Chapter 6

Real Rectangular Matrices, Singular Values and Vectors

6.1 Introduction

The FORTRAN codes in this Chapter address the question of computing distinct singular values and corresponding left and right singular vectors of real rectangular matrices, using a single-vector Lanczos procedure. For a given real rectangular mxn matrix A, these codes compute nonnegative scalars σ and corresponding real vectors $x \neq 0$ and $y \neq 0$ such that

$$\begin{aligned} Ax &= \sigma y \\ A^T y &= \sigma x. \end{aligned} \tag{6.1.1}$$

Every real rectangular mxn, $m \geq n$, matrix has a singular value decomposition,

$$A = Y \Sigma X^T, \quad X^T X = I, \quad Y^T Y = I, \quad \Sigma = \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix} \tag{6.1.2}$$

where Σ is $m \times n$, and $\Sigma_1 = \text{diag}\{\sigma_1, \ldots\ldots, \sigma_n\}$ with $\sigma_i, 1 \leq i \leq n$, the singular values of A. X is a $n \times n$ orthogonal matrix, Y is a $m \times m$ orthogonal matrix, and the columns of X and of Y are respectively, right and left singular vectors of A. There are many applications for this type of decomposition. Singular values and vectors are discussed in detail for example in Stewart [24].

Using Eqn(6.1.1), it is not difficult to demonstrate that the singular values of a given real matrix A are just the nonnegative square roots of the eigenvalues of the associated real symmetric matrix $A^T A$. Thus from the perturbation theorems for real symmetric matrices, we have that a small perturbation in the given matrix A causes small perturbations in the singular values. The same arguments demonstrate that the right singular vectors of a matrix A are eigenvectors of the matrix $A^T A$, and the left singular vectors are eigenvectors of the matrix AA^T. Therefore, we also have that the perturbation theorems for eigenvectors of real symmetric matrices apply to the singular vectors.

The Lanczos recursion as presented in Eqns(1.2.1) and (1.2.2) is only applicable to real symmetric matrices. Therefore we ask the question: How do we construct a real symmetric matrix which will give us the desired singular values? Obviously, we could just apply the real symmetric Lanczos recursion to $A^T A$. However in general, these matrices are not suitable because of the effects that squaring a matrix can have on the eigenvalues. Small singular values of A which are close together correspond to eigenvalues of $A^T A$ which are smaller and even closer together. Large singular values of A which are far apart correspond to eigenvalues of A^TA which are larger and further apart. When a matrix A has both small and large singular values, dealing numerically with the square of that matrix is difficult. Lanczos [15] suggested the use of an alternative real symmetric matrix. He proposed that the following larger but real symmetric $[m+n] \times [m+n]$ matrix be used.

$$B = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}. \tag{6.1.3}$$

The relationships between the eigenvalues and the eigenvectors of B and the singular values and singular vectors of A are discussed in detail in Section 5.4 of Chapter 5 in Volume 1.

We could apply the real symmetric version of the Lanczos recursion directly to the matrix B in Eqn(6.1.3). However, because this matrix is considerably larger than the A-matrix, we use a modification of the real symmetric Lanczos recursion which incorporates the following choice of starting vector suggested by Golub and Kahan [11]. We choose a starting vector either of the form $(0, u^T)^T$ or of the form $(v^T, 0)^T$ where u is of length n, the column order of the A-matrix, and v is of length m , the row order of the A-matrix. If we use such a starting vector in the basic Lanczos recursion in Eqns (1.2.1) and (1.2.2), we obtain a version of the Lanczos recursion designed specifically for the B-matrix in Eqn(6.1.3). The Lanczos vectors generated by this recursion alternate in form from either $(0, u^T)^T$ to $(v^T, 0)^T$ or vice-versa, as the iterations proceed. Furthermore, on each iteration of this recursion it is only necessary to either compute Au_i or A^Tv_i . Therefore, the amount of work per iteration of this recursion is no more than applying the real symmetric Lanczos recursion to a real symmetric matrix of order $\max m, n$. For details on the corresponding Lanczos recursion see Section 5.4 of Chapter 5 in Volume 1.

These codes can compute either a very few or very many of the distinct singular values of a given real rectangular matrix. As the documentation in Section 6.2 indicates, the A-multiplicity of a computed singular value 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 which we use generate a family of real symmetric, tridiagonal matrices (T-matrices). The diagonal entries of each of these T-matrices are all 0. The eigenvalues of any even-ordered T-matrix occur in $\pm$ pairs. This latter property is inherited from the B-matrix whose eigenvalues are just $\pm\sigma_i$, the $\pm$ pairs of singular values plus $m - 2n$ additional zero eigenvalues if $m \geq n$. Only even-ordered T-matrices may be used in the Lanczos computations. There is no reorthogonalization of the Lanczos vectors at any stage in any of the computations.

LSVAL, the main program for the single-vector, Lanczos singular value computations, calls the subroutine BISEC to compute eigenvalues of those Lanczos tridiagonal matrices specified by the user and on those subintervals specified by the user. The BISEC subroutine used in this chapter is a modification of the BISEC subroutine given in LESUB in Chapter 2 which assumes that the diagonal entries of the T-matrices supplied to it are all 0. BISEC simultaneously computes the T-eigenvalues and T-multiplicities and then sorts the computed T-eigenvalues into two categories, the 'good' T-eigenvalues and the 'spurious' T-eigenvalues. The 'good' T-eigenvalues are accepted as approximations to singular values of the user-specified matrix A. The accuracy of these 'good' T-eigenvalues as singular values of A is then estimated using error estimates computed by subroutine INVERR. The subroutine INVERR in this chapter is a modification of the INVERR subroutine in Chapter 2 which assumes the diagonal entries of the tridiagonal matrices supplied to it are all 0. Error estimates are computed only for isolated 'good' T-eigenvalues. All other 'good' T-eigenvalues are assumed to have converged. If convergence has not yet occurred and a larger Lanczos matrix has been specified by the user, these programs will continue on repeating the above procedure on a larger Lanczos matrix.

Once the singular values have been computed accurately enough, the user can select a subset of the 'converged' singular values for which singular vectors are to be computed. The main program LSVEC, for computing singular vectors of real rectangular matrices, is then used to compute these desired singular vectors. These singular vectors are obtained by computing Ritz vectors for the B-matrix and then splitting each of these $(m+n)$-dimensional Ritz vectors into approximate left and right singular vectors of A. The user should note that if the singular value being considered is very small, then LSVEC is not able to accurately compute both a left and a right singular vector approximation simultaneously. In this situation one of the two singular vectors will be more accurate than the other one is. If the starting vector is of the form $(0, u^T)^T$, then the right singular vector will be more accurate than the corresponding left vector. Similarly, if we use a starting vector of the form $(v^T, 0)^T$, then the left vector will be more accurate than the right vector will be. This loss in accuracy in one of the two vectors increases as the size of the singular value is decreased, and in the limit for a zero singular value, one of the two computed singular vectors will have no accuracy at all. See Section 5.4 of Chapter 5 in Volume 1.

All computations are in double precision real arithmetic. The user must supply a subroutine USPEC which defines and initializes the user-specified matrix A, and subroutines SVMAT and STRAN which compute respectively, matrix-vector multiplies Ax and A^Ty for any given vectors x and y. These subroutines must be constructed in such a way as to take advantage of the sparsity (and/or structure) of the user-supplied A-matrix and such that these computations are done accurately. More details about these real rectangular, single-vector Lanczos procedures are given in Section 5.4 of Chapter 5 in Volume 1.

6.2 Documentation for the Codes in Chapters 6

```
C-----LSVALHED-----------------------------------------------------
C  Authors:  Jane Cullum and Ralph A. Willoughby (Deceased)
C            Los Alamos National Laboratory
C            Los Alamos, New Mexico 87544
C
C            E-mail:  cullumj@lanl.gov
C
C  These codes are copyrighted by the authors.  These codes
C  and modifications of them or portions of them are NOT to be
C  incorporated into any commercial codes or used for any other
C  commercial purposes such as consulting for other companies,
C  without legal agreements with the authors of these Codes.
C  If these Codes or portions of them are used in other scientific or
C  engineering research works the names of the authors of these codes
C  and appropriate references to their written work are to be
C  incorporated in the derivative works.
C
C  This header is not to be removed from these codes.
C
C
C
C         REFERENCE: Cullum and Willoughby, Chapter 5
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     DOCUMENTATION FOR THE SINGLE-VECTOR
C     LANCZOS SINGULAR VALUE/VECTOR PROGRAMS
C     FOR REAL, RECTANGULAR MATRICES
C
C-----------------------------------------------------------------------
C
C     GIVEN A REAL RECTANGULAR MATRIX A OF ORDER M X N THE THREE
C     SETS OF FORTRAN FILES LABELLED LSVAL, LSSUB, AND LSMULT
C     CAN BE USED TO COMPUTE DISTINCT SINGULAR VALUES OF A IN
C     USER-SPECIFIED INTERVALS.
C
C     CORRESPONDING SINGULAR VECTORS FOR SELECTED, COMPUTED
C     SINGULAR VALUES CAN BE COMPUTED USING THE SETS OF FILES
C     LABELLED LSVEC, LSSUB AND LSMULT.
C
C     THESE PROGRAMS USE LANCZOS TRIDIAGONALIZATION WITHOUT
C     REORTHOGONALIZATION ON THE ASSOCIATED REAL SYMMETRIC MATRIX
C
C                                    ----                 ----
C                                   |  0                   A  |
C                    B  =           |                         |
C                                   |  A-TRANSPOSE         0  |
C                                    ----                 ----
```

```
C
C     OF ORDER M + N  TO GENERATE REAL SYMMETRIC TRIDIAGONAL
C     MATRICES, T(1,MEV), OF ORDER MEV.  SUBSETS OF THE EIGENVALUES OF
C     THESE T-MATRICES, LABELLED AS THE 'GOOD EIGENVALUES' OF T(1,MEV),
C     ARE APPROXIMATIONS TO THE DESIRED SINGULAR VALUES OF A.
C     CORRESPONDING RITZ VECTORS FOR B ARE APPROXIMATIONS TO
C     EIGENVECTORS OF B WHICH IN TURN CONTAIN APPROXIMATIONS TO
C     THE DESIRED LEFT AND RIGHT SINGULAR VECTORS OF A.  THIS
C     PROCEDURE USES A SPECIAL STARTING VECTOR SUGGESTED BY GOLUB
C     AND KAHAN.  THUS, THE STARTING LANCZOS VECTOR IS EITHER OF
C     THE FORM (V1,0) OR (0,V2) WHERE V1 IS MX1 AND V2 IS NX1 AND
C     ALL SUCCEEDING LANCZOS VECTORS GENERATED ALTERNATE BETWEEN
C     THESE 2 FORMS.  THIS SPECIAL CHOICE OF STARTING VECTOR RESULTS
C     IN SIGNIFICANT GAINS IN STORAGE AND OPERATION COUNTS AND
C     ALSO IN CONVERGENCE RELATIVE TO A 'BRUTE FORCE' APPLICATION
C     OF THE REAL SYMMETRIC LANCZOS PROCEDURE DIRECTLY TO THE
C     MATRIX B ABOVE.  FOR MORE DETAILS SEE REFERENCE 1 BELOW.
C     IN THE DISCUSSIONS T(1,MEV) DENOTES THE LANCZOS T-MATRIX
C     OF SIZE MEV.
C
C     THE IDEAS USED IN THESE PROGRAMS ARE DISCUSSED IN THE FOLLOWING
C     REFERENCES.
C
C     1.  JANE CULLUM, RALPH A. WILLOUGHBY AND MARK LAKE, A LANCZOS
C         ALGORITHM FOR COMPUTING SINGULAR VALUES AND VECTORS OF LARGE
C         MATRICES, SIAM J. SCIENTIFIC AND STATISTICAL COMPUTING,
C         VOL. 4, JUNE 1983, PP. 197-215.
C
C     2.  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, 1984.
C
C     3.  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     4. 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     5. IBID, COMPUTING EIGENVALUES OF VERY LARGE SYMMETRIC MATRICES-
C        AN IMPLEMENTATION OF A LANCZOS ALGORITHM WITHOUT
C        REORTHOGONALIZATION, J. COMPUT. PHYS. 44(1981), 329-358.
C
C
C-----PORTABILITY-----------------------------------------------------
C
C
C     PROGRAMS WERE TESTED FOR PORTABILITY USING THE PFORT VERIFIER.
C     FOR DETAILS OF THE VERIFIER SEE FOR EXAMPLE, B. G. RYDER AND
```

```
C     A. D. HALL, "THE PFORT VERIFIER", COMPUTING SCIENCE TECHNICAL     LSV01080
C     REPORT 12, BELL LABORATORIES, MURRAY HILL, NEW JERSEY 07974,      LSV01090
C     (REVISED), JANUARY 1981.                                          LSV01100
C                                                                       LSV01110
C     EXCEPT FOR THE FOLLOWING CONSTRUCTIONS WHICH CAN BE EASILY        LSV01120
C     MODIFIED BY THE USER TO MATCH THE PARTICULAR COMPUTER BEING       LSV01130
C     USED, THE PROGRAM STATEMENTS ARE PORTABLE.                        LSV01140
C                                                                       LSV01150
C    NONPORTABLE CONSTRUCTIONS.                                         LSV01160
C                                                                       LSV01170
C    IN LSVAL AND IN LSVEC                                              LSV01180
C         1.  DATA/MACHEP STATEMENT                                     LSV01190
C         2.  ALL READ(5,*) STATEMENTS (FREE FORMAT)                    LSV01200
C         3.  FORMAT(20A4) USED FOR THE EXPLANATORY HEADER ARRAY, EXPLANLSV01210
C         4.  FORMAT(4Z20) USED TO READ AND WRITE BETA FILES 1 AND 2.   LSV01220
C    IN LSMULT                                                          LSV01230
C         1.  IN SVMAT, STRAN, AND  USPEC THE ENTRY THAT PASSES THE     LSV01240
C             STORAGE LOCATIONS OF THE ARRAYS DEFINING THE              LSV01250
C             USER-SPECIFIED MATRIX.                                    LSV01260
C         2.  IN SAMPLE USPEC FOR 'DIAGONAL' MATRICES:  THE FREE        LSV01270
C             FORMAT (8,*) AND THE FORMAT (20A4).                       LSV01280
C    IN LSSUB                                                           LSV01290
C         1.  ALL STATEMENTS ARE PORTABLE.                              LSV01300
C                                                                       LSV01310
C                                                                       LSV01320
C     IN THE COMMENTS BELOW:                                            LSV01330
C     COMPLEX*16 = COMPLEX VARIABLE, 16 BYTES OF STORAGE                LSV01340
C     REAL*8 = REAL VARIABLE, 8 BYTES OF STORAGE                        LSV01350
C     REAL*4 = REAL VARIABLE, 4 BYTES OF STORAGE                        LSV01360
C     INTEGER*4 = INTEGER VARIABLE, 4 BYTES                             LSV01370
C                                                                       LSV01380
C                                                                       LSV01390
C-----A-MATRIX SPECIFICATION--------------------------------------------LSV01400
C                                                                       LSV01410
C                                                                       LSV01420
C     SUBROUTINE USPEC IS USED TO SPECIFY THE USER-SUPPLIED A-MATRIX.   LSV01430
C     SUBROUTINES SVMAT AND STRAN ARE, RESPECTIVELY, CORRESPONDING      LSV01440
C     MATRIX-VECTOR MULTIPLE SUBROUTINES FOR A AND FOR A-TRANSPOSE.     LSV01450
C     THESE SUBROUTINES SHOULD BE DESIGNED TO TAKE ADVANTAGE OF         LSV01460
C     ANY SPECIAL PROPERTIES OF THE USER-SUPPLIED MATRIX.  THE          LSV01470
C     MATRIX-VECTOR MULTIPLIES REQUIRED BY THE LANCZOS PROCEDURES       LSV01480
C     MUST BE COMPUTED RAPIDLY AND ACCURATELY.                          LSV01490
C                                                                       LSV01500
C     SUBROUTINE USPEC HAS THE CALLING SEQUENCE                         LSV01510
C                                                                       LSV01520
C            CALL USPEC(M,N,MATNO)                                      LSV01530
C                                                                       LSV01540
C     WHERE M IS THE NUMBER OF ROWS IN THE USER-SPECIFIED               LSV01550
C     A-MATRIX AND N IS THE NUMBER OF COLUMNS. MATNO IS A               LSV01560
C     <= 8 DIGIT INTEGER USED AS A MATRIX AND TEST IDENTIFICATION       LSV01570
C     NUMBER.  THIS SUBROUTINE DEFINES (DIMENSIONS) THE ARRAYS          LSV01580
C     REQUIRED TO SPECIFY THE A-MATRIX.  THIS SUBROUTINE ALSO           LSV01590
C     INITIALIZES THESE ARRAYS AND ANY OTHER PARAMETERS NEEDED TO       LSV01600
C     DEFINE THE MATRIX.  THE STORAGE LOCATIONS OF THESE PARAMETERS     LSV01610
C     AND ARRAYS ARE THEN PASSED TO THE MATRIX-VECTOR MULTIPLY          LSV01620
```

```
C     SUBROUTINES SVMAT AND STRAN VIA ENTRIES.  SAMPLE SUBROUTINES
C     ARE INCLUDED IN THE FORTRAN FILE LSMULT.
C
C     IMPORTANT NOTE:
C     THE SAMPLE MATRIX-VECTOR MULTIPLY SUBROUTINES IN LSMULT
C     ASSUME THAT M >= N.  THEY ALSO ASSUME THAT THE USER-SUPPLIED
C     INFORMATION ABOUT THE GIVEN MATRIX IS STORED ON FILE 8.
C     THE USER SHOULD SEE THE LSMULT PROGRAMS FOR MORE DETAILS.
C
C     SUBROUTINE SVMAT HAS THE CALLING SEQUENCE
C
C            CALL SVMAT(W,U,SUM)
C
C     WHERE U AND W ARE REAL*8 VECTORS AND SUM IS A REAL*8
C     SCALAR.  SVMAT CALCULATES U = A*W - SUM*U FOR THE
C     USER-SPECIFIED A-MATRIX.  SUBROUTINE STRAN HAS THE
C     CALLING SEQUENCE
C
C            CALL STRAN(W,U,SUM)
C
C     STRAN CALCULATES U = (A-TRANSPOSE)*W - SUM*U FOR THE
C     TRANSPOSE OF THE USER-SUPPLIED A-MATRIX.  THE ARRAY AND PARAMETER
C     INFORMATION NEEDED TO PERFORM THE MATRIX-VECTOR MULTIPLIES
C     IS PASSED TO THE SVMAT AND THE STRAN SUBROUTINES FROM THE
C     USPEC SUBROUTINE VIA ENTRIES.  ONE SET OF THE SAMPLE SVMAT
C     AND STRAN SUBROUTINES INCLUDED IN LSMULT COMPUTES
C     MATRIX-VECTOR MULTIPLIES FOR AN ARBITRARY SPARSE,
C     RECTANGULAR MATRIX STORED IN THE SPARSE FORMAT SPECIFIED
C     IN THE CORRESPONDING SAMPLE USPEC SUBROUTINE.  THE LANCZS
C     SUBROUTINE CALLS SVMAT AND STRAN IN THE GENERATION OF
C     THE LANCZOS T-MATRICES FOR THE B MATRIX.
C
C     THE DATA FOR THE A-MATRIX IS ASSUMED TO BE ON FILE 8 AND
C     IN THE FOLLOWING SPARSE FORMAT:
C     NZ = NUMBER OF NONZERO ELEMENTS OF A
C     ICOL(K), K = 1,N, NUMBER OF NONZEROS OF A IN COLUMN K.
C     IROW(K), K = 1,NZ, ROW INDEX OF A(K).
C     A(K), K=1,NZ  CONTAINS THE ELEMENTS OF A BY COLUMN.
C
C
C     SVMATV AND STRAN ARE CALLED FROM THE SUBROUTINE LANCZS
C     WHICH GENERATES THE LANCZOS TRIDIAGONAL MATRICES, THE
C     BETA HISTORY.   SIMILARLY, THSE SUBROUTINES ARE CALLED FROM
C     THE CORRESPONDING SINGULAR VECTOR PROGRAM, LSVEC.
C     SVMAT AND STRAN ARE DECLARED AS EXTERNAL VARIABLES.
C     EACH IS AN ARGUMENT FOR THE LANCZS SUBROUTINE.
C
C     USPEC, SVMAT, AND STRAN SUBROUTINES SUITABLE FOR THE
C     USER-SPECIFIED MATRIX MUST BE SUPPLIED BY THE USER.
C
C     THE MAIN PROGRAMS FOR THE SINGULAR VALUE AND SINGULAR VECTOR
C     CALCULATIONS ASSUME THAT INPUT FILE 5 CONTAINS THE ROW ORDER
C     M AND THE COLUMN ORDER N OF THE GIVEN A-MATRIX AND MATNO,
C     AN IDENTIFICATION NUMBER OF <= 8 DIGITS FOR THE GIVEN MATRIX.
C
```

```
C
C-----MACHEP------------------------------------------------------------
C
C
C     MACHEP IS A MACHINE DEPENDENT PARAMETER SPECIFYING THE RELATIVE
C     PRECISION OF THE FLOATING POINT ARITHMETIC USED.
C     MACHEP = 2.2 * 10**-16 FOR DOUBLE PRECISION ARITHMETIC ON
C     IBM 370-3081.
C
C     THE USER WILL HAVE TO RESET THIS PARAMETER TO
C     THE CORRESPONDING VALUE FOR THE MACHINE BEING USED.  NOTE THAT
C     IF A MACHINE WITH A MACHINE EPSILON THAT IS MUCH LARGER THAN THE
C     VALUE GIVEN HERE IS BEING USED, THEN THERE COULD BE
C     PROBLEMS WITH THE TOLERANCES.
C
C
C-----SUBROUTINES AND FUNCTIONS USER MUST SUPPLY------------------------
C
C
C     GENRAN,  FINPRO,  MASK,  USPEC,  SVMAT AND  STRAN
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     FINPRO = DOUBLE PRECISION FUNCTION WHICH COMPUTES THE INNER
C              PRODUCT OF 2 DOUBLE PRECISION VECTORS OF DIMENSION  K.
C              TESTS REPORTED IN THE REFERENCES USED THE HARWELL
C              LIBRARY SUBROUTINE FM02AD.
C              EXISTING CALLING SEQUENCE IS
C
C                      CALL FINPRO(N,V,J,W,K).
C
C              COMPUTES THE INNER PRODUCT OF DIMENSION N OF THE VECTORS
C              V AND W.  SUCCESSIVE COMPONENTS OF V AND OF W ARE STORED
C              AT LOCATIONS THAT ARE ,RESPECTIVELY, J AND K UNITS APART.
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 A-MATRIX.  SEE A-MATRIX SPECIFICATION SECTION
C
```

```
C    SVMAT = MATRIX-VECTOR MULTIPLY FOR USER-SUPPLIED A-MATRIX.         LSV02730
C            SEE A-MATRIX SPECIFICATION SECTION.                        LSV02740
C                                                                       LSV02750
C    STRAN = MATRIX-VECTOR MULTIPLY FOR TRANSPOSE OF USER-SUPPLIED      LSV02760
C            A-MATRIX.  SEE A-MATRIX SPECIFICATION SECTION.             LSV02770
C                                                                       LSV02780
C                                                                       LSV02790
C-----------------------------------------------------------------------LSV02800
C                                                                       LSV02810
C     COMMENTS FOR SINGULAR VALUE COMPUTATIONS                          LSV02820
C                                                                       LSV02830
C-----------------------------------------------------------------------LSV02840
C                                                                       LSV02850
C                                                                       LSV02860
C-----PARAMETER CONTROLS FOR SINGULAR VALUE PROGRAMS--------------------LSV02870
C                                                                       LSV02880
C                                                                       LSV02890
C     PARAMETER CONTROLS ARE INTRODUCED TO ALLOW SEGMENTATION OF THE    LSV02900
C     SINGULAR VALUE COMPUTATIONS AND TO ALLOW VARIOUS COMBINATIONS     LSV02910
C     OF READ/WRITES.                                                   LSV02920
C                                                                       LSV02930
C     THE FLAG ISTART CONTROLS THE T-MATRIX (BETA HISTORY)              LSV02940
C     GENERATION.                                                       LSV02950
C                                                                       LSV02960
C     ISTART = (0,1)  MEANS                                             LSV02970
C                                                                       LSV02980
C              (0) THERE IS NO EXISTING BETA HISTORY AND ONE            LSV02990
C                  MUST BE GENERATED.                                   LSV03000
C                                                                       LSV03010
C              (1) THERE IS AN EXISTING BETA HISTORY AND IT IS          LSV03020
C                  TO BE READ IN FROM FILE 2 AND EXTENDED IF NECESSARY. LSV03030
C                                                                       LSV03040
C     THE FLAG ISTOP CAN BE USED IN CONJUNCTION WITH THE FLAG ISTART TO LSV03050
C     ALLOW SEGMENTATION OF THE SINGULAR VALUE COMPUTATIONS.            LSV03060
C                                                                       LSV03070
C     ISTOP  = (0,1)  MEANS                                             LSV03080
C                                                                       LSV03090
C              (0) PROGRAM COMPUTES ONLY THE REQUESTED BETAS,           LSV03100
C                  STORES THEM AND THE LAST 2 LANCZOS VECTORS GENERATED LSV03110
C                  IN FILE 1 AND THEN TERMINATES.                       LSV03120
C                                                                       LSV03130
C              (1) PROGRAM COMPUTES REQUESTED BETAS AND THEN            LSV03140
C                  USES THE BISEC SUBROUTINE TO CALCULATE EIGENVALUES   LSV03150
C                  OF THE TRIDIAGONAL MATRICES GENERATED FOR THE ORDERS LSV03160
C                  SPECIFIED BY THE USER AND ON THE USER-SPECIFIED      LSV03170
C                  INTERVALS.  PROGRAM THEN USES THE SUBROUTINE INVERR  LSV03180
C                  TO COMPUTE ERROR ESTIMATES FOR THE ISOLATED GOOD     LSV03190
C                  T-EIGENVALUES WHICH ARE USED TO CHECK THE            LSV03200
C                  CONVERGENCE OF THESE T-EIGENVALUES.                  LSV03210
C                                                                       LSV03220
C     CONTROL PARAMETERS FOR WRITES                                     LSV03230
C                                                                       LSV03240
C     IHIS  = (0,1)  MEANS                                              LSV03250
C                                                                       LSV03260
C             (0) IF ISTOP .GT. 0 THEN BETAS ARE NOT SAVED ON FILE 1.   LSV03270
```

```
C                                                                       LSV03280
C             (1) PROGRAM WRITES BETAS AND LAST 2 LANCZOS               LSV03290
C                 VECTORS TO FILE 1 SO THAT THE T-MATRIX GENERATION     LSV03300
C                 MAY BE REUSED OR CONTINUED LATER IF NECESSARY.        LSV03310
C                 TYPICALLY ONE WOULD ALWAYS DO THIS ON ANY RUN WHERE   LSV03320
C                 A HISTORY FILE IS BEING GENERATED.  HISTORY MUST BE   LSV03330
C                 SAVED IN MACHINE FORMAT ((4Z20) FOR IBM/3081) SO      LSV03340
C                 THAT NO ERRORS DUE TO FORMAT CONVERSIONS OCCUR.       LSV03350
C                                                                       LSV03360
C     IDIST  = (0,1)  MEANS                                             LSV03370
C                                                                       LSV03380
C              (0) DISTINCT EIGENVALUES OF T-MATRICES ARE NOT SAVED.    LSV03390
C                                                                       LSV03400
C              (1) PROGRAM WRITES COMPUTED DISTINCT EIGENVALUES OF      LSV03410
C                  T-MATRICES ALONG WITH THEIR T-MULTIPLICITIES         LSV03420
C                  TO FILE 11.                                          LSV03430
C                                                                       LSV03440
C     IWRITE = (0,1)   MEANS                                            LSV03450
C                                                                       LSV03460
C              (0) NO EXTENDED OUTPUT FROM SUBROUTINES BISEC AND INVERR LSV03470
C                  IS SENT TO FILE 6.                                   LSV03480
C                                                                       LSV03490
C              (1) INDIVIDUAL COMPUTED T-EIGENVALUES AND CORRESPONDING  LSV03500
C                  ERROR ESTIMATES FROM THE SUBROUTINES BISEC AND INVERRLSV03510
C                  ARE PRINTED OUT TO FILE 6 AS THEY ARE COMPUTED.      LSV03520
C                                                                       LSV03530
C     THE PROGRAM ALWAYS MAKES A SEPARATE LIST OF THE COMPUTED GOOD     LSV03540
C     EIGENVALUES OF THE LANCZOS MATRICES T(1,MEV) CONSIDERED,          LSV03550
C     THESE ARE THE APPROXIMATIONS TO THE DESIRED SINGULAR VALUES,      LSV03560
C     ALONG WITH THEIR MINIMAL GAPS AS SINGULAR VALUES OF A AND         LSV03570
C     WRITES THEM TO FILE 3.  CORRESPONDING ERROR ESTIMATES FOR ANY     LSV03580
C     ISOLATED COMPUTED GOOD T-EIGENVALUES (SINGULAR VALUES OF A)       LSV03590
C     ARE ALWAYS WRITTEN TO FILE 4.                                     LSV03600
C                                                                       LSV03610
C                                                                       LSV03620
C-----INPUT/OUTPUT FILES FOR SINGULAR VALUE PROGRAMS--------------------LSV03630
C                                                                       LSV03640
C     ANY INPUT DATA OTHER THAN THE BETA HISTORY SHOULD BE STORED       LSV03650
C     ON FILE 5. SEE SAMPLE INPUT/OUTPUT FROM TYPICAL RUN.              LSV03660
C     THE READ STATEMENTS IN THE GIVEN FORTRAN PROGRAM ASSUME THAT      LSV03670
C     THE DATA STORED ON FILE 5 IS IN FREE FORMAT.  USER SHOULD NOTE    LSV03680
C     THAT 'FREE FORMAT' IS NOT CLASSIFIED AS PORTABLE BY PFORT SO THAT LSV03690
C     THE USER MAY HAVE TO MODIFY THE READ STATEMENTS FROM FILE 5 TO    LSV03700
C     CONFORM TO WHAT IS PERMISSIBLE ON THE MACHINE BEING USED.         LSV03710
C                                                                       LSV03720
C     FILE 6 WAS USED AS THE INTERACTIVE TERMINAL OUTPUT FILE.          LSV03730
C     THIS FILE PROVIDES A RUNNING ACCOUNT OF THE PROGRESS OF THE       LSV03740
C     COMPUTATIONS.  THE AMOUNT OF INFORMATION PRINTED OUT IS           LSV03750
C     CONTROLLED BY THE PARAMETER IWRITE.                               LSV03760
C                                                                       LSV03770
C DESCRIPTION OF OTHER I/O FILES                                        LSV03780
C                                                                       LSV03790
C FILE (K)     CONTAINS:                                                LSV03800
C                                                                       LSV03810
C      (1)     OUTPUT FILE:                                             LSV03820
```

```
C              HISTORY FILE OF NEWLY-GENERATED T-MATRIX
C              (BETA VECTOR) AND LAST 2 LANCZOS VECTORS USED
C              IN THE T-MATRIX GENERATION.
C              IF IHIS = 0 AND ISTOP = 1, FILE 1 IS NOT WRITTEN.
C
C       (2)    INPUT FILE:
C              SAME AS FILE 1 EXCEPT THAT IT CONTAINS A
C              PREVIOUSLY-GENERATED T-MATRIX (IF ANY). IF ISTART = 1,
C              PROGRAM ASSUMES THAT THERE IS A HISTORY FILE OF
C              BETAS ON FILE 2.  THESE BETAS AND THE LAST TWO LANCZOS
C              VECTORS USED IN THE T-MATRIX GENERATION ARE READ IN.
C
C       (3)    OUTPUT FILE:
C              COMPUTED GOOD EIGENVALUES OF THE T-MATRICES CONSIDERED.
C              ALSO CONTAINS T-MULTIPLICITIES OF THESE T-EIGENVALUES AS
C              EIGENVALUES OF THE T-MATRIX, AND THEIR GAPS AS
C              EIGENVALUES IN THE B MATRIX AND IN THE T-MATRIX.
C              NOTE THAT THESE GOOD T-EIGENVALUES ARE THE COMPUTED
C              SINGULAR VALUES OF THE A-MATRIX AND THAT THE GAPS
C              OF THESE EIGENVALUES AS EIGENVALUES OF THE B-MATRIX
C              ARE EQUAL TO THEIR GAPS AS SINGULAR VALUES OF A.  FILE
C              3 IS ALWAYS WRITTEN.
C
C       (4)    OUTPUT FILE:
C              ERROR ESTIMATES FOR THE ISOLATED COMPUTED SINGULAR
C              SINGULAR VALUES (ISOLATED GOOD EIGENVALUES OF T(1,MEV))
C              THESE ARE OBTAINED USING THE SUBROUTINE INVERR. THESE
C              ESTIMATES USE THE LAST COMPONENTS OF THE ASSOCIATED
C              T-EIGENVECTORS WHICH ARE COMPUTED USING INVERSE
C              ITERATION.  FILE 4 IS ALWAYS WRITTEN.
C
C
C       (8)    INPUT FILE:
C              SAMPLE USPEC SUBROUTINE ASSUMES THAT THE ARRAYS
C              REQUIRED TO SPECIFY THE USER'S MATRIX ARE STORED ON
C              FILE 8.  USERS MUST MAKE WHATEVER DEFINITIONS ARE
C              APPROPRIATE FOR THEIR MATRICES.
C
C       (9)    OUTPUT FILE:  OPTIONAL
C              CAN BE USED TO STORE THE TRUE SINGULAR VALUES OF
C              A GIVEN TEST MATRIX, WHEN THE SINGULAR VALUE PROCEDURE
C              IS BEING EXERCISED ON A TEST MATRIX.
C
C      (11)    OUTPUT FILE:
C              COMPUTED DISTINCT EIGENVALUES OF T-MATRICES USED.
C              ALSO CONTAINS THEIR T-MULTIPLICITIES AND T-GAPS TO
C              NEAREST DISTINCT T-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 SINGULAR VALUE PROGRAMS---------------------
C
C
C     THESE PARAMETERS ARE SET INTERNALLY IN THE PROGRAM
```

```
C                                                                       LSV04380
C     SCALEK     K = 1,2,3,4                                            LSV04390
C                                                                       LSV04400
C                THE SCALING FACTORS SCALEK HAVE BEEN INTRODUCED IN AN  LSV04410
C                ATTEMPT TO MAKE THE TOLERANCES USED IN THE             LSV04420
C                T-MULTIPLICITY, SPURIOUS, ISOLATION AND PRTESTS ADJUST LSV04430
C                TO THE SCALE OF THE GIVEN MATRIX.  THESE FACTORS MUST  LSV04440
C                NOT BE MODIFIED.                                       LSV04450
C                                                                       LSV04460
C     NOTE:    THE USER SHOULD NOTE THAT IF THE MATRIX BEING            LSV04470
C     PROCESSED IS VERY STIFF, THAT IS THE RATIO OF THE LARGEST         LSV04480
C     SINGULAR VALUE TO THE SMALLEST SINGULAR VALUE IS VERY             LSV04490
C     LARGE, THEN THE TOLERANCES BEING USED IN BISEC, LUMP, ISOEV       LSV04500
C     AND PRTEST MAY NOT TREAT THE SMALLEST SINGULAR VALUES             LSV04510
C     VERY WELL.  IN SOME SUCH CASES A USER-INTRODUCED REDUCTION        LSV04520
C     IN THE SIZE OF TKMAX AND THE SUBSEQUENT RECOMPUTATION OF          LSV04530
C     THE T-MATRIX EIGENVALUES CORRESPONDING TO THE SMALLEST            LSV04540
C     SINGULAR VALUES USING THIS TKMAX MAY RESULT IN IMPROVED           LSV04550
C     COMPUTATIONS AT THE LOW END.                                      LSV04560
C                                                                       LSV04570
C     THE LUMP, ISOEV, AND PRTEST TOLERANCES THAT WERE USED             LSV04580
C     MOST IN  THE TESTING OF THIS ALGORITHM WERE NOT                   LSV04590
C     SCALE INVARIANT BUT SEEMED TO WORK WELL ON MATRICES THAT          LSV04600
C     HAD SINGULAR VALUES BOTH GREATER THAN AND LESS                    LSV04610
C     THAN 1.  THESE TOLERANCES ARE ALSO INCLUDED IN THESE THREE        LSV04620
C     SUBROUTINES BUT AS COMMENTED OUT STATEMENTS.  THEY CAN BE          LSV04630
C     REVIVED BY COMMENTING OUT THE CORRESPONDING TOLERANCES            LSV04640
C     SPECIFIED IN THE STATEMENT ABOVE EACH OF THESE.                   LSV04650
C                                                                       LSV04660
C     IMPORTANT TOLERANCES OR SCALES THAT ARE USED REPEATEDLY           LSV04670
C     THROUGHOUT THIS PROGRAM ARE THE FOLLOWING:                        LSV04680
C     SCALED MACHINE EPSILON:  TTOL = TKMAX*EPSM WHERE                  LSV04690
C     EPSM = 2*MACHINE EPSILON AND                                      LSV04700
C     TKMAX = MAX(BETA(J), J = 1,MEV)                                   LSV04710
C     BISEC CONVERGENCE TOLERANCE:  BISTOL = DSQRT(1000+MEV)*TTOL       LSV04720
C     BISEC T-MULTIPLICITY TOLERANCE:  MULTOL = (1000+MEV)*TTOL         LSV04730
C     LANCZOS CONVERGENCE TOLERANCE:   CONTOL = BETA(MEV+1)*1.D-10      LSV04740
C                                                                       LSV04750
C                                                                       LSV04760
C     BTOL = RELATIVE TOLERANCE USED TO ESTIMATE ANY LOSS OF LOCAL      LSV04770
C            ORTHOGONALITY OF THE LANCZOS VECTORS AFTER THE T-MATRIX    LSV04780
C            HAS BEEN GENERATED.  THE LANCZOS PROCEDURE WORKS WELL      LSV04790
C            ONLY IF LOCAL ORTHOGONALITY BETWEEN SUCCESSIVE LANCZOS     LSV04800
C            VECTORS IS MAINTAINED.  THE TNORM SUBROUTINE TESTS         LSV04810
C            WHETHER OR NOT                                             LSV04820
C                                                                       LSV04830
C                 MINIMUM  |BETA(I)|/||A|| > BTOL.                      LSV04840
C                 I=2,KMAX                                              LSV04850
C                                                                       LSV04860
C            IF THIS TEST IS VIOLATED BY SOME BETA AND A T-MATRIX THAT  LSV04870
C            WOULD INCLUDE SUCH A BETA IS REQUESTED, THEN THE LANCZOS   LSV04880
C            PROCEDURE WILL TERMINATE FOR THE USER TO DECIDE WHAT TO    LSV04890
C            DO.  THE USER CAN OVER-RIDE THIS TEST BY SIMPLY DECREASING LSV04900
C            THE SIZE OF BTOL, BUT THEN CONVERGENCE IS NOT AS CERTAIN.  LSV04910
C            THE PROGRAM SETS BTOL = 1.D-8 WHICH IS A VERY CONSERVATIVE LSV04920
```

```
C            CHOICE. THE || A || IS ESTIMATED BY USING AN ESTIMATE
C            OF THE NORM OF THE T-MATRIX, T(1,KMAX).
C
C     GAPTOL = RELATIVE TOLERANCE USED IN THE SUBROUTINE ISOEV
C              TO DETERMINE FOR WHICH OF THE GOOD T-EIGENVALUES,
C              THE COMPUTED SINGULAR VALUES, ERROR ESTIMATES SHOULD
C              BE COMPUTED.  THE PROGRAM SETS GAPTOL = 1.D-8.
C              IF FOR A GIVEN 'GOOD' T-EIGENVALUE OF THE GIVEN
C              T-MATRIX  THE COMPUTED GAP IN THE T-MATRIX IS TOO
C              SMALL AND IS DUE TO A 'SPURIOUS' EIGENVALUE OF
C              THE T-MATRIX, THEN THE 'GOOD' T-EIGENVALUE IS ASSUMED
C              TO HAVE CONVERGED AND AN ERROR ESTIMATE IS NOT
C              COMPUTED.
C
C
C-----USER-SPECIFIED PARAMETERS FOR SINGULAR VALUE PROGRAMS-------------
C
C
C     RELTOL = RELATIVE TOLERANCE USED IN 'COMBINING' COMPUTED
C              EIGENVALUES OF T(1,MEV) PRIOR TO COMPUTING ERROR
C              ESTIMATES.
C
C     THE LUMPING OF T-EIGENVALUES OCCURS IN SUBROUTINE LUMP.
C     LUMPING IS NECESSARY BECAUSE IT IS IMPOSSIBLE TO ACCURATELY
C     PREDICT THE ACCURACY OF THE BISEC SUBROUTINE.  LUMP 'COMBINES'
C     T-EIGENVALUES THAT HAVE SLIPPED BY THE TOLERANCE THAT WAS USED
C     IN THE T-MULTIPLICITY TESTS.  IN PARTICULAR IF FOR SOME J,
C
C   |EVALUE(J)-EVALUE(J-1)| < DMAX1(RELTOL*|EVALUE(J)|,SCALE2*MULTOL)
C
C     THEN THESE T-EIGENVALUES ARE 'COMBINED'.  MULTOL IS THE TOLERANCE
C     THAT WAS USED IN THE T-MULTIPLICITY TEST IN BISEC.  SEE THE HEADER
C     ON THE LUMP SUBROUTINE FOR MORE DETAILS.
C
C     RELTOL IS SET TO 1.D-10.
C
C     MXINIT = MAXIMUM NUMBER OF INVERSE ITERATIONS ALLOWED IN
C              SUBROUTINE INVERR FOR EACH ISOLATED GOOD T-EIGENVALUE
C              CONSIDERED.  TYPICALLY ONLY ONE IS REQUIRED.
C
C     SEEDS FOR RANDOM NUMBER GENERATORS = INTEGER*4 SCALARS.
C
C               (1) SVSEED = SEED FOR STARTING VECTOR USED IN
C                   T-MATRIX GENERATION IN LANCZS SUBROUTINE
C
C               (2) RHSEED = SEED FOR RIGHT-HAND SIDE USED IN
C                   INVERSE ITERATION COMPUTATIONS IN INVERR.
C
C     BISEC DATA
C
C     (1) NINT  =  NUMBER OF SUBINTERVALS ON WHICH SINGULAR VALUES
C                  ARE TO BE COMPUTED.
C
C     (2) LB(J) = (J = 1,NINT)  =  LEFT END POINTS OF THESE INTERVALS.
C                 MUST BE PROVIDED IN INCREASING ORDER.  THAT IS,
```

```
C               LB(J) < LB(J+1) FOR J = 1,NINT.                         LSV05480
C                                                                       LSV05490
C     (3) UB(J) = (J = 1,NINT) =  RIGHT END POINTS OF THESE INTERVALS.  LSV05500
C               MUST BE PROVIDED IN INCREASING ORDER.  THAT IS,         LSV05510
C               UB(J) < UB(J+1) FOR J = 1,NINT.                         LSV05520
C                                                                       LSV05530
C     (4) MXSTUR  =  MAXIMUM NUMBER OF STURM ITERATIONS ALLOWED FOR     LSV05540
C                  ENTIRE SET OF SINGULAR VALUE CALCULATIONS OVER       LSV05550
C                  ALL SPECIFIED SIZE T-MATRICES.  PROGRAM WILL         LSV05560
C                  TERMINATE IF THIS LIMIT IS EXCEEDED.                 LSV05570
C                                                                       LSV05580
C     T-MATRICES                                                        LSV05590
C                                                                       LSV05600
C     SIZES OF T-MATRICES                                               LSV05610
C                                                                       LSV05620
C             (1) KMAX= MAXIMUM ORDER FOR T-MATRIX THAT USER IS WILLING LSV05630
C                     TO CONSIDER.                                      LSV05640
C                                                                       LSV05650
C             (2) NMEVS = MAXIMUM NUMBER OF T-MATRICES THAT WILL BE     LSV05660
C                       CONSIDERED.                                     LSV05670
C                                                                       LSV05680
C             (3) NMEV(J)  (J=1,NMEVS)  = SIZES OF T-MATRIX TO BE       LSV05690
C                                         CONSIDERED SEQUENTIALLY.      LSV05700
C                                                                       LSV05710
C     T-MATRIX-GENERATION                                               LSV05720
C                                                                       LSV05730
C     IPAR   = (1,2)   MEANS                                            LSV05740
C                                                                       LSV05750
C              (1) STARTING VECTOR IS OF FORM (0,V2) WHERE V2 IS        LSV05760
C                  NX1.  USE WHEN M > N .                               LSV05770
C                                                                       LSV05780
C              (2) STARTING VECTOR IF OF FORM (V1,0) WHERE V1 IS        LSV05790
C                  MX1.  USE WHEN M < N .                               LSV05800
C                                                                       LSV05810
C     USER SHOULD NOTE THAT THIS PROGRAM FIRST COMPUTES A T-MATRIX      LSV05820
C     OF ORDER KMAX AND THEN CYCLES THROUGH THE T-MATRICES SPECIFIED    LSV05830
C     A PRIORI BY THE USER, USING THE SUBROUTINE BISEC TO COMPUTE       LSV05840
C     EIGENVALUES OF THE T-MATRICES ON THE INTERVALS SPECIFIED BY       LSV05850
C     THE USER.  SUBSETS OF THESE T-EIGENVALUES ARE THEN SELECTED       LSV05860
C     AS APPROXIMATIONS TO THE DESIRED SINGULAR VALUES.                 LSV05870
C                                                                       LSV05880
C     IDEALLY, ONE WOULD COMPUTE THE SINGULAR VALUE APPROXIMATIONS      LSV05890
C     AT A REASONABLE SIZE T-MATRIX, LOOK AT THE ACCURACY OF THE        LSV05900
C     COMPUTED RESULTS AND USE THAT TO DETERMINE AN APPROPRIATE         LSV05910
C     INCREMENT FOR THE SIZE OF THE T-MATRIX BASED UPON WHAT            LSV05920
C     HAS ALREADY CONVERGED AND UPON THE SIZES OF THE ERROR ESTIMATES   LSV05930
C     ON THOSE SINGULAR VALUES THAT ARE DESIRED BUT THAT HAVE NOT       LSV05940
C     YET CONVERGED.  HOWEVER, IN THE INTERESTS OF GENERALITY AND       LSV05950
C     SIMPLICITY WE CHOSE NOT TO DO THAT HERE.                          LSV05960
C                                                                       LSV05970
C                                                                       LSV05980
C-----CONVERGENCE TESTS FOR THE SINGULAR VALUE PROGRAMS-----------------LSV05990
C                                                                       LSV06000
C                                                                       LSV06010
C     THE CONVERGENCE TEST INCORPORATED IN THIS PROGRAM IS              LSV06020
```

```
C     BASED UPON THE ASSUMPTION THAT THOSE T-EIGENVALUES AND            LSV06030
C     THEIR ASSOCIATED T-EIGENVECTORS THAT CORRESPOND TO                LSV06040
C     THE SINGULAR VALUES AND VECTORS WHICH WE WISH TO COMPUTE          LSV06050
C     CONVERGE AS THE T-SIZE IS INCREASED.                              LSV06060
C                                                                       LSV06070
C     AS CURRENTLY PROGRAMMED, CONVERGENCE IS CHECKED BY EXAMINING      LSV06080
C     THE SIZES OF ALL OF THE COMPUTED ERROR ESTIMATES ON ALL OF THE    LSV06090
C     INTERVALS SPECIFIED BY THE USER.  IDEALLY CONVERGENCE SHOULD      LSV06100
C     BE CHECKED ONLY ON THOSE SINGULAR VALUES OF INTEREST AND          LSV06110
C     ONCE THE SINGULAR VALUES ON SUB-INTERVALS OF THESE INTERVALS      LSV06120
C     HAVE CONVERGED, ANY SUBSEQUENT SINGULAR VALUE COMPUTATIONS        LSV06130
C     SHOULD BE MADE ONLY ON THE UNCONVERGED PORTIONS.  OBVIOUSLY,      LSV06140
C     IT WOULD BE DIFFICULT TO INCORPORATE CODE TO DO THE ABOVE         LSV06150
C     WITHOUT KNOWING A PRIORI PRECISELY WHAT THE USER IS TRYING        LSV06160
C     TO COMPUTE.  THEREFORE, WE DID NOT ATTEMPT TO DO THIS.  IF        LSV06170
C     ONE WISHES TO MAKE SUCH A MODIFICATION THEN ONE MUST ALSO         LSV06180
C     MODIFY THE PROGRAM SO THAT IT CREATES AN OVERALL LIST OF THE      LSV06190
C     CONVERGED SINGULAR VALUES AS THEY ARE COMPUTED, SINCE             LSV06200
C     CONVERGED SINGULAR VALUES OBTAINED AT A PARTICULAR VALUE OF       LSV06210
C     MEV WOULD NO LONGER BE RECOMPUTED AT LARGER VALUES OF MEV.        LSV06220
C                                                                       LSV06230
C     IF ONLY A FEW SINGULAR VALUES ARE TO BE COMPUTED THEN SUCH        LSV06240
C     CHANGES WOULD NOT MAKE MUCH DIFFERENCE IN THE RUNNING TIME.       LSV06250
C                                                                       LSV06260
C                                                                       LSV06270
C-----ARRAYS REQUIRED BY THE SINGULAR VALUE PROGRAMS--------------------LSV06280
C                                                                       LSV06290
C                                                                       LSV06300
C     BETA(J) = REAL*8 ARRAY.  ITS DIMENSION MUST BE AT LEAST KMAX+1.   LSV06310
C               THE LENGTH OF THE LARGEST T-MATRIX ALLOWED.  THIS       LSV06320
C               ARRAY CONTAINS THE SUBDIAGONAL ENTRIES OF THE           LSV06330
C               T-MATRICES.  THE DIAGONAL ENTRIES ARE ALL ZERO.         LSV06340
C                                                                       LSV06350
C               THE BETA VECTOR IS NOT ALTERED DURING THE               LSV06360
C               CALCULATIONS.  IMPORTANT NOTE:  ONLY EVEN ORDER         LSV06370
C               T-MATRICES ARE PERMISSIBLE.                             LSV06380
C                                                                       LSV06390
C     V1(J),V2(J),VS(J) = REAL*8 ARRAYS.   VS MUST BE OF                LSV06400
C                         DIMENSION AT LEAST KMAX.  V1 MUST BE          LSV06410
C                         OF DIMENSION AT LEAST MAX(M,KMAX+1).          LSV06420
C                         V2 MUST BE OF DIMENSION AT LEAST              LSV06430
C                         MAX(N,KMAX). M IS THE ROW DIMENSION OF        LSV06440
C                         A, AND N IS THE COLUMN DIMENSION.             LSV06450
C                         HOWEVER, THE DIMENSION                        LSV06460
C                         FOR V2 IS VALID ONLY IF NO MORE               LSV06470
C                         THAN KMAX/2 EIGENVALUES OF THE GIVEN          LSV06480
C                         T-MATRICES ARE TO BE COMPUTED IN ANY GIVEN    LSV06490
C                         SUBINTERVAL. V2 IS USED IN THE SUBROUTINE     LSV06500
C                         BISEC TO HOLD THE UPPER AND LOWER             LSV06510
C                         ENDPOINTS OF THE SUBINTERVALS GENERATED       LSV06520
C                         DURING THE BISECTIONS.  THEREFORE, ITS        LSV06530
C                         DIMENSION MUST ALWAYS BE AT LEAST 2*Q         LSV06540
C                         WHERE Q IS THE MAXIMUM NUMBER OF              LSV06550
C                         EIGENVALUES OF THE SPECIFIED T-MATRIX IN ANY  LSV06560
C                         ONE OF THE SPECIFIED INTERVALS.               LSV06570
```

```
C                                                                       LSV06580
C     LB(J),UB(J) = REAL*8 ARRAYS. EACH MUST BE OF DIMENSION AT LEAST   LSV06590
C                   NINT, THE NUMBER OF SUBINTERVALS TO BE CONSIDERED.  LSV06600
C                   LB CONTAINS THE LEFT-END POINTS OF THE INTERVALS    LSV06610
C                   ON WHICH SINGULAR VALUES ARE TO BE COMPUTED.        LSV06620
C                   UB CONTAINS THE RIGHT-END POINTS.                   LSV06630
C                                                                       LSV06640
C     EXPLAN(J) = REAL*4 ARRAY.  ITS DIMENSION IS 20.  THIS ARRAY IS    LSV06650
C                 USED TO ALLOW EXPLANATORY COMMENTS IN THE INPUT FILES.LSV06660
C                                                                       LSV06670
C     G(J) = REAL*4 ARRAY.  ITS DIMENSION MUST BE >= MAX(2*KMAX,M,N)    LSV06680
C            IT IS USED FOR HOLDING THE RANDOM VECTORS GENERATED,       LSV06690
C            HOLDING THE COMPUTED ERROR ESTIMATES AND THE COMPUTED      LSV06700
C            MINIMAL GAPS FOR THE SINGULAR VALUES.                      LSV06710
C                                                                       LSV06720
C     MP(J) = INTEGER*4 ARRAY.  ITS DIMENSION MUST BE AT LEAST KMAX,    LSV06730
C             THE MAXIMUM SIZE OF THE T-MATRICES ALLOWED.  IT CONTAINS  LSV06740
C             THE T-MULTIPLICITIES OF THE COMPUTED T-EIGENVALUES OF     LSV06750
C             THE T-MATRICES.  NOTE THAT 'SPURIOUS' EIGENVALUES         LSV06760
C             OF THE T-MATRICES ARE DENOTED BY A T-MULTIPLICITY OF      LSV06770
C             0.  T-EIGENVALUES THAT THE SUBROUTINE PRTEST HAS          LSV06780
C             IDENTIFIED AS 'GOOD' BUT HIDDEN ARE IDENTIFIED BY A       LSV06790
C             T-MULTIPLICITY OF -10 AND SUBSEQUENTLY ADDED TO THE LIST  LSV06800
C             OF COMPUTED SINGULAR VALUES.                              LSV06810
C                                                                       LSV06820
C     NMEV(J) = INTEGER*4 ARRAY.  ITS DIMENSION MUST BE AT LEAST THE    LSV06830
C               NUMBER OF T-MATRICES ALLOWED.  IT CONTAINS THE ORDERS   LSV06840
C               OF THE T-MATRICES TO BE CONSIDERED.                     LSV06850
C                                                                       LSV06860
C                                                                       LSV06870
C     OTHER ARRAYS                                                      LSV06880
C                                                                       LSV06890
C     THE USER MUST SPECIFY IN THE SUBROUTINE USPEC WHATEVER ARRAYS     LSV06900
C     ARE REQUIRED TO DEFINE THE MATRIX BEING USED.                     LSV06910
C                                                                       LSV06920
C                                                                       LSV06930
C-----SUBROUTINES INCLUDED----------------------------------------------LSV06940
C                                                                       LSV06950
C                                                                       LSV06960
C     LANCZS = COMPUTES THE BETA HISTORY. USES SUBROUTINES              LSV06970
C              FINPRO, GENRAN, SVMAT AND STRAN.                         LSV06980
C                                                                       LSV06990
C     BISEC = COMPUTES EIGENVALUES OF THE SPECIFIED T-MATRIX USING      LSV07000
C             STURM SEQUENCING, ON SEQUENCE OF INTERVALS SPECIFIED      LSV07010
C             BY THE USER.  EACH SUBINTERVAL IS TREATED AS OPEN         LSV07020
C             ON THE LEFT AND CLOSED ON THE RIGHT.  EIGENVALUES         LSV07030
C             ARE COMPUTED WITH SIMULTANEOUS DETERMINATION OF THE       LSV07040
C             T-MULTIPLICITIES AND OF WHICH T-EIGENVALUES ARE SPURIOUS. LSV07050
C                                                                       LSV07060
C     INVERR = USES INVERSE ITERATION ON T-MATRICES TO COMPUTE ERROR    LSV07070
C              ESTIMATES ON COMPUTED SINGULAR VALUES. (USES GENRAN)     LSV07080
C                                                                       LSV07090
C     LUMP = 'COMBINES' EIGENVALUES OF T-MATRIX USING THE RELATIVE      LSV07100
C             TOLERANCE RELTOL.                                         LSV07110
C                                                                       LSV07120
```

```
C     ISOEV = CALCULATES GAPS BETWEEN DISTINCT EIGENVALUES OF T-MATRIX  LSV07130
C             AND THEN USES THESE GAPS TO LABEL THOSE 'GOOD'            LSV07140
C             T-EIGENVALUES FOR WHICH ERROR ESTIMATES ARE NOT COMPUTED. LSV07150
C                                                                       LSV07160
C     TNORM = COMPUTES THE SCALE TKMAX USED IN DETERMINING THE          LSV07170
C             TOLERANCES FOR THE SPURIOUS, T-MULTIPLICITY AND PRTESTS.  LSV07180
C             IT ALSO CHECKS FOR LOCAL ORTHOGONALITY OF THE LANCZOS     LSV07190
C             VECTORS BY TESTING THE RELATIVE SIZE OF THE BETAS USING   LSV07200
C             THE RELATIVE TOLERANCE BTOL.                              LSV07210
C                                                                       LSV07220
C     PRTEST = LOOKS FOR 'GOOD' T-EIGENVALUES THAT HAVE BEEN MISLABELLEDLSV07230
C              BY THE SPURIOUS TEST BECAUSE THEY HAD 'TOO SMALL' A      LSV07240
C              PROJECTION ON THE STARTING LANCZOS VECTOR.               LSV07250
C              (LESS THAN SINGLE PRECISION)                             LSV07260
C              TESTS INDICATE THAT SUCH T-EIGENVALUES ARE RARE.         LSV07270
C              PRTEST SHOULD BE CALLED ONLY AFTER CONVERGENCE           LSV07280
C              HAS BEEN ESTABLISHED.                                    LSV07290
C                                                                       LSV07300
C     INVERM = USED TO COMPUTE ERROR ESTIMATES FOR ANY T-EIGENVALUES    LSV07310
C              WHICH PRTEST INDICATES MAY HAVE BEEN MISLABELLED.        LSV07320
C              SUCH T-EIGENVALUES ARE RELABELLED ONLY IF THEIR ERROR    LSV07330
C              ESTIMATES ARE SUFFICIENTLY SMALL.  PRIMARY USE OF        LSV07340
C              INVERM IS IN THE CORRESPONDING SINGULAR VECTOR PROGRAM.  LSV07350
C                                                                       LSV07360
C     SAMPLE USPEC, SVMAT AND STRAN SUBROUTINES ARE INCLUDED.           LSV07370
C                                                                       LSV07380
C     ALSO INCLUDED IS A STAND-ALONE PROGRAM, LSCOMPAC, THAT            LSV07390
C     TRANSLATES A MATRIX GIVEN IN THE I,J, A(I,J) FORMAT INTO          LSV07400
C     THE PARTICULAR SPARSE MATRIX FORMAT USED IN THE SAMPLE USPEC,     LSV07410
C     SVMAT AND STRAN SUBROUTINES PROVIDED.                             LSV07420
C                                                                       LSV07430
C                                                                       LSV07440
C-----OTHER PROGRAMS PROVIDED-------------------------------------------LSV07450
C                                                                       LSV07460
C                                                                       LSV07470
C     LSCOMPAC = STAND-ALONE PROGRAM THAT TRANSLATES A SPARSE           LSV07480
C                RECTANGULAR M X N MATRIX A, GIVEN AS I, J, A(I,J),     LSV07490
C                INTO THE SPARSE MATRIX FORMAT REQUIRED BY THE SAMPLE   LSV07500
C                USPEC, STRAN AND SVMAT SUBROUTINES PROVIDED FOR USE    LSV07510
C                IN THE SINGULAR VALUE/VECTOR PROGRAMS.                 LSV07520
C                THIS PROGRAM ASSUMES THAT THE MATRIX ENTRIES ARE       LSV07530
C                GIVEN EITHER COLUMN BY COLUMN OR ROW BY ROW.  IT       LSV07540
C                CANNOT HANDLE ANY OTHER ORDERINGS.  IN FACT IF         LSV07550
C                THE ENTRIES ARE GIVEN ROW BY ROW, THE DATA SET         LSV07560
C                CREATED ON FILE 8 CORRESPONDS TO A-TRANSPOSE AND       LSV07570
C                NOT TO A.  THUS, IN THIS SITUATION, IN ANY             LSV07580
C                SUBSEQUENT USE OF THE LANCZOS SINGULAR VALUE/VECTOR    LSV07590
C                PROGRAMS THE USER WILL HAVE TO INTERCHANGE THE         LSV07600
C                ROLES OF M AND OF N.                                   LSV07610
C                                                                       LSV07620
C                                                                       LSV07630
C-----COMMENTS ON THE STORAGE REQUIRED FOR SINGULAR VALUE PROGRAMS------LSV07640
C                                                                       LSV07650
C                                                                       LSV07660
C     THE ARRAYS IN THE REAL SINGULAR VALUE PROGRAM REQUIRE             LSV07670
```

```
C     APPROXIMATELY THE EQUIVALENT OF ONE REAL*8 ARRAY OF DIMENSION
C
C        2.5*KMAX + MAX(KMAX,M) + MAX(KMAX,N) + .5* MAX(2*KMAX,M,N)
C
C     PLUS WHATEVER IS NEEDED TO GENERATE A*X FOR THE GIVEN MATRIX A.
C     THE ARRAYS BETA, VS AND MP CONSUME 2.5*KMAX*8 BYTES.
C     THE ARRAY V1 CONSUMES MAXIMUM(KMAX+1,M)*8 BYTES,  THE
C     ARRAY V2 CONSUMES MAXIMUM(KMAX,N)*8 BYTES, WITH THE
C     QUALIFICATION STATED ABOVE WHERE V2 IS DEFINED.  THE G-ARRAY
C     CONSUMES .5*MAX(2*KMAX,M,N)*8 BYTES.
C
C
C-----------------------------------------------------------------------
C
C     COMMENTS FOR SINGULAR VECTOR COMPUTATIONS
C
C-----------------------------------------------------------------------
C
C
C     THE SINGULAR VALUES WHOSE SINGULAR VECTORS ARE TO BE COMPUTED
C     MUST HAVE BEEN COMPUTED USING THE CORRESPONDING LANCZOS
C     SINGULAR VALUE PROGRAMS FOR REAL RECTANGULAR MATRICES BECAUSE
C     THESE SINGULAR VECTOR PROGRAMS USE THE SAME FAMILY OF LANCZOS
C     TRIDIAGONAL MATRICES THAT WAS USED IN THE CORRESPONDING
C     SINGULAR VALUE COMPUTATIONS.
C
C     THESE PROGRAMS ASSUME THAT THE SINGULAR VALUES SUPPLIED TO IT
C     HAVE BEEN COMPUTED ACCURATELY, AS MEASURED BY THE
C     ERROR ESTIMATES COMPUTED IN THE CORRESPONDING LANCZOS
C     SINGULAR VALUE COMPUTATIONS, ALTHOUGH THESE ESTIMATES
C     ARE TYPICALLY CONSERVATIVE.   THE SINGULAR VALUES SUPPLIED
C     ARE STORED IN THE ARRAY GOODSV(J), J=1,NGOOD.
C
C     FOR EACH GOODSV(J), THE SUBROUTINE STURMI COMPUTES THE
C     SMALLEST SIZE LANCZOS TRIDIAGONAL MATRIX, T(1,M1(J)), FOR
C     WHICH  GOODSV(J) IS A T-EIGENVALUE TO WITHIN A SPECIFIED
C     TOLERANCE.  IT ALSO ATTEMPTS TO COMPUTE THE SIZE, M2(J),
C     BY WHICH THE GIVEN SINGULAR VALUE BECOMES A DOUBLE
C     T-EIGENVALUE TO WITHIN THE GIVEN TOLERANCE.  THESE SIZES ARE
C     USED TO DETERMINE 1ST GUESSES AT SIZES FOR THE T-EIGENVECTORS
C     THAT WILL BE USED IN THE SINGULAR VECTOR COMPUTATIONS.
C     SUBROUTINE INVERM SUCCESSIVELY COMPUTES CORRESPONDING
C     T-EIGENVECTORS OF ENLARGED T-MATRICES UNTIL A SUITABLE
C     SIZE T-MATRIX IS DETERMINED FOR EACH J.  UP TO 10 SUCH
C     T-EIGENVECTOR COMPUTATIONS ARE ALLOWED FOR EACH SINGULAR
C     VALUE SUPPLIED.
C
C     AFTER APPROPRIATE T-EIGENVECTORS HAVE BEEN COMPUTED,
C     RITZ VECTORS FOR THE MATRIX B CORRESPONDING TO THESE
C     T-EIGENVECTORS ARE THEN COMPUTED.  SECTIONS OF THESE
C     RITZ VECTORS ARE THEN TAKEN AS APPROXIMATE LEFT AND
C     RIGHT SINGULAR VECTORS CORRESPONDING TO THE GIVEN
C     SINGULAR VALUES GOODSV(J),  J = 1,...,NGOOD.
C
C     THIS IMPLEMENTATION FIRST COMPUTES ALL OF THE RELEVANT
```

```
C     T-EIGENVECTORS OF THE SYMMETRIC TRIDIAGONAL MATRICES
C     IN THE VECTOR, TVEC.
C
C     THEN, AS EACH OF THE LANCZOS VECTORS IS REGENERATED, ALL
C     OF THE B-MATRIX 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 B-MATRIX  RITZ VECTOR.  THE ARRAY RITVEC
C     CONTAINS THE SUCCESSIVE RITZ VECTORS WHICH ARE THEN
C     SPLIT INTO APPROXIMATIONS TO THE LEFT AND RIGHT SINGULAR
C     VECTORS OF THE USER-SUPPLIED MATRIX A.
C
C
C-----PARAMETER CONTROLS FOR SINGULAR VECTOR PROGRAMS-------------------
C
C
C     PARAMETER CONTROLS ARE INTRODUCED TO ALLOW SEGMENTATION OF THE
C     SINGULAR VECTOR COMPUTATIONS AND TO ALLOW VARIOUS COMBINATIONS
C     OF 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     SINGULAR VALUES WHOSE SINGULAR VECTORS 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                   SINGULAR VALUES SUPPLIED AND THEN CONTINUES
C                   WITH THE CORRESPONDING T-EIGENVECTOR
C                   COMPUTATIONS.
C
C              (1)  PROGRAM COMPUTES FIRST GUESSES AT THE SIZES
C                   OF THE T-MATRICES REQUIRED BY EACH OF THE
C                   SINGULAR VALUES SUPPLIED, STORES THESE IN FILE
C                   10 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     CRITERIA FOR INTERMEDIATE POINTS IN THE LANCZOS PROCEDURE.  THEY
C     TERMINATE THE PROCEDURE IF VARIOUS SPECIFIED QUANTITIES COULD
C     NOT 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     LSV08780
C                    THE USER-SPECIFIED DIMENSION OF THE TVEC ARRAY.    LSV08790
C                    IN THIS SITUATION THE PROGRAM COMPUTES AS MANY     LSV08800
C                    T-EIGENVECTORS AS IT HAS ROOM FOR, IN THE SAME     LSV08810
C                    ORDER IN WHICH THE SINGULAR VALUES ARE SUPPLIED.   LSV08820
C                                                                       LSV08830
C      SVTVEC = (0,1) MEANS                                             LSV08840
C                                                                       LSV08850
C               (0)  DO NOT STORE THE COMPUTED T-EIGENVECTORS ON        LSV08860
C                    FILE 11 UNLESS ALSO HAVE THE FLAG TVSTOP = 1,      LSV08870
C                    IN WHICH CASE THE T-EIGENVECTORS ARE ALWAYS        LSV08880
C                    WRITTEN TO FILE 11.                                LSV08890
C                                                                       LSV08900
C               (1)  STORE THE COMPUTED T-EIGENVECTORS ON FILE 11.      LSV08910
C                                                                       LSV08920
C      TVSTOP = (0,1) MEANS                                             LSV08930
C                                                                       LSV08940
C               (0)  ATTEMPT TO CONTINUE ON TO THE COMPUTATION          LSV08950
C                    OF THE B-MATRIX RITZVECTORS AFTER COMPLETING THE   LSV08960
C                    COMPUTATION OF THE T-EIGENVECTORS.                 LSV08970
C                                                                       LSV08980
C               (1)  TERMINATE AFTER COMPUTING THE                      LSV08990
C                    T-EIGENVECTORS AND STORING THEM ON FILE 11.        LSV09000
C                                                                       LSV09010
C      LVCONT = (0,1) MEANS                                             LSV09020
C                                                                       LSV09030
C               (0)  IF SOME OF THE T-EIGENVECTORS THAT WERE            LSV09040
C                    REQUESTED WERE NOT COMPUTED, EXIT                  LSV09050
C                    FROM THE PROGRAM WITHOUT COMPUTING THE             LSV09060
C                    CORRESPONDING RITZ VECTORS.                        LSV09070
C                                                                       LSV09080
C               (1)  CONTINUE ON TO THE RITZ VECTOR COMPUTATIONS        LSV09090
C                    EVEN IF NOT ALL OF THE T-EIGENVECTORS THAT         LSV09100
C                    WERE REQUESTED WERE COMPUTED.                      LSV09110
C                                                                       LSV09120
C      ERCONT = (0,1) MEANS                                             LSV09130
C                                                                       LSV09140
C               (0)  PROGRAM WILL NOT COMPUTE THE RITZ                  LSV09150
C                    VECTOR FOR ANY SINGULAR VALUE FOR WHICH NO         LSV09160
C                    T-EIGENVECTOR WHICH SATISFIES THE ERROR            LSV09170
C                    ESTIMATE TEST (ERTOL) HAS BEEN IDENTIFIED.         LSV09180
C                                                                       LSV09190
C               (1)  A RITZ VECTOR WILL BE COMPUTED FOR EVERY           LSV09200
C                    SINGULAR VALUE FOR WHICH A T-EIGENVECTOR HAS BEEN  LSV09210
C                    COMPUTED REGARDLESS OF WHETHER OR NOT THAT         LSV09220
C                    T-EIGENVECTOR SATISFIES THE ERROR ESTIMATE TEST.   LSV09230
C                                                                       LSV09240
C                                                                       LSV09250
C-----INPUT/OUTPUT FILES FOR THE SINGULAR VECTOR COMPUTATIONS-----------LSV09260
C                                                                       LSV09270
C                                                                       LSV09280
C     ANY INPUT DATA OTHER THAN THE T-MATRIX HISTORY FILE AND THE       LSV09290
C     PREVIOUSLY COMPUTED SINGULAR VALUES AND ERROR ESTIMATES           LSV09300
C     SHOULD BE STORED ON FILE 5 IN FREE FORMAT.  SEE SAMPLE            LSV09310
C     INPUT/OUTPUT FOR TYPICAL INPUT FILE.                              LSV09320
```

```
C
C     FILE 6 WAS USED AS THE INTERACTIVE TERMINAL OUTPUT FILE.
C     THIS FILE PROVIDES A RUNNING ACCOUNT OF THE PROGRESS OF THE
C     COMPUTATIONS.  ADDITIONAL PRINTOUT IS GENERATED WHEN
C     THE FLAG IWRITE = 1.
C
C
C DESCRIPTION OF OTHER I/O FILES
C
C FILE (K)      CONTAINS:
C
C      (2)      INPUT FILE:
C               PREVIOUSLY-GENERATED T-MATRICES (BETA ARRAY)
C               AND THE FINAL TWO LANCZOS VECTORS USED ON THAT
C               COMPUTATION.  THIS PROGRAM ALLOWS ENLARGEMENT
C               OF ANY T-MATRICES PROVIDED ON FILE 2.
C
C      (3)      INPUT FILE:
C               THE SINGULAR VALUES FOR WHICH CORRESPONDING
C               SINGULAR VECTORS ARE REQUESTED.  FILE 3 ALSO
C               CONTAINS THE T-MULTIPLICITIES OF THESE SINGULAR
C               VALUES (AS T-EIGENVALUES) AND THEIR COMPUTED GAPS
C               BOTH THE T-MATRICES AND IN THE USER-SUPPLIED MATRIX.
C               THIS FILE IS CREATED IN THE LANCZOS SINGULAR
C               VALUE COMPUTATIONS.
C
C      (4)      INPUT FILE:
C               ERROR ESTIMATES FOR THE ISOLATED SINGULAR VALUES
C               OF FILE 3.  THIS FILE IS CREATED DURING THE LANCZOS
C               SINGULAR VALUE COMPUTATIONS.
C
C      (8)      INPUT FILE:
C               USPEC SUBROUTINE ASSUMES THAT THE USER-
C               SUPPLIED MATRIX IS ON FILE 8.
C
C      (9)      OUTPUT FILE:
C               ERROR ESTIMATES FOR THE COMPUTED RITZ VECTORS CONSIDERED
C               AS EIGENVECTORS OF THE B-MATRIX. THESE ESTIMATES
C               ARE OF THE FORM
C                     BERROR = || B*RITVEC - SVAL*RITVEC ||
C               WHERE B DENOTES THE M+N ORDER SYMMETRIC MATRIX
C               ASSOCIATED WITH THE USER-SUPPLIED MATRIX A, SVAL
C               DENOTES THE SINGULAR VALUE BEING CONSIDERED AND
C               RITVEC DENOTES THE ASSOCIATED COMPUTED RITZ VECTOR.
C
C      (10)     OUTPUT FILE:
C               GUESSES AT APPROPRIATE SIZE T-MATRICES FOR THE
C               T-EIGENVECTORS FOR EACH SUPPLIED SINGULAR VALUE
C               IN THE ARRAY GOODSV(J),  J = 1,...,NGOOD.
C
C      (11)     OUTPUT FILE:
C               COMPUTED T-EIGENVECTORS CORRESPONDING TO SINGULAR
C               VALUES IN THE GOODSV ARRAY.  NOTE THAT IT IS POSSIBLE
C               IN CERTAIN SITUATIONS THAT FOR SOME SINGULAR VALUES
C               SUPPLIED IN THE GOODSV ARRAY A T-EIGENVECTOR WILL
```

```
C           NOT BE COMPUTED.                                            LSV09880
C                                                                       LSV09890
C     (12)    OUTPUT FILE:                                              LSV09900
C             CONTAINS COMPUTED RITZ VECTORS CORRESPONDING TO           LSV09910
C             THE T-EIGENVECTORS ON FILE 11. NOTE THAT IN               LSV09920
C             SOME SITUATIONS THAT FOR SOME SINGULAR VALUES IN          LSV09930
C             THE GOODSV ARRAY FOR WHICH T-EIGENVECTORS HAVE            LSV09940
C             BEEN COMPUTED NO CORRESPONDING RITZ VECTOR WILL           LSV09950
C             HAVE BEEN COMPUTED.                                       LSV09960
C                                                                       LSV09970
C     (13)    OUTPUT FILE:                                              LSV09980
C             ADDITIONAL INFORMATION ABOUT THE BOUNDS AND ERROR         LSV09990
C             ESTIMATES OBTAINED.                                       LSV10000
C                                                                       LSV10010
C                                                                       LSV10020
C-----SEEDS FOR SINGULAR VECTOR PROGRAMS--------------------------------LSV10030
C                                                                       LSV10040
C     SEEDS FOR RANDOM NUMBER GENERATOR GENRAN                          LSV10050
C               (1) SVSEED = INTEGER*4 SCALAR USED IN THE SUBROUTINE    LSV10060
C                            GENRAN TO GENERATE THE STARTING VECTOR FORLSV10070
C                            THE REGENERATION OF THE LANCZOS VECTORS.   LSV10080
C                                                                       LSV10090
C               (2) RHSEED = INTEGER*4 SCALAR USED IN THE SUBROUTINE    LSV10100
C                            GENRAN TO GENERATE A RANDOM VECTOR FOR     LSV10110
C                            USE IN SUBROUTINE INVERM.                  LSV10120
C                                                                       LSV10130
C     USER SHOULD NOTE THAT SVSEED MUST BE THE SAME SEED THAT           LSV10140
C     WAS USED TO GENERATE THE T-MATRICES THAT WERE USED TO             LSV10150
C     COMPUTE THE SINGULAR VALUES WHOSE SINGULAR VECTORS ARE TO BE      LSV10160
C     COMPUTED.  SVSEED IS READ IN FROM FILE 3.                         LSV10170
C                                                                       LSV10180
C                                                                       LSV10190
C-----USER-SPECIFIED PARAMETERS FOR THE SINGULAR VECTOR PROGRAMS--------LSV10200
C                                                                       LSV10210
C                                                                       LSV10220
C     NGOOD   = NUMBER OF SINGULAR VALUES READ INTO THE GOODSV ARRAY    LSV10230
C               READ FROM FILE 3.                                       LSV10240
C                                                                       LSV10250
C     M       = ROW ORDER OF THE USER-SUPPLIED MATRIX.                  LSV10260
C                                                                       LSV10270
C     N       = COLUMN ORDER OF THE USER-SUPPLIED MATRIX.               LSV10280
C                                                                       LSV10290
C     MEV     = SIZE OF THE T-MATRIX THAT WAS USED TO COMPUTE           LSV10300
C               THE SINGULAR VALUES WHOSE SINGULAR VECTORS ARE          LSV10310
C               REQUESTED.  MEV IS READ IN FROM FILE 3.                 LSV10320
C                                                                       LSV10330
C     KMAX    =  SIZE OF THE T-MATRIX PROVIDED ON FILE 2.               LSV10340
C                                                                       LSV10350
C     MDIMTV  = MAXIMUM CUMULATIVE SIZE OF THE TVEC ARRAY ALLOWED       LSV10360
C               FOR ALL OF THE T-EIGENVECTORS REQUIRED.  MDIMTV         LSV10370
C               MUST NOT EXCEED THE USER-SPECIFIED DIMENSION OF         LSV10380
C               THE TVEC ARRAY.  PROGRAM CAN BE RUN WITH THE FLAG       LSV10390
C               MBOUND = 1 TO DETERMINE AN EDUCATED GUESS ON AN         LSV10400
C               APPROPRIATE DIMENSION FOR THE TVEC ARRAY.               LSV10410
C                                                                       LSV10420
```

```
C     MDIMRV = MAXIMUM CUMULATIVE SIZE OF THE RITVEC ARRAY ALLOWED      LSV10430
C              FOR ALL OF THE RITZ VECTORS TO BE COMPUTED. MDIMRV       LSV10440
C              MUST NOT EXCEED THE USER-SPECIFIED DIMENSION OF          LSV10450
C              THE RITVEC ARRAY.  MUST BE SELECTED SO THAT              LSV10460
C              THERE IS ENOUGH ROOM FOR A RITZ VECTOR FOR EVERY         LSV10470
C              GOODEV(J) READ INTO PROGRAM.  (>= NGOOD*(M+N))           LSV10480
C                                                                       LSV10490
C                                                                       LSV10500
C-----ARRAYS REQUIRED BY THE SINGULAR VECTOR PROGRAMS-------------------LSV10510
C                                                                       LSV10520
C                                                                       LSV10530
C     BETA(J) = REAL*8 ARRAY.  ITS DIMENSION MUST BE AT LEAST           LSV10540
C                KMAXN+1,  WHERE KMAXN IS THE LARGEST SIZE T-MATRIX     LSV10550
C                CONSIDERED BY THE PROGRAM.  NOTE THAT KMAXN IS THE     LSV10560
C                LARGER OF THE SIZE OF THE BETA HISTORY PROVIDED        LSV10570
C                ON FILE 2 (IF ANY ) AND THE SIZE WHICH THE PROGRAM     LSV10580
C                SPECIFIES INTERNALLY, THIS LATTER IS ALWAYS            LSV10590
C                < = 11*MEV / 8  +  12, WHERE MEV IS THE SIZE           LSV10600
C                T-MATRIX THAT WAS USED IN THE CORRESPONDING            LSV10610
C                SINGULAR VALUE COMPUTATIONS.  BETA CONTAINS THE        LSV10620
C                NONZERO ENTRIES OF THE LANCZOS T-MATRICES.             LSV10630
C                BETA IS NOT DESTROYED IN THE COMPUTATIONS.             LSV10640
C                THE DIAGONAL ENTRIES OF THE T-MATRICES ARE ALL ZERO.   LSV10650
C                                                                       LSV10660
C     RITVEC(J) = REAL*8 ARRAY. IT DIMENSION MUST BE > = NGOOD*(M+N)    LSV10670
C                 WHERE THE USER-SUPPLIED MATRIX IS MXN                 LSV10680
C                 AND NGOOD IS THE NUMBER OF SINGULAR VALUES WHOSE      LSV10690
C                 SINGULAR VECTORS ARE TO BE COMPUTED.  IT CONTAINS     LSV10700
C                 THE COMPUTED APPROXIMATE SINGULAR VECTORS OF A.       LSV10710
C                 THESE COMPUTED RITZ VECTORS ARE STORED ON FILE 12.    LSV10720
C                                                                       LSV10730
C     TVEC(J)   = REAL*8 ARRAY.  ITS DIMENSION MUST BE AT LEAST         LSV10740
C                 MTOL = |MA(1)| + |MA(2)| + ... + |MA(NGOOD)|          LSV10750
C                 WHERE NGOOD IS THE NUMBER OF SINGULAR VALUES BEING    LSV10760
C                 CONSIDERED AND |MA(J)| IS THE SIZE OF THE             LSV10770
C                 T-MATRIX BEING USED FOR THE B-MATRIX RITZ VECTOR      LSV10780
C                 COMPUTATION FOR GOODSV(J).  THESE SIZES               LSV10790
C                 ARE COMPUTED BY THE PROGRAM.  AN ESTIMATE OF          LSV10800
C                 MTOL CAN BE OBTAINED BY SETTING MBOUND = 1,           LSV10810
C                 RUNNING THE PROGRAM, AND THEN MULTIPLYING THE         LSV10820
C                 RESULTING TOTAL T-SIZE SPECIFIED BY 5/4.  THE TVEC    LSV10830
C                 ARRAY CONTAINS THE COMPUTED T-EIGENVECTORS.  IF       LSV10840
C                 THE FLAG SVTVEC = 1 OR THE FLAG TVSTOP = 1, THEN      LSV10850
C                 THESE VECTORS ARE SAVED ON FILE 11.                   LSV10860
C                                                                       LSV10870
C     V1(J)     = REAL*8 ARRAY.  ITS DIMENSION MUST BE GREATER          LSV10880
C                 THAN THE MAXIMUM OF KMAX AND M, WHERE M IS            LSV10890
C                 THE ROW ORDER OF THE GIVEN MATRIX.  V1 IS USED        LSV10900
C                 IN THE SUBROUTINE INVERM AND IN THE REGENERATION      LSV10910
C                 OF THE LANCZOS VECTORS.                               LSV10920
C                                                                       LSV10930
C     V2(J)     = REAL*8 ARRAY.  ITS DIMENSION MUST BE GREATER          LSV10940
C                 THAN MAX(KMAX,N), WHERE N IS THE COLUMN ORDER OF      LSV10950
C                 THE GIVEN MATRIX.  IT IS USED IN THE REGENERATION     LSV10960
C                 OF THE LANCZOS VECTORS AND IN SUBROUTINE INVERM.      LSV10970
```

```
C
C     GOODSV(J),  = REAL*8 ARRAYS EACH OF DIMENSION AT LEAST NGOOD.
C     SVNEW(J)      CONTAIN THE SINGULAR VALUES FOR WHICH
C                   SINGULAR VECTORS ARE REQUESTED.  SINGULAR VALUES
C                   IN GOODSV ARE READ IN FROM FILE 3.
C
C     BMINGP(J), = REAL*4 ARRAYS OF DIMENSION AT LEAST NGOOD.
C     TMINGP(J)    CONTAIN, RESPECTIVELY, THE MINIMAL GAPS FOR
C                  CORRESPONDING SINGULAR VALUES IN GOODSV ARRAY IN
C                  B-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 MUST BE OF DIMENSION
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,M,N).  USED IN SUBROUTINE GENRAN TO HOLD
C                RANDOM NUMBERS NEEDED FOR THE LANCZOS VECTOR
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 SINGULAR
C             VALUE GOODSV(J) AS AN EIGENVALUE OF THE T-MATRIX.
C             USED TO FLAG SINGULAR VALUES 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 SINGULAR VALUE
C               IN GOODSV 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-----SUBROUTINES INCLUDED FOR THE SINGULAR VECTOR COMPUTATIONS---------
C
C
C     STURMI = FOR EACH GIVEN SINGULAR VALUE GOODSV(J) DETERMINES
C              THE SMALLEST SIZE T-MATRIX FOR WHICH GOODSV(J) IS
C              A T-EIGENVALUE (TO WITHIN A GIVEN TOLERANCE) AND IF
C              POSSIBLE THE SMALLEST SIZE T-MATRIX FOR WHICH
C              IT IS A DOUBLE T-EIGENVALUE (TO WITHIN THE SAME
C              TOLERANCE).  THE SIZE T-MATRIX USED IN THE
C              T-EIGENVECTOR COMPUTATIONS IS THEN DETERMINED BY
```

```
C            STARTING WITH AN INITIAL GUESS BASED ON THE
C            INFORMATION FROM STURMI, AND THEN LOOPING ON THE
C            SIZE OF THE T-EIGENVECTOR COMPUTATIONS.
C
C     LBISEC = RECOMPUTES THE VALUE OF THE GIVEN SINGULAR VALUE
C            AT THE T-SIZE SPECIFIED FOR THE T-EIGENVECTOR
C            COMPUTATION.  LBISEC IS A SIMPLIFICATION OF THE
C            BISEC SUBROUTINE USED IN THE LANCZOS SINGULAR
C            VALUE COMPUTATIONS.
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 SINGULAR VALUES
C            IN THE GOODSV ARRAY.
C
C     LANCZS AND TNORM  SUBROUTINES ARE ALSO USED HERE AS WELL AS
C     IN THE CORRESPONDING SINGULAR VALUE COMPUTATIONS.
C
C
C-----------------------------------------------------------------------
```

6.3 LSVAL: Main Program, Eigenvalue Computations

```
C-----LSVAL  (SINGULAR VALUES OF REAL, RECTANGULAR 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 are used in other scientific or
C  engineering research works the names of the authors of these codes
C  and appropriate references to their written work are to be
C  incorporated in the derivative works.
C
C  This header is not to be removed from these codes.
C
C       REFERENCE: Cullum and Willoughby, Chapter 5
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 MAIN PROGRAM FOR COMPUTING DISTINCT SINGULAR VALUES OF
C    A REAL M X N MATRIX USING LANCZOS TRIDIAGONALIZATION WITHOUT
C    REORTHOGONALIZATION AND WITH SPECIAL STARTING VECTORS.
C
C    FOR A GIVEN REAL MATRIX A OF ORDER M X N THE LANCZOS RECURSION
C    IS APPLIED TO THE ASSOCIATED REAL SYMMETRIC MATRIX B OF ORDER
C    MN = M + N
C
C                                ----                ----
C                               |  0                 A  |
C                   B  =        |                       |
C                               |  A-TRANSPOSE       0  |
C                                ----                ----
C
C    USING SPECIAL STARTING VECTORS.  PLEASE NOTE: ONLY EVEN ORDER
C    LANCZOS TRIDIAGONAL MATRICES AND ONLY NONNEGATIVE SUBINTERVALS
C    ARE PERMISSIBLE.
C
C    PFORT VERIFIER IDENTIFIED THE FOLLOWING NONPORTABLE
C    CONSTRUCTIONS
C
C    1.  DATA/MACHEP/ STATEMENT
C    2.  ALL READ(5,*) STATEMENTS (FREE FORMAT)
C    3.  FORMAT(20A4) USED WITH EXPLANATORY HEADER EXPLAN.
```

```
C     4.  HEXADECIMAL FORMAT (4Z20) USED IN BETA FILES.                 LSV00450
C                                                                       LSV00460
C-----------------------------------------------------------------------LSV00470
      DOUBLE PRECISION  BETA(5001),V1(5000),V2(5000),VS(5000)           LSV00480
      DOUBLE PRECISION  LB(20),UB(20)                                   LSV00490
      DOUBLE PRECISION  BTOL,GAPTOL,TTOL,MACHEP,EPSM,RELTOL             LSV00500
      DOUBLE PRECISION  SCALE1,SCALE2,SCALE3,SCALE4,BISTOL,CONTOL,MULTOLLSV00510
      DOUBLE PRECISION  ONE,ZERO,TEMP,TKMAX,BETAM,BKMIN,T0,T1           LSV00520
      REAL  G(5000),EXPLAN(20)                                          LSV00530
      INTEGER  MP(5000),NMEV(20)                                        LSV00540
      INTEGER  SVSEED,RHSEED,SVSOLD                                     LSV00550
      INTEGER IABS                                                      LSV00560
      REAL  ABS                                                         LSV00570
      DOUBLE PRECISION  DABS, DSQRT, DFLOAT                             LSV00580
      EXTERNAL SVMAT,STRAN                                              LSV00590
C-----------------------------------------------------------------------LSV00600
      DATA MACHEP/Z3410000000000000/                                    LSV00610
      EPSM = 2.0D0*MACHEP                                               LSV00620
C-----------------------------------------------------------------------LSV00630
C                                                                       LSV00640
C     ARRAYS MUST BE DIMENSIONED AS FOLLOWS:                            LSV00650
C     1.  BETA: >= (KMAX+1) WHERE KMAX IS READ IN AND IS                LSV00660
C         THE SIZE OF THE LARGEST T-MATRIX THAT CAN BE CONSIDERED.      LSV00670
C     2.  V1:  >= MAX(M,KMAX+1)                                         LSV00680
C     3.  V2:  >= MAX(N,KMAX)                                           LSV00690
C     4.  VS:  >= KMAX                                                  LSV00700
C     5.  G:  >= MAX(2*KMAX,M,N)                                        LSV00710
C     6.  MP:  >= KMAX                                                  LSV00720
C     7.  LB,UB:  >= NUMBER OF SUBINTERVALS SUPPLIED TO BISEC.          LSV00730
C     8.  NMEV:  >= NUMBER OF T-MATRICES ALLOWED.                       LSV00740
C     9.  EXPLAN:  DIMENSION IS 20.                                     LSV00750
C                                                                       LSV00760
C                                                                       LSV00770
C     IMPORTANT TOLERANCES OR SCALES THAT ARE USED REPEATEDLY           LSV00780
C     THROUGHOUT THIS PROGRAM ARE THE FOLLOWING:                        LSV00790
C     SCALED MACHINE EPSILON:  TTOL = TKMAX*EPSM WHERE                  LSV00800
C     EPSM = 2*MACHINE EPSILON AND                                      LSV00810
C     TKMAX = MAX(BETA(J), J = 1,MEV)                                   LSV00820
C     BISEC CONVERGENCE TOLERANCE:  BISTOL = DSQRT(1000+MEV)*TTOL       LSV00830
C     BISEC MULTIPLICITY TOLERANCE:  MULTOL = (1000+MEV)*TTOL           LSV00840
C     LANCZOS CONVERGENCE TOLERANCE:   CONTOL = BETA(MEV+1)*1.D-10      LSV00850
C-----------------------------------------------------------------------LSV00860
C     OUTPUT HEADER                                                     LSV00870
      WRITE(6,10)                                                       LSV00880
   10 FORMAT(/' LANCZOS PROCEDURE FOR REAL, RECTANGULAR MATRICES'/)     LSV00890
C                                                                       LSV00900
C     SET PROGRAM PARAMETERS                                            LSV00910
C     SCALEK ARE USED IN TOLERANCES NEEDED IN SUBROUTINES LUMP,         LSV00920
C     ISOEV AND PRTEST.  USER MUST NOT MODIFY THESE SCALES.             LSV00930
      SCALE1 = 5.0D2                                                    LSV00940
      SCALE2 = 5.0D0                                                    LSV00950
      SCALE3 = 5.0D0                                                    LSV00960
      SCALE4 = 1.0D4                                                    LSV00970
      ONE  = 1.0D0                                                      LSV00980
      ZERO = 0.0D0                                                      LSV00990
```

```
      BTOL = 1.0D-8                                                     LSV01000
C     BTOL = EPSM                                                       LSV01010
      GAPTOL = 1.0D-8                                                   LSV01020
      ICONV = 0                                                         LSV01030
      MOLD = 0                                                          LSV01040
      MOLD1 = 1                                                         LSV01050
      ICT = 0                                                           LSV01060
      MMB = 0                                                           LSV01070
      IPROJ = 0                                                         LSV01080
C                                                                       LSV01090
C     READ USER-SPECIFIED PARAMETERS FROM INPUT FILE 5 (FREE FORMAT)    LSV01100
C                                                                       LSV01110
C     READ USER-PROVIDED HEADERS FOR RUN                                LSV01120
      READ(5,20) EXPLAN                                                 LSV01130
      WRITE(6,20) EXPLAN                                                LSV01140
      READ(5,20) EXPLAN                                                 LSV01150
      WRITE(6,20) EXPLAN                                                LSV01160
   20 FORMAT(20A4)                                                      LSV01170
C                                                                       LSV01180
C     READ THE ROW ORDER M OF THE MATRIX AND THE COLUMN ORDER N.        LSV01190
C     READ THE MAXIMUM ORDER OF THE T-MATRICES ALLOWED (KMAX),          LSV01200
C     THE NUMBER OF T-MATRICES ALLOWED (NMEVS), AND A                   LSV01210
C     MATRIX IDENTIFICATION NUMBER (MATNO).                             LSV01220
      READ(5,20) EXPLAN                                                 LSV01230
      READ(5,*) M,N,KMAX,NMEVS,MATNO                                    LSV01240
      NM = M + N                                                        LSV01250
C                                                                       LSV01260
C     READ SEEDS FOR LANCZS AND INVERR SUBROUTINES (SVSEED AND RHSEED)  LSV01270
C     READ MAXIMUM NUMBER OF ITERATIONS ALLOWED FOR EACH INVERSE         LSV01280
C     ITERATION (MXINIT) AND MAXIMUM NUMBER OF STURM SEQUENCES          LSV01290
C     ALLOWED (MXSTUR)                                                  LSV01300
      READ(5,20) EXPLAN                                                 LSV01310
      READ(5,*) SVSEED,RHSEED,MXINIT,MXSTUR                             LSV01320
C                                                                       LSV01330
C     ISTART = (0,1):  ISTART = 0 MEANS BETA FILE IS NOT                LSV01340
C     AVAILABLE.  ISTART = 1 MEANS BETA FILE IS AVAILABLE ON            LSV01350
C     FILE 2.                                                           LSV01360
C     ISTOP = (0,1):  ISTOP = 0 MEANS PROCEDURE GENERATES BETA          LSV01370
C     FILE AND THEN TERMINATES.  ISTOP = 1 MEANS PROCEDURE GENERATES    LSV01380
C     BETAS IF NEEDED AND THEN COMPUTES SINGULAR VALUES AND             LSV01390
C     ERROR ESTIMATES AND THEN TERMINATES.                              LSV01400
      READ(5,20) EXPLAN                                                 LSV01410
      READ(5,*) ISTART,ISTOP                                            LSV01420
C                                                                       LSV01430
C     IHIS = (0,1):  IHIS = 0 MEANS BETA FILE IS NOT WRITTEN            LSV01440
C     TO FILE 1.  IHIS = 1 MEANS BETA FILE IS WRITTEN TO FILE 1.        LSV01450
C     IDIST = (0,1):  IDIST = 0 MEANS DISTINCT T-EIGENVALUES            LSV01460
C     ARE NOT WRITTEN TO FILE 11.  IDIST = 1 MEANS DISTINCT             LSV01470
C     T-EIGENVALUES ARE WRITTEN TO FILE 11.                             LSV01480
C     IWRITE = (0,1):  IWRITE = 0 MEANS NO INTERMEDIATE OUTPUT          LSV01490
C     FROM THE COMPUTATIONS IS WRITTEN TO FILE 6.  IWRITE = 1 MEANS     LSV01500
C     T-EIGENVALUES AND ERROR ESTIMATES ARE WRITTEN TO FILE 6           LSV01510
C     AS THEY ARE COMPUTED.  SPECIFY THE PARITY (IPAR) OF THE           LSV01520
C     LANCZOS STARTING VECTOR.  IF M > N, THEN IPAR = 1,                LSV01530
C     IF M < N, THEN IPAR = 2.                                          LSV01540
```

```
      READ(5,20) EXPLAN                                                 LSV01550
      READ(5,*) IHIS,IDIST,IWRITE,IPAR                                  LSV01560
      IF(M.GT.N) IPAR = 1                                               LSV01570
      IF(M.LT.N) IPAR = 2                                               LSV01580
      IPARO = IPAR                                                      LSV01590
C                                                                       LSV01600
C     READ IN THE RELATIVE TOLERANCE (RELTOL) FOR USE IN THE            LSV01610
C     SPURIOUS, T-MULTIPLICITY, AND PRTEST TESTS.                       LSV01620
      READ(5,20) EXPLAN                                                 LSV01630
      READ(5,*) RELTOL                                                  LSV01640
C                                                                       LSV01650
C     READ IN THE SIZES OF THE T-MATRICES TO BE CONSIDERED.             LSV01660
C     NOTE THAT ONLY EVEN ORDER T-SIZES ARE PERMISSIBLE.                LSV01670
      READ(5,20) EXPLAN                                                 LSV01680
      READ(5,*) (NMEV(J), J=1,NMEVS)                                    LSV01690
C                                                                       LSV01700
C     CHECK TO SEE THAT ALL T-SIZES PROVIDED ARE EVEN ORDERED.          LSV01710
C     TERMINATE IF THAT IS NOT THE CASE.                                LSV01720
      DO 30 I = 1,NMEVS                                                 LSV01730
      NMEV2 = NMEV(I)/2                                                 LSV01740
      IF(2*NMEV2.NE.NMEV(I)) GO TO 670                                  LSV01750
   30 CONTINUE                                                          LSV01760
C                                                                       LSV01770
C     READ IN THE NUMBER OF SUBINTERVALS TO BE CONSIDERED.              LSV01780
      READ(5,20) EXPLAN                                                 LSV01790
      READ(5,*) NINT                                                    LSV01800
C                                                                       LSV01810
C     READ IN THE LEFT-END POINTS OF THE SUBINTERVALS TO BE CONSIDERED. LSV01820
C     THESE MUST BE IN ALGEBRAICALLY INCREASING ORDER                   LSV01830
      READ(5,20) EXPLAN                                                 LSV01840
      READ(5,*) (LB(J), J=1,NINT)                                       LSV01850
C                                                                       LSV01860
C     READ IN THE RIGHT-END POINTS OF THE SUBINTERVALS TO BE CONSIDERED.LSV01870
C     THESE MUST BE IN ALGEBRAICALLY INCREASING ORDER                   LSV01880
      READ(5,20) EXPLAN                                                 LSV01890
      READ(5,*) (UB(J), J=1,NINT)                                       LSV01900
C                                                                       LSV01910
C-----------------------------------------------------------------------LSV01920
C     INITIALIZE THE ARRAYS FOR THE USER-SPECIFIED MATRIX               LSV01930
C     AND PASS THE STORAGE LOCATIONS OF THESE ARRAYS TO THE             LSV01940
C     MATRIX-VECTOR MULTIPLY SUBROUTINES SVMAT AND STRAN.               LSV01950
C                                                                       LSV01960
      CALL USPEC(M,N,MATNO)                                             LSV01970
C                                                                       LSV01980
C-----------------------------------------------------------------------LSV01990
C     MASK UNDERFLOW AND OVERFLOW                                       LSV02000
C                                                                       LSV02010
      CALL MASK                                                         LSV02020
C                                                                       LSV02030
C-----------------------------------------------------------------------LSV02040
C                                                                       LSV02050
C     WRITE TO FILE 6, A SUMMARY OF THE PARAMETERS FOR THIS RUN         LSV02060
C                                                                       LSV02070
      WRITE(6,40) MATNO,M,N,KMAX                                        LSV02080
   40 FORMAT(/3X,'MATRIX ID',5X,'M',5X,'N',4X,'MAX ORDER OF T'/         LSV02090
```

```
     1 I12,2I6,I18/)
C
      WRITE(6,50) ISTART,ISTOP
   50 FORMAT(/2X,'ISTART',3X,'ISTOP'/2I8/)
C
      WRITE(6,60) IHIS,IDIST,IWRITE,IPAR
   60 FORMAT(/4X,'IHIS',3X,'IDIST',2X,'IWRITE',4X,'IPAR'/4I8/)
C
      WRITE(6,70) SVSEED,RHSEED
   70 FORMAT(/' SEEDS FOR RANDOM NUMBER GENERATOR'//
     1 4X,'LANCZS SEED',4X,'INVERR SEED'/2I15/)
C
      WRITE(6,80) (NMEV(J), J=1,NMEVS)
   80 FORMAT(/' SIZES OF T-MATRICES TO BE CONSIDERED'/(6I12))
C
      WRITE(6,90) RELTOL,GAPTOL,BTOL
   90 FORMAT(/' RELATIVE TOLERANCE USED TO COMBINE COMPUTED T-EIGENVALUE
     1S'/E15.3/' RELATIVE GAP TOLERANCES USED IN INVERSE ITERATION'/
     1E15.3/' RELATIVE TOLERANCE FOR CHECK ON SIZE OF BETAS'/E15.3/)
C
      WRITE(6,100) (J,LB(J),UB(J), J=1,NINT)
  100 FORMAT(/' BISEC WILL BE USED ON THE FOLLOWING INTERVALS'/
     1 (I6,2E20.6)/)
C
      IF (ISTART.EQ.0.AND.IPAR.EQ.1) WRITE(6,110)
      IF (ISTART.EQ.0.AND.IPAR.EQ.2) WRITE(6,120)
  110 FORMAT(/' STARTING VECTOR IS OF FORM (0,V2)'/)
  120 FORMAT(/' STARTING VECTOR IS OF FORM (V1,0)'/)
C
      IF (ISTART.EQ.0) GO TO 170
C
C     READ IN BETA HISTORY FROM FILE 2
C
      READ(2,130)MOLD,MO,NO,IPARO,IPAR,SVSOLD,MATOLD
  130 FORMAT(3I6,2I3,I12,I8)
C
      IF (KMAX.LT.MOLD) KMAX = MOLD
      KMAX1 = KMAX + 1
C
C     CHECK THAT M, N, MATRIX ID MATNO, AND RANDOM SEED SVSEED
C     AGREE WITH THOSE IN THE HISTORY FILE.  IF NOT PROCEDURE STOPS.
C
      ITEMP = (MO-M)**2+(NO-N)**2+(MATNO-MATOLD)**2+(SVSEED-SVSOLD)**2
C
      IF (ITEMP.EQ.0) GO TO 150
C
      WRITE(6,140)
  140 FORMAT(' PROGRAM TERMINATES'/    ' READ FROM FILE 2 CORRESPONDS TO
     1 DIFFERENT MATRIX THAN MATRIX SPECIFIED'/)
      GO TO 690
C
  150 CONTINUE
      MOLD1 = MOLD+1
C
      READ(2,160)(BETA(J), J=1,MOLD1)
```

```
  160 FORMAT(4Z20)                                                      LSV02650
C                                                                       LSV02660
      IF (KMAX.EQ.MOLD) GO TO 190                                       LSV02670
C                                                                       LSV02680
      READ(2,160)(V1(J), J=1,M)                                         LSV02690
      READ(2,160)(V2(J), J=1,N)                                         LSV02700
C                                                                       LSV02710
  170 CONTINUE                                                          LSV02720
      IIX = SVSEED                                                      LSV02730
C                                                                       LSV02740
C-----------------------------------------------------------------------LSV02750
C                                                                       LSV02760
      CALL LANCZS(SVMAT,STRAN,BETA,V1,V2,G,KMAX,MOLD1,M,N,IPAR,IIX)     LSV02770
C                                                                       LSV02780
C-----------------------------------------------------------------------LSV02790
C                                                                       LSV02800
      KMAX1 = KMAX + 1                                                  LSV02810
C                                                                       LSV02820
      IF (IHIS.EQ.0.AND.ISTOP.GT.0) GO TO 190                           LSV02830
C                                                                       LSV02840
      WRITE(1,180) KMAX,M,N,IPARO,IPAR,SVSEED,MATNO                     LSV02850
  180 FORMAT(3I6,2I3,I12,I8,' = KMAX,M,N,IPARO,IPAR,SVSEED,MATNO')      LSV02860
C                                                                       LSV02870
      WRITE(1,160)(BETA(I), I=1,KMAX1)                                  LSV02880
C                                                                       LSV02890
      WRITE(1,160)(V1(I), I=1,M)                                        LSV02900
      WRITE(1,160)(V2(I), I=1,N)                                        LSV02910
C                                                                       LSV02920
      IF (ISTOP.EQ.0) GO TO 570                                         LSV02930
C                                                                       LSV02940
  190 CONTINUE                                                          LSV02950
      BKMIN = BTOL                                                      LSV02960
      WRITE(6,200)                                                      LSV02970
  200 FORMAT(/' T-MATRICES (BETA) ARE NOW AVAILABLE'/)                  LSV02980
C                                                                       LSV02990
C-----------------------------------------------------------------------LSV03000
C     SUBROUTINE TNORM CHECKS MIN(BETA)/(ESTIMATED NORM(A)) > BTOL .    LSV03010
C     IF THIS IS VIOLATED IB IS SET EQUAL TO THE NEGATIVE OF THE INDEX  LSV03020
C     OF THE MINIMAL BETA.  IF(IB < 0) THEN SUBROUTINE TNORM IS         LSV03030
C     CALLED FOR EACH VALUE OF MEV TO DETERMINE WHETHER OR NOT THERE    LSV03040
C     IS A BETA IN THE T-MATRIX SPECIFIED THAT VIOLATES THIS TEST.      LSV03050
C     IF THERE IS SUCH A BETA THE PROGRAM TERMINATES FOR THE USER       LSV03060
C     TO DECIDE WHAT TO DO.  THIS TEST CAN BE OVER-RIDDEN BY            LSV03070
C     SIMPLY MAKING BTOL SMALLER, BUT THEN THERE IS THE POSSIBILITY     LSV03080
C     THAT LOSSES IN THE LOCAL ORTHOGONALITY MAY HURT THE COMPUTATIONS. LSV03090
C     BTOL = 1.D-8 IS HOWEVER A CONSERVATIVE CHOICE FOR BTOL.           LSV03100
C                                                                       LSV03110
C     TNORM ALSO COMPUTES TKMAX = MAX(BETA(K), K=1,KMAX).               LSV03120
C     TKMAX IS USED TO SCALE THE TOLERANCES USED IN THE                 LSV03130
C     T-MULTIPLICITY AND SPURIOUS TESTS IN BISEC. TKMAX IS ALSO USED IN LSV03140
C     THE PROJECTION TEST FOR HIDDEN T-EIGENVALUES THAT HAD 'TOO SMALL' LSV03150
C     A PROJECTION ON THE STARTING VECTOR.                              LSV03160
C                                                                       LSV03170
      CALL TNORM(BETA,BKMIN,TKMAX,KMAX,IB)                              LSV03180
C                                                                       LSV03190
```

```
C-----------------------------------------------------------------------
C
      TTOL = EPSM*TKMAX
C
C     LOOP ON THE SIZE OF THE T-MATRIX
C
  210 CONTINUE
      MMB = MMB + 1
C     NOTE THAT ONLY EVEN ORDER T-SIZES ARE PERMISSIBLE.
      MEV = NMEV(MMB)
C     IS MEV TOO LARGE ?
      IF(MEV.LE.KMAX) GO TO 230
      WRITE(6,220) MMB, MEV, KMAX
  220 FORMAT(/' TERMINATE PRIOR TO CONSIDERING THE',I6,'TH T-MATRIX'/
     1' BECAUSE THE SIZE REQUESTED',I6,' IS GREATER THAN THE MAXIMUM SIZ
     1E ALLOWED',I6/)
      GO TO 570
C
  230 MP1 = MEV + 1
      BETAM = BETA(MP1)
C
      IF (IB.GE.0) GO TO 240
C
      T0 = BTOL
C
C-----------------------------------------------------------------------
C
      CALL TNORM(BETA,T0,T1,MEV,IBMEV)
C
C-----------------------------------------------------------------------
C
      TEMP = T0/TKMAX
      IBMEV = IABS(IBMEV)
      IF (TEMP.GE.BTOL) GO TO 240
      IBMEV = -IBMEV
      GO TO 630
C
  240 CONTINUE
      IC = MXSTUR-ICT
C
C-----------------------------------------------------------------------
C     BISEC LOOP. THE SUBROUTINE BISEC INCORPORATES DIRECTLY THE
C     T-MULTIPLICITY AND SPURIOUS TESTS. T-EIGENVALUES WILL BE
C     CALCULATED BY BISEC SEQUENTIALLY ON INTERVALS
C     (LB(J),UB(J)), J = 1,NINT).
C
C     ON RETURN FROM BISEC
C     NDIS = NUMBER OF DISTINCT EIGENVALUES OF T(1,MEV) ON UNION
C            OF THE (LB,UB) INTERVALS
C     VS = DISTINCT T-EIGENVALUES IN ALGEBRAICALLY INCREASING ORDER
C     MP = T-MULTIPLICITIES OF THE T-EIGENVALUES STORED IN VS
C     MP(I) = (0,1,MI), MI>1, I=1,NDIS  MEANS:
C        (0)  VS(I) IS SPURIOUS
C        (1)  VS(I) IS T-SIMPLE AND GOOD
C        (MI) VS(I) IS T-MULTIPLE AND IS THEREFORE NOT ONLY GOOD BUT
```

```
C             ALSO A CONVERGED GOOD T-EIGENVALUE.
C
C
      CALL BISEC(BETA,V1,V2,VS,LB,UB,EPSM,TTOL,MP,NINT,
     1 MEV,NDIS,IC,IWRITE)
C
C-----------------------------------------------------------------------
C
      IF (NDIS.EQ.0) GO TO 650
C
C     COMPUTE THE TOTAL NUMBER OF STURM SEQUENCES USED TO DATE
C     COMPUTE THE BISEC CONVERGENCE AND T-MULTIPLICITY TOLERANCES USED.
C     COMPUTE THE CONVERGENCE TOLERANCE FOR T-EIGENVALUES.
      ICT = ICT + IC
      TEMP = DFLOAT(MEV+1000)
      MULTOL = TEMP*TTOL
      TEMP = DSQRT(TEMP)
      BISTOL = TTOL*TEMP
      CONTOL = BETAM*1.D-10
C
C-----------------------------------------------------------------------
C    SUBROUTINE LUMP 'COMBINES' T-EIGENVALUES THAT ARE 'TOO CLOSE'.
C    NOTE HOWEVER THAT CLOSE SPURIOUS T-EIGENVALUES ARE NOT AVERAGED
C    WITH GOOD ONES. HOWEVER, THEY MAY BE USED TO INCREASE THE
C    T-MULTIPLICITY OF A GOOD T-EIGENVALUE.
C
      LOOP = NDIS
      CALL LUMP(VS,RELTOL,MULTOL,SCALE2,MP,LOOP)
C
C-----------------------------------------------------------------------
C
      IF(NDIS.EQ.LOOP) GO TO 260
C
      WRITE(6,250) NDIS, MEV, LOOP
  250 FORMAT(/I6,' DISTINCT T-EIGENVALUES WERE COMPUTED IN BISEC AT MEV
    1=',I6/ 2X,' LUMP SUBROUTINE REDUCES NUMBER OF DISTINCT T-EIGENVALU
    1ES TO',I6)
C
  260 CONTINUE
      NDIS = LOOP
      BETA(MP1) = BETAM
C
C-----------------------------------------------------------------------
C     THE SUBROUTINE ISOEV LABELS THOSE SIMPLE T-EIGENVALUES OF T(1,MEV)
C     WITH VERY SMALL GAPS BETWEEN NEIGHBORING T-EIGENVALUES OF T(1,MEV)
C     TO AVOID COMPUTING ERROR ESTIMATES FOR ANY SIMPLE GOOD
C     T-EIGENVALUE THAT IS TOO CLOSE TO A SPURIOUS T-EIGENVALUE.
C     ON RETURN FROM ISOEV, G CONTAINS CODED MINIMAL GAPS
C     BETWEEN THE DISTINCT EIGENVALUES OF T(1,MEV). (G IS REAL).
C     G(I) < 0 MEANS MINGAP IS DUE TO LEFT GAP G(I) > 0 MEANS DUE TO
C     RIGHT GAP. MP(I) = -1 MEANS THAT THE GOOD T-EIGENVALUE IS SIMPLE
C     AND HAS A VERY SMALL MINGAP IN T(1,MEV) DUE TO A SPURIOUS
C     T-EIGENVALUE.
C     NG = NUMBER OF GOOD T-EIGENVALUES.
C     NISO = NUMBER OF ISOLATED, GOOD T-EIGENVALUES.
```

```
C                                                                       LSV04300
      CALL ISOEV(VS,GAPTOL,MULTOL,SCALE1,G,MP,NDIS,NG,NISO)             LSV04310
C                                                                       LSV04320
C-----------------------------------------------------------------------LSV04330
C                                                                       LSV04340
      WRITE(6,270)NG,NISO,NDIS                                          LSV04350
  270 FORMAT(/I6,' SINGULAR VALUES HAVE BEEN COMPUTED'/                 LSV04360
     1 I6,' OF THESE ARE ISOLATED'/                                     LSV04370
     2 I6,' = NUMBER OF DISTINCT T-EIGENVALUES COMPUTED'/)              LSV04380
C                                                                       LSV04390
C     DO WE WRITE DISTINCT T-EIGENVALUES TO FILE 11?                    LSV04400
      IF (IDIST.EQ.0) GO TO 310                                         LSV04410
C                                                                       LSV04420
      WRITE(11,280) NDIS,NISO,MEV,M,N,SVSEED,MATNO                      LSV04430
  280 FORMAT(5I5,I12,I8,' = NDIS,NISO,MEV,M,N,SVSEED,MATNO'/)           LSV04440
C                                                                       LSV04450
      WRITE(11,290) (MP(I),VS(I),G(I), I=1,NDIS)                        LSV04460
  290 FORMAT(2(I3,E25.16,E12.3))                                        LSV04470
C                                                                       LSV04480
      WRITE(11,300) NDIS, (MP(I), I=1,NDIS)                             LSV04490
  300 FORMAT(/I6,' = NDIS, T-MULTIPLICITIES (0 MEANS  SPURIOUS)'/(20I4))LSV04500
C                                                                       LSV04510
  310 CONTINUE                                                          LSV04520
C                                                                       LSV04530
      IF (NISO.NE.0) GO TO 340                                          LSV04540
C                                                                       LSV04550
      WRITE(4,320) MEV                                                  LSV04560
  320 FORMAT(/' AT MEV = ',I6,' THERE ARE NO ISOLATED T-EIGENVALUES'/   LSV04570
     1' SO NO ERROR ESTIMATES WERE COMPUTED/')                          LSV04580
C                                                                       LSV04590
      WRITE(6,330)                                                      LSV04600
  330 FORMAT(/' ALL COMPUTED SINGULAR VALUES ARE T-MULTIPLE'/           LSV04610
     1 ' THEREFORE ALL COMPUTED SINGULAR VALUES ARE ASSUMED TO HAVE CONVLSV04620
     1ERGED'/)                                                          LSV04630
C                                                                       LSV04640
      ICONV = 1                                                         LSV04650
      GO TO 380                                                         LSV04660
C                                                                       LSV04670
  340 CONTINUE                                                          LSV04680
C                                                                       LSV04690
C-----------------------------------------------------------------------LSV04700
C     SUBROUTINE INVERR COMPUTES ERROR ESTIMATES FOR ISOLATED GOOD      LSV04710
C     T-EIGENVALUES USING INVERSE ITERATION ON T(1,MEV). ON RETURN      LSV04720
C     G(J) = MINIMUM GAP IN T(1,MEV) FOR EACH VS(J), J=1,NDIS           LSV04730
C     G(MEV+I) = BETAM*|U(MEV)| = ERROR ESTIMATE FOR ISOLATED GOOD      LSV04740
C              T-EIGENVALUES, WHERE I = 1, NISO AND  BETAM = BETA(MEV+1)LSV04750
C              U(MEV) IS MEVTH COMPONENT OF THE UNIT EIGENVECTOR OF T   LSV04760
C              CORRESPONDING TO THE ITH ISOLATED GOOD T-EIGENVALUE.     LSV04770
C     A NEGATIVE ERROR ESTIMATE MEANS THAT FOR THAT PARTICULAR          LSV04780
C     T-EIGENVALUE THE INVERSE ITERATION DID NOT CONVERGE IN <= MXINIT  LSV04790
C     STEPS AND THAT THE CORRESPONDING ERROR ESTIMATE IS QUESTIONABLE.  LSV04800
C                                                                       LSV04810
C     V2 CONTAINS THE ISOLATED GOOD T-EIGENVALUES                       LSV04820
C     V1 CONTAINS THE MINGAPS TO THE NEAREST DISTINCT  EIGENVALUE       LSV04830
C        OF T(1,MEV) FOR EACH ISOLATED GOOD T-EIGENVALUE IN V2.         LSV04840
```

```
C     VS CONTAINS THE NDIS DISTINCT EIGENVALUES OF T(1,MEV)
C     MP CONTAINS THE CORRESPONDING CODED T-MULTIPLICITIES
C
      IT = MXINIT
      CALL INVERR(BETA,V1,V2,VS,EPSM,G,MP,MEV,MMB,NDIS,NISO,NM,
     1 RHSEED,IT,IWRITE)
C
C-----------------------------------------------------------------------
C
C     SIMPLE CHECK FOR CONVERGENCE. CHECKS TO SEE IF ALL OF THE ERROR
C     ESTIMATES ARE SMALLER THAN CONTOL.
C     IF THIS TEST IS SATISFIED, THEN CONVERGENCE FLAG, ICONV IS SET
C     TO 1.  TYPICALLY ERROR ESTIMATES ARE VERY CONSERVATIVE.
C
      WRITE(6,350) CONTOL
  350 FORMAT(/' CONVERGENCE IS TESTED USING THE CONVERGENCE TOLERANCE',
     1E13.4/)
C
      II = MEV +1
      IF = MEV+NISO
      DO 360 I = II,IF
      IF (ABS(G(I)).GT.CONTOL) GO TO 380
  360 CONTINUE
      ICONV = 1
      MMB = NMEVS
C
      WRITE(6,370) CONTOL
  370 FORMAT(' ALL COMPUTED ERROR ESTIMATES WERE LESS THAN',E15.4/
     1 ' THEREFORE PROCEDURE TERMINATES'/)
C
  380 CONTINUE
C
C     IF CONVERGENCE IS INDICATED, THAT IS ICONV = 1 ,THEN
C     THE SUBROUTINE PRTEST IS CALLED TO CHECK FOR ANY CONVERGED
C     T-EIGENVALUES THAT HAVE BEEN MISLABELLED AS SPURIOUS BECAUSE
C     THE PROJECTION OF THEIR SINGULAR VECTOR ON THE STARTING
C     VECTOR WAS TOO SMALL.  NUMERICAL TESTS INDICATE THAT
C     SUCH SINGULAR VALUES ARE RARE.  THEREFORE, IF MANY OF
C     THESE HIDDEN SINGULAR VALUES APPEAR ON SOME RUN, THE USER
C     CAN BE CERTAIN THAT SOMETHING IS FOULED UP.
C
      IF (ICONV.EQ.0) GO TO 510
C
C-----------------------------------------------------------------------
C
      CALL PRTEST (BETA,VS,TKMAX,EPSM,RELTOL,SCALE3,SCALE4,
     1 MP,NDIS,MEV,IPROJ)
C
C-----------------------------------------------------------------------
C
      IF(IPROJ.EQ.0) GO TO 500
C
      IF(IDIST.EQ.1)  WRITE(11,390) IPROJ
  390 FORMAT(' SUBROUTINE PRTEST WANTS TO RELABEL',I6,' SPURIOUS T-EIGEN
     1VALUES'/' WE ACCEPT RELABELLING ONLY IF LAST COMPONENT OF T-EIGENV
```

```
     1ECTOR IS L.T. 1.D-10')/)
C
      IIX = RHSEED
C
C-----------------------------------------------------------------------
C
      CALL GENRAN(IIX,G,MEV)
C
C-----------------------------------------------------------------------
C
      ITEN = -10
      NISOM = NISO + MEV
      IWRITO = IWRITE
      IWRITE = 0
C
      DO 420 J = 1,NDIS
      IF(MP(J).NE.ITEN) GO TO 420
      T0 = VS(J)
C
C-----------------------------------------------------------------------
C
      IT = MXINIT
      CALL INVERM(BETA,V1,V2,T0,TEMP,T1,EPSM,G,MEV,IT,IWRITE)
C
C-----------------------------------------------------------------------
C
      IF(TEMP.LE.1.D-10) GO TO 410
C     ERROR ESTIMATE WAS NOT SMALL REJECT RELABELLING OF THIS
C     T-EIGENVALUE.
      IF(IDIST.EQ.1) WRITE(11,400) J,T0,TEMP
  400 FORMAT(/' LAST COMPONENT FOR',I6,'TH T-EIGENVALUE',E20.12/' IS TOO
     1 LARGE = ',E15.6,' SO DO NOT ACCEPT PRTEST RELABELLING'/)
      MP(J) = 0
      IPROJ = IPROJ - 1
      GO TO 420
C     RELABELLING ACCEPTED
  410 NISOM = NISOM + 1
      G(NISOM) = BETAM*TEMP
  420 CONTINUE
      IWRITE = IWRITO
C
      IF(IPROJ.EQ.0) GO TO 460
      WRITE(6,430) IPROJ
  430 FORMAT(/I6,' T-EIGENVALUES WERE RECLASSIFIED AS GOOD.'/
     1' THESE ARE IDENTIFIED IN FILE 3 BY A T-MULTIPLICITY OF -10'/' USE
     2R SHOULD INSPECT EACH TO MAKE SURE NEIGHBORS HAVE CONVERGED'/)
C
      IF(IDIST.EQ.1)  WRITE(11,440) IPROJ
  440 FORMAT(/I6,' T-EIGENVALUES WERE RELABELLED AS GOOD'/
     1' BELOW IS CORRECTED T-MULTIPLICITY PATTERN'/)
C
      WRITE(6,450) NDIS, (MP(I), I=1,NDIS)
      IF(IDIST.EQ.1) WRITE(11,450) NDIS, (MP(I), I=1,NDIS)
  450 FORMAT(/I6,' = NDIS, T-MULTIPLICITIES (0 MEANS  SPURIOUS)'/
     1 6X, ' (-10) MEANS SPURIOUS T-EIGENVALUE RELABELLED AS GOOD'/(20I4
```

```
     1))                                                                LSV05950
C                                                                       LSV05960
C     RECALCULATE MINGAPS FOR DISTINCT T(1,MEV) EIGENVALUES.            LSV05970
  460 NDIS1 = NDIS - 1                                                  LSV05980
      G(NDIS) = VS(NDIS1)-VS(NDIS)                                      LSV05990
      G(1) = VS(2)-VS(1)                                                LSV06000
C                                                                       LSV06010
      DO 470 J = 2,NDIS1                                                LSV06020
      T0 = VS(J)-VS(J-1)                                                LSV06030
      T1 = VS(J+1)-VS(J)                                                LSV06040
      G(J) = T1                                                         LSV06050
      IF (T0.LT.T1) G(J) = -T0                                          LSV06060
  470 CONTINUE                                                          LSV06070
      IF(IPROJ.EQ.0) GO TO 500                                          LSV06080
C     WRITE TO FILE 4 ERROR ESTIMATES FOR THOSE T-EIGENVALUES RELABELLEDLSV06090
      NGOOD = 0                                                         LSV06100
      DO 480 J = 1,NDIS                                                 LSV06110
      IF(MP(J).EQ.0) GO TO 480                                          LSV06120
      NGOOD = NGOOD + 1                                                 LSV06130
      IF(MP(J).NE.ITEN) GO TO 480                                       LSV06140
      T0 = VS(J)                                                        LSV06150
      NISO = NISO + 1                                                   LSV06160
      NISOM = MEV + NISO                                                LSV06170
      WRITE(4,490) NGOOD,T0,G(NISOM),G(J)                               LSV06180
  480 CONTINUE                                                          LSV06190
  490 FORMAT(I10,E25.16,2E14.3)                                         LSV06200
C                                                                       LSV06210
  500 CONTINUE                                                          LSV06220
C                                                                       LSV06230
C     WRITE THE COMPUTED SINGULAR VALUES TO FILE 3.  FIRST TRANSFER THEMLSV06240
C     TO V2 AND THEIR T-MULTIPLICITIES TO THE CORRESPONDING POSITIONS   LSV06250
C     IN MP AND COMPUTE THE B-MINGAPS, THE MINIMAL GAPS BETWEEN THE     LSV06260
C     SINGULAR VALUES CONSIDERED AS EIGENVALUES OF THE B-MATRIX.        LSV06270
C     THESE GAPS WILL BE PUT IN THE ARRAY G.                            LSV06280
C     SINCE G CURRENTLY CONTAINS THE MINIMAL GAPS BETWEEN THE DISTINCT  LSV06290
C     EIGENVALUES OF THE T-MATRIX, THESE GAPS WILL FIRST BE             LSV06300
C     TRANSFERRED TO V1.  NOTE THAT V1<0 MEANS THAT THAT MINIMAL GAP    LSV06310
C     IN THE T-MATRIX IS DUE TO A SPURIOUS T-EIGENVALUE.                LSV06320
C     ALL THIS INFORMATION IS PRINTED TO FILE 3                         LSV06330
C                                                                       LSV06340
  510 CONTINUE                                                          LSV06350
C                                                                       LSV06360
      NG = 0                                                            LSV06370
      DO 520 I = 1,NDIS                                                 LSV06380
      IF (MP(I).EQ.0) GO TO 520                                         LSV06390
      NG = NG+1                                                         LSV06400
      MP(NG) = MP(I)                                                    LSV06410
      V2(NG) = VS(I)                                                    LSV06420
      TEMP = G(I)                                                       LSV06430
      TEMP = DABS(TEMP)                                                 LSV06440
      J = I+1                                                           LSV06450
      IF (G(I).LT.ZERO) J = I-1                                         LSV06460
      IF (MP(J).EQ.0) TEMP = -TEMP                                      LSV06470
      V1(NG) = TEMP                                                     LSV06480
  520 CONTINUE                                                          LSV06490
```

```
C
      WRITE(6,530)MEV
  530 FORMAT(//' SINGULAR VALUE CALCULATION AT MEV = ',I6,'    IS COMPLE
     1TE'//)
C
C     NG = NUMBER OF COMPUTED DISTINCT GOOD T-EIGENVALUES.  NEXT
C     GENERATE GAPS BETWEEN GOOD T-EIGENVALUES (BMINGAPS) AND PUT THEM
C     IN G.  G(J) < 0 MEANS THE BMINGAP IS DUE TO THE LEFT-HAND GAP.
C
      NGM1 = NG - 1
      G(NG) = V2(NGM1)-V2(NG)
      G(1) = V2(2)-V2(1)
C
      DO 540 J = 2,NGM1
      T0 = V2(J)-V2(J-1)
      T1 = V2(J+1)-V2(J)
      G(J) = T1
      IF (T0.LT.T1) G(J) = -T0
  540 CONTINUE
C
C     WRITE GOOD T-EIGENVALUES (COMPUTED SINGULAR VALUES) OUT TO FILE 3.
C
      WRITE(3,550)NG,NDIS,MEV,M,N,SVSEED,MATNO,IPARO,MULTOL,IB,BTOL
  550 FORMAT(5I6,I12,I8,I2,'=NG,ND,MEV,M,N,SEED,MN,IPARO'/
     1 E20.12,I6,E13.4,' = MUTOL,INDEX MINIMAL BETA,BTOL'/
     1' SV NO',2X,'T-MULT',10X,'SINGULAR VALUE',7X,'BMINGAP',7X,'TMINGAP
     1')
C
      WRITE(3,560)(I,MP(I),V2(I),G(I),V1(I),  I=1,NG)
  560 FORMAT(I6,I8,E25.16,2E14.3)
C
C     IF CONVERGENCE FLAG ICONV.NE.1 AND NUMBER OF T-MATRICES
C     CONSIDERED TO DATE IS LESS THAN NUMBER ALLOWED, INCREMENT MEV.
C     AND LOOP BACK TO 210 TO REPEAT COMPUTATIONS.  RESTORE BETA(MEV+1).
C
      BETA(MP1) = BETAM
C
      IF (MMB.LT.NMEVS.AND.ICONV.NE.1) GO TO 210
C
C     END OF LOOP ON DIFFERENT SIZE T-MATRICES ALLOWED.
C
  570 CONTINUE
C
      IF(ISTOP.EQ.0)  WRITE(6,580)
  580 FORMAT(/' T-MATRICES (BETA) ARE NOW AVAILABLE, TERMINATE'/)
      IF (IHIS.EQ.1.AND.KMAX.NE.MOLD) WRITE(1,590)
  590 FORMAT(/' ABOVE ARE THE FOLLOWING VECTORS '/
     2 '  BETA(I), I = 1,KMAX+1'/
     3 ' FINAL TWO LANCZOS VECTORS OF ORDERS M,N FOR I = KMAX,KMAX+1'/
     4 ' ALL VECTORS IN THIS FILE HAVE FORMAT 4Z20'/
     5 ' ----- END OF FILE 1 NEW BETA HISTORY---------------'///)
C
      IF (ISTOP.EQ.0) GO TO 690
C
      WRITE(3,600)
```

```
  600 FORMAT(/' ABOVE ARE COMPUTED SINGULAR VALUES'/                     LSV07050
     1 ' NG = NUMBER OF SINGULAR VALUES COMPUTED'/                       LSV07060
     2 ' NDIS = NUMBER OF COMPUTED DISTINCT EIGENVALUES OF T(1,MEV)'/   LSV07070
     3 ' M = ROW ORDER OF A N = COLUMN ORDER,  MATNO = MATRIX IDENT'/   LSV07080
     4 ' MULTOL = T-MULTIPLICITY TOLERANCE FOR T-EIGENVALUES IN BISEC'/ LSV07090
     4 ' T-MULT IS THE T-MULTIPLICITY OF SINGULAR VALUE'/               LSV07100
     5 ' T-MULT = -1 MEANS SPURIOUS T-EIGENVALUE TOO CLOSE'/            LSV07110
     6 ' DO NOT COMPUTE ERROR ESTIMATES FOR SUCH T-EIGENVALUES'/        LSV07120
     7 ' BMINGAP = MINIMAL GAP BETWEEN THE COMPUTED SINGULAR VALUES'/   LSV07130
     8 ' BMINGAP .LT. 0. MEANS MINIMAL GAP IS DUE TO LEFT-HAND GAP'/    LSV07140
     9 ' TMINGAP= MINIMAL GAP W.R.T.  DISTINCT EIGENVALUES IN T(1,MEV)'/LSV07150
     1 ' TMINGAP .LT. 0. MEANS MINGAP IS DUE TO SPURIOUS T-EIGENVALUE'/ LSV07160
     2 ' ----- END OF FILE 3 SINGULAR VALUES------------------------'//)LSV07170
C                                                                       LSV07180
      IF (IDIST.EQ.1) WRITE(11,610)                                     LSV07190
  610 FORMAT(/' ABOVE ARE THE DISTINCT EIGENVALUES OF T(1,MEV).'/       LSV07200
     2 ' THE FORMAT IS      T-MULTIPLICITY   T-EIGENVALUE   TMINGAP'/   LSV07210
     3 '        THIS FORMAT IS REPEATED TWICE ON EACH LINE.'/           LSV07220
     4 ' T-MULTIPLICITY = -1 MEANS THAT THE SUBROUTINE ISOEV HAS TAGGED'LSV07230
     5 /'   THIS COMPUTED SINGULAR VALUE AS HAVING A VERY CLOSE SPURIOUSLSV07240
     6 '/'   T-EIGENVALUE SO THAT NO ERROR ESTIMATE WILL BE COMPUTED'/  LSV07250
     7 '   FOR THAT SINGULAR VALUE IN SUBROUTINE INVERR.'/              LSV07260
     8 ' TMINGAP .LT. 0, TMINGAP IS DUE TO LEFT GAP .GT. 0, RIGHT GAP.'/LSV07270
     9 ' EACH OF THE DISTINCT T-EIGENVALUE TABLES IS FOLLOWED'/         LSV07280
     9 ' BY THE T-MULTIPLICITY PATTERN.'/                               LSV07290
     1 ' NDIS = NUMBER OF COMPUTED DISTINCT EIGENVALUES OF T(1,MEV).'/  LSV07300
     2 ' NG = NUMBER OF COMPUTED SINGULAR VALUES. '/                    LSV07310
     3 ' NISO = NUMBER OF ISOLATED (IN T-MATRIX) SINGULAR VALUES. '/    LSV07320
     4 ' NISO ALSO IS THE COUNT OF +1 ENTRIES IN T-MULTIPLICITY PATTERN.LSV07330
     5 '/' ----- END OF FILE 11 DISTINCT T-EIGENVALUES--------------'//)LSV07340
C                                                                       LSV07350
      IF(NISO.NE.0)  WRITE(4,620)                                       LSV07360
  620 FORMAT(/' ABOVE ARE THE ERROR ESTIMATES OBTAINED FOR THE ISOLATED LSV07370
     1GOOD T-EIGENVALUES'/                                              LSV07380
     1' OBTAINED VIA INVERSE ITERATION IN THE SUBROUTINE INVERR.'/      LSV07390
     1' ALL OTHER GOOD T-EIGENVALUES HAVE CONVERGED.'/                  LSV07400
     2' ERROR ESTIMATE = BETAM*ABS(UM)'/                                LSV07410
     2' WHERE BETAM = BETA(MEV+1) AND UM = U(MEV).'/                    LSV07420
     3' U = UNIT EIGENVECTOR OF T WHERE T*U = SV*U AND SV = ISOLATED GOOLSV07430
     3D T-EIGENVALUE.'/                                                 LSV07440
     4' TMINGAP = GAP TO NEAREST DISTINCT EIGENVALUE OF T(1,MEV).'/     LSV07450
     5' TMINGAP .LT. 0. MEANS MINGAP IS DUE TO A SPURIOUS T-EIGENVALUE.'LSV07460
     6/' ------ END OF FILE 4 ERRINV -------------------------------'//)LSV07470
      GO TO 690                                                         LSV07480
C                                                                       LSV07490
  630 CONTINUE                                                          LSV07500
C                                                                       LSV07510
      IBB = IABS(IBMEV)                                                 LSV07520
      IF (IBMEV.LT.0) WRITE(6,640) MEV,IBB,BETA(IBB)                    LSV07530
  640 FORMAT(/' PROGRAM TERMINATES BECAUSE MEV REQUESTED = ',I6,' IS .GTLSV07540
     1',I6/' AT WHICH AN ABNORMALLY SMALL BETA = ' , E13.4,' OCCURRED'/)LSV07550
      GO TO 690                                                         LSV07560
C                                                                       LSV07570
  650 IF (NDIS.EQ.0.AND.ISTOP.GT.0) WRITE(6,660)                        LSV07580
  660 FORMAT(/' INTERVALS SPECIFIED FOR BISECT DID NOT CONTAIN ANY T-EIGLSV07590
```

```
     1ENVALUES'/' PROGRAM TERMINATES')                                  LSV07600
      GO TO 690                                                         LSV07610
C                                                                       LSV07620
  670 WRITE(6,680) I, NMEV(I)                                           LSV07630
  680 FORMAT(//I6,'TH T-SIZE REQUESTED ',I6,' IS ODD'/                  LSV07640
     1' BUT ONLY EVEN T-SIZES ARE PERMISSIBLE.  PROGRAM TERMINATES FOR ULSV07650
     1SER TO FIX'//)                                                    LSV07660
      GO TO 690                                                         LSV07670
C                                                                       LSV07680
  690 CONTINUE                                                          LSV07690
C                                                                       LSV07700
      STOP                                                              LSV07710
C-----END OF MAIN PROGRAM FOR LANCZOS SINGULAR VALUE COMPUTATIONS-------LSV07720
      END                                                               LSV07730
```

6.4 LSVEC: Main Program, Eigenvector Computations

```
C-----LSVEC (SINGULAR VECTORS OF REAL RECTANGULAR 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 are used in other scientific or
C  engineering research works the names of the authors of these codes
C  and appropriate references to their written work are to be
C  incorporated in the derivative works.
C
C  This header is not to be removed from these codes.
C
C        REFERENCE: Cullum and Willoughby, Chapter 5
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 MAIN PROGRAM FOR COMPUTING A LEFT AND A
C     RIGHT SINGULAR VECTOR CORRESPONDING TO EACH OF A SET
C     OF SINGULAR VALUES WHICH HAVE BEEN COMPUTED ACCURATELY BY THE
C     CORRESPONDING LANCZOS SINGULAR VALUE PROGRAM (LSVAL)
C     FOR REAL RECTANGULAR MATRICES.  THIS PROGRAM COULD BE
C     MODIFIED TO COMPUTE ADDITIONAL SINGULAR VECTORS FOR ANY
C     SINGULAR VALUE THAT IS A MULTIPLE SINGULAR VALUE OF A.
C     THE AMOUNT OF ADDITIONAL COMPUTATION REQUIRED BY SUCH A
C     MODIFICATION DEPENDS UPON THE GIVEN A-MATRIX AND UPON
C     THE PART OF THE SPECTRUM INVOLVED.
C
C     FOR A GIVEN REAL MATRIX A OF ORDER M X N THE LANCZOS RECURSION
C     IS APPLIED TO THE ASSOCIATED REAL SYMMETRIC MATRIX B OF ORDER
C     MN = M+N
C
C                                 ----                ----
C                                |  0                  A  |
C                     B  =       |                        |
C                                |  A-TRANSPOSE        0  |
C                                 ----                ----
C     USING SPECIAL STARTING VECTORS.
C
C     THESE SINGULAR VECTOR COMPUTATIONS ASSUME THAT EACH
C     SINGULAR VALUE THAT IS BEING CONSIDERED HAS CONVERGED AS
```

```
C     AN EIGENVALUE OF THE LANCZOS TRIDIAGONAL MATRICES GENERATED.
C
C     THE EIGENVALUES OF EACH EVEN-ORDERED LANCZOS MATRIX OCCUR
C     IN + AND - PAIRS, AND THE RITZ VECTOR COMPUTATION RESTS ON
C     AN INVERSE ITERATION COMPUTATION FOR A LANCZOS MATRIX.
C     THIS CAUSES AN ANOMALY IN THE SINGULAR VECTOR COMPUTATIONS
C     FOR VERY SMALL SINGULAR VALUES.  IN PRACTICE WE SEE THAT
C     FOR ANY SUCH SINGULAR VALUE THAT ONE MEMBER OF EACH PAIR OF
C     APPROXIMATE SINGULAR VECTORS WILL BE MORE ACCURATE THAN THE
C     OTHER MEMBER OF THAT PAIR IS.  IF IPAR = 1 (STARTING LANCZOS
C     VECTOR IS OF FORM (0,V2) WHERE V2 IS NX1) THEN THE RIGHT
C     SINGULAR VECTOR WILL BE OBTAINED MORE ACCURATELY THAN THE
C     LEFT SINGULAR VECTOR.  IF IPAR = 2 (STARTING LANCZOS VECTOR
C     IS OF FORM (V1,0) WHERE V1 IS MX1) THEN THE LEFT SINGULAR
C     VECTOR WILL BE MORE ACCURATE THAN THE RIGHT SINGULAR VECTOR.
C     PRIOR TO NORMALIZATION THE SIZES OF THESE INACCURATE VECTORS
C     WILL BE THE SAME AS THE SIZE OF THE ASSOCIATED VERY SMALL
C     SINGULAR VALUE.  IN FACT IN THE LIMIT, FOR A ZERO SINGULAR VALUE
C     AND IPAR = 1, THE VECTOR COMPUTED AS THE APPROXIMATION TO THE
C     LEFT SINGULAR VECTOR WILL BE THE 0 VECTOR.  (IF IPAR = 2 THEN
C     THIS WOULD BE THE RIGHT SINGULAR VECTOR).  THE CORRESPONDING
C     ERROR ESTIMATES WILL REFLECT THE INACCURACY OF THE ONE MEMBER
C     OF EACH SUCH PAIR, SINCE THESE ESTIMATES ARE A SUM OF ESTIMATES
C     FOR THE INDIVIDUAL MEMBERS OF THE PAIR.  THEREFORE, FOR ANY VERY
C     SMALL SINGULAR VALUE A CORRESPONDING SINGULAR VECTOR WILL BE
C     COMPUTED ONLY IF THE USER HAS SET THE FLAG ERCONT TO 1.
C
C-----------------------------------------------------------------------
C
C     PFORT VERIFIER IDENTIFIED THE FOLLOWING NONPORTABLE
C     CONSTRUCTIONS
C
C     1.  DATA/MACHEP/ STATEMENT
C     2.  ALL READ(5,*) STATEMENTS (FREE FORMAT)
C     3.  FORMAT(20A4) USED WITH THE EXPLANATORY HEADER, EXPLAN
C     4.  HEXADECIMAL FORMAT (4Z20) USED FOR BETA HISTORY.
C
C     IMPORTANT NOTE:  THIS PROGRAM ALLOWS ENLARGEMENT OF THE
C     BETA ARRAY.  IN PARTICULAR, IF ANY ONE OF THE SINGULAR VALUES
C     SUPPLIED IS T-SIMPLE AND AS AN EIGENVALUE OF THE ASSOCIATED
C     LANCZOS TRIDIAGONAL MATRIX IS NOT CLOSE TO A SPURIOUS
C     EIGENVALUE OF THAT MATRIX, THIS PROGRAM WILL REQUIRE
C     THAT KMAX BE AT LEAST THE LARGEST EVEN NUMBER LESS
C     THAN OR EQUAL TO  (11*MEV)/8 + 13.  IF KMAX IS NOT THAT
C     LARGE, THEN THIS PROGRAM WILL RESET KMAX TO THIS SIZE
C     AND EXTEND THE BETA HISTORY IF REQUIRED.
C     THUS, THE DIMENSION OF THE BETA ARRAY MUST BE
C     LARGE ENOUGH TO ALLOW FOR THIS POSSIBILITY.
C     REMEMBER THAT THE BETA ARRAY, BETA(J), IS SUCH THAT
C     J = 1,..., KMAX+1.  SO IF THE KMAX USED BY THE PROGRAM
C     IS TO BE 3000, THEN BETA MUST BE OF LENGTH AT LEAST 3001.
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION  BETA(5001),V1(5000),V2(5000),RITVEC(30000)
      DOUBLE PRECISION  TVEC(30000),GOODSV(50),SVNEW(50),TLAST(50)
```

```
      DOUBLE PRECISION  SVAL,SVALN,TOLN,TTOL,ERTOL,BATA
      DOUBLE PRECISION  MULTOL,SCALE0,STUTOL,BTOL,LB,UB
      DOUBLE PRECISION  ONE,ZERO,MACHEP,EPSM,TEMP,SUM
      DOUBLE PRECISION  RELTOL,ERROR,TERROR,ERRMIN,BKMIN
      REAL     G(10000),BMINGP(50),TMINGP(50),EXPLAN(20)
      REAL     TERR(50),BERR(50),BERRGP(50),RNORM(50),TBETA(50)
      INTEGER  MP(50),M1(50),M2(50),MA(50),ML(50),MINT(50),MFIN(50)
      INTEGER  SVSEED,SVSOLD,RHSEED,IDELTA(50)
      INTEGER  MBOUND,NTVCON,SVTVEC,TVSTOP,LVCONT,ERCONT,TFLAG
      DOUBLE PRECISION  FINPRO
      DOUBLE PRECISION  DABS, DMAX1, DSQRT, DFLOAT
      REAL ABS
      INTEGER  IABS
C-----------------------------------------------------------------------
      EXTERNAL SVMAT, STRAN
      DATA MACHEP/Z3410000000000000/
      EPSM = 2.D0*MACHEP
C-----------------------------------------------------------------------
C
C     ARRAYS MUST BE DIMENSIONED AS FOLLOWS:
C     1.  BETA: >= (KMAX+1) WHERE KMAX, THE LARGEST SIZE
C               T-MATRIX CONSIDERED BY THE PROGRAM, IS THE
C               LARGER OF THE SIZE OF THE BETA HISTORY PROVIDED
C               ON FILE 2 (IF ANY ) AND THE SIZE WHICH THE PROGRAM
C               SPECIFIES INTERNALLY, THIS LATTER IS ALWAYS
C               < = (11*MEV)/8 + 13, WHERE MEV IS THE SIZE
C               T-MATRIX THAT WAS USED IN THE CORRESPONDING
C               SINGULAR VALUE COMPUTATIONS.  NOTE THAT ALL
C               T-MATRICES CONSIDERED MUST HAVE EVEN ORDER.
C     2.  V1:  >= MAX(M,KMAX)
C     3.  V2:   >= N
C     4.  G:  >= MAX(M,N,KMAX)
C     5.  RITVEC:  >= (N+M)*NGOOD, WHERE NGOOD IS THE NUMBER OF
C                 SINGULAR VALUES SUPPLIED TO THIS PROGRAM.
C     6.  TVEC:  >= CUMULATIVE LENGTH OF ALL THE T-EIGENVECTORS
C               NEEDED TO GENERATE THE DESIRED RITZ VECTORS.  AN
C               EDUCATED GUESS AT AN APPROPRIATE LENGTH CAN BE
C               OBTAINED BY RUNNING THE PROGRAM WITH THE FLAG
C               MBOUND = 1 AND MULTIPLYING THE RESULTING SIZE BY 5/4.
C     7.  GOODSV, TMINGP, BMINGP,TERR, BERR, BERRGP, RNORM,
C         TBETA, TLAST, SVNEW, MP, MA, M1, M2, MINT, MFIN AND
C         IDELTA MUST ALL BE >= NGOOD.
C
C-----------------------------------------------------------------------
C     OUTPUT HEADER
      WRITE(6,10)
   10 FORMAT(/' LANCZOS PROCEDURE FOR REAL, RECTANGULAR MATRICES'/
     1'      COMPUTE SINGULAR VECTORS')
C
C     SET PROGRAM PARAMETERS
C     USER MUST NOT MODIFY SCALE0
      SCALE0 = 5.0D0
      ZERO = 0.0D0
      ONE = 1.0D0
      MPMIN = -1000
```

```
C     CONVERGENCE TOLERANCE FOR T-EIGENVECTORS FOR RITZ COMPUTATIONS    LSV01550
      ERTOL = 1.D-10                                                    LSV01560
C                                                                       LSV01570
C     READ USER-SPECIFIED PARAMETER FROM INPUT FILE 5 (FREE FORMAT)     LSV01580
C                                                                       LSV01590
C     READ USER-PROVIDED HEADER FOR RUN                                 LSV01600
      READ(5,20) EXPLAN                                                 LSV01610
      WRITE(6,20) EXPLAN                                                LSV01620
   20 FORMAT(20A4)                                                      LSV01630
C                                                                       LSV01640
C     READ IN MATNO = MATRIX/RUN IDENTIFICATION NUMBER, 8 DIGITS OR LESSLSV01650
C     AND THE ORDER OF THE MATRIX M X N .                               LSV01660
C                                                                       LSV01670
      READ(5,20) EXPLAN                                                 LSV01680
      READ(5,*) MATNO, M, N                                             LSV01690
      MN = M + N                                                        LSV01700
C                                                                       LSV01710
C     READ IN THE MAXIMUM PERMISSIBLE DIMENSIONS FOR THE TVEC ARRAY     LSV01720
C     (MDIMTV), FOR THE RITVEC ARRAY (MDIMRV), AND FOR THE BETA         LSV01730
C     ARRAY (MBETA).                                                    LSV01740
C                                                                       LSV01750
      READ(5,20) EXPLAN                                                 LSV01760
      READ(5,*) MDIMTV, MDIMRV, MBETA                                   LSV01770
C                                                                       LSV01780
C     READ IN RELATIVE TOLERANCE USED IN DETERMINING APPROPRIATE        LSV01790
C     SIZES FOR THE T-MATRICES USED IN THE SINGULAR VECTOR COMPUTATIONS.LSV01800
C                                                                       LSV01810
      READ(5,20) EXPLAN                                                 LSV01820
      READ(5,*) RELTOL                                                  LSV01830
C                                                                       LSV01840
C     SET FLAGS TO 0 OR 1:                                              LSV01850
C     MBOUND = 1:  PROGRAM TERMINATES AFTER COMPUTING 1ST GUESSES       LSV01860
C                  ON APPROPRIATE T-SIZES FOR USE IN THE RITZ VECTOR    LSV01870
C                  COMPUTATIONS                                         LSV01880
C     NTVCON = 0:  PROGRAM TERMINATES IF THE TVEC ARRAY IS NOT          LSV01890
C                  LARGE ENOUGH TO HOLD ALL THE T-EIGENVECTORS REQUIRED.LSV01900
C     SVTVEC = 0:  THE T-EIGENVECTORS ARE NOT WRITTEN TO FILE 11        LSV01910
C                  UNLESS TVSTOP = 1                                    LSV01920
C     SVTVEC = 1:  WRITE THE T-EIGENVECTORS TO FILE 11.                 LSV01930
C     TVSTOP = 1:  PROGRAM TERMINATES AFTER COMPUTING THE               LSV01940
C                  T-EIGENVECTORS                                       LSV01950
C     LVCONT = 0:  PROGRAM TERMINATES IF THE NUMBER OF T-EIGENVECTORS   LSV01960
C                  COMPUTED IS NOT EQUAL TO THE NUMBER OF RITZ          LSV01970
C                  VECTORS (SINGULAR VECTORS) REQUESTED.                LSV01980
C     ERCONT = 0:  MEANS FOR ANY GIVEN SINGULAR VALUE, A RITZ VECTOR    LSV01990
C                  WILL NOT BE COMPUTED FOR THAT SINGULAR VALUE UNLESS  LSV02000
C                  A T-EIGENVECTOR HAS BEEN IDENTIFIED WITH A LAST      LSV02010
C                  COMPONENT WHICH SATISFIES THE SPECIFIED              LSV02020
C                  CONVERGENCE CRITERION.                               LSV02030
C     ERCONT = 1:  MEANS FOR ANY GIVEN SINGULAR VALUE, A RITZ VECTOR    LSV02040
C                  WILL BE COMPUTED.  IF A T-EIGENVECTOR CANNOT         LSV02050
C                  BE IDENTIFIED WHICH SATISFIES THE LAST               LSV02060
C                  COMPONENT CRITERION, THEN THE PROGRAM WILL           LSV02070
C                  USE THE T-VECTOR THAT CAME CLOSEST TO                LSV02080
C                  SATISFYING THE CRITERION                             LSV02090
```

```
C     IWRITE = 1:  EXTENDED OUTPUT OF INTERMEDIATE COMPUTATIONS
C                  IS WRITTEN TO FILE 6
C     IREAD = 0:   BETA FILE IS REGENERATED.
C     IREAD = 1:   BETA FILE USED IN SINGULAR VALUE 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 THE INVERSE ITERATION ON THE T-MATRICES.
C
      READ(5,20) EXPLAN
      READ(5,*) RHSEED
C
C-----------------------------------------------------------------------
C     INITIALIZE THE ARRAYS FOR THE USER-SPECIFIED MATRIX AND
C     PASS THE STORAGE LOCATIONS OF THESE ARRAYS TO THE MATRIX-VECTOR
C     MULTIPLY SUBROUTINES SVMAT AND STRAN.
C
      CALL USPEC(M,N,MATNO)
C
C-----------------------------------------------------------------------
C     MASK UNDERFLOW AND OVERFLOW
      CALL MASK
C
C-----------------------------------------------------------------------
C     WRITE RUN PARAMETERS OUT TO FILE 6
C
      WRITE(6,30) M,N,MATNO
   30 FORMAT(/' MATRIX ORDER =',I5,' BY ',I5/
     1 ' A-MATRIX AND CASE IDENTIFIER = ',I10/)
C
      WRITE(6,40) MBOUND,NTVCON,SVTVEC,IREAD
   40 FORMAT(/3X,'MBOUND',3X,'NTVCON',3X,'SVTVEC',3X,'IREAD'/3I9,I8/)
C
      WRITE(6,50) TVSTOP,LVCONT,ERCONT,IWRITE
   50 FORMAT(/3X,'TVSTOP',3X,'LVCONT',3X,'ERCONT',3X,'IWRITE'/4I9)
C
      WRITE(6,60) MDIMTV,MDIMRV,MBETA
   60 FORMAT(/3X,'MDIMTV',3X,'MDIMRV',3X,'MBETA'/2I9,I8)
C
      WRITE(6,70) RELTOL,RHSEED
   70 FORMAT(/7X,'RELTOL',3X,'RHSEED'/E13.4,I9)
C
C     FROM FILE 3 READ IN THE NUMBER OF SINGULAR VALUES (NGOOD)
C     FOR WHICH SINGULAR VECTORS ARE REQUESTED, THE ORDER (MEV) OF
C     THE LANCZOS TRIDIAGONAL MATRIX USED IN COMPUTING THESE
```

```
C     SINGULAR VALUES, THE ORDER MOLD X NOLD OF THE USER-SPECIFIED
C     MATRIX USED IN THOSE COMPUTATIONS, THE SEED (SVSEED) USED FOR
C     GENERATING THE STARTING VECTOR THAT WAS USED IN THOSE
C     COMPUTATIONS, AND THE MATRIX/RUN IDENTIFICATION NUMBER (MATOLD)
C     USED IN THOSE COMPUTATIONS.  ALSO READ IN THE NUMBER (NDIS) OF
C     DISTINCT EIGENVALUES OF THE MATRIX T(1,MEV) THAT WERE COMPUTED
C     BUT THIS VALUE IS NOT USED IN THE SINGULAR VECTOR
C     COMPUTATIONS.
C
      READ(3,80) NGOOD,NDIS,MEV,MOLD,NOLD,SVSEED,MATOLD,IPARO
   80 FORMAT(5I6,I12,I8,I2)
C
C     READ IN THE T-MULTIPLICITY TOLERANCE USED IN THE BISEC SUBROUTINE
C     DURING THE COMPUTATION OF THE GIVEN SINGULAR VALUES.
C     ALSO READ IN THE FLAG IB.  IF IB < 0, THEN SOME BETA(I) IN THE
C     T-MATRIX FILE PROVIDED ON FILE 2 FAILED THE ORTHOGONALITY
C     TEST IN THE TNORM SUBROUTINE.  USER SHOULD NOTE THAT THIS
C     PROGRAM PROCEEDS INDEPENDENTLY OF THE SIZE OF THE BETA USED.
C
      READ(3,90) MULTOL,IB,BTOL
   90 FORMAT(E20.12,I6,E13.4)
C
      TEMP = DFLOAT(MEV+1000)
      TTOL = MULTOL/TEMP
      WRITE(6,100) MULTOL,TTOL
  100 FORMAT(/' T-MULTIPLICITY TOLERANCE USED IN THE SINGULAR VALUE COMP
     1UTATIONS WAS',E13.4/' SCALED MACHINE EPSILON IS',E13.4)
C
C     CONTINUE WRITE TO FILE 6 OF THE PARAMETERS FOR THIS RUN
C
      WRITE(6,110)NGOOD,NDIS,MEV,MOLD,NOLD,MATOLD,SVSEED,MULTOL,IB,
     1BTOL,IPARO
  110 FORMAT(/' SINGULAR VALUES SUPPLIED ARE READ IN FROM FILE 3'/
     1 6X,'NG',2X,'NDIS',3X,'MEV',2X,'MOLD',2X,'NOLD',2X,'MATOLD',4X/
     1I8,4I6,I8//6X,'SVSEED',6X,'MULTOL',9X,'IB',8X,'BTOL',4X,'IPARO'/
     1I12,E12.3,I11,E12.4,I9/)
C
C     IS THE ARRAY RITVEC LONG ENOUGH TO HOLD ALL OF THE DESIRED
C     RITZ VECTORS (APPROXIMATE EIGENVECTORS OF B)?
      MNMAX = NGOOD*MN
      IF(MBOUND.EQ.1) GO TO 120
      IF(TVSTOP.NE.1.AND.MNMAX.GT.MDIMRV) GO TO 1600
C
C     CHECK THAT THE ORDERS M,N AND THE MATRIX IDENTIFICATION NUMBER
C     MATNO SPECIFIED BY THE USER AGREE WITH THOSE READ IN FROM
C     FILE 3.
  120 ITEMP = (MOLD-M)**2+(NOLD-N)**2+(MATOLD-MATNO)**2
      IF (ITEMP.NE.0) GO TO 1620
C
C     READ IN FROM FILE 3, THE T-MULTIPLICITIES OF THE SINGULAR VALUES
C     WHOSE SINGULAR VECTORS ARE TO BE COMPUTED, THE VALUES OF THESE
C     SINGULAR VALUES AND THEIR MINIMAL GAPS AS SINGULAR VALUES OF THE
C     USER-SPECIFIED MATRIX AND OF THE RELATED T-MATRIX.
C
      READ(3,20) EXPLAN
```

```
      READ(3,130) (MP(J),GOODSV(J),BMINGP(J),TMINGP(J), J=1,NGOOD)
  130 FORMAT(6X,I8,E25.16,2E14.3)
C
      WRITE(6,140) (J,GOODSV(J),MP(J),BMINGP(J),  J=1,NGOOD)
  140 FORMAT(/' SINGULAR VALUES READ IN FROM FILE 3 AND THEIR T-MULTIPLI
     1CITIES'/4X,' J ',4X,' SINGULAR VALUE',5X,'TMULT',4X,'BMINGP'/
     1(I6,E20.12,I6,E13.4))
C
      WRITE(6,150) MEV,SVSEED
  150 FORMAT(/' THESE SINGULAR VALUES WERE COMPUTED USING A T-MATRIX OF
     1ORDER ',I5/' AND SEED FOR RANDOM NUMBER GENERATOR =',I12)
C
C     READ IN THE ERROR ESTIMATES
C
C     CHECK WHETHER OR NOT THERE ARE ANY ISOLATED T-EIGENVALUES IN
C     THE T-EIGENVALUES PROVIDED (HERE THE SINGULAR VALUES ARE
C     CONSIDERED AS EIGENVALUES OF THE ASSOCIATED LANCZOS TRIDIAGONAL
C     MATRICES.)
      DO 160 J=1,NGOOD
      IF(MP(J).EQ.1) GO TO 170
  160 CONTINUE
      GO TO 190
  170 READ(4,20) EXPLAN
      READ(4,20) EXPLAN
      READ(4,20) EXPLAN
      READ(4,180) NISO
  180 FORMAT(18X,I6)
      READ(4,20) EXPLAN
      READ(4,20) EXPLAN
      READ(4,20) EXPLAN
  190 DO 220 J=1,NGOOD
      BERR(J) = 0.D0
      IF(MP(J).NE.1) GO TO 220
      READ(4,200) SVAL, BERR(J)
  200 FORMAT(10X,E25.16,E14.3)
      IF(DABS(SVAL - GOODSV(J)).LT.1.D-10) GO TO 220
      WRITE(6,210) SVAL,GOODSV(J)
  210 FORMAT(' PROBLEM WITH READ IN OF ERROR ESTIMATES'/' SINGULAR VALUE
     1READ IN',E20.12,' DOES NOT MATCH GOODSV(J) ='/E20.12)
      GO TO 1860
C
  220 CONTINUE
C
      WRITE(6,230) (J,GOODSV(J),BERR(J), J=1,NGOOD)
  230 FORMAT(' ERROR ESTIMATES ='/4X,' J',3X,'SINGULAR VALUE',8X,
     1'ESTIMATE'/(I6,E20.12,E14.3))
C
      IF(IREAD.EQ.0) IPAR = IPARO
      IF(IREAD.EQ.0)  GO TO 350
C
C     READ IN THE SIZE OF THE T-MATRIX PROVIDED ON FILE 2.  READ IN
C     THE ORDER OF THE USER-SPECIFIED MATRIX , THE FLAGS IPARO
C     AND IPAR WHICH INDICATE RESPECTIVELY THE PARITY OF THE
C     STARTING VECTOR USED IN THE GENERATION OF THE EXISTING
C     BETA AND THE PARITY OF THE NEXT LANCZOS VECTOR THAT
```

```
C     HAS TO BE GENERATED IF THE BETA HISTORY IS EXTENDED,
C     THE SEED USED BY THE RANDOM NUMBER GENERATOR WHEN
C     GENERATING THE STARTING VECTOR THAT WAS USED, AND THE
C     MATRIX/TEST IDENTIFICATION NUMBER THAT WERE USED IN
C     THE LANCZOS SINGULAR VALUE COMPUTATIONS.  IF THE FLAG
C     IREAD = 0, REGENERATE HISTORY AND DO NOT READ ANYTHING
C     FROM FILE 2.  HISTORY MUST BE STORED IN MACHINE FORMAT,
C     ((4Z20) FOR IBM 3081).
C
      READ(2,240) KMAX,MOLD,NOLD,IPARO,IPAR,SVSOLD,MATOLD
  240 FORMAT(3I6,2I3,I12,I8)
C
      WRITE(6,250) KMAX,MOLD,NOLD,IPARO,IPAR,SVSOLD,MATOLD
  250 FORMAT(/' READ IN HEADER FROM BETA FILE 2'/
     1 2X,'KMAX',2X,'MOLD',2X,'NOLD',2X,'IPARO',2X,'IPAR',6X,'SVSOLD
     1 ',2X,'MATOLD'/3I6,I7,I6,I12,I12)
C
C     CHECK THAT THE PARAMETERS READ IN AGREE WITH WHAT THE USER
C     HAS SPECIFIED
      IF(MOLD.NE.M.OR.NOLD.NE.N.OR.MATOLD.NE.MATNO.OR.SVSOLD.NE.SVSEED)
     1 GO TO 1640
C
      IF(IPARO.EQ.1) WRITE(6,260)
      IF(IPARO.EQ.2) WRITE(6,270)
  260 FORMAT(/' STARTING VECTOR USED IN EXISTING SINGULAR VALUE HISTORY
     1WAS'/' OF THE FORM (0,V2)')
  270 FORMAT(/' STARTING VECTOR USED IN EXISTING SINGULAR VALUE HISTORY
     1WAS'/' OF THE FORM (V1,0)')
C
      KMAX1 = KMAX + 1
C
C     READ IN THE T-MATRICES FROM FILE 2.  THESE ARE USED TO GENERATE
C     THE T-EIGENVECTORS THAT WILL BE USED IN THE RITZ VECTOR
C     COMPUTATIONS.  HISTORY MUST BE STORED IN 4Z20 FORMAT.
C
      READ(2,280) (BETA(J), J=1,KMAX1)
  280 FORMAT(4Z20)
C
      READ(2,280) (V1(J), J=1,M)
      READ(2,280) (V2(J), J=1,N)
C
C     KMAX MAY BE ENLARGED IF THE SIZE AT WHICH THE SINGULAR VALUE
C     COMPUTATIONS WERE PERFORMED IS ESSENTIALLY KMAX AND
C     THERE IS AT LEAST ONE SINGULAR VALUE THAT IS SIMPLE AS AN
C     EIGENVALUE OF T(1,MEV), AND IF ITS NEAREST NEIGHBOR IN THE
C     T-MATRIX IS TOO CLOSE, THAT NEIGHBOR IS A 'GOOD' T-EIGENVALUE.
      DO 290 J = 1,NGOOD
      IF(MP(J).EQ.1) GO TO 310
  290 CONTINUE
      WRITE(6,300)
  300 FORMAT(/' ALL SINGULAR VALUES USED ARE T-MULTIPLE OR CLOSE TO SPUR
     1IOUS EIGENVALUES'/' (AS EIGENVALUES OF T(1,MEV)) SO KMAX IS NOT CH
     1ANGED'/)
      IF(KMAX.LT.MEV) GO TO 1660
      GO TO 330
```

```
C                                                                       LSV04300
  310 KMAXN= (11*MEV)/8 + 12                                            LSV04310
      IF((KMAXN/2)*2.NE.KMAXN) KMAXN = KMAXN + 1                        LSV04320
      IF(MBETA.LE.KMAXN) GO TO 1840                                     LSV04330
      IF(KMAX.GE.KMAXN )  GO TO 330                                     LSV04340
      WRITE(6,320) KMAX, KMAXN                                          LSV04350
  320 FORMAT(' ENLARGE KMAX FROM ',I6,' TO ',I6)                        LSV04360
      MOLD1 = KMAX + 1                                                  LSV04370
      KMAX = KMAXN                                                      LSV04380
      GO TO 420                                                         LSV04390
C                                                                       LSV04400
  330 WRITE(6,340) KMAX                                                 LSV04410
  340 FORMAT(/' T-MATRICES HAVE BEEN READ IN FROM FILE 2'/' THE LARGEST LSV04420
     1SIZE T-MATRIX ALLOWED IS',I6/)                                    LSV04430
C                                                                       LSV04440
      IF(IREAD.EQ.1) GO TO 460                                          LSV04450
C                                                                       LSV04460
C     REGENERATE THE BETA                                               LSV04470
C                                                                       LSV04480
  350 MOLD1 = 1                                                         LSV04490
C                                                                       LSV04500
      IF(IPAR.EQ.1) WRITE(6,360)                                        LSV04510
      IF(IPAR.EQ.2) WRITE(6,370)                                        LSV04520
  360 FORMAT(/' STARTING VECTOR USED IN HISTORY REGENERATION IS OF THE  LSV04530
     1FORM (0,V2)')                                                     LSV04540
  370 FORMAT(/' STARTING VECTOR USED IN HISTORY REGENERATION IS OF THE  LSV04550
     1FORM (V1,0)')                                                     LSV04560
C                                                                       LSV04570
      DO 380 J = 1,NGOOD                                                LSV04580
      IF(MP(J).EQ.1) GO TO 400                                          LSV04590
  380 CONTINUE                                                          LSV04600
      KMAX = MEV + 12                                                   LSV04610
      IF((KMAX/2)*2.NE.KMAX) GO TO 1680                                 LSV04620
      WRITE(6,390) KMAX                                                 LSV04630
  390 FORMAT(/' ALL SINGULAR VALUES FOR WHICH SINGULAR VECTORS ARE TO BELSV04640
     1COMPUTED ARE EITHER T-MULTIPLE OR CLOSE TO'/' A SPURIOUS T-EIGENVALSV04650
     1LUE THEREFORE SET KMAX = MEV + 12 = ',I7)                         LSV04660
      GO TO 420                                                         LSV04670
C                                                                       LSV04680
  400 KMAXN = (11*MEV)/8 + 12                                           LSV04690
      IF((KMAXN/2)*2.NE.KMAXN) KMAXN = KMAXN + 1                        LSV04700
      IF(MBETA.LE.KMAXN) GO TO 1840                                     LSV04710
      WRITE(6,410) KMAXN                                                LSV04720
  410 FORMAT(' SET KMAX EQUAL TO ',I6)                                  LSV04730
      KMAX = KMAXN                                                      LSV04740
C                                                                       LSV04750
  420 KMAX1 = KMAX + 1                                                  LSV04760
      WRITE(6,430) MOLD1,KMAX1                                          LSV04770
  430 FORMAT(/' LANCZS SUBROUTINE GENERATES BETA(J+1), J =',            LSV04780
     1 I6,' TO ', I6/)                                                  LSV04790
      IF(IREAD.EQ.1.AND.IPAR.EQ.1) WRITE(6,440)                         LSV04800
      IF(IREAD.EQ.1.AND.IPAR.EQ.2) WRITE(6,450)                         LSV04810
  440 FORMAT(/' FIRST LANCZOS VECTOR IN HISTORY EXTENSION IF OF THE FORMLSV04820
     1 (0,V2)')                                                         LSV04830
  450 FORMAT(/' FIRST LANCZOS VECTOR IN HISTORY EXTENSION IF OF THE FORMLSV04840
```

```
     1 (V1,0)')
C
C-----------------------------------------------------------------------
C
      CALL LANCZS(SVMAT,STRAN,BETA,V1,V2,G,KMAX,MOLD1,M,N,IPAR,SVSEED)
C
C-----------------------------------------------------------------------
C
  460 CONTINUE
C
C     THE SUBROUTINE STURMI DETERMINES THE SMALLEST SIZE T-MATRIX FOR
C     WHICH THE SINGULAR VALUE IN QUESTION IS AN EIGENVALUE (TO
C     WITHIN A SPECIFIED TOLERANCE) AND IF POSSIBLE THE SMALLEST
C     SIZE T-MATRIX FOR WHICH THE SINGULAR VALUE IS A DOUBLE
C     EIGENVALUE (TO WITHIN THE SAME TOLERANCE).  THE SIZE
C     T-MATRIX THAT WILL BE USED IN EACH OF THE RITZ VECTOR COMPUTATIONS
C     IS THEN DETERMINED BY LOOPING ON THE SIZE OF THE T-EIGENVECTOR
C     COMPUTATIONS, STARTING WITH A SIZE DETERMINED FROM THE
C     INFORMATION OBTAINED FROM STURMI.
C
      STUTOL = SCALE0*MULTOL
      IF(IWRITE.EQ.1) WRITE(6,470)
  470 FORMAT(' FROM STURMI')
      DO 510 J = 1,NGOOD
      SVAL = GOODSV(J)
C     COMPUTE THE TOLERANCES USED BY STURMI TO DETERMINE AN INTERVAL
C     CONTAINING THE SINGULAR VALUE SVAL.
      TEMP = DABS(SVAL)*RELTOL
      TOLN = DMAX1(TEMP,STUTOL)
C
C-----------------------------------------------------------------------
C
      CALL STURMI(BETA,SVAL,TOLN,EPSM,KMAX,MK1,MK2,IC,IWRITE)
C
C-----------------------------------------------------------------------
C
C     STORE THE COMPUTED ORDERS OF T-MATRICES FOR LATER PRINTOUT
      IF(MK1.GT.1) GO TO 475
C     SVAL IS VERY SMALL SINGULAR VALUE, RESET MK1 TO CORRECT VALUE
      MK1 = MK2
      MK2 = MIN0(2*MK1,KMAX)
      M1(J) = MK1
      M2(J) = MK2
      ML(J) = MK2
      GO TO 476
  475 M1(J) = MK1
      M2(J) = MK2
      ML(J) = (MK1 + 3*MK2)/4
      IF(MK2.EQ.KMAX)  ML(J) = KMAX
C
  476 IF(IC.GT.0) GO TO 490
C     IC = 0 MEANS THERE WAS NO T-EIGENVALUE IN THE DESIGNATED INTERVAL
C     EVEN BY T-SIZE KMAX.  THIS MEANS THAT THE SINGULAR VALUE
C     PROVIDED HAS NOT YET CONVERGED SO PROGRAM DOES NOT COMPUTE
C     A SINGULAR VECTOR FOR IT.
```

```
      WRITE(6,480) J,GOODSV(J),MK1,MK2
  480 FORMAT(I6,'TH SINGULAR VALUE',E20.12,' HAS NOT CONVERGED '/
     1' SO DO NOT COMPUTE ANY T-EIGENVECTOR OR RITZ VECTOR FOR IT'
     1/' MK1 AND MK2 FOR THIS SINGULAR VALUE WERE',2I6)
      MP(J) = MPMIN
      MA(J) = -2*KMAX
      GO TO 510
C     COMPUTE AN APPROPRIATE SIZE T-MATRIX FOR THE GIVEN SINGULAR
C     VALUE.
  490 IF(M2(J).EQ.KMAX) GO TO 500
C     M1 AND M2 WERE BOTH DETERMINED
      MAJ = (3*M1(J) + M2(J))/4  + 1
      IF((MAJ/2)*2.NE.MAJ)  MAJ = MAJ + 1
      MA(J) = MAJ
      GO TO 510
C     M2 NOT DETERMINED
  500 MAJ = (5*M1(J))/4  + 1
      IF((MAJ/2)*2.NE.MAJ)  MAJ = MAJ + 1
      MA(J) = MAJ
C
  510 CONTINUE
C
      IF (IWRITE.EQ.1) WRITE(6,520) (MA(JJ), JJ=1,NGOOD)
  520 FORMAT(/' 1ST GUESS AT APPROPRIATE SIZE T-MATRICES'/
     1 ' ACTUAL VALUES WILL PROBABLY BE 1/4 AGAIN AS MUCH'/(13I6))
C
C     PRINT OUT TO FILE 10 1ST GUESSES AT SIZES OF THE T-MATRICES TO
C     BE USED IN THE SINGULAR VECTOR COMPUTATIONS.
C     PROGRAM LOOPS ON T-SIZE TO DETERMINE APPROPRIATE SIZE T-MATRIX.
      WRITE(10,530) N,KMAX
  530 FORMAT(2I8,' = ORDER OF USER MATRIX AND MAX ORDER OF T(1,MEV)')
C
      WRITE(10,540)
  540 FORMAT(/' 1ST GUESS AT APPROPRIATE SIZE T-MATRICES'/
     1 ' ACTUAL VALUES WILL PROBABLY BE 1/4 AGAIN AS MUCH'/)
C
      WRITE(10,550)
  550 FORMAT(4X,'J',7X,'GOODSV(J)',4X,'M1(J)',1X,'M2(J)',1X,'MA(J)')
C
      WRITE(10,560) (J,GOODSV(J),M1(J),M2(J), MA(J), J=1,NGOOD)
  560 FORMAT(I5,E19.12,3I6)
C
      IF(MBOUND.EQ.1) WRITE(10,570)
  570 FORMAT(/' GOODSV(J) IS A GOOD EIGENVALUE OF T(1,MEV)'/
     1 ' M1 = SMALLEST VALUE OF M SUCH THAT T(1,M) HAS AT LEAST'/
     1 '      ONE EIGENVALUE IN THE INTERVAL (SV-TOLN,SV+TOLN)'/
     1 '      TOLN(J) = DMAX1(GOODSV(J)*RELTOL, SCALE0*MULTOL)'/
     1 ' M2 = SMALLEST M (IF ANY) SUCH THAT IN THE ABOVE INTERVAL'/
     1 '      T(1,M) HAS AT LEAST TWO EIGENVALUES '/
     1 ' INITIAL VALUE OF MA(J) IS CHOSEN HEURISTICALLY'/
     1 ' PROGRAM LOOPS ON SIZE OF T-MATRIX TO GET APPROPRIATE SIZE'/
     1 ' END OF SIZES OF T-MATRICES FILE 10'///)
C
C
C     TERMINATE AFTER COMPUTING 1ST GUESSES AT  SIZES OF THE
```

```
C     T-MATRICES REQUIRED FOR THE GIVEN SINGULAR VALUES?
      IF(MBOUND.EQ.1) GO TO 1700
C
C
C     IS THERE ROOM FOR ALL OF THE REQUESTED T-EIGENVECTORS?
      MTOL = 0
      DO 580 J = 1,NGOOD
      IF(MP(J).EQ.MPMIN) GO TO 580
      MTOL = MTOL + IABS(MA(J))
  580 CONTINUE
      MTOL = (5*MTOL)/4
      IF(MTOL.GT.MDIMTV.AND.NTVCON.EQ.0) GO TO 1720
C
C-----------------------------------------------------------------------
C     GENERATE A RANDOM VECTOR TO BE USED REPEATEDLY BY
C     SUBROUTINE INVERM
C
      IIL = RHSEED
      CALL GENRAN(IIL,G,KMAX)
C
C-----------------------------------------------------------------------
C
C     FOR EACH SINGULAR VALUE LOOP ON T-EIGENVECTOR COMPUTATIONS
C     TO COMPUTE AN APPROPRIATE T-EIGENVECTOR TO USE IN THE
C     RITZ VECTOR COMPUTATIONS.
C
      MTOL = 0
      NTVEC = 0
      ILBIS = 0
      DO 770 J = 1,NGOOD
      ICOUNT = 0
      ERRMIN = 10.D0
      MABEST = MPMIN
      IF(MP(J).EQ.MPMIN) GO TO 770
      TFLAG = 0
      SVAL = GOODSV(J)
      TEMP = RELTOL*DABS(SVAL)
      UB = SVAL + DMAX1(STUTOL,TEMP)
      LB = SVAL - DMAX1(STUTOL,TEMP)
      LB = DMAX1(LB,ZERO)
  590 KMAXU = IABS(MA(J))
C
C     SELECT A SUITABLE INCREMENT FOR THE ORDERS OF THE T-MATRICES
C     TO BE CONSIDERED IN DETERMINING APPROPRIATE SIZES FOR THE RITZ
C     VECTOR COMPUTATIONS.  ALL ORDERS CONSIDERED MUST BE EVEN.
      IF(ICOUNT.GT.0) GO TO 610
C     SELECT IDELTA(J) BASED UPON THE T-MULTIPLICITY OBTAINED
      IF(M2(J).EQ.KMAX) GO TO 600
C     M2 DETERMINED
      IDEL = ((3*M1(J) + 5*M2(J))/8  +  1 - IABS(MA(J)))/10  +  1
      IF((IDEL/2)*2.NE.IDEL) IDEL = IDEL + 1
      IDELTA(J) = IDEL
      GO TO 610
C     M2 NOT DETERMINED
  600 MAMAX = MIN0((11*MEV)/8 + 12, (13*M1(J))/8 + 1)
```

```
      IDEL = (MAMAX - IABS(MA(J)))/10  +  1
      IF((IDEL/2)*2.NE.IDEL)  IDEL = IDEL + 1
      IDELTA(J) = IDEL
  610 ICOUNT = ICOUNT + 1
C
C-----------------------------------------------------------------------
C     TO MIMIMIZE THE EFFECT OF THE ONE-SIDED ACCEPTANCE TEST FOR
C     EIGENVALUES IN THE BISEC SUBROUTINE, RECOMPUTE THE GIVEN
C     SINGULAR VALUE AT THE SPECIFIED KMAXU
C
      CALL LBISEC(BETA,EPSM,SVAL,SVALN,LB,UB,TTOL,KMAXU,NEVT)
C
C-----------------------------------------------------------------------
C
C     CHECK WHETHER OR NOT GIVEN T-MATRIX HAS AN EIGENVALUE IN THE
C     SPECIFIED INTERVAL AND IF SO WHAT ITS T-MULTIPLICITY IS.
C
      IF(NEVT.EQ.1) GO TO 650
      IF(NEVT.NE.0)  GO TO 630
      ILBIS = 1
      WRITE(6,620) SVAL,KMAXU
  620 FORMAT(/' PROBLEM ENCOUNTERED IN RECOMPUTATION OF USER-SUPPLIED SI
     1NGULAR VALUE',E20.12/' THE SIZE T-MATRIX SPECIFIED',I6,' DOES NOT
     1HAVE A SINGULAR VALUE IN THE INTERVAL SPECIFIED'/' INCREASE SIZE A
     1ND TRY AGAIN'/)
      GO TO 670
C
  630 IF(NEVT.GT.1)  WRITE(6,640) SVAL,KMAXU
  640 FORMAT(/' PROBLEM ENCOUNTERED IN RECOMPUTATION OF USER-SUPPLIED
     1SINGULAR VALUE',E20.12/' FOR THE SIZE T-MATRIX SPECIFIED =',I6,' T
     1HE GIVEN SINGULAR VALUE IS T-MULTIPLE IN THE INTERVAL SPECIFIED'/'
     1SOMETHING IS WRONG, THEREFORE NO SINGULAR VECTORS WILL BE COMPUTED
     1 FOR THIS SINGULAR VALUE'/)
C
      MP(J) = MPMIN
      MA(J) = -2*KMAX
      GO TO 770
C
  650 CONTINUE
      ILBIS = 0
C
C
      SVNEW(J) = SVALN
      SVAL = SVALN
      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 780
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) - SVAL)*U = RHS  FOR EACH SINGULAR VALUE TO
C     OBTAIN THE DESIRED T-EIGENVECTOR.
C
      IF(IWRITE.EQ.1)  WRITE(6,660) J
  660 FORMAT(/I6,'TH SINGULAR VALUE ')
C
      CALL INVERM(BETA,V1,TVEC(KINT),SVAL,ERROR,TERROR,EPSM,G,KMAXU,
     1 IT,IWRITE)
C
C-----------------------------------------------------------------------
C
      TERR(J) = TERROR
      TLAST(J) = ERROR
      KMAXU1 = KMAXU + 1
      TBETA(J) = BETA(KMAXU1)*ERROR
C
C     AFTER COMPUTING EACH OF THE T-EIGENVECTORS,
C     CHECK THE SIZE OF THE ERROR ESTIMATE, ERROR.
C     IF THIS ESTIMATE IS NOT AS SMALL AS DESIRED AND
C     |MA(J)| < ML(J), ATTEMPT TO INCREASE THE SIZE OF |MA(J)|
C     AND REPEAT THE T-EIGENVECTOR COMPUTATIONS.
C
      IF(ERROR.LT.ERTOL.OR.TFLAG.EQ.1)  GO TO 760
C
      IF(ERROR.GE.ERRMIN) GO TO 670
C     LAST COMPONENT IS LESS THAN MINIMAL TO DATE
      ERRMIN = ERROR
      MABEST = MA(J)
  670 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 690
C     NEW MA(J) IS GREATER THAN MAXIMUM ALLOWED.
      IF(ERCONT.EQ.0.OR.MABEST.EQ.MPMIN) GO TO 710
      TFLAG = 1
      MA(J) = MABEST
      IF(ILBIS.EQ.0)   MTOL = MTOL - KMAXU
      WRITE(6,680) MA(J)
  680 FORMAT(' 10 ORDERS WERE CONSIDERED.  NONE SATISFIED THE ERROR TEST
     1'/' THEREFORE USE THE BEST ORDER OBTAINED FOR THE T-EIGENVECTORS'
     1,I6)
      GO TO 590
C
  690 MA(J) = ITEST
C
      MT = IABS(MA(J))
      IF(IWRITE.EQ.1.AND.ILBIS.EQ.0) WRITE(6,700) MT
  700 FORMAT(/' CHANGE SIZE OF T-MATRIX TO ',I6,' RECOMPUTE T-EIGENVECTO
     1R')
C
```

```
      IF(ILBIS.EQ.0)  MTOL = MTOL - KMAXU                               LSV07600
C                                                                       LSV07610
      GO TO 590                                                         LSV07620
C                                                                       LSV07630
C     APPROPRIATE SIZE T-MATRIX WAS NOT OBTAINED                        LSV07640
  710 CONTINUE                                                          LSV07650
      WRITE(10,720) J,SVAL,MP(J)                                        LSV07660
  720 FORMAT(/' ON 10 INCREMENTS NOT ABLE TO IDENTIFY APPROPRIATE SIZE  LSV07670
     1T-MATRIX FOR'/                                                    LSV07680
     1I4,' TH SINGULAR VALUE  = ',E20.12,' T-MULTIPLICITY =',I4/)       LSV07690
      IF(M2(J).EQ.KMAX) WRITE(10,730)                                   LSV07700
      IF(M2(J).LT.KMAX) WRITE(10,740)                                   LSV07710
  730 FORMAT(/' ORDERS TESTED RANGED FROM 5*M1(J)/4 TO APPROXIMATELY'/  LSV07720
     1 '  MIN(11*MEV/8, 13*M1(J)/8)'/)                                  LSV07730
  740 FORMAT(/' ORDERS TESTED RANGED FROM (3*M1(J)+M2(J))/4  TO APPROXIMLSV07740
     1ATELY'/'  (3*M1(J)+5*M2(J))/8'/)                                  LSV07750
      WRITE(10,750)                                                     LSV07760
  750 FORMAT(' ALLOWING LARGER ORDERS FOR THE T-MATRICES MAY RESULT IN  LSV07770
     1 SUCCESS'/' BUT PROBABLY WILL NOT.  PROBLEM IS PROBABLY DUE TO'   LSV07780
     1 /' LACK OF CONVERGENCE OF GIVEN SINGULAR VALUE, CHECK THE ERROR ELSV07790
     1STIMATE')                                                         LSV07800
      MP(J) = MPMIN                                                     LSV07810
      IF(ILBIS.EQ.0)  MTOL = MTOL - KMAXU                               LSV07820
      GO TO 770                                                         LSV07830
  760 NTVEC = NTVEC + 1                                                 LSV07840
C                                                                       LSV07850
  770 CONTINUE                                                          LSV07860
      NGOODC = NGOOD                                                    LSV07870
      GO TO 800                                                         LSV07880
C                                                                       LSV07890
C     COME HERE IF THERE IS NOT ENOUGH ROOM FOR ALL OF T-EIGENVECTORS   LSV07900
  780 NGOODC = J-1                                                      LSV07910
      WRITE(6,790) J,MTOL,MDIMTV                                        LSV07920
  790 FORMAT(/' NOT ENOUGH ROOM IN TVEC ARRAY FOR ',I4,'TH T-EIGENVECTORLSV07930
     1'/'  TVEC DIMENSION REQUESTED = ',I6,' BUT TVEC HAS DIMENSION ',I6LSV07940
     1/)                                                                LSV07950
      IF(NGOODC.EQ.0) GO TO 1740                                        LSV07960
      MTOL = MTOL-KMAXU                                                 LSV07970
C                                                                       LSV07980
  800 CONTINUE                                                          LSV07990
C                                                                       LSV08000
C     THE LOOP ON T-EIGENVECTOR COMPUTATIONS IS COMPLETE.               LSV08010
C     WRITE OUT THE SIZE T-MATRICES THAT WILL BE USED FOR               LSV08020
C     THE RITZ VECTOR COMPUTATIONS.                                     LSV08030
C                                                                       LSV08040
      WRITE(10,810)                                                     LSV08050
  810 FORMAT(/' SIZES OF T-MATRICES THAT WILL BE USED IN THE RITZ COMPUTLSV08060
     1ATIONS'/5X,'J',8X,' SINGULAR VALUE  ',1X,'MA(J)')                 LSV08070
C                                                                       LSV08080
      WRITE(10,820)   (J,GOODSV(J),MA(J),  J=1,NGOOD)                   LSV08090
  820 FORMAT(I6,E25.14,I6)                                              LSV08100
      WRITE(10,570)                                                     LSV08110
C                                                                       LSV08120
      WRITE(6,830) MTOL                                                 LSV08130
  830 FORMAT(/' THE CUMULATIVE LENGTH OF THE T-EIGENVECTORS IS',I18)    LSV08140
```

```
C                                                                       LSV08150
      WRITE(6,840) NTVEC,NGOOD                                          LSV08160
  840 FORMAT(/I6,' T-EIGENVECTORS OUT OF',I6,' REQUESTED WERE COMPUTED')LSV08170
C                                                                       LSV08180
C     SAVE THE T-EIGENVECTORS ON FILE 11?                               LSV08190
      IF(TVSTOP.NE.1.AND.SVTVEC.EQ.0) GO TO 900                         LSV08200
C                                                                       LSV08210
      WRITE(11,850) NTVEC,MTOL,MATNO,SVSEED                             LSV08220
  850 FORMAT(I6,3I12,' = NTVEC,MTOL,MATNO,SVSEED')                      LSV08230
C                                                                       LSV08240
      DO 880 J=1,NGOODC                                                 LSV08250
C     IF MP(J) = MPMIN THEN NO SUITABLE T-EIGENVECTOR IS AVAILABLE      LSV08260
C     FOR THAT SINGULAR VALUE.                                          LSV08270
      IF(MP(J).EQ.MPMIN) WRITE(11,860) J,MA(J),GOODSV(J),MP(J)          LSV08280
  860 FORMAT(2I6,E20.12,I6/' TH SINGVAL,T-SIZE,SVALUE,FLAG,NO EIGVEC')  LSV08290
      IF(MP(J).NE.MPMIN) WRITE(11,870) J,MA(J),GOODSV(J),MP(J)          LSV08300
  870 FORMAT(I6,I6,E20.12,I6/' T-EIGVEC,SIZE T,SVALUE OF A,MP(J)')      LSV08310
      IF(MP(J).EQ.MPMIN) GO TO 880                                      LSV08320
      KI = MINT(J)                                                      LSV08330
      KF = MFIN(J)                                                      LSV08340
C                                                                       LSV08350
      WRITE(11,280) (TVEC(K), K=KI,KF)                                  LSV08360
C                                                                       LSV08370
  880 CONTINUE                                                          LSV08380
C                                                                       LSV08390
      IF(TVSTOP.NE.1) GO TO 900                                         LSV08400
C                                                                       LSV08410
      WRITE(6,890) TVSTOP, NTVEC,NGOOD                                  LSV08420
  890 FORMAT(/' USER SET TVSTOP = ',I1/                                 LSV08430
     1' THEREFORE PROGRAM TERMINATES AFTER T-EIGENVECTOR COMPUTATIONS'/ LSV08440
     1' T-EIGENVECTORS THAT WERE COMPUTED ARE SAVED ON FILE 11'/        LSV08450
     1I8,' T-EIGENVECTORS WERE COMPUTED OUT OF',I7,' REQUESTED'/)       LSV08460
C                                                                       LSV08470
      GO TO 1860                                                        LSV08480
C                                                                       LSV08490
  900 CONTINUE                                                          LSV08500
C     IF NOT ALL OF THE REQUESTED T-EIGENVECTORS WERE COMPUTED,         LSV08510
C     ARE THE LANCZOS SINGULAR VECTOR COMPUTATIONS CONTINUED?           LSV08520
C                                                                       LSV08530
      IF(NTVEC.NE.NGOOD.AND.LVCONT.EQ.0) GO TO 1760                     LSV08540
C                                                                       LSV08550
C     COMPUTE THE MAXIMUM SIZE OF THE T-MATRIX USED FOR THOSE           LSV08560
C     SINGULAR VALUES WITH GOOD ERROR ESTIMATES.                        LSV08570
C                                                                       LSV08580
      KMAXU = 0                                                         LSV08590
      DO 910 J = 1,NGOODC                                               LSV08600
      MT = IABS(MA(J))                                                  LSV08610
      IF(MT.LT.KMAXU.OR.MP(J).EQ.MPMIN) GO TO 910                       LSV08620
      KMAXU = MT                                                        LSV08630
  910 CONTINUE                                                          LSV08640
C                                                                       LSV08650
      IF(KMAXU.EQ.0) GO TO 1800                                         LSV08660
C                                                                       LSV08670
      WRITE(6,920) KMAXU                                                LSV08680
  920 FORMAT(/I6,' = LARGEST SIZE T-MATRIX TO BE USED IN THE RITZ VECTORLSV08690
```

```
     1 COMPUTATIONS')
C
C     COUNT THE NUMBER OF RITZ VECTORS NOT BEING COMPUTED
      MREJEC = 0
      DO 930 J=1,NGOODC
  930 IF(MP(J).EQ.MPMIN)  MREJEC = MREJEC + 1
      MREJET = MREJEC + (NGOOD-NGOODC)
      IF(MREJET.NE.0) WRITE(6,940) MREJET
  940 FORMAT(/' RITZ VECTORS ARE NOT COMPUTED FOR',I6,' OF THE SINGULAR
     1VALUES'/)
      NACT = NGOODC - MREJEC
      WRITE(6,950) NGOOD,NTVEC,NACT
  950 FORMAT(/I6,' RITZ VECTORS WERE REQUESTED'/I6,' T-EIGENVECTORS WERE
     1 COMPUTED'/I6,' RITZ VECTORS WILL BE COMPUTED'/)
C     CHECK IF THERE ARE ANY RITZ VECTORS TO COMPUTE
      IF(MREJEC.EQ.NGOODC) GO TO 1780
C
C     CONTINUE WITH THE LANCZOS VECTOR COMPUTATIONS?
      IF(LVCONT.EQ.0.AND.MREJEC.NE.0) GO TO 1760
C
C     NOW COMPUTE THE RITZ VECTORS.  REGENERATE THE
C     LANCZOS VECTORS.
C
      DO 960 I = 1,MNMAX
  960 RITVEC(I) = ZERO
C
C     REGENERATE THE STARTING VECTOR. THIS MUST BE GENERATED AND
C     NORMALIZED PRECISELY THE WAY IT WAS DONE IN THE CORRESPONDING
C     SINGULAR VALUE COMPUTATIONS, OTHERWISE THERE WILL BE A
C     MISMATCH BETWEEN THE T-EIGENVECTORS THAT HAVE BEEN COMPUTED
C     FROM THE T-MATRICES READ IN FROM FILE 2 (IF THEY WERE READ IN)
C     AND THE LANCZOS TRIDIAGONAL MATRICES  THAT ARE BEING REGENERATED.
C
C     STARTING VECTORS ARE OF THE FORM (V1,0) OR (0,V2) WHERE V1 IS
C     OF LENGTH M AND V2 IS OF LENGTH N.  SUCCEEDING LANCZOS VECTORS
C     ALTERNATE BETWEEN THESE TWO FORMS AND THE DIAGONAL ENTRIES OF THE
C     T-MATRICES ALL VANISH.  THE PARAMETER IPAR0 DETERMINES THE SHAPE
C     OF THE STARTING VECTOR.  IF IPAR0=1, THEN STARTING VECTOR WAS
C     OF THE FORM (0,V2).   IF IPAR0=2, THEN STARTING VECTOR WAS OF
C     THE FORM (V1,0).
C     REGENERATE STARTING VECTOR
      BATA = ZERO
      IPAR = IPAR0
      ITNUM = 1
      IF (IPAR.EQ.2) GO TO 1020
C
C-----------------------------------------------------------------------
C     IPAR = 1 SO SET V2 TO RANDOM UNIT VECTOR AND SET V1 = 0.
C
      IIL = SVSEED
      CALL GENRAN(IIL,G,N)
C
C-----------------------------------------------------------------------
C
      DO 970 J = 1,N
```

```
  970 V2(J) = G(J)                                                      LSV09250
C-----------------------------------------------------------------------LSV09260
      SUM = ONE/DSQRT(FINPRO(N,V2,1,V2,1))                              LSV09270
C-----------------------------------------------------------------------LSV09280
C                                                                       LSV09290
      DO 980 J = 1,M                                                    LSV09300
  980 V1(J) = ZERO                                                      LSV09310
C                                                                       LSV09320
      DO 990 J = 1,N                                                    LSV09330
  990 V2(J) = V2(J)*SUM                                                 LSV09340
C                                                                       LSV09350
C     INITIALIZE RITZ VECTORS                                           LSV09360
      DO 1010 J = 1,NGOODC                                              LSV09370
      IF (MP(J).EQ.MPMIN) GO TO 1010                                    LSV09380
      LL = MN*J - N                                                     LSV09390
      II = MINT(J)                                                      LSV09400
      TEMP = TVEC(II)                                                   LSV09410
C                                                                       LSV09420
      DO 1000 K = 1,N                                                   LSV09430
      LL = LL + 1                                                       LSV09440
 1000 RITVEC(LL) = TEMP*V2(K)                                           LSV09450
C                                                                       LSV09460
 1010 CONTINUE                                                          LSV09470
C                                                                       LSV09480
      GO TO 1150                                                        LSV09490
C                                                                       LSV09500
 1020 CONTINUE                                                          LSV09510
C                                                                       LSV09520
C-----------------------------------------------------------------------LSV09530
C     IPAR = 2 SO SET V1 TO RANDOM UNIT VECTOR AND SET V2 = 0.          LSV09540
C                                                                       LSV09550
      CALL GENRAN(SVSEED,G,M)                                           LSV09560
C                                                                       LSV09570
C-----------------------------------------------------------------------LSV09580
C                                                                       LSV09590
      DO 1030 J = 1,M                                                   LSV09600
 1030 V1(J) = G(J)                                                      LSV09610
C-----------------------------------------------------------------------LSV09620
      SUM = ONE/DSQRT(FINPRO(M,V1,1,V1,1))                              LSV09630
C-----------------------------------------------------------------------LSV09640
C                                                                       LSV09650
      DO 1040 J = 1,N                                                   LSV09660
 1040 V2(J) = ZERO                                                      LSV09670
C                                                                       LSV09680
      DO 1050 J = 1,M                                                   LSV09690
 1050 V1(J) = V1(J)*SUM                                                 LSV09700
C                                                                       LSV09710
C     INITIALIZE RITZ VECTORS                                           LSV09720
      DO 1070 J = 1,NGOODC                                              LSV09730
      IF (MP(J).EQ.MPMIN) GO TO 1070                                    LSV09740
      LL = MN*(J-1)                                                     LSV09750
      II = MINT(J)                                                      LSV09760
      TEMP = TVEC(II)                                                   LSV09770
C                                                                       LSV09780
      DO 1060 K = 1,M                                                   LSV09790
```

```
      LL = LL + 1                                                       LSV09800
 1060 RITVEC(LL) = TEMP*V1(K)                                           LSV09810
C                                                                       LSV09820
 1070 CONTINUE                                                          LSV09830
C                                                                       LSV09840
 1080 CONTINUE                                                          LSV09850
C                                                                       LSV09860
C  DO ONE ITERATION OF LANCZOS WHERE NEW LANCZOS VECTOR WILL HAVE THE   LSV09870
C  FORM  (0,V2).                                                        LSV09880
C                                                                       LSV09890
C-----------------------------------------------------------------------LSV09900
C                                                                       LSV09910
      CALL STRAN(V1,V2,BATA)                                            LSV09920
C                                                                       LSV09930
C-----------------------------------------------------------------------LSV09940
C                                                                       LSV09950
C-----------------------------------------------------------------------LSV09960
      BATA = DSQRT(FINPRO(N,V2,1,V2,1))                                 LSV09970
C-----------------------------------------------------------------------LSV09980
      SUM = ONE/BATA                                                    LSV09990
      ITNUM = ITNUM + 1                                                 LSV10000
      IPAR = 2                                                          LSV10010
C                                                                       LSV10020
      TEMP = BETA(ITNUM)                                                LSV10030
      TEMP = DABS(BATA - TEMP)/TEMP                                     LSV10040
      IF (TEMP.LT.1.0D-10) GO TO 1110                                   LSV10050
C                                                                       LSV10060
C     HISTORY MISMATCH ON REGENERATION THUS DEFAULT                     LSV10070
 1090 WRITE(6,1100) ITNUM,IPAR,BATA,BETA(ITNUM),TEMP                    LSV10080
 1100 FORMAT(1X,'ITNUM',2X,'IPAR',16X,'BATA',16X,'BETA',14X,'RELERR'/   LSV10090
     1 2I6,3E20.12/' BATA AND BETA DO NOT AGREE SO PROGRAM STOPS'/)     LSV10100
      GO TO 1860                                                        LSV10110
C                                                                       LSV10120
 1110 CONTINUE                                                          LSV10130
C     NORMALIZE LANCZOS VECTOR                                          LSV10140
      DO 1120 J = 1,N                                                   LSV10150
 1120 V2(J) = V2(J)*SUM                                                 LSV10160
C                                                                       LSV10170
C  UPDATE RITZ VECTORS                                                  LSV10180
      DO 1140 J = 1,NGOODC                                              LSV10190
      IF (IABS(MA(J)).LT.ITNUM.OR.MP(J).EQ.MPMIN) GO TO 1140            LSV10200
      LL = MN*J - N                                                     LSV10210
      II = MINT(J) + ITNUM - 1                                          LSV10220
      TEMP = TVEC(II)                                                   LSV10230
C                                                                       LSV10240
      DO 1130 K = 1,N                                                   LSV10250
      LL = LL + 1                                                       LSV10260
 1130 RITVEC(LL) = TEMP*V2(K) + RITVEC(LL)                              LSV10270
C                                                                       LSV10280
 1140 CONTINUE                                                          LSV10290
C     HAVE ALL REQUIRED LANCZOS VECTORS BEEN REGENERATED ?              LSV10300
C                                                                       LSV10310
      IF(ITNUM.EQ.KMAXU) GO TO 1190                                     LSV10320
C                                                                       LSV10330
 1150 CONTINUE                                                          LSV10340
```

```
C
C     DO ONE ITERATION OF LANCZOS WHERE NEW LANCZOS VECTOR WILL HAVE
C     THE FORM  (V1,0).
C
C-----------------------------------------------------------------------
C
      CALL SVMAT(V2,V1,BATA)
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      BATA = DSQRT(FINPRO(M,V1,1,V1,1))
C-----------------------------------------------------------------------
      SUM = ONE/BATA
      ITNUM = ITNUM + 1
      IPAR = 1
C
      TEMP = BETA(ITNUM)
      TEMP = DABS(BATA - TEMP)/TEMP
      IF (TEMP.GE.1.0D-10) GO TO 1090
C
C     NORMALIZE LANCZOS VECTOR
      DO 1160 J = 1,M
 1160 V1(J) = V1(J)*SUM
C
C     UPDATE RITZ VECTORS
      DO 1180 J = 1,NGOODC
      IF (IABS(MA(J)).LT.ITNUM.OR.MP(J).EQ.MPMIN) GO TO 1180
      LL = MN*(J-1)
      II = MINT(J) + ITNUM - 1
      TEMP = TVEC(II)
C
      DO 1170 K = 1,M
      LL = LL + 1
 1170 RITVEC(LL) = TEMP*V1(K) + RITVEC(LL)
C
 1180 CONTINUE
C  HAVE ALL REQUIRED LANCZOS VECTORS BEEN COMPUTED ?
      IF (ITNUM.LT.KMAXU) GO TO 1080
C
 1190 CONTINUE
C
C     RITZVECTOR GENERATION IS COMPLETE. NORMALIZE EACH RITZVECTOR
C     AS AN EIGENVECTOR OF THE ASSOCIATED SYMMETRIC MATRIX B.
C     THEN COMPUTE THE ERRORS IN THESE VECTORS AS EIGENVECTORS
C     OF B AND WRITE THESE OUT TO FILE 9.  THEN INDIVIDUALLY
C     NORMALIZE THE FIRST M AND THE LAST N COMPONENTS OF EACH OF
C     THESE RITZ VECTORS AND TAKE THESE NORMALIZED VECTORS AS
C     RESPECTIVELY APPROXIMATIONS TO THE LEFT AND TO THE RIGHT
C     SINGULAR VECTORS OF THE CORRESPONDING SINGULAR VALUE OF
C     THE ORIGINAL MATRIX.
C
C
C     NORMALIZE THE RITZ VECTORS AS EIGENVECTORS OF B
      DO 1280 J = 1,NGOODC
```

```
      IF (MP(J).EQ.MPMIN) GO TO 1280                                    LSV10900
      LINT = MN*(J-1) + 1                                               LSV10910
      LFIN = MN*J                                                       LSV10920
      SUM = ZERO                                                        LSV10930
      SVAL = SVNEW(J)                                                   LSV10940
C                                                                       LSV10950
      DO 1200 K = LINT,LFIN                                             LSV10960
 1200 SUM = SUM + RITVEC(K)*RITVEC(K)                                   LSV10970
C                                                                       LSV10980
      SUM = DSQRT(SUM)                                                  LSV10990
      RNORM(J) = SUM                                                    LSV11000
      TEMP = ONE - SUM                                                  LSV11010
      SUM = ONE/SUM                                                     LSV11020
C                                                                       LSV11030
      DO 1210 K = LINT,LFIN                                             LSV11040
 1210 RITVEC(K) = RITVEC(K)*SUM                                         LSV11050
C                                                                       LSV11060
C     COMPUTE ERROR IN RITZ VECTOR CONSIDERED AS AN EIGENVECTOR OF B.   LSV11070
      LINTM = LINT + M                                                  LSV11080
      L = LINT - 1                                                      LSV11090
      DO 1220 K = 1,M                                                   LSV11100
      L = L + 1                                                         LSV11110
 1220 V1(K) = RITVEC(L)                                                 LSV11120
      DO 1230 K = 1,N                                                   LSV11130
      L = L + 1                                                         LSV11140
 1230 V2(K) = RITVEC(L)                                                 LSV11150
C                                                                       LSV11160
C-----------------------------------------------------------------------LSV11170
C                                                                       LSV11180
      CALL SVMAT(RITVEC(LINTM),V1,SVAL)                                 LSV11190
      CALL STRAN(RITVEC(LINT),V2,SVAL)                                  LSV11200
C                                                                       LSV11210
C-----------------------------------------------------------------------LSV11220
C                                                                       LSV11230
      SUM = ZERO                                                        LSV11240
      DO 1240 JJ = 1,M                                                  LSV11250
 1240 SUM = SUM + V1(JJ)*V1(JJ)                                         LSV11260
C                                                                       LSV11270
      DO 1250 JJ = 1,N                                                  LSV11280
 1250 SUM = SUM + V2(JJ)*V2(JJ)                                         LSV11290
C                                                                       LSV11300
      IF(IWRITE.NE.0)  WRITE(6,1260) J,GOODSV(J)                        LSV11310
 1260 FORMAT(/I5,'TH SINGULAR VALUE CONSIDERED =',E20.12/)              LSV11320
C                                                                       LSV11330
      IF(IWRITE.NE.0) WRITE(6,1270) TERR(J), TBETA(J), RNORM(J)         LSV11340
 1270 FORMAT(' RESIDUAL FOR T-EIGENVECTOR = ',E14.3/                    LSV11350
     1' DABS(BETA(MA(J)+1)*U(MA(J)) = ' ,E14.3/                         LSV11360
     1' NORM(RITZVEC) = ', E14.3/)                                      LSV11370
C                                                                       LSV11380
      SUM = DSQRT(SUM)                                                  LSV11390
      BERR(J) = SUM                                                     LSV11400
      BERRGP(J) = SUM/ABS(BMINGP(J))                                    LSV11410
 1280 CONTINUE                                                          LSV11420
C                                                                       LSV11430
C     RITZVECTORS ARE NORMALIZED AND B-MATRIX ESTIMATES ARE IN BERR     LSV11440
```

```
C     AND IN BERRGP ARRAYS.   STORE THESE ESTIMATES BUT NOT THE
C     VECTORS.
C
      WRITE(9,1290)
 1290 FORMAT(11X,'GOODSV(J)',3X,'MA(J)',2X,'  BMINGAP',6X,' BERROR',2X,
     1 'BERROR/BGAP',4X,' TERROR')
C
      WRITE(13,1300)
 1300 FORMAT(11X,'GOODSV(J)',5X,'RITZNORM',5X,' BMINGAP',5X,'TBETA(J)',
     1 5X,'TLAST(J)')
C
      DO 1330 J=1,NGOODC
C
      IF(MP(J).EQ.MPMIN) GO TO 1330
C
      WRITE(9,1310)SVNEW(J),MA(J),BMINGP(J),BERR(J),BERRGP(J),TERR(J)
 1310 FORMAT(E20.12,I6,4E13.5)
C
      WRITE(13,1320) SVNEW(J),RNORM(J),BMINGP(J),TBETA(J),TLAST(J)
 1320 FORMAT(E20.12,4E13.5)
C
 1330 CONTINUE
C
      IF (MREJEC.EQ.0) GO TO 1410
C
      WRITE(9,1340)
 1340 FORMAT(/' RITZ VECTORS WERE NOT COMPUTED FOR THE FOLLOWING SINGULA
     1R VALUES'/' EITHER BECAUSE THEY HAD NOT CONVERGED OR BECAUSE THE E
     1RROR ESTIMATE'/'  WAS NOT AS SMALL AS DESIRED'/)
C
      WRITE(13,1350)
 1350 FORMAT(/' RITZ VECTORS WERE NOT COMPUTED FOR THE FOLLOWING SINGULA
     1R VALUES'/' EITHER BECAUSE THEY HAD NOT CONVERGED OR BECAUSE THE E
     1ERROR ESTIMATE'/' WAS NOT AS SMALL AS DESIRED'/)
C
      DO 1400 J = 1,NGOODC
      IF(MP(J).NE.MPMIN) GO TO 1400
C     EACH SINGULAR VALUE FOR WHICH NO SINGULAR VECTOR WAS CALCULATED
C     HAS INFORMATION OUTPUTTED TO FILES 4 AND 13
C
      WRITE(9,1360)
 1360 FORMAT(6X,'GOODSV(J)',3X,'MA(J)',5X,'BMINGP(J)',6X,'TLAST(J)',
     1 6X,'TBETA(J)',3X,'MP(J)')
      WRITE(9,1370) GOODSV(J),MA(J),BMINGP(J),TLAST(J),TBETA(J),MP(J)
 1370 FORMAT(E15.8,I8,3E14.4,I8)
C
      WRITE(13,1380)
 1380 FORMAT(6X,'GOODSV(J)',3X,'MA(J)',3X,'M1(J)',3X,'M2(J)',3X,'MP(J)'
     1/)
      WRITE(13,1390) GOODSV(J),MA(J),M1(J),M2(J),MP(J)
 1390 FORMAT(E15.8,4I8)
C
 1400 CONTINUE
C
 1410 CONTINUE
```

```
C
      WRITE(9,1420)
 1420 FORMAT(/' ABOVE ARE ERROR ESTIMATES FOR THE B AND T EIGENVECTORS'/
     1 ' ASSOCIATED WITH THE GOODSV LISTED IN COLUMN 1'/
     1 ' BERROR = NORM(B*X - SV*X)  TERROR = NORM(T*Y - SV*Y) '/
     1 ' WHERE T = T(1,MA(J))    X = RITZ VECTOR = V*Y  V = SUCCESSIVE'/
     1 ' LANCZOS VECTORS. BMINGAP = GAP TO NEAREST B-EIGENVALUE'//)
C
      WRITE(13,1430)
 1430 FORMAT(/' ABOVE ARE ERROR ESTIMATES FOR THE GOODSVS'/
     1 ' RITZNORM = NORM(COMPUTED RITZ VECTOR FOR B-MATRIX'/
     1 ' TBETA(J) = BETA(MA(J)+1)*Y(MA(J)),  T*Y = SV*Y '/
     1 ' TLAST(J) = DABS(Y(MA(J)) '/)
C
C     NUMBER OF RITZ VECTORS COMPUTED
      NCOMPU = NGOODC - MREJEC
      WRITE(12,1440) N,NCOMPU,NGOODC,MATNO
 1440 FORMAT(3I6,I12,' SIZE A, NO.RITZVECS, NO.SVALUES,MATNO')
C
C     INDIVIDUALLY NORMALIZE THE FIRST M AND THE LAST N COMPONENTS OF
C     EACH RITZ VECTOR.
C
      LFIN = 0
      DO 1560 J = 1,NGOODC
C
      IF(MP(J).EQ.MPMIN) GO TO 1540
C
C     RITZ VECTOR WAS COMPUTED
      LINT = MN*(J-1) + 1
      LFIN = MN*J
      LFIN1 = LINT + M - 1
      LINT1 = LFIN1 + 1
C
      SUM = 0.D0
      TEMP = 0.D0
      DO 1450 I = LINT,LFIN1
 1450 SUM = SUM + RITVEC(I)*RITVEC(I)
      SUM = ONE/DSQRT(SUM)
      DO 1460 I = LINT,LFIN1
 1460 RITVEC(I) = SUM*RITVEC(I)
      DO 1470 I = LINT1,LFIN
 1470 TEMP = TEMP + RITVEC(I)*RITVEC(I)
      TEMP = ONE/DSQRT(TEMP)
      DO 1480 I = LINT1,LFIN
 1480 RITVEC(I) = TEMP*RITVEC(I)
C
      WRITE(12,1490) J, GOODSV(J), MP(J)
 1490 FORMAT(/I6,4X,E20.12,I6,' J, SINGULAR VALUE, MP(J)')
C
      WRITE(12,1500) BERR(J),BERRGP(J)
 1500 FORMAT(2E15.5,' = NORM(B*Z-SVAL*Z) AND  NORM(B*Z-SVAL*Z)/BMINGAP')
C
      WRITE(12,1510) J
 1510 FORMAT(/I6,'TH LEFT SINGULAR VECTOR'/)
C     WRITE(12,170) (RITVEC(LL), LL=LINT,LFIN1)
```

```
      WRITE(12,1520) (RITVEC(LL), LL=LINT,LFIN1)                        LSV12550
 1520 FORMAT(4E20.12)                                                   LSV12560
C                                                                       LSV12570
      WRITE(12,1530) J                                                  LSV12580
 1530 FORMAT(/I6,'TH RIGHT SINGULAR VECTOR'/)                           LSV12590
C     WRITE(12,170) (RITVEC(LL), LL=LINT1,LFIN)                         LSV12600
      WRITE(12,1520) (RITVEC(LL), LL=LINT1,LFIN)                        LSV12610
C                                                                       LSV12620
      GO TO 1560                                                        LSV12630
C                                                                       LSV12640
C     NO RITZ VECTOR WAS COMPUTED FOR THIS SINGULAR VALUE               LSV12650
 1540 WRITE(12,1550) J,GOODSV(J),MP(J)                                  LSV12660
 1550 FORMAT(I6,4X,E20.12,I6,' J,SINGVALUE,MP(J),NO RITZ VECTOR COMPUTEDLSV12670
     1')                                                                LSV12680
C                                                                       LSV12690
 1560 CONTINUE                                                          LSV12700
C                                                                       LSV12710
C     DID ANY T-MATRICES INCLUDE OFF-DIAGONAL ENTRIES SMALLER THAN      LSV12720
C     DESIRED, AS SPECIFIED BY BTOL?                                    LSV12730
C                                                                       LSV12740
      IF(IB.GT.0) GO TO 1590                                            LSV12750
      WRITE(6,1570) KMAXU                                               LSV12760
 1570 FORMAT(/' FOR LARGEST T-MATRIX CONSIDERED',I7,' CHECK THE SIZE OF LSV12770
     1BETAS')                                                           LSV12780
C                                                                       LSV12790
C-----------------------------------------------------------------------LSV12800
C                                                                       LSV12810
      CALL TNORM(BETA,BKMIN,TEMP,KMAXU,IBMT)                            LSV12820
C                                                                       LSV12830
C-----------------------------------------------------------------------LSV12840
C                                                                       LSV12850
      IF(IBMT.LT.0) WRITE (6,1580)                                      LSV12860
 1580 FORMAT(/' WARNING THE T-MATRICES FOR ONE OR MORE OF THE SINGULAR VLSV12870
     1ALUES CONSIDERED'/' HAD AN OFF-DIAGONAL ENTRY THAT WAS SMALLER THALSV12880
     1N THE BETA TOLERANCE THAT WAS SPECIFIED'/)                        LSV12890
 1590 CONTINUE                                                          LSV12900
C                                                                       LSV12910
       GO TO 1860                                                       LSV12920
C                                                                       LSV12930
 1600 WRITE(6,1610) NGOOD,MNMAX,MDIMRV                                  LSV12940
 1610 FORMAT(/I4,' RITZ VECTORS WERE REQUESTED BUT THE REQUIRED DIMENSIOLSV12950
     1N',I6/' IS LARGER THAN THE USER-SPECIFIED DIMENSION OF RITVEC',I6 LSV12960
     1/' THEREFORE, THE SINGULAR VECTOR PROCEDURE TERMINATES FOR THE USELSV12970
     1R TO INTERVENE')                                                  LSV12980
C                                                                       LSV12990
      GO TO 1860                                                        LSV13000
C                                                                       LSV13010
 1620 WRITE(6,1630)  MOLD,M,NOLD,N,MATOLD,MATNO                         LSV13020
 1630 FORMAT(/' GOODSV PARAMETERS READ FROM FILE 3 DO NOT AGREE WITH THOLSV13030
     1SE'/' SPECIFIED BY THE USER. MOLD,M,NOLD,N,MATOLD,MATNO ='/       LSV13040
     14I6, 2I12/' THEREFORE PROGRAM TERMINATES FOR USER TO RESOLVE DIFFELSV13050
     1RENCES'/)                                                         LSV13060
C                                                                       LSV13070
      GO TO 1860                                                        LSV13080
C                                                                       LSV13090
```

```
 1640 WRITE(6,1650)
 1650 FORMAT(/' PARAMETERS IN BETA FILE DO NOT AGREE WITH THOSE SPECIFIE
     1D BY THE USER.'/' THEREFORE, THE PROGRAM TERMINATES FOR THE USER T
     1O RESOLVE THE DIFFERENCES'/)
C
      GO TO 1860
C
 1660 WRITE(6,1670) KMAX,MEV
 1670 FORMAT(/' IN BETA HISTORY HEADER KMAX =',I6/
     1' BUT SINGULAR VALUES WERE COMPUTED AT MEV = ',I6,' PROGRAM STOPS'
     1)
C
      GO TO 1860
C
 1680 WRITE(6,1690) MEV
 1690 FORMAT(/' SOMETHING IS WRONG.'/' HEADER SAYS THAT SIZE T-MATRIX US
     1ED IN THE SINGULAR VALUE COMPUTATIONS WAS = ',I6/' BUT THIS IS AN 
     1ODD ORDER AND THAT IS NOT ALLOWED.  PROGRAM STOPS'/)
C
      GO TO 1860
C
 1700 WRITE(6,1710)
 1710 FORMAT(/' PROGRAM COMPUTED 1ST GUESSES AT T-MATRIX SIZES, READ THE
     1M TO FILE 10'/' THEN TERMINATED AS REQUESTED.')
      GO TO 1860
C
 1720 WRITE(6,1730) MTOL, MDIMTV
 1730 FORMAT(/' PROGRAM TERMINATES BECAUSE THE TVEC DIMENSION ANTICIPATE
     1D',I7/'  IS LARGER THAN THE TVEC DIMENSION',I7,' SPECIFIED BY THE 
     1USER.'/'  USER MAY RESET THE TVEC DIMENSION AND RESTART THE PROGRA
     1M'/)
      GO TO 1860
C
 1740 WRITE(6,1750)
 1750 FORMAT(/' PROGRAM TERMINATES BECAUSE NO SUITABLE T-EIGENVECTORS WE
     1RE IDENTIFIED'/' FOR ANY OF THE SINGULAR VALUES SUPPLIED.  PROBLEM
     1 COULD BE CAUSED BY'/' TOO SMALL A TVEC DIMENSION OR SIMPLY BE THA
     1T NO SUITABLE T-VECTORS'/' WERE IDENTIFIED.  USER SHOULD CHECK OUT
     1PUT'/)
      GO TO 1860
C
 1760 WRITE(6,1770) LVCONT,NTVEC,NGOOD
 1770 FORMAT(/' LVCONT FLAG =',I2,' AND NUMBER ',I5,' OF T-EIGENVECTORS 
     1 COMPUTED N.E.'/' NUMBER',I5,' REQUESTED SO PROGRAM TERMINATES'/)
      GO TO 1860
 1780 WRITE(6,1790)
 1790 FORMAT(/' PROGRAM TERMINATES WITHOUT COMPUTING ANY RITZ VECTORS'/ 
     1/' BECAUSE ALL OF THE T-EIGENVECTORS WERE REJECTED AS NOT SUITABLE
     1 FOR'/' THE RITZ VECTOR COMPUTATIONS.  PROBABLE CAUSE WAS LACK OF 
     1CONVERGENCE'/' OF THE SINGULAR VALUES'/)
       GO TO 1860
C
 1800 WRITE(6,1810)
 1810 FORMAT(/' PROGRAM INDICATES THAT IT IS NOT POSSIBLE TO COMPUTE ANY
     1 OF THE'/' REQUESTED EIGENVECTORS. THEREFORE PROGRAM TERMINATES')
```

```
      DO 1820 J=1,NGOODC                                                LSV13650
 1820 WRITE(6,1830)  J,GOODSV(J),MP(J)                                  LSV13660
 1830 FORMAT(/4X,' J',11X,'GOODSV(J)',4X,'MP(J)'/I6,E20.12,I9)          LSV13670
      GO TO 1860                                                        LSV13680
C                                                                       LSV13690
 1840 WRITE(6,1850) MBETA,KMAXN                                         LSV13700
 1850 FORMAT(/' PROGRAM TERMINATES BECAUSE THE STORAGE ALLOTTED FOR THE LSV13710
     1BETA ARRAY',I8/' IS NOT SUFFICIENT FOR THE ENLARGED KMAX =',I8,' TLSV13720
     1HAT THE PROGRAM WANTS'/' USER CAN ENLARGE THE BETA ARRAY AND RERUNLSV13730
     1 THE PROGRAM'/)                                                   LSV13740
C                                                                       LSV13750
 1860 CONTINUE                                                          LSV13760
C                                                                       LSV13770
      STOP                                                              LSV13780
C-----END OF MAIN PROGRAM FOR LANCZOS SINGULAR VECTOR COMPUTATIONS------LSV13790
      END                                                               LSV13800
```

6.5 LSMULT: LANCZS and Sample Matrix-Vector Multiply Subroutines

```
C-----LSMULT-----------------------------------------------------------
C  Authors:  Jane Cullum and Ralph A. Willoughby (Deceased)
C            Los Alamos National Laboratory
C            Los Alamos, New Mexico 87544
C
C            E-mail:  cullumj@lanl.gov
C
C  These codes are copyrighted by the authors.  These codes
C  and modifications of them or portions of them are NOT to be
C  incorporated into any commercial codes or used for any other
C  commercial purposes such as consulting for other companies,
C  without legal agreements with the authors of these Codes.
C  If these Codes or portions of them are used in other scientific or
C  engineering research works the names of the authors of these codes
C  and appropriate references to their written work are to be
C  incorporated in the derivative works.
C
C  This header is not to be removed from these codes.
C
C         REFERENCE: Cullum and Willoughby, Chapter 5
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 SUBROUTINES LANCZS, USPECS, STRAN, AND SVMAT
C     FOR USE WITH THE LANCZOS SINGULAR VALUE/VECTOR PROGRAMS
C
C     NONPORTABLE CONSTRUCTIONS:
C     1.  THE ENTRY MECHANISM USED TO PASS THE STORAGE LOCATIONS
C         OF THE USER-SPECIFIED MATRIX FROM THE SUBROUTINE USPEC
C         TO THE MATRIX-VECTOR MULTIPLY SUBROUTINES SVMAT AND
C         STRAN.
C     2.  IN THE SAMPLE USPEC PROVIDED:  THE FREE FORMAT (8,*),
C         AND THE FORMAT (20A4).
C
C-----START OF LANCZS--------------------------------------------------
C
      SUBROUTINE LANCZS(MATVEC,MTRAN,BETA,V1,V2,G,KMAX,MOLD1,
     1                  M,N,IPAR,IIX)
C
C----------------------------------------------------------------------
      DOUBLE PRECISION BETA(1),V1(1),V2(1),SUM,TEMP,ONE,ZERO
      REAL G(1)
      DOUBLE PRECISION FINPRO
      INTEGER IPAR
      EXTERNAL MATVEC,MTRAN
C----------------------------------------------------------------------
```

```
C     COMPUTE T(1,MEV) FOR SYMMETRIZED VERSION OF GIVEN A-MATRIX.
C
C                               ----                 ----
C                              |  0                  A  |
C                B  =          |                        |
C                              |  A-TRANSPOSE        0  |
C                               ----                 ----
C
C     WHERE A IS AN M BY N REAL SPARSE MATRIX, USING STARTING
C     VECTORS OF THE FORM (V1,0) WHEN THE FLAG IPAR = 2 AND
C     OF THE FORM (0,V2) WHEN THE FLAG IPAR = 1.  V1 IS OF
C     DIMENSION M, THE ROW DIMENSION OF A, AND V2 IS OF DIMENSION
C     N, THE COLUMN DIMENSION OF A.
C
C     WITH STARTING VECTORS OF THESE FORMS, THE LANCZOS VECTORS
C     GENERATED ALTERNATE BETWEEN THESE 2 FORMS AND ALL OF THE
C     DIAGONAL ENTRIES OF THE LANCZOS TRIDIAGONAL MATRICES T(1,MEV)
C     GENERATED ARE 0.
C
C     LANCZS USES 2 USER-SUPPLIED SUBROUTINES MATVEC AND MTRAN.
C     MAIN PROGRAM CALLS THESE SVMAT AND STRAN, RESPECTIVELY.
C     CALLING SEQUENCES ARE
C
C           CALL MATVEC(V2,V1,SUM)
C           CALL MTRAN(V1,V2,SUM)
C
C     MATVEC  COMPUTES V1 = A*V2 - SUM*V1.
C     MTRAN COMPUTES V2 = (A-TRANSPOSE)*V1 - SUM*V2.
C
C     ON EXIT V1 AND V2 CONTAIN THE NONZERO PARTS OF THE
C     LAST TWO LANCZOS VECTORS.
C
C     IF MOLD1 = 1 THEN T(1,KMAX) IS GENERATED FROM SCRATCH.
C     IF MOLD1 > 1 THEN A PREVIOUSLY-GENERATED T-MATRIX OF SIZE
C     (MOLD1-1) IS EXTENDED TO ONE OF SIZE KMAX. SINGULAR VALUE
C     PRGORAMS CAN ONLY UTILIZE T-MATRICES OF EVEN ORDER.
C     BETA(KMAX+1) IS ALSO COMPUTED FOR USE IN THE ERROR ESTIMATES.
C
C-----------------------------------------------------------------------
      ONE  = 1.0D0
      ZERO = 0.0D0
      ITNUM = MOLD1
C
      IF (ITNUM .GT. 1) GO TO (80,100), IPAR
C
C     NO PREVIOUS BETA HISTORY
      BETA(1) = ZERO
      IIL = IIX
      IF (IPAR .EQ. 2) GO TO 40
C
C-----------------------------------------------------------------------
C     IPAR = 1 SO SET V2 EQUAL TO A UNIT RANDOM VECTOR AND SET V1 = 0.
      CALL GENRAN(IIL,G,N)
C-----------------------------------------------------------------------
C
```

```
      DO 10 J = 1,N                                                     LSM00990
   10 V2(J) = G(J)                                                      LSM01000
C                                                                       LSM01010
C-----------------------------------------------------------------------LSM01020
      TEMP = FINPRO(N,V2(1),1,V2(1),1)                                  LSM01030
C-----------------------------------------------------------------------LSM01040
C                                                                       LSM01050
      SUM = ONE/DSQRT(TEMP)                                             LSM01060
      DO 20 J = 1,M                                                     LSM01070
   20 V1(J) = ZERO                                                      LSM01080
C                                                                       LSM01090
      DO 30 J = 1,N                                                     LSM01100
   30 V2(J) = V2(J)*SUM                                                 LSM01110
      GO TO 100                                                         LSM01120
C                                                                       LSM01130
   40 CONTINUE                                                          LSM01140
C                                                                       LSM01150
C-----------------------------------------------------------------------LSM01160
C     IPAR = 2 SO SET V1 EQUAL TO A UNIT RANDOM VECTOR AND SET V2 = 0.  LSM01170
      CALL GENRAN(IIL,G,M)                                              LSM01180
C-----------------------------------------------------------------------LSM01190
C                                                                       LSM01200
      DO 50 J=1,M                                                       LSM01210
   50 V1(J) = G(J)                                                      LSM01220
C                                                                       LSM01230
C-----------------------------------------------------------------------LSM01240
      TEMP = FINPRO(M,V1(1),1,V1(1),1)                                  LSM01250
C-----------------------------------------------------------------------LSM01260
C                                                                       LSM01270
      SUM = ONE/DSQRT(TEMP)                                             LSM01280
      DO 60 J = 1,N                                                     LSM01290
   60 V2(J) = ZERO                                                      LSM01300
      DO 70 J = 1,M                                                     LSM01310
   70 V1(J) = V1(J)*SUM                                                 LSM01320
C                                                                       LSM01330
C     BELOW IS START FOR MOLD1 > 1 AND IPAR = 1                         LSM01340
C     DO ONE ITERATION OF LANCZOS TO OBTAIN (0,V2)                      LSM01350
C                                                                       LSM01360
   80 CONTINUE                                                          LSM01370
      SUM = BETA(ITNUM)                                                 LSM01380
C                                                                       LSM01390
C-----------------------------------------------------------------------LSM01400
      CALL MTRAN(V1,V2,SUM)                                             LSM01410
C-----------------------------------------------------------------------LSM01420
C                                                                       LSM01430
C-----------------------------------------------------------------------LSM01440
      SUM = FINPRO(N,V2(1),1,V2(1),1)                                   LSM01450
C-----------------------------------------------------------------------LSM01460
C                                                                       LSM01470
      ITNUM = ITNUM + 1                                                 LSM01480
      BETA(ITNUM) = DSQRT(SUM)                                          LSM01490
      SUM = ONE/BETA(ITNUM)                                             LSM01500
C                                                                       LSM01510
      DO 90 J = 1,N                                                     LSM01520
   90 V2(J) = V2(J)*SUM                                                 LSM01530
```

```
C
      IPAR = 2
      IF (ITNUM .GT. KMAX) GO TO 120
C
C     BELOW IS START FOR MOLD1 > 1 AND IPAR = 2
C     DO ONE ITERATION OF LANCZOS TO OBTAIN (V1,0)
C
  100 CONTINUE
      SUM = BETA(ITNUM)
C
C-----------------------------------------------------------------------
      CALL MATVEC(V2,V1,SUM)
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
      SUM = FINPRO(M,V1(1),1,V1(1),1)
C-----------------------------------------------------------------------
C
      ITNUM = ITNUM + 1
      BETA(ITNUM) = DSQRT(SUM)
      SUM = ONE/BETA(ITNUM)
C
      DO 110 J = 1,M
  110 V1(J)= V1(J) * SUM
C
      IPAR = 1
      IF (ITNUM .GT. KMAX) GO TO 120
      GO TO 80
C
  120 CONTINUE
C
      RETURN
C-----END OF LANCZS-----------------------------------------------------
      END
C
C-----START OF USPEC (GENERAL SPARSE, RECTANGULAR MATRIX)---------------
C
C     SUBROUTINE USPEC(M,N,MATNO)
      SUBROUTINE SUSPEC(M,N,MATNO)
C
C--------------------------------------------------------------------
      DOUBLE PRECISION A(10000)
      INTEGER IROW(10000),ICOL(3010)
C--------------------------------------------------------------------
C    DIMENSIONS ARRAYS NEEDED TO DEFINE THE USER-SUPPLIED
C    M X N RECTANGULAR A-MATRIX, READS IN VALUES OF THESE
C    ARRAYS AND THEN PASSES THE STORAGE LOCATIONS OF THESE
C    ARRAYS TO THE CORRESPONDING MATRIX-VECTOR MULTIPLY
C    SUBROUTINES SVMAT AND STRAN.
C
C    THE A-MATRIX IS STORED IN THE FOLLOWING SPARSE FORMAT:
C    M = NUMBER OF ROWS IN A.
C    N = NUMBER OF COLUMNS IN A.
C    NZ = NUMBER OF NONZERO ENTRIES IN A-MATRIX.
C    ICOL(J), J=1,N IS NUMBER OF NONZERO ENTRIES IN COLUMN J.
```

```
C    IROW(K), K = 1,NZ IS THE ROW INDEX FOR CORRESPONDING A(K).
C    A(K), K=1,NZ IS NONZERO ENTRIES IN A, COLUMN BY COLUMN.
C    IT IS ASSUMED THAT ICOL(J) > 0 FOR ALL J
C
C     NOTE:  ASSOCIATED SUBROUTINES SVMAT AND STRAN ASSUME THAT
C             M >= N.
C
C-----------------------------------------------------------------------
C     READ IN  MATRIX FROM  FILE 8
C
      READ(8,10) NZ,MOLD,NOLD,MATOLD
   10 FORMAT(I10,2I6,I8)
C
      WRITE(6,20) NZ,MOLD,NOLD,MATOLD
   20 FORMAT(6X,'NZ',4X,'MOLD',4X,'NOLD',4X,'MATOLD'/I10,2I6,I10/)
C
C     TEST OF PARAMETER CORRECTNESS
      ITEMP = (MOLD-M)**2 + (NOLD-N)**2 + (MATOLD-MATNO)**2
C
      IF (ITEMP.EQ.0) GO TO 40
C
      WRITE(6,30)
   30 FORMAT(' PROGRAM TERMINATES BECAUSE EITHER ORDERS OF OR LABELS FOR
     1 MATRIX DISAGREE')
      GO TO 70
C
   40 CONTINUE
C
C     NUMBER OF NONZERO ENTRIES IN EACH COLUMN IS READ IN
C     THEN THE CORRESPONDING ROW INDEX FOR EACH SUCH ENTRY IS READ
      READ(8,50) (ICOL(K), K=1,N)
      READ(8,50) (IROW(K), K=1,NZ)
   50 FORMAT(13I6)
C
C     READ IN THE NONZERO ENTRIES IN THE MATRIX
      READ(8,60) (A(K), K=1,NZ)
   60 FORMAT(3E25.16)
C  50 FORMAT(4E19.10)
C
C-----------------------------------------------------------------------
C     PASS STORAGE LOCATIONS OF ARRAYS THAT DEFINE THE MATRIX TO
C     THE MATRIX-VECTOR MULTIPLY SUBROUTINES SVMAT AND STRAN
      CALL SMATVE(A,ICOL,IROW,M,N)
      CALL STRANE(A,ICOL,IROW,M,N)
C-----------------------------------------------------------------------
C
C-----END OF USPEC------------------------------------------------------
      RETURN
   70 STOP
      END
C
C-----STRAN (GENERAL SPARSE MATRIX)-------------------------------------
C
C     SUBROUTINE STRAN(W,U,SUM)
      SUBROUTINE SSTRAN(W,U,SUM)
```

```
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION W(1),U(1),A(1),SUM,TEMP
      INTEGER IROW(1),ICOL(1)
C-----------------------------------------------------------------------
C     SUBROUTINE TO COMPUTE U = (A-TRANSPOSE)*W - SUM*U WHERE A IS
C     A GENERAL, SPARSE M X N MATRIX WITH M >= N.
C
C     ASSUMES MATRIX IS STORED IN SPARSE FORMAT GIVEN IN
C     CORRESPONDING USPEC SUBROUTINE.
C-----------------------------------------------------------------------
      JLAST = 0
      DO 20 J = 1,N
      JFIRST = JLAST + 1
      JLAST  = JLAST + ICOL(J)
      TEMP   = -SUM*U(J)
C
      DO 10 K = JFIRST,JLAST
      IK = IROW(K)
   10 TEMP = A(K)*W(IK) + TEMP
C
   20 U(J) = TEMP
C
      RETURN
C
C-----------------------------------------------------------------------
      ENTRY STRANE(A,ICOL,IROW,M,N)
C-----------------------------------------------------------------------
C
C-----END OF STRAN FOR GENERAL SPARSE MATRIX----------------------------
      RETURN
      END
C
C-----SVMAT (GENERAL SPARSE MATRIX)-------------------------------------
C
C     SUBROUTINE SVMAT(W,U,SUM)
      SUBROUTINE SSVMAT(W,U,SUM)
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION  W(1),U(1),A(1),SUM,TEMP
      INTEGER IROW(1),ICOL(1)
C-----------------------------------------------------------------------
C     SUBROUTINE TO COMPUTE U = A*W - SUM*U WHERE A IS A
C     GENERAL, SPARSE M X N MATRIX WITH M >= N.
C
C     ASSUMES THAT THE MATRIX IS STORED IN THE SPARSE FORMAT
C     GIVEN IN THE CORRESPONDING USPEC SUBROUTINE.
C-----------------------------------------------------------------------
      DO 10 I = 1,M
   10 U(I) = -SUM*U(I)
C
C  MAIN LOOP.  PROCESSING PROCEEDS COL BY COL.  JFIRST AND JLAST ARE
C  POINTERS TO THE FIRST AND LAST NONZEROS IN COLUMN J.
C
      JLAST = 0
```

```
      DO 30 J = 1,N                                                     LSM03190
      JFIRST = JLAST + 1                                                LSM03200
      JLAST = JLAST + ICOL(J)                                           LSM03210
      TEMP = W(J)                                                       LSM03220
C                                                                       LSM03230
      DO 20 K = JFIRST,JLAST                                            LSM03240
      IK = IROW(K)                                                      LSM03250
   20 U(IK) = U(IK) + A(K)*TEMP                                         LSM03260
C                                                                       LSM03270
   30 CONTINUE                                                          LSM03280
C                                                                       LSM03290
      RETURN                                                            LSM03300
C                                                                       LSM03310
C-----------------------------------------------------------------------LSM03320
      ENTRY SMATVE(A,ICOL,IROW,M,N)                                     LSM03330
C-----------------------------------------------------------------------LSM03340
C                                                                       LSM03350
C----END OF SVMAT FOR GENERAL SPARSE MATRICES---------------------------LSM03360
      RETURN                                                            LSM03370
      END                                                               LSM03380
C                                                                       LSM03390
C-----ROUTINES FOR 'DIAGONAL' TEST MATRICES-----------------------------LSM03400
C     DMATV,DMTRAN,DIAGSP SUBROUTINES ARE FOR RECTANGULAR DIAGONAL      LSM03410
C     TEST MATRICES.                                                    LSM03420
C                                                                       LSM03430
C-----START OF USPEC FOR 'DIAGONAL' TEST MATRIX-------------------------LSM03440
C                                                                       LSM03450
      SUBROUTINE USPEC(M,N,MATNO)                                       LSM03460
C     SUBROUTINE DIAGSP(M,N,MATNO)                                      LSM03470
C                                                                       LSM03480
C     DEFINES 'DIAGONAL' MATRIX OF FOLLOWING FORM                       LSM03490
C                                                                       LSM03500
C                          -----         -----                          LSM03510
C                         | 0       0       D |                         LSM03520
C                 A =     | 0       0       0 |                         LSM03530
C                         |D-TRANS  0       0 |                         LSM03540
C                          -----         -----                          LSM03550
C                                                                       LSM03560
C     WHERE D IS DIAGONAL MATRIX OF ORDER N, AND IN THE                 LSM03570
C     MIDDLE THERE ARE (M-N) ROWS OF ZEROES.                            LSM03580
C     CALLS ENTRY TO MATRIX-VECTOR MULTIPLY SUBROUTINE TO PASS          LSM03590
C     STORAGE LOCATION OF THE D-ARRAY AND THE ORDERS M AND N.           LSM03600
C                                                                       LSM03610
C     NOTE:  ASSOCIATED MATRIX-VECTOR SUBROUTINES ASSUME THAT           LSM03620
C            M >= N.                                                    LSM03630
C-----------------------------------------------------------------------LSM03640
      DOUBLE PRECISION  D(1000), SPACE                                  LSM03650
      REAL  EXPLAN(20)                                                  LSM03660
C-----------------------------------------------------------------------LSM03670
C                                                                       LSM03680
      READ(8,10) EXPLAN                                                 LSM03690
   10 FORMAT(20A4)                                                      LSM03700
      READ(8,*) MOLD,NOLD,NUNIF,SPACE,D(1)                              LSM03710
C                                                                       LSM03720
      IF(N.NE.NOLD.OR.M.NE.MOLD) GO TO 80                               LSM03730
```

```
C     COMPUTE THE UNIFORM PORTION OF THE SPECTRUM                       LSM03740
      DO 20 J=2,NUNIF                                                   LSM03750
   20 D(J) = D(1) - DFLOAT(J-1)*SPACE                                   LSM03760
      NUNIF1=NUNIF + 1                                                  LSM03770
      READ(8,10)  EXPLAN                                                LSM03780
      DO 30 J=NUNIF1,N                                                  LSM03790
   30 READ(8,*) D(J)                                                    LSM03800
      NNUNIF = NOLD - NUNIF                                             LSM03810
      WRITE(6,40) NOLD,SPACE,NNUNIF,D(1)                                LSM03820
   40 FORMAT(/' DIAGONAL TEST MATRIX, SIZE = ',I4/' MOST ENTRIES ARE ', LSM03830
     1E10.3,' UNITS APART.',I3,' ENTRIES'/' ARE IRREGULARLY SPACED. FIRSLSM03840
     1T ENTRY IS ',E10.3/)                                              LSM03850
      NB = NUNIF - 2                                                    LSM03860
C                                                                       LSM03870
C                                                                       LSM03880
C     PRINT OUT DIAGONAL PORTION OF A-MATRIX                            LSM03890
      WRITE(6,50) (D(I), I=1,10 )                                       LSM03900
      WRITE(6,60) (D(I), I = NB,N)                                      LSM03910
      MNDIF = MOLD - NOLD                                               LSM03920
      IF(MNDIF.NE.0) WRITE(6,70) MNDIF                                  LSM03930
   50 FORMAT(/' SINGULAR VALUE LANCZOS TEST, 1ST 10 ENTRIES OF DIAGONAL LSM03940
     1A-MATRIX = '/(3E22.14))                                           LSM03950
   60 FORMAT(/' MIDDLE UNIFORM PORTION OF MATRIX IS NOT PRINTED OUT'/   LSM03960
     1' END OF UNIFORM PLUS NONUNIFORM SECTION = '/(3E22.14))           LSM03970
   70 FORMAT(I4,' ZERO ROWS ARE ADDED TO THE DIAGONAL TO MAKE IT RECTANGLSM03980
     1ULAR'/)                                                           LSM03990
C                                                                       LSM04000
C     DIAGONAL GENERATION COMPLETE                                      LSM04010
C                                                                       LSM04020
C-----------------------------------------------------------------------LSM04030
C     CALL ENTRY TO MATRIX-VECTOR MULTIPLY SUBROUTINES  TO PASS         LSM04040
C     STORAGE LOCATION OF D-ARRAY AND ORDER OF A-MATRIX.                LSM04050
      CALL DMATVE(D,M,N)                                                LSM04060
      CALL DMTRAE(D,M,N)                                                LSM04070
C-----------------------------------------------------------------------LSM04080
C                                                                       LSM04090
      RETURN                                                            LSM04100
   80 WRITE(6,90) MOLD,NOLD,M,N                                         LSM04110
   90 FORMAT(' PROGRAM TERMINATES MOLD=',I5,' N.E. M=',I5,' OR NOLD=',  LSM04120
     1I5,' N.E. N=',I5)                                                 LSM04130
C-----END OF USPEC SUBROUTINE FOR 'DIAGONAL' TEST MATRICES--------------LSM04140
      STOP                                                              LSM04150
      END                                                               LSM04160
C                                                                       LSM04170
C-----DSVMAT ('DIAGONAL' TEST MATRICES)---------------------------------LSM04180
C                                                                       LSM04190
C     SUBROUTINE DSVMAT(Z,W,SUM)                                        LSM04200
      SUBROUTINE SVMAT(Z,W,SUM)                                         LSM04210
C                                                                       LSM04220
C-----------------------------------------------------------------------LSM04230
      DOUBLE PRECISION  A(1),Z(1),W(1),SUM                              LSM04240
C-----------------------------------------------------------------------LSM04250
C                                                                       LSM04260
C     COMPUTES W = A*Z - SUM*W .  ASSUMES THAT M >= N.                  LSM04270
      DO 10 I = 1,N                                                     LSM04280
```

```
   10 W(I) = A(I)*Z(I) - SUM *W(I)                                      LSM04290
      IF(M.EQ.N) RETURN                                                 LSM04300
      N1 = N+1                                                          LSM04310
      DO 20 I = N1,M                                                    LSM04320
   20 W(I) = -SUM*W(I)                                                  LSM04330
      RETURN                                                            LSM04340
C                                                                       LSM04350
C-----------------------------------------------------------------------LSM04360
C     STORAGE LOCATIONS OF THE A-ARRAY                                  LSM04370
C     AND THE ORDER OF THE A-MATRIX ARE PASSED TO THE MATVEC SUBROUTINE.LSM04380
C     ENTRY MATVE(A,M,N)                                                LSM04390
      ENTRY DMATVE(A,M,N)                                               LSM04400
C-----------------------------------------------------------------------LSM04410
C                                                                       LSM04420
C-----END OF MATRIX -VECTOR MULTIPLY 'DIAGONAL' TEST PROBLEMS-----------LSM04430
      RETURN                                                            LSM04440
      END                                                               LSM04450
C                                                                       LSM04460
C-----MATRIX-VECTOR MULTIPLY FOR 'DIAGONAL' TEST MATRICES---------------LSM04470
C                                                                       LSM04480
      SUBROUTINE STRAN(Z,W,SUM)                                         LSM04490
C     SUBROUTINE DSTRAN(Z,W,SUM)                                        LSM04500
C                                                                       LSM04510
C-----------------------------------------------------------------------LSM04520
      DOUBLE PRECISION  A(1),Z(1),W(1),SUM                              LSM04530
C-----------------------------------------------------------------------LSM04540
C                                                                       LSM04550
C     COMPUTES W = A-TRANSPOSE*Z - SUM*W .  ASSUMES M >= N.             LSM04560
      DO 10 I = 1,N                                                     LSM04570
   10 W(I) = A(I)*Z(I)- SUM*W(I)                                        LSM04580
      RETURN                                                            LSM04590
C                                                                       LSM04600
C-----------------------------------------------------------------------LSM04610
C     STORAGE LOCATIONS OF THE A-ARRAY AND THE ORDER                    LSM04620
C     OF THE A-MATRIX ARE OBTAINED FROM USPEC SUBROUTINE.               LSM04630
C     ENTRY MTRANE(A,M,N)                                               LSM04640
      ENTRY DMTRAE(A,M,N)                                               LSM04650
C-----------------------------------------------------------------------LSM04660
C                                                                       LSM04670
C-----END OF SPARSE SYMMETRIC MATRIX-VECTOR MULTIPLY--------------------LSM04680
      RETURN                                                            LSM04690
      END                                                               LSM04700
```

6.6 LSSUB: Other Subroutines used by the Codes in Chapter 6

```
C-----LSSUB-------(SINGULAR VALUES AND VECTORS)------------------------LSS00010
C  Authors:  Jane Cullum and Ralph A. Willoughby (Deceased)           LSS00020
C            Los Alamos National Laboratory                           LSS00030
C            Los Alamos, New Mexico 87544                             LSS00040
C                                                                     LSS00050
C            E-mail:  cullumj@lanl.gov                                LSS00060
C                                                                     LSS00070
C  These codes are copyrighted by the authors.  These codes           LSS00080
C  and modifications of them or portions of them are NOT to be        LSS00090
C  incorporated into any commercial codes or used for any other       LSS00100
C  commercial purposes such as consulting for other companies,        LSS00110
C  without legal agreements with the authors of these Codes.          LSS00120
C  If these Codes or portions of them are used in other scientific or LSS00130
C  engineering research works the names of the authors of these codes LSS00140
C  and appropriate references to their written work are to be         LSS00150
C  incorporated in the derivative works.                              LSS00160
C                                                                     LSS00170
C  This header is not to be removed from these codes.                 LSS00180
C                                                                     LSS00190
C         REFERENCE: Cullum and Willoughby, Chapter 5                 LSS00191
C         Lanczos Algorithms for Large Symmetric Eigenvalue ComputationsLSS00192
C         VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in LSS00193
C         Applied Mathematics, 2002. SIAM Publications,               LSS00194
C         Philadelphia, PA. USA                                       LSS00195
C                                                                     LSS00196
C                                                                     LSS00197
C     ACCORDING TO PFORT THESE SUBROUTINES ARE PORTABLE               LSS00200
C                                                                     LSS00210
C                                                                     LSS00220
C     SUBROUTINES    BISEC, INVERR, TNORM, LUMP, ISOEV, PRTEST, AND   LSS00230
C                    INVERM ARE USED WITH LANCZOS SINGULAR VALUE      LSS00240
C                    PROGRAM LSVAL.  STURMI, INVERM, LBISEC, TNORM    LSS00250
C                    ARE USED WITH THE LANCZOS SINGULAR VECTOR        LSS00260
C                    PROGRAM LSVEC.                                   LSS00270
C                                                                     LSS00280
C                                                                     LSS00290
C-----COMPUTE T-EIGENVALUES BY BISECTION-------------------------------LSS00300
C                                                                     LSS00310
      SUBROUTINE BISEC(BETA,BETA2,VB,VS,LBD,UBD,EPS,TTOL,MP,          LSS00320
     1 NINT,MEV,NDIS,IC,IWRITE)                                       LSS00330
C                                                                     LSS00340
C---------------------------------------------------------------------LSS00350
      DOUBLE PRECISION  BETA(1),BETA2(1),VB(1),VS(1)                  LSS00360
      DOUBLE PRECISION  LBD(1),UBD(1),EPS,EPT,EP0,EP1,TEMP,TTOL       LSS00370
      DOUBLE PRECISION  ZERO,ONE,HALF,YU,YV,LB,UB,XL,XU,X1,X0,XS,BETAM LSS00380
      INTEGER  MP(1),IDEF(10)                                         LSS00390
      DOUBLE PRECISION  DABS, DSQRT, DMAX1, DMIN1, DFLOAT             LSS00400
C---------------------------------------------------------------------LSS00410
C     COMPUTES EIGENVALUES OF T(1,MEV) BY LOOPING INTERNALLY ON THE    LSS00420
C     USER-SPECIFIED INTERVALS,  (LB(J),UB(J)), J = 1,NINT.  INTERVALS LSS00430
```

```
C     ARE TREATED AS OPEN ON THE LEFT AND CLOSED ON THE RIGHT.          LSS00440
C     THE BISEC SUBROUTINE SIMULTANEOUSLY LABELS SPURIOUS T-EIGENVALUES LSS00450
C     AND DETERMINES THE T-MULTIPLICITIES OF EACH GOOD T-EIGENVALUE.    LSS00460
C     SPURIOUS T-EIGENVALUES ARE LABELLED BY A T-MULTIPLICITY = 0.      LSS00470
C     ANY T-EIGENVALUE WITH A T-MULTIPLICITY >= 1 IS 'GOOD'.            LSS00480
C                                                                       LSS00490
C     IF IWRITE = 0 THEN MOST OF THE WRITES TO FILE 6 ARE NOT           LSS00500
C     ACTIVATED.                                                        LSS00510
C                                                                       LSS00520
C     NOTE THAT PROGRAM ASSUMES THAT NO MORE THAN MMAX/2 T-EIGENVALUES  LSS00530
C     OF T(1,MEV) ARE TO BE COMPUTED IN ANY ONE OF THE SUBINTERVALS     LSS00540
C     CONSIDERED, WHERE MMAX = DIMENSION OF VB SPECIFIED BY THE USER    LSS00550
C     IN THE MAIN PROGRAM LEVAL.                                        LSS00560
C                                                                       LSS00570
C     ON ENTRY                                                          LSS00580
C     BETA2(J) IS SET = BETA(J)*BETA(J).  THE STORAGE FOR BETA2 COULD   LSS00590
C     BE ELIMINATED BY RECOMPUTING THE BETA(J)**2 FOR EACH STURM        LSS00600
C     SEQUENCE.                                                         LSS00610
C                                                                       LSS00620
C     EPS = 2*MACHEP =  4.4 * 10**-16 ON IBM 3081.                      LSS00630
C     TTOL = EPS*TKMAX WHERE                                            LSS00640
C     TKMAX = MAX(BETA(K), K=1,KMAX)                                    LSS00650
C                                                                       LSS00660
C     ON EXIT                                                           LSS00670
C     NDIS = TOTAL NUMBER OF COMPUTED DISTINCT T-EIGENVALUES OF         LSS00680
C            T(1,MEV) ON THE UNION OF THE (LB,UB) INTERVALS.            LSS00690
C     VS = COMPUTED DISTINCT T-EIGENVALUES OF T(1,MEV) IN ALGEBRAICALLY-LSS00700
C          INCREASING ORDER                                             LSS00710
C     MP = CORRESPONDING T-MULTIPLICITIES OF THESE T-EIGENVALUES        LSS00720
C     MP(I) = (0,1,MI), MI>1, I=1,NDIS  MEANS:                          LSS00730
C        (0)  V(I) IS SPURIOUS                                          LSS00740
C        (1)  V(I) IS T-ISOLATED AND GOOD                               LSS00750
C        (MI) V(I) IS T-MULTIPLE AND HENCE A CONVERGED GOOD T-EIGENVALUELSS00760
C     IC = TOTAL NUMBER OF STURMS USED                                  LSS00770
C                                                                       LSS00780
C     DEFAULTS                                                          LSS00790
C     ISKIP = 0 INITIALLY. IF DEFAULT OCCURS ON J-TH SUB-INTERVAL, SET  LSS00800
C             ISKIP=ISKIP+1 AND IDEF(ISKIP) = J                         LSS00810
C             DEFAULTS OCCUR IF THERE ARE NO T-EIGENVALUES  IN THE      LSS00820
C             SUBINTERVAL SPECIFIED OR IF THE NUMBER                    LSS00830
C             OF STURMS SEQUENCES REQUIRED EXCEEDS MXSTUR.              LSS00840
C             WHEN A DEFAULT OCCURS THE PROGRAM                         LSS00850
C             SKIPS THE INTERVAL INVOLVED AND GOES ON TO THE NEXT       LSS00860
C             INTERVAL.                                                 LSS00870
C                                                                       LSS00880
C-----------------------------------------------------------------------LSS00890
C     SPECIFY PARAMETERS                                                LSS00900
      ZERO = 0.0D0                                                      LSS00910
      ONE  = 1.0D0                                                      LSS00920
      HALF = 0.5D0                                                      LSS00930
      MXSTUR = IC                                                       LSS00940
      NDIS = 0                                                          LSS00950
      IC = 0                                                            LSS00960
      ISKIP = 0                                                         LSS00970
      MP1 = MEV+1                                                       LSS00980
```

```
C     SAVE THEN SET BETA(MEV+1) = 0. GENERATE BETA**2
      BETAM = BETA(MP1)
      BETA(MP1) = ZERO
C
      DO 10 I = 1,MP1
   10 BETA2(I) = BETA(I)*BETA(I)
C
C     EP0 IS USED IN T-MULTIPLICITY AND SPURIOUS TESTS
C     EP1 AND EPS ARE USED IN THE BISEC CONVERGENCE TEST
C
      TEMP = DFLOAT(MEV+1000)
      EP0  = TEMP*TTOL
      EP1  = DSQRT(TEMP)*TTOL
C
      WRITE(6,20)MEV,NINT
   20 FORMAT(/' BISEC CALCULATION'/' ORDER OF T IS',I6/
     1' NUMBER OF INTERVALS IS',I6/)
C
      WRITE(6,30) EP0,EP1
   30 FORMAT(/' MULTOL, TOLERANCE USED IN T-MULTIPLICITY AND SPURIOUS TE
     1STS = ',E10.3/' BISTOL, TOLERANCE USED IN BISEC CONVERGENCE TEST =
     1',E10.3/)
C
C     LOOP ON THE NINT INTERVALS  (LB(J),UB(J)), J=1,NINT
      DO 430 JIND = 1,NINT
      LB = LBD(JIND)
      UB = UBD(JIND)
C
      WRITE(6,40)JIND,LB,UB
   40 FORMAT(//1X,'BISEC INTERVAL NO',2X,'LOWER BOUND',2X,'UPPER BOUND'/
     1I18,2E13.5/)
C
C     INITIALIZATION AND PARAMETER SPECIFICATION
C     ICT IS TOTAL STURM COUNT ON (LB,UB)
C
      NA = 0
      MD = 0
      NG = 0
      ICT = 0
C
C     START OF T-EIGENVALUE CALCULATIONS
      X1 = UB
      ISTURM = 1
      GO TO 330
C     FORWARD STURM CALCULATION TO DETERMINE NA = NO. T-EIGENVALUES > UB
   50 NA = NEV
C
      X1 = LB
      ISTURM = 2
      GO TO 330
C     FORWARD STURM CALC TO DETERMINE MT = NO. T-EIGENVALUES ON (LB,UB)
   60 CONTINUE
      MT=NEV
      ICT = ICT +2
C
```

```
      WRITE(6,70)MT,NA
   70 FORMAT(/2I6,' = NO. TMEV ON (LB,UB) AND NO. .GT. UB'/)
C
C     DEFAULT TEST: IS ESTIMATED NUMBER OF STURMS > MXSTUR?
      IEST = 30*MT
      IF (IEST.LT.MXSTUR) GO TO 90
C
      WRITE(6,80)
   80 FORMAT(//' ESTIMATED NUMBER OF STURMS REQUIRED EXCEEDS USER LIMIT'
     1/' SKIP THIS SUBINTERVAL')
      GO TO 110
C
   90 CONTINUE
C
      IF (MT.GE.1) GO TO 120
C
      WRITE(6,100)
  100 FORMAT(//' THERE ARE NO T-EIGENVALUES ON THIS INTERVAL)'/)
C
  110 ISKIP = ISKIP+1
      IDEF(ISKIP) = JIND
      GO TO 430
C
C     REGULAR CASE.
  120 CONTINUE
C
      IF (IWRITE.NE.0) WRITE(6,130)
  130 FORMAT(/' DISTINCT T-EIGENVALUES COMPUTED USING BISEC'/
     1 13X,'T-EIGENVALUE',2X,'TMULT',3X,'MD',4X,'NG')
C
C     SET UP INITIAL UPPER AND LOWER BOUNDS FOR T-EIGENVALUES
      DO 140 I=1,MT
      VB(I) = LB
      MTI = MT + I
  140 VB(MTI) = UB
C
C     CALCULATE T-EIGENVALUES FROM LB UP TO UB  K = MT,...,1
C     MAIN LOOP FOR FINDING KTH T-EIGENVALUE
C
      K = MT
  150 CONTINUE
      ICO = 0
      XL = VB(K)
      MTK = MT+K
      XU = VB(MTK)
C
      ISTURM = 3
      X1 = XU
      ICO = ICO + 1
      GO TO 330
C     FORWARD STURM CALCULATION AT XU
  160 NU=NEV
C
C     BISECTION LOOP FOR KTH T-EIGENVALUE. TEST X1=MIDPOINT OF (XL,XU)
      ISTURM = 4
```

```
  170 CONTINUE                                                          LSS02090
      X1 = (XL+XU)*HALF                                                 LSS02100
      XS = DABS(XL)+DABS(XU)                                            LSS02110
      X0 = XU-XL                                                        LSS02120
      EPT = EPS*XS+EP1                                                  LSS02130
C                                                                       LSS02140
C     EPT IS CONVERGENCE TOLERANCE FOR KTH T-EIGENVALUE                 LSS02150
C                                                                       LSS02160
      IF (X0.LE.EPT) GO TO 230                                          LSS02170
C                                                                       LSS02180
C     T-EIGENVALUE HAS NOT YET CONVERGED                                LSS02190
C                                                                       LSS02200
      IC0 = IC0 + 1                                                     LSS02210
      GO TO 330                                                         LSS02220
C     FORWARD STURM CALCULATION AT CURRENT T-EIGENVALUE APPROXIMATION.  LSS02230
  180 CONTINUE                                                          LSS02240
C                                                                       LSS02250
C     UPDATE T-EIGENVALUE INTERVAL (XL,XU)                              LSS02260
C                                                                       LSS02270
      IF (NEV.LT.K) GO TO 190                                           LSS02280
C                                                                       LSS02290
C     NUMBER OF T-EIGENVALUES NEV = K                                   LSS02300
      XL = X1                                                           LSS02310
      GO TO 170                                                         LSS02320
  190 CONTINUE                                                          LSS02330
C     NUMBER OF T-EIGENVALUES NEV<K                                     LSS02340
      XU = X1                                                           LSS02350
      NU = NEV                                                          LSS02360
C                                                                       LSS02370
C     UPDATE OF T-EIGENVALUE BOUNDS                                     LSS02380
C                                                                       LSS02390
      IF (NEV.EQ.0) GO TO 210                                           LSS02400
C                                                                       LSS02410
      DO 200 I = 1,NEV                                                  LSS02420
  200 VB(I) = DMAX1(X1,VB(I))                                           LSS02430
C                                                                       LSS02440
  210 NEV1 = NEV+1                                                      LSS02450
C                                                                       LSS02460
      DO 220 II = NEV1,K                                                LSS02470
      I = MT+II                                                         LSS02480
  220 VB(I) = DMIN1(X1,VB(I))                                           LSS02490
C                                                                       LSS02500
      GO TO 170                                                         LSS02510
C                                                                       LSS02520
C     END (XL,XU) BISECTION LOOP FOR KTH T-EIGENVALUE ON (LB,UB)        LSS02530
C     TEST FOR T-MULTIPLICITY AND IF SIMPLE THEN TEST FOR SPURIOUSNESS  LSS02540
C                                                                       LSS02550
  230 CONTINUE                                                          LSS02560
      NDIS = NDIS+1                                                     LSS02570
      MD = MD+1                                                         LSS02580
      VS(NDIS) = X1                                                     LSS02590
C                                                                       LSS02600
      JSTURM = 1                                                        LSS02610
      X1 = XL-EP0                                                       LSS02620
      GO TO 370                                                         LSS02630
```

```
C     BACKWARD STURM CALCULATION
  240 KL = KEV
      JL = JEV
C
      JSTURM = 2
      IC0 = IC0 + 2
      X1 = XU+EP0
      GO TO 370
C     BACKWARD STURM CALCULATION
  250 JU = JEV
      KU = KEV
C
C     FOR T(1,MEV)
C     NU - KU = NO. T-EIGENVALUES ON (XU, XU + EP0)
C     KL - KU = NO. T-EIGENVALUES ON (XL - EP0, XU + EP0)
C
C     FOR T(2,MEV)
C     JL -JU = NO. T-EIGENVALUES ON (XL - EP0, XU + EP0)
C
C     IS THIS A SIMPLE T-EIGENVALUE?
C
      IF (KL-KU-1.EQ.0) GO TO 290
C
C     VS(NDIS) = KTH-T-EIGENVALUE OF (LB,UB) IS T-MULTIPLE AND HENCE
C     GOOD
      IF (KU.EQ.NU) GO TO 280
C     CONTINUE TO CHECK FOR T-MULTIPLICITY
  260 CONTINUE
      ISTURM = 5
      X1 = X1+EP0
      IC0 = IC0 + 1
      GO TO 330
C     FORWARD STURM CALCULATION
  270 KNE = KU-NEV
      KU = NEV
      IF (KNE.NE.0) GO TO 260
C     SPECIFY T-MULTIPLICITY = MP(NDIS)
  280 MPEV = KL-KU
      KNEW = KU
      GO TO 300
C     END T-MULTIPLE CASE
C
C     T-EIGENVALUE IS SIMPLE   CHECK IF IT IS SPURIOUS
  290 CONTINUE
      MPEV = 1
      IF (JU.LT.JL) MPEV=0
      KNEW = K-1
C
C     X1 >= XU+EP0
C     SPURIOUS TEST AND SIMPLE CASE COMPLETED
C     START OF NEXT T-EIGENVALUE COMPUTATION
C
  300 K = KNEW
      MP(NDIS) = MPEV
      IF (MPEV.GE.1) NG = NG + 1
```

```
C                                                                       LSS03190
      IF (IWRITE.NE.0) WRITE(6,310) VS(NDIS),MPEV,MD,NG                 LSS03200
  310 FORMAT(E25.16,3I6)                                                LSS03210
C                                                                       LSS03220
C     UPDATE STURM COUNT. IC0 = STURM COUNT FOR KTH T-EIGENVALUE        LSS03230
      ICT = ICT + IC0                                                   LSS03240
C                                                                       LSS03250
C     EXIT TEST FOR K DO LOOP                                           LSS03260
C                                                                       LSS03270
      IF (K.LE.0) GO TO 410                                             LSS03280
C                                                                       LSS03290
C     UPDATE LOWER BOUNDS                                               LSS03300
      DO 320 I=1,KNEW                                                   LSS03310
  320 VB(I) = DMAX1(X1,VB(I))                                           LSS03320
C                                                                       LSS03330
      GO TO 150                                                         LSS03340
C     END OF BISECTION LOOP FOR KTH EIGENVALUE                          LSS03350
C                                                                       LSS03360
C     FORWARD STURM CALCULATION                                         LSS03370
  330 NEV = -NA                                                         LSS03380
      YU  = ONE                                                         LSS03390
C                                                                       LSS03400
      DO 360 I = 1,MEV                                                  LSS03410
      IF (YU.NE.ZERO) GO TO 340                                         LSS03420
      YV = BETA(I)/EPS                                                  LSS03430
      GO TO 350                                                         LSS03440
  340 YV = BETA2(I)/YU                                                  LSS03450
  350 YU = X1 - YV                                                      LSS03460
      IF (YU.GE.ZERO) GO TO 360                                         LSS03470
      NEV = NEV + 1                                                     LSS03480
  360 CONTINUE                                                          LSS03490
C     NEV = NUMBER OF T-EIGENVALUES ON (X1,UB)                          LSS03500
C                                                                       LSS03510
      GO TO (50,60,160,180,270), ISTURM                                 LSS03520
C                                                                       LSS03530
C     BACKWARD STURM CALCULATION FOR T(1,MEV) AND T(2,MEV)              LSS03540
  370 KEV = -NA                                                         LSS03550
      YU  = ONE                                                         LSS03560
C                                                                       LSS03570
      DO 400 II = 1,MEV                                                 LSS03580
      I = MP1-II                                                        LSS03590
      IF (YU.NE.ZERO) GO TO 380                                         LSS03600
      YV = BETA(I+1)/EPS                                                LSS03610
      GO TO 390                                                         LSS03620
  380 YV = BETA2(I+1)/YU                                                LSS03630
  390 YU = X1-YV                                                        LSS03640
      JEV = 0                                                           LSS03650
      IF (YU.GE.ZERO) GO TO 400                                         LSS03660
      KEV = KEV+1                                                       LSS03670
      JEV = 1                                                           LSS03680
  400 CONTINUE                                                          LSS03690
      JEV = KEV-JEV                                                     LSS03700
C                                                                       LSS03710
      GO TO (240,250), JSTURM                                           LSS03720
C                                                                       LSS03730
```

```
C     KEV = -NA + (NUMBER OF T(1,MEV) T-EIGENVALUES) > X1
C     JEV = -NA + (NUMBER OF T(2,MEV) T-EIGENVALUES) > X1
C
C     SET PARAMETERS FOR NEXT INTERVAL
  410 CONTINUE
      IC = ICT+IC
      MXSTUR = MXSTUR-ICT
C
      WRITE(6,420) JIND,NG,MD,ICT
  420 FORMAT(/' T-EIGENVALUE CALCULATION ON INTERVAL',I6,'  IS COMPLETE'
     1 /3X,'NO. GOOD',3X,'NO. DISTINCT',4X,'STURMS'/I10,I13,I10)
C
  430 CONTINUE
C
C     END LOOP ON THE SUBINTERVALS (LB(J),UB(J)), J=1,NINT
C     ISKIP OUTPUT
C
      IF (ISKIP.GT.0) WRITE(6,440)ISKIP
  440 FORMAT(' BISEC DEFAULTED ON',I3,3X,'INTERVALS'/
     1 ' DEFAULTS OCCUR IF AN INTERVAL HAS NO T-EIGENVALUES'/
     2 ' OR THE STURM ESTIMATE EXCEEDS THE USER-SPECIFIED LIMIT'/)
C
      IF (ISKIP.GT.0) WRITE(6,450)(IDEF(I), I=1,ISKIP)
  450 FORMAT(' BISEC DEFAULTED ON INTERVALS'/(10I8))
C
C     RESET BETA AT I = MP1
      BETA(MP1) = BETAM
C-----END OF BISEC------------------------------------------------------
      RETURN
      END
C
C-----INVERSE ITERATION ON T(1,MEV)-------------------------------------
C
      SUBROUTINE INVERR(BETA,V1,V2,VS,EPS,G,MP,MEV,MMB,NDIS,NISO,
     1 NM,IKL,IT,IWRITE)
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION BETA(1),V1(1),V2(1),VS(1)
      DOUBLE PRECISION X1,U,Z,EST,TEMP,T0,T1,RATIO,SUM,XU,NORM,TSUM
      DOUBLE PRECISION  BETAM,EPS,EPS3,EPS4,ZERO,ONE
      REAL G(1)
      INTEGER  MP(1)
C-----------------------------------------------------------------------
      DOUBLE PRECISION FINPRO
      REAL  ABS
      DOUBLE PRECISION  DABS, DMIN1, DSQRT, DFLOAT
C-----------------------------------------------------------------------
C     COMPUTES ERROR ESTIMATES FOR COMPUTED ISOLATED GOOD T-EIGENVALUES
C     IN VS AND WRITES THESE T-EIGENVALUES AND ESTIMATES TO FILE 4.
C     BY DEFINITION A GOOD T-EIGENVALUE IS ISOLATED IF ITS
C     CLOSEST NEIGHBOR IS ALSO GOOD, OR IF ONE OF ITS NEIGHBORS IS
C     SPURIOUS BUT THAT NEIGHBOR IS FAR ENOUGH AWAY.  SO
C     IN PARTICULAR, WE COMPUTE ESTIMATES FOR GOOD T-EIGENVALUES
C     THAT ARE IN CLUSTERS OF GOOD T-EIGENVALUES.
C
```

```
C     USES INVERSE ITERATION ON T(1,MEV) SOLVING THE EQUATION           LSS04290
C     (T - X1*I)V2 = RIGHT-HAND SIDE (RANDOMLY-GENERATED)               LSS04300
C     FOR EACH SUCH GOOD T-EIGENVALUE X1.                               LSS04310
C                                                                       LSS04320
C     PROGRAM REFACTORS T-X1*I ON EACH ITERATION OF INVERSE ITERATION.  LSS04330
C     TYPICALLY ONLY ONE ITERATION IS NEEDED PER T-EIGENVALUE X1.       LSS04340
C                                                                       LSS04350
C     POSSIBLE STORAGE COMPRESSION                                      LSS04360
C     G STORAGE COULD BE ELIMINATED BY REGENERATING THE RANDOM          LSS04370
C     RIGHT-HAND SIDE ON EACH ITERATION AND PRINTING OUT THE            LSS04380
C     ERROR ESTIMATES AS THEY ARE GENERATED.                            LSS04390
C                                                                       LSS04400
C     ON ENTRY AND EXIT                                                 LSS04410
C     MEV = ORDER OF T                                                  LSS04420
C     BETA CONTAINS THE NONZERO ENTRIES OF THE T-MATRIX                 LSS04430
C     VS = COMPUTED DISTINCT EIGENVALUES OF T(1,MEV)                    LSS04440
C     MP = T-MULTIPLICITY OF EACH EIGENVALUE IN VS. MP(I) = -1 MEANS    LSS04450
C          VS(I) IS A GOOD T-EIGENVALUE BUT THAT IT IS SITTING CLOSE TO LSS04460
C          A SPURIOUS T-EIGENVALUE.  MP(I) = 0 MEANS VS(I) IS SPURIOUS. LSS04470
C          ESTIMATES ARE COMPUTED ONLY FOR THOSE T-EIGENVALUES          LSS04480
C          WITH MP(I) = 1. FLAGGING WAS DONE IN SUBROUTINE ISOEV         LSS04490
C          PRIOR TO ENTERING INVERR.                                    LSS04500
C     NISO = NUMBER OF ISOLATED GOOD T-EIGENVALUES CONTAINED IN VS      LSS04510
C     NDIS =  NUMBER OF DISTINCT T-EIGENVALUES IN VS                    LSS04520
C     IKL = SEED FOR RANDOM NUMBER GENERATOR                            LSS04530
C     EPS = 2. * MACHINE EPSILON                                        LSS04540
C                                                                       LSS04550
C     IN PROGRAM:                                                       LSS04560
C     ITER = MAXIMUM NUMBER OF INVERSE ITERATION STEPS ALLOWED FOR EACH LSS04570
C            X1.  ITER = IT ON ENTRY.                                   LSS04580
C     G = ARRAY OF DIMENSION AT LEAST MEV + NISO.  USED TO STORE        LSS04590
C         RANDOMLY-GENERATED RIGHT-HAND SIDE.  THIS IS NOT              LSS04600
C         REGENERATED FOR EACH X1. G IS ALSO USED TO STORE ERROR        LSS04610
C         ESTIMATES AS THEY ARE COMPUTED FOR LATER PRINTOUT.            LSS04620
C     V1,V2 = WORK SPACES USED IN THE FACTORIZATION OF T(1,MEV).        LSS04630
C     AT THE END OF THE INVERSE ITERATION COMPUTATION FOR X1, V2        LSS04640
C     CONTAINS THE UNIT EIGENVECTOR OF T(1,MEV) CORRESPONDING TO X1.    LSS04650
C     V1 AND V2 MUST BE OF DIMENSION AT LEAST MEV.                      LSS04660
C                                                                       LSS04670
C     ON EXIT                                                           LSS04680
C     G(J) = MINIMUM GAP IN T(1,MEV) FOR EACH VS(J), J=1,NDIS           LSS04690
C     G(MEV+I) = BETAM*|V2(MEV)| = ERROR ESTIMATE FOR ISOLATED GOOD     LSS04700
C              T-EIGENVALUES, WHERE I = 1,NISO  AND  BETAM = BETA(MEV+1)LSS04710
C              V2(MEV) IS LAST COMPONENT OF THE UNIT T-EIGENVECTOR OF   LSS04720
C              T(1,MEV) CORRESPONDING TO ITH ISOLATED GOOD T-EIGENVALUE.LSS04730
C                                                                       LSS04740
C     IF FOR SOME X1 IT.GT.ITER THEN THE ERROR ESTIMATE IN G IS MARKED  LSS04750
C     WITH A - SIGN.                                                    LSS04760
C                                                                       LSS04770
C     V2 = ISOLATED GOOD T-EIGENVALUES                                  LSS04780
C     V1 = MINIMAL T-GAPS FOR THE EIGENVALUES IN V2.                    LSS04790
C     THESE ARE CONSTRUCTED FOR WRITE-OUT PURPOSES ONLY AND NOT         LSS04800
C     NEEDED ELSEWHERE IN THE PROGRAM.                                  LSS04810
C-----------------------------------------------------------------------LSS04820
C                                                                       LSS04830
```

```
C     LABEL OUTPUT FILE  4                                              LSS04840
      IF (MMB.EQ.1) WRITE(4,10)                                         LSS04850
   10 FORMAT(' INVERSE ITERATION ERROR ESTIMATES'/)                     LSS04860
C                                                                       LSS04870
C     FILE 6 (TERMINAL) OUTPUT OF ERROR ESTIMATES                       LSS04880
      IF (IWRITE.NE.0.AND.NISO.NE.0) WRITE(6,20)                        LSS04890
   20 FORMAT(/' INVERSE ITERATION ERROR ESTIMATES'/'  JISO',' JDIST',8X LSS04900
     1,'GOOD T-EIGENVALUE',4X,'BETAM*UM',5X,'TMINGAP')                  LSS04910
C                                                                       LSS04920
C     INITIALIZATION AND PARAMETER SPECIFICATION                        LSS04930
      ZERO = 0.0D0                                                      LSS04940
      ONE = 1.0D0                                                       LSS04950
      NG = 0                                                            LSS04960
      NISO = 0                                                          LSS04970
      ITER = IT                                                         LSS04980
      MP1 = MEV+1                                                       LSS04990
      MM1 = MEV-1                                                       LSS05000
      BETAM = BETA(MP1)                                                 LSS05010
      BETA(MP1) = ZERO                                                  LSS05020
C                                                                       LSS05030
C     CALCULATE SCALE AND TOLERANCES                                    LSS05040
      TSUM = ZERO                                                       LSS05050
      DO 30 I = 2,MEV                                                   LSS05060
   30 TSUM = TSUM + BETA(I)                                             LSS05070
C                                                                       LSS05080
      EPS3 = EPS*TSUM                                                   LSS05090
      EPS4 = DFLOAT(MEV)*EPS3                                           LSS05100
C                                                                       LSS05110
C     GENERATE SCALED RANDOM RIGHT-HAND SIDE                            LSS05120
      ILL = IKL                                                         LSS05130
C                                                                       LSS05140
C-----------------------------------------------------------------------LSS05150
      CALL GENRAN(ILL,G,MEV)                                            LSS05160
C-----------------------------------------------------------------------LSS05170
C                                                                       LSS05180
      GSUM = ZERO                                                       LSS05190
      DO 40 I = 1,MEV                                                   LSS05200
   40 GSUM = GSUM+ABS(G(I))                                             LSS05210
      GSUM = EPS4/GSUM                                                  LSS05220
C                                                                       LSS05230
      DO 50 I = 1,MEV                                                   LSS05240
   50 G(I) = GSUM*G(I)                                                  LSS05250
C                                                                       LSS05260
C     LOOP ON ISOLATED GOOD T-EIGENVALUES IN VS (MP(I) = 1) TO          LSS05270
C     CALCULATE CORRESPONDING UNIT T-EIGENVECTOR OF T(1,MEV)            LSS05280
C                                                                       LSS05290
      DO 180 JEV = 1,NDIS                                               LSS05300
      IF (MP(JEV).EQ.0) GO TO 180                                       LSS05310
      NG = NG + 1                                                       LSS05320
      IF (MP(JEV).NE.1) GO TO 180                                       LSS05330
      IT = 1                                                            LSS05340
      NISO = NISO + 1                                                   LSS05350
      X1 = VS(JEV)                                                      LSS05360
C                                                                       LSS05370
C     INITIALIZE RIGHT HAND SIDE FOR INVERSE ITERATION                  LSS05380
```

```
      DO 60 I = 1,MEV                                                   LSS05390
   60 V2(I) = G(I)                                                      LSS05400
C                                                                       LSS05410
C     TRIANGULAR FACTORIZATION WITH NEAREST NEIGHBOR PIVOT              LSS05420
C     STRATEGY. INTERCHANGES ARE LABELLED BY SETTING BETA < 0.          LSS05430
C                                                                       LSS05440
   70 CONTINUE                                                          LSS05450
      U = -X1                                                           LSS05460
      Z = BETA(2)                                                       LSS05470
C                                                                       LSS05480
      DO 90 I = 2,MEV                                                   LSS05490
      IF (BETA(I).GT.DABS(U)) GO TO 80                                  LSS05500
C     NO INTERCHANGE                                                    LSS05510
      V1(I-1) = Z/U                                                     LSS05520
      V2(I-1) = V2(I-1)/U                                               LSS05530
      V2(I) = V2(I)-BETA(I)*V2(I-1)                                     LSS05540
      RATIO = BETA(I)/U                                                 LSS05550
      U = -X1-Z*RATIO                                                   LSS05560
      Z = BETA(I+1)                                                     LSS05570
      GO TO 90                                                          LSS05580
   80 CONTINUE                                                          LSS05590
C     INTERCHANGE CASE                                                  LSS05600
      RATIO = U/BETA(I)                                                 LSS05610
      BETA(I) = -BETA(I)                                                LSS05620
      V1(I-1) = -X1                                                     LSS05630
      U = Z-RATIO*V1(I-1)                                               LSS05640
      Z = -RATIO*BETA(I+1)                                              LSS05650
      TEMP = V2(I-1)                                                    LSS05660
      V2(I-1) = V2(I)                                                   LSS05670
      V2(I) = TEMP-RATIO*V2(I)                                          LSS05680
   90 CONTINUE                                                          LSS05690
      IF (U.EQ.ZERO) U = EPS3                                           LSS05700
C                                                                       LSS05710
C     SMALLNESS TEST AND DEFAULT VALUE FOR LAST COMPONENT               LSS05720
C     PIVOT(I-1) = |BETA(I)| FOR INTERCHANGE CASE                       LSS05730
C     (I-1,I+1) ELEMENT IN RIGHT FACTOR = BETA(I+1)                     LSS05740
C     END OF FACTORIZATION AND FORWARD SUBSTITUTION                     LSS05750
C                                                                       LSS05760
C     BACK SUBSTITUTION                                                 LSS05770
      V2(MEV) = V2(MEV)/U                                               LSS05780
      DO 110 II = 1,MM1                                                 LSS05790
      I = MEV-II                                                        LSS05800
      IF (BETA(I+1).LT.ZERO) GO TO 100                                  LSS05810
C     NO INTERCHANGE                                                    LSS05820
      V2(I) = V2(I)-V1(I)*V2(I+1)                                       LSS05830
      GO TO 110                                                         LSS05840
C     INTERCHANGE CASE                                                  LSS05850
  100 BETA(I+1) = -BETA(I+1)                                            LSS05860
      V2(I) = (V2(I)-V1(I)*V2(I+1)-BETA(I+2)*V2(I+2))/BETA(I+1)         LSS05870
  110 CONTINUE                                                          LSS05880
C                                                                       LSS05890
C     TESTS FOR CONVERGENCE OF INVERSE ITERATION                        LSS05900
C     IF SUM |V2| COMPS. LE. 1 AND IT. LE. ITER DO ANOTHER INVIT STEP   LSS05910
C                                                                       LSS05920
      NORM = DABS(V2(MEV))                                              LSS05930
```

```
      DO 120 II = 1,MM1
      I = MEV-II
  120 NORM = NORM+DABS(V2(I))
C
      IF (NORM.GE.ONE) GO TO 140
      IT = IT+1
      IF (IT.GT.ITER) GO TO 140
      XU = EPS4/NORM
C
      DO 130 I = 1,MEV
  130 V2(I) = V2(I)*XU
C
      GO TO 70
C     ANOTHER INVERSE ITERATION STEP
C
C     INVERSE ITERATION FINISHED
C     NORMALIZE COMPUTED T-EIGENVECTOR : V2 = V2/||V2||
  140 CONTINUE
      SUM = FINPRO(MEV,V2(1),1,V2(1),1)
      SUM = ONE/DSQRT(SUM)
C
      DO 150 II = 1,MEV
  150 V2(II) = SUM*V2(II)
C
C     SAVE ERROR ESTIMATE FOR LATER OUTPUT
      EST = BETAM*DABS(V2(MEV))
      IF (IT.GT.ITER) EST = -EST
      MEVPNI = MEV + NISO
      G(MEVPNI) = EST
      IF (IWRITE.EQ.0) GO TO 180
C
C     FILE 6 (TERMINAL) OUTPUT OF ERROR ESTIMATES.
      IF (JEV.EQ.1) GAP = VS(2) - VS(1)
      IF (JEV.EQ.MEV) GAP = VS(MEV) - VS(MEV-1)
      IF (JEV.EQ.MEV.OR.JEV.EQ.1) GO TO 160
      TEMP = DMIN1(VS(JEV+1)-VS(JEV),VS(JEV)-VS(JEV-1))
      GAP = TEMP
  160 CONTINUE
C
      WRITE(6,170) NISO,JEV,X1,EST,GAP
  170 FORMAT(2I6,E25.16,2E12.3)
C
  180 CONTINUE
C
C     END ERROR ESTIMATE LOOP ON ISOLATED GOOD T-EIGENVALUES.
C     GENERATE DISTINCT MINGAPS FOR T(1,MEV).  THIS IS USEFUL AS AN
C     INDICATOR OF THE GOODNESS OF THE INVERSE ITERATION ESTIMATES.
C     TRANSFER ISOLATED GOOD T-EIGENVALUES AND CORRESPONDING TMINGAPS
C     TO V2 AND V1 FOR OUTPUT PURPOSES ONLY.
C
      NM1 = NDIS - 1
      G(NDIS) = VS(NM1)-VS(NDIS)
      G(1) = VS(2)-VS(1)
C
      DO 190 J = 2,NM1
```

```
      T0 = VS(J)-VS(J-1)
      T1 = VS(J+1)-VS(J)
      G(J) = T1
      IF (T0.LT.T1) G(J)=-T0
  190 CONTINUE
      ISO = 0
      DO 200 J = 1,NDIS
      IF (MP(J).NE.1) GO TO 200
      ISO = ISO+1
      V1(ISO) = G(J)
      V2(ISO) = VS(J)
  200 CONTINUE
C
      IF(NISO.EQ.0) GO TO 250
C
C     ERROR ESTIMATES ARE WRITTEN TO FILE 4
      WRITE(4,210)MEV,NDIS,NG,NISO,NM,IKL,ITER,BETAM
  210 FORMAT(1X,'TSIZE',2X,'NDIS',1X,'NGOOD',2X,'NISO',3X,'M+N'/5I6/
     1 4X,'RHSEED',2X,'MXINIT',5X,'BETAM'/I10,I8,E10.3/
     2 2X,'GOODEVNO',8X,'GOOD T-EIGENVALUE',6X,'BETAM*UM',7X,'TMINGAP')
C
      ISPUR = 0
      I = 0
      DO 240 J = 1,NDIS
      IF(MP(J).NE.0) GO TO 220
      ISPUR = ISPUR + 1
      GO TO 240
  220 IF(MP(J).NE.1) GO TO 240
      I = I + 1
      MEVI = MEV + I
      IGOOD = J - ISPUR
      WRITE(4,230) IGOOD,V2(I),G(MEVI),V1(I)
  230 FORMAT(I10,E25.16,2E14.3)
  240 CONTINUE
      GO TO 270
C
  250 WRITE(4,260)
  260 FORMAT(/' THERE ARE NO ISOLATED T-EIGENVALUES SO NO ERROR ESTIMATE
     1S WERE COMPUTED')
C     RESTORE BETA(MEV+1) = BETAM
  270 BETA(MP1) = BETAM
C-----END OF INVERR---------------------------------------------------
      RETURN
      END
C
C-----START OF TNORM--------------------------------------------------
C
      SUBROUTINE TNORM(BETA,BMIN,TMAX,MEV,IB)
C
C--------------------------------------------------------------------
      DOUBLE PRECISION  BETA(1)
      DOUBLE PRECISION  TMAX,BMIN,BSIZE,BTOL
      DOUBLE PRECISION  DABS, DMAX1
C--------------------------------------------------------------------
C     COMPUTE SCALING FACTOR USED IN THE T-MULTIPLICITY, SPURIOUS AND
```

```
C     PRTESTS.   CHECK RELATIVE SIZE OF THE BETA(K), K=1,MEV
C     AS A TEST ON THE LOCAL ORTHOGONALITY OF THE LANCZOS VECTORS.
C
C             TMAX = MAX (BETA(I),  I=1,MEV)
C             BMIN = MIN (BETA(I) I=2,MEV)
C             BSIZE = BMIN/TMAX
C             |IB| = INDEX OF MINIMAL(BETA)
C             IB < 0 IF BMIN/TMAX < BTOL
C-----------------------------------------------------------------------
C     SPECIFY PARAMETERS
      IB = 2
      BTOL = BMIN
      BMIN = BETA(2)
      TMAX = BETA(2)
C
      DO 20 I = 2,MEV
      IF (BETA(I).GE.BMIN) GO TO 10
      IB = I
      BMIN = BETA(I)
   10 TMAX = DMAX1(TMAX,BETA(I))
   20 CONTINUE
C
C     TEST OF LOCAL ORTHOGONALITY USING SCALED BETAS
      BSIZE = BMIN/TMAX
      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
     1OVER 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
C-----START OF LUMP-----------------------------------------------------
C
      SUBROUTINE LUMP(V1,RELTOL,MULTOL,SCALE2,LINDEX,LOOP)
C
C-----------------------------------------------------------------------
```

```
      DOUBLE PRECISION  V1(1),SUM,RELTOL,MULTOL,THOLD,ZERO,SCALE2       LSS07590
      INTEGER  LINDEX(1)                                                LSS07600
      DOUBLE PRECISION  DABS, DFLOAT, DMAX1                             LSS07610
C-----------------------------------------------------------------------LSS07620
C     LINDEX(J) = T-MULTIPLICITY OF JTH DISTINCT T-EIGENVALUE           LSS07630
C     LOOP = NUMBER OF DISTINCT T-EIGENVALUES                           LSS07640
C     LUMP 'COMBINES' COMPUTED 'GOOD' T-EIGENVALUES THAT ARE            LSS07650
C     'TOO CLOSE'.                                                      LSS07660
C     VALUE FOR RELTOL IS 1.D-10.                                       LSS07670
C                                                                       LSS07680
C     IF IN A SET OF T-EIGENVALUES TO BE COMBINED THERE IS AN EIGENVALUELSS07690
C     WITH LINDEX=1, THEN THE VALUE OF THE COMBINED T-EIGENVALUES IS SETLSS07700
C     EQUAL TO THE VALUE OF THAT T-EIGENVALUE.  NOTE THAT IF A SPURIOUS LSS07710
C     T-EIGENVALUE IS TO BE 'COMBINED' WITH A GOOD T-EIGENVALUE, THEN   LSS07720
C     THIS IS DONE ONLY BY INCREASING THE INDEX, LINDEX, FOR THAT       LSS07730
C     T-EIGENVALUE. NUMERICAL VALUES OF SPURIOUS T-EIGENVALUES ARE      LSS07740
C     NEVER COMBINED WITH THOSE OF GOOD T-EIGENVALUES.                   LSS07750
C-----------------------------------------------------------------------LSS07760
      ZERO = 0.0D0                                                      LSS07770
      NLOOP = 0                                                         LSS07780
      J = 0                                                             LSS07790
      ICOUNT = 1                                                        LSS07800
      JI = 1                                                            LSS07810
      THOLD = DMAX1(RELTOL*DABS(V1(1)),SCALE2*MULTOL)                   LSS07820
C     THOLD = DMAX1(RELTOL*DABS(V1(1)),RELTOL)                          LSS07830
C                                                                       LSS07840
   10 J = J+1                                                           LSS07850
      IF (J.EQ.LOOP) GO TO 20                                           LSS07860
      SUM = DABS(V1(J)-V1(J+1))                                         LSS07870
      IF (SUM.LT.THOLD) GO TO 60                                        LSS07880
   20 JF = JI + ICOUNT - 1                                              LSS07890
      INDSUM = 0                                                        LSS07900
      ISPUR = 0                                                         LSS07910
C                                                                       LSS07920
      DO 30 KK = JI,JF                                                  LSS07930
      IF (LINDEX(KK).NE.0) GO TO 30                                     LSS07940
      ISPUR = ISPUR + 1                                                 LSS07950
      INDSUM = INDSUM + 1                                               LSS07960
   30 INDSUM = INDSUM + LINDEX(KK)                                      LSS07970
C                                                                       LSS07980
C     IF (JF-JI.GE.1) WRITE(6,40) (V1(KKK), KKK=JI,JF)                  LSS07990
   40 FORMAT(/' LUMP LUMPS THE T-EIGENVALUES'/(4E20.13))                LSS08000
C                                                                       LSS08010
C     COMPUTE THE 'COMBINED' T-EIGENVALUE AND THE RESULTING             LSS08020
C     T-MULTIPLICITY                                                    LSS08030
      K = JI - 1                                                        LSS08040
   50 K = K+1                                                           LSS08050
      IF (K.GT.JF) GO TO 70                                             LSS08060
      IF (LINDEX(K) .NE.1) GO TO 50                                     LSS08070
      NLOOP = NLOOP + 1                                                 LSS08080
      V1(NLOOP) = V1(K)                                                 LSS08090
      GO TO 100                                                         LSS08100
   60 ICOUNT = ICOUNT + 1                                               LSS08110
      GO TO 10                                                          LSS08120
C                                                                       LSS08130
```

```
C     ALL INDICES WERE 0 OR >1                                          LSS08140
   70 NLOOP = NLOOP + 1                                                 LSS08150
      IDIF = INDSUM - ISPUR                                             LSS08160
      IF (IDIF.EQ.0) GO TO 90                                           LSS08170
C                                                                       LSS08180
      SUM = ZERO                                                        LSS08190
      DO 80 KK = JI,JF                                                  LSS08200
   80 SUM = SUM + V1(KK) * DFLOAT(LINDEX(KK))                           LSS08210
C                                                                       LSS08220
      V1(NLOOP) = SUM/DFLOAT(IDIF)                                      LSS08230
      GO TO 100                                                         LSS08240
   90 V1(NLOOP) = V1(JI)                                                LSS08250
  100 LINDEX(NLOOP) = INDSUM                                            LSS08260
      IDIF = INDSUM - ISPUR                                             LSS08270
      IF (IDIF.EQ.0.AND.ISPUR.EQ.1) LINDEX(NLOOP) = 0                   LSS08280
      IF (J.EQ.LOOP) GO TO 110                                          LSS08290
      ICOUNT = 1                                                        LSS08300
      JI= J+1                                                           LSS08310
      THOLD = DMAX1(RELTOL*DABS(V1(JI)),SCALE2*MULTOL)                  LSS08320
C     THOLD = DMAX1(RELTOL*DABS(V1(JI)),RELTOL)                         LSS08330
      IF (JI.LT.LOOP) GO TO 10                                          LSS08340
      NLOOP = NLOOP + 1                                                 LSS08350
      V1(NLOOP)= V1(JI)                                                 LSS08360
      LINDEX(NLOOP) = LINDEX(JI)                                        LSS08370
  110 CONTINUE                                                          LSS08380
C                                                                       LSS08390
C     ON RETURN V1 CONTAINS THE DISTINCT T-EIGENVALUES                  LSS08400
C     LINDEX CONTAINS THE CORRESPONDING T-MULTIPLICITIES                LSS08410
C                                                                       LSS08420
      LOOP = NLOOP                                                      LSS08430
      RETURN                                                            LSS08440
C-----END OF LUMP-------------------------------------------------------LSS08450
      END                                                               LSS08460
C                                                                       LSS08470
C                                                                       LSS08480
C-----START OF ISOEV----------------------------------------------------LSS08490
C                                                                       LSS08500
      SUBROUTINE ISOEV(VS,GAPTOL,MULTOL,SCALE1,G,MP,NDIS,NG,NISO)       LSS08510
C                                                                       LSS08520
C-----------------------------------------------------------------------LSS08530
      DOUBLE PRECISION  VS(1),T0,T1,MULTOL,GAPTOL,SCALE1,TEMP           LSS08540
      REAL   G(1),GAP                                                   LSS08550
      INTEGER  MP(1)                                                    LSS08560
      REAL  ABS                                                         LSS08570
      DOUBLE PRECISION  DABS, DMAX1                                     LSS08580
C-----------------------------------------------------------------------LSS08590
C     GENERATE DISTINCT TMINGAPS AND USE THEM TO LABEL THE ISOLATED     LSS08600
C     GOOD T-EIGENVALUES THAT ARE VERY CLOSE TO SPURIOUS ONES.          LSS08610
C     ERROR ESTIMATES WILL NOT BE COMPUTED FOR THESE T-EIGENVALUES.     LSS08620
C                                                                       LSS08630
C     ON ENTRY AND EXIT                                                 LSS08640
C     VS CONTAINS THE COMPUTED DISTINCT T-EIGENVALUES OF T(1,MEV)       LSS08650
C     MP CONTAINS THE CORRESPONDING T-MULTIPLICITIES                    LSS08660
C     NDIS = NUMBER OF DISTINCT T-EIGENVALUES                           LSS08670
C     GAPTOL = RELATIVE GAP TOLERANCE SET IN MAIN                       LSS08680
```

```
C
C     ON EXIT
C     G CONTAINS THE TMINGAPS.
C     G(I) < 0 MEANS MINGAP IS DUE TO LEFT GAP
C     MP(I) IS NOT CHANGED EXCEPT THAT  MP(I)=-1, IF MP(I)=1,
C     TMINGAP WAS TOO SMALL AND DUE TO A SPURIOUS T-EIGENVALUE.
C
C     IF MP(I)=-1 THAT SIMPLE GOOD T-EIGENVALUE WILL BE SKIPPED
C     IN THE SUBSEQUENT ERROR ESTIMATE COMPUTATIONS IN INVERR
C     THAT IS, WE COMPUTE ERROR ESTIMATES ONLY FOR THOSE GOOD
C     T-EIGENVALUES WITH MP(I)=1.
C-----------------------------------------------------------------------
C     CALCULATE MINGAPS FOR DISTINCT T(1,MEV) EIGENVALUES.
      NM1 = NDIS - 1
      G(NDIS) = VS(NM1)-VS(NDIS)
      G(1) = VS(2)-VS(1)
C
      DO 10 J = 2,NM1
      T0 = VS(J)-VS(J-1)
      T1 = VS(J+1)-VS(J)
      G(J) = T1
      IF (T0.LT.T1) G(J) = -T0
   10 CONTINUE
C
C     SET MP(I)=-1 FOR SIMPLE GOOD T-EIGENVALUES WHOSE MINGAPS  ARE
C     'TOO SMALL' AND DUE TO SPURIOUS T-EIGENVALUES.
C
      NISO = 0
      NG = 0
      DO 20 J = 1,NDIS
      IF (MP(J).EQ.0) GO TO 20
      NG = NG+1
      IF (MP(J).NE.1) GO TO 20
C     VS(J) IS NEXT TO SIMPLE GOOD T-EIGENVALUE
      NISO = NISO + 1
      I = J+1
      IF (G(J).LT.0.0) I = J-1
      IF (MP(I).NE.0) GO TO 20
      GAP = ABS(G(J))
      T0 = DMAX1(SCALE1*MULTOL,GAPTOL*DABS(VS(J)))
C     T0 = DMAX1(GAPTOL,GAPTOL*DABS(VS(J)))
      TEMP = T0
      IF (GAP.GT.TEMP) GO TO 20
      MP(J) = -MP(J)
      NISO = NISO-1
   20 CONTINUE
C
C-----END OF ISOEV------------------------------------------------------
      RETURN
      END
C
C-----START OF PRTEST---------------------------------------------------
C
      SUBROUTINE PRTEST(BETA,TEIG,TKMAX,EPSM,RELTOL,SCALE3,SCALE4,
     1 TMULT,NDIST,MEV,IPROJ)
```

```
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION  BETA(1),TEIG(1),SIGMA(4)
      DOUBLE PRECISION  EPSM,RELTOL,PRTOL,TKMAX,LRATIO,URATIO
      DOUBLE PRECISION  EPS,EPS1,BETAM,LBD,UBD,SIG,YU,YV,LRATS,URATS
      DOUBLE PRECISION  ZERO,ONE,TEN,BISTOL,SCALE3,SCALE4,AEV,TEMP
      INTEGER  TMULT(1),ISIGMA(4)
      DOUBLE PRECISION  DABS, DMAX1, DSQRT, DFLOAT
C-----------------------------------------------------------------------
C     AFTER CONVERGENCE HAS BEEN ESTABLISHED, SUBROUTINE PRTEST
C     TESTS COMPUTED EIGENVALUES OF T(1,MEV) THAT HAVE BEEN LABELLED
C     SPURIOUS TO DETERMINE IF ANY SINGULAR VALUES OF A HAVE BEEN
C     MISSED BY LANCZOS PROCEDURE.  A SINGULAR VALUE WHOSE
C     SINGULAR VECTOR(S) HAS A VERY SMALL PROJECTION ON THE
C     STARTING VECTOR (< SINGLE PRECISION) CAN BE MISSED BECAUSE
C     IT WILL THEN ALSO BE AN EIGENVALUE OF T(2,MEV) TO WITHIN
C     THE SQUARE OF THIS ORIGINAL PROJECTION.  HOWEVER,
C     OUR EXPERIENCE IS THAT SUCH SMALL PROJECTIONS OCCUR ONLY
C     VERY INFREQUENTLY.
C
C     THIS SUBROUTINE IS CALLED ONLY AFTER CONVERGENCE HAS BEEN
C     ESTABLISHED. ONCE CONVERGENCE HAS BEEN OBSERVED ON THE
C     OTHER SINGULAR VALUES, THEN ONE CAN EXPECT TO ALSO HAVE
C     CONVERGENCE ON ANY SUCH 'HIDDEN' SINGULAR VALUES. (IF THERE
C     ARE ANY).  PROCEDURE CONSIDERS ONLY SPURIOUS T-EIGENVALUES AND
C     ONLY THOSE SPURIOUS T-EIGENVALUES THAT ARE ISOLATED FROM GOOD
C     T-EIGENVALUES.  FOR EACH SUCH T-EIGENVALUE IT DOES 2 STURM
C     SEQUENCES AND A FEW SCALAR MULTIPLICATIONS.  UPON RETURN TO MAIN
C     PROGRAM ERROR ESTIMATES WILL BE COMPUTED FOR ANY T-EIGENVALUES
C     THAT HAVE BEEN LABELLED AS 'HIDDEN'.  SUCH T-EIGENVALUES
C     WILL BE RELABELLED AS 'GOOD' ONLY IF THESE ERROR ESTIMATES
C     ARE SUFFICIENTLY SMALL.
C-----------------------------------------------------------------------
      ZERO = 0.0D0
      ONE  = 1.0D0
      TEN  = 10.0D0
      PRTOL = 1.D-6
      TEMP = DFLOAT(MEV+1000)
      TEMP = DSQRT(TEMP)
      BISTOL = TKMAX*EPSM*TEMP
      NSIGMA = 4
      SIGMA(1) = TEN*TKMAX
C
      DO 10 J = 2,NSIGMA
   10 SIGMA(J) = TEN*SIGMA(J-1)
C
      IFIN = 0
      MF = 1
      ML = MEV
      BETAM = BETA(MF)
      BETA(MF) = ZERO
      IPROJ = 0
      J = 1
C
      IF (TMULT(1).NE.0) GO TO 110
```

```
C
      AEV = DABS(TEIG(1))
      TEMP = PRTOL*AEV
      EPS1 = DMAX1(TEMP,SCALE4*BISTOL)
C     EPS1 = DMAX1(TEMP,PRTOL)
      TEMP = RELTOL*AEV
      EPS  = DMAX1(TEMP,SCALE3*BISTOL)
C     EPS  = DMAX1(TEMP,RELTOL)
C
      IF (TEIG(2)-TEIG(1).LT.EPS1.AND.TMULT(2).NE.0) GO TO 110
C
   20 LBD = TEIG(J) - EPS
      UBD = TEIG(J) + EPS
      MEVL = 0
      IL = 0
      YU = ONE
C
      DO 50 I=MF,ML
      IF (YU.NE.ZERO) GO TO 30
      YV = BETA(I)/EPSM
      GO TO 40
   30 YV = BETA(I)*BETA(I)/YU
   40 YU = -LBD-YV
      IF (YU.GE.ZERO) GO TO 50
C     MEVL INCREMENTED
      MEVL = MEVL + 1
      IL = I
   50 CONTINUE
C
      LRATIO = YU
      MEV1L = MEVL
      IF (IL.EQ.ML) MEV1L=MEVL-1
C
C     MEVL = NUMBER OF EVS OF T(1,MEV) WHICH ARE < LBD
C     MEV1L = NUMBER OF EVS OF T(1,MEV-1) WHICH ARE < LBD
C     LRATIO = DET(T(1,MEV)-LBD)/DET(T(1,MEV-1)-LBD):
C
      MEVU = 0
      IL = 0
      YU = ONE
C
      DO 80 I=MF,ML
      IF (YU.NE.ZERO) GO TO 60
      YV = BETA(I)/EPSM
      GO TO 70
   60 YV = BETA(I)*BETA(I)/YU
   70 YU = -UBD-YV
      IF (YU.GE.ZERO) GO TO 80
C     MEVU INCREMENTED
      MEVU = MEVU + 1
      IL = I
   80 CONTINUE
C
      URATIO = YU
      MEV1U = MEVU
```

```
      IF (IL.EQ.ML) MEV1U=MEVU-1
C
C     MEVU = NUMBER OF EVS OF T(MEV) WHICH ARE < UBD
C     MEV1U = NUMBER OF EVS OF T(MEV-1) WHICH ARE < UBD
C     URATIO = DET(TM-UBD)/DET(T(M-1)-UBD): TM=T(MF,ML)
C
      NEV1 = MEV1U-MEV1L
C
      DO 90 K=1,NSIGMA
      SIG = SIGMA(K)
      LRATS = LRATIO-SIG
      URATS = URATIO-SIG
C     NOTE THE INCREMENT IS ON NUMBER OF EVALUES OF T(M-1)
      MEVLS = MEV1L
      IF (LRATS.LT.0.) MEVLS=MEV1L+1
      MEVUS = MEV1U
      IF (URATS.LT.0.) MEVUS=MEV1U+1
      ISIGMA(K) = MEVUS - MEVLS
   90 CONTINUE
C
      ICOUNT = 0
      DO 100 K=1,NSIGMA
  100 IF (ISIGMA(K).EQ.1) ICOUNT=ICOUNT + 1
C
      IF (ICOUNT.LT.2.OR.NEV1.EQ.0) GO TO 110
      TMULT(J) = -10
      IPROJ=IPROJ+1
C
  110 J=J+1
C
      IF (J.GE.NDIST) GO TO 120
      IF (TMULT(J).NE.0) GO TO 110
C
      AEV = DABS(TEIG(J))
      TEMP = PRTOL*AEV
      EPS1 = DMAX1(TEMP,SCALE4*BISTOL)
C     EPS1 = DMAX1(TEMP,PRTOL)
      TEMP = RELTOL*AEV
      EPS  = DMAX1(TEMP,SCALE3*BISTOL)
C     EPS  = DMAX1(TEMP,RELTOL)
C
      IF (TEIG(J)-TEIG(J-1).LT.EPS1.AND.TMULT(J-1).NE.0) GO TO 110
      IF (TEIG(J+1)-TEIG(J).LT.EPS1.AND.TMULT(J+1).NE.0) GO TO 110
C
      GO TO 20
C
  120 IF (IFIN.EQ.1) GO TO 130
      IF (TMULT(NDIST).NE.0) GO TO 130
C
      AEV = DABS(TEIG(NDIST))
      TEMP = PRTOL*AEV
      EPS1 = DMAX1(TEMP,SCALE4*BISTOL)
C     EPS1 = DMAX1(TEMP,PRTOL)
      TEMP = RELTOL*AEV
      EPS  = DMAX1(TEMP,SCALE3*BISTOL)
```

```
C     EPS  = DMAX1(TEMP,RELTOL)
C
      NDIST1=NDIST -1
      TEMP = TEIG(NDIST)-TEIG(NDIST1)
      IF (TEMP.LT.EPS1.AND.TMULT(NDIST1).NE.0) GO TO 130
      IFIN = 1
C
      GO TO 20
C
  130 BETA(MF) = BETAM
C
C-----END OF PRTEST-------------------------------------------------------
      RETURN
      END
C
C------START OF STURMI----------------------------------------------------
C
      SUBROUTINE STURMI(BETA,X1,TOLN,EPSM,MMAX,MK1,MK2,IC,IWRITE)
C
C------------------------------------------------------------------------
      DOUBLE PRECISION  BETA(1)
      DOUBLE PRECISION  EPSM,X1,TOLN,EVL,EVU,BETA2
      DOUBLE PRECISION  U1,U2,V1,V2,ZERO,ONE
      INTEGER I,IC,ICD,IC0,IC1,IC2,MK1,MK2,MMAX
C------------------------------------------------------------------------
C
C     FOR ANY GOOD T-EIGENVALUE THAT HAS CONVERGED AS AN EIGENVALUE
C     OF THE T-MATRICES THIS SUBROUTINE CALCULATES
C     THE SMALLEST SIZE OF THE T-MATRIX, T(1,MK1) DEFINED
C     BY THE BETA ARRAY SUCH THAT MK1.LE.MMAX
C     AND THE INTERVAL (X1-TOLN,X1+TOLN) CONTAINS AT LEAST ONE
C     EIGENVALUE OF T(1,MK1). IT ALSO CALCULATES MK2 <= MMAX
C     AS THE SMALLEST SIZE T-MATRIX (IF ANY) SUCH THAT THIS INTERVAL
C     CONTAINS AT LEAST TWO EIGENVALUES OF T(1,MK2).
C     IF NO T-MATRIX OF ORDER < MMAX SATISFIES THIS REQUIREMENT
C     THEN MK2 IS SET EQUAL TO MMAX.  THE SINGULAR VECTOR PROGRAM
C     USES THESE VALUES TO DETERMINE A 1ST GUESS AT AN APPROPRIATE
C     SIZE T-MATRIX FOR THE SINGULAR VALUE  X1.
C
C     ON EXIT IC = NUMBER OF EIGENVALUES OF T(1,MK2) IN THIS INTERVAL
C
C     STURMI REGENERATES THE QUANTITIES BETA(I)**2 EACH TIME IT IS
C     CALLED, OBVIOUSLY FOR THE PRICE OF ANOTHER VECTOR OF LENGTH
C     MMAX THIS GENERATION COULD BE DONE ONCE IN THE MAIN
C     PROGRAM BEFORE THE LOOP ON THE CALLS TO SUBROUTINE STURMI.
C
C     IF ANY OF THE GOOD T-EIGENVALUES BEING CONSIDERED WERE MULTIPLE
C     AS SINGULAR VALUES OF THE USER-SPECIFIED MATRIX, THEN
C     THIS SUBROUTINE COULD BE MODIFIED TO COMPUTE ADDITIONAL
C     SIZES MKJ, J = 3, ...  WHICH COULD THEN BE USED IN THE
C     MAIN LANCZOS SINGULAR VECTOR PROGRAM TO COMPUTE ADDITIONAL
C     SINGULAR VECTORS CORRESPONDING TO THESE MULTIPLE SINGULAR
C     VALUES. THE MAIN PROGRAM LSVEC PROVIDED DOES NOT INCLUDE
C     THIS OPTION.
C
```

```
C-----------------------------------------------------------------------
C     INITIALIZATION OF PARAMETERS
      MK1 = 0
      MK2 = 0
      ZERO = 0.0D0
      ONE  = 1.0D0
      BETA(1) = ZERO
      EVL = X1-TOLN
      EVU = X1+TOLN
      U1 = ONE
      U2 = ONE
      IC0 = 0
      IC1 = 0
      IC2 = 0
C
C     MAIN LOOP FOR CALCULATING THE SIZES MK1,MK2
      DO 60 I = 1,MMAX
      BETA2 = BETA(I)*BETA(I)
      IF (U1.NE.ZERO) GO TO 10
      V1 = BETA(I)/EPSM
      GO TO 20
   10 V1 = BETA2/U1
   20 U1 = EVL - V1
      IF (U1.LT.ZERO) IC1 = IC1+1
      IF (U2.NE.ZERO) GO TO 30
      V2 = BETA(I)/EPSM
      GO TO 40
   30 V2 = BETA2/U2
   40 U2 = EVU - V2
      IF (U2.LT.ZERO) IC2 = IC2+1
C     TEST FOR CHANGE IN NUMBER OF T-EIGENVALUES ON (EVL,EVU)
      ICD = IC1-IC2
      IC = ICD-IC0
      IF (IC.GE.1) GO TO 50
      GO TO 60
   50 CONTINUE
      IF (IC0.EQ.0) MK1 = I
      IC0 = IC0+1
      IF (IC0.GT.1) GO TO 70
   60 CONTINUE
C
      I = I-1
      IF (IC0.EQ.0) MK1 = MMAX
   70 MK2 = I
      IC = ICD
C
      IF (IWRITE.EQ.1) WRITE(6,80) X1,MK1,MK2,IC
   80 FORMAT(' EVAL =',E20.12,' MK1 =',I6,' MK2 =',I6,' IC =',I3/)
C
      RETURN
C-----END OF STURMI-----------------------------------------------------
      END
C
C
C-----START OF INVERM---------------------------------------------------
```

```
C
      SUBROUTINE INVERM(BETA,V1,V2,X1,ERROR,ERRORV,EPS,G,MEV,IT,
     1 IWRITE)
C
C-----------------------------------------------------------------------
      DOUBLE PRECISION  BETA(1),V1(1),V2(1)
      DOUBLE PRECISION  X1,U,Z,TEMP,RATIO,SUM,XU,NORM,TSUM,BETAM
      DOUBLE PRECISION  EPS,EPS3,EPS4,ERROR,ERRORV,ZERO,ONE
      REAL  G(1)
      DOUBLE PRECISION  DABS, DSQRT, DFLOAT
      DOUBLE PRECISION FINPRO
      REAL  ABS
C-----------------------------------------------------------------------
C
C     COMPUTES T-EIGENVECTORS FOR ISOLATED GOOD T-EIGENVALUES X1
C     USING INVERSE ITERATION ON T(1,MEV(X1)) SOLVING EQUATION
C     (T - X1*I)V2 = RIGHT-HAND SIDE (RANDOMLY-GENERATED) .
C     PROGRAM REFACTORS T- X1*I ON EACH ITERATION OF INVERSE ITERATION.
C     TYPICALLY ONLY ONE ITERATION IS NEEDED PER T-EIGENVALUE X1.
C
C     IF IWRITE = 1 THEN THERE ARE EXTENDED WRITES TO FILE 6 (TERMINAL)
C
C     ON ENTRY G CONTAINS A REAL*4 RANDOM VECTOR WHICH WAS GENERATED
C     IN MAIN PROGRAM.
C
C     ON ENTRY AND EXIT
C     MEV = ORDER OF T
C     BETA CONTAINS THE OFFDIAGONAL ENTRIES OF T.
C     EPS = 2. * MACHINE EPSILON
C
C     IN PROGRAM:
C     ITER = MAXIMUM NUMBER STEPS ALLOWED FOR INVERSE ITERATION
C     ITER = IT ON ENTRY.
C     V1,V2 = WORK SPACES USED IN THE FACTORIZATION OF T(1,MEV).
C     V1 AND V2 MUST BE OF DIMENSION AT LEAST MEV.
C
C     ON EXIT
C     V2 = THE UNIT EIGENVECTOR OF T(1,MEV) CORRESPONDING TO X1.
C     ERROR = |V2(MEV)| = ERROR ESTIMATE FOR CORRESPONDING
C             RITZ VECTOR FOR X1.
C
C     ERRORV = || T*V2 - X1*V2 || = ERROR ESTIMATE ON T-EIGENVECTOR.
C     IF IT.GT.ITER THEN ERRORV = -ERRORV
C     IT = NUMBER OF ITERATIONS ACTUALLY REQUIRED
C-----------------------------------------------------------------------
C     INITIALIZATION AND PARAMETER SPECIFICATION
      ONE  = 1.0D0
      ZERO = 0.0D0
      ITER = IT
      MP1 = MEV+1
      MM1 = MEV-1
      BETAM = BETA(MP1)
      BETA(MP1) = ZERO
C
C     CALCULATE SCALE AND TOLERANCES
```

```
      TSUM = ZERO
      DO 10 I = 2,MEV
   10 TSUM = TSUM + BETA(I)
C
      EPS3 = EPS*TSUM
      EPS4 = DFLOAT(MEV)*EPS3
C
C     GENERATE SCALED RANDOM RIGHT-HAND SIDE
      GSUM = ZERO
      DO 20 I = 1,MEV
   20 GSUM = GSUM+ABS(G(I))
      GSUM = EPS4/GSUM
C
C     INITIALIZE RIGHT HAND SIDE FOR INVERSE ITERATION
      DO 30 I = 1,MEV
   30 V2(I) = GSUM*G(I)
      IT = 1
C
C     CALCULATE UNIT EIGENVECTOR OF T(1,MEV) FOR ISOLATED GOOD
C     T-EIGENVALUE X1.
C
C     TRIANGULAR FACTORIZATION WITH NEAREST NEIGHBOR PIVOT
C     STRATEGY. INTERCHANGES ARE LABELLED BY SETTING BETA < 0.
C
   40 CONTINUE
      U = -X1
      Z = BETA(2)
C
      DO 60 I=2,MEV
      IF (BETA(I).GT.DABS(U)) GO TO 50
C     NO PIVOT INTERCHANGE
      V1(I-1) = Z/U
      V2(I-1) = V2(I-1)/U
      V2(I) = V2(I)-BETA(I)*V2(I-1)
      RATIO = BETA(I)/U
      U = -X1-Z*RATIO
      Z = BETA(I+1)
      GO TO 60
C     PIVOT INTERCHANGE
   50 CONTINUE
      RATIO = U/BETA(I)
      BETA(I) = -BETA(I)
      V1(I-1) = -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 (U.EQ.ZERO) U=EPS3
C
C     SMALLNESS TEST AND DEFAULT VALUE FOR LAST COMPONENT
C     PIVOT(I-1) = |BETA(I)| FOR INTERCHANGE CASE
C     (I-1,I+1) ELEMENT IN RIGHT FACTOR = BETA(I+1)
```

```
C     END OF FACTORIZATION AND FORWARD SUBSTITUTION
C
C     BACK SUBSTITUTION
      V2(MEV) = V2(MEV)/U
      DO 80 II = 1,MM1
      I = MEV-II
      IF (BETA(I+1).LT.ZERO) GO TO 70
C     NO PIVOT INTERCHANGE
      V2(I) = V2(I)-V1(I)*V2(I+1)
      GO TO 80
C     PIVOT INTERCHANGE
   70 BETA(I+1) = -BETA(I+1)
      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 = DABS(V2(MEV))
      DO 90 II = 1,MM1
      I = MEV-II
   90 NORM = NORM+DABS(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
  100 V2(I) = V2(I)*XU
C
      GO TO 40
C
C     NORMALIZE COMPUTED T-EIGENVECTOR : V2 = V2/||V2||
C
  110 CONTINUE
C
      SUM = FINPRO(MEV,V2(1),1,V2(1),1)
      SUM = ONE/DSQRT(SUM)
      DO 120 II = 1,MEV
  120 V2(II) = SUM*V2(II)
C
C     SAVE ERROR ESTIMATE FOR LATER OUTPUT
      ERROR = DABS(V2(MEV))
C
C     GENERATE ERRORV = ||T*V2 - X1*V2||.
      V1(MEV) = BETA(MEV)*V2(MEV-1)-X1*V2(MEV)
      DO 130 J = 2,MM1
      JM = MP1 - J
      V1(JM) = BETA(JM)*V2(JM-1) + BETA(JM+1)*V2(JM+1
```

```
     1) - X1*V2(JM)
  130 CONTINUE
C
      V1(1) = BETA(2)*V2(2) - X1*V2(1)
      ERRORV = FINPRO(MEV,V1(1),1,V1(1),1)
      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(' INVERSE ITERATION OUTPUT'/
     1 2X,'TSIZE',13X,'T-EIGENVALUE',11X,'U(M)',9X,'ERRORV'/
     1 I6,E25.16,2E15.5)
C
C     RESTORE BETA(MEV+1) = BETAM
  150 CONTINUE
      BETA(MP1) = BETAM
C-----END OF INVERM----------------------------------------------------
      RETURN
      END
C
C-----START OF LBISEC--------------------------------------------------
C
      SUBROUTINE LBISEC(BETA,EPSM,EVAL,EVALN,LB,UB,TTOL,M,NEVT)
C
C---------------------------------------------------------------------
      DOUBLE PRECISION  BETA(1),X0,X1,XL,XU,YU,YV,LB,UB
      DOUBLE PRECISION  EPSM,EP1,EVAL,EVALN,EVD,EPT
      DOUBLE PRECISION  ZERO,ONE,HALF,TTOL,TEMP
      DOUBLE PRECISION  DABS,DSQRT,DFLOAT
C---------------------------------------------------------------------
C     SPECIFY PARAMETERS
      ZERO = 0.0D0
      HALF = 0.5D0
      ONE  = 1.0D0
      XL = LB
      XU = UB
C
C     EP1 = DSQRT(1000+M)*TTOL     TTOL = EPSM*TKMAX
C     TKMAX = MAX(BETA(K), K= 1,KMAX)
C
      TEMP = DFLOAT(1000+M)
      EP1 = DSQRT(TEMP)*TTOL
C
      NA = 0
      X1 = XU
      JSTURM = 1
      GO TO 60
C     FORWARD STURM CALCULATION
   10 NA = NEV
      X1 = XL
      JSTURM = 2
      GO TO 60
C     FORWARD STURM CALCULATION
```

```
   20 NEVT = NEV                                                        LSS14190
C                                                                       LSS14200
C     WRITE(6,30) M,EVAL,NEVT,EP1                                       LSS14210
   30 FORMAT(/3X,'TSIZE',23X,'EV',9X/I8,E25.16/                         LSS14220
     1 I6,' = NUMBER OF T(1,M) EIGENVALUES ON TEST INTERVAL'/           LSS14230
     1 E12.3,' = CONVERGENCE TOLERANCE'/)                               LSS14240
C                                                                       LSS14250
      IF (NEVT.NE.1) GO TO 120                                          LSS14260
C                                                                       LSS14270
C     BISECTION LOOP                                                    LSS14280
      JSTURM = 3                                                        LSS14290
   40 X1 = HALF*(XL+XU)                                                 LSS14300
      X0 = XU-XL                                                        LSS14310
      EPT = EPSM*(DABS(XL) + DABS(XU)) + EP1                            LSS14320
C     CONVERGENCE TEST                                                  LSS14330
      IF (X0.LE.EPT) GO TO 100                                          LSS14340
      GO TO 60                                                          LSS14350
C     FORWARD STURM CALCULATION                                         LSS14360
   50 CONTINUE                                                          LSS14370
      IF(NEV.EQ.0) XU = X1                                              LSS14380
      IF(NEV.EQ.1) XL = X1                                              LSS14390
      GO TO 40                                                          LSS14400
C     NEV = NUMBER OF EIGENVALUES OF T(1,M) ON (X1,XU)                  LSS14410
C     THERE IS EXACTLY ONE EIGENVALUE OF T(1,M) ON (XL,XU)              LSS14420
C                                                                       LSS14430
C     FORWARD STURM CALCULATION                                         LSS14440
   60 NEV = -NA                                                         LSS14450
      YU = ONE                                                          LSS14460
      DO 90 I = 1,M                                                     LSS14470
      IF (YU.NE.ZERO) GO TO 70                                          LSS14480
      YV = BETA(I)/EPSM                                                 LSS14490
      GO TO 80                                                          LSS14500
   70 YV = BETA(I)*BETA(I)/YU                                           LSS14510
   80 YU = X1 - YV                                                      LSS14520
      IF (YU.GE.ZERO) GO TO 90                                          LSS14530
      NEV = NEV+1                                                       LSS14540
   90 CONTINUE                                                          LSS14550
      GO TO (10,20,50), JSTURM                                          LSS14560
C                                                                       LSS14570
  100 CONTINUE                                                          LSS14580
C                                                                       LSS14590
      EVALN = X1                                                        LSS14600
      EVD = DABS(EVALN-EVAL)                                            LSS14610
C     WRITE(6,110) EVALN,EVAL,EVD                                       LSS14620
  110 FORMAT(/20X,'EVALN',21X,'EVAL',6X,'CHANGE'/2E25.16,E12.3/)        LSS14630
C                                                                       LSS14640
  120 CONTINUE                                                          LSS14650
      RETURN                                                            LSS14660
C-----END OF LBISEC-----------------------------------------------------LSS14670
      END                                                               LSS14680
```

6.7 LSVAL: LSVEC: File Definitions, Sample Input Files

Below is a listing of the input/output files which are accessed by the Lanczos program LSVAL for computing singular values of real rectangular matrices on user-specified intervals. Included also is a sample of the input file which LSVAL requires on file 5. The parameters in this file are supplied in free format. File 8 contains the data for the rectangular mxn matrix A.

```
Sample Specifications for Input/Output Files for LSVAL
----------------------------------------------------------------
 LSVAL EXEC FOR LANCZOS SINGULAR VALUE CALCULATIONS
FI 06 TERM
FILEDEF  1 DISK &1        NSHISTOR  A (RECFM F LRECL 80 BLOCK 80
FILEDEF  2 DISK &1        SVHISTOR  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 LSVAL     INPUT     A (RECFM F LRECL 80 BLOCK 80
FILEDEF  8 DISK &1        INPUT     A (RECFM F LRECL 80 BLOCK 80
FILEDEF 11 DISK &1        DISTINCT  A (RECFM F LRECL 80 BLOCK 80
LOAD   LSVAL   LSSUB    LSMULT
----------------------------------------------------------------
```

```
Sample Input File for LSVAL
----------------------------------------------------------------------
 LANCZOS SINGULAR VALUE PROCEDURE,
 WITHOUT REORTHOGONALIZATION BUT WITH BIDIAGONALIZATION.
LINE 1      M    N    KMAX   NMEVS    MATNO
         100   100    300       1     2220
LINE 2   SVSEED    RHSEED    MXINIT    MXSTUR
       49302312   7549309        5    100000
LINE 3      ISTART    ISTOP
               0        1
LINE 4      IHIS    IDIST    IWRITE    IPAR
              1        0        1       2
LINE 5  RELTOL(RELATIVE TOLERANCE USED IN 'COMBINING'  GOOD EVALS
   .0000000001
LINE 6  MB(1)   MB(2)   MB(3)   MB(4)  (SIZE OF T(1,MEV) MUST BE EVEN)
          280
LINE 7   NINT    (NUMBER OF BISEC INTERVALS)
            1
LINE 8    LB(1)    LB(2)   LB(3)   LB(4)  (LOWER BOUNDS INTERVALS)
            0.0
LINE 9    UB(1)    UB(2)   UB(3)   UB(4)  (UPPER BOUNDS INTERVALS)
           1.0
----------------------------------------------------------------------
```

Below is a listing of the input/output files which are accessed by the Lanczos program for computing singular vectors, LSVEC. Included also is a sample of the input file which LSVEC requires on file 5. The parameters in this file are supplied in free format.

File 8 contains the data for the rectangular mxn matrix A. LSVEC computes singular vectors for each of a user-specified subset of the singular values computed by the companion program LSVAL.

```
Sample Specifications of the Input/Output Files for LSVEC
---------------------------------------------------------------
 LSVEC EXEC TO RUN LANCZOS SINGULAR VECTOR PROGRAM
FI 06 TERM
FILEDEF  2 DISK &1       SVHISTOR  A (RECFM F LRECL 80 BLOCK 80
FILEDEF  3 DISK &1       GOODSV    A (RECFM F LRECL 80 BLOCK 80
FILEDEF  4 DISK &1       ERRINV    A (RECFM F LRECL 80 BLOCK 80
FILEDEF  5 DISK LSVEC    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   LSVEC   LSSUB   LSMULT
---------------------------------------------------------------
```

```
Sample Input File for LEVEC
-----------------------------------------------------------------------
 LSVEC SINGULAR VECTORS, NO REORTHOGONALIZATION BUT BIDIAGONALIZATION
LINE 1  MATNO      M     N
          100    100    80
LINE 2  MDIMTV    MDIMRV  MBETA (MAX.DIMENSIONS,TVEC,RITVEC AND BETA
         10000     10000   2000
LINE 3      RELTOL
        .0000000001
LINE 4  MBOUND   NTVCON SVTVEC IREAD (FLAGS
             0        1      0     1
LINE 5  TVSTOP   LVCONT ERCONT  IWRITE (FLAGS
             0        1      1       1
LINE 6    RHSEED  (RANDOM GENERATOR SEED FOR STARTING VECTOR IN INVERM)
        45329517
-----------------------------------------------------------------------
```