ISOLATED T-EIGENVALUES SO NO ERROR ESTIMATECSL03230
     1S WERE COMPUTED')                                                 CSL03240
C     RESTORE BETA(MEV+1) = BETAM                                       CSL03250
  290 BETA(MP1) = BETAM                                                 CSL03260
C-----END OF INVERR-----------------------------------------------------CSL03270
      RETURN                                                            CSL03280
      END                                                               CSL03290
C-----START OF TNORM----------------------------------------------------CSL03300
C                                                                       CSL03310
      SUBROUTINE TNORM(ALPHA,BETA,BMIN,TMAX,MEV,IB)                     CSL03320
C                                                                       CSL03330
C-----------------------------------------------------------------------CSL03340
      COMPLEX*16  ALPHA(1),BETA(1)                                      CSL03350
      DOUBLE PRECISION  TMAX,BMIN,BMAX,BSIZE,BTOL,ABATA,AALFA           CSL03360
      DOUBLE PRECISION  DMAX1, CDABS                                    CSL03370
C     COMPLEX*16 DCMPLX                                                 CSL03380
C-----------------------------------------------------------------------CSL03390
C     IN REAL SYMMETRIC AND HERMITIAN VERSIONS TMAX IS USED             CSL03400
C     TO DETERMINE THE TOLERANCES USED IN THE T-MULTIPLICITY AND IN     CSL03410
C     THE SPURIOUS TESTS.  FOR THE COMPLEX SYMMETRIC CASE WE            CSL03420
C     HAVE TO COMPUTE ALL OF THE T-EIGENVALUES SO WE USE THEM INSTEAD   CSL03430
C     OF TMAX TO DETERMINE THESE TOLERANCES.  WE USE TMAX TO            CSL03440
C     CHECK THE RELATIVE SIZES OF THE BETA(K), K=1,...,MEV AS A         CSL03450
C     TEST ON THE LOCAL ORTHOGONALITY OF THE LANCZOS VECTORS.           CSL03460
C                                                                       CSL03470
C             TMAX = MAX (|ALPHA(I)|, |BETA(I)|,  I=1,MEV)              CSL03480
C             BMIN = MIN (|BETA(I)|, I=2,MEV)                           CSL03490
C             BSIZE = BMIN/TMAX                                         CSL03500
C             |IB| = INDEX OF MINIMAL(BETA)                             CSL03510
C             IB < 0 IF BMIN/TMAX < BTOL                                CSL03520
C-----------------------------------------------------------------------CSL03530
C     SPECIFY PARAMETERS                                                CSL03540
      IB = 2                                                            CSL03550
      BTOL = BMIN                                                       CSL03560
      BMIN = CDABS(BETA(2))                                             CSL03570
      BMAX = BMIN                                                       CSL03580
      TMAX = CDABS(ALPHA(1))                                            CSL03590
C                                                                       CSL03600
      DO 20 I = 2,MEV                                                   CSL03610
      ABATA = CDABS(BETA(I))                                            CSL03620
      IF (ABATA.GE.BMIN) GO TO 10                                       CSL03630
      IB = I                                                            CSL03640
      BMIN = ABATA                                                      CSL03650
   10 AALFA = CDABS(ALPHA(I))                                           CSL03660
      TMAX = DMAX1(TMAX,AALFA)                                          CSL03670
      BMAX = DMAX1(ABATA,BMAX)                                          CSL03680
   20 CONTINUE                                                          CSL03690
      TMAX = DMAX1(BMAX,TMAX)                                           CSL03700
C                                                                       CSL03710
C     TEST OF LOCAL ORTHOGONALITY USING SCALED BETAS                    CSL03720
      BSIZE = BMIN/TMAX                                                 CSL03730
```

```
      IF (BSIZE.GE.BTOL) GO TO 40
C
C     DEFAULT.  BSIZE IS SMALLER THAN TOLERANCE BTOL SPECIFIED IN MAIN
C     PROGRAM.  PROGRAM TERMINATES FOR USER TO DECIDE WHAT TO DO
C     BECAUSE LOCAL ORTHOGONALITY OF THE LANCZOS VECTORS COULD BE
C     LOST.
C
      IB = -IB
      WRITE(6,30) MEV
   30 FORMAT(/' BETA TEST INDICATES POSSIBLE LOSS OF LOCAL ORTHOGONALITY
     1 OVER 1ST',I6,' LANCZOS VECTORS'/)
C
   40 CONTINUE
C
      WRITE(6,50) IB
   50 FORMAT(/' MINIMUM BETA RATIO OCCURS AT',I6,' TH BETA'/)
C
      WRITE(6,60) MEV,BMIN,TMAX,BSIZE
   60 FORMAT(/1X,'TSIZE',6X,'MIN BETA',5X,'TKMAX',6X,'MIN RATIO'/
     1 I6,E14.3,E10.3,E15.3/)
C
C-----END OF TNORM------------------------------------------------------
      RETURN
      END
C
C-----START OF LUMP-----------------------------------------------------
C
      SUBROUTINE LUMP(VC,V1,VA,RELTOL,SPUTOL,SCALE2,LINDEX,TFLAG,LOOP)
C
C-----------------------------------------------------------------------
      COMPLEX*16 VC(1),V1(1),ZEROC,SUMC
      DOUBLE PRECISION  VA(1),RELTOL,SPUTOL,SCALE2
      DOUBLE PRECISION  THOLD,TH1,TH2,DGAP,ZERO,ONE
      INTEGER  LINDEX(1),TFLAG(1)
      DOUBLE PRECISION  DFLOAT, DMAX1, CDABS
C     COMPLEX*16 DCMPLX
C-----------------------------------------------------------------------
C     VC(J) = JTH DISTINCT T-EIGENVALUE,  VA(J) = |VC(J)|, IN ORDER
C     OF INCREASING MAGNITUDE.
C     LINDEX(J) = T-MULTIPLICITY OF JTH DISTINCT T-EIGENVALUE
C     LOOP = NUMBER OF DISTINCT T-EIGENVALUES
C     LUMP 'COMBINES' COMPUTED 'GOOD' T-EIGENVALUES THAT ARE 'TOO CLOSE'
C     VALUE OF RELTOL IS 1.D-8.
C
C     IF IN A SET OF T-EIGENVALUES TO BE COMBINED THERE IS AN EIGENVALUE
C     WITH LINDEX=1, THEN THE VALUE OF THE COMBINED T-EIGENVALUES IS SET
C     EQUAL TO THE VALUE OF THAT EIGENVALUE.  NOTE THAT IF A SPURIOUS
C     T-EIGENVALUE IS TO BE 'COMBINED' WITH A GOOD EIGENVALUE, THEN THIS
C     IS DONE ONLY BY INCREASING THE INDEX, LINDEX, FOR THAT EIGENVALUE
C     NUMERICAL VALUES OF SPURIOUS T-EIGENVALUES ARE NEVER COMBINED WITH
C     THOSE OF GOOD T-EIGENVALUES.
C-----------------------------------------------------------------------
      ZERO = 0.0D0
      ONE = 1.D0
      ZEROC = DCMPLX(ZERO,ZERO)
```

```
      TH2 = SCALE2*SPUTOL                                               CSL04290
      DO 10 K = 1,LOOP                                                  CSL04300
   10 TFLAG(K) = 0                                                      CSL04310
      NLOOP = 0                                                         CSL04320
      J = 0                                                             CSL04330
   20 J = J+1                                                           CSL04340
      IF (J.GT.LOOP) GO TO 130                                          CSL04350
      IF (TFLAG(J).EQ.1) GO TO 20                                       CSL04360
      NLOOP = NLOOP + 1                                                 CSL04370
      TFLAG(J) = 1                                                      CSL04380
      V1(1) = VC(J)                                                     CSL04390
      ICOUNT = 1                                                        CSL04400
      JN = LINDEX(J)                                                    CSL04410
      TH1 = RELTOL*VA(J)                                                CSL04420
      THOLD = DMAX1(TH1,TH2)                                            CSL04430
C     THOLD = RELTOL*DMAX1(ONE,VA(J))                                   CSL04440
      IF (JN.EQ.0) GO TO 30                                             CSL04450
      INDSUM = JN                                                       CSL04460
      ISPUR = 0                                                         CSL04470
      SUMC = DFLOAT(JN)*VC(J)                                           CSL04480
      GO TO 40                                                          CSL04490
   30 INDSUM = 1                                                        CSL04500
      ISPUR = 1                                                         CSL04510
      SUMC = ZEROC                                                      CSL04520
   40 IF (J.EQ.LOOP) GO TO 70                                           CSL04530
      I = J                                                             CSL04540
   50 I = I + 1                                                         CSL04550
      IF (I.GT.LOOP) GO TO 70                                           CSL04560
      IF (TFLAG(I).EQ.1) GO TO 50                                       CSL04570
      DGAP = VA(I) - VA(J)                                              CSL04580
      IF (DGAP.GE.THOLD) GO TO 70                                       CSL04590
      DGAP = CDABS(VC(I)-VC(J))                                         CSL04600
      IF (DGAP.GE.THOLD) GO TO 50                                       CSL04610
C     LUMP VC(I) WITH VC(J)                                             CSL04620
      ICOUNT = ICOUNT + 1                                               CSL04630
      TFLAG(I) = 1                                                      CSL04640
      V1(ICOUNT) = VC(I)                                                CSL04650
      IN = LINDEX(I)                                                    CSL04660
      IF (IN.NE.0) GO TO 60                                             CSL04670
      ISPUR = ISPUR + 1                                                 CSL04680
      INDSUM = INDSUM + 1                                               CSL04690
      GO TO 50                                                          CSL04700
   60 INDSUM = INDSUM + IN                                              CSL04710
      SUMC = SUMC + DFLOAT(IN)*VC(I)                                    CSL04720
      GO TO 50                                                          CSL04730
C     COMPUTE THE 'COMBINED' T-EIGENVALUE AND THE RESULTING             CSL04740
C     T-MULTIPLICITY                                                    CSL04750
   70 CONTINUE                                                          CSL04760
C                                                                       CSL04770
C     IF (ICOUNT.GT.1) WRITE(6,80) (K,V1(K), K = 1,ICOUNT)              CSL04780
   80 FORMAT(/' T-EIGENVALUES ARE LUMPED '/                             CSL04790
     1 5X,'J',12X,'REAL(EV)',12X,'IMAG(EV)'/(I6,2E20.12))               CSL04800
C                                                                       CSL04810
      IF (ICOUNT.EQ.1) INDSUM = JN                                      CSL04820
      IDIF = INDSUM - ISPUR                                             CSL04830
```

```
      IF (IDIF.EQ.0.AND.ICOUNT.GT.1) GO TO 90                           CSL04840
      IF (ICOUNT.EQ.1) GO TO 90                                         CSL04850
C     ICOUNT.GT.1 AND IDIF.GT.0                                         CSL04860
      SUMC = SUMC/DFLOAT(IDIF)                                          CSL04870
      VC(NLOOP) = SUMC                                                  CSL04880
      VA(NLOOP) = CDABS(SUMC)                                           CSL04890
      GO TO 100                                                         CSL04900
   90 VC(NLOOP) = VC(J)                                                 CSL04910
      VA(NLOOP) = VA(J)                                                 CSL04920
  100 LINDEX(NLOOP) = INDSUM                                            CSL04930
      GO TO 20                                                          CSL04940
C     INDEX J IS FINISHED                                               CSL04950
C                                                                       CSL04960
C     ON RETURN VC CONTAINS THE DISTINCT T-EIGENVALUES  VA = |VC|       CSL04970
C     LINDEX CONTAINS THE CORRESPONDING T-MULTIPLICITIES                CSL04980
C                                                                       CSL04990
  130 CONTINUE                                                          CSL05000
      LOOP = NLOOP                                                      CSL05010
      RETURN                                                            CSL05020
C-----END OF LUMP-------------------------------------------------------CSL05030
      END                                                               CSL05040
C                                                                       CSL05050
C-----START OF ISOEV----------------------------------------------------CSL05060
C                                                                       CSL05070
      SUBROUTINE ISOEV(VS,GR,GG,GAPTOL,SPUTOL,SCALE1,MP,NDIS,NG,NISO)   CSL05080
C                                                                       CSL05090
C-----------------------------------------------------------------------CSL05100
      COMPLEX*16 VS(1),T0                                               CSL05110
      DOUBLE PRECISION  GR(1),SPUTOL,GAPTOL,SCALE1,TEMP,TOL,TJ,DGAP,ONE CSL05120
      REAL GG(1)                                                        CSL05130
      INTEGER  MP(1)                                                    CSL05140
      REAL  ABS                                                         CSL05150
      DOUBLE PRECISION  DMAX1, CDABS                                    CSL05160
C-----------------------------------------------------------------------CSL05170
C     USE TMINGAPS TO LABEL THE ISOLATED GOOD T-EIGENVALUES             CSL05180
C     THAT ARE VERY CLOSE TO SPURIOUS ONES.  ERROR ESTIMATES            CSL05190
C     WILL NOT BE COMPUTED FOR THESE T-EIGENVALUES.                     CSL05200
C                                                                       CSL05210
C     ON ENTRY AND EXIT                                                 CSL05220
C     VS CONTAINS THE COMPUTED DISTINCT EIGENVALUES OF T(1,MEV)         CSL05230
C     GR(K) = |VS(K)|, K = 1,NDIS, GR(K).LE.GR(K+1)                     CSL05240
C     GG(K) = MIN(J.NE.K,|VS(K)-VS(J)|)   MINGAP                        CSL05250
C     MP CONTAINS THE CORRESPONDING T-MULTIPLICITIES                    CSL05260
C     NDIS = NUMBER OF DISTINCT T-EIGENVALUES                           CSL05270
C     GAPTOL = RELATIVE GAP TOLERANCE SET IN MAIN                       CSL05280
C                                                                       CSL05290
C     ON EXIT                                                           CSL05300
C     MP(J) IS NOT CHANGED EXCEPT THAT  MP(J)=-1, IF MP(J)=1,           CSL05310
C     AND A SPURIOUS T-EIGENVALUE IS TOO CLOSE.                         CSL05320
C                                                                       CSL05330
C     IF MP(I)=-1 THAT SIMPLE GOOD T-EIGENVALUE WILL BE SKIPPED         CSL05340
C     IN THE SUBSEQUENT ERROR ESTIMATE COMPUTATIONS IN INVERR           CSL05350
C     THAT IS, WE COMPUTE ERROR ESTIMATES ONLY FOR THOSE GOOD           CSL05360
C     T-EIGENVALUES WITH MP(J)=1.                                       CSL05370
C-----------------------------------------------------------------------CSL05380
```

```
      ONE = 1.0D0                                                       CSL05390
      DGAP = SCALE1*SPUTOL                                              CSL05400
      NISO = 0                                                          CSL05410
      NG = 0                                                            CSL05420
      DO 40 J = 1,NDIS                                                  CSL05430
      IF (MP(J).EQ.0) GO TO 40                                          CSL05440
      NG = NG+1                                                         CSL05450
      IF (MP(J).NE.1) GO TO 40                                          CSL05460
      TJ = GR(J)                                                        CSL05470
      T0 = VS(J)                                                        CSL05480
      TOL = DMAX1(DGAP,GAPTOL*TJ)                                       CSL05490
C     TOL = DMAX1(ONE,TJ)*GAPTOL                                        CSL05500
C     VS(J) IS NEXT SIMPLE GOOD T-EIGENVALUE                            CSL05510
      NISO = NISO + 1                                                   CSL05520
      IF (ABS(GG(J)).GT.TOL) GO TO 40                                   CSL05530
      I = J                                                             CSL05540
   10 I = I-1                                                           CSL05550
      IF (I.LT.1) GO TO 20                                              CSL05560
      IF (TJ-GR(I).GT.TOL) GO TO 20                                     CSL05570
      IF (MP(I).NE.0) GO TO 10                                          CSL05580
      TEMP = CDABS(T0-VS(I))                                            CSL05590
      IF (TEMP.GT.TOL) GO TO 10                                         CSL05600
      MP(J) = -MP(J)                                                    CSL05610
      NISO = NISO-1                                                     CSL05620
      GO TO 40                                                          CSL05630
   20 I = J                                                             CSL05640
   30 I = I+1                                                           CSL05650
      IF (I.GT.NDIS) GO TO 40                                           CSL05660
      IF (GR(I)-TJ.GT.TOL) GO TO 40                                     CSL05670
      IF (MP(I).NE.0) GO TO 30                                          CSL05680
      TEMP = CDABS(T0-VS(I))                                            CSL05690
      IF (TEMP.GT.TOL) GO TO 30                                         CSL05700
      MP(J) = -MP(J)                                                    CSL05710
      NISO = NISO-1                                                     CSL05720
   40 CONTINUE                                                          CSL05730
C                                                                       CSL05740
C-----END OF ISOEV------------------------------------------------------CSL05750
      RETURN                                                            CSL05760
      END                                                               CSL05770
C---COMPEV------------------------------------------------------------  CSL05780
C                                                                       CSL05790
      SUBROUTINE COMPEV(ALPHA,BETA,V1,V2,VS,EVMAG,MULTOL,SPUTOL,        CSL05800
     1MP,T2FLAG,MEV,NDIS,SAVTEV)                                        CSL05810
C                                                                       CSL05820
C     USES COMPLEX SYMMETRIC VERSION OF IMTQL1, CMTQL1, TO              CSL05830
C     COMPUTE EIGENVALUES OF THE T-MATRIX T(1,MEV).                     CSL05840
C                                                                       CSL05850
C-----------------------------------------------------------------------CSL05860
      COMPLEX*16 ALPHA(1),BETA(1),VS(1),V1(1),V2(1),EVAL,CTEMP          CSL05870
      DOUBLE PRECISION EVMAG(1)                                         CSL05880
      DOUBLE PRECISION TEMP,DGAP,TOL,DELMIN                             CSL05890
      DOUBLE PRECISION MULTOL,SPUTOL,EVALR,EVALC                        CSL05900
      INTEGER MP(1),T2FLAG(1),SAVTEV                                    CSL05910
      DOUBLE PRECISION CDABS, DFLOAT                                    CSL05920
C-----------------------------------------------------------------------CSL05930
```

```
C
      MEV1 = MEV - 1
C
      IF (SAVTEV.GE.0) GO TO 40
C
      READ(10,10) MEV
   10 FORMAT(I6)
   20 FORMAT(20A4)
      MEV1 = MEV - 1
      READ(10,30) (VS(K), K = 1,MEV)
   30 FORMAT(4Z20)
      READ(10,20) EXPLAN
      READ(10,20) EXPLAN
      READ(10,30) (V2(K), K = 1,MEV1)
      GO TO 90
C
   40 CONTINUE
C
      DO 50 J = 1,MEV
      VS(J) = ALPHA(J)
   50 V1(J) = BETA(J)
C
      WRITE(6,60) MEV
   60 FORMAT(/' COMPUTE EIGENVALUES OF T(1,',I4,') USING CMTQL1'/)
C
C-----------------------------------------------------------------------
      CALL CMTQL1(MEV,VS,V1,IERR)
C-----------------------------------------------------------------------
C
C     WRITE(6,70) IERR
   70 FORMAT(' T-EIGENVALUES VIA CMTQL1'/' IERR = ',I6/)
C
      IF (IERR.EQ.0) GO TO 90
C
      WRITE(6,80)
   80 FORMAT(' ON RETURN FROM CMTQL1 ERROR FLAG WAS NOT ZERO'/)
      GO TO 410
C
   90 CONTINUE
C
C     T-EIGENVALUES ARE IN VS IN INCREASING ORDER OF MAGNITUDE
      DO 100 J = 1,MEV
  100 EVMAG(J) = CDABS(VS(J))
C
C     THE MAGNITUDES OF THE T-EIGENVALUES ARE IN EVMAG, IN ORDER OF
C     INCREASING MAGNITUDE
C     WRITE(13,105) (EVMAG(J), J = 1,MEV)
C 105 FORMAT(' MAGNITUDES OF T-EIGENVALUES'/(4E20.12))
C
      IF(SAVTEV.NE.1) GO TO 130
      WRITE(10,110) MEV
  110 FORMAT(I6,' = ORDER OF T-MATRIX, T-EIGVALS =')
      WRITE(10,120) (VS(J), J = 1,MEV)
C 120 FORMAT(4Z20)
  120 FORMAT(4E20.12)
```

```
C                                                                       CSL06490
C                                                                       CSL06500
  130 CONTINUE                                                          CSL06510
      MULTOL = MULTOL*EVMAG(MEV)                                        CSL06520
      SPUTOL = SPUTOL*EVMAG(MEV)                                        CSL06530
      TOL = 1000.0D0*SPUTOL                                             CSL06540
      WRITE(6,140) MULTOL,SPUTOL                                        CSL06550
  140 FORMAT(/' TOLERANCES USED IN T-MULTIPLICITY AND SPURIOUS TESTS =' CSL06560
     1 ,2E10.3/)                                                        CSL06570
C                                                                       CSL06580
C     T-MULTIPLICITY DETERMINATION                                      CSL06590
      J = 0                                                             CSL06600
      NDIS = 0                                                          CSL06610
      DO 150 I = 1,MEV                                                  CSL06620
  150 T2FLAG(I) = 0                                                     CSL06630
C                                                                       CSL06640
  160 J = J+1                                                           CSL06650
      IF (J.GT.MEV) GO TO 190                                           CSL06660
      IF (T2FLAG(J).EQ.1) GO TO 160                                     CSL06670
      CTEMP = VS(J)                                                     CSL06680
      EVAL = CTEMP                                                      CSL06690
      TEMP = EVMAG(J)                                                   CSL06700
      NDIS = NDIS + 1                                                   CSL06710
      INDEX = 1                                                         CSL06720
      T2FLAG(J) = 1                                                     CSL06730
      I = J                                                             CSL06740
  170 I = I+1                                                           CSL06750
      IF (I.GT.MEV) GO TO 180                                           CSL06760
      IF (T2FLAG(I).EQ.1) GO TO 170                                     CSL06770
      DGAP = EVMAG(I)-TEMP                                              CSL06780
      IF (DGAP.GT.MULTOL) GO TO 180                                     CSL06790
      DGAP = CDABS(EVAL-VS(I))                                          CSL06800
      IF (DGAP.GT.MULTOL) GO TO 170                                     CSL06810
C     T-MULTIPLICITY INCREASES                                          CSL06820
      INDEX = INDEX + 1                                                 CSL06830
      CTEMP = CTEMP + VS(I)                                             CSL06840
      T2FLAG(I) = 1                                                     CSL06850
      GO TO 170                                                         CSL06860
C     T-MULTIPLICITY FOR VS(NDIS) HAS BEEN DETERMINED                   CSL06870
  180 VS(NDIS) = CTEMP/DFLOAT(INDEX)                                    CSL06880
      MP(NDIS) = INDEX                                                  CSL06890
      GO TO 160                                                         CSL06900
  190 CONTINUE                                                          CSL06910
C     T-MULTIPLICITY CALCULATION IS COMPLETE                            CSL06920
C                                                                       CSL06930
C     T(2,MEV) EIGENVALUE CALCULATION AND SPURIOUS TESTS                CSL06940
C                                                                       CSL06950
      IF (SAVTEV.LT.0) GO TO 240                                        CSL06960
C                                                                       CSL06970
      WRITE(6,200) MEV1                                                 CSL06980
  200 FORMAT(/' COMPUTE T(2,',I4,') EIGENVALUES'/)                      CSL06990
C                                                                       CSL07000
      DO 210 J = 1,MEV1                                                 CSL07010
      JP1 = J+1                                                         CSL07020
      V2(J) = ALPHA(JP1)                                                CSL07030
```

```
  210 V1(J) = BETA(JP1)                                                 CSL07040
C                                                                       CSL07050
C-----------------------------------------------------------------------CSL07060
      CALL CMTQL1(MEV1,V2,V1,IERR)                                      CSL07070
C-----------------------------------------------------------------------CSL07080
C                                                                       CSL07090
C     WRITE(6,220) IERR                                                 CSL07100
  220 FORMAT(' T2-HAT EIGENVALUES VIA CMTQL1'/' IERR = ',I6/)           CSL07110
C                                                                       CSL07120
      IF (IERR.EQ.0) GO TO 240                                          CSL07130
C                                                                       CSL07140
      WRITE(6,230)                                                      CSL07150
  230 FORMAT(' ON RETURN FROM CMTQL1 ERROR FLAG WAS NOT ZERO'/)         CSL07160
      GO TO 410                                                         CSL07170
C                                                                       CSL07180
  240 CONTINUE                                                          CSL07190
C                                                                       CSL07200
      DO 250 J = 1,MEV1                                                 CSL07210
  250 EVMAG(J) = CDABS(V2(J))                                           CSL07220
C                                                                       CSL07230
C     WRITE(13,255) (EVMAG(J), J = 1,MEV)                               CSL07240
C 255 FORMAT(/' MAGNITUDES OF T2 EIGENVALUES'/(4E20.12))                CSL07250
C                                                                       CSL07260
      IF(SAVTEV.NE.1) GO TO 270                                         CSL07270
      WRITE(10,260) MEV1                                                CSL07280
  260 FORMAT(/I6,' = ORDER OF T2-HAT, T2EIGVALS = ')                    CSL07290
      WRITE(10,120) (V2(J), J = 1,MEV1)                                 CSL07300
  270 CONTINUE                                                          CSL07310
C                                                                       CSL07320
C     SPURIOUS TESTS                                                    CSL07330
      DO 280 I = 1,MEV1                                                 CSL07340
  280 T2FLAG(I) = 0                                                     CSL07350
C                                                                       CSL07360
C     GO THROUGH THE EIGENVALUES OF T2-HAT.  FIND THE CLOSEST EIGENVALUECSL07370
C     OF T(1,MEV).  IF IT IS T-MULTIPLE GO ON.  IF IT IS SIMPLE DECLARE CSL07380
C     SPURIOUS WHENEVER DELMIN < SPUTOL BY SETTING MP(I) = 0            CSL07390
      J = 0                                                             CSL07400
  290 J = J+1                                                           CSL07410
      IF (J.GT.MEV1) GO TO 390                                          CSL07420
C                                                                       CSL07430
C     WRITE(14,300) J,V2(J)                                             CSL07440
  300 FORMAT('EIGENVALUE T2-HAT =', I6,2E22.14)                         CSL07450
C                                                                       CSL07460
      TEMP = EVMAG(J)                                                   CSL07470
      EVAL = V2(J)                                                      CSL07480
      EVALR = TEMP + SPUTOL                                             CSL07490
      EVALC = TEMP - SPUTOL                                             CSL07500
      DELMIN = 2.D0*CDABS(VS(MEV))                                      CSL07510
      IMIN = 0                                                          CSL07520
C     BACKWARD SEARCH                                                   CSL07530
      I = J + 1                                                         CSL07540
  310 I = I - 1                                                         CSL07550
      IF(I.LT.1) GO TO 320                                              CSL07560
      IF(I.GT.NDIS) I = NDIS                                            CSL07570
C                                                                       CSL07580
```

```
      TEMP = CDABS(VS(I))                                               CSL07590
      IF (TEMP.LT.EVALC) GO TO 320                                      CSL07600
      IF(MP(I).EQ.0) GO TO 310                                          CSL07610
      DGAP = CDABS(VS(I) - EVAL)                                        CSL07620
      IF (DGAP.GE.DELMIN) GO TO 310                                     CSL07630
      DELMIN = DGAP                                                     CSL07640
      IMIN = I                                                          CSL07650
C                                                                       CSL07660
      GO TO 310                                                         CSL07670
C     FORWARD SEARCH                                                    CSL07680
  320 I = J                                                             CSL07690
  330 I = I + 1                                                         CSL07700
      IF(I.GT.NDIS) GO TO 340                                           CSL07710
C                                                                       CSL07720
      TEMP = CDABS(VS(I))                                               CSL07730
      IF (TEMP.GT.EVALR) GO TO 340                                      CSL07740
      IF(MP(I).EQ.0) GO TO 330                                          CSL07750
      DGAP = CDABS(VS(I) - EVAL)                                        CSL07760
      IF (DGAP.GE.DELMIN) GO TO 330                                     CSL07770
      DELMIN = DGAP                                                     CSL07780
      IMIN = I                                                          CSL07790
C                                                                       CSL07800
      GO TO 330                                                         CSL07810
C                                                                       CSL07820
  340 CONTINUE                                                          CSL07830
      IF(IMIN.EQ.0) GO TO 370                                           CSL07840
C     WRITE(14,350) IMIN, MP(IMIN),VS(IMIN),DELMIN,J                    CSL07850
  350 FORMAT(/I6,' TH EVALUE, MP =',I3,' EVALUE =',2E22.13/             CSL07860
     1' MINDEL = ',E14.3,' OCCURS FOR',I6,' TH T2-HAT EVALUE')          CSL07870
      IF(DELMIN.GT.SPUTOL) GO TO 290                                    CSL07880
      IF(MP(IMIN).GT.1)  GO TO 290                                      CSL07890
      MP(IMIN) = 0                                                      CSL07900
C     WRITE(14,360)                                                     CSL07910
  360 FORMAT(' ABOVE T-EIGENVALUE IS SPURIOUS')                         CSL07920
      GO TO 290                                                         CSL07930
  370 CONTINUE                                                          CSL07940
      GO TO 290                                                         CSL07950
  390 CONTINUE                                                          CSL07960
C     END OF SPURIOUS TESTS                                             CSL07970
C                                                                       CSL07980
      DO 400 J = 1,NDIS                                                 CSL07990
  400 EVMAG(J) = CDABS(VS(J))                                           CSL08000
C                                                                       CSL08010
      RETURN                                                            CSL08020
C-----END OF COMPEV-----------------------------------------------------CSL08030
  410 STOP                                                              CSL08040
      END                                                               CSL08050
C-----CMTQL1 (EIGENVALUES OF COMPLEX SYMMETRIC TRIDIAGONAL)-------------CSL08060
C                                                                       CSL08070
      SUBROUTINE CMTQL1(N,D,E,IERR)                                     CSL08080
C                                                                       CSL08090
C-----------------------------------------------------------------------CSL08100
      INTEGER I,J,L,M,N,II,MML,IERR                                     CSL08110
      COMPLEX*16 D(1),E(1),B,C,F,G,P,R,S,W,CZERO,CONE                   CSL08120
      COMPLEX*16 CDSQRT,DCMPLX                                          CSL08130
```

```
      DOUBLE PRECISION MACHEP,EPS,TEMP,T0,T1,ZERO,HALF,ONE,TWO          CSL08140
      DOUBLE PRECISION CDABS,DSQRT                                      CSL08150
C-----------------------------------------------------------------------CSL08160
      DATA MACHEP/Z3410000000000000/                                    CSL08170
      EPS = 100.D0*MACHEP                                               CSL08180
C-----------------------------------------------------------------------CSL08190
      ZERO = 0.0D0                                                      CSL08200
      HALF = 0.5D0                                                      CSL08210
      ONE  = 1.0D0                                                      CSL08220
      TWO  = 2.0D0                                                      CSL08230
      CZERO = DCMPLX(ZERO,ZERO)                                         CSL08240
      CONE  = DCMPLX(ONE,ZERO)                                          CSL08250
      IERR = 0                                                          CSL08260
      IF (N.EQ.1) GO TO 160                                             CSL08270
C                                                                       CSL08280
      DO 10 I = 2,N                                                     CSL08290
   10 E(I-1) = E(I)                                                     CSL08300
      E(N) = CZERO                                                      CSL08310
C                                                                       CSL08320
      DO 140 L = 1,N                                                    CSL08330
      J = 0                                                             CSL08340
C                                                                       CSL08350
C     DETERMINE FIRST NEGLIGIBLE SUBDIAGONAL ELEMENT IF ANY             CSL08360
   20 DO 30 M = L,N                                                     CSL08370
      IF (M.EQ.N) GO TO 40                                              CSL08380
      TEMP = CDABS(D(M)) + CDABS(D(M+1))                                CSL08390
      IF (CDABS(E(M)).LE.TEMP*MACHEP) GO TO 40                          CSL08400
   30 CONTINUE                                                          CSL08410
C                                                                       CSL08420
   40 P = D(L)                                                          CSL08430
C                                                                       CSL08440
      IF (M.EQ.L) GO TO 100                                             CSL08450
      IF (J.EQ.100) GO TO 150                                           CSL08460
      J = J+1                                                           CSL08470
C                                                                       CSL08480
C     FORM SHIFT AS EIGENVALUE OF (L,L+1) 2X2 CLOSEST TO D(L)           CSL08490
      G = (D(L+1) - P)*HALF                                             CSL08500
      T0 = CDABS(G)                                                     CSL08510
      T1 = CDABS(E(L))                                                  CSL08520
      IF (T0.GT.T1) GO TO 50                                            CSL08530
      W = G/E(L)                                                        CSL08540
      R = CDSQRT(CONE + W**2)                                           CSL08550
      T0 = CDABS(W + R)                                                 CSL08560
      T1 = CDABS(W - R)                                                 CSL08570
      TEMP = ONE                                                        CSL08580
      IF (T1.GT.T0) TEMP = -ONE                                         CSL08590
      G = D(M) - P + E(L)/(W + TEMP*R)                                  CSL08600
      GO TO 60                                                          CSL08610
   50 CONTINUE                                                          CSL08620
      W = E(L)/G                                                        CSL08630
      R = CDSQRT(CONE + W**2)                                           CSL08640
      T0 = CDABS(CONE + R)                                              CSL08650
      T1 = CDABS(CONE - R)                                              CSL08660
      TEMP = ONE                                                        CSL08670
      IF (T1.GT.T0) TEMP = -ONE                                         CSL08680
```

```
      G = D(M) - P + W*E(L)/(CONE + TEMP*R)                            CSL08690
   60 CONTINUE                                                         CSL08700
C                                                                      CSL08710
C     G IS SHIFTED D(M)                                                CSL08720
C     SPECIFY PARAMETERS FOR I = M-1 CASE, I = M-1,M-2,...,L           CSL08730
C                                                                      CSL08740
      S = CONE                                                         CSL08750
      C = -CONE                                                        CSL08760
      P = CZERO                                                        CSL08770
      MML = M - L                                                      CSL08780
C                                                                      CSL08790
      DO 90 II = 1,MML                                                 CSL08800
      I = M - II                                                       CSL08810
C                                                                      CSL08820
C     FOR I<M-1 F=T(I+2,I), B=NEW E(I), AIM OF (I,I+1) TRANSFORMATION  CSL08830
C     IS TO ZERO OUT F                                                 CSL08840
C                                                                      CSL08850
      F =  S*E(I)                                                      CSL08860
      B = -C*E(I)                                                      CSL08870
      T0 = CDABS(G)                                                    CSL08880
      T1 = CDABS(F)                                                    CSL08890
      IF (T1.GT.T0) GO TO 70                                           CSL08900
C     |G| >= |F|                                                       CSL08910
      W = F/G                                                          CSL08920
      R = CDSQRT(CONE + W**2)                                          CSL08930
      E(I+1) = G*R                                                     CSL08940
      C = CONE/R                                                       CSL08950
      S = W*C                                                          CSL08960
      GO TO 80                                                         CSL08970
C     |F| > |G|                                                        CSL08980
   70 CONTINUE                                                         CSL08990
      W = G/F                                                          CSL09000
      R = CDSQRT(CONE + W**2)                                          CSL09010
      E(I+1) = F*R                                                     CSL09020
      S = CONE/R                                                       CSL09030
      C = W*S                                                          CSL09040
   80 CONTINUE                                                         CSL09050
      TEMP = CDABS(W)**2 + ONE                                         CSL09060
      T0 = DSQRT(TEMP)                                                 CSL09070
      T1 = CDABS(R)                                                    CSL09080
      IERR = -L                                                        CSL09090
      IF (T1.LE.EPS*T0) GO TO 160                                      CSL09100
      IERR = 0                                                         CSL09110
C                                                                      CSL09120
C     C**2 + S**2 = CONE, -Q(I,I) = Q(I+1,I+1) = C, Q(I,I+1) = S       CSL09130
C     Q = Q-TRANSPOSE = Q-INVERSE  RR = CDSQRT(G**2 +F**2)             CSL09140
C     G = D(I+1) AFTER PREVIOUS TRANSFORMATION THEN G = NEW E(I)       CSL09150
C     NEW D(I) = D(I) - S*RR,   NEW D(I+1) = D(I+1) + S*RR             CSL09160
C     NEW E(I) = E(I) - C*RR,   NEW E(I+1) = RR,   P = S*RR            CSL09170
C                                                                      CSL09180
      G = D(I+1) - P                                                   CSL09190
      R = (D(I) - G)*S + TWO*C*B                                       CSL09200
      P = S*R                                                          CSL09210
      D(I+1) = G + P                                                   CSL09220
      G = B - C*R                                                      CSL09230
```

```
   90 CONTINUE
C     END OF I LOOP
C
C     UPDATE PARAMETERS FOR I = L CASE
      D(L) = D(L) - P
      E(L) = G
      E(M) = CZERO
      GO TO 20
C
C     ORDER EIGENVALUES   P = D(L)
  100 IF (L.EQ.1) GO TO 120
      DO 110 II = 2,L
      I = L+2-II
      IF (CDABS(P).GE.CDABS(D(I-1))) GO TO 130
      D(I) = D(I-1)
  110 CONTINUE
C
  120 I = 1
C
  130 D(I) = P
C
  140 CONTINUE
      GO TO 160
C
  150 IERR = L
C-----END OF CMTQL1----------------------------------------------------
  160 RETURN
      END
C
C-----COMGAP-----------------------------------------------------------
C
      SUBROUTINE COMGAP(VC,VA,GG,MP,IND,M,IGAP,ITAG)
C
C----------------------------------------------------------------------
      COMPLEX*16 VC(1),Z
      DOUBLE PRECISION  VA(1),T0,T1,TU,TK
      REAL  GG(1),GTEMP
      INTEGER  MP(1),IND(1)
      REAL ABS
      DOUBLE PRECISION CDABS
C----------------------------------------------------------------------
C     IF IGAP = 0 WE DO NOT ORDER EIGENVALUES BY INCREASING GAP SIZE
C     AND WE DO NOT WRITE GAP OUTPUT TO FILE 12
C
C     VA(K) = |VC(K)|  VA(K) <= VA(K+1)
C     GG(K) = MIN |VC(K)-VC(J)|  J .NE. K.
C----------------------------------------------------------------------
      TU = VA(M) + VA(M)
      K = 0
   10 K = K+1
      IF (K.GT.M) GO TO 60
      INDEX = 0
      T1 = TU
      TK = VA(K)
      Z = VC(K)
```

```
      J = K                                                             CSL09790
C     BACKWARDS                                                         CSL09800
   20 J = J-1                                                           CSL09810
      IF (J.LT.1) GO TO 30                                              CSL09820
      T0 = TK - VA(J)                                                   CSL09830
      IF (T0.GT.T1) GO TO 30                                            CSL09840
      T0 = CDABS(Z - VC(J))                                             CSL09850
      IF (T1.LE.T0) GO TO 20                                            CSL09860
      T1 = T0                                                           CSL09870
      INDEX = J                                                         CSL09880
      GO TO 20                                                          CSL09890
C     FORWARDS                                                          CSL09900
   30 J = K                                                             CSL09910
   40 J = J+1                                                           CSL09920
      IF (J.GT.M) GO TO 50                                              CSL09930
      T0 = VA(J) - TK                                                   CSL09940
      IF (T0.GT.T1) GO TO 50                                            CSL09950
      T0 = CDABS(Z - VC(J))                                             CSL09960
      IF (T1.LE.T0) GO TO 40                                            CSL09970
      T1 = T0                                                           CSL09980
      INDEX = J                                                         CSL09990
      GO TO 40                                                          CSL10000
   50 IND(K) = INDEX                                                    CSL10010
      GG(K) = T1                                                        CSL10020
      IF(ITAG.EQ.0) GO TO 10                                            CSL10030
      IF(MP(INDEX).EQ.0)  GG(K) = -GG(K)                                CSL10040
      GO TO 10                                                          CSL10050
C                                                                       CSL10060
   60 CONTINUE                                                          CSL10070
      IF (IGAP.EQ.0) GO TO 140                                          CSL10080
C                                                                       CSL10090
C     WRITE(12,70)                                                      CSL10100
   70 FORMAT(' MINGAPS FOR GOOD T-EIGENVALUES'/                         CSL10110
     1 1X,'EVNUM',1X,'NEIGH',15X,'R(EV)',15X,'I(EV)',4X,'MINGAP')       CSL10120
C     WRITE(12,80) (K,IND(K),VC(K),GG(K), K = 1,M)                      CSL10130
   80 FORMAT(2I6,2E20.12,E10.3)                                         CSL10140
C                                                                       CSL10150
C     ORDER VC G BY INCREASING MINGAP SIZE                              CSL10160
      DO 90 J = 1,M                                                     CSL10170
      IND(J) = J                                                        CSL10180
   90 CONTINUE                                                          CSL10190
C                                                                       CSL10200
      DO 110 K = 2,M                                                    CSL10210
      KM1 = K-1                                                         CSL10220
      DO 100 L = 1,KM1                                                  CSL10230
      KK = K-L                                                          CSL10240
      KP1 = KK+1                                                        CSL10250
      IF (ABS(GG(KP1)).GE.ABS(GG(KK))) GO TO 110                        CSL10260
      Z = VC(KK)                                                        CSL10270
      VC(KK) = VC(KP1)                                                  CSL10280
      VC(KP1) = Z                                                       CSL10290
      GTEMP = GG(KK)                                                    CSL10300
      GG(KK) = GG(KP1)                                                  CSL10310
      GG(KP1) = GTEMP                                                   CSL10320
      ITEMP = IND(KK)                                                   CSL10330
```

```
      IND(KK) = IND(KP1)                                                CSL10340
      IND(KP1) = ITEMP                                                  CSL10350
  100 CONTINUE                                                          CSL10360
  110 CONTINUE                                                          CSL10370
C                                                                       CSL10380
C     WRITE(12,120)                                                     CSL10390
  120 FORMAT(' T-EIGENVALUES ORDERED BY INCREASING MINGAP'/             CSL10400
     1 1X,'GAPNUM',1X,'EVNUM',15X,'R(EV)',15X,'I(EV)',4X,'MINGAP')      CSL10410
C                                                                       CSL10420
C     WRITE(12,130) (K,IND(K),VC(K),GG(K), K = 1,M)                     CSL10430
  130 FORMAT(I7,I6,2E20.12,E10.3)                                       CSL10440
C                                                                       CSL10450
  140 CONTINUE                                                          CSL10460
C-----END OF COMGAP-----------------------------------------------------CSL10470
      RETURN                                                            CSL10480
      END                                                               CSL10490
C                                                                       CSL10500
C-----START OF INVERM FOR TRIDIAGONAL COMPLEX SYMMETRIC MATRICES--------CSL10510
C                                                                       CSL10520
      SUBROUTINE INVERM(ALPHA,BETA,V1,V2,X1,ERROR,ERRORV,EPS,GR,GC,     CSL10530
     1INTERC,MEV,IT,IWRITE)                                             CSL10540
C                                                                       CSL10550
C-----------------------------------------------------------------------CSL10560
      COMPLEX*16  ALPHA(1),BETA(1),V1(1),V2(1)                          CSL10570
      COMPLEX*16  X1,U,Z,TEMP,RATIO,BETAM,ZEROC                         CSL10580
      DOUBLE PRECISION  SUM,XU,NORM,TSUM,GSUM                           CSL10590
      DOUBLE PRECISION  EPS,EPS3,EPS4,ERROR,ERRORV,ZERO,ONE             CSL10600
      DOUBLE PRECISION GR(1),GC(1)                                      CSL10610
      INTEGER INTERC(1)                                                 CSL10620
      DOUBLE PRECISION  DABS, DSQRT, DFLOAT, CDABS                      CSL10630
C     COMPLEX*16 DCMPLX                                                 CSL10640
C-----------------------------------------------------------------------CSL10650
C                                                                       CSL10660
C     COMPUTES T-EIGENVECTORS FOR ISOLATED GOOD T-EIGENVALUES X1        CSL10670
C     USING INVERSE ITERATION ON T(1,MEV(X1)) SOLVING EQUATION          CSL10680
C     (T - X1*I)V2 = RIGHT-HAND SIDE (RANDOMLY-GENERATED) .             CSL10690
C     PROGRAM REFACTORS T- X1*I ON EACH ITERATION OF INVERSE ITERATION. CSL10700
C     TYPICALLY ONLY ONE ITERATION IS NEEDED PER T-EIGENVALUE X1.       CSL10710
C                                                                       CSL10720
C     IF IWRITE = 1 THEN THERE ARE EXTENDED WRITES TO FILE 6 (TERMINAL) CSL10730
C                                                                       CSL10740
C     ON ENTRY G CONTAINS A REAL*4 RANDOM VECTOR WHICH WAS GENERATED    CSL10750
C     IN MAIN PROGRAM.                                                  CSL10760
C                                                                       CSL10770
C     ON ENTRY AND EXIT                                                 CSL10780
C     MEV = ORDER OF T                                                  CSL10790
C     ALPHA, BETA CONTAIN THE DIAGONAL AND OFFDIAGONAL ENTRIES OF T.    CSL10800
C     EPS = 2. * MACHINE EPSILON                                        CSL10810
C                                                                       CSL10820
C     IN PROGRAM:                                                       CSL10830
C     ITER = MAXIMUM NUMBER STEPS ALLOWED FOR INVERSE ITERATION         CSL10840
C     ITER = IT ON ENTRY.                                               CSL10850
C     V1,V2 = WORK SPACES USED IN THE FACTORIZATION OF T(1,MEV).        CSL10860
C     V1 AND V2 MUST BE OF DIMENSION AT LEAST MEV.                      CSL10870
C                                                                       CSL10880
```

```
C     ON EXIT                                                           CSL10890
C     V2 = THE UNIT EIGENVECTOR OF T(1,MEV) CORRESPONDING TO X1.        CSL10900
C     ERROR = |V2(MEV)| = ERROR ESTIMATE FOR CORRESPONDING              CSL10910
C             RITZ VECTOR FOR X1.                                       CSL10920
C                                                                       CSL10930
C     ERRORV = || T*V2 - X1*V2 || = ERROR ESTIMATE ON T-EIGENVECTOR.    CSL10940
C     IF IT.GT.ITER THEN ERRORV = -ERRORV                               CSL10950
C     IT = NUMBER OF ITERATIONS ACTUALLY REQUIRED                       CSL10960
C-----------------------------------------------------------------------CSL10970
C     INITIALIZATION AND PARAMETER SPECIFICATION                        CSL10980
      ONE  = 1.0D0                                                      CSL10990
      ZERO = 0.0D0                                                      CSL11000
      ZEROC = DCMPLX(ZERO,ZERO)                                         CSL11010
      ITER = IT                                                         CSL11020
      MP1 = MEV+1                                                       CSL11030
      MM1 = MEV-1                                                       CSL11040
      BETAM = BETA(MP1)                                                 CSL11050
      BETA(MP1) = ZEROC                                                 CSL11060
C                                                                       CSL11070
C     CALCULATE SCALE AND TOLERANCES                                    CSL11080
      TSUM = CDABS(ALPHA(1))                                            CSL11090
      DO 10 I = 2,MEV                                                   CSL11100
   10 TSUM = TSUM + CDABS(ALPHA(I)) + CDABS(BETA(I))                    CSL11110
C                                                                       CSL11120
      EPS3 = EPS*TSUM                                                   CSL11130
      EPS4 = DFLOAT(MEV)*EPS3                                           CSL11140
C                                                                       CSL11150
C     GENERATE SCALED RANDOM RIGHT-HAND SIDE                            CSL11160
      GSUM = ZERO                                                       CSL11170
      DO 20 I = 1,MEV                                                   CSL11180
   20 GSUM = GSUM + DABS(GR(I)) + DABS(GC(I))                           CSL11190
      GSUM = EPS4/GSUM                                                  CSL11200
C                                                                       CSL11210
C     INITIALIZE RIGHT HAND SIDE FOR INVERSE ITERATION                  CSL11220
      DO 30 I = 1,MEV                                                   CSL11230
      INTERC(I) = 0                                                     CSL11240
   30 V2(I) = GSUM*DCMPLX(GR(I),GC(I))                                  CSL11250
      IT = 1                                                            CSL11260
C                                                                       CSL11270
C     CALCULATE UNIT EIGENVECTOR OF T(1,MEV) FOR ISOLATED GOOD          CSL11280
C     T-EIGENVALUE X1.                                                  CSL11290
C                                                                       CSL11300
C     TRIANGULAR FACTORIZATION WITH NEAREST NEIGHBOR PIVOT              CSL11310
C     STRATEGY. INTERCHANGES ARE LABELLED BY SETTING INTERC(I)=0        CSL11320
C                                                                       CSL11330
   40 CONTINUE                                                          CSL11340
      U = ALPHA(1)-X1                                                   CSL11350
      Z = BETA(2)                                                       CSL11360
C                                                                       CSL11370
      DO 60 I=2,MEV                                                     CSL11380
      IF (CDABS(BETA(I)).GT.CDABS(U)) GO TO 50                          CSL11390
C     NO PIVOT INTERCHANGE                                              CSL11400
      V1(I-1) = Z/U                                                     CSL11410
      V2(I-1) = V2(I-1)/U                                               CSL11420
      V2(I) = V2(I)-BETA(I)*V2(I-1)                                     CSL11430
```

```
      RATIO = BETA(I)/U
      U = ALPHA(I)-X1-Z*RATIO
      Z = BETA(I+1)
      GO TO 60
C     PIVOT INTERCHANGE
   50 CONTINUE
      RATIO = U/BETA(I)
      INTERC(I) = 1
      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)
   60 CONTINUE
C
      IF (CDABS(U).EQ.ZERO) U= DCMPLX(EPS3,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 80 II = 1,MM1
      I = MEV-II
      IF (INTERC(I+1).EQ.1) GO TO 70
C     NO PIVOT INTERCHANGE
      V2(I) = V2(I)-V1(I)*V2(I+1)
      GO TO 80
C     PIVOT INTERCHANGE
   70 V2(I) = (V2(I)-V1(I)*V2(I+1)-BETA(I+2)*V2(I+2))/BETA(I+1)
   80 CONTINUE
C
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 = CDABS(V2(MEV))
      DO 90 II = 1,MM1
      I = MEV-II
   90 NORM = NORM+CDABS(V2(I))
C
C     IS DESIRED GROWTH IN VECTOR ACHIEVED ?
C     IF NOT, DO ANOTHER INVERSE ITERATION STEP UNLESS NUMBER ALLOWED IS
C     EXCEEDED.
      IF (NORM.GE.ONE) GO TO 110
C
      IT=IT+1
      IF (IT.GT.ITER) GO TO 110
C
      XU = EPS4/NORM
      DO 100 I=1,MEV
      INTERC(I) = 0
```

```
  100 V2(I) = V2(I)*XU
C
      GO TO 40
C
C     NORMALIZE COMPUTED T-EIGENVECTOR : V2 = V2/||V2||
C
  110 CONTINUE
C
C-----------------------------------------------------------------------
      CALL CINPRD(V2,V2,SUM,MEV)
C-----------------------------------------------------------------------
C
      SUM = ONE/DSQRT(SUM)
      DO 120 II = 1,MEV
  120 V2(II) = SUM*V2(II)
C
C     SAVE ERROR ESTIMATE FOR LATER OUTPUT
      ERROR = CDABS(V2(MEV))
C
C     GENERATE ERRORV = ||T*V2 - X1*V2||.
C     LOOP IS BOTTOM UP BECAUSE LAST COMPONENTS MAY BE VERY SMALL
      V1(MEV) = ALPHA(MEV)*V2(MEV)+BETA(MEV)*V2(MEV-1)-X1*V2(MEV)
      DO 130 J = 2,MM1
      JM = MP1 - J
      V1(JM) = ALPHA(JM)*V2(JM) + BETA(JM)*V2(JM-1) + BETA(JM+1)*V2(JM+1
     1) - X1*V2(JM)
  130 CONTINUE
C
      V1(1) = ALPHA(1)*V2(1) + BETA(2)*V2(2) - X1*V2(1)
C
C-----------------------------------------------------------------------
      CALL CINPRD(V1,V1,ERRORV,MEV)
C-----------------------------------------------------------------------
C
      ERRORV = DSQRT(ERRORV)
      IF (IT.GT.ITER) ERRORV = -ERRORV
      IF (IWRITE.EQ.0) GO TO 150
C
C     FILE 6 (TERMINAL) OUTPUT OF ERROR ESTIMATES.
      WRITE(6,140) MEV,X1,ERROR,ERRORV
  140 FORMAT(1X,'TSIZE',10X,'RE(GOODEV)',10X,'IM(GOODEV)',11X,'U(M)',
     1 9X,'TERROR'/I6,2E20.12,2E15.5)
C
C     RESTORE BETA(MEV+1) = BETAM
  150 CONTINUE
      BETA(MP1) = BETAM
C-----END OF INVERM-----------------------------------------------------
      RETURN
      END
C
C-----START OF INNER PRODUCT ROUTINE------------------------------------
C
C     COMPUTES EUCLIDEAN INNER PRODUCT OF 2 COMPLEX VECTORS
C     SUMC = (V2-TRANSPOSE)*V1
C
```

```
      SUBROUTINE INPRDC(V2,V1,SUMC,N)                                   CSL12540
C                                                                       CSL12550
C-----------------------------------------------------------------------CSL12560
      DOUBLE PRECISION  ZERO                                            CSL12570
      COMPLEX*16 V2(1),V1(1),SUMC                                       CSL12580
C-----------------------------------------------------------------------CSL12590
C                                                                       CSL12600
      ZERO = 0.D0                                                       CSL12610
      SUMC = DCMPLX(ZERO,ZERO)                                          CSL12620
      DO 10 J=1,N                                                       CSL12630
   10 SUMC = SUMC + V2(J)*V1(J)                                         CSL12640
C                                                                       CSL12650
      RETURN                                                            CSL12660
C-----END OF EUCLIDEAN INNER PRODUCT SUBROUTINE-------------------------CSL12670
      END                                                               CSL12680
C                                                                       CSL12690
C-----START OF HERMITIAN INNER PRODUCT ROUTINE--------------------------CSL12700
C     COMPLEX INNER PRODUCT                                             CSL12710
C                                                                       CSL12720
      SUBROUTINE CINPRD(V2,V1,SUM,N)                                    CSL12730
C-----------------------------------------------------------------------CSL12740
      DOUBLE PRECISION  ZERO,SUM                                        CSL12750
      COMPLEX*16 V2(1),V1(1),SUMC                                       CSL12760
C-----------------------------------------------------------------------CSL12770
C     COMPUTES THE INNER PRODUCT OF THE CONJUGATE OF V2 WITH V1.        CSL12780
      ZERO = 0.D0                                                       CSL12790
      SUMC = DCMPLX(ZERO,ZERO)                                          CSL12800
      DO 10 J=1,N                                                       CSL12810
   10 SUMC = SUMC + DCONJG(V2(J))*V1(J)                                 CSL12820
      SUM = DREAL(SUMC)                                                 CSL12830
C                                                                       CSL12840
      RETURN                                                            CSL12850
C-----END OF COMPLEX INNER PRODUCT SUBROUTINE---------------------------CSL12860
      END                                                               CSL12870
```

7.7 CSLEVAL: CSLEVEC: File Definitions, Sample Input Files

Below is a listing of the input/output files which are accessed by the complex symmetric Lanczos eigenvalue program, CSLEVAL. Included also is a sample of the input file which CSLEVAL requires on file 5. The parameters in this file are supplied in free format.File 8 contains the data for the nxn complex symmetric matrix A.

CSLEVAL computes eigenvalues of diagonalizable complex symmetric matrices.

```
Sample Specifications of Input/Output Files for CSLEVAL
-------------------------------------------------------------------
 CSLEVAL EXEC LANCZOS EIGENVALUE CALCULATION COMPLEX SYMMETRIC CASE
FI 06 TERM
FILEDEF  1 DISK &1        NHISTORY  A (RECFM F LRECL 80 BLOCK 80
FILEDEF  2 DISK &1        HISTORY   A (RECFM F LRECL 80 BLOCK 80
FILEDEF  3 DISK &1        GOODEV    A (RECFM F LRECL 80 BLOCK 80
FILEDEF  4 DISK &1        ERRINV    A (RECFM F LRECL 80 BLOCK 80
FILEDEF  5 DISK CSLEVAL   INPUT     A (RECFM F LRECL 80 BLOCK 80
FILEDEF  8 DISK &1        INPUT     A (RECFM F LRECL 80 BLOCK 80
FILEDEF 10 DISK &1        T-T2EVAL  A (RECFM F LRECL 80 BLOCK 80
FILEDEF 11 DISK &1        DISTINCT  A (RECFM F LRECL 80 BLOCK 80
LOAD   CSLEVAL   CSLESUB   CSLEMULT
-------------------------------------------------------------------
```

```
Sample Input File for CSLEVAL
---------------------------------------------------------------------
 CSLEVAL INPUT LANCZOS EIGENVALUE COMPUTATION, NO REORTHOGONALIZATION
 OF A NONDEFECTIVE COMPLEX SYMMETRIC MATRIX.
LINE 1     N    KMAX    NMEVS     MATNO
         528     792        1       528
LINE 2   SVSEED     RHSEED      MXINIT
       49302312    5731029           5
LINE 3   ISTART      ISTOP
             0          1
LINE 4  IHIS  IDIST  SAVTEV  IWRITE (SAVE HIST.,DISTINCT EV,TEV,WRITE
           1      0       1       1
LINE 5   RELTOL (RELATIVE TOLERANCE IN 'COMBINING' GOODEV)
    .0000000001
LINE 6   MB(1)   MB(2)   MB(3)   MB(4)    (ORDERS OF T(1,MEV) )
           528
C     NOTE THAT WHEN READING IN PREVIOUSLY COMPUTED EIGENVALUES
C     THE VALUE OF MB(1) MUST BE EQUAL TO THE SIZE AT WHICH
C     THOSE EIGENVALUES WERE COMPUTED AND KMAX MUST BE LISTED AS
C     LARGER THAN MB(1).
---------------------------------------------------------------------
```

Below is a listing of the input/output files which are accessed by the complex symmetric Lanczos eigenvector program, CSLEVEC. Included also is a sample of the input file which CSLEVEC requires on file 5. The parameters in this file are supplied in free format.

File 8 contains the data for the nxn complex symmetric matrix A. CSLEVEC computes eigenvectors for each of a user-specified subset of the eigenvalues computed by the companion program CSLEVAL.

```
Sample Specifications of the Input/Output Files for CSLEVEC
----------------------------------------------------------------
 CSLEVEC EXEC LANCZOS EIGENVECTOR PROGRAM COMPLEX SYMMETRIC CASE
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 CSLEVEC  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  CSLEVEC  CSLESUB  CSLEMULT
----------------------------------------------------------------
```

```
Sample Input File for CSLEVEC
-----------------------------------------------------------------------
 CSLEVEC EIGENVECTORS COMPLEX SYMMETRIC CASE 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
-----------------------------------------------------------------------
```