Volume 2

Documentation and Listings
Original Lanczos Codes

Lanczos Algorithms for Large Symmetric Eigenvalue Computations

Jane K. Cullum

Ralph A. Willoughby

- ## Volume 1: Theory

  Volume 1: Theory has been republished verbatim in 2002 as Volume 41 in the SIAM Book Series, Classics in Applied Mathematics. SIAM Publications, Philadelphia, PA. ISBN 0-89871-523-7. (A short Errata for Volume 1 is included in this web version of Volume 2 as Chapter 10.)

- ## Uni-Processor and Parallel/Fortran90 Versions

  Currently, Leonard Hoffnung (Math. Dept, U. Kentucky), Spencer Shellman, (Comp. Sc. Dept, U. Utah) and Jane Cullum (*cullumj@lanl.gov*) are working on uni-processor and on MPI parallel Fortran90 versions of the codes contained in Volume 2. The Hoffnung and Shellman contributions are supported currently by a U.S. Department of Energy, Office of Science, MICS, Los Alamos AMS Program grant. The resulting codes will be made available via the Netlib software repository.

- ## Matrix Size

  This book was published 17 years ago. Computers of today are orders of magnitude faster and have orders of magnitude more memory and storage than those which were availabe when this book was written. Seventeen years ago, a matrix of size $10,000$ was considered very large. Since 1985 some of the algorithms which are included in this book have been used on problems of size a million or more. The requirements are *accurate* matrix computations and *sufficient* computer arithmetic precision.

- **Thanks to Leonard Hoffnung for his help in converting ancient 'script' files for Volume 2.**

Contents

Chapter 1

Lanczos procedures

1.1 Introduction

The FORTRAN codes contained in this volume are designed for computing eigenvalues and eigenvectors or singular values and singular vectors of large, sparse matrices. Large means of order several hundred to perhaps 10,000. The largest matrix which we tested was real symmetric and had order 4900. This book is divided into 9 chapters. In this first chapter we give a brief description of Lanczos eigenelement procedures and then make some comments about what the Lanczos codes in this book can and cannot be expected to compute. Detailed analyses of the ideas used in these procedures are contained in Volume 1 of this book.

Chapters 2 through 7 contain procedures which are based upon the single-vector Lanczos recursion with no reorthogonalization of any kind. Six different classes of problems are addressed in these 6 chapters: Eigenelement computations for

1. Real symmetric matrices (Chapter 2)

2. Hermitian matrices (Chapter 3)

3. Factored inverses of real symmetric matrices (Chapter 4)

4. Real symmetric, generalized problems (Chapter 5)

5. Nondefective, complex symmetric matrices (Chapter 7)

6. Singular value and vector computations for real, rectangular matrices (Chapter 6).

Chapters 8 and 9 contain Lanczos procedures which are based upon 'block' versions of the Lanczos recursions. These iterative block procedures include some reorthogonalization within each iteration, but this reorthogonalization is limited to reorthogonalizations w.r.t. certain vectors in each first Lanczos block.

The single-vector procedures can be used to compute anywhere from a very few to very many eigenvalues (singular values). These eigenvalues (singular values) need not be at the extremes of the spectrum. For some matrices it is even possible to compute all of the eigenvalues. The iterative block procedures can only be used to compute a few extreme eigenvalues of the specified matrix. The single vector codes consist of two phases. First eigenvalues or singular values are computed and then corresponding eigenvectors or singular vectors are computed. The iterative 'block' codes compute eigenvalues and corresponding eigenvector approximations simultaneously. Block codes for computing singular values are not included in this book. See for example, Golub, Luk, and Overton [13] for an example of such a block algorithm.

With three exceptions which are given below, each Chapter 2 through 9 contains the following types of information for the particular class of problems considered in that chapter: documentation; main program(s); LANCZS subroutine for computing Lanczos matrices; sample matrix-vector multiply and/or solve subroutines; other subroutines needed by the codes in that chapter; and definitions of the files used by the programs together with sample input files. Because of the similarities between the variables, flags, etc., the documentation for the codes contained in Chapters 2, 3, 4, and 5 was combined and is contained in Section 2.2 of Chapter 2. The codes in Chapters 2, 3, 4, and 5 use essentially (with 2 exceptions) the same set of 'other or additional subroutines' so these subroutines were combined and are given only in Chapter 2, Section 2.6. Similarly, the block codes in Chapters 8 and 9 use the same set of additional subroutines and these are given only in Section 8.5. Some additional optional, preprocessing codes are also provided, and again each of these is included in only one of the chapters and not in each of the ones where it might be useful.

Each set of codes contains many write statements. These write statements serve two major functions: to provide consistency checks on the information supplied by the user, and to provide running commentary on the progress of the computations. Much of the code has been modularized to help make the program logic more transparent to the user. These codes are not designed as efficiently as they could be. Many internal comments have been included. Numerous consistency checks have been used to verify that the user has set up the procedure properly. Basically, we have compromised some efficiency for safety and robustness.

Each LANCZS subroutine together with the corresponding sample matrix-vector multiply and solve subroutines are in files labelled as *MULT. For example in Chapter 2 where real symmetric matrices are discussed this file is labelled LEMULT. The user should note that within a given *MULT file, each sample USPEC* and *MATV subroutine has been given two names so that these subroutines can co-exist with similar subroutines for other test matrices. However, two different *MULT files cannot co-exist because subroutine names are reused in going from one category of matrices to another category. In particular for the codes in Chapters 2, 3, and 7, the matrix-vector multiply subroutine is called CMATV. Moreover, in all of the chapters, the matrix specification subroutines are called USPEC. This reuse of names makes it easier for the user to pass from one set of codes to another. Furthermore, from category to category, subroutines with similar function were typically given the same name. For example, all of the subroutines which generate families of Lanczos matrices are named LANCZS. There are two BISEC bisection subroutines for computing eigenvalues of real symmetric tridiagonal matrices, one for Chapters 2, 3, 4, and 5 and the other one is for Chapter 6. If these sets of codes had to co-exist in one computer file, then it would be necessary for the user to devise a scheme for renaming those subroutines which have the same names.

With respect to portability, each of these programs and subroutines has been individually checked for portability by the PFORT Verifier [22], but the communications between these subroutines have not been checked. Obvious problems with portability like non-Fortran items in the format statements have all been removed. However, certain nonportable constructions have been retained because they make the programs somewhat easier to use. The header of each of the programs contains a list of those constructions in that program which were identified by the PFORT verifier as being nonportable. These headers can be used to locate the nonportable items so that if necessary they can be modified. A list of most of the nonportable items and the reasons for retaining them are given in Table 1.1.

The single vector Lanczos codes in Chapters 2 through 7 are essentially self-contained. The user must provide the matrix-vector multiply and/or solve subroutines which are required by these codes, together with a matrix specification subroutine which defines, dimensions and initializes the matrix which will be used by the Lanczos procedure. The sample matrix-specification subroutines and sample matrix-vector multiply and solve subroutines contained with these codes can be modified and used if appropriate or they can be replaced completely. All of these procedures require a random number generator subroutine, inner product subroutines, and a subroutine to mask underflow. These procedures assume that each time the random number generator is called that the seed for this generator is automatically reset to a different value.

The iterative 'block' Lanczos codes in Chapters 8 and 9 require matrix specification and matrix-vector

Nonportable Construction	Where Used	Why Used
Entry	Passes storage locations of arrays and parameters needed to define user-specified matrix from subroutine USPEC where arrays are dimensioned and initialized to the corresponding matrix-vector multiply or solve subroutine.	Codes do not need to 'see' the user-specified matrix. Codes need only output from matrix-vector multiply or solve subroutines for the matrix being used. User does not have to alter the calling sequences to these subroutines every time the number or kind of arrays needed to define the given matrix is changed.
Formats (20A4) and (4Z20)	(20A4) is used to read and write explanatory comments within the main programs and in sample USPEC subroutines. Machine format (4Z20) is used to read in and write out the Lanczos tridiagonal matrices generated and other quantities for which conversion errors could cause numerical problems.	Allows the user to easily modify headers describing the matrix and code being used. Prevents format conversion errors incurred in input/output conversions.
Free Format Read (5,*)	Used in main program and in sample USPEC subroutines on read-ins of user-specified parameters from input file 5.	Ease of input. User does not have to have the input values properly aligned in the input file.
Complex*16 Variables	Used only in the Hermitian and in the complex symmetric Lanczos codes.	Computations require double precision complex arithmetic.
Specification of Machine Epsilon	Used in main programs	Required to define tolerances used at various points in the computations.

Table 1.1: Nonportable Constructions Used in the Codes

multiply and solve subroutines very similar to those used in the single vector codes, plus the same type of random number generating subroutine, inner product subroutine, and mask subroutine. However, as implemented here the block codes are not self-contained. These codes call two subroutines from the EISPACK Library [23, 8], TRED2 and IMTQL2, which are used repeatedly to compute the eigenvalues and eigenvectors of the small Lanczos matrices generated on each iteration of the block procedures. The user can of course replace these calls by calls to subroutines which perform similar functions, if the EISPACK Library is not available.

The optional preprocessing programs in Sections 2.7, 4.5, 6.7, and 7.7 are stand-alone (if one includes the programs which must be supplied by the user), except for the subroutine PERMUT given in Section 4.5. PERMUT can be used in conjunction with the procedures in Chapters 4, 5, and 9. It calls the SPARSPAK Library [9] (A. George, J. Liu, E. Ng, U. Waterloo) to try to determine a reordering of the given sparse matrix for which the sparsity of the given matrix translates into a sparse factorization of the reordered matrix.

1.2   What are Lanczos procedures?

Lanczos procedures for computing eigenvalues and eigenvectors of real symmetric matrices are based upon one or more variants of the basic single-vector Lanczos recursion for tridiagonalizing a real symmetric matrix A. Given a starting vector v_1 which is typically-generated randomly, the Lanczos recursion implements a Gram-Schmidt orthogonalization of the matrix-vector products Av_i corresponding to the Lanczos vectors v_i generated by the recursion. See for example Bjorck [1]. Specifically, we have that for $i = 2, \ldots, m$,

$$\beta_{i+1}v_{i+1} = Av_i - \alpha_i v_i - \beta_i v_{i-1} \tag{1.2.1}$$

where $\alpha_i \equiv v_i^T A v_i$ and $\beta_{i+1} \equiv v_{i+1}^T A v_i$. By definition $\alpha_i v_i$ and $\beta_i v_{i-1}$ are the projections of Av_i onto the two most recently-generated Lanczos vectors v_i and v_{i-1}. In practice to improve the numerical stability of this recursion, the above formulas are replaced by the following ones.

$$\alpha_i \equiv v_i^T (Av_i - \beta_i v_{i-1}) \text{ and } \beta_{i+1} \equiv \|Av_i - \alpha_i v_i - \beta_i v_{i-1}\|. \tag{1.2.2}$$

The α_i as defined in Eqn(1.2.2) correspond to a modified Gram-Schmidt orthogonalization procedure. The formula for β_{i+1} given in Eqn(1.2.2) is theoretically equivalent to the one given with Eqn(1.2.1). However, it is superior numerically because this choice directly controls the sizes of the Lanczos vectors. See Paige [19].

Rewriting Eqn(1.2.1) in matrix form, we obtain

$$AV_j = V_j T_j + \beta_{j+1}v_{j+1}e_j^T \tag{1.2.3}$$

where T_j denotes the real symmetric tridiagonal Lanczos matrix of order j whose diagonal entries are the scalars α_i, $1 \leq i \leq j$, and whose subdiagonal (superdiagonal) entries are the scalars β_{i+1}, $1 \leq i \leq j - 1$, generated by the Lanczos recursion. In Eqn(1.2.3), $V_j = (v_1, v_2, \ldots, v_j)$, the matrix whose columns are the Lanczos vectors generated by the recursion, and e_j is the coordinate vector whose j-th component is 1 and whose other components are 0.

It is easy to demonstrate by induction that in exact arithmetic each set of vectors V_j generated by the recursion in Eqns(1.2.1) and (1.2.2) is an orthonormal set. Therefore for any A-matrix with n distinct eigenvalues and any starting vector v_1 which has a projection on every eigenspace of A, we have that for each $j \leq n$,

$$T_j = V_j^T A V_j. \tag{1.2.4}$$

Thus the symmetric tridiagonal matrices T_j are representations of the projections of the given matrix A onto the subspaces spanned by the corresponding sets of Lanczos vectors V_j. The eigenvalues of these matrices are the eigenvalues of the A-matrix restricted to these subspaces. Since the Lanczos vectors are obtained by orthogonalizing vectors of the form $\{v_1, Av_1, A^2v_1, \ldots, \}$, we expect the eigenvalues of the T_j to provide good approximations to some of the eigenvalues of A, if j is sufficiently large. Clearly, at least theoretically, if we extend the recursion to $j = n$, then the eigenvalues of T_n will be the eigenvalues of A. T_n is simply an orthogonal transformation of A and must therefore have the same eigenvalues as A. Moreover, any Ritz vector $V_j u$ obtained from an eigenvector u of some T_j is an approximation to a corresponding eigenvector of A.

Basic steps in any Lanczos procedure for computing eigenvalues and eigenvectors of 'symmetric' matrices are the following.

1. Use a variant of the Lanczos recursion to transform the given 'symmetric' matrix A into a family of 'symmetric' tridiagonal matrices of varying sizes.

2. Compute eigenvalues and eigenvectors of certain members of this family. Because of the real symmetric tridiagonal structure this is a much simpler problem than computing the eigenvalues and eigenvectors of A directly.

3. Take some or all of these eigenvalues as approximations to eigenvalues of A and map the corresponding eigenvectors of the tridiagonal matrix into Ritz vectors for the matrix A.

4. Use these Ritz vectors as approximations to the eigenvectors of A.

The Lanczos recursion in Eqn(1.2.1) has several properties which make it particularly attractive for dealing with large but sparse matrices. First the given matrix enters the recursion only through the matrix-vector multiply terms Av_i. Thus contrary to what is done in the standard methods for solving small or medium size eigenvalue problems, see for example EISPACK [23, 8], the given matrix is not explicitly modified. The user must provide only a subroutine which computes Ax for any given vector x. If the matrix A is sparse, this computation can be done using an amount of storage that is only linear in the size of the matrix instead of quadratic. Second, the recursion uses only the two most recently-generated Lanczos vectors. The Gram-Schmidt orthogonalization of an arbitrary set of vectors would require that at any given stage in the process that all of the vectors which have already been orthogonalized be available for orthogonalizing each additional vector as it is considered. Thus, the storage requirements for implementing the basic Lanczos recursion are minimal. If we use Eqns(1.2.1) and (1.2.2) then only 2 n-vectors are needed for the two most recently-generated Lanczos vectors plus storage for the α and β arrays.

There are however numerical problems if only a simple direct implementation of this recursion is programmed. In general such an implementation yields Lanczos matrices which have extra eigenvalues in addition to the 'good' eigenvalues which are approximations to eigenvalues of A. These extraneous or 'spurious' eigenvalues are caused by the losses in the orthogonality of the Lanczos vectors which in turn are caused by the combination of the roundoff errors resulting from the finite computer arithmetic and the convergence (as j is increased) of eigenvalues of the Lanczos matrices to eigenvalues of the original matrix A. This interaction between the computer arithmetic and the convergence of eigenvalues is discussed in Paige [17, 20].

During the past $5-10$ years many different types of Lanczos eigenelement algorithms have been proposed. See Volume 1, Chapter 2 of this book for a brief survey of the literature. Most of these procedures incorporate modifications to the basic Lanczos recursion in Eqns(1.2.1) and (1.2.2) which force the Lanczos vectors to stay nearly orthonormal. These approaches require either the repeated computation of Ritz vectors or the repeated reorthogonalization of the Lanczos vectors as they are generated or some combination of these two computations. In either case as the size of the Lanczos matrix generated is increased to be able to compute more eigenvalues, the associated Ritz vectors or the Lanczos vectors needed for

the reorthogonalizations require more and more storage. These modifications often work well but destroy much of the simplicity of the basic procedure, and because of the added storage requirements resulting from the reorthogonalizations they limit the number of eigenelements which can be computed.

The approach which we have chosen and which is implemented in the enclosed FORTRAN programs in Chapters 2 through 7 is not to force the orthogonality of the Lanczos vectors by reorthogonalizing, but to work directly with the basic Lanczos recursion, accepting the losses in orthogonality, and then unraveling the effects of these losses. This approach allows us to retain the basic simplicity of the Lanczos recursion, to minimize the storage requirements, and to therefore maximize the number of eigenvalues of A which can be computed. In our approach in the single-vector algorithms in Chapters 2 through 7, Ritz vectors are not computed until after the eigenvalues have been computed accurately. Consequently, the basic storage requirements for our eigenvalue (singular value) algorithms are only a small multiple of the size of the largest Lanczos matrix used in the computations. Thus, we can compute many eigenvalues of very large but sparse matrices. Depending upon what is to be computed and upon the eigenvalue distribution in the given matrix A, the sizes of the Lanczos matrices used in these computations may be much smaller or considerably larger than the original A-matrix. However the Lanczos matrices generated by the procedures in Chapters 2 through 6 are real symmetric and tridiagonal so that these matrices can be very large and still not present insurmountable computational problems. Eigenvalue and eigenvector computations for such matrices require minimal amounts of storage and fairly reasonable numbers of arithmetic operations.

The computational problems which arise from not maintaining near orthogonality of the Lanczos vectors and which we must address in our single-vector codes are of two types. First and most importantly, we must deal with the question of sorting the eigenvalues of the Lanczos matrices into 2 classes, one corresponding to the 'good' eigenvalues which are approximations to the eigenvalues of A and the other corresponding to the extra or 'spurious' eigenvalues caused by the losses in orthogonality. The identification test used for doing this is discussed in Volume 1, Chapter 4, Section 4.5. For the procedures discussed in Chapters 2 through 6, this identification test is an integral and inexpensive part of the eigenvalue (singular value) computations. For the complex symmetric procedure discussed in Chapter 7 this test is handled in a considerably less eloquent manner and is expensive.

The second but much less serious difficulty we must address is the question of false multiplicities. The multiplicity of a particular 'good' eigenvalue as an eigenvalue of the Lanczos matrices is not related to the multiplicity of that eigenvalue as an eigenvalue of the A-matrix. 'Good' eigenvalues may replicate many times as eigenvalues of a Lanczos matrix, but be only simple eigenvalues of the original A-matrix. Thus, these single-vector procedures cannot directly determine the true multiplicities of the computed 'good' eigenvalues. Of course, this latter comment is also applicable to any single-vector Lanczos procedure not just to our procedures. Theoretically, at most one eigenvector for each distinct eigenvalue of the A-matrix can be obtained using the single-vector Lanczos recursion given in Eqns(1.2.1) and (1.2.2). (This of course is not true for iterative block Lanczos procedures.) It is interesting to note however that if the Lanczos recursion is used without any reorthogonalization, then it can yield sets of linearly independent eigenvectors for eigenvalues which are multiple in the A-matrix. The amount of work required to compute these additional eigenvectors depends upon the particular matrix in question and upon the particular eigenvalue. The codes provided in Chapters 2 through 7 of this book do not however incorporate this capability.

The iterative 'block' Lanczos procedures for real symmetric matrices given in Chapters 8 through 9 are based upon a block version of the Lanczos recursion

$$Q_{j+1}B_{j+1} = AQ_j - Q_jA_j - Q_{j-1}B_j^T \qquad (1.2.5)$$

for $j = 1, 2, \ldots, s$ where Q_1 is $n \times q$ and the coefficient matrices A_j and B_{j+1} are block analogs of the scalar coefficients in the single-vector Lanczos recursion in Eqns(1.2.1) and (1.2.2). The number of blocks s used on each iteration is chosen such that $qs \ll n$, where n is the order of the given A-matrix and q is chosen such that $q \geq q'$, the number of eigenvalues and eigenvectors desired. The Lanczos matrices are real symmetric, block tridiagonal matrices. In Eqn(1.2.5) we used Q_j instead of V_j because in our block Lanczos procedures we maintain near-orthogonality of the blocks generated within each iteration

by incorporating reorthogonalization of the blocks of Lanczos vectors with respect to certain vectors in the first Lanczos block.

The 'block' procedures provided in Chapters 8 and 9 are really hybrid algorithms, something between a true block Lanczos procedure, see for example, Cullum and Donath [4, 3] and Chapter 7 in [5], and the single-vector Lanczos procedures given in Chapters 2 through 7. The sequence of 'blocks' generated on each iteration of this hybrid method has the property that the first Q-block contains at least as many vectors as the user is trying to compute, but the second and succeeding blocks each contain only one vector. The corresponding resulting Lanczos matrices are not block tridiagonal. Each Lanczos matrix has a border of blocks in the first q rows and columns and is tridiagonal below this border.

At the beginning of each chapter, a brief description is given of the particular variant of the Lanczos recursion used in the Lanczos codes included in that chapter, along with some additional comments relevant to the particular types of problems being considered in that chapter.

1.3 Comments and disclaimers

The single-vector Lanczos procedures contained in Chapters 2 through 7 do not behave like standard eigenelement procedures. Their behavior is both non-classical and somewhat unorthodox. If one of these codes were run on two different kinds of computers but with the same original matrix and the same initial specifications, the computed results could be quite different. A primary cause for such differences can of course be a difference in the starting vector caused by a difference in the random number generators. However even if the same starting vector were read in, the results would almost surely differ due to the differences in the computer arithmetic. In practice, the Lanczos matrices generated on two different kinds of computers may agree for a certain number of Lanczos steps but will begin to diverge upon the convergence of one or more of the eigenvalues of these Lanczos matrices to eigenvalues of the A-matrix. If after a reasonable number of steps in the Lanczos recursion we were to compare the entries in the Lanczos matrices generated by the two different computers, the values would probably be very different.

Furthermore, if we were to compute the eigenvalues of the two sets of Lanczos matrices for various sizes and 'spurious' eigenvalues were present, then these spurious eigenvalues would be different and even appear in different portions of the spectrum. In fact, prior to the convergence of a particular 'good' eigenvalue, the values of that good eigenvalue, in terms of how accurate it is at any given stage in the computations, may differ. However once a 'good' eigenvalue in either set has converged, that 'good' eigenvalue will agree with a true eigenvalue of the original user-specified matrix to as many digits as can be expected.

Therefore, if the user carries out the sample eigenvalue computation provided in Chapter 2, he/she should not be alarmed or surprised if the output from the computer being used does not agree with what is shown in the sample, as long as the converged 'good' eigenvalues agree. Actually one may observe different rates of convergence on different kinds of computers, depending upon the computer arithmetic. With increased arithmetic precision in all of the computations, these procedures may converge more rapidly. With decreased precision, they will converge less rapidly. All of our codes use double precision arithmetic (for an IBM 3083) and any precision less than that is not recommended.

Each of these procedures requires the user to supply either a matrix-vector multiply subroutine or a matrix-vector solve subroutine. (Both types of subroutines are required for the codes in Chapter 5 .) Such subroutines should perform the required computations rapidly and accurately, taking advantage of any special properties or structure in the given matrix. Our Lanczos programs see the original matrix as the outputs of these subroutines. The codes provided include sample matrix-vector multiply subroutines for a general sparse 'symmetric' matrix given in a particular sparse format. These are available for the user to use or modify as desired. Note that similar programs are also provided for the singular value/vector computations. Accuracy is important in these subroutines because consistency must be maintained in the information being provided to the LANCZS subroutine which is generating the Lanczos matrices. There is no built-in mechanism for preserving symmetry. Therefore, the matrix-vector multiply and solve subroutines must be coded with care. Without such consistency the Lanczos codes will not function

properly.

The convergence characteristics of the two types of Lanczos procedures considered are quite different. These differences are discussed in Chapters 4 and 7 of Volume 1 of this book. However, in both cases, the degree of difficulty in computing the desired eigenvalues depends upon the eigenvalue gaps. For the single-vector procedures the primary factor in determining whether or not it is feasible to compute either large numbers of eigenvalues or the eigenvalues with the smallest gaps, is the gap ratio, the ratio of the largest gap between two neighboring eigenvalues to the smallest such gap. The smaller this ratio, the easier it is to compute all of the eigenvalues of the given matrix. The larger this ratio, the harder it is to compute those eigenvalues with the smallest gaps. The locations of the desired eigenvalues in the spectrum of the given matrix also play a significant role in the rate of convergence of individual eigenvalues. Both types of Lanczos procedures favor extreme eigenvalues. The iterative block codes, in fact, can only compute a few extreme eigenvalues. However for the single-vector codes, it is possible for interior eigenvalues which have gaps which are significantly larger than the gaps for some of the extreme eigenvalues to converge prior to the convergence of those extreme eigenvalues. Examples of the convergence achievable are given in Volume 1, Chapter 4 of this book.

The convergence of the iterative block procedures depends primarily upon the gaps between the eigenvalues being computed and the closest eigenvalue not being approximated, the spread of the matrix, and the overall eigenvalue distribution. The block procedures discussed in Chapters 8 and 9 are iterative and the codes track the rate of convergence. If the observed rate is too slow (as specified by the user), these block procedures will terminate without achieving convergence. The user then has the option of restarting the block procedure with a different choice of parameters and using the current approximation to the basis for the desired eigenspace as the starting vectors.

Thus, the amount of work required for a particular eigenelement computation for a given matrix using a particular method depends directly upon the eigenvalue distribution in that matrix and upon which portion of the spectrum is being computed. Some problems are 'easy', others are hard. Therefore failure can occur, in the sense that these procedures may not be able to compute the information desired by the user within the computational bounds specified by the user. However the single-vector Lanczos procedures, even in 'failure', provide a great deal of information about the eigenvalue spectrum of the given matrix.

In deciding which procedure to use on a given problem, our preference is a single-vector procedure, although the iterative block procedures can often quickly provide simultaneously the desired eigenvalues and eigenvectors. If the user wants extreme eigenvalues and the user knows or suspects that one or more of these is multiple, then the block procedure is probably preferable. More details about the Lanczos procedures contained in this book can be found in Volume 1. Any questions about these programs including the question of obtaining copies of these codes or of problems with these codes, should be addressed directly to the authors. We hope that these codes will prove useful in many different applications in the engineering and scientific community.

Chapter 2

Real Symmetric Matrices

2.1 Introduction

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

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

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

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

1. Real symmetric matrices have complete eigensystems. That is, the dimension of the eigenspace corresponding to any given eigenvalue of the given matrix A is the same as the multiplicity of that eigenvalue as a root of the characteristic polynomial of A.

2. For any two distinct eigenvalues of A, λ and μ , and corresponding eigenvectors x and y, $x^T y = 0$. Thus, eigenvectors corresponding to different eigenvalues are orthogonal, and we can construct an eigenvector basis which is orthonormal. Vectors are orthonormal if they are orthogonal and each has a Euclidean norm of 1. (The Euclidean norm of a vector is just the square root of the sum of the squares of its components.)

3. Small perturbations in the matrix cause only small perturbations in the eigenvalues. Of the classes of matrices which we consider, the class of real symmetric matrices is the most well-behaved and thus the 'easiest'.

The Lanczos codes contained in this chapter correspond to the most straight-forward implementation of the Lanczos recursion included in this book. These codes can be used to compute either a very few or very many of the distinct eigenvalues of the given real symmetric matrix. As the documentation in the next section indicates, the A-multiplicity of a given computed 'good' Lanczos eigenvalue can be obtained only with additional computation, and the modifications required to do this additional computation are not included in these versions of the codes. This implementation uses the basic Lanczos recursion given in Eqns (1.2.1) and (1.2.2) in Section 1.2 of Chapter 1 to generate a family of real symmetric, tridiagonal

matrices (T-matrices) whose sizes are specified by the user. There is no reorthogonalization of the Lanczos vectors at any stage in any of the computations.

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

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

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

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

```
C-----LEVALHED-----------------------------------------------------------LEV00010
C  Authors:  Jane Cullum and Ralph A. Willoughby (Deceased)      LEV00020
C            Los Alamos National Laboratory                      LEV00030
C            Los Alamos, New Mexico 87544                        LEV00040
C                                                                LEV00050
C            E-mail:  cullumj@lanl.gov                           LEV00060
C                                                                LEV00070
C  These codes are copyrighted by the authors.  These codes      LEV00080
C  and modifications of them or portions of them are NOT to be   LEV00090
C  incorporated into any commercial codes or used for any other  LEV00100
C  commercial purposes such as consulting for other companies,   LEV00110
C  without legal agreements with the authors of these Codes.     LEV00120
C  If these Codes or portions of them are used in other scientific or LEV00130
C  engineering research works the names of the authors of these codes LEV00140
C  and appropriate references to their written work are to be    LEV00150
C  incorporated in the derivative works.                         LEV00160
C                                                                LEV00170
C  This header is not to be removed from these codes.            LEV00180
C                                                                LEV00190
C        REFERENCE: Cullum and Willoughby, Chapters 1,2,3,4      LEV00200
C        Lanczos Algorithms for Large Symmetric Eigenvalue ComputationsLEV00210
C        VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in LEV00220
C        Applied Mathematics, 2002. SIAM Publications,           LEV00230
C        Philadelphia, PA. USA                                   LEV00240
C                                                                LEV00250
C                                                                LEV00260
C-----DOCUMENTATION FOR SINGLE-VECTOR------------------------------------LEV00270
C                                                                LEV00280
C     LANCZOS EIGENVALUE/EIGENVECTOR PROGRAMS FOR                LEV00290
C     (1) REAL SYMMETRIC MATRICES                                LEV00300
C     (2) HERMITIAN MATRICES                                     LEV00310
C     (3) FACTORED INVERSES OF REAL SYMMETRIC MATRICES           LEV00320
C     (4) REAL SYMMETRIC, GENERALIZED PROBLEMS WHERE ONE OF THE  LEV00330
C         MATRICES IS POSITIVE DEFINITE AND ITS CHOLESKY FACTORS ARE LEV00340
C         AVAILABLE                                              LEV00350
C                                                                LEV00360
C                                                                LEV00370
C                                                                LEV00380
C-----------------------------------------------------------------------LEV00390
C                                                                LEV00400
C     REAL SYMMETRIC MATRICES:                                   LEV00410
C                                                                LEV00420
C     GIVEN A REAL SYMMETRIC MATRIX A OF ORDER N THE THREE SETS OF LEV00430
C     FORTRAN FILES LABELLED LEVAL, LESUB, AND LEMULT CAN BE USED TO LEV00440
C     COMPUTE DISTINCT EIGENVALUES OF THE USER-SPECIFIED MATRIX  LEV00450
C     IN USER-SPECIFIED INTERVALS.                               LEV00460
C                                                                LEV00470
C     CORRESPONDING EIGENVECTORS FOR SELECTED, COMPUTED EIGENVALUES CAN LEV00480
C     BE COMPUTED USING THE SETS OF FILES LABELLED LEVEC, LESUB, AND LEV00490
C     LEMULT.                                                    LEV00500
C                                                                LEV00510
```

```
C                                                                        LEV00520
C       HERMITIAN MATRICES:                                              LEV00530
C                                                                        LEV00540
C       GIVEN A HERMITIAN MATRIX A OF ORDER N THE THREE SETS OF          LEV00550
C       FORTRAN FILES LABELLED HLEVAL, LESUB, AND HLEMULT CAN BE USED    LEV00560
C       TO COMPUTE DISTINCT EIGENVALUES IN USER-SPECIFIED INTERVALS.     LEV00570
C                                                                        LEV00580
C       CORRESPONDING EIGENVECTORS FOR SELECTED, COMPUTED EIGENVALUES    LEV00590
C       CAN BE COMPUTED USING THE SETS OF PROGRAMS LABELLED HLEVEC,      LEV00600
C       LESUB, AND HLEMULT.                                              LEV00610
C                                                                        LEV00620
C                                                                        LEV00630
C       FACTORED INVERSES OF REAL SYMMETRIC MATRICES:                    LEV00640
C                                                                        LEV00650
C       GIVEN A REAL SYMMETRIC MATRIX A, THE LANCZOS RECURSION IS        LEV00660
C       APPLIED TO THE INVERSE OF A, USING A FACTORIZATION               LEV00670
C       OF A.  THE SETS OF FILES LIVAL, LESUB, AND LIMULT                LEV00680
C       CAN BE USED TO COMPUTE THE DISTINCT EIGENVALUES OF THE           LEV00690
C       INVERSE OF THE A-MATRIX AND OF A IN  USER-SPECIFIED              LEV00700
C       INTERVALS.  THE PROGRAMS ACTUALLY ALLOW ONE TO WORK WITH         LEV00710
C       ANY MATRIX B = PCP' WHERE  C = SO*A + SHIFT*I, WHERE             LEV00720
C       SO AND SHIFT ARE SCALARS CHOSEN BY THE USER AND P IS A           LEV00730
C       PERMUTATION MATRIX CHOSEN SUCH THAT THE FACTORIZATION            LEV00740
C       OF THE B-MATRIX RETAINS SPARSITY.  IN THE                        LEV00750
C       SAMPLE LIMULT SUBROUTINES PROVIDED, SO AND SHIFT MUST BE         LEV00760
C       CHOSEN SO THAT THE RESULTING B-MATRIX IS POSITIVE DEFINITE,      LEV00770
C       AND THE CHOLESKY FACTORS ARE USED TO SOLVE B*U = V.              LEV00780
C       HOWEVER, THE USER CAN EASILY REPLACE THE SAMPLE USPEC AND        LEV00790
C       BSOLV SUBROUTINES PROVIDED BY SUBROUTINES THAT ALLOW THE         LEV00800
C       GENERAL FACTORIZATION L*D*(L-TRANSPOSE).  THESE LANCZOS          LEV00810
C       PROGRAMS APPLY THE LANCZOS RECURSION TO B-INVERSE, USING         LEV00820
C       THE FACTORIZATION PROVIDED.  OPTIONAL PREPROCESSING PROGRAMS     LEV00830
C       PERMUT, LORDER, LFACT, AND LTEST ARE PROVIDED FOR SET-UP PURPOSES.LEV00840
C       PERMUT USES THE SPARSPAK PACKAGE OF A. GEORGE, J. LIU AND        LEV00850
C       E. NG TO OBTAIN A REORDERING OF THE GIVEN MATRIX THAT            LEV00860
C       PRESERVES SPARSENESS ON SUBEQUENT FACTORIZATION.  LORDER         LEV00870
C       CAN BE USED TO REORDER A GIVEN MATRIX, USING A GIVEN             LEV00880
C       PERMUTATION.  LFACT CAN BE USED TO COMPUTE THE CHOLESKY          LEV00890
C       FACTORS OF A GIVEN POSITIVE DEFINITE B-MATRIX.  LTEST CAN        LEV00900
C       BE USED TO ESTIMATE THE NUMERICAL CONDITION OF THE               LEV00910
C       B-MATRIX.                                                        LEV00920
C                                                                        LEV00930
C       CORRESPONDING EIGENVECTORS FOR SELECTED, COMPUTED                LEV00940
C       EIGENVALUES CAN BE COMPUTED USING THE SETS OF FILES              LEV00950
C       LABELLED LIVEC, LESUB, AND LIMULT.                               LEV00960
C                                                                        LEV00970
C       GENERALIZED REAL SYMMETRIC PROBLEMS:                             LEV00980
C                                                                        LEV00990
C       GIVEN 2 REAL SYMMETRIC MATRICES A AND B WHERE IN ADDITION B IS   LEV01000
C       POSITIVE DEFINITE AND ITS CHOLESKY FACTORS ARE AVAILABLE,        LEV01010
C       THE SETS OF FILES LGVAL, LGMULT, AND LESUB CAN BE USED           LEV01020
C       TO COMPUTE THE DISTINCT EIGENVALUES OF THE GENERALIZED           LEV01030
C       PROBLEM A*X = EVAL*B*X.                                          LEV01040
C                                                                        LEV01050
C       CORRESPONDING EIGENVECTORS CAN BE COMPUTED USING THE PROGRAMS    LEV01060
```

```
C     LGVEC, LGMULT, AND LESUB.  NOTE THAT THE PREPROCESSING PROGRAMS  LEV01070
C     AVAILABLE FOR USE IN CASE (3) (PERMUT, LORDER, LFACT, AND LTEST)  LEV01080
C     CAN ALSO BE USED IN THIS CASE TO OBTAIN A SUITABLE PERMUTATION,   LEV01090
C     AND A FACTORIZATION OF THE RESULTING B-MATRIX.  THE A-MATRIX      LEV01100
C     CAN THEN BE PERMUTED USING LORDER.                               LEV01110
C                                                                      LEV01120
C                                                                      LEV01130
C     THESE PROGRAMS ALL USE LANCZOS TRIDIAGONALIZATION WITHOUT        LEV01140
C     REORTHOGONALIZATION TO GENERATE REAL SYMMETRIC TRIDIAGONAL       LEV01150
C     MATRICES, T(1,MEV), OF ORDER MEV.  SUBSETS OF THE EIGENVALUES OF  LEV01160
C     THESE T-MATRICES, LABELLED AS THE 'GOOD EIGENVALUES', YIELD      LEV01170
C     APPROXIMATIONS TO THE DESIRED EIGENVALUES.  CORRESPONDING        LEV01180
C     RITZ VECTORS ARE APPROXIMATIONS TO THE DESIRED EIGENVECTORS.     LEV01190
C     NOTE THAT FOR CASE (4) THE GENERALIZED LANCZOS RECURSION         LEV01200
C     B*V(I+1)*BETA(I+1) = A*V(I) - B*V(I)*ALPHA(I) - B*V(I-1)*BETA(I)  LEV01210
C     IS USED, ALONG WITH THE B-NORM.                                  LEV01220
C                                                                      LEV01230
C     THE IDEAS USED IN THESE PROGRAMS ARE DISCUSSED IN THE FOLLOWING  LEV01240
C     REFERENCES.                                                      LEV01250
C                                                                      LEV01260
C     1.   JANE CULLUM AND RALPH A. WILLOUGHBY, LANCZOS ALGORITHMS     LEV01270
C          FOR LARGE SYMMETRIC MATRICES, VOLUME ?, PROGRESS IN         LEV01280
C          SCIENTIFIC COMPUTING, EDITORS, G. GOLUB, H.O. KREISS,       LEV01290
C          S. ARBARBANEL, AND R. GLOWINSKI,  BIRKHAUSER BOSTON INC.,   LEV01300
C          CAMBRIDGE, MASSACHUSETTS, 1983.                             LEV01310
C                                                                      LEV01320
C     2.   JANE CULLUM AND RALPH A. WILLOUGHBY, COMPUTING EIGENVECTORS LEV01330
C          (AND EIGENVALUES) OF LARGE, SYMMETRIC MATRICES USING        LEV01340
C          LANCZOS TRIDIAGONALIZATION, LECTURE NOTES IN MATHEMATICS,   LEV01350
C          773, NUMERICAL ANALYSIS PROCEEDINGS, DUNDEE 1979, EDITED BY LEV01360
C          G. A. WATSON, SPRINGER-VERLAG, (1980), BERLIN, PP.46-63.    LEV01370
C                                                                      LEV01380
C     3. IBID, LANCZOS AND THE COMPUTATION IN SPECIFIED INTERVALS OF   LEV01390
C          THE SPECTRUM OF LARGE SPARSE, REAL SYMMETRIC MATRICES, SPARSE LEV01400
C          MATRIX PROCEEDINGS 1978, ED. I.S. DUFF AND G. W. STEWART,   LEV01410
C          SIAM, PHILADELPHIA, PP.220-255, 1979.                      LEV01420
C                                                                      LEV01430
C     4. IBID, COMPUTING EIGENVALUES OF VERY LARGE SYMMETRIC MATRICES- LEV01440
C          AN IMPLEMENTATION OF A LANCZOS ALGORITHM WITHOUT           LEV01450
C          REORTHOGONALIZATION, J. COMPUT. PHYS. 44(1981), 329-358.    LEV01460
C                                                                      LEV01470
C                                                                      LEV01480
C-----PORTABILITY-----------------------------------------------------LEV01490
C                                                                      LEV01500
C                                                                      LEV01510
C     PROGRAMS WERE TESTED FOR PORTABILITY USING THE PFORT VERIFIER.   LEV01520
C     FOR DETAILS OF THE VERIFIER SEE FOR EXAMPLE, B. G. RYDER AND     LEV01530
C     A. D. HALL, "THE PFORT VERIFIER", COMPUTING SCIENCE TECHNICAL    LEV01540
C     REPORT 12, BELL LABORATORIES, MURRAY HILL, NEW JERSEY 07974,     LEV01550
C     (REVISED), JANUARY 1981.                                        LEV01560
C                                                                      LEV01570
C     WITH THE EXCEPTION OF THE PROGRAMS FOR HERMITIAN MATRICES WHICH  LEV01580
C     ARE NOT PORTABLE BECAUSE OF THEIR USE OF COMPLEX*16 VARIABLES,   LEV01590
C     THE OTHER PROGRAMS INCLUDED ARE PORTABLE EXCEPT FOR A FEW        LEV01600
C     CONSTRUCTIONS WHICH, IF NECESSARY, WILL HAVE TO BE MODIFIED      LEV01610
```

```
C     BY THE USER FOR THE PARTICULAR COMPUTER BEING USED.          LEV01620
C                                                                  LEV01630
C     NONPORTABLE CONSTRUCTIONS:                                   LEV01640
C                                                                  LEV01650
C     REAL SYMMETRIC MATRICES:                                     LEV01660
C     IN LEVAL AND IN LEVEC                                        LEV01670
C         1.  DATA/MACHEP STATEMENT                                LEV01680
C         2.  ALL READ(5,*) STATEMENTS (FREE FORMAT)               LEV01690
C         3.  FORMAT(20A4) USED FOR THE EXPLANATORY HEADER ARRAY, EXPLANLEV01700
C         4.  FORMAT(4Z20) USED TO READ AND WRITE ALPHA/BETA FILES.LEV01710
C     IN LEMULT                                                    LEV01720
C         1.  IN CMATV AND  USPEC THE ENTRY THAT PASSES THE STORAGELEV01730
C             LOCATIONS OF THE ARRAYS DEFINING THE USER-SPECIFIED  LEV01740
C             MATRIX.                                              LEV01750
C         2.  IN THE SAMPLE USPEC PROVIDED:  FREE FORMAT (8,*),    LEV01760
C             THE FORMAT (20A4), AND DATA/MACHEP STATEMENT.        LEV01770
C                                                                  LEV01780
C     HERMITIAN MATRICES:                                          LEV01790
C     IN HLEVAL AND IN HLEVEC                                      LEV01800
C         1.  DATA/MACHEP STATEMENT                                LEV01810
C         2.  ALL READ(5,*) STATEMENTS (FREE FORMAT)               LEV01820
C         3.  FORMAT(20A4) USED FOR THE EXPLANATORY HEADER ARRAY, EXPLANLEV01830
C         4.  COMPLEX*16 VARIABLES AND FUNCTIONS SUCH AS DCMPLX.   LEV01840
C         5.  FORMAT (4Z20) USED TO READ AND WRITE ALPHA/BETA FILES.LEV01850
C     IN HLEMULT                                                   LEV01860
C         1.  IN CMATV AND  USPEC THE ENTRY THAT PASSES THE STORAGELEV01870
C             LOCATIONS OF THE ARRAYS DEFINING THE USER-SPECIFIED  LEV01880
C             MATRIX.                                              LEV01890
C         2.  COMPLEX*16 VARIABLES AND FUNCTIONS SUCH AS DCMPLX.   LEV01900
C         3.  IN THE SAMPLE USPEC PROVIDED:  FREE FORMAT (8,*),    LEV01910
C             THE FORMAT (20A4), AND DATA/MACHEP STATEMENT.        LEV01920
C                                                                  LEV01930
C     FACTORED INVERSES OF REAL SYMMETRIC MATRICES:                LEV01940
C     IN LIVAL AND IN LIVEC                                        LEV01950
C         1.  DATA/MACHEP STATEMENT                                LEV01960
C         2.  ALL READ(5,*) STATEMENTS (FREE FORMAT)               LEV01970
C         3.  FORMAT(20A4) USED FOR THE EXPLANATORY HEADER ARRAY, EXPLANLEV01980
C         4.  FORMAT(4Z20) USED TO READ AND WRITE ALPHA/BETA FILES.LEV01990
C     IN LIMULT                                                    LEV02000
C         1.  IN USPEC AND BSOLV, THE ENTRIES THAT PASS            LEV02010
C             THE STORAGE LOCATIONS OF THE ARRAYS DEFINING THE     LEV02020
C             USER-SPECIFIED MATRIX.                               LEV02030
C         2.  IN THE SAMPLE USPEC SUBROUTINES PROVIDED:            LEV02040
C             FORMATS (20A4) AND (4Z20), FREE FORMAT (8,*), AND    LEV02050
C             DATA/MACHEP STATEMENTS.                              LEV02060
C                                                                  LEV02070
C                                                                  LEV02080
C     GENERALIZED SYMMETRIC PROBLEM, B-MATRIX POSITIVE             LEV02090
C     DEFINITE AND CHOLESKY FACTORS AVAILABLE:                     LEV02100
C     IN LGVAL AND IN LGVEC                                        LEV02110
C         1.  DATA/MACHEP STATEMENT                                LEV02120
C         2.  ALL READ(5,*) STATEMENTS (FREE FORMAT)               LEV02130
C         3.  FORMAT(20A4) USED FOR THE EXPLANATORY HEADER ARRAY, EXPLANLEV02140
C         4.  FORMAT(4Z20) USED TO READ AND WRITE ALPHA/BETA FILES.LEV02150
C     IN LGMULT                                                    LEV02160
```

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

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

```
C       LET L DENOTE THE CHOLESKY FACTOR OF THE B-MATRIX USED IN LANCZS.  LEV03270
C       WHEN ISOLV = 1, LSOLV COMPUTES U = L*V.  WHEN ISOLV = 2,          LEV03280
C       LSOLV COMPUTES U = (L-TRANSPOSE)*V.  WHEN ISOLV = 3, LSOLV        LEV03290
C       COMPUTES U = (L-INVERSE)*V.  WHEN ISOLV = 4, LSOLV               LEV03300
C       COMPUTES  U = ((L-TRANSPOSE)-INVERSE)*V.                          LEV03310
C                                                                         LEV03320
C       SAMPLE PROGRAMS ASSUME THAT THE A-MATRIX (CASES (1),(2),(4))      LEV03330
C       IS ON FILE 8 AND STORED IN THE FOLLOWING SPARSE FORMAT:           LEV03340
C       ICOL(K), K = 1,NZL, NUMBER OF SUBDIAGONAL NONZEROS IN COLUMN K.   LEV03350
C       IROW(K), K = 1,NZS, ROW INDEX OF ASD(K).                         LEV03360
C       AD(K), K=1,N, CONTAINS THE DIAGONAL ELEMENTS OF THE A-MATRIX.     LEV03370
C       ASD(K), K=1,NZS  CONTAINS THE SUBDIAGONAL ELEMENTS OF A BY COLUMN.LEV03380
C       NZS = NUMBER OF NONZERO ELEMENTS BELOW THE DIAGONAL OF A          LEV03390
C       NZL = INDEX OF LAST COLUMN WITH NONZERO SUBDIAGONAL ENTRIES       LEV03400
C       N = ORDER OF THE A-MATRIX.                                       LEV03410
C                                                                         LEV03420
C       NOTE THAT THE OPTIONAL PREPROCESSING PROGRAMS PERMUT AND          LEV03430
C       LORDER ASSUME THAT THE GIVEN MATRIX IS ON FILE 8.  CASES (3)      LEV03440
C       AND (4) ASSUME THAT THE SPARSE FACTORIZATION OF B IS STORED ON    LEV03450
C       FILE 7.  THE SAMPLE BSOLV SUBROUTINE SUPPLIED ASSUMES             LEV03460
C       THAT THE B-MATRIX IS POSITIVE DEFINITE AND THAT ITS CHOLESKY      LEV03470
C       FACTOR IS PROVIDED ON FILE 7, STORED IN SPARSE FORMAT IN          LEV03480
C       ARRAYS BD AND BSD.  THE USER CAN EASILY REPLACE THIS SAMPLE       LEV03490
C       BSOLV SUBROUTINE AND THE CORRESPONDING SAMPLE USPEC               LEV03500
C       SUBROUTINE BY SUBROUTINES THAT DEFINE AND USE A GENERAL           LEV03510
C       FACTORIZATION L*D*(L-TRANSPOSE).                                  LEV03520
C                                                                         LEV03530
C       THE SAMPLE USPEC, CMATV (CASES (1) AND (2)), AMATV (CASE (4)),    LEV03540
C       BSOLV (CASE (3)), AND LSOLV (CASE(4)) MUST BE MODIFIED BY         LEV03550
C       THE USER TO ACCOMODATE THE USER-SPECIFIED MATRIX OR MATRICES.     LEV03560
C                                                                         LEV03570
C                                                                         LEV03580
C-----MACHEP-------------------------------------------------------------LEV03590
C                                                                         LEV03600
C                                                                         LEV03610
C       MACHEP IS A MACHINE DEPENDENT PARAMETER SPECIFYING THE RELATIVE   LEV03620
C       PRECISION OF THE FLOATING POINT ARITHMETIC USED.                 LEV03630
C       MACHEP = 2.2 * 10**-16 FOR DOUBLE PRECISION ARITHMETIC ON         LEV03640
C       IBM 370-3081.                                                    LEV03650
C                                                                         LEV03660
C       THE USER WILL HAVE TO RESET THIS PARAMETER TO                     LEV03670
C       THE CORRESPONDING VALUE FOR THE MACHINE BEING USED.  NOTE THAT    LEV03680
C       IF A MACHINE WITH A MACHINE EPSILON THAT IS MUCH LARGER THAN THE  LEV03690
C       VALUE GIVEN HERE IS BEING USED, THEN THERE COULD BE               LEV03700
C       PROBLEMS WITH THE TOLERANCES.                                     LEV03710
C                                                                         LEV03720
C                                                                         LEV03730
C-----SUBROUTINES AND FUNCTIONS USER MUST SUPPLY------------------------LEV03740
C                                                                         LEV03750
C                                                                         LEV03760
C       GENRAN,  FINPRO,  MASK,  USPEC,  AND                              LEV03770
C       CASES (1) AND (2),  CMATV:  CASE (3), BSOLV:                      LEV03780
C       CASE (4),  AMATV AND LSOLV.                                       LEV03790
C                                                                         LEV03800
C       GENRAN = COMPUTES K PSEUDO-RANDOM NUMBERS AND STORES THEM IN      LEV03810
```

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

```
C--------------------------------------------------------------------LEV04370
C                                                                     LEV04380
C                                                                     LEV04390
C-----PARAMETER CONTROLS FOR EIGENVALUE PROGRAMS---------------------LEV04400
C                                                                     LEV04410
C                                                                     LEV04420
C     PARAMETER CONTROLS ARE INTRODUCED TO ALLOW SEGMENTATION OF THE  LEV04430
C     EIGENVALUE COMPUTATIONS AND TO ALLOW VARIOUS COMBINATIONS OF    LEV04440
C     READ/WRITES.                                                    LEV04450
C                                                                     LEV04460
C     THE FLAG ISTART CONTROLS THE T-MATRIX (ALPHA/BETA HISTORY)      LEV04470
C     GENERATION.                                                     LEV04480
C                                                                     LEV04490
C     ISTART = (0,1)  MEANS                                           LEV04500
C                                                                     LEV04510
C             (0) THERE IS NO EXISTING ALPHA/BETA HISTORY AND ONE     LEV04520
C                 MUST BE GENERATED.                                  LEV04530
C                                                                     LEV04540
C             (1) THERE IS AN EXISTING ALPHA/BETA HISTORY AND IT IS   LEV04550
C                 TO BE READ IN FROM FILE 2 AND EXTENDED IF NECESSARY.LEV04560
C                                                                     LEV04570
C     THE FLAG ISTOP CAN BE USED IN CONJUNCTION WITH THE FLAG ISTART TOLEV04580
C     ALLOW SEGMENTATION OF THE EIGENVALUE COMPUTATIONS.             LEV04590
C                                                                     LEV04600
C     ISTOP  = (0,1)  MEANS                                           LEV04610
C                                                                     LEV04620
C             (0) PROGRAM COMPUTES ONLY THE REQUESTED ALPHAS/BETAS,   LEV04630
C                 STORES THEM AND THE LAST 2 LANCZOS VECTORS GENERATED LEV04640
C                 IN FILE 1 AND THEN TERMINATES.  IN CASE (4) THERE   LEV04650
C                 ARE ACTUALLY 3 VECTORS TO BE SAVED.                 LEV04660
C                                                                     LEV04670
C             (1) PROGRAM COMPUTES REQUESTED ALPHAS/BETAS AND THEN    LEV04680
C                 USES THE BISEC SUBROUTINE TO CALCULATE EIGENVALUES  LEV04690
C                 OF THE TRIDIAGONAL MATRICES GENERATED FOR THE ORDERS LEV04700
C                 SPECIFIED BY THE USER AND ON THE USER-SPECIFIED     LEV04710
C                 INTERVALS.  PROGRAM THEN USES THE SUBROUTINE INVERR LEV04720
C                 TO COMPUTE ERROR ESTIMATES FOR THE ISOLATED GOOD    LEV04730
C                 T-EIGENVALUES WHICH ARE USED TO CHECK THE           LEV04740
C                 CONVERGENCE OF THESE T-EIGENVALUES.                 LEV04750
C                                                                     LEV04760
C     CONTROL PARAMETERS FOR WRITES                                   LEV04770
C                                                                     LEV04780
C     IHIS  = (0,1)  MEANS                                            LEV04790
C                                                                     LEV04800
C             (0) IF ISTOP .GT. 0 THEN ALPHA/BETAS ARE NOT SAVED ON   LEV04810
C                 FILE 1.                                             LEV04820
C                                                                     LEV04830
C             (1) PROGRAM WRITES ALPHAS/BETAS AND LAST 2 LANCZOS      LEV04840
C                 VECTORS TO FILE 1 SO THAT THE T-MATRIX GENERATION   LEV04850
C                 MAY BE REUSED OR CONTINUED LATER IF NECESSARY.      LEV04860
C                 TYPICALLY ONE WOULD ALWAYS DO THIS ON ANY RUN WHERE LEV04870
C                 A HISTORY FILE IS BEING GENERATED.  HISTORY MUST BE LEV04880
C                 SAVED IN MACHINE FORMAT ((4Z20) FOR IBM 3081) SO    LEV04890
C                 THAT NO ERRORS ARE INTRODUCED BY FORMAT CONVERSIONS.LEV04900
C                                                                     LEV04910
```

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

```
C                    WERE GENERATED.  IN CASE (4) THREE 'LANCZOS' VECTORS      LEV05470
C                    ARE READ IN FROM FILE 2.                                  LEV05480
C                                                                              LEV05490
C       (3)     OUTPUT FILE:                                                   LEV05500
C                    COMPUTED GOOD EIGENVALUES OF THE T-MATRICES USED. ALSO    LEV05510
C                    CONTAINS T-MULTIPLICITIES OF THESE EIGENVALUES AS         LEV05520
C                    EIGENVALUES OF THE T-MATRIX, AND THEIR GAPS AS            LEV05530
C                    EIGENVALUES IN THE A-MATRIX AND IN THE T-MATRIX.          LEV05540
C                    FILE 3 IS ALWAYS WRITTEN.  IN CASE (3) THIS OUTPUT        LEV05550
C                    CONTAINS THE EIGENVALUES OF THE B-INVERSE MATRIX          LEV05560
C                    SINCE IN THIS CASE THE T-MATRICES CORRESPOND TO           LEV05570
C                    THE B-INVERSE MATRIX AND NOT TO THE A-MATRIX.  IN         LEV05580
C                    THIS CASE, 3 SETS OF GAPS ARE GIVEN, THOSE IN             LEV05590
C                    THE T-MATRIX, IN THE B-INVERSE MATRIX AND THOSE           LEV05600
C                    FOR THE CORRESPONDING EIGENVALUES IN THE A-MATRIX.        LEV05610
C                                                                              LEV05620
C       (4)     OUTPUT FILE:                                                   LEV05630
C                    ERROR ESTIMATES FOR THE ISOLATED GOOD T-EIGENVALUES       LEV05640
C                    WHICH ARE OBTAINED USING THE SUBROUTINE INVERR. THESE     LEV05650
C                    ESITMATES USE THE LAST COMPONENTS OF THE ASSOCIATED       LEV05660
C                    T-EIGENVECTORS WHICH ARE COMPUTED USING INVERSE           LEV05670
C                    ITERATION.  FILE 4 IS ALWAYS WRITTEN.                     LEV05680
C                                                                              LEV05690
C                                                                              LEV05700
C       (7)     INPUT FILE:                                                    LEV05710
C                    USED ONLY IN CASES (3) AND (4), FACTORED INVERSES         LEV05720
C                    OF REAL SYMMETRIC MATRICES AND GENERALIZED EIGENVALUE     LEV05730
C                    PROBLEM.  CONTAINS THE REQUIRED FACTORIZATION OF THE      LEV05740
C                    B-MATRIX.                                                 LEV05750
C                                                                              LEV05760
C       (8)     INPUT FILE:                                                    LEV05770
C                    SAMPLE USPEC SUBROUTINE ASSUMES THAT THE ARRAYS           LEV05780
C                    REQUIRED TO SPECIFY THE USER'S-MATRIX ARE STORED ON       LEV05790
C                    FILE 8.  USERS MUST MAKE WHATEVER DEFINITIONS ARE         LEV05800
C                    APPROPRIATE FOR THEIR MATRICES.  NOTE THAT IN CASE        LEV05810
C                    (3) THE LANCZS SUBROUTINE DOES NOT USE THE MATRIX         LEV05820
C                    ON FILE 8 IN THE T-MATRIX GENERATION, RATHER IT           LEV05830
C                    USES THE FACTORIZATION OF AN ASSOCIATED                   LEV05840
C                    B-MATRIX WHICH IS STORED ON FILE 7.  IN CASE (4),         LEV05850
C                    THE INFORMATION STORED ON BOTH FILES 7 AND 8 IS USED.     LEV05860
C                                                                              LEV05870
C       (9)     INPUT AND OUTPUT FILE:                                         LEV05880
C                    CAN BE USED TO STORE THE TRUE EIGENVALUES OF THE          LEV05890
C                    GIVEN PROBLEM,  WHEN THE PROCEDURE                        LEV05900
C                    IS BEING EXERCISED ON A TEST MATRIX.                      LEV05910
C                                                                              LEV05920
C       (11)    OUTPUT FILE:                                                   LEV05930
C                    COMPUTED DISTINCT EIGENVALUES OF T-MATRICES USED.         LEV05940
C                    ALSO CONTAINS THEIR T-MULTIPLICITIES AND T-GAPS TO        LEV05950
C                    NEAREST DISTINCT EIGENVALUES, AND THE T-MULTIPLICITY      LEV05960
C                    PATTERN OF THE GOOD AND THE SPURIOUS T-EIGENVALUES.       LEV05970
C                    FILE 11 IS WRITTEN ONLY IF IDIST = 1.                     LEV05980
C                                                                              LEV05990
C                                                                              LEV06000
C-----PARAMETERS SET BY THE EIGENVALUE PROGRAMS----------------------- LEV06010
```

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

```
C            PROCEDURE WILL TERMINATE FOR THE USER TO DECIDE WHAT TO    LEV06570
C            DO.  THE USER CAN OVER-RIDE THIS TEST BY SIMPLY DECREASING LEV06580
C            THE SIZE OF BTOL, BUT THEN CONVERGENCE IS NOT AS CERTAIN.  LEV06590
C            THE PROGRAM SETS BTOL = 1.D-8 WHICH IS A VERY CONSERVATIVE LEV06600
C            CHOICE. THE || A || IS ESTIMATED BY USING AN ESTIMATE      LEV06610
C            OF THE NORM OF THE T-MATRIX, T(1,KMAX).                    LEV06620
C                                                                       LEV06630
C     GAPTOL = RELATIVE TOLERANCE USED IN THE SUBROUTINE ISOEV          LEV06640
C              TO DETERMINE WHICH OF THE GOOD T-EIGENVALUES NEED         LEV06650
C              ERROR ESTIMATES.  THE PROGRAM SETS GAPTOL = 1.D-8.        LEV06660
C              IF FOR A GIVEN 'GOOD' T-EIGENVALUE THE COMPUTED GAP       LEV06670
C              IS TOO SMALL AND IS DUE TO A 'SPURIOUS' T-EIGENVALUE      LEV06680
C              THEN THE 'GOOD' T-EIGENVALUE IS ASSUMED TO HAVE CONVERGEDLEV06690
C              AND NO ERROR ESTIMATES ARE COMPUTED.                     LEV06700
C                                                                       LEV06710
C                                                                       LEV06720
C-----USER-SPECIFIED PARAMETERS FOR EIGENVALUE PROGRAMS-----------------LEV06730
C                                                                       LEV06740
C                                                                       LEV06750
C     RELTOL = RELATIVE TOLERANCE USED IN 'COMBINING' COMPUTED          LEV06760
C              EIGENVALUES OF T(1,MEV) PRIOR TO COMPUTING ERROR         LEV06770
C              ESTIMATES.                                               LEV06780
C                                                                       LEV06790
C     THE LUMPING OF T-EIGENVALUES OCCURS IN SUBROUTINE LUMP.           LEV06800
C     LUMPING IS NECESSARY BECAUSE IT IS IMPOSSIBLE TO ACCURATELY       LEV06810
C     PREDICT THE ACCURACY OF THE BISEC SUBROUTINE.  LUMP 'COMBINES'    LEV06820
C     T-EIGENVALUES THAT HAVE SLIPPED BY THE TOLERANCE THAT WAS USED    LEV06830
C     IN THE T-MULTIPLICITY TESTS.  IN PARTICULAR IF FOR SOME J,        LEV06840
C                                                                       LEV06850
C     |EVALUE(J)-EVALUE(J-1)| < DMAX1(RELTOL*|EVALUE(J)|,SCALE2*MULTOL) LEV06860
C                                                                       LEV06870
C     THEN THESE T-EIGENVALUES ARE 'COMBINED'.  MULTOL IS THE TOLERANCE LEV06880
C     THAT WAS USED IN THE T-MULTIPLICITY TEST IN BISEC.  SEE THE HEADERLEV06890
C     ON THE LUMP SUBROUTINE FOR MORE DETAILS.                         LEV06900
C                                                                       LEV06910
C     RELTOL IS SET TO 1.D-10.                                         LEV06920
C                                                                       LEV06930
C     MXINIT = MAXIMUM NUMBER OF INVERSE ITERATIONS ALLOWED IN          LEV06940
C              SUBROUTINE INVERR FOR EACH ISOLATED GOOD T-EIGENVALUE.   LEV06950
C              TYPICALLY ONLY ONE ITERATION IS REQUIRED.               LEV06960
C                                                                       LEV06970
C     SEEDS FOR RANDOM NUMBER GENERATORS = INTEGER*4 SCALARS.           LEV06980
C                                                                       LEV06990
C            (1) SVSEED = SEED FOR STARTING VECTOR USED IN              LEV07000
C                 T-MATRIX GENERATION IN LANCZS SUBROUTINE              LEV07010
C                                                                       LEV07020
C            (2) RHSEED = SEED FOR RIGHT-HAND SIDE USED IN              LEV07030
C                 INVERSE ITERATION COMPUTATIONS IN INVERR.             LEV07040
C                                                                       LEV07050
C     BISEC DATA                                                        LEV07060
C                                                                       LEV07070
C     (1) NINT  =  NUMBER OF SUBINTERVALS ON WHICH EIGENVALUES ARE      LEV07080
C                  TO BE COMPUTED.                                      LEV07090
C                                                                       LEV07100
C     (2) LB(J) = (J = 1,NINT)  =  LEFT END POINTS OF THESE INTERVALS.  LEV07110
```

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

```
C      BE CHECKED ONLY ON THOSE EIGENVALUES OF INTEREST AND         LEVO7670
C      ONCE THE EIGENVALUES ON SUB-INTERVALS OF THESE INTERVALS HAVE LEVO7680
C      CONVERGED, ANY SUBSEQUENT EIGENVALUE COMPUTATIONS SHOULD BE    LEVO7690
C      MADE ONLY ON THE UNCONVERGED PORTIONS.  OBVIOUSLY, IT WOULD BE LEVO7700
C      DIFFICULT TO INCORPORATE CODE TO DO THE ABOVE WITHOUT KNOWING  LEVO7710
C      A PRIORI PRECISELY WHAT THE USER IS TRYING TO COMPUTE.         LEVO7720
C      THEREFORE, WE DID NOT ATTEMPT TO DO THIS.  IF ONE WISHES TO    LEVO7730
C      MAKE SUCH A MODIFICATION THEN ONE MUST ALSO MODIFY THE PROGRAM LEVO7740
C      SO THAT IT CREATES AN OVERALL LIST OF THE CONVERGED 'GOOD'     LEVO7750
C      T-EIGENVALUES AS THEY ARE COMPUTED, SINCE CONVERGED 'GOOD'     LEVO7760
C      T-EIGENVALUES WHICH WERE COMPUTED AT A PARTICULAR VALUE OF MEV LEVO7770
C      WOULD NO LONGER BE RECOMPUTED AT LARGER VALUES OF MEV.         LEVO7780
C                                                                     LEVO7790
C      IF ONLY A FEW EIGENVALUES ARE TO BE COMPUTED THEN SUCH CHANGES LEVO7800
C      WOULD NOT MAKE MUCH DIFFERENCE IN THE RUNNING TIME.            LEVO7810
C                                                                     LEVO7820
C                                                                     LEVO7830
C-----ARRAYS REQUIRED BY THE EIGENVALUE PROGRAMS---------------------LEVO7840
C                                                                     LEVO7850
C                                                                     LEVO7860
C      ALL 4 CASES                                                    LEVO7870
C                                                                     LEVO7880
C      ALPHA(J) = REAL*8 ARRAY. ITS DIMENSION MUST BE AT LEAST KMAX,  LEVO7890
C                 THE LENGTH OF THE LARGEST T-MATRIX ALLOWED.  THIS   LEVO7900
C                 ARRAY CONTAINS THE DIAGONAL ENTRIES OF THE T-MATRICES. LEVO7910
C                                                                     LEVO7920
C      BETA(J) = REAL*8 ARRAY.  ITS DIMENSION MUST BE AT LEAST KMAX+1. LEVO7930
C                THIS ARRAY CONTAINS THE SUBDIAGONAL ENTRIES OF THE   LEVO7940
C                T-MATRICES.                                          LEVO7950
C                                                                     LEVO7960
C                THE ALPHA AND BETA VECTORS ARE NOT ALTERED           LEVO7970
C                DURING THE CALCULATIONS.                             LEVO7980
C                                                                     LEVO7990
C      V1(J),V2(J),VS(J) = REAL*8 ARRAYS IN REAL SYMMETRIC CASES.     LEVO8000
C                          V1 AND V2 ARE COMPLEX*16 IN HERMITIAN CASE. LEVO8010
C                          IN CASES (1) AND (2) VS MUST BE OF         LEVO8020
C                          DIMENSION AT LEAST KMAX.  IN CASES (3) AND LEVO8030
C                          (4) VS MUST BE OF DIMENSION AT LEAST       LEVO8040
C                          MAX(N,KMAX).  IN REAL SYMMETRIC CASES      LEVO8050
C                          V1 MUST BE OF DIMENSION AT LEAST           LEVO8060
C                          MAX(KMAX+1,N) AND V2 MUST BE OF DIMENSION  LEVO8070
C                          MAX(KMAX,N).  IN HERMITIAN CASES V1        LEVO8080
C                          MUST BE OF DIMENSION MAX(N,(KMAX+1)/2)     LEVO8090
C                          AND V2 OF DIMENSION AT LEAST MAX(N,KMAX/2). LEVO8100
C                          IN ALL CASES HOWEVER, THE ABOVE DIMENSIONS LEVO8110
C                          FOR V2 ARE VALID ONLY IF NO MORE           LEVO8120
C                          THAN KMAX/2 EIGENVALUES OF THE GIVEN       LEVO8130
C                          T-MATRICES ARE TO BE COMPUTED IN ANY GIVEN LEVO8140
C                          SUBINTERVAL. V2 IS USED IN THE SUBROUTINE  LEVO8150
C                          BISEC TO HOLD THE UPPER AND LOWER          LEVO8160
C                          ENDPOINTS OF THE SUBINTERVALS GENERATED    LEVO8170
C                          DURING THE BISECTIONS.  THEREFORE, ITS     LEVO8180
C                          REAL*8 DIMENSION MUST ALWAYS BE AT LEAST   LEVO8190
C                          2*Q WHERE Q IS THE MAXIMUM NUMBER OF       LEVO8200
C                          EIGENVALUES OF THE SPECIFIED T-MATRIX IN ANY LEVO8210
```

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

```
C     IS REFERRED TO THOSE PROGRAMS FOR DETAILS.            LEV08770
C                                                           LEV08780
C                                                           LEV08790
C-----SUBROUTINES INCLUDED-----------------------------------LEV08800
C                                                           LEV08810
C                                                           LEV08820
C     LANCZS = COMPUTES THE ALPHA/BETA HISTORY. IN CASES (1) AND (2)  LEV08830
C              REAL SYMMETRIC AND HERMITIAN MATRICES, USES SUBROUTINES LEV08840
C              FINPRO, GENRAN AND CMATV.  IN CASE (3), INVERSES OF  LEV08850
C              REAL SYMMETRIC MATRICES, USES SUBROUTINES FINPRO,  LEV08860
C              GENRAN AND BSOLV.  IN CASE (4), GENERALIZED EIGENVALUE  LEV08870
C              PROBLEM, USES SUBROUTINES FINPRO, GENRAN, AMATV AND  LEV08880
C              LSOLV.                                       LEV08890
C                                                           LEV08900
C     BISEC = COMPUTES EIGENVALUES OF THE SPECIFIED T-MATRIX  LEV08910
C             USING STURM SEQUENCING, ON SEQUENCE OF INTERVALS  LEV08920
C             SPECIFIED BY THE USER.  EACH SUBINTERVAL IS TREATED  LEV08930
C             AS OPEN ON THE LEFT AND CLOSED ON THE RIGHT.  LEV08940
C             EIGENVALUES ARE COMPUTED WITH SIMULTANEOUS DETERMINATION LEV08950
C             OF THE T-MULTIPLICITIES AND OF SPURIOUS T-EIGENVALUES. LEV08960
C                                                           LEV08970
C     INVERR = USES INVERSE ITERATION ON T-MATRICES TO COMPUTE ERROR  LEV08980
C              ESTIMATES ON COMPUTED GOOD T-EIGENVALUES. (USES GENRAN) LEV08990
C                                                           LEV09000
C     LUMP = 'COMBINES' EIGENVALUES OF T-MATRIX USING THE RELATIVE  LEV09010
C            TOLERANCE RELTOL.                              LEV09020
C                                                           LEV09030
C     ISOEV = CALCULATES GAPS BETWEEN DISTINCT EIGENVALUES OF T-MATRIX LEV09040
C             AND THEN USES THESE GAPS TO LABEL THOSE 'GOOD'  LEV09050
C             T-EIGENVALUES FOR WHICH ERROR ESTIMATES ARE NOT COMPUTED. LEV09060
C                                                           LEV09070
C     TNORM = COMPUTES THE SCALE TKMAX USED IN DETERMINING THE  LEV09080
C             TOLERANCES FOR THE SPURIOUS, T-MULTIPLICITY AND PRTESTS. LEV09090
C             IT ALSO CHECKS FOR LOCAL ORTHOGONALITY OF THE LANCZOS  LEV09100
C             VECTORS BY TESTING THE RELATIVE SIZE OF THE BETAS USING  LEV09110
C             THE RELATIVE TOLERANCE BTOL.                  LEV09120
C                                                           LEV09130
C     PRTEST = LOOKS FOR GOOD T-EIGENVALUES THAT HAVE BEEN MISLABELLED LEV09140
C              BY THE SPURIOUS TEST BECAUSE THEY HAD 'TOO SMALL' A  LEV09150
C              PROJECTION ON THE STARTING LANCZOS VECTOR.   LEV09160
C              (LESS THAN SINGLE PRECISION)                 LEV09170
C              TESTS INDICATE THAT SUCH EIGENVALUES ARE RARE.  LEV09180
C              PRTEST SHOULD BE CALLED ONLY AFTER CONVERGENCE  LEV09190
C              HAS BEEN ESTABLISHED.                        LEV09200
C                                                           LEV09210
C     INVERM = USED TO COMPUTE ERROR ESTIMATES FOR ANY T-EIGENVALUES  LEV09220
C              WHICH PRTEST INDICATES MAY HAVE BEEN MISLABELLED.  LEV09230
C              SUCH T-EIGENVALUES ARE RELABELLED ONLY IF THEIR ERROR  LEV09240
C              ESTIMATES ARE SUFFICIENTLY SMALL.  PRIMARY USE OF  LEV09250
C              INVERM IS IN THE CORRESPONDING EIGENVECTOR COMPUTATIONS. LEV09260
C                                                           LEV09270
C     CASES (3) AND (4) ONLY, FACTORED INVERSES:            LEV09280
C                                                           LEV09290
C     FOR OPTIONAL, PRELIMINARY PROCESSING:                 LEV09300
C     PERMUT  (PROGRAM CALLS SPARSPAK PACKAGE) :            LEV09310
```

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

```
C     THE ARRAYS V1 AND V2 CONSUME 2*MAXIMUM(KMAX,N)*8 BYTES, WITH THE  LEV09870
C     QUALIFICATION STATED ABOVE WHERE V2 IS DEFINED.  THE G-ARRAY       LEV09880
C     CONSUMES .5*MAX(2*KMAX,N)*8 BYTES.                                 LEV09890
C                                                                        LEV09900
C     CASE (2), HERMITIAN MATRICES:                                      LEV09910
C                                                                        LEV09920
C     THE ARRAYS IN THE HERMITIAN EIGENVALUE PROGRAMS REQUIRE            LEV09930
C     THE EQUIVALENT OF ONE REAL*8 ARRAY OF DIMENSION                    LEV09940
C                                                                        LEV09950
C         3.5*KMAX + 4*MAX(KMAX/2,N) + .5*MAX(2*KMAX,N) + 2*N            LEV09960
C                                                                        LEV09970
C     PLUS WHATEVER IS NEEDED TO GENERATE A*X FOR THE GIVEN MATRIX A.    LEV09980
C     THE ARRAYS ALPHA, BETA, VS, AND MP CONSUME 3.5*KMAX*8 BYTES.       LEV09990
C     THE ARRAYS V1 AND V2 CONSUME 4*MAXIMUM(KMAX/2,N)*8 BYTES, WITH THELEV10000
C     QUALIFICATION STATED ABOVE WHERE V2 IS DEFINED.  THE G-ARRAY       LEV10010
C     CONSUMES .5*MAX(2*KMAX,N)*8 BYTES.  GR REQUIRES                    LEV10020
C     AND GC REQUIRE 2*N*8BYTES.                                         LEV10030
C                                                                        LEV10040
C                                                                        LEV10050
C     CASE (3), INVERSES OF REAL SYMMETRIC MATRICES:                     LEV10060
C                                                                        LEV10070
C     THE ARRAYS IN THE EIGENVALUE PROGRAMS DESIGNED FOR                 LEV10080
C     CASE (3), INVERSES OF REAL SYMMETRIC MATRICES USING               LEV10090
C     REORDERING AND FACTORIZATION, REQUIRE                             LEV10100
C     THE EQUIVALENT OF ONE REAL*8 ARRAY OF DIMENSION                    LEV10110
C                                                                        LEV10120
C         3*KMAX + 3*MAX(KMAX,N) + .5*MAX(2*KMAX,N)                     LEV10130
C                                                                        LEV10140
C     PLUS WHATEVER IS NEEDED TO GENERATE B(INVERSE)*X  FOR THE          LEV10150
C     SCALED, SHIFTED AND PERMUTED VERSION OF A WHICH WE DENOTE          LEV10160
C     BY B.  THE ARRAYS ALPHA, BETA, MP, AND MP2 CONSUME 3*KMAX*8       LEV10170
C     BYTES.  THE ARRAYS V1, V2, AND VS CONSUME 3*MAX(KMAX,N)*8 BYTES,   LEV10180
C     WITH THE QUALIFICATION STATED ABOVE WHERE V2 IS DEFINED.          LEV10190
C     THE G ARRAY CONSUMES .5*MAX(2*KMAX,N)*8 BYTES. THESE NUMBERS       LEV10200
C     DO NOT INCLUDE THE STORAGE REQUIRED BY THE PREPROCESSING PROGRAMS LEV10210
C     PERMUT, LORDER, LFACT, AND LTEST.                                  LEV10220
C                                                                        LEV10230
C                                                                        LEV10240
C     A SYMMETRIC, SPARSE MATRIX OF ORDER N WITH NZS NONZERO ELEMENTS    LEV10250
C     BELOW THE MAIN DIAGONAL WOULD REQUIRE THE EQUIVALENT OF ONE        LEV10260
C     REAL*8 ARRAY OF DIMENSION 1.5*(NZS + N) IF THE POINTERS USED       LEV10270
C     ARE INTEGER*4.                                                     LEV10280
C                                                                        LEV10290
C     SOME OF THE ARRAY STORAGE IS NOT ESSENTIAL AND COULD BE            LEV10300
C     ELIMINATED IF STORAGE IS A PROBLEM.                                LEV10310
C     THE FOLLOWING COMMENTS APPLY DIRECTLY ONLY TO CASE (1),            LEV10320
C     THE PROGRAMS FOR REAL SYMMETRIC MATRICES, HOWEVER, SIMILAR         LEV10330
C     STATEMENTS COULD BE MADE ABOUT THE OTHER CASES.                    LEV10340
C                                                                        LEV10350
C     CASE (1), REAL SYMMETRIC PROGRAMS:                                 LEV10360
C     THE G ARRAY COULD BE REMOVED IF THE USER IS WILLING TO             LEV10370
C                                                                        LEV10380
C             (1) REGENERATE THE RANDOM STARTING VECTOR IN INVERR        LEV10390
C                 FOR EACH ERROR ESTIMATE                                LEV10400
C             (2) WRITE OUT THE ERROR ESTIMATES AND VARIOUS GAPS AS      LEV10410
```

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

```
C     EIGENVALUES. AS PROGRAMMED, ALL THE COMPUTED DISTINCT EIGENVALUES LEV10970
C     OF T(1,MEV) ARE STORED IN VS.  THEREFORE TO TAKE ADVANTAGE OF      LEV10980
C     SUCH A REDUCTION IN STORAGE THE USER WOULD HAVE TO MODIFY THAT     LEV10990
C     PART OF THE PROGRAM AND ALSO COMMENT OUT THE CALL TO THE           LEV11000
C     SUBROUTINE PRTEST.                                                 LEV11010
C                                                                        LEV11020
C                                                                        LEV11030
C------------------------------------------------------------------------LEV11040
C                                                                        LEV11050
C     COMMENTS FOR EIGENVECTOR COMPUTATIONS                              LEV11060
C                                                                        LEV11070
C------------------------------------------------------------------------LEV11080
C                                                                        LEV11090
C                                                                        LEV11100
C     THE EIGENVALUES WHOSE EIGENVECTORS ARE TO BE COMPUTED MUST         LEV11110
C     HAVE BEEN COMPUTED USING THE CORRESPONDING LANCZOS EIGENVALUE      LEV11120
C     PROGRAMS BECAUSE THE EIGENVECTOR PROGRAMS WILL USE THE SAME        LEV11130
C     FAMILY OF LANCZOS TRIDIAGONAL MATRICES THAT WAS USED IN THE        LEV11140
C     CORRESPONDING EIGENVALUE COMPUTATIONS.                             LEV11150
C                                                                        LEV11160
C     THESE PROGRAMS ASSUME THAT THE EIGENVALUES SUPPLIED TO IT          LEV11170
C     HAVE BEEN COMPUTED ACCURATELY, AS MEASURED BY THE                  LEV11180
C     ERROR ESTIMATES COMPUTED IN THE CORRESPONDING LANCZOS             LEV11190
C     EIGENVALUE COMPUTATIONS, ALTHOUGH THESE ESTIMATES ARE             LEV11200
C     TYPICALLY CONSERVATIVE.  IN CASES (1), (2) AND (4), THE           LEV11210
C     EIGENVALUES OF INTEREST ARE STORED IN THE ARRAY GOODEV(J),        LEV11220
C     J=1,NGOOD.  IN CASE (3) THE PROGRAM WORKS WITH THE                LEV11230
C     EIGENVALUES OF B(INVERSE) WHICH ARE STORED IN THE ARRAY           LEV11240
C     GOODBI(J), J=1,NGOOD.  THE CORRESPONDING EIGENVALUES             LEV11250
C     OF A ARE STORED IN GOODA(J), J=1,NGOOD.                          LEV11260
C                                                                        LEV11270
C     FOR EACH GOODEV(J), THE SUBROUTINE STURMI COMPUTES THE            LEV11280
C     SMALLEST SIZE LANCZOS TRIDIAGONAL MATRIX, T(1,M1(J)), FOR         LEV11290
C     WHICH  GOODEV(J) IS AN EIGENVALUE TO WITHIN A SPECIFIED           LEV11300
C     TOLERANCE.  IT ALSO ATTEMPTS TO COMPUTE THE SIZE, M2(J),          LEV11310
C     BY WHICH THE GIVEN EIGENVALUE BECOMES A DOUBLE EIGENVALUE         LEV11320
C     TO WITHIN THE GIVEN TOLERANCE.  THESE VALUES ARE USED             LEV11330
C     TO DETERMINE 1ST GUESSES AT SIZES FOR THE T-EIGENVECTORS          LEV11340
C     THAT WILL BE USED IN THE RITZ VECTOR COMPUTATIONS.               LEV11350
C     SUBROUTINE INVERM SUCCESSIVELY COMPUTES CORRESPONDING            LEV11360
C     EIGENVECTORS OF ENLARGED T-MATRICES UNTIL A SUITABLE             LEV11370
C     SIZE T-MATRIX IS DETERMINED FOR EACH J.  UP TO 10 SUCH           LEV11380
C     EIGENVECTOR COMPUTATIONS ARE ALLOWED FOR EACH EIGENVALUE.        LEV11390
C                                                                        LEV11400
C     AFTER APPROPRIATE T-EIGENVECTORS HAVE BEEN COMPUTED,             LEV11410
C     RITZ VECTOR CORRESPONDING TO THESE T-EIGENVECTORS ARE THEN       LEV11420
C     COMPUTED AND TAKEN AS APPROXIMATE EIGENVECTORS OF A FOR THE      LEV11430
C     GIVEN EIGENVALUES, GOODEV(J), J = 1, ..., NGOOD.                 LEV11440
C                                                                        LEV11450
C     THIS IMPLEMENTATION FIRST COMPUTES ALL OF THE RELEVANT           LEV11460
C     EIGENVECTORS OF THE SYMMETRIC TRIDIAGONAL MATRICES               LEV11470
C     IN THE VECTOR, TVEC.                                              LEV11480
C                                                                        LEV11490
C     THEN, AS EACH OF THE LANCZOS VECTORS IS REGENERATED, ALL         LEV11500
C     OF THE RITZ VECTORS CORRESPONDING TO THESE                       LEV11510
```

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

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

```
C                    COMPUTED REGARDLESS OF WHETHER OR NOT THAT        LEV12620
C                    T-EIGENVECTOR SATISFIED THE ERROR ESTIMATE TEST.  LEV12630
C                                                                      LEV12640
C                                                                      LEV12650
C-----INPUT/OUTPUT FILES FOR THE EIGENVECTOR COMPUTATIONS-------------LEV12660
C                                                                      LEV12670
C                                                                      LEV12680
C     ANY INPUT DATA OTHER THAN THE T-MATRIX HISTORY FILE AND THE      LEV12690
C     PREVIOUSLY COMPUTED EIGENVALUES AND CORRESPONDING ERROR          LEV12700
C     ESTIMATES SHOULD BE STORED ON FILE 5 IN FREE FORMAT.             LEV12710
C     SEE SAMPLE INPUT/OUTPUT FOR TYPICAL INPUT FILE.                  LEV12720
C                                                                      LEV12730
C     FILE 6 WAS USED AS THE INTERACTIVE TERMINAL OUTPUT FILE.         LEV12740
C     THIS FILE PROVIDES A RUNNING ACCOUNT OF THE PROGRESS OF THE      LEV12750
C     COMPUTATIONS.  ADDITIONAL PRINTOUT IS GENERATED WHEN             LEV12760
C     THE FLAG IWRITE = 1.                                             LEV12770
C                                                                      LEV12780
C                                                                      LEV12790
C DESCRIPTION OF OTHER I/O FILES                                       LEV12800
C                                                                      LEV12810
C FILE (K)     CONTAINS:                                               LEV12820
C                                                                      LEV12830
C    (2)     INPUT FILE:                                               LEV12840
C            PREVIOUSLY-GENERATED T-MATRICES (ALPHA/BETA ARRAYS)       LEV12850
C            AND THE FINAL TWO LANCZOS VECTORS USED ON THAT            LEV12860
C            COMPUTATION.  THIS PROGRAM ALLOWS ENLARGEMENT             LEV12870
C            OF ANY T-MATRICES PROVIDED ON FILE 2.  NOTE THAT          LEV12880
C            IN CASE (4), THREE 'LANCZOS' VECTORS ARE ON FILE 2.       LEV12890
C                                                                      LEV12900
C    (3)     INPUT FILE:                                               LEV12910
C            THE GOOD EIGENVALUES OF THE T-MATRIX  T(1,MEV)            LEV12920
C            FOR WHICH RITZ VECTORS ARE REQUESTED.                     LEV12930
C            FILE 3 ALSO CONTAINS THE T-MULTIPLICITIES OF THESE        LEV12940
C            EIGENVALUES AND THEIR COMPUTED GAPS IN THE                LEV12950
C            T-MATRICES AND IN THE USER-SUPPLIED MATRIX. IN            LEV12960
C            CASE (3) FILE 3 CONTAINS THE EIGENVALUES OF THE           LEV12970
C            B-INVERSE MATRIX AND 3 SETS OF CORRESPONDING GAPS.        LEV12980
C            THIS FILE IS CREATED IN THE LANCZOS EIGENVALUE            LEV12990
C            COMPUTATIONS.                                             LEV13000
C                                                                      LEV13010
C    (4)     INPUT FILE:                                               LEV13020
C            ERROR ESTIMATES FOR THE ISOLATED GOOD T-EIGENVALUES       LEV13030
C            ON FILE 3.  THIS FILE IS CREATED DURING THE LANCZOS       LEV13040
C            EIGENVALUE COMPUTATIONS.                                  LEV13050
C                                                                      LEV13060
C    (7)     INPUT FILE:                                               LEV13070
C            IN CASE (3) ((4)),                                        LEV13080
C            CONTAINS SPARSE MATRIX REPRESENTATION OF FACTORIZATION    LEV13090
C            OF MATRIX (B-MATRIX) USED BY LANCZS SUBROUTINE.           LEV13100
C                                                                      LEV13110
C    (8)     INPUT FILE:                                               LEV13120
C            IN CASES (1),(2) AND (4), USPEC SUBROUTINE ASSUMES        LEV13130
C            THAT USER-SUPPLIED A-MATRIX IS ON FILE 8.  IN CASE (3)    LEV13140
C            A-MATRIX CAN BE STORED ON FILE 8, BUT IT IS NOT           LEV13150
C            USED BY THE LANCZOS PROGRAMS.                             LEV13160
```

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

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

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

```
C                                                                    LEV14820
C      MP(J) = INTEGER*4 ARRAY WHOSE DIMENSION IS AT LEAST NGOOD.     LEV14830
C              INITIALLY CONTAINS THE MULTIPLICITY OF THE EIGENVALUE  LEV14840
C              GOODEV(J) AS AN EIGENVALUE OF THE T-MATRIX T(1,MEV).   LEV14850
C              USED TO FLAG EIGENVALUES FOR WHICH NO T-EIGENVECTOR    LEV14860
C              OR NO RITZ VECTOR IS TO BE COMPUTED.                   LEV14870
C                                                                    LEV14880
C      MA(J)   = INTEGER*4 ARRAYS EACH OF WHOSE DIMENSIONS            LEV14890
C                IS AT LEAST NGOOD.  USED IN DETERMINING              LEV14900
C                AN APPROPRIATE T-MATRIX FOR EACH EIGENVALUE          LEV14910
C                IN GOODEV ARRAY.                                     LEV14920
C                                                                    LEV14930
C      MINT(J),MFIN(J) = INTEGER*4 ARRAYS WHOSE DIMENSIONS MUST BE AT LEV14940
C                        LEAST NGOOD.  USED TO POINT TO THE BEGINNINGS LEV14950
C                        AND THE ENDS OF THE COMPUTED EIGENVECTOR     LEV14960
C                        OF THE T-MATRIX, T(1,|MA(J)|).              LEV14970
C                                                                    LEV14980
C      IDELTA(J)  = INTEGER*4 ARRAY WHOSE DIMENSION MUST BE AT        LEV14990
C                   LEAST NGOOD.  CONTAINS INCREMENTS USED IN LOOPS   LEV15000
C                   ON APPROPRIATE SIZE T-MATRIX FOR THE T-EIGENVECTOR LEV15010
C                   COMPUTATIONS.                                     LEV15020
C                                                                    LEV15030
C                                                                    LEV15040
C      CASE (2) ONLY, HERMITIAN MATRICES:                            LEV15050
C                                                                    LEV15060
C      GR(J),GC(J)     = REAL*8 ARRAYS USED ONLY IN CASE (2),         LEV15070
C                        HERMITIAN MATRICES.  EACH MUST BE AT         LEV15080
C                        LEAST MAX(N,KMAX).  USED TO HOLD             LEV15090
C                        STARTING VECTORS FOR LANCZS                  LEV15100
C                        COMPUTATIONS AND FOR INVERM SUBROUTINES.     LEV15110
C                                                                    LEV15120
C      CASES (3) AND (4) ONLY, FACTORED INVERSES OF REAL SYMMETRIC    LEV15130
C      MATRICES AND GENERALIZED EIGENVALUE PROBLEMS:                  LEV15140
C                                                                    LEV15150
C      VS(J) =  REAL*8 ARRAY WHOSE DIMENSION MUST BE AT LEAST N.      LEV15160
C               USED IN REGENERATION OF THE LANCZOS VECTORS.         LEV15170
C                                                                    LEV15180
C      IPR(J), IPT(J) = INTEGER*4 ARRAYS.  EACH MUST BE OF DIMENSION  LEV15190
C                       AT LEAST N, THE ORDER OF A. USED TO STORE     LEV15200
C                       THE REORDERING OF THE GIVEN MATRIX.           LEV15210
C                                                                    LEV15220
C      CASE (3) ONLY, INVERSES OF REAL SYMMETRIC MATRICES:           LEV15230
C                                                                    LEV15240
C      GOODA(J), GOODBI(J) = REAL*8 ARRAYS.  EACH MUST BE OF DIMENSION LEV15250
C                            AT LEAST NGOOD, THE NUMBER OF EIGENVALUES LEV15260
C                            BEING CONSIDERED.  GOODA CONTAINS THE    LEV15270
C                            EIGENVALUES OF A AND GOODBI CONTAINS THE LEV15280
C                            EIGENVALUES OF B(INVERSE).  THE PROGRAM  LEV15290
C                            WORKS DIRECTLY WITH THE GOODBI ARRAY.    LEV15300
C                                                                    LEV15310
C                                                                    LEV15320
C-----SUBROUTINES INCLUDED FOR THE EIGENVECTOR COMPUTATIONS------------LEV15330
C                                                                    LEV15340
C                                                                    LEV15350
C      STURMI = FOR EACH GIVEN EIGENVALUE GOODEV(J) DETERMINES        LEV15360
```

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

2.3   LEVAL: Main Program, Eigenvalue Computations

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

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

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

```
C      READ IN THE NUMBER OF SUBINTERVALS TO BE CONSIDERED.          LEV01570
       READ(5,20) EXPLAN                                             LEV01580
       READ(5,*) NINT                                                LEV01590
C                                                                    LEV01600
C      READ IN THE LEFT-END POINTS OF THE SUBINTERVALS TO BE CONSIDERED. LEV01610
C      THESE MUST BE IN ALGEBRAICALLY-INCREASING ORDER               LEV01620
       READ(5,20) EXPLAN                                             LEV01630
       READ(5,*) (LB(J), J=1,NINT)                                   LEV01640
C                                                                    LEV01650
C      READ IN THE RIGHT-END POINTS OF THE SUBINTERVALS TO BE CONSIDERED.LEV01660
C      THESE MUST BE IN ALGEBRAICALLY-INCREASING ORDER               LEV01670
       READ(5,20) EXPLAN                                             LEV01680
       READ(5,*) (UB(J), J=1,NINT)                                   LEV01690
C                                                                    LEV01700
C-------------------------------------------------------------------LEV01710
C      INITIALIZE THE ARRAYS FOR THE USER-SPECIFIED MATRIX           LEV01720
C      AND PASS THE STORAGE LOCATIONS OF THESE ARRAYS TO THE         LEV01730
C      MATRIX-VECTOR MULTIPLY SUBROUTINE CMATV.                      LEV01740
C                                                                    LEV01750
       CALL USPEC(N,MATNO)                                           LEV01760
C                                                                    LEV01770
C-------------------------------------------------------------------LEV01780
C      MASK UNDERFLOW AND OVERFLOW                                   LEV01790
C                                                                    LEV01800
       CALL MASK                                                     LEV01810
C                                                                    LEV01820
C-------------------------------------------------------------------LEV01830
C                                                                    LEV01840
C      WRITE TO FILE 6, A SUMMARY OF THE PARAMETERS FOR THIS RUN     LEV01850
C                                                                    LEV01860
       WRITE(6,30) MATNO,N,KMAX                                      LEV01870
   30 FORMAT(/3X,'MATRIX ID',4X,'ORDER OF A',4X,'MAX ORDER OF T'/    LEV01880
     1 I12,I14,I18/)                                                 LEV01890
C                                                                    LEV01900
       WRITE(6,40) ISTART,ISTOP                                      LEV01910
   40 FORMAT(/2X,'ISTART',3X,'ISTOP'/2I8/)                           LEV01920
C                                                                    LEV01930
       WRITE(6,50) IHIS,IDIST,IWRITE                                 LEV01940
   50 FORMAT(/4X,'IHIS',3X,'IDIST',2X,'IWRITE'/3I8/)                 LEV01950
C                                                                    LEV01960
       WRITE(6,60) SVSEED,RHSEED                                     LEV01970
   60 FORMAT(/' SEEDS FOR RANDOM NUMBER GENERATOR'//                 LEV01980
     1 4X,'LANCZS SEED',4X,'INVERR SEED'/2I15/)                      LEV01990
C                                                                    LEV02000
       WRITE(6,70) (NMEV(J), J=1,NMEVS)                              LEV02010
   70 FORMAT(/' SIZES OF T-MATRICES TO BE CONSIDERED'/(6I12))        LEV02020
C                                                                    LEV02030
       WRITE(6,80) RELTOL,GAPTOL,BTOL                                LEV02040
   80 FORMAT(/' RELATIVE TOLERANCE USED TO COMBINE COMPUTED T-EIGENVALUELEV02050
     1S'/E15.3/' RELATIVE GAP TOLERANCES USED IN INVERSE ITERATION'/ LEV02060
     1E15.3/' RELATIVE TOLERANCE FOR CHECK ON SIZE OF BETAS'/E15.3/) LEV02070
C                                                                    LEV02080
       WRITE(6,90) (J,LB(J),UB(J), J=1,NINT)                         LEV02090
   90 FORMAT(/' BISEC WILL BE USED ON THE FOLLOWING INTERVALS'/      LEV02100
     1 (I6,2E20.6))                                                  LEV02110
```

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

```
      WRITE(1,130)(V2(I), I=1,N)                                 LEV02670
C                                                                LEV02680
      IF (ISTOP.EQ.0) GO TO 540                                  LEV02690
C                                                                LEV02700
  160 CONTINUE                                                   LEV02710
      BKMIN = BTOL                                               LEV02720
C                                                                LEV02730
      WRITE(6,170)                                               LEV02740
  170 FORMAT(/' T-MATRICES (ALPHA AND BETA) ARE NOW AVAILABLE'/) LEV02750
C                                                                LEV02760
C----------------------------------------------------------------LEV02770
C     SUBROUTINE TNORM CHECKS MIN(BETA)/(ESTIMATED NORM(A)) > BTOL . LEV02780
C     IF THIS IS VIOLATED IB IS SET EQUAL TO THE NEGATIVE OF THE INDEX LEV02790
C     OF THE MINIMAL BETA.  IF(IB < 0) THEN SUBROUTINE TNORM IS  LEV02800
C     CALLED FOR EACH VALUE OF MEV TO DETERMINE WHETHER OR NOT THERE LEV02810
C     IS A BETA IN THE T-MATRIX SPECIFIED THAT VIOLATES THIS TEST. LEV02820
C     IF THERE IS SUCH A BETA THE PROGRAM TERMINATES FOR THE USER LEV02830
C     TO DECIDE WHAT TO DO.  THIS TEST CAN BE OVER-RIDDEN BY     LEV02840
C     SIMPLY MAKING BTOL SMALLER, BUT THEN THERE IS THE POSSIBILITY LEV02850
C     THAT LOSSES IN THE LOCAL ORTHOGONALITY MAY HURT THE COMPUTATIONS. LEV02860
C     BTOL = 1.D-8 IS HOWEVER A CONSERVATIVE CHOICE FOR BTOL.    LEV02870
C                                                                LEV02880
C     TNORM ALSO COMPUTES TKMAX = MAX(|ALPHA(K)|,BETA(K), K=1,KMAX). LEV02890
C     TKMAX IS USED TO SCALE THE TOLERANCES USED IN THE          LEV02900
C     T-MULTIPLICITY AND SPURIOUS TESTS IN BISEC. TKMAX IS ALSO USED IN LEV02910
C     THE PROJECTION TEST FOR HIDDEN EIGENVALUES THAT HAD 'TOO SMALL' LEV02920
C     A PROJECTION ON THE STARTING VECTOR.                       LEV02930
C                                                                LEV02940
      CALL TNORM(ALPHA,BETA,BKMIN,TKMAX,KMAX,IB)                 LEV02950
C                                                                LEV02960
C----------------------------------------------------------------LEV02970
C                                                                LEV02980
      TTOL = EPSM*TKMAX                                          LEV02990
C                                                                LEV03000
C     LOOP ON THE SIZE OF THE T-MATRIX                           LEV03010
C                                                                LEV03020
  180 CONTINUE                                                   LEV03030
      MMB = MMB + 1                                              LEV03040
      MEV = NMEV(MMB)                                            LEV03050
C     IS MEV TOO LARGE ?                                         LEV03060
      IF(MEV.LE.KMAX) GO TO 200                                  LEV03070
      WRITE(6,190) MMB, MEV, KMAX                                LEV03080
  190 FORMAT(/' TERMINATE PRIOR TO CONSIDERING THE',I6,'TH T-MATRIX'/ LEV03090
     1' BECAUSE THE SIZE REQUESTED',I6,' IS GREATER THAN THE MAXIMUM SIZLEV03100
     1E ALLOWED',I6/)                                            LEV03110
      GO TO 540                                                  LEV03120
C                                                                LEV03130
  200 MP1 = MEV + 1                                              LEV03140
      BETAM = BETA(MP1)                                          LEV03150
C                                                                LEV03160
      IF (IB.GE.0) GO TO 210                                     LEV03170
C                                                                LEV03180
      T0 = BTOL                                                  LEV03190
C                                                                LEV03200
C----------------------------------------------------------------LEV03210
```

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

```
      LOOP = NDIS                                           LEV03770
      CALL LUMP(VS,RELTOL,MULTOL,SCALE2,MP,LOOP)            LEV03780
C                                                           LEV03790
C-----------------------------------------------------------LEV03800
C                                                           LEV03810
      IF(NDIS.EQ.LOOP) GO TO 230                            LEV03820
C                                                           LEV03830
      WRITE(6,220) NDIS, MEV, LOOP                          LEV03840
  220 FORMAT(/I6,' DISTINCT T-EIGENVALUES WERE COMPUTED IN BISEC AT MEV LEV03850
     1',I6/ 2X,' LUMP SUBROUTINE REDUCES NUMBER OF DISTINCT EIGENVALUES LEV03860
     10',I6)                                                LEV03870
C                                                           LEV03880
  230 CONTINUE                                              LEV03890
      NDIS = LOOP                                           LEV03900
      BETA(MP1) = BETAM                                     LEV03910
C                                                           LEV03920
C-----------------------------------------------------------LEV03930
C     THE SUBROUTINE ISOEV LABELS THOSE SIMPLE EIGENVALUES OF T(1,MEV)  LEV03940
C     WITH VERY SMALL GAPS BETWEEN NEIGHBORING EIGENVALUES OF T(1,MEV)  LEV03950
C     TO AVOID COMPUTING ERROR ESTIMATES FOR ANY SIMPLE GOOD           LEV03960
C     T-EIGENVALUE THAT IS TOO CLOSE TO A SPURIOUS EIGENVALUE.         LEV03970
C     ON RETURN FROM ISOEV, G CONTAINS CODED MINIMAL GAPS             LEV03980
C     BETWEEN THE DISTINCT EIGENVALUES OF T(1,MEV). (G IS REAL).      LEV03990
C     G(I) < 0 MEANS MINGAP IS DUE TO LEFT GAP G(I) > 0 MEANS DUE TO  LEV04000
C     RIGHT GAP. MP(I) = -1 MEANS THAT THE GOOD T-EIGENVALUE IS SIMPLE LEV04010
C     AND HAS A VERY SMALL MINGAP IN T(1,MEV) DUE TO A SPURIOUS       LEV04020
C     EIGENVALUE.  NG = NUMBER OF GOOD T-EIGENVALUES.                 LEV04030
C     NISO = NUMBER OF ISOLATED GOOD T-EIGENVALUES.                   LEV04040
C                                                           LEV04050
      CALL ISOEV(VS,GAPTOL,MULTOL,SCALE1,G,MP,NDIS,NG,NISO) LEV04060
C                                                           LEV04070
C-----------------------------------------------------------LEV04080
C                                                           LEV04090
      WRITE(6,240)NG,NISO,NDIS                              LEV04100
  240 FORMAT(/I6,' GOOD T-EIGENVALUES HAVE BEEN COMPUTED'/  LEV04110
     1 I6,' OF THESE ARE T-ISOLATED'/                       LEV04120
     2 I6,' = NUMBER OF DISTINCT T-EIGENVALUES COMPUTED'/)  LEV04130
C                                                           LEV04140
C     DO WE WRITE DISTINCT EIGENVALUES OF T-MATRIX TO FILE 11?         LEV04150
      IF (IDIST.EQ.0) GO TO 280                             LEV04160
C                                                           LEV04170
      WRITE(11,250) NDIS,NISO,MEV,N,SVSEED,MATNO            LEV04180
  250 FORMAT(/4I6,I12,I8,' = NDIS,NISO,MEV,N,SVSEED,MATNO'/)LEV04190
C                                                           LEV04200
      WRITE(11,260) (MP(I),VS(I),G(I), I=1,NDIS)            LEV04210
  260 FORMAT(2(I3,E25.16,E12.3))                            LEV04220
C                                                           LEV04230
      WRITE(11,270) NDIS, (MP(I), I=1,NDIS)                 LEV04240
  270 FORMAT(/I6,' = NDIS, T-MULTIPLICITIES (0 MEANS  SPURIOUS)'/(20I4))LEV04250
C                                                           LEV04260
  280 CONTINUE                                              LEV04270
C                                                           LEV04280
      IF (NISO.NE.0) GO TO 310                              LEV04290
C                                                           LEV04300
      WRITE(4,290) MEV                                      LEV04310
```

```
  290 FORMAT(/' AT MEV = ',I6,' THERE ARE NO ISOLATED T-EIGENVALUES'/   LEV04320
     1' SO NO ERROR ESTIMATES WERE COMPUTED/')                          LEV04330
C                                                                       LEV04340
      WRITE(6,300)                                                      LEV04350
  300 FORMAT(/' ALL COMPUTED GOOD T-EIGENVALUES ARE MULTIPLE'/          LEV04360
     1 ' THEREFORE ALL SUCH EIGENVALUES ARE ASSUMED TO HAVE CONVERGED') LEV04370
C                                                                       LEV04380
      ICONV = 1                                                         LEV04390
      GO TO 350                                                         LEV04400
C                                                                       LEV04410
  310 CONTINUE                                                          LEV04420
C                                                                       LEV04430
C-----------------------------------------------------------------------LEV04440
C     SUBROUTINE INVERR COMPUTES ERROR ESTIMATES FOR ISOLATED GOOD      LEV04450
C     T-EIGENVALUES USING INVERSE ITERATION ON T(1,MEV). ON RETURN      LEV04460
C     G(J) = MINIMUM GAP IN T(1,MEV) FOR EACH VS(J), J=1,NDIS           LEV04470
C     G(MEV+I) = BETAM*|U(MEV)| = ERROR ESTIMATE FOR ISOLATED GOOD      LEV04480
C             T-EIGENVALUES, WHERE I = 1, NISO AND  BETAM = BETA(MEV+1) LEV04490
C             U(MEV) IS MEVTH COMPONENT OF THE UNIT EIGENVECTOR OF T    LEV04500
C             CORRESPONDING TO THE ITH ISOLATED GOOD T-EIGENVALUE.      LEV04510
C     A NEGATIVE ERROR ESTIMATE MEANS THAT FOR THAT PARTICULAR          LEV04520
C     EIGENVALUE THE INVERSE ITERATION DID NOT CONVERGE IN <= MXINIT    LEV04530
C     STEPS AND THAT THE CORRESPONDING ERROR ESTIMATE IS QUESTIONABLE.  LEV04540
C                                                                       LEV04550
C     V2 CONTAINS THE ISOLATED GOOD T-EIGENVALUES                       LEV04560
C     V1 CONTAINS THE MINGAPS TO THE NEAREST DISTINCT  EIGENVALUE       LEV04570
C        OF T(1,MEV) FOR EACH ISOLATED GOOD T-EIGENVALUE IN V2.         LEV04580
C     VS CONTAINS THE NDIS DISTINCT EIGENVALUES OF T(1,MEV)             LEV04590
C     MP CONTAINS THE CORRESPONDING CODED T-MULTIPLICITIES              LEV04600
C                                                                       LEV04610
      IT = MXINIT                                                       LEV04620
      CALL INVERR(ALPHA,BETA,V1,V2,VS,EPSM,G,MP,MEV,MMB,NDIS,NISO,N,    LEV04630
     1    RHSEED,IT,IWRITE)                                             LEV04640
C                                                                       LEV04650
C-----------------------------------------------------------------------LEV04660
C                                                                       LEV04670
C     SIMPLE CHECK FOR CONVERGENCE. CHECKS TO SEE IF ALL OF THE ERROR   LEV04680
C     ESTIMATES ARE SMALLER THAN CONTOL = BETAM*1.D-10.                 LEV04690
C     IF THIS TEST IS SATISFIED, THEN CONVERGENCE FLAG, ICONV IS SET    LEV04700
C     TO 1.  TYPICALLY ERROR ESTIMATES ARE VERY CONSERVATIVE.          LEV04710
C                                                                       LEV04720
      WRITE(6,320) CONTOL                                               LEV04730
  320 FORMAT(/' CONVERGENCE IS TESTED USING THE CONVERGENCE TOLERANCE', LEV04740
     1E13.4/)                                                           LEV04750
C                                                                       LEV04760
      II = MEV +1                                                       LEV04770
      IF = MEV+NISO                                                     LEV04780
      DO 330 I = II,IF                                                  LEV04790
      IF (ABS(G(I)).GT.CONTOL) GO TO 350                                LEV04800
  330 CONTINUE                                                          LEV04810
      ICONV = 1                                                         LEV04820
      MMB = NMEVS                                                       LEV04830
C                                                                       LEV04840
      WRITE(6,340) CONTOL                                               LEV04850
  340 FORMAT(' ALL COMPUTED ERROR ESTIMATES WERE LESS THAN',E15.4/      LEV04860
```

```
      1 ' THEREFORE PROCEDURE TERMINATES'/)                         LEV04870
C                                                                   LEV04880
  350 CONTINUE                                                      LEV04890
C                                                                   LEV04900
C    IF CONVERGENCE IS INDICATED, THAT IS ICONV = 1 ,THEN          LEV04910
C    THE SUBROUTINE PRTEST IS CALLED TO CHECK FOR ANY CONVERGED    LEV04920
C    T-EIGENVALUES THAT HAVE BEEN MISLABELLED AS SPURIOUS BECAUSE  LEV04930
C    THE PROJECTION OF THEIR EIGENVECTOR(S) ON THE STARTING        LEV04940
C    VECTOR WERE TOO SMALL.                                        LEV04950
C    NUMERICAL TESTS INDICATE THAT SUCH EIGENVALUES ARE RARE.      LEV04960
C    IF FOR SOME REASON MANY OF THESE HIDDEN EIGENVALUES APPEAR    LEV04970
C    ON SOME RUN, YOU CAN BE CERTAIN THAT SOMETHING IS FOULED UP.  LEV04980
C                                                                   LEV04990
      IF (ICONV.EQ.0) GO TO 480                                     LEV05000
C                                                                   LEV05010
C------------------------------------------------------------------LEV05020
C                                                                   LEV05030
      CALL PRTEST (ALPHA,BETA,VS,TKMAX,EPSM,RELTOL,SCALE3,SCALE4,   LEV05040
     1 MP,NDIS,MEV,IPROJ)                                           LEV05050
C                                                                   LEV05060
C------------------------------------------------------------------LEV05070
C                                                                   LEV05080
      IF(IPROJ.EQ.0) GO TO 470                                      LEV05090
C                                                                   LEV05100
      IF(IDIST.EQ.1)  WRITE(11,360) IPROJ                           LEV05110
  360 FORMAT(' SUBROUTINE PRTEST WANTS TO RELABEL',I6,' SPURIOUS T-EIGENLEV05120
     1VALUES'/' WE ACCEPT RELABELLING ONLY IF LAST COMPONENT OF T-EIGENVLEV05130
     1ECTOR IS L.T. 1.D-10'/)                                      LEV05140
C                                                                   LEV05150
      IIX = RHSEED                                                  LEV05160
C                                                                   LEV05170
C------------------------------------------------------------------LEV05180
C                                                                   LEV05190
      CALL GENRAN(IIX,G,MEV)                                        LEV05200
C                                                                   LEV05210
C------------------------------------------------------------------LEV05220
C                                                                   LEV05230
      ITEN = -10                                                    LEV05240
      NISOM = NISO + MEV                                            LEV05250
      IWRITO = IWRITE                                               LEV05260
      IWRITE = 0                                                    LEV05270
C                                                                   LEV05280
      DO 390 J = 1,NDIS                                             LEV05290
      IF(MP(J).NE.ITEN) GO TO 390                                   LEV05300
      TO = VS(J)                                                    LEV05310
C                                                                   LEV05320
C------------------------------------------------------------------LEV05330
C                                                                   LEV05340
      IT = MXINIT                                                   LEV05350
      CALL INVERM(ALPHA,BETA,V1,V2,TO,TEMP,T1,EPSM,G,MEV,IT,IWRITE) LEV05360
C                                                                   LEV05370
C------------------------------------------------------------------LEV05380
C                                                                   LEV05390
      IF(TEMP.LE.1.D-10) GO TO 380                                  LEV05400
C    ERROR ESTIMATE WAS NOT SMALL REJECT RELABELLING OF THIS EIGENVALUELEV05410
```

```
      IF(IDIST.EQ.1)  WRITE(11,370) J,T0,TEMP                   LEV05420
  370 FORMAT(/' LAST COMPONENT FOR',I6,'TH EIGENVALUE',E20.12/' IS TOO LLEV05430
     1ARGE = ',E15.6,' SO DO NOT ACCEPT PRTEST RELABELLING'/)    LEV05440
      MP(J) = 0                                                  LEV05450
      IPROJ = IPROJ - 1                                          LEV05460
      GO TO 390                                                  LEV05470
C     RELABELLING ACCEPTED                                       LEV05480
  380 NISOM = NISOM + 1                                          LEV05490
      G(NISOM) = BETAM*TEMP                                      LEV05500
  390 CONTINUE                                                   LEV05510
      IWRITE = IWRIT0                                            LEV05520
C                                                                LEV05530
      IF(IPROJ.EQ.0) GO TO 430                                   LEV05540
      WRITE(6,400) IPROJ                                         LEV05550
  400 FORMAT(/I6,' T-EIGENVALUES WERE RECLASSIFIED AS GOOD.'/    LEV05560
     1' THESE ARE IDENTIFIED IN FILE 3 BY A T-MULTIPLICITY OF -10'/' USELEV05570
     2R SHOULD INSPECT EACH TO MAKE SURE NEIGHBORS HAVE CONVERGED'/)   LEV05580
C                                                                LEV05590
      IF(IDIST.EQ.1)  WRITE(11,410) IPROJ                        LEV05600
  410 FORMAT(/I6,' T-EIGENVALUES WERE RELABELLED AS GOOD'/       LEV05610
     1' BELOW IS CORRECTED T-MULTIPLICITY PATTERN'/)             LEV05620
C                                                                LEV05630
      WRITE(6,420) NDIS, (MP(I), I=1,NDIS)                       LEV05640
      IF(IDIST.EQ.1)  WRITE(11,420) NDIS, (MP(I), I=1,NDIS)      LEV05650
  420 FORMAT(/I6,' = NDIS, T-MULTIPLICITIES (0 MEANS  SPURIOUS)'/  LEV05660
     1 6X, ' (-10) MEANS SPURIOUS T-EIGENVALUE RELABELLED AS GOOD'/(20I4LEV05670
     1))                                                         LEV05680
C                                                                LEV05690
C     RECALCULATE MINGAPS FOR DISTINCT T(1,MEV) EIGENVALUES.     LEV05700
  430 NM1 = NDIS - 1                                             LEV05710
      G(NDIS) = VS(NM1)-VS(NDIS)                                 LEV05720
      G(1) = VS(2)-VS(1)                                         LEV05730
C                                                                LEV05740
      DO 440 J = 2,NM1                                           LEV05750
      T0 = VS(J)-VS(J-1)                                         LEV05760
      T1 = VS(J+1)-VS(J)                                         LEV05770
      G(J) = T1                                                  LEV05780
      IF (T0.LT.T1) G(J) = -T0                                   LEV05790
  440 CONTINUE                                                   LEV05800
      IF(IPROJ.EQ.0) GO TO 470                                   LEV05810
C     WRITE TO FILE 4 ERROR ESTIMATES FOR THOSE T-EIGENVALUES RELABELLEDLEV05820
      NGOOD = 0                                                  LEV05830
      DO 450 J = 1,NDIS                                          LEV05840
      IF(MP(J).EQ.0) GO TO 450                                   LEV05850
      NGOOD = NGOOD + 1                                          LEV05860
      IF(MP(J).NE.ITEN) GO TO 450                                LEV05870
      T0 = VS(J)                                                 LEV05880
      NISO = NISO + 1                                            LEV05890
      NISOM = MEV + NISO                                         LEV05900
      WRITE(4,460) NGOOD,T0,G(NISOM),G(J)                        LEV05910
  450 CONTINUE                                                   LEV05920
  460 FORMAT(I10,E25.16,2E14.3)                                  LEV05930
C                                                                LEV05940
  470 CONTINUE                                                   LEV05950
C                                                                LEV05960
```

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

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

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

2.4   LEVEC: Main Program, Eigenvector Computations

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

```
C       THE PROGRAM REQUIRES THAT KMAX BE AT LEAST 11*MEV/8 + 12.  IF    LEV00460
C       KMAX IS NOT THIS LARGE, THEN THE PROGRAM RESETS KMAX TO THIS     LEV00470
C       SIZE AND EXTENDS THE ALPHA, BETA HISTORY IF REQUIRED.           LEV00480
C       THUS, THE DIMENSIONS OF THE ALPHA AND BETA ARRAYS MUST BE       LEV00490
C       LARGE ENOUGH TO ALLOW FOR THIS POSSIBILITY.                     LEV00500
C       REMEMBER THAT THE BETA ARRAY, BETA(J), IS SUCH THAT             LEV00510
C       J = 1,..., KMAX+1.  SO IF THE KMAX USED BY THE PROGRAM          LEV00520
C       IS TO BE 3000, THEN BETA MUST BE OF LENGTH AT LEAST 3001.       LEV00530
C                                                                       LEV00540
C-----------------------------------------------------------------------LEV00550
      DOUBLE PRECISION  ALPHA(5000),BETA(5001)                          LEV00560
      DOUBLE PRECISION  V1(5000),V2(5000)                               LEV00570
      DOUBLE PRECISION  RITVEC(30000),TVEC(30000),GOODEV(50),EVNEW(50)  LEV00580
      DOUBLE PRECISION  EVAL,EVALN,TOLN,TTOL,ERTOL,ALFA,BATA            LEV00590
      DOUBLE PRECISION  MULTOL,SCALE0,STUTOL,BTOL,LB,UB                 LEV00600
      DOUBLE PRECISION  ONE,ZERO,MACHEP,EPSM,TEMP,SUM,ERRMIN,BKMIN      LEV00610
      DOUBLE PRECISION  RELTOL,ERROR,TERROR,TLAST(50)                   LEV00620
      REAL  G(10000),AMINGP(50),TMINGP(50),EXPLAN(20)                   LEV00630
      REAL  TERR(50),ERR(50),ERRDGP(50),RNORM(50),TBETA(50)            LEV00640
      INTEGER  MP(50),M1(50),M2(50),MA(50),ML(50),MINT(50),MFIN(50)     LEV00650
      INTEGER  SVSEED,SVSOLD,RHSEED,IDELTA(50)                          LEV00660
      INTEGER  MBOUND,NTVCON,SVTVEC,TVSTOP,LVCONT,ERCONT,TFLAG          LEV00670
      DOUBLE PRECISION  FINPRO                                          LEV00680
      DOUBLE PRECISION  DABS, DMAX1, DSQRT, DFLOAT                      LEV00690
      REAL ABS                                                          LEV00700
      INTEGER  IABS                                                     LEV00710
C-----------------------------------------------------------------------LEV00720
      EXTERNAL CMATV                                                    LEV00730
      DATA MACHEP/Z3410000000000000/                                   LEV00740
      EPSM = 2.D0*MACHEP                                                LEV00750
C-----------------------------------------------------------------------LEV00760
C                                                                       LEV00770
C       ARRAYS MUST BE DIMENSIONED AS FOLLOWS:                          LEV00780
C       1.   ALPHA:  >= KMAXN,  BETA: >= (KMAXN+1) WHERE KMAXN, THE     LEV00790
C                    LARGEST SIZE T-MATRIX CONSIDERED BY THE PROGRAM,   LEV00800
C                    IS THE LARGER OF THE SIZE OF THE ALPHA, BETA HISTORYLEV00810
C                    PROVIDED ON FILE 2 (IF ANY ) AND THE SIZE WHICH THE LEV00820
C                    PROGRAM SPECIFIES INTERNALLY, THIS LATTER IS ALWAYS LEV00830
C                    < = 11*MEV / 8  +  12, WHERE MEV IS THE SIZE       LEV00840
C                    T-MATRIX THAT WAS USED IN THE CORRESPONDING EIGENVALUE LEV00850
C                    COMPUTATIONS.                                      LEV00860
C       2.   V1:  >= MAX(N,KMAX)                                        LEV00870
C       3.   V2:  >= N                                                  LEV00880
C       4.   G:  >= MAX(N,KMAX)                                         LEV00890
C       5.   RITVEC:  >= N*NGOOD, WHERE NGOOD IS NUMBER OF EIGENVALUES  LEV00900
C                    SUPPLIED TO THIS PROGRAM.                          LEV00910
C       6.   TVEC:  >= CUMULATIVE LENGTH OF ALL THE T-EIGENVECTORS      LEV00920
C            NEEDED TO GENERATE THE DESIRED RITZ VECTORS.  AN EDUCATED  LEV00930
C            GUESS AT AN APPROPRIATE LENGTH CAN BE OBTAINED BY RUNNING THE LEV00940
C            PROGRAM WITH THE FLAG MBOUND = 1 AND MULTIPLYING THE       LEV00950
C            RESULTING SIZE BY 5/4.                                     LEV00960
C       7.   GOODEV, AMINGP, TMINGP, TERR, ERR, ERRGDP, RNORM, TBETA,   LEV00970
C            TLAST, EVNEW, MP, MA, M1, M2, MINT, MFIN AND IDELTA ALL MUST LEV00980
C            BE >= NGOOD.                                               LEV00990
C                                                                       LEV01000
```

```
C-------------------------------------------------------------------LEV01010
C     OUTPUT HEADER                                                  LEV01020
      WRITE(6,10)                                                    LEV01030
   10 FORMAT(/' LANCZOS EIGENVECTOR PROCEDURE FOR REAL SYMMETRIC MATRICELEV01040
     1S'/)                                                           LEV01050
C                                                                    LEV01060
C     SET PROGRAM PARAMETERS                                         LEV01070
C     USER MUST NOT MODIFY SCALE0                                    LEV01080
      SCALE0 = 5.0D0                                                 LEV01090
      ZERO = 0.0D0                                                   LEV01100
      ONE = 1.0D0                                                    LEV01110
      MPMIN = -1000                                                  LEV01120
C     SET CONVERGENCE CRITERION FOR T-EIGENVECTORS.                  LEV01130
      ERTOL = 1.D-10                                                 LEV01140
C                                                                    LEV01150
C     READ USER-SPECIFIED PARAMETER FROM INPUT FILE 5 (FREE FORMAT)  LEV01160
C                                                                    LEV01170
C     READ USER-PROVIDED HEADER FOR RUN                             LEV01180
      READ(5,20) EXPLAN                                             LEV01190
      WRITE(6,20) EXPLAN                                            LEV01200
   20 FORMAT(20A4)                                                  LEV01210
C                                                                    LEV01220
C     READ IN THE MAXIMUM PERMISSIBLE DIMENSIONS FOR THE TVEC ARRAY  LEV01230
C     (MDIMTV), FOR THE RITVEC ARRAY (MDIMRV), AND FOR THE BETA      LEV01240
C     ARRAY (MBETA).                                                 LEV01250
      READ(5,20) EXPLAN                                             LEV01260
      READ(5,*) MDIMTV, MDIMRV, MBETA                              LEV01270
C                                                                    LEV01280
C     READ IN RELATIVE TOLERANCE (RELTOL) USED IN DETERMINING        LEV01290
C     APPROPRIATE SIZES FOR THE T-MATRICES TO BE USED IN THE RITZ    LEV01300
C     VECTOR COMPUTATIONS.                                           LEV01310
      READ(5,20) EXPLAN                                             LEV01320
      READ(5,*) RELTOL                                             LEV01330
C                                                                    LEV01340
C     SET FLAGS TO 0 OR 1:                                           LEV01350
C     MBOUND = 1:  PROGRAM TERMINATES AFTER COMPUTING 1ST GUESSES    LEV01360
C                  ON APPROPRIATE T-SIZES FOR USE IN THE RITZ VECTOR LEV01370
C                  COMPUTATIONS                                      LEV01380
C     NTVCON = 0:  PROGRAM TERMINATES IF THE TVEC ARRAY IS NOT       LEV01390
C                  LARGE ENOUGH TO HOLD ALL THE T-EIGENVECTORS REQUIRED.LEV01400
C     SVTVEC = 0:  THE T-EIGENVECTORS ARE NOT WRITTEN TO FILE 11     LEV01410
C                  UNLESS TVSTOP = 1                                 LEV01420
C     SVTVEC = 1:  WRITE THE T-EIGENVECTORS TO FILE 11.              LEV01430
C     TVSTOP = 1:  PROGRAM TERMINATES AFTER COMPUTING THE            LEV01440
C                  T-EIGENVECTORS                                    LEV01450
C     LVCONT = 0:  PROGRAM TERMINATES IF THE NUMBER OF T-EIGENVECTORS LEV01460
C                  COMPUTED IS NOT EQUAL TO THE NUMBER OF RITZ       LEV01470
C                  VECTORS REQUESTED.                                LEV01480
C     ERCONT = 0:  MEANS FOR ANY GIVEN EIGENVALUE, A RITZ VECTOR     LEV01490
C                  WILL NOT BE COMPUTED FOR THAT EIGENVALUE UNLESS   LEV01500
C                  A T-EIGENVECTOR HAS BEEN IDENTIFIED WITH A LAST   LEV01510
C                  COMPONENT WHICH SATISFIES THE SPECIFIED           LEV01520
C                  CONVERGENCE CRITERION.                            LEV01530
C     ERCONT = 1:  MEANS FOR ANY GIVEN EIGENVALUE, A RITZ VECTOR     LEV01540
C                  WILL BE COMPUTED.  IF A T-EIGENVECTOR CANNOT      LEV01550
```

```
C                BE IDENTIFIED WHICH SATISFIES THE LAST           LEV01560
C                COMPONENT CRITERION, THEN THE PROGRAM WILL        LEV01570
C                USE THE T-VECTOR THAT CAME CLOSEST TO             LEV01580
C                SATISFYING THE CRITERION.                         LEV01590
C      IWRITE = 1:  EXTENDED OUTPUT OF INTERMEDIATE COMPUTATIONS   LEV01600
C                IS WRITTEN TO FILE 6                              LEV01610
C      IREAD = 0:   ALPHA/BETA FILE IS REGENERATED.               LEV01620
C      IREAD = 1:   ALPHA/BETA FILE USED IN EIGENVALUE COMPUTATIONS LEV01630
C                IS READ IN AND EXTENDED IF NECESSARY.  IN BOTH    LEV01640
C                CASES IREAD = 0 OR 1, THE LANCZOS VECTORS ARE     LEV01650
C                ALWAYS REGENERATED FOR THE RITZ VECTOR            LEV01660
C                COMPUTATIONS                                      LEV01670
C                                                                  LEV01680
      READ(5,20) EXPLAN                                            LEV01690
      READ(5,*) MBOUND,NTVCON,SVTVEC,IREAD                         LEV01700
C                                                                  LEV01710
      READ(5,20) EXPLAN                                            LEV01720
      READ(5,*) TVSTOP,LVCONT,ERCONT,IWRITE                        LEV01730
      IF (TVSTOP.EQ.1) SVTVEC = 1                                  LEV01740
C                                                                  LEV01750
C     READ IN SEED (RHSEED) FOR GENERATING RANDOM STARTING VECTOR  LEV01760
C     FOR INVERSE ITERATION ON THE T-MATRICES.                     LEV01770
      READ(5,20) EXPLAN                                            LEV01780
      READ(5,*) RHSEED                                             LEV01790
C                                                                  LEV01800
C     READ IN MATNO = MATRIX/RUN IDENTIFICATION NUMBER AND         LEV01810
C     N = ORDER OF A-MATRIX                                        LEV01820
      READ(5,20) EXPLAN                                            LEV01830
      READ(5,*) MATNO,N                                            LEV01840
C                                                                  LEV01850
C------------------------------------------------------------------LEV01860
C     INITIALIZE THE ARRAYS FOR THE USER-SPECIFIED MATRIX          LEV01870
C     AND PASS THE STORAGE LOCATIONS OF THESE ARRAYS TO THE        LEV01880
C     MATRIX-VECTOR MULTIPLY SUBROUTINE CMATV.                     LEV01890
C                                                                  LEV01900
      CALL USPEC(N,MATNO)                                          LEV01910
C                                                                  LEV01920
C------------------------------------------------------------------LEV01930
C     MASK UNDERFLOW AND OVERFLOW                                  LEV01940
      CALL MASK                                                    LEV01950
C                                                                  LEV01960
C------------------------------------------------------------------LEV01970
C     WRITE RUN PARAMETERS OUT TO FILE 6                           LEV01980
C                                                                  LEV01990
      WRITE(6,30) MATNO,N                                          LEV02000
   30 FORMAT(/' MATRIX IDENTIFICATION NO. = ',I10,' ORDER OF A = ',I5) LEV02010
C                                                                  LEV02020
      WRITE(6,40) MBOUND,NTVCON,SVTVEC,IREAD                       LEV02030
   40 FORMAT(/3X,'MBOUND',3X,'NTVCON',3X,'SVTVEC',3X,'IREAD'/3I9,I8) LEV02040
C                                                                  LEV02050
      WRITE(6,50) TVSTOP,LVCONT,ERCONT,IWRITE                      LEV02060
   50 FORMAT(/3X,'TVSTOP',3X,'LVCONT',3X,'ERCONT',3X,'IWRITE'/4I9) LEV02070
C                                                                  LEV02080
      WRITE(6,60) MDIMTV,MDIMRV,MBETA                              LEV02090
   60 FORMAT(/3X,'MDIMTV',3X,'MDIMRV',3X,'MBETA'/2I9,I8)           LEV02100
```

```
C                                                                      LEV02110
      WRITE(6,70) RELTOL,RHSEED                                         LEV02120
   70 FORMAT(/7X,'RELTOL',3X,'RHSEED'/E13.4,I9)                         LEV02130
C                                                                      LEV02140
C                                                                      LEV02150
C     FROM FILE 3 READ IN THE NUMBER OF EIGENVALUES (NGOOD) FOR WHICH   LEV02160
C     EIGENVECTORS ARE REQUESTED, THE ORDER (MEV) OF THE LANCZOS        LEV02170
C     TRIDIAGONAL MATRIX USED IN COMPUTING THESE EIGENVALUES, THE       LEV02180
C     ORDER (NOLD) OF THE USER-SPECIFIED MATRIX USED IN THE EIGENVALUE  LEV02190
C     COMPUTATIONS, THE SEED (SVSEED) USED FOR GENERATING THE STARTING  LEV02200
C     VECTOR THAT WAS USED IN THOSE LANCZOS EIGENVALUE COMPUTATIONS,    LEV02210
C     AND THE MATRIX/RUN IDENTIFICATION NUMBER (MATOLD) USED IN THOSE   LEV02220
C     COMPUTATIONS.  ALSO READ IN THE NUMBER (NDIS) OF DISTINCT         LEV02230
C     EIGENVALUES OF T(1,MEV) THAT WERE COMPUTED BUT THIS VALUE IS      LEV02240
C     NOT USED IN THE EIGENVECTOR COMPUTATIONS.                         LEV02250
C                                                                      LEV02260
      READ(3,80) NGOOD,NDIS,MEV,NOLD,SVSEED,MATOLD                      LEV02270
   80 FORMAT(4I6,I12,I8)                                                LEV02280
C                                                                      LEV02290
C     READ IN THE T-MULTIPLICITY TOLERANCE USED IN THE BISEC SUBROUTINE LEV02300
C     DURING THE COMPUTATION OF THE GIVEN EIGENVALUES.                  LEV02310
C     ALSO READ IN THE FLAG IB.  IF IB < 0, THEN SOME BETA(I) IN THE    LEV02320
C     T-MATRIX FILE PROVIDED ON FILE 2 FAILED THE ORTHOGONALITY         LEV02330
C     TEST IN THE TNORM SUBROUTINE.  USER SHOULD NOTE THAT THIS VECTOR  LEV02340
C     PROGRAM PROCEEDS INDEPENDENTLY OF THE SIZE OF THE BETA USED.      LEV02350
C                                                                      LEV02360
      READ(3,90) MULTOL,IB,BTOL                                         LEV02370
   90 FORMAT(E20.12,I6,E13.4)                                           LEV02380
C                                                                      LEV02390
      TEMP = DFLOAT(MEV+1000)                                           LEV02400
      TTOL = MULTOL/TEMP                                                LEV02410
      WRITE(6,100) MULTOL,TTOL                                          LEV02420
  100 FORMAT(/' T-MULTIPLICITY TOLERANCE USED IN THE EIGENVALUE COMPUTATLEV02430
     1IONS WAS',E13.4/' SCALED MACHINE EPSILON IS',E13.4)               LEV02440
C                                                                      LEV02450
C     CONTINUE WRITE TO FILE 6 OF THE PARAMETERS FOR THIS RUN           LEV02460
C                                                                      LEV02470
      WRITE(6,110)NGOOD,NDIS,MEV,NOLD,MATOLD,SVSEED,MULTOL,IB,BTOL      LEV02480
  110 FORMAT(/' EIGENVALUES SUPPLIED ARE READ IN FROM FILE 3'/' FILE 3  LEV02490
     1HEADER IS'/4X,'NG',2X,'NDIS',3X,'MEV',2X,'NOLD',2X,'MATOLD',4X,   LEV02500
     1'SVSEED',6X,'MULTOL',6X,'IB',9X,'BTOL'/4I6,I8,I10,E12.3,I8,E13.4/)LEV02510
C                                                                      LEV02520
C     IS THE ARRAY RITVEC LONG ENOUGH TO HOLD ALL OF THE DESIRED        LEV02530
C     RITZ VECTORS (APPROXIMATE EIGENVECTORS)?                          LEV02540
      NMAX = NGOOD*N                                                    LEV02550
      IF(MBOUND.NE.0) GO TO 120                                         LEV02560
      IF(TVSTOP.NE.1.AND.NMAX.GT.MDIMRV) GO TO 1310                     LEV02570
C                                                                      LEV02580
C     CHECK THAT THE ORDER N AND THE MATRIX IDENTIFICATION NUMBER       LEV02590
C     MATNO SPECIFIED BY THE USER AGREE WITH THOSE READ IN FROM         LEV02600
C     FILE 3.                                                          LEV02610
  120 ITEMP = (NOLD-N)**2+(MATOLD-MATNO)**2                             LEV02620
      IF (ITEMP.NE.0) GO TO 1330                                        LEV02630
C                                                                      LEV02640
C     READ IN FROM FILE 3, THE T-MULTIPLICITIES OF THE EIGENVALUES      LEV02650
```

```
C      WHOSE EIGENVECTORS ARE TO BE COMPUTED, THE VALUES OF THESE      LEV02660
C      EIGENVALUES AND THEIR MINIMAL GAPS AS EIGENVALUES OF THE        LEV02670
C      USER-SPECIFIED MATRIX AND AS EIGENVALUES OF THE T-MATRIX.       LEV02680
C                                                                      LEV02690
       READ(3,20) EXPLAN                                               LEV02700
       READ(3,130) (MP(J),GOODEV(J),TMINGP(J),AMINGP(J), J=1,NGOOD)    LEV02710
   130 FORMAT(6X,I6,E25.16,2E14.3)                                     LEV02720
C                                                                      LEV02730
       WRITE(6,140) (J,GOODEV(J),MP(J),TMINGP(J),AMINGP(J), J=1,NGOOD) LEV02740
   140 FORMAT(/' EIGENVALUES READ IN, T-MULTIPLICITIES, T-GAPS AND A-GAPSLEV02750
      1 '/4X,' J ',5X,'GOOD EIGENVALUE',5X,'MULT',4X,' TMINGAP ',4X,   LEV02760
      1'  AMINGAP '/(I6,E25.16,I4,2E15.4))                             LEV02770
C                                                                      LEV02780
C      READ IN ERROR ESTIMATES                                         LEV02790
       WRITE(6,150) MEV,SVSEED                                         LEV02800
   150 FORMAT(/' THESE EIGENVALUES WERE COMPUTED USING A T-MATRIX OF   LEV02810
      1ORDER ',I5/' AND SEED FOR RANDOM NUMBER GENERATOR =',I12)       LEV02820
C      CHECK WHETHER OR NOT THERE ARE ANY T-ISOLATED EIGENVALUES IN    LEV02830
C      THE EIGENVALUES PROVIDED                                        LEV02840
       DO 160 J=1,NGOOD                                                LEV02850
       IF(MP(J).EQ.1) GO TO 170                                        LEV02860
   160 CONTINUE                                                        LEV02870
       GO TO 190                                                       LEV02880
   170 READ(4,20) EXPLAN                                               LEV02890
       READ(4,20) EXPLAN                                               LEV02900
       READ(4,20) EXPLAN                                               LEV02910
       READ(4,180) NISO                                                LEV02920
   180 FORMAT(18X,I6)                                                  LEV02930
       READ(4,20) EXPLAN                                               LEV02940
       READ(4,20) EXPLAN                                               LEV02950
       READ(4,20) EXPLAN                                               LEV02960
   190 DO 220 J=1,NGOOD                                                LEV02970
       ERR(J) = 0.D0                                                   LEV02980
       IF(MP(J).NE.1) GO TO 220                                        LEV02990
       READ(4,200) EVAL, ERR(J)                                        LEV03000
   200 FORMAT(10X,E25.16,E14.3)                                        LEV03010
       IF(DABS(EVAL - GOODEV(J)).LT.1.D-10) GO TO 220                  LEV03020
       WRITE(6,210) EVAL,GOODEV(J)                                     LEV03030
   210 FORMAT(' PROBLEM WITH READ IN OF ERROR ESTIMATES'/' EIGENVALUE REALEV03040
      1D IN',E20.12,' DOES NOT MATCH GOODEV(J) ='/E20.12)             LEV03050
       GO TO 1550                                                      LEV03060
C                                                                      LEV03070
   220 CONTINUE                                                        LEV03080
C                                                                      LEV03090
       WRITE(6,230) (J,GOODEV(J),ERR(J), J=1,NGOOD)                    LEV03100
   230 FORMAT(' ERROR ESTMATES ='/4X,' J',5X,'EIGENVALUE',10X,' ESTIMATE LEV03110
      1'/(I6,E20.12,E14.3))                                            LEV03120
C                                                                      LEV03130
       IF(IREAD.EQ.0)  GO TO 330                                       LEV03140
C                                                                      LEV03150
C      READ IN THE SIZE OF THE T-MATRIX PROVIDED ON FILE 2.  READ IN   LEV03160
C      THE ORDER OF THE USER-SPECIFIED MATRIX , THE SEED FOR THE       LEV03170
C      RANDOM NUMBER GENERATOR, AND THE MATRIX/TEST IDENTIFICATION     LEV03180
C      NUMBER THAT WERE USED IN THE LANCZOS EIGENVALUE COMPUTATIONS.   LEV03190
C      THESE ARE USED IN A CONSISTENCY CHECK                           LEV03200
```

```
C     IF FLAG IREAD = 0 REGENERATE ALPHA, BETA                 LEV03210
C                                                              LEV03220
      READ(2,240) KMAX,NOLD,SVSOLD,MATOLD                      LEV03230
  240 FORMAT(2I6,I12,I8)                                       LEV03240
C                                                              LEV03250
      WRITE(6,250) KMAX,NOLD,SVSOLD,MATOLD                     LEV03260
  250 FORMAT(/' READ IN THE T-MATRICES STORED ON FILE 2'/' FILE 2 HEADERLEV03270
     1 IS'/2X,'KMAX',2X,'NOLD',6X,'SVSOLD',2X,'MATOLD'/2I6,I12,I8/) LEV03280
C                                                              LEV03290
C     CHECK THAT THE ORDER, THE MATRIX/TEST IDENTIFICATION NUMBER LEV03300
C     AND THE SEED FOR THE RANDOM NUMBER GENERATOR USED IN THE LEV03310
C     LANCZOS COMPUTATIONS THAT GENERATED THE ALPHA,BETA FILE  LEV03320
C     BEING USED AGREE WITH WHAT THE USER HAS SPECIFIED.       LEV03330
      IF (NOLD.NE.N.OR.MATOLD.NE.MATNO.OR.SVSOLD.NE.SVSEED) GO TO 1350 LEV03340
C                                                              LEV03350
      KMAX1 = KMAX + 1                                         LEV03360
C                                                              LEV03370
C     READ IN THE T-MATRICES FROM FILE 2.  THESE ARE USED TO GENERATE LEV03380
C     THE T-EIGENVECTORS THAT WILL BE USED IN THE RITZ VECTOR  LEV03390
C     COMPUTATIONS.  HISTORY MUST BE STORED IN MACHINE FORMAT  LEV03400
C     ((4Z20) FOR IBM/3081)                                    LEV03410
C                                                              LEV03420
      READ(2,260) (ALPHA(J), J=1,KMAX)                         LEV03430
      READ(2,260) (BETA(J), J=1,KMAX1)                         LEV03440
  260 FORMAT(4Z20)                                             LEV03450
C                                                              LEV03460
      READ(2,260) (V1(J), J=1,N)                               LEV03470
      READ(2,260) (V2(J), J=1,N)                               LEV03480
C                                                              LEV03490
C     KMAX MAY BE ENLARGED IF THE SIZE AT WHICH THE EIGENVALUE LEV03500
C     COMPUTATIONS WERE PERFORMED IS ESSENTIALLY KMAX AND      LEV03510
C     THERE IS AT LEAST ONE EIGENVALUE THAT IS T-SIMPLE AND    LEV03520
C     T-ISOLATED, IN THE SENSE THAT IF ITS NEAREST NEIGHBOR IS TOO LEV03530
C     CLOSE THAT NEIGHBOR IS A 'GOOD' T-EIGENVALUE.            LEV03540
      DO 270 J = 1,NGOOD                                       LEV03550
      IF(MP(J).EQ.1) GO TO 290                                 LEV03560
  270 CONTINUE                                                 LEV03570
      WRITE(6,280)                                             LEV03580
  280 FORMAT(/' ALL EIGENVALUES USED ARE T-MULTIPLE OR CLOSE TO SPURIOUSLEV03590
     1 T-EIGENVALUES'/' SO KMAX IS NOT INCREASED')            LEV03600
      IF(KMAX.LT.MEV) GO TO 1370                               LEV03610
      GO TO 310                                                LEV03620
C                                                              LEV03630
  290 KMAXN= 11*MEV/8 + 12                                     LEV03640
      IF(MBETA.LE.KMAXN) GO TO 1530                            LEV03650
      IF(KMAX.GE.KMAXN )  GO TO 310                            LEV03660
      WRITE(6,300) KMAX, KMAXN                                 LEV03670
  300 FORMAT(' ENLARGE KMAX FROM ',I6,' TO ',I6)               LEV03680
      MOLD1 = KMAX + 1                                         LEV03690
      KMAX = KMAXN                                             LEV03700
      GO TO 380                                                LEV03710
C                                                              LEV03720
  310 WRITE(6,320) KMAX                                        LEV03730
  320 FORMAT(/' T-MATRICES HAVE BEEN READ IN FROM FILE 2'/' THE LARGEST LEV03740
     1SIZE T-MATRIX ALLOWED IS',I6/)                           LEV03750
```

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

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

```
  510 FORMAT(/' EV = GOODEV(J) IS A GOOD EIGENVALUE OF T(1,MEV)'/       LEV04860
     1' M1 = SMALLEST VALUE OF M SUCH THAT T(1,M) HAS AT LEAST'/        LEV04870
     1 '     ONE EIGENVALUE IN THE INTERVAL (EV-TOLN,EV+TOLN)'/         LEV04880
     1 '     TOLN(J) = DMAX1(GOODEV(J)*RELTOL, SCALE0*MULTOL)'/         LEV04890
     1 ' M2 = SMALLEST M (IF ANY) SUCH THAT IN THE ABOVE INTERVAL'/     LEV04900
     1 '     T(1,M) HAS AT LEAST TWO EIGENVALUES '/                     LEV04910
     1 ' IABS(MA(J)) = APPROPRIATE SIZE T-MATRIX FOR GOODEV(J)'/        LEV04920
     1 ' INITIAL VALUE OF MA(J) IS CHOSEN HEURISTICALLY'/               LEV04930
     1 ' PROGRAM LOOPS ON SIZE OF T-MATRIX TO GET BETTER SIZE'/         LEV04940
     1 ' END OF SIZES OF T-MATRICES FILE 10'///)                        LEV04950
C                                                                       LEV04960
C     TERMINATE AFTER COMPUTING 1ST GUESSES AT SIZES OF THE            LEV04970
C     T-MATRICES REQUIRED FOR THE GIVEN EIGENVALUES?                   LEV04980
      IF(MBOUND.EQ.1) GO TO 1390                                        LEV04990
C                                                                       LEV05000
C     IS THERE ROOM FOR ALL OF THE REQUESTED T-EIGENVECTORS?           LEV05010
      MTOL = 0                                                          LEV05020
      DO 520 J = 1,NGOOD                                                LEV05030
      IF(MP(J).EQ.MPMIN) GO TO 520                                      LEV05040
      MTOL = MTOL + IABS(MA(J))                                         LEV05050
  520 CONTINUE                                                          LEV05060
      MTOL = (5*MTOL)/4                                                 LEV05070
      IF(MTOL.GT.MDIMTV.AND.NTVCON.EQ.0) GO TO 1410                     LEV05080
C                                                                       LEV05090
C-----------------------------------------------------------------------LEV05100
C     GENERATE A RANDOM VECTOR TO BE USED REPEATEDLY BY               LEV05110
C     SUBROUTINE INVERM                                                LEV05120
C                                                                       LEV05130
      CALL GENRAN(RHSEED,G,KMAX)                                        LEV05140
C                                                                       LEV05150
C---------------------------------------------------------------------  LEV05160
C                                                                       LEV05170
C     LOOP ON GIVEN EIGENVALUES TO COMPUTE THE CORRESPONDING          LEV05180
C     T-EIGENVECTOR.                                                   LEV05190
C                                                                       LEV05200
      MTOL = 0                                                          LEV05210
      NTVEC = 0                                                         LEV05220
      ILBIS = 0                                                         LEV05230
      DO 710 J = 1,NGOOD                                                LEV05240
      ICOUNT = 0                                                        LEV05250
      ERRMIN = 10.D0                                                    LEV05260
      MABEST = MPMIN                                                    LEV05270
      IF(MP(J).EQ.MPMIN) GO TO 710                                      LEV05280
      TFLAG = 0                                                         LEV05290
      EVAL = GOODEV(J)                                                  LEV05300
      TEMP = DABS(EVAL)*RELTOL                                          LEV05310
      UB = EVAL + DMAX1(STUTOL,TEMP)                                    LEV05320
      LB = EVAL - DMAX1(STUTOL,TEMP)                                    LEV05330
  530 KMAXU = IABS(MA(J))                                               LEV05340
C                                                                       LEV05350
C     SELECT A SUITABLE INCREMENT FOR THE ORDERS OF THE T-MATRICES    LEV05360
C     TO BE CONSIDERED IN DETERMINING APPROPRIATE SIZES FOR THE RITZ  LEV05370
C     VECTOR COMPUTATIONS.                                             LEV05380
      IF(ICOUNT.GT.0) GO TO 550                                         LEV05390
C     SELECT IDELTA(J) BASED UPON THE T-MULTIPLICITY OBTAINED         LEV05400
```

```
      IF(M2(J).EQ.KMAX) GO TO 540                       LEV05410
C     M2 DETERMINED                                     LEV05420
      IDELTA(J) = ((3*M1(J) + 5*M2(J))/8  +  1 - IABS(MA(J)))/10 + 1  LEV05430
      GO TO 550                                         LEV05440
C     M2 NOT DETERMINED                                 LEV05450
  540 MAMAX = MINO((11*MEV)/8 + 12, (13*M1(J))/8 + 1)   LEV05460
      IDELTA(J) = (MAMAX - IABS(MA(J)))/10  +  1        LEV05470
  550 ICOUNT = ICOUNT + 1                               LEV05480
C                                                       LEV05490
C-----------------------------------------------------------------LEV05500
C     TO MIMIMIZE THE EFFECT OF THE ONE-SIDED ACCEPTANCE TEST FOR  LEV05510
C     EIGENVALUES IN THE BISEC SUBROUTINE, RECOMPUTE THE GIVEN     LEV05520
C     EIGENVALUE AT THE SPECIFIED KMAXU                 LEV05530
C                                                       LEV05540
      CALL LBISEC(ALPHA,BETA,EPSM,EVAL,EVALN,LB,UB,TTOL,KMAXU,NEVT)  LEV05550
C                                                       LEV05560
C-----------------------------------------------------------------LEV05570
C                                                       LEV05580
C     CHECK WHETHER OR NOT GIVEN T-MATRIX HAS AN EIGENVALUE IN THE  LEV05590
C     SPECIFIED INTERVAL AND IF SO WHAT ITS T-MULTIPLICITY IS.     LEV05600
C                                                       LEV05610
      IF(NEVT.EQ.1) GO TO 590                           LEV05620
      IF(NEVT.NE.0) GO TO 570                           LEV05630
      ILBIS = 1                                         LEV05640
      WRITE(6,560) EVAL,KMAXU                           LEV05650
  560 FORMAT(/' PROBLEM ENCOUNTERED IN RECOMPUTATION OF USER-SUPPLIED EILEV05660
     1GENVALUE',E20.12/' THE SIZE T-MATRIX SPECIFIED',I6,' DOES NOT     LEV05670
     1HAVE AN EIGENVALUE IN THE INTERVAL SPECIFIED'/' THEREFORE NO EIGENLEV05680
     1VECTOR WILL BE COMPUTED FOR THIS PARTICULAR EIGENVALUE'/)  LEV05690
      GO TO 610                                         LEV05700
C                                                       LEV05710
  570 IF(NEVT.GT.1)  WRITE(6,580) EVAL,KMAXU            LEV05720
  580 FORMAT(/' PROBLEM ENCOUNTERED IN RECOMPUTATION OF USER-SUPPLIED  LEV05730
     1EIGENVALUE',E20.12/' FOR THE SIZE T-MATRIX SPECIFIED =',I6,' THE  LEV05740
     1GIVEN EIGENVALUE IS T-MULTIPLE IN THE INTERVAL SPECIFIED'/' SOMETHLEV05750
     1ING IS WRONG, THEREFORE NO EIGENVECTOR WILL BE COMPUTED FOR THIS ELEV05760
     1IGENVALUE'/)                                      LEV05770
C                                                       LEV05780
      MP(J) = MPMIN                                     LEV05790
      MA(J) = -2*KMAX                                   LEV05800
      GO TO 710                                         LEV05810
C                                                       LEV05820
  590 CONTINUE                                          LEV05830
      ILBIS = 0                                         LEV05840
C                                                       LEV05850
      EVNEW(J) = EVALN                                  LEV05860
      EVAL = EVALN                                      LEV05870
      MTOL = MTOL+KMAXU                                 LEV05880
C                                                       LEV05890
C     IS THERE ROOM IN TVEC ARRAY FOR THE NEXT T-EIGENVECTOR?  LEV05900
C     IF NOT, SKIP TO RITZ VECTOR COMPUTATIONS.         LEV05910
      IF (MTOL.GT.MDIMTV) GO TO 720                     LEV05920
C                                                       LEV05930
      IT = 3                                            LEV05940
      KINT = MTOL - KMAXU +1                            LEV05950
```

```
C                                                                    LEV05960
C     RECORD THE BEGINNING AND END OF THE T-EIGENVECTOR BEING COMPUTED LEV05970
      MINT(J) = KINT                                                 LEV05980
      MFIN(J) = MTOL                                                 LEV05990
C                                                                    LEV06000
C----------------------------------------------------------------------LEV06010
C     SUBROUTINE INVERM DOES INVERSE ITERATION, I.E. SOLVES          LEV06020
C     (T(1,KMAXU) - EVAL)*U = RHS  FOR EACH EIGENVALUE TO OBTAIN THE LEV06030
C     DESIRED T-EIGENVECTOR.                                         LEV06040
C                                                                    LEV06050
      IF(IWRITE.EQ.1)  WRITE(6,600) J                               LEV06060
  600 FORMAT(/I6,'TH EIGENVALUE')                                    LEV06070
C                                                                    LEV06080
      CALL INVERM(ALPHA,BETA,V1,TVEC(KINT),EVAL,ERROR,TERROR,EPSM,   LEV06090
     1 G,KMAXU,IT,IWRITE)                                           LEV06100
C                                                                    LEV06110
C----------------------------------------------------------------------LEV06120
C                                                                    LEV06130
      TERR(J) = TERROR                                               LEV06140
      TLAST(J) = ERROR                                               LEV06150
      KMAXU1 = KMAXU + 1                                             LEV06160
      TBETA(J) = BETA(KMAXU1)*ERROR                                  LEV06170
C                                                                    LEV06180
C     AFTER EACH OF THE T-EIGENVECTORS IS COMPUTED, THE             LEV06190
C     SIZE OF THE ERROR ESTIMATE, ERROR IS CHECKED.                 LEV06200
C     IF THIS ESTIMATE IS NOT AS SMALL AS DESIRED AND               LEV06210
C     |MA(J)| < ML(J), PROGRAM ATTEMPTS TO INCREASE THE SIZE OF |MA(J)| LEV06220
C     AND REPEAT THE T-EIGENVECTOR COMPUTATIONS.                    LEV06230
C                                                                    LEV06240
      IF(ERROR.LT.ERTOL.OR.TFLAG.EQ.1)  GO TO 700                   LEV06250
C                                                                    LEV06260
      IF(ERROR.GE.ERRMIN) GO TO 610                                 LEV06270
C     LAST COMPONENT IS LESS THAN MINIMAL TO DATE                   LEV06280
      ERRMIN = ERROR                                                LEV06290
      MABEST = MA(J)                                                LEV06300
  610 CONTINUE                                                       LEV06310
C                                                                    LEV06320
      IF(MA(J).GT.0)  ITEST = MA(J) + IDELTA(J)                     LEV06330
      IF(MA(J).LT.0)  ITEST = -(IABS(MA(J)) + IDELTA(J))           LEV06340
      IF(IABS(ITEST).LE.ML(J).AND.ICOUNT.LE.10) GO TO 630          LEV06350
C     NEW MA(J) IS GREATER THAN MAXIMUM ALLOWED.                    LEV06360
      IF(ERCONT.EQ.0.OR.MABEST.EQ.MPMIN) GO TO 650                 LEV06370
      TFLAG = 1                                                     LEV06380
      MA(J) = MABEST                                                LEV06390
      IF(ILBIS.EQ.0)  MTOL = MTOL - KMAXU                          LEV06400
      WRITE(6,620) MA(J)                                           LEV06410
  620 FORMAT(' 10 ORDERS WERE CONSIDERED.  NONE SATISFIED THE ERROR TESTLEV06420
     1'/' THEREFORE USE THE BEST ORDER OBTAINED FOR THE EIGENVECTORS' LEV06430
     1,I6)                                                         LEV06440
      GO TO 530                                                     LEV06450
C                                                                    LEV06460
  630 MA(J) = ITEST                                                 LEV06470
C                                                                    LEV06480
      MT = IABS(MA(J))                                              LEV06490
      IF(IWRITE.EQ.1) WRITE(6,640)  MT                             LEV06500
```

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

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

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

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

```
C                                                             LEV08710
      IF(IABS(MA(J)).LT.IVEC.OR.MP(J).EQ.MPMIN) GO TO 1000    LEV08720
      II = IVEC + MINT(J) - 1                                 LEV08730
      TEMP = TVEC(II)                                         LEV08740
C     II IS THE (IVEC)TH COMPONENT OF THE T-EIGENVECTOR CONTAINED  LEV08750
C     IN TVEC(MINT(J)).                                       LEV08760
C                                                             LEV08770
      DO 990 K = 1,N                                          LEV08780
      LL = LL + 1                                             LEV08790
  990 RITVEC(LL) = TEMP*V2(K) + RITVEC(LL)                    LEV08800
C                                                             LEV08810
 1000 CONTINUE                                                LEV08820
C                                                             LEV08830
      IVEC = IVEC + 1                                         LEV08840
      IF (IVEC.LE.KMAXU) GO TO 930                            LEV08850
C                                                             LEV08860
C                                                             LEV08870
C     RITZVECTOR GENERATION IS COMPLETE. NORMALIZE EACH RITZVECTOR.  LEV08880
C     NOTE THAT IF CERTAIN RITZ VECTORS WERE NOT COMPUTED THEN THAT  LEV08890
C     PORTION OF THE RITVEC ARRAY WAS NOT UTILIZED.           LEV08900
C                                                             LEV08910
      LFIN = 0                                                LEV08920
      DO 1050 J = 1,NGOODC                                    LEV08930
C                                                             LEV08940
      KK = LFIN                                               LEV08950
      LFIN = LFIN + N                                         LEV08960
      IF(MP(J).EQ.MPMIN) GO TO 1050                           LEV08970
C                                                             LEV08980
      DO 1010 K = 1,N                                         LEV08990
      KK = KK + 1                                             LEV09000
 1010 V2(K) = RITVEC(KK)                                      LEV09010
C                                                             LEV09020
      SUM = FINPRO(N,V2(1),1,V2(1),1)                         LEV09030
      SUM = DSQRT(SUM)                                        LEV09040
      RNORM(J) = SUM                                          LEV09050
      TEMP = DABS(ONE-SUM)                                    LEV09060
      SUM = ONE/SUM                                           LEV09070
C                                                             LEV09080
      KK = LFIN - N                                           LEV09090
      DO 1020 K = 1,N                                         LEV09100
      KK = KK + 1                                             LEV09110
      V2(K) = SUM*V2(K)                                       LEV09120
 1020 RITVEC(KK) = V2(K)                                      LEV09130
C                                                             LEV09140
      IF (IWRITE.NE.0) WRITE(6,1030) J,GOODEV(J)              LEV09150
 1030 FORMAT(/I5,' TH EIGENVALUE CONSIDERED = ',E20.12/)      LEV09160
C                                                             LEV09170
      IF (IWRITE.NE.0) WRITE(6,1040) TERR(J),TBETA(J),TEMP    LEV09180
 1040 FORMAT(' NORM OF ERROR IN T-EIGENVECTOR = ',E14.3/      LEV09190
     1 ' BETA(MA(J)+1)*U(MA(J)) = ',E14.3/                    LEV09200
     1 ' ABS(NORM(RITVEC) - 1.0)  = ',E14.3/)                 LEV09210
C                                                             LEV09220
      LINT = LFIN - N + 1                                     LEV09230
      EVAL = EVNEW(J)                                         LEV09240
C                                                             LEV09250
```

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

```
      1'MP(J)')                                                 LEV09810
         WRITE(9,1130) GOODEV(J),MA(J),AMINGP(J),TBETA(J),MP(J) LEV09820
 1130 FORMAT(E15.8,I8,2E14.4,I8)                                LEV09830
C                                                              LEV09840
         WRITE(13,1140)                                         LEV09850
 1140 FORMAT(/' RITZ VECTORS WERE NOT COMPUTED FOR THE FOLLOWING EIGENVALEV09860
      1LUES'/' BECAUSE THEY HAD NOT CONVERGED'/)                LEV09870
C                                                              LEV09880
         WRITE(13,1150)                                         LEV09890
 1150 FORMAT(6X,'GOODEV(J)',3X,'MA(J)',3X,'M1(J)',3X,'M2(J)',3X,'MP(J)' LEV09900
      1/)                                                       LEV09910
         WRITE(13,1160) GOODEV(J),MA(J),M1(J),M2(J),MP(J)       LEV09920
 1160 FORMAT(E15.8,4I8)                                         LEV09930
C                                                              LEV09940
 1170 CONTINUE                                                  LEV09950
 1180 CONTINUE                                                  LEV09960
C                                                              LEV09970
         WRITE(9,1190)                                          LEV09980
 1190 FORMAT(/' ABOVE ARE ERROR ESTIMATES FOR THE A AND T EIGENVECTORS'/LEV09990
      1 ' ASSOCIATED WITH THE GOODEV LISTED IN COLUMN 1'/       LEV10000
      1 ' AERROR = NORM(A*X - EV*X)   TERROR = NORM(T*Y - EV*Y) '/ LEV10010
      1 ' WHERE T = T(1,MA(J))     X = RITZ VECTOR = V*Y  V = SUCCESSIVE'/LEV10020
      1 ' LANCZOS VECTORS. AMINGAP = GAP TO NEAREST A-EIGENVALUE'//) LEV10030
C                                                              LEV10040
         WRITE(13,1200)                                         LEV10050
 1200 FORMAT(/' ABOVE ARE ERROR ESTIMATES ASSOCIATED WITH THE GOODEV'/ LEV10060
      1 ' RITZNORM = NORM(COMPUTED RITZ VECTOR)'/               LEV10070
      1 ' TBETA(J) = BETA(MA(J)+1)*Y(MA(J)),   T*Y = EVAL*Y'/   LEV10080
      1 ' TLAST(J) = Y(MA(J))'/                                 LEV10090
      1 ' AMINGAP = GAP TO NEAREST A-EIGENVALUE'/)              LEV10100
C                                                              LEV10110
C     NUMBER OF RITZ VECTORS COMPUTED                          LEV10120
      NCOMPU = NGOODC - MREJEC                                  LEV10130
      WRITE(12,1210) N,NCOMPU,NGOODC,MATNO                      LEV10140
 1210 FORMAT(3I6,I12,' SIZE A, NO.RITZVECS, NO.EVALUES,MATNO') LEV10150
C                                                              LEV10160
      LFIN = 0                                                 LEV10170
      DO 1270 J = 1,NGOODC                                     LEV10180
      LINT = LFIN + 1                                          LEV10190
      LFIN = LFIN + N                                          LEV10200
C                                                              LEV10210
      IF(MP(J).EQ.MPMIN) GO TO 1250                            LEV10220
C     RITZ VECTOR WAS COMPUTED                                 LEV10230
      WRITE(12,1220) J, GOODEV(J), MP(J)                       LEV10240
 1220 FORMAT(I6,4X,E20.12,I6,' J, EIGENVAL, MP(J)')            LEV10250
C                                                              LEV10260
      WRITE(12,1230) ERR(J),ERRDGP(J)                          LEV10270
 1230 FORMAT(2E15.5,' = NORM(A*Z-EVAL*Z) AND  NORM(A*Z-EVAL*Z)/MINGAP') LEV10280
C                                                              LEV10290
      WRITE(12,1240) (RITVEC(LL), LL=LINT,LFIN)                LEV10300
 1240 FORMAT(4E20.12)                                          LEV10310
      GO TO 1270                                               LEV10320
C     NO RITZ VECTOR WAS COMPUTED FOR THIS EIGENVALUE          LEV10330
 1250 WRITE(12,1260) J,GOODEV(J),MP(J)                         LEV10340
 1260 FORMAT(I6,4X,E20.12,I6,' J,EIGVALUE,NO RITZ VECTOR COMPUTED') LEV10350
```

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

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

2.5   LEMULT: LANCZS and Sample Matrix-Vector Multiply Subroutines

```
C-----LEMULT----------------------------------------------------------LEM00010
C  Authors:   Jane Cullum and Ralph A. Willoughby (Deceased)          LEM00020
C             Los Alamos National Laboratory                          LEM00030
C             Los Alamos, New Mexico 87544                            LEM00040
C                                                                     LEM00050
C             E-mail:  cullumj@lanl.gov                               LEM00060
C                                                                     LEM00070
C  These codes are copyrighted by the authors.  These codes           LEM00080
C  and modifications of them or portions of them are NOT to be        LEM00090
C  incorporated into any commercial codes or used for any other       LEM00100
C  commercial purposes such as consulting for other companies,        LEM00110
C  without legal agreements with the authors of these Codes.          LEM00120
C  If these Codes or portions of them are used in other scientific or LEM00130
C  engineering research works the names of the authors of these codes LEM00140
C  and appropriate references to their written work are to be         LEM00150
C  incorporated in the derivative works.                              LEM00160
C                                                                     LEM00170
C  This header is not to be removed from these codes.                 LEM00180
C                                                                     LEM00190
C          REFERENCE: Cullum and Willoughby, Chapters 1,2,3,4         LEM00191
C          Lanczos Algorithms for Large Symmetric Eigenvalue ComputationsLEM00192
C          VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in LEM00193
C          Applied Mathematics, 2002. SIAM Publications,              LEM00194
C          Philadelphia, PA. USA                                      LEM00195
C                                                                     LEM00196
C                                                                     LEM00200
C     CONTAINS SUBROUTINES  LANCZS, USPEC, AND CMATV                  LEM00210
C     TO BE USED WITH THE REAL SYMMETRIC VERSION OF THE LANCZOS       LEM00220
C     EIGENVALUE/EIGENVECTOR PROCEDURES.                              LEM00230
C     ALSO CONTAINS SUBROUTINES FOR POISSON TEST MATRICES THAT ALLOW  LEM00240
C     COMPUTATION OF TRUE ERRORS IN COMPUTED EIGENVALUES AND          LEM00250
C     IN CORRESPONDING EIGENVECTORS.                                  LEM00260
C                                                                     LEM00270
C     NONPORTABLE CONSTRUCTIONS:                                      LEM00280
C     1.  THE ENTRY MECHANISM USED TO PASS THE STORAGE               LEM00290
C         LOCATIONS OF THE USER-SPECIFIED MATRIX FROM THE            LEM00300
C         SUBROUTINE USPEC TO THE MATRIX-VECTOR SUBROUTINE           LEM00310
C         CMATV.                                                     LEM00320
C     2.  IN THE SAMPLE USPEC AND CMATV FOR DIAGONAL TEST MATRICES:  LEM00330
C         FREE FORMAT (8,*) AND THE FORMAT (20A4).                   LEM00340
C     3.  IN THE POISSON SUBROUTINES PROVIDED, THE DATA MACHEP       LEM00350
C         DEFINITION AND MANY OF THE INDICES FOR ARRAYS ARE NOT      LEM00360
C         IN A PORTABLE CONSTRUCTION.  THESE PROGRAMS SHOULD BE      LEM00370
C         REMOVED FROM THE LEMULT FILE IF THE USER IS NOT USING THEM. LEM00380
C                                                                     LEM00390
C-----LANCZS-COMPUTE THE LANCZOS TRIDIAGONAL MATRICES-----------------LEM00400
C                                                                     LEM00410
      SUBROUTINE LANCZS(MATVEC,ALPHA,BETA,V1,V2,G,KMAX,MOLD1,N,IIX)   LEM00420
C                                                                     LEM00430
C--------------------------------------------------------------------LEM00440
```

```
      DOUBLE PRECISION  ALPHA(1),BETA(1),V1(1),V2(1),SUM,TEMP,ONE,ZERO   LEM00450
      REAL  G(1)                                                        LEM00460
      DOUBLE PRECISION FINPRO, DSQRT                                    LEM00470
      EXTERNAL MATVEC                                                   LEM00480
C----------------------------------------------------------------------LEM00490
C                                                                       LEM00500
      ZERO = 0.D0                                                       LEM00510
      ONE = 1.D0                                                        LEM00520
C                                                                       LEM00530
      IF(MOLD1.GT.1)GO TO 30                                            LEM00540
C                                                                       LEM00550
C   ALPHA/BETA GENERATION STARTS AT I = 1                               LEM00560
C   MOLD1 = 1 SET V1 = 0. AND V2 = RANDOM UNIT VECTOR                   LEM00570
      BETA(1) = ZERO                                                    LEM00580
      IIL=IIX                                                           LEM00590
C                                                                       LEM00600
C----------------------------------------------------------------------LEM00610
      CALL GENRAN(IIL,G,N)                                              LEM00620
C----------------------------------------------------------------------LEM00630
C                                                                       LEM00640
      DO 10 I = 1,N                                                     LEM00650
   10 V2(I) = G(I)                                                      LEM00660
C                                                                       LEM00670
C----------------------------------------------------------------------LEM00680
      SUM = FINPRO(N,V2(1),1,V2(1),1)                                   LEM00690
C----------------------------------------------------------------------LEM00700
C                                                                       LEM00710
      SUM = ONE/DSQRT(SUM)                                              LEM00720
      DO 20 I = 1,N                                                     LEM00730
      V1(I) = ZERO                                                      LEM00740
   20 V2(I) = V2(I)*SUM                                                 LEM00750
C                                                                       LEM00760
C   ALPHA BETA GENERATION LOOP                                          LEM00770
   30 CONTINUE                                                          LEM00780
C                                                                       LEM00790
      DO 60 I=MOLD1,KMAX                                                LEM00800
      SUM = BETA(I)                                                     LEM00810
C   MATVEC(V2,V1,SUM) CALCULATES  V1 = A*V2 - SUM*V1                    LEM00820
C                                                                       LEM00830
C----------------------------------------------------------------------LEM00840
      CALL MATVEC(V2,V1,SUM)                                            LEM00850
C----------------------------------------------------------------------LEM00860
C                                                                       LEM00870
C----------------------------------------------------------------------LEM00880
      SUM = FINPRO(N,V1(1),1,V2(1),1)                                   LEM00890
C----------------------------------------------------------------------LEM00900
C                                                                       LEM00910
      ALPHA(I) = SUM                                                    LEM00920
      DO 40 J=1,N                                                       LEM00930
   40 V1(J) = V1(J)-SUM*V2(J)                                           LEM00940
C                                                                       LEM00950
C----------------------------------------------------------------------LEM00960
      SUM = FINPRO(N,V1(1),1,V1(1),1)                                   LEM00970
C----------------------------------------------------------------------LEM00980
C                                                                       LEM00990
```

```
      IN = I+1                                               LEM01000
      BETA(IN) = DSQRT(SUM)                                  LEM01010
      SUM = ONE/BETA(IN)                                     LEM01020
      DO 50 J=1,N                                            LEM01030
      TEMP = SUM*V1(J)                                       LEM01040
      V1(J) = V2(J)                                          LEM01050
   50 V2(J) = TEMP                                           LEM01060
   60 CONTINUE                                               LEM01070
C                                                            LEM01080
C     END ALPHA, BETA GENERATION LOOP                        LEM01090
C                                                            LEM01100
      RETURN                                                 LEM01110
C-----END OF LANCZS-----------------------------------------LEM01120
      END                                                    LEM01130
C                                                            LEM01140
C-----USPEC (GENERAL SYMMETRIC SPARSE MATRICES)-------------LEM01150
C                                                            LEM01160
C     SUBROUTINE USPEC(N,MATNO)                              LEM01170
      SUBROUTINE GUSPEC(N,MATNO)                             LEM01180
C                                                            LEM01190
C----------------------------------------------------------LEM01200
      DOUBLE PRECISION  A(10000),AD(5010)                    LEM01210
      INTEGER  IROW(10000),ICOL(5010)                        LEM01220
C----------------------------------------------------------LEM01230
C   USPEC DIMENSIONS AND INITIALIZES THE ARRAYS NEEDED TO DEFINE   LEM01240
C   THE USER-SPECIFIED MATRIX AND THEN PASSES THE STORAGE LOCATIONS LEM01250
C   OF THESE ARRAYS TO THE MULTIPLY SUBROUTINE CMATV.      LEM01260
C                                                            LEM01270
C   MATRIX IS STORED IN FOLLOWING SPARSE MATRIX FORMAT:     LEM01280
C   N = ORDER OF A-MATRIX,                                  LEM01290
C   NZS = NUMBER OF NONZERO SUBDIAGONAL ENTRIES,            LEM01300
C   NZL = INDEX OF LAST COLUMN CONTAINING NONZERO SUBDIAGONAL ENTRIES, LEM01310
C   ICOL(J), J=1,NZL IS THE NUMBER OF NONZERO SUBDIAGONAL ELEMENTS LEM01320
C           IN COLUMN J.                                    LEM01330
C   IROW(K), K = 1,NZS IS THE CORRESPONDING ROW INDEX FOR A(K). LEM01340
C   AD(I), I=1,N CONTAINS DIAGONAL ENTRIES (INCLUDING ANY 0 LEM01350
C        DIAGONAL ENTRIES).                                 LEM01360
C   A(K), K=1,NZS CONTAINS NONZERO SUBDIAGONAL ENTRIES, BY COLUMN LEM01370
C   FOR J > NZL THERE ARE NO NONZERO SUBDIAGONAL ELEMENTS IN COLUMN J. LEM01380
C   ICOL(J) = 0 IS ALLOWED                                  LEM01390
C                                                            LEM01400
C----------------------------------------------------------LEM01410
C     ARRAYS THAT DEFINE THE MATRIX ARE READ IN FROM FILE 8 LEM01420
C                                                            LEM01430
      READ(8,10) NZS,NOLD,NZL,MATOLD                         LEM01440
   10 FORMAT(I10,2I6,I8)                                     LEM01450
C                                                            LEM01460
      WRITE(6,20) NZS,NOLD,NZL,MATOLD                        LEM01470
   20 FORMAT(I10,2I6,I8,' = NZS,NOLD,NZL,MATOLD'/)           LEM01480
C                                                            LEM01490
C     TEST OF PARAMETER CORRECTNESS                          LEM01500
      ITEMP = (NOLD-N)**2 + (MATNO-MATOLD)**2                LEM01510
C                                                            LEM01520
      IF(ITEMP.EQ.0) GO TO 40                                LEM01530
C                                                            LEM01540
```

```
      WRITE(6,30)                                            LEM01550
   30 FORMAT(' PROGRAM TERMINATES BECAUSE EITHER ORDERS OF OR LABELS FORLEM01560
     1 MATRIX DISAGREE')                                     LEM01570
      GO TO 70                                               LEM01580
C                                                            LEM01590
   40 CONTINUE                                               LEM01600
C                                                            LEM01610
C     NUMBER OF NONZERO SUBDIAGONAL ENTRIES IN EACH COLUMN IS READ   LEM01620
C     THEN THE CORRESPONDING ROW INDEX FOR EACH SUCH ENTRY IS READ   LEM01630
      READ(8,50) (ICOL(K), K=1,NZL)                          LEM01640
      READ(8,50) (IROW(K), K=1,NZS)                          LEM01650
   50 FORMAT(13I6)                                           LEM01660
C                                                            LEM01670
C     DIAGONAL IS READ FIRST, THEN NONZERO BELOW DIAGONAL ENTRIES    LEM01680
      READ(8,60) (AD(K), K=1,N)                              LEM01690
      READ(8,60) (A(K), K=1,NZS)                             LEM01700
   60 FORMAT(4E19.10)                                        LEM01710
C                                                            LEM01720
C------------------------------------------------------------LEM01730
C     PASS STORAGE LOCATIONS OF ARRAYS THAT DEFINE THE MATRIX TO     LEM01740
C     THE MATRIX-VECTOR MULTIPLY SUBROUTINE CMATV            LEM01750
C                                                            LEM01760
      CALL CMATVE(A,AD,ICOL,IROW,N,NZL)                      LEM01770
C                                                            LEM01780
C------------------------------------------------------------LEM01790
C                                                            LEM01800
      RETURN                                                 LEM01810
   70 STOP                                                   LEM01820
C-----END OF USPEC-------------------------------------------LEM01830
      END                                                    LEM01840
C                                                            LEM01850
C-----MATRIX-VECTOR MULTIPLY FOR REAL SPARSE SYMMETRIC MATRICES------LEM01860
C                                                            LEM01870
C     SUBROUTINE CMATV(W,U,SUM)                              LEM01880
      SUBROUTINE GCMATV(W,U,SUM)                             LEM01890
C                                                            LEM01900
C------------------------------------------------------------LEM01910
      DOUBLE PRECISION   U(1),W(1),A(1),AD(1),SUM            LEM01920
      INTEGER  IROW(1),ICOL(1)                               LEM01930
C------------------------------------------------------------LEM01940
C     SPARSE MATRIX-VECTOR MULTIPLY FOR LANCZS  U = A*W - SUM*U      LEM01950
C     SEE USPEC SUBROUTINE FOR DESCRIPTION OF THE ARRAYS THAT DEFINE LEM01960
C     THE MATRIX                                             LEM01970
C------------------------------------------------------------LEM01980
C                                                            LEM01990
      GO TO 3                                                LEM02000
C     STORAGE LOCATIONS OF ARRAYS ARE PASSED TO CMATV FROM USPEC     LEM02010
      ENTRY CMATVE(A,AD,ICOL,IROW,N,NZL)                     LEM02020
      GO TO 4                                                LEM02030
C------------------------------------------------------------LEM02040
C     COMPUTE THE DIAGONAL TERMS                             LEM02050
    3 DO 10 I = 1,N                                          LEM02060
   10 U(I) = AD(I)*W(I)-SUM*U(I)                             LEM02070
C                                                            LEM02080
C     COMPUTE BY COLUMN                                      LEM02090
```

```
      LLAST = 0                                                 LEM02100
      DO 30 J = 1,NZL                                           LEM02110
C                                                               LEM02120
      IF (ICOL(J).EQ.0) GO TO 30                                LEM02130
      LFIRST = LLAST + 1                                        LEM02140
      LLAST = LLAST + ICOL(J)                                   LEM02150
C                                                               LEM02160
      DO 20 L = LFIRST,LLAST                                    LEM02170
      I = IROW(L)                                               LEM02180
C                                                               LEM02190
      U(I) = U(I) + A(L)*W(J)                                   LEM02200
      U(J) = U(J) + A(L)*W(I)                                   LEM02210
C                                                               LEM02220
   20 CONTINUE                                                  LEM02230
C                                                               LEM02240
   30 CONTINUE                                                  LEM02250
C                                                               LEM02260
    4 RETURN                                                    LEM02270
C                                                               LEM02280
C-----END OF CMATV--------------------------------------------LEM02290
      END                                                       LEM02300
C                                                               LEM02310
C-----MATRIX-VECTOR MULTIPLY FOR DIAGONAL TEST MATRICES---------------LEM02320
C                                                               LEM02330
C     SUBROUTINE CMATV(W,U,SUM)                                 LEM02340
      SUBROUTINE DCMATV(W,U,SUM)                                LEM02350
C                                                               LEM02360
C     CMATV COMPUTES  U = (DIAGONAL MATRIX) * W - SUM * U       LEM02370
C----------------------------------------------------------------LEM02380
      DOUBLE PRECISION  W(1),U(1),SUM                           LEM02390
      DOUBLE PRECISION  D(1)                                    LEM02400
C----------------------------------------------------------------LEM02410
      GO TO 3                                                   LEM02420
      ENTRY MVDIAE(D,N)                                         LEM02430
      GO TO 4                                                   LEM02440
C----------------------------------------------------------------LEM02450
C                                                               LEM02460
    3 DO 10 I=1,N                                               LEM02470
   10 U(I)= D(I)*W(I) - SUM*U(I)                                LEM02480
    4 RETURN                                                    LEM02490
C                                                               LEM02500
C-----END OF DIAGONAL TEST MATRIX  MULTIPLY--------------------------LEM02510
      END                                                       LEM02520
C                                                               LEM02530
C                                                               LEM02540
C-----START OF USPEC FOR DIAGONAL TEST MATRIX------------------------LEM02550
C                                                               LEM02560
C     SUBROUTINE USPEC(N,MATNO)                                 LEM02570
      SUBROUTINE DUSPEC(N,MATNO)                                LEM02580
C                                                               LEM02590
C----------------------------------------------------------------LEM02600
      DOUBLE PRECISION  D(1000), SHIFT, SPACE                   LEM02610
      DOUBLE PRECISION  DABS, DFLOAT                            LEM02620
      REAL  EXPLAN(20)                                          LEM02630
C----------------------------------------------------------------LEM02640
```

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

```
C-------------------------------------------------------------------LEM03200
      DOUBLE PRECISION  C0,C1,C2,HALF,ONE                            LEM03210
      REAL  EXPLAN(20)                                               LEM03220
C-----------------------------------------------------------------   LEM03230
      HALF = 0.5D0                                                   LEM03240
      ONE  = 1.0D0                                                   LEM03250
C                                                                    LEM03260
C     READ USER-SPECIFIED PARAMETERS FROM INPUT FILE 8 (FREE FORMAT) LEM03270
C                                                                    LEM03280
      READ(8,10) EXPLAN                                              LEM03290
      WRITE(6,10) EXPLAN                                             LEM03300
   10 FORMAT(20A4)                                                   LEM03310
C                                                                    LEM03320
      READ(8,10) EXPLAN                                              LEM03330
      READ(8,*) KX,KY,C0                                             LEM03340
      N = KX*KY                                                      LEM03350
      C1 = HALF-C0                                                   LEM03360
      C2 = ONE                                                       LEM03370
C                                                                    LEM03380
      WRITE(6,20) N,KX,KY,C2,C0,C1                                   LEM03390
   20 FORMAT(/5X,'N',4X,'KX',4X,'KY',7X,'DIAGONAL',3X,'X-CODIAGONAL', LEM03400
     1 3X,'Y-CODIAGONAL'/3I6,3E15.8/)                                LEM03410
C                                                                    LEM03420
C-------------------------------------------------------------------LEM03430
      CALL PMATVE(C0,C1,C2,KX,KY)                                    LEM03440
      CALL EXEVE(C0,C1,C2,KX,KY)                                     LEM03450
      CALL EXERRP(C0,C1,C2,KX,KY)                                    LEM03460
C     CALL EXVECP(C0,C1,C2,KX,KY)                                    LEM03470
C-------------------------------------------------------------------LEM03480
C                                                                    LEM03490
      RETURN                                                         LEM03500
C-----END OF USPEC--------------------------------------------------LEM03510
      END                                                           LEM03520
C                                                                    LEM03530
C-----START OF CMATV------------------------------------------------LEM03540
C     CALCULATE U = A*W - SUM*U FOR REAL POISSON MATRICES            LEM03550
C                                                                    LEM03560
      SUBROUTINE CMATV(W,U,SUM)                                      LEM03570
C     SUBROUTINE PMATV(W,U,SUM)                                      LEM03580
C                                                                    LEM03590
C-------------------------------------------------------------------LEM03600
      DOUBLE PRECISION  U(1),W(1)                                    LEM03610
      DOUBLE PRECISION  C0,C1,C2,CC0,CC1,SUM                         LEM03620
C-------------------------------------------------------------------LEM03630
      GO TO 3                                                        LEM03640
      ENTRY  PMATVE(CC0,CC1,C2,KX,KY)                                LEM03650
C-------------------------------------------------------------------LEM03660
C                                                                    LEM03670
      C0 = -CC0                                                      LEM03680
      C1 = -CC1                                                      LEM03690
      GO TO 4                                                        LEM03700
C                                                                    LEM03710
    3 N = KX*KY                                                      LEM03720
      KX1 = KX-1                                                     LEM03730
      KY1 = KY-1                                                     LEM03740
```

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

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

```
      MP(NEXACT) = IDEX                                        LEM04850
C     MP(K) = MULTIPLICITY OF KTH EIGENVALUE CLUSTER FOR A     LEM04860
      IDEX = 1                                                 LEM04870
      I = J                                                    LEM04880
      IF (I.GT.N) GO TO 90                                     LEM04890
      GO TO 70                                                 LEM04900
   90 CONTINUE                                                 LEM04910
C                                                              LEM04920
C     MULTIPLICITIES HAVE BEEN DETERMINED                      LEM04930
C     NEXACT = NUMBER OF DISTINCT A-EIGENVALUES                LEM04940
C                                                              LEM04950
      WRITE(9,100)NEXACT                                       LEM04960
      WRITE(6,100)NEXACT                                       LEM04970
  100 FORMAT(I6,' = NUMBER OF TRUE A-EIGENVALUES WHICH ARE DISTINCT'/)  LEM04980
C                                                              LEM04990
C     MINGAP CALCULATION FOR DISTINCT A-EIGENVALUES            LEM05000
      NM1 = NEXACT - 1                                         LEM05010
      G(NEXACT) = U(NM1)-U(NEXACT)                             LEM05020
      G(1) = U(2)-U(1)                                         LEM05030
C                                                              LEM05040
      DO 110 J = 2,NM1                                         LEM05050
      T0 = U(J)-U(J-1)                                         LEM05060
      T1 = U(J+1)-U(J)                                         LEM05070
      G(J) = T1                                                LEM05080
      IF (T0.LT.T1) G(J) = -T0                                 LEM05090
  110 CONTINUE                                                 LEM05100
C                                                              LEM05110
C     NEXACT DISTINCT A-EIGENVALUES ARE IN U IN ASCENDING ORDER LEM05120
C     MP = MULTIPLICITIES OF THE DISTINCT EIGENVALUES OF A     LEM05130
C     G = TRUE MINIMUM GAP IN A FOR EACH OF THESE EIGENVALUES  LEM05140
C     G < 0 INDICATES THE LEFT-HAND GAP WAS MINIMAL.           LEM05150
C     OUTPUT MULTIPLICITIES, DISTINCT EVS, AND MINGAPS TO FILE 9 LEM05160
C                                                              LEM05170
      WRITE(9,120)                                             LEM05180
  120 FORMAT(5X,'I',1X,'AMULT',5X,'TRUE A-EIGENVALUE(I)',      LEM05190
     1 3X,'A-MINGAP(I)')                                       LEM05200
C                                                              LEM05210
      WRITE(9,130)(J,MP(J),U(J),G(J), J=1,NEXACT)              LEM05220
  130 FORMAT(2I6,E25.16,E14.3)                                 LEM05230
C                                                              LEM05240
      WRITE(9,140)                                             LEM05250
  140 FORMAT(' NEXACT DISTINCT A-EIGENVALUES ARE IN ASCENDING ORDER'/ LEM05260
     1 ' AMULT = MULTIPLICITIES OF THE DISTINCT EIGENVALUES OF A.'/ LEM05270
     2 ' A-MINGAP(I) = TRUE MINIMUM GAP IN A FOR EACH EIGENVALUE.'/ LEM05280
     3 ' A-MINGAP(I) LT 0 INDICATES THE LEFT-HAND GAP WAS MINIMAL.'//) LEM05290
C                                                              LEM05300
C     WE ORDER U VECTOR BY INCREASING SIZE OF THE GAPS         LEM05310
C                                                              LEM05320
      DO 150 K = 1,NEXACT                                      LEM05330
  150 MP(K) = K                                                LEM05340
C                                                              LEM05350
      DO 170 K = 2,NEXACT                                      LEM05360
      KM1 = K-1                                                LEM05370
C                                                              LEM05380
      DO 160 L = 1,KM1                                         LEM05390
```

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

```
C-----------------------------------------------------------------LEM05950
      DATA MACHEP/Z3410000000000000/                              LEM05960
      EPSM = 2.0D0*MACHEP                                         LEM05970
C-----------------------------------------------------------------LEM05980
C     ON ENTRY V CONTAINS NG GOOD EIGENVALUES OF T(1,MEV)          LEM05990
C     MP CONTAINS THE MULTIPLICITIES OF THESE EIGENVALUES.         LEM06000
C     U(I) = GAP TO NEAREST DISTINCT TMEV I=1,NG                   LEM06010
C     U < 0. MEANS GAP IS DUE TO SPURIOUS EV                       LEM06020
C                                                                  LEM06030
C     ON EXIT G(MEV+I) = ERROR FOR V(I) < 0. IF MULT EV > 1        LEM06040
C     K = MP(I) MEANS |V(I) - U(K)| = MIN                          LEM06050
C     MP < 0 MEANS MORE THAN ONE I USES SAME K                     LEM06060
C                                                                  LEM06070
C     T0 = C2 - 2*C1*COS(PIL*J)                                    LEM06080
C     U(KP) = T0 - 2*C0*COS(PIK*I)                                 LEM06090
C     KP = (J-1)*KX + I                                            LEM06100
C     C2 = ONE                                                     LEM06110
C     C0 = X-CODIAGONAL = INPUT                                    LEM06120
C     C1 = Y-CODIAGONAL = HALF - C0                                LEM06130
C-----------------------------------------------------------------LEM06140
      GO TO 3                                                      LEM06150
      ENTRY EXERRP(C0,C1,C2,KX,KY)                                 LEM06160
      GO TO 4                                                      LEM06170
C-----------------------------------------------------------------LEM06180
    3 N = KX*KY                                                    LEM06190
      ZERO = 0.0D0                                                 LEM06200
      ONE  = 1.0D0                                                 LEM06210
      TWO  = 2.0D0                                                 LEM06220
C                                                                  LEM06230
C     SET G(I) = GAP FROM GOOD T(MEV) TO NEAREST DISTINCT TMEV I=1,NG LEM06240
      DO 10 I = 1,NG                                               LEM06250
      G(I) = U(I)                                                  LEM06260
   10 CONTINUE                                                     LEM06270
C                                                                  LEM06280
C     REGENERATE A-EIGENVALUES                                     LEM06290
      T0 = DARCOS(-ONE)                                            LEM06300
      T1 = DFLOAT(KX+1)                                            LEM06310
      PIK = T0/T1                                                  LEM06320
      T1 = DFLOAT(KY+1)                                            LEM06330
      PIL = T0/T1                                                  LEM06340
      KP = 0                                                       LEM06350
C                                                                  LEM06360
      DO 30 J = 1,KY                                               LEM06370
      T1 = PIL*DFLOAT(J)                                           LEM06380
      T0 = C2 - TWO*C1*DCOS(T1)                                    LEM06390
      DO 20 I = 1,KX                                               LEM06400
      KP = KP+1                                                    LEM06410
      T1 = PIK*DFLOAT(I)                                           LEM06420
   20  U(KP) = T0 - TWO*C0*DCOS(T1)                                LEM06430
   30 CONTINUE                                                     LEM06440
C                                                                  LEM06450
C     ORDER U VECTOR BY INCREASING ALGEBRAIC SIZE                  LEM06460
      DO 50 K = 2,N                                                LEM06470
      KM1 = K-1                                                    LEM06480
      DO 40 L = 1,KM1                                              LEM06490
```

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

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

```
C                                                                    LEM07600
C     IIX = SEED FOR RANDOM NUMBER GENERATOR USED TO CALCULATE        LEM07610
C           STARTING LANCZOS VECTOR IN LANCZS                         LEM07620
C     V = RANDOM UNIT STARTING VECTOR FOR LANCZS                      LEM07630
C     A*U = EV*U   ||U|| = ONE                                        LEM07640
C                                                                     LEM07650
C     C2 = DIAGONAL OF KX BY KX MATRIX                                LEM07660
C    -C0 = CO-DIAGONAL OF THE KX BY KX MATRIX.                        LEM07670
C    -C1 = Y-CODIAGONAL.                                              LEM07680
C                                                                     LEM07690
C     NOTE THAT THE VECTORS WI,WJ,WII ARE DIMENSIONED INTERNALLY      LEM07700
C     THEY ARE USED JUST TO KEEP FROM REGENERATING INFORMATION.       LEM07710
C     WI,WII = REAL*8 ARRAYS OF DIMENSION AT LEAST KX                 LEM07720
C     WJ     = REAL*8 ARRAY  OF DIMENSION AT LEAST KY.                LEM07730
C                                                                     LEM07740
C     NOTATION USED IN PROGRAM                                        LEM07750
C                                                                     LEM07760
C     PIK = ARCOS(-1)/(KX+1)    PIL = ARCOS(-1)/(KY+1)                LEM07770
C     WI(I) = PIK*I        WJ(J) = PIL*J                              LEM07780
C                                                                     LEM07790
C     U(K) IS A-EV ORDERED BY INCREASING SIZE, K = 1,N                LEM07800
C     LATER U IS USED TO STORE THE TRUE EIGENVECTOR                   LEM07810
C     T0 = C2 - 2*C1*COS(PIL*J)    EV(I,J) = T0 - 2*C0*COS(PIK*I)     LEM07820
C     I = 1,KX    J = 1,KY    KP = (J-1)*KX + I                       LEM07830
C                                                                     LEM07840
C     U(KV) = SIN(PIK*I*IK)*SIN(PIL*J*JK)                             LEM07850
C     IK = 1,KX    JK = 1,KY    KV = (JK-1)*KX + IK                   LEM07860
C     U IS UNSCALED EIGENVECTOR FOR EV(I,J) = Y1                      LEM07870
C     WS = 1/||U||: ||U|| = .5*DSQRT(T2*T3)  T2 = KX+1  T3 = KY+1     LEM07880
C                                                                     LEM07890
C     JVEC  = (1,2) FLAGS COMPUTATIONS TO BE PERFORMED.               LEM07900
C                                                                     LEM07910
C           = (1) MEANS GIVEN X1 FIND Y1 AND KVEC SUCH THAT           LEM07920
C                       Y1 = EV(KVEC) AND |X1-Y1| = MIN               LEM07930
C                 ALSO GIVEN UNIT RITZ VECTOR ASSOCIATED WITH X1      LEM07940
C                     CALCULATE UNIT EIGENVECTOR U, A*U = Y1*U        LEM07950
C                     T2 = ||V-U||   T1 = MAX(|V(K)-U(K)|, K= 1,N)    LEM07960
C                     MAX OCCURS FIRST AT K = KK                      LEM07970
C                                                                     LEM07980
C           = (2) MEANS CALCULATION OF THE PROJECTION OF THE STARTING LEM07990
C                 LANCZOS VECTOR ON EACH EIGENVECTOR OF A.            LEM08000
C                                                                     LEM08010
C----------------------------------------------------------------LEM08020
      DATA MACHEP/Z3410000000000000/                                 LEM08030
      EPSM = 2.0D0*MACHEP                                            LEM08040
C--------------------------------------------------------------- LEM08050
      GO TO 3                                                        LEM08060
      ENTRY EXVECP(C0,C1,C2,KX,KY)                                   LEM08070
      GO TO 4                                                        LEM08080
C----------------------------------------------------------------LEM08090
C     SPECIFY PARAMETERS                                             LEM08100
    3 N = KX*KY                                                      LEM08110
      ZERO = 0.0D0                                                   LEM08120
      HALF = 0.5D0                                                   LEM08130
      ONE  = 1.0D0                                                   LEM08140
```

```
      TWO  = 2.0D0                                         LEM08150
      T0 = DARCOS(-ONE)                                    LEM08160
      T1 = DFLOAT(KX+1)                                    LEM08170
      PIK = T0/T1                                          LEM08180
      T2 = DFLOAT(KY+1)                                    LEM08190
      PIL = T0/T2                                          LEM08200
      WS = TWO/DSQRT(T1*T2)                                LEM08210
C                                                          LEM08220
C     GENERATE WI WJ VECTORS                               LEM08230
      KP = 0                                               LEM08240
      DO 20 J = 1,KY                                       LEM08250
      T1 = PIL*DFLOAT(J)                                   LEM08260
      WJ(J) = T1                                           LEM08270
      T0 = C2 - TWO*C1*DCOS(T1)                            LEM08280
      DO 10 I = 1,KX                                       LEM08290
      KP = KP+1                                            LEM08300
      T1 = PIK*DFLOAT(I)                                   LEM08310
      WI(I) = T1                                           LEM08320
   10 U(KP) = T0 - TWO*C0*DCOS(T1)                         LEM08330
   20 CONTINUE                                             LEM08340
C     U(KP) = EV(I,J) = C2 - 2*C1*COS(PIL*J) - 2*C0*COS(PIK*I)   LEM08350
C                                                          LEM08360
C     INITIALIZE MP VECTOR                                 LEM08370
      DO 30 K = 1,N                                        LEM08380
   30 MP(K) = K                                            LEM08390
C                                                          LEM08400
C     WE ORDER U VECTOR BY INCREASING SIZE OF THE EVS      LEM08410
      DO 50 K = 2,N                                        LEM08420
      KM1 = K-1                                            LEM08430
C                                                          LEM08440
      DO 40 L = 1,KM1                                      LEM08450
      JJ = K - L                                           LEM08460
      IF (U(JJ+1).GE.U(JJ)) GO TO 50                       LEM08470
      EE = U(JJ)                                           LEM08480
      U(JJ) = U(JJ+1)                                      LEM08490
      U(JJ+1) = EE                                         LEM08500
      IEE = MP(JJ)                                         LEM08510
      MP(JJ) = MP(JJ+1)                                    LEM08520
   40 MP(JJ+1) = IEE                                       LEM08530
C                                                          LEM08540
   50 CONTINUE                                             LEM08550
C                                                          LEM08560
      ATOLN = DMAX1(DABS(U(1)),DABS(U(N)))*EPSM            LEM08570
C                                                          LEM08580
      IF (ICOUNT.EQ.1) WRITE(6,60) N,KX,KY,JVEC,C2,C0,C1,ATOLN   LEM08590
   60 FORMAT(/' TRUE ERRORS FOR CONVERGED GOODEV'/         LEM08600
     1 4I6,' = N KX KY JVEC'//                             LEM08610
     1 4E12.5,' = C2 C0 C1 ATOLN'//)                       LEM08620
C                                                          LEM08630
C     KP = MP(K) MEANS EIGENVALUE U(K) CORRESPONDS TO EIGENVECTOR W(KP)  LEM08640
C     COMPUTE TOLERANCE USED IN COMPUTING TRUE MULTIPLICITIES   LEM08650
C                                                          LEM08660
      IF (JVEC.EQ.1) GO TO 180                             LEM08670
C                                                          LEM08680
C     JVEC = 2 SO CALCULATE PROJECTIONS AND WRITE IN FILE 12    LEM08690
```

```
C                                                               LEM08700
      WRITE(12,70)                                              LEM08710
   70 FORMAT(' PROJECTIONS OF LANCZOS STARTING VECTOR ON A-EIGENVECS')  LEM08720
C                                                               LEM08730
      WRITE(12,80)N,KX,KY,IIX,C2,C0,C1,ATOLN                    LEM08740
   80 FORMAT(1X,'A-SIZE',2X,'X-DIM',2X,'Y-DIM',6X,'SVSEED'/3I7,I12/  LEM08750
     1 5X,'A-DIAGONAL',3X,'X-CODIAGONAL',3X,'Y-CODIAGONAL',5X,'ATOLN'/  LEM08760
     2 3E15.8,E10.3)                                            LEM08770
C                                                               LEM08780
      WRITE(12,90)                                              LEM08790
   90 FORMAT(5X,'PROJECTION',8X,'TRUE A-EIGENVALUE',1X,'EV NO'   LEM08800
     1,2X,'VEC NO')                                             LEM08810
C                                                               LEM08820
C     GENERATE SAME RANDOM UNIT VECTOR USED IN THE LANCZS RECURSIONS.  LEM08830
      IIL=IIX                                                   LEM08840
C                                                               LEM08850
C---------------------------------------------------------------LEM08860
      CALL GENRAN(IIL,G,N)                                      LEM08870
C---------------------------------------------------------------LEM08880
C                                                               LEM08890
      DO 100 I = 1,N                                            LEM08900
  100 V(I) = G(I)                                               LEM08910
C                                                               LEM08920
C---------------------------------------------------------------LEM08930
      SUM = FINPRO(N,V(1),1,V(1),1)                             LEM08940
C---------------------------------------------------------------LEM08950
C                                                               LEM08960
      SUM = 1.D0/DSQRT(SUM)                                     LEM08970
C                                                               LEM08980
      DO 110 I = 1,N                                            LEM08990
  110 V(I) = V(I)*SUM                                           LEM09000
C                                                               LEM09010
C     DETERMINE UNIT EIGENVECTOR W ASSOCIATED WITH EACH EV(I,J) = Y1  LEM09020
C     AND CALCULATE THE PROJECTION G(K) OF U ON THE STARTING VECTOR V  LEM09030
C     A*U = EV*U       WS = 1/||WU||: WU = UNSCALED EIGENVECTOR  LEM09040
C                                                               LEM09050
      DO 160 K =1,N                                             LEM09060
C     DETERMINE I J FROM K: MP(K) = KP = (J-1)*KX+I             LEM09070
      KP = MP(K)                                                LEM09080
      I = MOD(KP,KX)                                            LEM09090
      IF (I.EQ.0) I = KX                                        LEM09100
      T1 = WI(I)                                                LEM09110
      J = 1 + (KP-1)/KX                                         LEM09120
      T0 = WJ(J)                                                LEM09130
      T0 = WJ(J)                                                LEM09140
C                                                               LEM09150
      Y1 = C2 - TWO*C1*DCOS(WJ(J)) - TWO*C0*DCOS(WI(I))         LEM09160
C     Y1 = EV(I,J)                                              LEM09170
C                                                               LEM09180
      DO 120 II = 1,KX                                          LEM09190
      T2 = T1*DFLOAT(II)                                        LEM09200
  120 WII(II) = WS*DSIN(T2)                                     LEM09210
C                                                               LEM09220
      KV = 0                                                    LEM09230
      DO 140 JJ = 1,KY                                          LEM09240
```

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

```
  200 Y1 = U(KX1)                                              LEM09800
C                                                              LEM09810
      IF (KX1.EQ.1) EE = U(2) - U(1)                           LEM09820
      IF (KX1.EQ.N) EE = U(N) - U(NM1)                         LEM09830
      IF (KX1.EQ.1.OR.KX1.EQ.N) GO TO 210                      LEM09840
      EE = DMIN1(U(KX1+1)-U(KX1),U(KX1)-U(KX1-1))              LEM09850
  210 CONTINUE                                                 LEM09860
C                                                              LEM09870
      T0 = DABS(ONE - X1/Y1)                                   LEM09880
C                                                              LEM09890
      WRITE(6,220) N,KX1,ICOUNT,Y1,X1,T0,EE                    LEM09900
  220 FORMAT(3I8,' = N, A-EV NUMBER,RITZ NUMBER'//             LEM09910
     1 18X,' TRUEEV',19X,'GOODEV',4X,'RELERROR',4X,'A-MINGAP'/ LEM09920
     1 2E25.16,2E12.3/)                                        LEM09930
C                                                              LEM09940
      IF (EE.GT.ATOLN) GO TO 240                               LEM09950
C                                                              LEM09960
      WRITE(6,230)                                             LEM09970
  230 FORMAT(' Y1 IS A MULTIPLE EIGENVALUE OF A SO WE EXIT'/)  LEM09980
C                                                              LEM09990
      GO TO 310                                                LEM10000
C                                                              LEM10010
C     Y1 IS TOEPLITZ EIGENVALUE CLOSEST TO X1.                 LEM10020
C     CALCULATION OF EIGENVECTOR ASSOCIATED WITH EIGENVALUE Y1 LEM10030
C     A*U = Y1*U                                               LEM10040
C     DETERMINE I J FROM K: MP(K) = KP = (J-1)*KX+I            LEM10050
  240 CONTINUE                                                 LEM10060
      K = KX1                                                  LEM10070
      KP = MP(K)                                               LEM10080
      I = MOD(KP,KX)                                           LEM10090
      IF (I.EQ.0) I = KX                                       LEM10100
      T1 = WI(I)                                               LEM10110
      J = 1 + (KP-1)/KX                                        LEM10120
      T2 = WJ(J)                                               LEM10130
C                                                              LEM10140
      DO 250 II = 1,KX                                         LEM10150
      T0 = T1*DFLOAT(II)                                       LEM10160
  250 WII(II) = WS*DSIN(T0)                                    LEM10170
C                                                              LEM10180
      KV = 0                                                   LEM10190
      DO 270 JJ = 1,KY                                         LEM10200
      T0 = T2*DFLOAT(JJ)                                       LEM10210
      T0 = DSIN(T0)                                            LEM10220
C                                                              LEM10230
      DO 260 II = 1,KX                                         LEM10240
      KV = KV + 1                                              LEM10250
  260 U(KV) = T0*WII(II)                                       LEM10260
C                                                              LEM10270
  270 CONTINUE                                                 LEM10280
C                                                              LEM10290
C     U IS UNIT TRUE EIGENVECTOR OF A ASSOCIATED WITH Y1       LEM10300
C     V IS UNIT RITZVECTOR OF A ASSOCIATED WITH X1             LEM10310
C                                                              LEM10320
      KK = 0                                                   LEM10330
      S = ONE                                                  LEM10340
```

```
      T1 = ZERO                                               LEM10350
C                                                             LEM10360
      DO 280 K = 1,N                                          LEM10370
      IF (DABS(U(K)).LE.T1) GO TO 280                         LEM10380
      T1 = DABS(U(K))                                         LEM10390
      KK = K                                                  LEM10400
  280 CONTINUE                                                LEM10410
      IF (U(KK)*V(KK).LT.ZERO) S = - ONE                      LEM10420
C                                                             LEM10430
      KK = 0                                                  LEM10440
      T1 = ZERO                                               LEM10450
      T2 = ZERO                                               LEM10460
      DO 290 K = 1,N                                          LEM10470
      TEMP = DABS(S*U(K) - V(K))                              LEM10480
      T2 = T2 + TEMP**2                                       LEM10490
      IF (TEMP.LE.T1) GO TO 290                               LEM10500
      KK = K                                                  LEM10510
      T1 = TEMP                                               LEM10520
  290 CONTINUE                                                LEM10530
C                                                             LEM10540
      T2 = DSQRT(T2)                                          LEM10550
      WRITE(6,300) KK,T1,T2                                   LEM10560
  300 FORMAT(' EIGENVECTOR ERROR. MAX ERROR AT COMPONENT = ',I6/  LEM10570
     1 ' MAX DABS(TRUEVEC(K)-RITZVEC(K)) = ',E12.5/           LEM10580
     1 ' NORM(TRUEVEC-RITZVEC)  = ',E12.5/)                   LEM10590
C                                                             LEM10600
  310 CONTINUE                                                LEM10610
C                                                             LEM10620
    4 RETURN                                                  LEM10630
C                                                             LEM10640
C-----END OF EXVEC-------------------------------------------------LEM10650
      END                                                     LEM10660
```

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

```
C---------------------------------------------------------------------  LES00010
C                                                                       LES00020
C     LESUB                                                             LES00030
C                                                                       LES00040
C----------------------------------------------------------------------LES00050
C  Authors:   Jane Cullum and Ralph A. Willoughby (Deceased)           LES00060
C             Los Alamos National Laboratory                           LES00070
C             Los Alamos, New Mexico 87544                             LES00080
C                                                                       LES00090
C             E-mail:  cullumj@lanl.gov                                 LES00100
C                                                                       LES00110
C  These codes are copyrighted by the authors.  These codes            LES00120
C  and modifications of them or portions of them are NOT to be         LES00130
C  incorporated into any commercial codes or used for any other        LES00140
C  commercial purposes such as consulting for other companies,         LES00150
C  without legal agreements with the authors of these Codes.           LES00160
C  If these Codes or portions of them are used in other scientific or  LES00170
C  engineering research works the names of the authors of these codes  LES00180
C  and appropriate references to their written work are to be          LES00190
C  incorporated in the derivative works.                               LES00200
C                                                                       LES00210
C  This header is not to be removed from these codes.                  LES00220
C                                                                       LES00230
C          REFERENCE: Cullum and Willoughby, Chapters 1,2,3,4          LES00231
C          Lanczos Algorithms for Large Symmetric Eigenvalue ComputationsLES00232
C          VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in LES00233
C          Applied Mathematics, 2002. SIAM Publications,               LES00234
C          Philadelphia, PA. USA                                       LES00235
C                                                                       LES00236
C                                                                       LES00237
C             (1) REAL SYMMETRIC                                        LES00240
C             (2) HERMITIAN MATRICES                                    LES00250
C             (3) FACTORED INVERSES OF REAL SYMMETRIC MATRICES AND      LES00260
C             (4) REAL SYMMETRIC GENERALIZED, A*X = EVAL*B*X WHERE      LES00270
C                 B IS POSITIVE DEFINITE, CHOLESKY FACTOR AVAILABLE     LES00280
C                                                                       LES00290
C     ACCORDING TO PFORT THESE SUBROUTINES ARE PORTABLE EXCEPT FOR:     LES00300
C     (1) THE COMPLEX*16 VARIABLES AND THE CORRESPONDING FUNCTIONS      LES00310
C         FOR COMPLEX VARIABLES, DCMPLX, DREAL AND DCONJG USED IN       LES00320
C         THE SUBROUTINE CINPRD (USED ONLY IN CASE (2), HERMITIAN)      LES00330
C     (2) THE ENTRY IN THE SUBROUTINE LPERM USED TO PASS THE           LES00340
C         PERMUTATION FROM THE UPSEC SUBROUTINE TO LPERM. (USED         LES00350
C         ONLY IN CASES (3) AND (4), INVERSE AND GENERALIZED).         LES00360
C                                                                       LES00370
C     SUBROUTINES    BISEC, INVERR, TNORM, LUMP, ISOEV, PRTEST, AND     LES00380
C                    INVERM ARE USED WITH THE LANCZOS EIGENVALUE        LES00390
C                    PROGRAMS LEVAL, HLEVAL, LIVAL AND LGVAL. STURMI,   LES00400
C                    INVERM, LBISEC, AND TNORM ARE USED WITH THE        LES00410
C                    EIGENVECTOR PROGRAMS LEVEC, HLEVEC, LIVEC AND      LES00420
C                    LGVEC.  LPERM IS USED WITH LIVEC AND LGVEC.        LES00430
```

```
C                     IN THE HERMITIAN CASE, THE SUBROUTINE CINPRD         LES00440
C                     IS ALSO USED.                                        LES00450
C                                                                          LES00460
C-----COMPUTE T-EIGENVALUES BY BISECTION-------------------------------LES00470
C                                                                          LES00480
      SUBROUTINE BISEC(ALPHA,BETA,BETA2,VB,VS,LBD,UBD,EPS,TTOL,MP,         LES00490
     1 NINT,MEV,NDIS,IC,IWRITE)                                           LES00500
C                                                                          LES00510
C--------------------------------------------------------------------LES00520
      DOUBLE PRECISION  ALPHA(1),BETA(1),BETA2(1),VB(1),VS(1)             LES00530
      DOUBLE PRECISION  LBD(1),UBD(1),EPS,EPT,EP0,EP1,TEMP,TTOL           LES00540
      DOUBLE PRECISION  ZERO,ONE,HALF,YU,YV,LB,UB,XL,XU,X1,X0,XS,BETAM    LES00550
      INTEGER  MP(1),IDEF(100)                                            LES00560
      DOUBLE PRECISION  DABS, DSQRT, DMAX1, DMIN1, DFLOAT                 LES00570
C--------------------------------------------------------------------LES00580
C     COMPUTES EIGENVALUES OF T(1,MEV) BY LOOPING INTERNALLY ON THE       LES00590
C     USER-SPECIFIED INTERVALS,  (LB(J),UB(J)), J = 1,NINT.   INTERVALS   LES00600
C     ARE TREATED AS OPEN ON THE LEFT AND CLOSED ON THE RIGHT.            LES00610
C     THE BISEC SUBROUTINE SIMULTANEOUSLY LABELS SPURIOUS T-EIGENVALUES   LES00620
C     AND DETERMINES THE T-MULTIPLICITIES OF EACH GOOD T-EIGENVALUE.      LES00630
C     SPURIOUS T-EIGENVALUES ARE LABELLED BY A T-MULTIPLICITY = 0.        LES00640
C     ANY T-EIGENVALUE WITH A T-MULTIPLICITY >= 1 IS 'GOOD'.              LES00650
C                                                                          LES00660
C     IF IWRITE = 0 THEN MOST OF THE WRITES TO FILE 6 ARE NOT             LES00670
C     ACTIVATED.                                                          LES00680
C                                                                          LES00690
C     NOTE THAT PROGRAM ASSUMES THAT NO MORE THAN MMAX/2 EIGENVALUES      LES00700
C     OF T(1,MEV) ARE TO BE COMPUTED IN ANY ONE OF THE SUBINTERVALS       LES00710
C     CONSIDERED, WHERE MMAX = DIMENSION OF VB SPECIFIED BY THE USER      LES00720
C     IN THE MAIN PROGRAM LEVAL.                                          LES00730
C                                                                          LES00740
C     ON ENTRY                                                            LES00750
C     BETA2(J) IS SET = BETA(J)*BETA(J).  THE STORAGE FOR BETA2 COULD     LES00760
C     BE ELIMINATED BY RECOMPUTING THE BETA(J)**2 FOR EACH STURM          LES00770
C     SEQUENCE.                                                           LES00780
C                                                                          LES00790
C     EPS = 2*MACHEP =  4.4 * 10**-16 ON IBM 3081.                        LES00800
C     TTOL = EPS*TKMAX WHERE                                              LES00810
C     TKMAX = MAX(|ALPHA(K)|,BETA(K), K=1,KMAX)                           LES00820
C                                                                          LES00830
C     ON EXIT                                                             LES00840
C     NDIS = TOTAL NUMBER OF COMPUTED DISTINCT EIGENVALUES OF             LES00850
C            T(1,MEV) ON THE UNION OF THE (LB,UB) INTERVALS.              LES00860
C     VS = COMPUTED DISTINCT EIGENVALUES OF T(1,MEV) IN ALGEBRAICALLY-    LES00870
C          INCREASING ORDER                                              LES00880
C     MP = CORRESPONDING T-MULTIPLICITIES OF THESE EIGENVALUES            LES00890
C     MP(I) = (0,1,MI), MI>1, I=1,NDIS  MEANS:                           LES00900
C        (0)  V(I) IS SPURIOUS                                           LES00910
C        (1)  V(I) IS ISOLATED AND GOOD                                  LES00920
C        (MI) V(I) IS MULTIPLE AND HENCE A CONVERGED GOOD T-EIGENVALUE.  LES00930
C     IC = TOTAL NUMBER OF STURMS USED                                   LES00940
C                                                                          LES00950
C     DEFAULTS                                                            LES00960
C     ISKIP = 0 INITIALLY. IF DEFAULT OCCURS ON J-TH SUB-INTERVAL, SET    LES00970
C             ISKIP=ISKIP+1 AND IDEF(ISKIP) = J                          LES00980
```

```
C               DEFAULTS OCCUR IF THERE ARE NO T-EIGENVALUES  IN THE     LES00990
C               SUBINTERVAL SPECIFIED OR IF THE NUMBER                   LES01000
C               OF STURMS SEQUENCES REQUIRED EXCEEDS MXSTUR.             LES01010
C               WHEN A DEFAULT OCCURS THE PROGRAM                        LES01020
C               SKIPS THE INTERVAL INVOLVED AND GOES ON TO THE NEXT      LES01030
C               INTERVAL.                                                LES01040
C                                                                        LES01050
C------------------------------------------------------------------------LES01060
C     SPECIFY PARAMETERS                                                 LES01070
      ZERO = 0.0D0                                                       LES01080
      ONE  = 1.0D0                                                       LES01090
      HALF = 0.5D0                                                       LES01100
      MXSTUR = IC                                                        LES01110
      NDIS = 0                                                           LES01120
      IC = 0                                                             LES01130
      ISKIP = 0                                                          LES01140
      MP1 = MEV+1                                                        LES01150
C     SAVE THEN SET BETA(MEV+1) = 0. GENERATE BETA**2                    LES01160
      BETAM = BETA(MP1)                                                  LES01170
      BETA(MP1) = ZERO                                                   LES01180
C                                                                        LES01190
      DO 10 I = 1,MP1                                                    LES01200
   10 BETA2(I) = BETA(I)*BETA(I)                                         LES01210
C                                                                        LES01220
C     EP0 IS USED IN T-MULTIPLICITY AND SPURIOUS TESTS                   LES01230
C     EP1 AND EPS ARE USED IN THE BISEC CONVERGENCE TEST                 LES01240
C                                                                        LES01250
      TEMP = DFLOAT(MEV+1000)                                            LES01260
      EP0  = TEMP*TTOL                                                   LES01270
      EP1  = DSQRT(TEMP)*TTOL                                            LES01280
C                                                                        LES01290
      WRITE(6,20)MEV,NINT                                                LES01300
   20 FORMAT(/' BISEC CALCULATION'/' ORDER OF T IS',I6/                  LES01310
     1' NUMBER OF INTERVALS IS',I6/)                                     LES01320
C                                                                        LES01330
      WRITE(6,30) EP0,EP1                                                LES01340
   30 FORMAT(/' MULTOL, TOLERANCE USED IN T-MULTIPLICITY AND SPURIOUS TELES01350
     1STS = ',E10.3/' BISTOL, TOLERANCE USED IN BISEC CONVERGENCE TEST =LES01360
     1 ',E10.3/)                                                        LES01370
C                                                                        LES01380
C     LOOP ON THE NINT INTERVALS  (LB(J),UB(J)), J=1,NINT                LES01390
      DO 430 JIND = 1,NINT                                               LES01400
      LB = LBD(JIND)                                                     LES01410
      UB = UBD(JIND)                                                     LES01420
C                                                                        LES01430
      WRITE(6,40)JIND,LB,UB                                              LES01440
   40 FORMAT(//1X,'BISEC INTERVAL NO',2X,'LOWER BOUND',2X,'UPPER BOUND'/LES01450
     1I18,2E13.5/)                                                       LES01460
C                                                                        LES01470
C     INITIALIZATION AND PARAMETER SPECIFICATION                         LES01480
C     ICT IS TOTAL STURM COUNT ON (LB,UB)                                LES01490
C                                                                        LES01500
      NA = 0                                                             LES01510
      MD = 0                                                             LES01520
      NG = 0                                                             LES01530
```

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

```
C                                                               LES02090
      K = MT                                                    LES02100
  150 CONTINUE                                                  LES02110
      IC0 = 0                                                   LES02120
      XL = VB(K)                                                LES02130
      MTK = MT+K                                                LES02140
      XU = VB(MTK)                                              LES02150
C                                                               LES02160
      ISTURM = 3                                                LES02170
      X1 = XU                                                   LES02180
      IC0 = IC0 + 1                                             LES02190
      GO TO 330                                                 LES02200
C     FORWARD STURM CALCULATION AT XU                           LES02210
  160 NU=NEV                                                    LES02220
C                                                               LES02230
C     BISECTION LOOP FOR KTH T-EIGENVALUE. TEST  X1=MIDPOINT OF (XL,XU) LES02240
      ISTURM = 4                                                LES02250
  170 CONTINUE                                                  LES02260
      X1 = (XL+XU)*HALF                                         LES02270
      XS = DABS(XL)+DABS(XU)                                    LES02280
      X0 = XU-XL                                                LES02290
      EPT = EPS*XS+EP1                                          LES02300
C                                                               LES02310
C     EPT IS CONVERGENCE TOLERANCE FOR KTH T-EIGENVALUE         LES02320
C                                                               LES02330
      IF (X0.LE.EPT) GO TO 230                                  LES02340
C                                                               LES02350
C     T-EIGENVALUE HAS NOT YET CONVERGED                        LES02360
C                                                               LES02370
      IC0 = IC0 + 1                                             LES02380
      GO TO 330                                                 LES02390
C     FORWARD STURM CALCULATION AT CURRENT T-EIGENVALUE APPROXIMATION.  LES02400
  180 CONTINUE                                                  LES02410
C                                                               LES02420
C     UPDATE T-EIGENVALUE INTERVAL (XL,XU)                      LES02430
C                                                               LES02440
      IF (NEV.LT.K) GO TO 190                                   LES02450
C                                                               LES02460
C     NUMBER OF T-EIGENVALUES NEV = K                           LES02470
      XL = X1                                                   LES02480
      GO TO 170                                                 LES02490
  190 CONTINUE                                                  LES02500
C     NUMBER OF T-EIGENVALUES NEV<K                             LES02510
      XU = X1                                                   LES02520
      NU = NEV                                                  LES02530
C                                                               LES02540
C     UPDATE OF T-EIGENVALUE BOUNDS                             LES02550
C                                                               LES02560
      IF (NEV.EQ.0) GO TO 210                                   LES02570
C                                                               LES02580
      DO 200 I = 1,NEV                                          LES02590
  200 VB(I) = DMAX1(X1,VB(I))                                   LES02600
C                                                               LES02610
  210 NEV1 = NEV+1                                              LES02620
C                                                               LES02630
```

```
      DO 220 II = NEV1,K                                         LES02640
      I = MT+II                                                  LES02650
  220 VB(I) = DMIN1(X1,VB(I))                                    LES02660
C                                                                LES02670
      GO TO 170                                                  LES02680
C                                                                LES02690
C     END (XL,XU) BISECTION LOOP FOR KTH T-EIGENVALUE ON (LB,UB) LES02700
C     TEST FOR T-MULTIPLICITY AND IF SIMPLE THEN TEST FOR SPURIOUSNESS LES02710
C                                                                LES02720
  230 CONTINUE                                                   LES02730
      NDIS = NDIS+1                                              LES02740
      MD = MD+1                                                  LES02750
      VS(NDIS) = X1                                              LES02760
C                                                                LES02770
      JSTURM = 1                                                 LES02780
      X1 = XL-EP0                                                LES02790
      GO TO 370                                                  LES02800
C     BACKWARD STURM CALCULATION                                 LES02810
  240 KL = KEV                                                   LES02820
      JL = JEV                                                   LES02830
C                                                                LES02840
      JSTURM = 2                                                 LES02850
      IC0 = IC0 + 2                                              LES02860
      X1 = XU+EP0                                                LES02870
      GO TO 370                                                  LES02880
C     BACKWARD STURM CALCULATION                                 LES02890
  250 JU = JEV                                                   LES02900
      KU = KEV                                                   LES02910
C                                                                LES02920
C     FOR T(1,MEV)                                               LES02930
C     NU - KU = NO. T-EIGENVALUES ON (XU, XU + EP0)              LES02940
C     KL - KU = NO. T-EIGENVALUES ON (XL - EP0, XU + EP0)        LES02950
C                                                                LES02960
C     FOR T(2,MEV)                                               LES02970
C     JL -JU = NO. T-EIGENVALUES ON (XL - EP0, XU + EP0)         LES02980
C                                                                LES02990
C     IS THIS A SIMPLE T-EIGENVALUE?                             LES03000
C                                                                LES03010
      IF (KL-KU-1.EQ.0) GO TO 290                                LES03020
C                                                                LES03030
C     VS(NDIS) = KTH-T-EIGENVALUE OF (LB,UB) IS MULTIPLE AND HENCE GOOD LES03040
      IF (KU.EQ.NU) GO TO 280                                    LES03050
C     CONTINUE TO CHECK FOR T-MULTIPLICITY                       LES03060
  260 CONTINUE                                                   LES03070
      ISTURM = 5                                                 LES03080
      X1 = X1+EP0                                                LES03090
      IC0 = IC0 + 1                                              LES03100
      GO TO 330                                                  LES03110
C     FORWARD STURM CALCULATION                                  LES03120
  270 KNE = KU-NEV                                               LES03130
      KU = NEV                                                   LES03140
      IF (KNE.NE.0) GO TO 260                                    LES03150
C     SPECIFY T-MULTIPLICITY = MP(NDIS)                          LES03160
  280 MPEV = KL-KU                                               LES03170
      KNEW = KU                                                  LES03180
```

```
      GO TO 300                                           LES03190
C     END MULTIPLE CASE                                   LES03200
C                                                         LES03210
C     T-EIGENVALUE IS SIMPLE   CHECK IF IT IS SPURIOUS    LES03220
  290 CONTINUE                                            LES03230
      MPEV = 1                                            LES03240
      IF (JU.LT.JL) MPEV=0                                LES03250
      KNEW = K-1                                          LES03260
C                                                         LES03270
C     X1 >= XU+EP0                                        LES03280
C     SPURIOUS TEST AND T-SIMPLE CASE COMPLETED           LES03290
C     START OF NEXT T-EIGENVALUE COMPUTATION              LES03300
C                                                         LES03310
  300 K = KNEW                                            LES03320
      MP(NDIS) = MPEV                                     LES03330
      IF (MPEV.GE.1) NG = NG + 1                          LES03340
C                                                         LES03350
      IF (IWRITE.NE.0) WRITE(6,310) VS(NDIS),MPEV,MD,NG   LES03360
  310 FORMAT(E25.16,3I6)                                  LES03370
C                                                         LES03380
C     UPDATE STURM COUNT. IC0 = STURM COUNT FOR KTH T-EIGENVALUE  LES03390
      ICT = ICT + IC0                                     LES03400
C                                                         LES03410
C     EXIT TEST FOR K DO LOOP                             LES03420
C                                                         LES03430
      IF (K.LE.0) GO TO 410                               LES03440
C                                                         LES03450
C     UPDATE LOWER BOUNDS                                 LES03460
      DO 320 I=1,KNEW                                     LES03470
  320 VB(I) = DMAX1(X1,VB(I))                             LES03480
C                                                         LES03490
      GO TO 150                                           LES03500
C     END OF BISECTION LOOP FOR KTH T-EIGENVALUE          LES03510
C                                                         LES03520
C     FORWARD STURM CALCULATION                           LES03530
  330 NEV = -NA                                           LES03540
      YU  = ONE                                           LES03550
C                                                         LES03560
      DO 360 I = 1,MEV                                    LES03570
      IF (YU.NE.ZERO) GO TO 340                           LES03580
      YV = BETA(I)/EPS                                    LES03590
      GO TO 350                                           LES03600
  340 YV = BETA2(I)/YU                                    LES03610
  350 YU = X1 - ALPHA(I) - YV                             LES03620
      IF (YU.GE.ZERO) GO TO 360                           LES03630
      NEV = NEV + 1                                       LES03640
  360 CONTINUE                                            LES03650
C     NEV = NUMBER OF T-EIGENVALUES ON (X1,UB)            LES03660
C                                                         LES03670
      GO TO (50,60,160,180,270), ISTURM                  LES03680
C                                                         LES03690
C     BACKWARD STURM CALCULATION FOR T(1,MEV) AND T(2,MEV)  LES03700
  370 KEV = -NA                                           LES03710
      YU  = ONE                                           LES03720
C                                                         LES03730
```

```
      DO 400 II = 1,MEV                                         LES03740
      I = MP1-II                                                LES03750
      IF (YU.NE.ZERO) GO TO 380                                 LES03760
      YV = BETA(I+1)/EPS                                        LES03770
      GO TO 390                                                 LES03780
  380 YV = BETA2(I+1)/YU                                        LES03790
  390 YU = X1-ALPHA(I)-YV                                       LES03800
      JEV = 0                                                   LES03810
      IF (YU.GE.ZERO) GO TO 400                                 LES03820
      KEV = KEV+1                                               LES03830
      JEV = 1                                                   LES03840
  400 CONTINUE                                                  LES03850
      JEV = KEV-JEV                                             LES03860
C                                                               LES03870
      GO TO (240,250), JSTURM                                   LES03880
C                                                               LES03890
C     KEV = -NA + (NUMBER OF T(1,MEV) EIGENVALUES) > X1         LES03900
C     JEV = -NA + (NUMBER OF T(2,MEV) EIGENVALUES) > X1         LES03910
C     SET PARAMETERS FOR NEXT INTERVAL                          LES03920
  410 CONTINUE                                                  LES03930
      IC = ICT+IC                                               LES03940
      MXSTUR = MXSTUR-ICT                                       LES03950
C                                                               LES03960
      WRITE(6,420) JIND,NG,MD                                   LES03970
  420 FORMAT(/' T-EIGENVALUE CALCULATION ON INTERVAL',I6,'  IS COMPLETE'LES03980
     1 /3X,'NO. GOOD',3X,'NO. DISTINCT'/I10,I13)                LES03990
C                                                               LES04000
  430 CONTINUE                                                  LES04010
C                                                               LES04020
C     END LOOP ON THE SUBINTERVALS (LB(J),UB(J)), J=1,NINT      LES04030
C     ISKIP OUTPUT                                              LES04040
C                                                               LES04050
      IF (ISKIP.GT.0) WRITE(6,440)ISKIP                         LES04060
  440 FORMAT(' BISEC DEFAULTED ON',I3,3X,'INTERVALS'/           LES04070
     1 ' DEFAULTS OCCUR IF AN INTERVAL HAS NO T-EIGENVALUES'/   LES04080
     2 ' OR THE STURM ESTIMATE EXCEEDS THE USER-SPECIFIED LIMIT'/) LES04090
C                                                               LES04100
      IF (ISKIP.GT.0) WRITE(6,450)(IDEF(I), I=1,ISKIP)          LES04110
  450 FORMAT(' BISEC DEFAULTED ON INTERVALS'/(10I8))            LES04120
C                                                               LES04130
C     RESET BETA AT I = MP1                                     LES04140
      BETA(MP1) = BETAM                                         LES04150
C-----END OF BISEC----------------------------------------------LES04160
      RETURN                                                    LES04170
      END                                                       LES04180
C                                                               LES04190
C-----INVERSE ITERATION ON T(1,MEV)-----------------------------LES04200
C                                                               LES04210
      SUBROUTINE INVERR(ALPHA,BETA,V1,V2,VS,EPS,G,MP,MEV,MMB,NDIS,NISO, LES04220
     1 N,IKL,IT,IWRITE)                                         LES04230
C                                                               LES04240
C--------------------------------------------------------------LES04250
      DOUBLE PRECISION ALPHA(1),BETA(*),V1(1),V2(*),VS(*)       LES04260
      DOUBLE PRECISION X1,U,Z,EST,TEMP,T0,T1,RATIO,SUM,XU,NORM,TSUM  LES04270
      DOUBLE PRECISION  BETAM,EPS,EPS3,EPS4,ZERO,ONE            LES04280
```

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

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

```
   50 G(I) = GSUM*G(I)                                          LES05390
C                                                               LES05400
C     LOOP ON ISOLATED GOOD T-EIGENVALUES IN VS (MP(I) = 1) TO  LES05410
C     CALCULATE CORRESPONDING UNIT EIGENVECTOR OF T(1,MEV)      LES05420
C                                                               LES05430
      DO 180 JEV = 1,NDIS                                       LES05440
C                                                               LES05450
      IF (MP(JEV).EQ.0) GO TO 180                               LES05460
      NG = NG + 1                                               LES05470
      IF (MP(JEV).NE.1) GO TO 180                               LES05480
C                                                               LES05490
      IT = 1                                                    LES05500
      NISO = NISO + 1                                           LES05510
      X1 = VS(JEV)                                              LES05520
C                                                               LES05530
C     INITIALIZE RIGHT HAND SIDE FOR INVERSE ITERATION          LES05540
      DO 60 I = 1,MEV                                           LES05550
   60 V2(I) = G(I)                                              LES05560
C                                                               LES05570
C     TRIANGULAR FACTORIZATION WITH NEAREST NEIGHBOR PIVOT      LES05580
C     STRATEGY. INTERCHANGES ARE LABELLED BY SETTING BETA < 0.  LES05590
C                                                               LES05600
   70 CONTINUE                                                  LES05610
      U = ALPHA(1)-X1                                           LES05620
      Z = BETA(2)                                               LES05630
C                                                               LES05640
      DO 90 I = 2,MEV                                           LES05650
      IF (BETA(I).GT.DABS(U)) GO TO 80                          LES05660
C     NO INTERCHANGE                                            LES05670
      V1(I-1) = Z/U                                             LES05680
      V2(I-1) = V2(I-1)/U                                       LES05690
      V2(I) = V2(I)-BETA(I)*V2(I-1)                             LES05700
      RATIO = BETA(I)/U                                         LES05710
      U = ALPHA(I)-X1-Z*RATIO                                   LES05720
      Z = BETA(I+1)                                             LES05730
      GO TO 90                                                  LES05740
   80 CONTINUE                                                  LES05750
C     INTERCHANGE CASE                                          LES05760
      RATIO = U/BETA(I)                                         LES05770
      BETA(I) = -BETA(I)                                        LES05780
      V1(I-1) = ALPHA(I)-X1                                     LES05790
      U = Z-RATIO*V1(I-1)                                       LES05800
      Z = -RATIO*BETA(I+1)                                      LES05810
      TEMP = V2(I-1)                                            LES05820
      V2(I-1) = V2(I)                                           LES05830
      V2(I) = TEMP-RATIO*V2(I)                                  LES05840
   90 CONTINUE                                                  LES05850
      IF (U.EQ.ZERO) U = EPS3                                   LES05860
C                                                               LES05870
C     SMALLNESS TEST AND DEFAULT VALUE FOR LAST COMPONENT       LES05880
C     PIVOT(I-1) = |BETA(I)| FOR INTERCHANGE CASE               LES05890
C     (I-1,I+1) ELEMENT IN RIGHT FACTOR = BETA(I+1)             LES05900
C     END OF FACTORIZATION AND FORWARD SUBSTITUTION             LES05910
C                                                               LES05920
C     BACK SUBSTITUTION                                         LES05930
```

```
        V2(MEV) = V2(MEV)/U                                    LES05940
        DO 110 II = 1,MM1                                       LES05950
        I = MEV-II                                             LES05960
        IF (BETA(I+1).LT.ZERO) GO TO 100                       LES05970
C       NO INTERCHANGE                                         LES05980
        V2(I) = V2(I)-V1(I)*V2(I+1)                            LES05990
        GO TO 110                                             LES06000
C       INTERCHANGE CASE                                       LES06010
  100 BETA(I+1) = -BETA(I+1)                                   LES06020
        V2(I) = (V2(I)-V1(I)*V2(I+1)-BETA(I+2)*V2(I+2))/BETA(I+1) LES06030
  110 CONTINUE                                                 LES06040
C                                                              LES06050
C       TESTS FOR CONVERGENCE OF INVERSE ITERATION             LES06060
C       IF SUM |V2| COMPS. LE. 1 AND IT. LE. ITER DO ANOTHER INVIT STEP LES06070
C                                                              LES06080
        NORM = DABS(V2(MEV))                                   LES06090
        DO 120 II = 1,MM1                                       LES06100
        I = MEV-II                                             LES06110
  120 NORM = NORM+DABS(V2(I))                                  LES06120
C                                                              LES06130
        IF (NORM.GE.ONE) GO TO 140                             LES06140
        IT = IT+1                                             LES06150
        IF (IT.GT.ITER) GO TO 140                              LES06160
        XU = EPS4/NORM                                         LES06170
C                                                              LES06180
        DO 130 I = 1,MEV                                        LES06190
  130 V2(I) = V2(I)*XU                                         LES06200
C                                                              LES06210
        GO TO 70                                              LES06220
C       ANOTHER INVERSE ITERATION STEP                         LES06230
C                                                              LES06240
C       INVERSE ITERATION FINISHED                             LES06250
C       NORMALIZE COMPUTED T-EIGENVECTOR : V2 = V2/||V2||      LES06260
  140 CONTINUE                                                 LES06270
        SUM = FINPRO(MEV,V2(1),1,V2(1),1)                      LES06280
        SUM = ONE/DSQRT(SUM)                                   LES06290
C                                                              LES06300
        DO 150 II = 1,MEV                                       LES06310
  150 V2(II) = SUM*V2(II)                                      LES06320
C                                                              LES06330
C       SAVE ERROR ESTIMATE FOR LATER OUTPUT                   LES06340
        EST = BETAM*DABS(V2(MEV))                              LES06350
        IF (IT.GT.ITER) EST = -EST                             LES06360
        MEVPNI = MEV + NISO                                   LES06370
        G(MEVPNI) = EST                                       LES06380
        IF (IWRITE.EQ.0) GO TO 180                             LES06390
C                                                              LES06400
C       FILE 6 (TERMINAL) OUTPUT OF ERROR ESTIMATES.           LES06410
        IF (JEV.EQ.1) GAP = VS(2) - VS(1)                      LES06420
        IF (JEV.EQ.MEV) GAP = VS(MEV) - VS(MEV-1)              LES06430
        IF (JEV.EQ.MEV.OR.JEV.EQ.1) GO TO 160                  LES06440
        TEMP = DMIN1(VS(JEV+1)-VS(JEV),VS(JEV)-VS(JEV-1))      LES06450
        GAP = TEMP                                            LES06460
  160 CONTINUE                                                 LES06470
C                                                              LES06480
```

```
      WRITE(6,170) NISO,JEV,X1,EST,GAP                          LES06490
  170 FORMAT(2I6,E25.16,2E12.3)                                 LES06500
C                                                               LES06510
  180 CONTINUE                                                  LES06520
C                                                               LES06530
C     END ERROR ESTIMATE LOOP ON ISOLATED GOOD T-EIGENVALUES.   LES06540
C     GENERATE DISTINCT MINGAPS FOR T(1,MEV).  THIS IS USEFUL AS AN  LES06550
C     INDICATOR OF THE GOODNESS OF THE INVERSE ITERATION ESTIMATES.  LES06560
C     TRANSFER ISOLATED GOOD T-EIGENVALUES AND CORRESPONDING TMINGAPS  LES06570
C     TO V2 AND V1 FOR OUTPUT PURPOSES ONLY.                    LES06580
C                                                               LES06590
      NM1 = NDIS - 1                                            LES06600
      G(NDIS) = VS(NM1)-VS(NDIS)                                LES06610
      G(1) = VS(2)-VS(1)                                        LES06620
C                                                               LES06630
      DO 190 J = 2,NM1                                          LES06640
      T0 = VS(J)-VS(J-1)                                        LES06650
      T1 = VS(J+1)-VS(J)                                        LES06660
      G(J) = T1                                                 LES06670
      IF (T0.LT.T1) G(J)=-T0                                    LES06680
  190 CONTINUE                                                  LES06690
      ISO = 0                                                   LES06700
      DO 200 J = 1,NDIS                                         LES06710
      IF (MP(J).NE.1) GO TO 200                                 LES06720
      ISO = ISO+1                                               LES06730
      V1(ISO) = G(J)                                            LES06740
      V2(ISO) = VS(J)                                           LES06750
  200 CONTINUE                                                  LES06760
C                                                               LES06770
      IF(NISO.EQ.0) GO TO 250                                   LES06780
C                                                               LES06790
C     ERROR ESTIMATES ARE WRITTEN TO FILE 4                     LES06800
      WRITE(4,210)MEV,NDIS,NG,NISO,N,IKL,ITER,BETAM             LES06810
  210 FORMAT(1X,'TSIZE',2X,'NDIS',1X,'NGOOD',2X,'NISO',1X,'ASIZE'/5I6/  LES06820
     1 4X,'RHSEED',2X,'MXINIT',5X,'BETAM'/I10,I8,E10.3/         LES06830
     2 2X,'GOODEVNO',8X,'GOOD T-EIGENVALUE',6X,'BETAM*UM',7X,'TMINGAP')  LES06840
C                                                               LES06850
      ISPUR = 0                                                 LES06860
      I = 0                                                     LES06870
      DO 240 J = 1,NDIS                                         LES06880
      IF(MP(J).NE.0) GO TO 220                                  LES06890
      ISPUR = ISPUR + 1                                         LES06900
      GO TO 240                                                 LES06910
  220 IF(MP(J).NE.1) GO TO 240                                  LES06920
      I = I + 1                                                 LES06930
      MEVI = MEV + I                                            LES06940
      IGOOD = J - ISPUR                                         LES06950
      WRITE(4,230) IGOOD,V2(I),G(MEVI),V1(I)                    LES06960
  230 FORMAT(I10,E25.16,2E14.3)                                 LES06970
  240 CONTINUE                                                  LES06980
      GO TO 270                                                 LES06990
C                                                               LES07000
  250 WRITE(4,260)                                              LES07010
  260 FORMAT(/' THERE ARE NO ISOLATED T-EIGENVALUES SO NO ERROR ESTIMATELES07020
     1S WERE COMPUTED')                                         LES07030
```

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

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

```
      IF (LINDEX(KK).NE.0) GO TO 30                           LES08140
      ISPUR = ISPUR + 1                                       LES08150
      INDSUM = INDSUM + 1                                     LES08160
   30 INDSUM = INDSUM + LINDEX(KK)                            LES08170
C                                                             LES08180
C     IF (JF-JI.GE.1) WRITE(6,40) (V1(KKK), KKK=JI,JF)        LES08190
   40 FORMAT(/' LUMP LUMPS THE T-EIGENVALUES'/(4E20.13))      LES08200
C                                                             LES08210
C     COMPUTE THE 'COMBINED' T-EIGENVALUE AND THE RESULTING   LES08220
C     T-MULTIPLICITY                                          LES08230
      K = JI - 1                                              LES08240
   50 K = K+1                                                 LES08250
      IF (K.GT.JF) GO TO 70                                   LES08260
      IF (LINDEX(K) .NE.1) GO TO 50                           LES08270
      NLOOP = NLOOP + 1                                       LES08280
      V1(NLOOP) = V1(K)                                       LES08290
      GO TO 100                                               LES08300
   60 ICOUNT = ICOUNT + 1                                     LES08310
      GO TO 10                                                LES08320
C                                                             LES08330
C     ALL INDICES WERE 0 OR >1                                LES08340
   70 NLOOP = NLOOP + 1                                       LES08350
      IDIF = INDSUM - ISPUR                                   LES08360
      IF (IDIF.EQ.0) GO TO 90                                 LES08370
C                                                             LES08380
      SUM = ZERO                                              LES08390
      DO 80 KK = JI,JF                                        LES08400
   80 SUM = SUM + V1(KK) * DFLOAT(LINDEX(KK))                 LES08410
C                                                             LES08420
      V1(NLOOP) = SUM/DFLOAT(IDIF)                            LES08430
      GO TO 100                                               LES08440
   90 V1(NLOOP) = V1(JI)                                      LES08450
  100 LINDEX(NLOOP) = INDSUM                                  LES08460
      IDIF = INDSUM - ISPUR                                   LES08470
      IF (IDIF.EQ.0.AND.ISPUR.EQ.1) LINDEX(NLOOP) = 0         LES08480
      IF (J.EQ.LOOP) GO TO 110                                LES08490
      ICOUNT = 1                                              LES08500
      JI= J+1                                                 LES08510
      THOLD = DMAX1(RELTOL*DABS(V1(JI)),SCALE2*MULTOL)        LES08520
C     THOLD = DMAX1(RELTOL*DABS(V1(JI)),RELTOL)               LES08530
      IF (JI.LT.LOOP) GO TO 10                                LES08540
      NLOOP = NLOOP + 1                                       LES08550
      V1(NLOOP)= V1(JI)                                       LES08560
      LINDEX(NLOOP) = LINDEX(JI)                              LES08570
  110 CONTINUE                                                LES08580
C                                                             LES08590
C     ON RETURN V1 CONTAINS THE DISTINCT T-EIGENVALUES        LES08600
C     LINDEX CONTAINS THE CORRESPONDING T-MULTIPLICITIES      LES08610
C                                                             LES08620
      LOOP = NLOOP                                            LES08630
      RETURN                                                  LES08640
C-----END OF LUMP-------------------------------------------------LES08650
      END                                                     LES08660
C                                                             LES08670
C                                                             LES08680
```

```
C-----START OF ISOEV----------------------------------------------------LES08690
C                                                                       LES08700
      SUBROUTINE ISOEV(VS,GAPTOL,MULTOL,SCALE1,G,MP,NDIS,NG,NISO)        LES08710
C                                                                       LES08720
C----------------------------------------------------------------------LES08730
      DOUBLE PRECISION  VS(*),T0,T1,MULTOL,GAPTOL,SCALE1,TEMP           LES08740
      REAL   G(1),GAP                                                    LES08750
      INTEGER  MP(1)                                                     LES08760
      REAL  ABS                                                          LES08770
      DOUBLE PRECISION  DABS, DMAX1                                      LES08780
C----------------------------------------------------------------------LES08790
C     GENERATE DISTINCT TMINGAPS AND USE THEM TO LABEL THE ISOLATED      LES08800
C     GOOD T-EIGENVALUES THAT ARE VERY CLOSE TO SPURIOUS ONES.           LES08810
C     ERROR ESTIMATES WILL NOT BE COMPUTED FOR THESE T-EIGENVALUES.      LES08820
C                                                                       LES08830
C     ON ENTRY AND EXIT                                                  LES08840
C     VS CONTAINS THE COMPUTED DISTINCT EIGENVALUES OF T(1,MEV)          LES08850
C     MP CONTAINS THE CORRESPONDING T-MULTIPLICITIES                     LES08860
C     NDIS = NUMBER OF DISTINCT EIGENVALUES                             LES08870
C     GAPTOL = RELATIVE GAP TOLERANCE SET IN MAIN                       LES08880
C                                                                       LES08890
C     ON EXIT                                                            LES08900
C     G CONTAINS THE TMINGAPS.                                           LES08910
C     G(I) < 0 MEANS MINGAP IS DUE TO LEFT GAP                          LES08920
C     MP(I) IS NOT CHANGED EXCEPT THAT  MP(I)=-1, IF MP(I)=1,            LES08930
C     TMINGAP WAS TOO SMALL AND DUE TO A SPURIOUS T-EIGENVALUE.          LES08940
C                                                                       LES08950
C     IF MP(I)=-1 THAT SIMPLE GOOD T-EIGENVALUE WILL BE SKIPPED          LES08960
C     IN THE SUBSEQUENT ERROR ESTIMATE COMPUTATIONS IN INVERR           LES08970
C     THAT IS, WE COMPUTE ERROR ESTIMATES ONLY FOR THOSE GOOD           LES08980
C     T-EIGENVALUES WITH MP(I)=1.                                        LES08990
C----------------------------------------------------------------------LES09000
C     CALCULATE MINGAPS FOR DISTINCT T(1,MEV) EIGENVALUES.              LES09010
      NM1 = NDIS - 1                                                     LES09020
      G(NDIS) = VS(NM1)-VS(NDIS)                                         LES09030
      G(1) = VS(2)-VS(1)                                                 LES09040
C                                                                       LES09050
      DO 10 J = 2,NM1                                                    LES09060
      T0 = VS(J)-VS(J-1)                                                 LES09070
      T1 = VS(J+1)-VS(J)                                                 LES09080
      G(J) = T1                                                          LES09090
      IF (T0.LT.T1) G(J) = -T0                                           LES09100
   10 CONTINUE                                                           LES09110
C                                                                       LES09120
C     SET MP(I)=-1 FOR SIMPLE GOOD T-EIGENVALUES WHOSE MINGAPS  ARE     LES09130
C     'TOO SMALL' AND DUE TO SPURIOUS T-EIGENVALUES.                    LES09140
C                                                                       LES09150
      NISO = 0                                                           LES09160
      NG = 0                                                             LES09170
      DO 20 J = 1,NDIS                                                   LES09180
      IF (MP(J).EQ.0) GO TO 20                                           LES09190
      NG = NG+1                                                          LES09200
      IF (MP(J).NE.1) GO TO 20                                           LES09210
C     VS(J) IS NEXT SIMPLE GOOD T-EIGENVALUE                            LES09220
      NISO = NISO + 1                                                    LES09230
```

```
      I = J+1                                                LES09240
      IF (G(J).LT.0.0) I = J-1                               LES09250
      IF (MP(I).NE.0) GO TO 20                               LES09260
      GAP = ABS(G(J))                                        LES09270
      TO = DMAX1(SCALE1*MULTOL,GAPTOL*DABS(VS(J)))           LES09280
C     TO = DMAX1(GAPTOL,GAPTOL*DABS(VS(J)))                  LES09290
      TEMP = TO                                              LES09300
      IF (GAP.GT.TEMP) GO TO 20                              LES09310
      MP(J) = -MP(J)                                         LES09320
      NISO = NISO-1                                          LES09330
   20 CONTINUE                                               LES09340
C                                                            LES09350
C-----END OF ISOEV-------------------------------------------LES09360
      RETURN                                                 LES09370
      END                                                    LES09380
C                                                            LES09390
C-----START OF PRTEST----------------------------------------LES09400
C                                                            LES09410
      SUBROUTINE PRTEST(ALPHA,BETA,TEIG,TKMAX,EPSM,RELTOL,SCALE3,SCALE4,LES09420
     1 TMULT,NDIST,MEV,IPROJ)                                LES09430
C                                                            LES09440
C-----------------------------------------------------------LES09450
      DOUBLE PRECISION  ALPHA(1), BETA(1),TEIG(*),SIGMA(10)  LES09460
      DOUBLE PRECISION  EPSM,RELTOL,PRTOL,TKMAX,LRATIO,URATIO LES09470
      DOUBLE PRECISION  EPS,EPS1,BETAM,LBD,UBD,SIG,YU,YV,LRATS,URATS LES09480
      DOUBLE PRECISION ZERO,ONE,TEN,BISTOL,SCALE3,SCALE4,AEV,TEMP LES09490
      INTEGER  TMULT(*),ISIGMA(10)                           LES09500
      DOUBLE PRECISION  DABS, DMAX1, DSQRT, DFLOAT           LES09510
C-----------------------------------------------------------LES09520
C     AFTER CONVERGENCE HAS BEEN ESTABLISHED, SUBROUTINE PRTEST LES09530
C     TESTS COMPUTED EIGENVALUES OF T(1,MEV) THAT HAVE BEEN LABELLED LES09540
C     SPURIOUS TO DETERMINE IF ANY EIGENVALUES OF A HAVE BEEN LES09550
C     MISSED BY LANCZOS PROCEDURE.  AN EIGENVALUE WITH A VERY SMALL LES09560
C     PROJECTION ON THE STARTING VECTOR (< SINGLE PRECISION) LES09570
C     CAN BE MISSED BECAUSE IT IS ALSO AN EIGENVALUE OF T(2,MEV) TO LES09580
C     WITHIN THE SQUARE OF THIS ORIGINAL PROJECTION.        LES09590
C     OUR EXPERIENCE IS THAT SUCH SMALL PROJECTIONS OCCUR ONLY LES09600
C     VERY INFREQUENTLY.                                    LES09610
C                                                            LES09620
C     THIS SUBROUTINE IS CALLED ONLY AFTER CONVERGENCE HAS BEEN LES09630
C     ESTABLISHED. ONCE CONVERGENCE HAS BEEN OBSERVED ON THE LES09640
C     OTHER EIGENVALUES THEN ONE CAN EXPECT TO ALSO HAVE CONVERGENCE LES09650
C     ON ANY SUCH HIDDEN EIGENVALUES.(IF THERE ARE ANY).  THIS LES09660
C     PROCEDURE CONSIDERS ONLY SPURIOUS T-EIGENVALUES AND ONLY THOSE LES09670
C     SPURIOUS T-EIGENVALUES THAT ARE ISOLATED FROM GOOD T-EIGENVALUES. LES09680
C     FOR EACH SUCH T-EIGENVALUE IT DOES 2 STURM SEQUENCES  LES09690
C     AND A FEW SCALAR MULTIPLICATIONS.  UPON RETURN TO MAIN LES09700
C     PROGRAM ERROR ESTIMATES WILL BE COMPUTED FOR ANY EIGENVALUES LES09710
C     THAT HAVE BEEN LABELLED AS 'HIDDEN'.  SUCH T-EIGENVALUES LES09720
C     WILL BE RELABELLED AS 'GOOD' ONLY IF THESE ERROR ESTIMATES LES09730
C     ARE SUFFICIENTLY SMALL.                               LES09740
C-----------------------------------------------------------LES09750
      ZERO = 0.0D0                                           LES09760
      ONE  = 1.0D0                                           LES09770
      TEN  = 10.0D0                                          LES09780
```

```
      PRTOL = 1.D-6                                           LES09790
      TEMP = DFLOAT(MEV+1000)                                 LES09800
      TEMP = DSQRT(TEMP)                                      LES09810
      BISTOL = TKMAX*EPSM*TEMP                                LES09820
      NSIGMA = 4                                              LES09830
      SIGMA(1) = TEN*TKMAX                                    LES09840
C                                                             LES09850
      DO 10 J = 2,NSIGMA                                      LES09860
   10 SIGMA(J) = TEN*SIGMA(J-1)                               LES09870
C                                                             LES09880
      IFIN = 0                                                LES09890
      MF = 1                                                  LES09900
      ML = MEV                                                LES09910
      BETAM = BETA(MF)                                        LES09920
      BETA(MF) = ZERO                                         LES09930
      IPROJ = 0                                               LES09940
      J = 1                                                   LES09950
C                                                             LES09960
      IF (TMULT(1).NE.0) GO TO 110                            LES09970
C                                                             LES09980
      AEV = DABS(TEIG(1))                                     LES09990
      TEMP = PRTOL*AEV                                        LES10000
      EPS1 = DMAX1(TEMP,SCALE4*BISTOL)                        LES10010
C     EPS1 = DMAX1(TEMP,PRTOL)                                LES10020
      TEMP = RELTOL*AEV                                       LES10030
      EPS  = DMAX1(TEMP,SCALE3*BISTOL)                        LES10040
C     EPS  = DMAX1(TEMP,RELTOL)                               LES10050
C                                                             LES10060
      IF (TEIG(2)-TEIG(1).LT.EPS1.AND.TMULT(2).NE.0) GO TO 110 LES10070
C                                                             LES10080
   20 LBD = TEIG(J) - EPS                                     LES10090
      UBD = TEIG(J) + EPS                                     LES10100
      MEVL = 0                                                LES10110
      IL = 0                                                  LES10120
      YU = ONE                                                LES10130
C                                                             LES10140
      DO 50 I=MF,ML                                           LES10150
      IF (YU.NE.ZERO) GO TO 30                                LES10160
      YV = BETA(I)/EPSM                                       LES10170
      GO TO 40                                                LES10180
   30 YV = BETA(I)*BETA(I)/YU                                 LES10190
   40 YU = ALPHA(I)-LBD-YV                                    LES10200
      IF (YU.GE.ZERO) GO TO 50                                LES10210
C     MEVL INCREMENTED                                        LES10220
      MEVL = MEVL + 1                                         LES10230
      IL = I                                                  LES10240
   50 CONTINUE                                                LES10250
C                                                             LES10260
      LRATIO = YU                                             LES10270
      MEV1L = MEVL                                            LES10280
      IF (IL.EQ.ML) MEV1L=MEVL-1                              LES10290
C                                                             LES10300
C     MEVL = NUMBER OF EVS OF T(1,MEV) WHICH ARE < LBD        LES10310
C     MEV1L = NUMBER OF EVS OF T(1,MEV-1) WHICH ARE < LBD     LES10320
C     LRATIO = DET(T(1,MEV)-LBD)/DET(T(1,MEV-1)-LBD):         LES10330
```

```
C                                                       LES10340
      MEVU = 0                                          LES10350
      IL = 0                                            LES10360
      YU = ONE                                          LES10370
C                                                       LES10380
      DO 80 I=MF,ML                                     LES10390
      IF (YU.NE.ZERO) GO TO 60                          LES10400
      YV = BETA(I)/EPSM                                 LES10410
      GO TO 70                                          LES10420
   60 YV = BETA(I)*BETA(I)/YU                           LES10430
   70 YU = ALPHA(I)-UBD-YV                              LES10440
      IF (YU.GE.ZERO) GO TO 80                          LES10450
C     MEVU INCREMENTED                                  LES10460
      MEVU = MEVU + 1                                   LES10470
      IL = I                                            LES10480
   80 CONTINUE                                          LES10490
C                                                       LES10500
      URATIO = YU                                       LES10510
      MEV1U = MEVU                                      LES10520
      IF (IL.EQ.ML) MEV1U=MEVU-1                        LES10530
C                                                       LES10540
C     MEVU = NUMBER OF EVS OF T(MEV) WHICH ARE < UBD    LES10550
C     MEV1U = NUMBER OF EVS OF T(MEV-1) WHICH ARE < UBD LES10560
C     URATIO = DET(TM-UBD)/DET(T(M-1)-UBD): TM=T(MF,ML) LES10570
C                                                       LES10580
      NEV1 = MEV1U-MEV1L                                LES10590
C                                                       LES10600
      DO 90 K=1,NSIGMA                                  LES10610
      SIG = SIGMA(K)                                    LES10620
      LRATS = LRATIO-SIG                                LES10630
      URATS = URATIO-SIG                                LES10640
C     NOTE THE INCREMENT IS ON NUMBER OF EVALUES OF T(M-1) LES10650
      MEVLS = MEV1L                                     LES10660
      IF (LRATS.LT.0.) MEVLS=MEV1L+1                    LES10670
      MEVUS = MEV1U                                     LES10680
      IF (URATS.LT.0.) MEVUS=MEV1U+1                    LES10690
      ISIGMA(K) = MEVUS - MEVLS                         LES10700
   90 CONTINUE                                          LES10710
C                                                       LES10720
      ICOUNT = 0                                        LES10730
      DO 100 K=1,NSIGMA                                 LES10740
  100 IF (ISIGMA(K).EQ.1) ICOUNT=ICOUNT + 1            LES10750
C                                                       LES10760
      IF (ICOUNT.LT.2.OR.NEV1.EQ.0) GO TO 110          LES10770
      TMULT(J) = -10                                    LES10780
      IPROJ=IPROJ+1                                     LES10790
C                                                       LES10800
  110 J=J+1                                             LES10810
C                                                       LES10820
      IF (J.GE.NDIST) GO TO 120                         LES10830
      IF (TMULT(J).NE.0) GO TO 110                      LES10840
C                                                       LES10850
      AEV = DABS(TEIG(J))                               LES10860
      TEMP = PRTOL*AEV                                  LES10870
      EPS1 = DMAX1(TEMP,SCALE4*BISTOL)                  LES10880
```

```
C     EPS1 = DMAX1(TEMP,PRTOL)                                   LES10890
      TEMP = RELTOL*AEV                                          LES10900
      EPS  = DMAX1(TEMP,SCALE3*BISTOL)                          LES10910
C     EPS  = DMAX1(TEMP,RELTOL)                                 LES10920
C                                                               LES10930
      IF (TEIG(J)-TEIG(J-1).LT.EPS1.AND.TMULT(J-1).NE.0) GO TO 110   LES10940
      IF (TEIG(J+1)-TEIG(J).LT.EPS1.AND.TMULT(J+1).NE.0) GO TO 110   LES10950
C                                                               LES10960
      GO TO 20                                                  LES10970
C                                                               LES10980
  120 IF (IFIN.EQ.1) GO TO 130                                  LES10990
      IF (TMULT(NDIST).NE.0) GO TO 130                          LES11000
C                                                               LES11010
      AEV = DABS(TEIG(NDIST))                                   LES11020
      TEMP = PRTOL*AEV                                          LES11030
      EPS1 = DMAX1(TEMP,SCALE4*BISTOL)                          LES11040
C     EPS1 = DMAX1(TEMP,PRTOL)                                  LES11050
      TEMP = RELTOL*AEV                                         LES11060
      EPS  = DMAX1(TEMP,SCALE3*BISTOL)                          LES11070
C     EPS  = DMAX1(TEMP,RELTOL)                                 LES11080
C                                                               LES11090
      NDIST1=NDIST -1                                           LES11100
      TEMP = TEIG(NDIST)-TEIG(NDIST1)                           LES11110
      IF (TEMP.LT.EPS1.AND.TMULT(NDIST1).NE.0) GO TO 130        LES11120
      IFIN = 1                                                  LES11130
C                                                               LES11140
      GO TO 20                                                  LES11150
C                                                               LES11160
  130 BETA(MF) = BETAM                                          LES11170
C                                                               LES11180
C-----END OF PRTEST----------------------------------------------LES11190
      RETURN                                                    LES11200
      END                                                       LES11210
C                                                               LES11220
C------START OF STURMI-------------------------------------------LES11230
C                                                               LES11240
      SUBROUTINE STURMI(ALPHA,BETA,X1,TOLN,EPSM,MMAX,MK1,MK2,IC,IWRITE) LES11250
C                                                               LES11260
C---------------------------------------------------------------LES11270
      DOUBLE PRECISION  ALPHA(1),BETA(1)                        LES11280
      DOUBLE PRECISION  EPSM,X1,TOLN,EVL,EVU,BETA2              LES11290
      DOUBLE PRECISION  U1,U2,V1,V2,ZERO,ONE                    LES11300
      INTEGER I,IC,ICD,IC0,IC1,IC2,MK1,MK2,MMAX                 LES11310
C---------------------------------------------------------------LES11320
C                                                               LES11330
C     FOR ANY EIGENVALUE OF A THAT HAS CONVERGED AS AN EIGENVALUE   LES11340
C     OF THE T-MATRICES THIS SUBROUTINE CALCULATES              LES11350
C     THE SMALLEST SIZE OF THE T-MATRIX, T(1,MK1) DEFINED       LES11360
C     BY THE ALPHA AND BETA ARRAYS SUCH THAT MK1.LE.MMAX        LES11370
C     AND THE INTERVAL (X1-TOLN,X1+TOLN) CONTAINS AT LEAST ONE  LES11380
C     EIGENVALUE OF T(1,MK1). IT ALSO CALCULATES MK2 <= MMAX    LES11390
C     AS THE SMALLEST SIZE T-MATRIX (IF ANY) SUCH THAT THIS INTERVAL   LES11400
C     CONTAINS AT LEAST TWO EIGENVALUES OF T(1,MK2).            LES11410
C     IF NO T-MATRIX OF ORDER < MMAX SATISFIES THIS REQUIREMENT LES11420
C     THEN MK2 IS SET EQUAL TO MMAX.  THE EIGENVECTOR PROGRAM   LES11430
```

```
C       USES THESE VALUES TO DETERMINE AN APPROPRIATE 1ST GUESS AT         LES11440
C       AN APPROPRIATE SIZE T-MATRIX FOR THE EIGENVALUE X1.                LES11450
C                                                                          LES11460
C       ON EXIT IC = NUMBER OF EIGENVALUES OF T(1,MK2) IN THIS INTERVAL    LES11470
C                                                                          LES11480
C       STURMI REGENERATES THE QUANTITIES BETA(I)**2 EACH TIME IT IS       LES11490
C       CALLED, OBVIOUSLY FOR THE PRICE OF ANOTHER VECTOR OF LENGTH        LES11500
C       MMAX THIS GENERATION COULD BE DONE ONCE IN THE MAIN               LES11510
C       PROGRAM BEFORE THE LOOP ON THE CALLS TO SUBROUTINE STURMI.         LES11520
C                                                                          LES11530
C       IF ANY OF THE EIGENVALUES BEING CONSIDERED WERE MULTIPLE           LES11540
C       AS EIGENVALUES OF THE USER-SPECIFIED MATRIX, THEN                  LES11550
C       THIS SUBROUTINE COULD BE MODIFIED TO COMPUTE ADDITIONAL            LES11560
C       SIZES MKJ, J = 3, ...  WHICH COULD THEN BE USED IN THE             LES11570
C       MAIN LANCZOS EIGENVECTOR PROGRAM TO COMPUTE ADDITIONAL             LES11580
C       EIGENVECTORS CORRESPONDING TO THESE MULTIPLE EIGENVALUES.          LES11590
C       THE MAIN PROGRAM PROVIDED DOES NOT INCLUDE THIS OPTION.            LES11600
C                                                                          LES11610
C-------------------------------------------------------------------------LES11620
C       INITIALIZATION OF PARAMETERS                                       LES11630
        MK1 = 0                                                            LES11640
        MK2 = 0                                                            LES11650
        ZERO = 0.0D0                                                       LES11660
        ONE  = 1.0D0                                                       LES11670
        BETA(1) = ZERO                                                     LES11680
        EVL = X1-TOLN                                                      LES11690
        EVU = X1+TOLN                                                      LES11700
        U1 = ONE                                                           LES11710
        U2 = ONE                                                           LES11720
        IC0 = 0                                                            LES11730
        IC1 = 0                                                            LES11740
        IC2 = 0                                                            LES11750
C                                                                          LES11760
C       MAIN LOOP FOR CALCULATING THE SIZES MK1,MK2                        LES11770
        DO 60 I = 1,MMAX                                                   LES11780
        BETA2 = BETA(I)*BETA(I)                                            LES11790
        IF (U1.NE.ZERO) GO TO 10                                           LES11800
        V1 = BETA(I)/EPSM                                                  LES11810
        GO TO 20                                                           LES11820
   10 V1 = BETA2/U1                                                        LES11830
   20 U1 = EVL - ALPHA(I) - V1                                            LES11840
        IF (U1.LT.ZERO) IC1 = IC1+1                                        LES11850
        IF (U2.NE.ZERO) GO TO 30                                           LES11860
        V2 = BETA(I)/EPSM                                                  LES11870
        GO TO 40                                                           LES11880
   30 V2 = BETA2/U2                                                        LES11890
   40 U2 = EVU - ALPHA(I) - V2                                            LES11900
        IF (U2.LT.ZERO) IC2 = IC2+1                                        LES11910
C       TEST FOR CHANGE IN NUMBER OF T-EIGENVALUES ON (EVL,EVU)            LES11920
        ICD = IC1-IC2                                                      LES11930
        IC = ICD-IC0                                                       LES11940
        IF (IC.GE.1) GO TO 50                                              LES11950
        GO TO 60                                                          LES11960
   50 CONTINUE                                                             LES11970
        IF (IC0.EQ.0) MK1 = I                                             LES11980
```

```
        ICO = ICO+1                                              LES11990
        IF (ICO.GT.1) GO TO 70                                   LES12000
   60 CONTINUE                                                   LES12010
C                                                                LES12020
        I = I-1                                                  LES12030
        IF (ICO.EQ.0) MK1 = MMAX                                 LES12040
   70 MK2 = I                                                    LES12050
        IC = ICD                                                 LES12060
C                                                                LES12070
        IF (IWRITE.EQ.1) WRITE(6,80) X1,MK1,MK2,IC               LES12080
   80 FORMAT(' EVAL =',E20.12,' MK1 =',I6,' MK2 =',I6,' IC =',I3/) LES12090
C                                                                LES12100
        RETURN                                                   LES12110
C-----END OF STURMI----------------------------------------------LES12120
        END                                                      LES12130
C                                                                LES12140
C                                                                LES12150
C-----START OF INVERM--------------------------------------------LES12160
C                                                                LES12170
        SUBROUTINE INVERM(ALPHA,BETA,V1,V2,X1,ERROR,ERRORV,EPS,G,MEV,IT, LES12180
      1 IWRITE)                                                  LES12190
C                                                                LES12200
C---------------------------------------------------------------LES12210
        DOUBLE PRECISION  ALPHA(1),BETA(*),V1(1),V2(*)           LES12220
        DOUBLE PRECISION  X1,U,Z,TEMP,RATIO,SUM,XU,NORM,TSUM,BETAM LES12230
        DOUBLE PRECISION  EPS,EPS3,EPS4,ERROR,ERRORV,ZERO,ONE    LES12240
        REAL  G(1)                                               LES12250
        DOUBLE PRECISION  DABS, DSQRT, DFLOAT                    LES12260
        DOUBLE PRECISION FINPRO                                  LES12270
        REAL  ABS                                                LES12280
C---------------------------------------------------------------LES12290
C                                                                LES12300
C     COMPUTES T-EIGENVECTORS FOR ISOLATED GOOD T-EIGENVALUES X1 LES12310
C     USING INVERSE ITERATION ON T(1,MEV(X1)) SOLVING EQUATION   LES12320
C     (T - X1*I)V2 = RIGHT-HAND SIDE (RANDOMLY-GENERATED) .      LES12330
C     PROGRAM REFACTORS T- X1*I ON EACH ITERATION OF INVERSE ITERATION. LES12340
C     TYPICALLY ONLY ONE ITERATION IS NEEDED PER T-EIGENVALUE X1. LES12350
C                                                                LES12360
C     IF IWRITE = 1 THEN THERE ARE EXTENDED WRITES TO FILE 6 (TERMINAL) LES12370
C                                                                LES12380
C     ON ENTRY G CONTAINS A REAL*4 RANDOM VECTOR WHICH WAS GENERATED LES12390
C     IN MAIN PROGRAM.                                           LES12400
C                                                                LES12410
C     ON ENTRY AND EXIT                                          LES12420
C     MEV = ORDER OF T                                           LES12430
C     ALPHA, BETA CONTAIN THE DIAGONAL AND OFFDIAGONAL ENTRIES OF T. LES12440
C     EPS = 2. * MACHINE EPSILON                                 LES12450
C                                                                LES12460
C     IN PROGRAM:                                                LES12470
C     ITER = MAXIMUM NUMBER STEPS ALLOWED FOR INVERSE ITERATION  LES12480
C     ITER = IT ON ENTRY.                                        LES12490
C     V1,V2 = WORK SPACES USED IN THE FACTORIZATION OF T(1,MEV). LES12500
C     V1 AND V2 MUST BE OF DIMENSION AT LEAST MEV.               LES12510
C                                                                LES12520
C     ON EXIT                                                    LES12530
```

```
C     V2 = THE UNIT EIGENVECTOR OF T(1,MEV) CORRESPONDING TO X1.      LES12540
C     ERROR = |V2(MEV)| = ERROR ESTIMATE FOR CORRESPONDING            LES12550
C           RITZ VECTOR FOR X1.                                       LES12560
C                                                                     LES12570
C     ERRORV = || T*V2 - X1*V2 || = ERROR ESTIMATE ON T-EIGENVECTOR.  LES12580
C     IF IT.GT.ITER THEN ERRORV = -ERRORV                             LES12590
C     IT = NUMBER OF ITERATIONS ACTUALLY REQUIRED                     LES12600
C---------------------------------------------------------------------LES12610
C     INITIALIZATION AND PARAMETER SPECIFICATION                      LES12620
      ONE  = 1.0D0                                                    LES12630
      ZERO = 0.0D0                                                    LES12640
      ITER = IT                                                       LES12650
      MP1 = MEV+1                                                     LES12660
      MM1 = MEV-1                                                     LES12670
      BETAM = BETA(MP1)                                               LES12680
      BETA(MP1) = ZERO                                                LES12690
C                                                                     LES12700
C     CALCULATE SCALE AND TOLERANCES                                  LES12710
      TSUM = DABS(ALPHA(1))                                           LES12720
      DO 10 I = 2,MEV                                                 LES12730
   10 TSUM = TSUM + DABS(ALPHA(I)) + BETA(I)                          LES12740
C                                                                     LES12750
      EPS3 = EPS*TSUM                                                 LES12760
      EPS4 = DFLOAT(MEV)*EPS3                                         LES12770
C                                                                     LES12780
C     GENERATE SCALED RANDOM RIGHT-HAND SIDE                          LES12790
      GSUM = ZERO                                                     LES12800
      DO 20 I = 1,MEV                                                 LES12810
   20 GSUM = GSUM+ABS(G(I))                                           LES12820
      GSUM = EPS4/GSUM                                                LES12830
C                                                                     LES12840
C     INITIALIZE RIGHT HAND SIDE FOR INVERSE ITERATION               LES12850
      DO 30 I = 1,MEV                                                 LES12860
   30 V2(I) = GSUM*G(I)                                               LES12870
      IT = 1                                                          LES12880
C                                                                     LES12890
C     CALCULATE UNIT EIGENVECTOR OF T(1,MEV) FOR ISOLATED GOOD        LES12900
C     T-EIGENVALUE X1.                                                LES12910
C                                                                     LES12920
C     TRIANGULAR FACTORIZATION WITH NEAREST NEIGHBOR PIVOT            LES12930
C     STRATEGY. INTERCHANGES ARE LABELLED BY SETTING BETA < 0.        LES12940
C                                                                     LES12950
   40 CONTINUE                                                        LES12960
      U = ALPHA(1)-X1                                                 LES12970
      Z = BETA(2)                                                     LES12980
C                                                                     LES12990
      DO 60 I=2,MEV                                                   LES13000
      IF (BETA(I).GT.DABS(U)) GO TO 50                                LES13010
C     NO PIVOT INTERCHANGE                                            LES13020
      V1(I-1) = Z/U                                                   LES13030
      V2(I-1) = V2(I-1)/U                                             LES13040
      V2(I) = V2(I)-BETA(I)*V2(I-1)                                   LES13050
      RATIO = BETA(I)/U                                               LES13060
      U = ALPHA(I)-X1-Z*RATIO                                         LES13070
      Z = BETA(I+1)                                                   LES13080
```

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

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

```
C                                                                      LES14190
C     EP1 = DSQRT(1000+M)*TTOL     TTOL = EPSM*TKMAX                    LES14200
C     TKMAX = MAX(|ALPHA(K)|,BETA(K), K= 1,KMAX)                       LES14210
C                                                                      LES14220
      TEMP = DFLOAT(1000+M)                                            LES14230
      EP1 = DSQRT(TEMP)*TTOL                                           LES14240
C                                                                      LES14250
      NA = 0                                                           LES14260
      X1 = XU                                                          LES14270
      JSTURM = 1                                                       LES14280
      GO TO 60                                                         LES14290
C     FORWARD STURM CALCULATION                                        LES14300
   10 NA = NEV                                                         LES14310
      X1 = XL                                                          LES14320
      JSTURM = 2                                                       LES14330
      GO TO 60                                                         LES14340
C     FORWARD STURM CALCULATION                                        LES14350
   20 NEVT = NEV                                                       LES14360
C                                                                      LES14370
C     WRITE(6,30) M,EVAL,NEVT,EP1                                      LES14380
   30 FORMAT(/3X,'TSIZE',23X,'EV',9X/I8,E25.16/                        LES14390
     1 I6,' = NUMBER OF T(1,M) EIGENVALUES ON TEST INTERVAL'/          LES14400
     1 E12.3,' =  CONVERGENCE TOLERANCE'/)                             LES14410
C                                                                      LES14420
      IF (NEVT.NE.1) GO TO 120                                         LES14430
C                                                                      LES14440
C     BISECTION LOOP                                                   LES14450
      JSTURM = 3                                                       LES14460
   40 X1 = HALF*(XL+XU)                                                LES14470
      X0 = XU-XL                                                       LES14480
      EPT = EPSM*(DABS(XL) + DABS(XU)) + EP1                           LES14490
C     CONVERGENCE TEST                                                 LES14500
      IF (X0.LE.EP1) GO TO 100                                         LES14510
      GO TO 60                                                         LES14520
C     FORWARD STURM CALCULATION                                        LES14530
   50 CONTINUE                                                         LES14540
      IF(NEV.EQ.0) XU = X1                                             LES14550
      IF(NEV.EQ.1) XL = X1                                             LES14560
      GO TO 40                                                         LES14570
C     NEV = NUMBER OF T-EIGENVALUES OF T(1,M) ON (X1,XU)               LES14580
C     THERE IS EXACTLY ONE T-EIGENVALUE OF T(1,M) ON (XL,XU)           LES14590
C                                                                      LES14600
C     FORWARD STURM CALCULATION                                        LES14610
   60 NEV = -NA                                                        LES14620
      YU = ONE                                                         LES14630
      DO 90 I = 1,M                                                    LES14640
      IF (YU.NE.ZERO) GO TO 70                                         LES14650
      YV = BETA(I)/EPSM                                                LES14660
      GO TO 80                                                         LES14670
   70 YV = BETA(I)*BETA(I)/YU                                          LES14680
   80 YU = X1 - ALPHA(I) - YV                                          LES14690
      IF (YU.GE.ZERO) GO TO 90                                         LES14700
      NEV = NEV+1                                                      LES14710
   90 CONTINUE                                                         LES14720
      GO TO (10,20,50), JSTURM                                         LES14730
```

```
C                                                                 LES14740
  100 CONTINUE                                                     LES14750
C                                                                 LES14760
      EVALN = X1                                                  LES14770
      EVD = DABS(EVALN-EVAL)                                      LES14780
C     WRITE(6,110) EVALN,EVAL,EVD                                 LES14790
  110 FORMAT(/20X,'EVALN',21X,'EVAL',6X,'CHANGE'/2E25.16,E12.3/)  LES14800
C                                                                 LES14810
  120 CONTINUE                                                     LES14820
      RETURN                                                      LES14830
C-----END OF LBISEC-------------------------------------------------LES14840
      END                                                         LES14850
C-----START OF COMPLEX INNER PRODUCT-------------------------------LES14860
C                                                                 LES14870
C     COMPLEX INNER PRODUCT                                       LES14880
C                                                                 LES14890
      SUBROUTINE CINPRD(V2,V1,SUM,N)                              LES14900
C-----------------------------------------------------------------LES14910
      DOUBLE PRECISION  ZERO,SUM                                  LES14920
      COMPLEX*16 V2(1),V1(1),SUMC                                 LES14930
C-----------------------------------------------------------------LES14940
C                                                                 LES14950
C     NOTE THAT THE ORDER MATTERS HERE                            LES14960
C     COMPUTES THE INNER PRODUCT OF THE CONJUGATE OF V2 WITH V1.  LES14970
      ZERO = 0.D0                                                 LES14980
      SUMC = DCMPLX(ZERO,ZERO)                                    LES14990
      DO 10 J=1,N                                                 LES15000
   10 SUMC = SUMC + DCONJG(V2(J))*V1(J)                           LES15010
      SUM = DREAL(SUMC)                                           LES15020
C                                                                 LES15030
      RETURN                                                      LES15040
C-----END OF COMPLEX INNER PRODUCT SUBROUTINE---------------------LES15050
      END                                                         LES15060
C                                                                 LES15070
C-----LPERM-PERMUTES VECTORS--------------------------------------LES15080
C                                                                 LES15090
      SUBROUTINE LPERM(W,U,IPERM)                                 LES15100
C                                                                 LES15110
C-----------------------------------------------------------------LES15120
      DOUBLE PRECISION  U(1),W(1)                                 LES15130
      INTEGER  IPR(1),IPT(1)                                      LES15140
C-----------------------------------------------------------------LES15150
C     SUBROUTINE HAS 2 BRANCHES:  IPERM = 1,  CALCULATES          LES15160
C     U = P*W                                                     LES15170
C     LET J = IPR(K) THEN U(K) = W(J), K = 1,N.                   LES15180
C     IPERM = 2, CALCULATES                                       LES15190
C     U = P'*W  LET J = IPT(K) THEN U(K) = W(J), K=1,N.           LES15200
C-----------------------------------------------------------------LES15210
      GO TO 3                                                     LES15220
      ENTRY LPERME(IPR,IPT,N)                                     LES15230
      GO TO 4                                                     LES15240
C-----------------------------------------------------------------LES15250
C                                                                 LES15260
    3 IF(IPERM.EQ.2) GO TO 30                                     LES15270
C                                                                 LES15280
```

```
C----------------------------------------------------------------LES15290
C     IPERM = 1                                                  LES15300
      DO 10 K = 1,N                                              LES15310
      J = IPR(K)                                                 LES15320
   10 U(K) = W(J)                                                LES15330
      GO TO 60                                                   LES15340
C--------------------------------------------------------------- LES15350
C     IPERM = 2                                                  LES15360
   30 DO 40 K = 1,N                                              LES15370
      J = IPT(K)                                                 LES15380
   40 U(K) = W(J)                                                LES15390
C--------------------------------------------------------------- LES15400
   60 CONTINUE                                                   LES15410
      DO 50 K = 1,N                                              LES15420
   50 W(K) = U(K)                                                LES15430
C                                                                LES15440
    4 RETURN                                                     LES15450
C                                                                LES15460
C-----END OF LPERM------------------------------------------     LES15470
      END                                                        LES15480
```

2.7   LECOMPAC: Optional Preprocessing Program

```
C-----LECOMPAC-(STAND-ALONE PROGRAM)------------------------------------LEC00010
C  AUTHOR:   RALPH A. WILLOUGHBY (Deceased)                  LEC00020
C                                                            LEC00030
C                                                            LEC00040
C                                                            LEC00050
C             E-mail:  cullumj@lanl.gov                      LEC00060
C                                                            LEC00070
C  These codes are copyrighted by the authors.  These codes  LEC00080
C  and modifications of them or portions of them are NOT to be LEC00090
C  incorporated into any commercial codes or used for any other LEC00100
C  commercial purposes such as consulting for other companies, LEC00110
C  without legal agreements with the authors of these Codes.  LEC00120
C  If these Codes or portions of them are used in other scientific or LEC00130
C  engineering research works the names of the authors of these codes LEC00140
C  and appropriate references to their written work are to be  LEC00150
C  incorporated in the derivative works.                     LEC00160
C                                                            LEC00170
C  This header is not to be removed from these codes.         LEC00180
C                                                            LEC00190
C                                                            LEC00200
C     THIS PROGRAM TRANSLATES A SPARSE SYMMETRIC N X N MATRIX A,  LEC00210
C     GIVEN AS I, J, A(I,J), INTO THE SPARSE MATRIX FORMAT     LEC00220
C     REQUIRED BY THE SAMPLE USPEC AND CMATV PROGRAMS PROVIDED  LEC00230
C     FOR USE WITH THE LANCZOS EIGENVALUE/EIGENVECTOR PROCEDURES. LEC00240
C     THIS PROGRAM ASSUMES THAT THE MATRIX ENTRIES ARE PROVIDED  LEC00250
C     EITHER COLUMN BY COLUMN OR ROW BY ROW.                  LEC00260
C     NOTE THAT THIS PROGRAM DOES NOT DIRECTLY APPLY TO THE   LEC00270
C     HERMITIAN CASE BECAUSE FOR HERMITIAN MATRICES THE DIAGONALS LEC00280
C     ARE REAL AND THE OFF-DIAGONAL ENTRIES ARE COMPLEX VARIABLES. LEC00290
C                                                            LEC00300
C     NONPORTABLE STATEMENTS:  PFORT VERIFIER INDICATES THAT THIS  LEC00310
C                              IS PORTABLE.                   LEC00320
C                                                            LEC00330
C----------------------------------------------------------------------LEC00340
      DOUBLE PRECISION A(15000), AD(2000)                   LEC00350
      DOUBLE PRECISION ZERO                                 LEC00360
      INTEGER IROW(15000),ICOL(15000)                       LEC00370
C----------------------------------------------------------------------LEC00380
C   INPUT FILE 7 CONTAINS THE SPARSE SYMMETRIC NXN MATRIX STORED AS: LEC00390
C                                                            LEC00400
C             NZ,M,N,MATNO                                   LEC00410
C             I(K)  J(K)  A(K)  K = 1,NZ                     LEC00420
C                                                            LEC00430
C     WHERE NZ IS THE TOTAL NUMBER OF NONZEROS IN THE MATRIX A,  LEC00440
C     N IS THE ROW AND COLUMN DIMENSION OF A,                LEC00450
C     AND A(K) ARE THE NONZERO ENTRIES STORED ROW BY ROW OR  LEC00460
C     COLUMN BY COLUMN.  PROGRAM READS THIS IN AS IROW(K) = I(K),  LEC00470
C     ICOL(K) = J(K), AND A(K) = A(K).                       LEC00480
C                                                            LEC00490
C     OUTPUT FILE = 8 CONTAINS THE A-MATRIX IN SPARSE FORMAT  LEC00500
C                                                            LEC00510
C             NZS,N,NZL,MATNO                                LEC00520
```

```
C                ICOL(K)    K = 1,NZL                              LEC00530
C                IROW(K)    K = 1,NZS                              LEC00540
C                  AD(K)    K = 1,N                                LEC00550
C                   A(K)    K = 1,NZS                              LEC00560
C                                                                 LEC00570
C     WHERE N IS THE ORDER OF THE INPUT MATRIX A,                 LEC00580
C     NZ EQUALS THE NUMBER OF NONZERO ELEMENTS IN A WHICH ARE ON  LEC00590
C     OR BELOW THE MAIN DIAGONAL.  NZL EQUALS THE NUMBER OF THE   LEC00600
C     LAST COLUMN HAVING NONZEROES BELOW THE DIAGONAL IN A.       LEC00610
C     NZS EQUALS THE NUMBER OF NONZERO ELEMENTS BELOW THE MAIN    LEC00620
C     DIAGONAL.  AD(K), K=1,N, CONTAINS THE DIAGONAL ELEMENTS OF A. LEC00630
C     A(K), K=1,NZS, CONTAINS THE KTH NONZERO SUB-DIAGONAL ELEMENT LEC00640
C     OF THE INPUT MATRIX.  A IS STORED COLUMN BY COLUMN.         LEC00650
C     IROW(K), K=1,NZS, CONTAINS THE ROW INDEX OF THE  NONZERO    LEC00660
C     STRICTLY LOWER TRIANGULAR ELEMENT A(K).                     LEC00670
C     ICOL(K), K=1,NZL, EQUALS THE NUMBER OF STRICTLY LOWER       LEC00680
C     TRIANGULAR NONZEROES IN COLUMN K OF THE INPUT MATRIX.       LEC00690
C                                                                 LEC00700
C-----------------------------------------------------------------LEC00710
      ZERO = 0.0D0                                                LEC00720
C                                                                 LEC00730
      READ(7,10) NZ,N,MATNO,IIROW                                 LEC00740
   10 FORMAT(2I6,I8,I4)                                           LEC00750
C                                                                 LEC00760
      WRITE(6,20) NZ,N,MATNO,IIROW                                LEC00770
   20 FORMAT(I10,I6,I10,' = NO. NONZERO AIJ J.GE.I, ORDER OF A, MATNO'/ LEC00780
     1 I6,' = IIROW IF IIROW=0 ORDERING IS BY COLS IIROW=1 BY ROWS'/) LEC00790
C                                                                 LEC00800
      DO 30 K = 1,N                                               LEC00810
   30 AD(K) = ZERO                                                LEC00820
C                                                                 LEC00830
      IF (IIROW.EQ.0) READ(7,40) (IROW(K),ICOL(K),A(K), K=1,NZ)   LEC00840
C                                                                 LEC00850
      IF (IIROW.EQ.1) READ(7,40) (ICOL(K),IROW(K),A(K), K=1,NZ)   LEC00860
   40 FORMAT(2I5,E14.7)                                           LEC00870
C                                                                 LEC00880
      LCOUNT = 0                                                  LEC00890
      K = 1                                                       LEC00900
C                                                                 LEC00910
C     START OF A NEW COLUMN                                       LEC00920
   50 CONTINUE                                                    LEC00930
      J = ICOL(K)                                                 LEC00940
      ICOL(J) = 0                                                 LEC00950
   60 CONTINUE                                                    LEC00960
C                                                                 LEC00970
      IF (J.NE.IROW(K)) GO TO 70                                  LEC00980
C                                                                 LEC00990
C     DIAGONAL CASE                                               LEC01000
      AD(J) = A(K)                                                LEC01010
      GO TO 80                                                    LEC01020
C                                                                 LEC01030
C     SUB-DIAGONAL NONZERO                                        LEC01040
   70 CONTINUE                                                    LEC01050
      NZL = J                                                     LEC01060
      LCOUNT = LCOUNT + 1                                         LEC01070
```

```
      A(LCOUNT) = A(K)                                         LEC01080
      IROW(LCOUNT) = IROW(K)                                   LEC01090
      ICOL(J) = ICOL(J) + 1                                    LEC01100
C                                                              LEC01110
   80 CONTINUE                                                 LEC01120
      K = K+1                                                  LEC01130
C                                                              LEC01140
      IF(K.GT.NZ) GO TO 90                                     LEC01150
C                                                              LEC01160
      IF(ICOL(K).GT.J) GO TO 50                                LEC01170
C                                                              LEC01180
      GO TO 60                                                 LEC01190
C                                                              LEC01200
   90 CONTINUE                                                 LEC01210
      NZS = LCOUNT                                             LEC01220
C                                                              LEC01230
      WRITE(8,100) NZS,N,NZL,MATNO                             LEC01240
      WRITE(6,100) NZS,N,NZL,MATNO                             LEC01250
  100 FORMAT(I10,2I6,I8,' = NZS N NZL MATNO')                  LEC01260
C                                                              LEC01270
      WRITE(8,110) (ICOL(I), I=1,NZL)                          LEC01280
      WRITE(8,110) (IROW(K), K=1,NZS)                          LEC01290
  110 FORMAT(13I6)                                             LEC01300
C                                                              LEC01310
      WRITE(8,120) (AD(K), K=1,N)                              LEC01320
      WRITE(8,120) (A(K), K=1,NZS)                             LEC01330
  120 FORMAT(4E19.10)                                          LEC01340
C                                                              LEC01350
C-----END LECOMPAC-------------------------------------------------LEC01360
      STOP                                                     LEC01370
      END                                                      LEC01380
```

2.8   LEVAL: LEVEC: File Definitions, Sample Input Files

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

```
Sample Specifications of the Input/Output Files for LEVAL
----------------------------------------------------------------------
 LEVAL EXEC LANCZOS EIGENVALUE CALCULATION REAL SYMMETRIC MATRICES
FI 06 TERM
FILEDEF  1 DISK &1        NHISTORY  A (RECFM F LRECL 80 BLOCK 80
FILEDEF  2 DISK &1        HISTORY   A (RECFM F LRECL 80 BLOCK 80
FILEDEF  3 DISK &1        GOODEV    A (RECFM F LRECL 80 BLOCK 80
FILEDEF  4 DISK &1        ERRINV    A (RECFM F LRECL 80 BLOCK 80
FILEDEF  5 DISK LEVAL     INPUT     A (RECFM F LRECL 80 BLOCK 80
FILEDEF  8 DISK &1        INPUT     A (RECFM F LRECL 80 BLOCK 80
FILEDEF 11 DISK &1        DISTINCT  A (RECFM F LRECL 80 BLOCK 80
LOAD   LEVAL   LESUB   LEMULT
----------------------------------------------------------------
```

```
Sample Input File for LEVAL
-------------------------------------------------------------
 LANCZOS EIGENVALUE COMPUTATIONS, NO REORTHOGONALIZATION
 TEST MATRIX
LINE 1    N    KMAX    NMEVS    MATNO
         143    429       1    706830
LINE 2   SVSEED   RHSEED     MXINIT   MXSTUR
        7892713   147935        5    100000
LINE 3   ISTART    ISTOP
            0        1
LINE 4      IHIS     IDIST  IWRITE
             1        0       1
LINE 5   RELTOL (RELATIVE TOLERANCE IN 'COMBINING' GOODEV)
     .0000000001
LINE 6   MB(1)  MB(2)  MB(3)  MB(4) (ORDERS OF T(1,MEV))
          190
LINE 7   NINT    (NUMBER OF SUB-INTERVALS FOR BISEC)
           1
LINE 8   LB(1)   LB(2)   LB(3)  (INTERVAL LOWER BOUNDS)
          0.0
LINE 9   UB(1)   UB(2)   UB(3)  (INTERVAL UPPER BOUNDS)
         1.001
-------------------------------------------------------------
```

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

```
Sample Specifications of the Input/Output Files for LEVEC
----------------------------------------------------------------------
 LEVEC EXEC TO RUN LANCZOS EIGENVECTOR PROGRAM, REAL SYMMETRIC MATRICES
FI  06 TERM
FILEDEF  2 DISK &1       HISTORY   A (RECFM F LRECL 80 BLOCK 80
FILEDEF  3 DISK &1       GOODEV    A (RECFM F LRECL 80 BLOCK 80
FILEDEF  4 DISK &1       ERRINV    A (RECFM F LRECL 80 BLOCK 80
FILEDEF  5 DISK LEVEC    INPUT     A (RECFM F LRECL 80 BLOCK 80
FILEDEF  8 DISK &1       INPUT     A (RECFM F LRECL 80 BLOCK 80
FILEDEF  9 DISK &1       ERREST    A (RECFM F LRECL 80 BLOCK 80
FILEDEF 10 DISK &1       BOUNDS    A (RECFM F LRECL 80 BLOCK 80
FILEDEF 11 DISK &1       TEIGVECS  A (RECFM F LRECL 80 BLOCK 80
FILEDEF 12 DISK &1       RITZVECS  A (RECFM F LRECL 80 BLOCK 80
FILEDEF 13 DISK &1       PAIGE     A (RECFM F LRECL 80 BLOCK 80
LOAD  LEVEC  LESUB  LEMULT
----------------------------------------------------------------------
```

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

Chapter 3

Hermitian Matrices

3.1 Introduction

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

$$Ax = \lambda x. \qquad (3.1.1)$$

Definition 2 *A complex nxn matrix A, $A \equiv (a_{ij}), 1 \leq i,j \leq n$, is a Hermitian matrix if and only if for every i and j, $a_{ij} = \overline{a_{ji}}$, where the overbar denotes the complex conjugate of the complex-valued entry a_{ij}.*

It is straight-forward to demonstrate from Definition 2 that for any Hermitian matrix $A = B + Ci$, where B and C are real matrices and $i = \sqrt{-1}$, that B must be a real symmetric matrix and C must be a skew symmetric matrix. That is, $B^T = B$ and $C^T = -C$. Futhermore, it is not difficult to see that Hermitian matrices must have real diagonal entries and real eigenvalues. However, the eigenvectors are complex-valued. Any Hermitian matrix can be transformed into a real symmetric tridiagonal matrix for the purposes of computing the eigenvalues of the Hermitian matrix, Stewart [24]. In fact, the Lanczos recursion which we use in the codes in this chapter transforms the given Hermitian matrix A into a family of real symmetric tridiagonal matrices rather than into a family of Hermitian tridiagonal matrices.

Hermitian matrices possess the 'same' properties as real symmetric matrices do, except that these properties are defined with respect to the complex or Hermitian norm, rather than with respect to the Euclidean norm, see Stewart [24]. The Hermitian norm of a given complex-valued vector $x \equiv (x(i)), 1 \leq i \leq n$, is defined as $\|x\|_C^2 \equiv \sum_{i=1}^{n} \overline{x(i)}x(i)$. Three properties which we use are:

1. Hermitian matrices have complete eigensystems. That is, the dimension of the eigenspace corresponding to any given eigenvalue of a Hermitian matrix is the same as the multiplicity of that eigenvalue as a root of the characteristic polynomial of that matrix.

2. For any two distinct eigenvalues λ, μ and corresponding eigenvectors x, y, $x^H y = 0$, where the superscript H denotes the complex conjugate transpose of the vector x. The complex conjugate transpose of a column vector x is the row vector whose i^{th} component is $\overline{x(i)}$. There is a complete set of eigenvectors $X_n \equiv (x_1, \ldots, x_n)$ such that X is a unitary matrix.

3. Small Hermitian perturbations in a Hermitian matrix cause only small perturbations in the eigenvalues.

The single-vector Lanczos codes in this chapter can be used to compute either a very few or very many of the distinct eigenvalues of the given Hermitian matrix. The documentation for these codes is contained in Chapter 2, Section 2.2. As in the real symmetric case, the A-multiplicity of a given computed 'good' Lanczos eigenvalue can be obtained only with additional computation, and the modifications required to do this additional computation are not included in these versions of the codes. This implementation uses a Hermitian analog of the basic Lanczos recursion contained in Eqns (1.2.1) and (1.2.2) to generate a family of real symmetric tridiagonal matrices whose sizes are specified by the user. There is no reorthogonalization of the Lanczos vectors at any stage in any of the computations.

The Hermitian version of the Lanczos recursion which we use is given below. For $i = 1, 2, ..., m$ and a randomly-generated complex starting vector v_1 with $\|v_1\|_C = 1$, generate Lanczos vectors v_i using the following recursion.
$$\beta_{i+1} v_{i+1} = A v_i - \alpha_i v_i - \beta_i v_{i-1}, \tag{3.1.2}$$
where
$$\alpha_i \equiv v_i^H A v_i, \quad \beta_{i+1} = \|A v_i - \alpha_i v_i - \beta_i v_{i-1}\|_C \tag{3.1.3}$$

We see from Eqns(3.1.3) that the Hermitian inner product is used. This is the 'natural' inner product for Hermitian matrices. Gram-Schmidt orthogonalization is used, unlike the real symmetric case where a modified Gram-Schmidt orthogonalization was used. This change in the local orthogonalization procedure increases the storage requirements for the implementation of the Lanczos recursion by one additional complex vector of length equal to the order of the original A-matrix. Modified Gram-Schmidt orthogonalization cannot be used in the Hermitian case because corrections to the α_i defined by this modification are complex-valued not real, and it would not be legitimate to accept the real portions of these corrections and simply ignore the complex portions.

It is easy to demonstrate that as we stated earlier, each Lanczos matrix (T-matrix) generated by this Hermitian recursion is a real symmetric tridiagonal matrix. In particular, we see from the formulas in Eqn(3.1.3) that the diagonal entries of each of these matrices are Rayleigh quotients of the given Hermitian matrix A, and therefore must all be real-valued. Furthermore by construction, the nonzero off-diagonal entries β_{i+1} are all real-valued. This use of real-valued β_i requires some justification. This justification is given in Section 4.9 of Chapter 4 of Volume 1 of this book.

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

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

The computations in the Lanczos recursion are a mixture of double precision real arithmetic and of double precision complex arithmetic. Once the Lanczos matrices have been computed, the remaining

computations are all done in double precision real arithmetic, using the same subroutines that are used in the real symmetric case. In addition to the programs and subroutines provided here, the user must supply a subroutine USPEC which defines and initializes the user-specified matrix A and a subroutine CMATV which computes matrix-vector multiplies Ax for any given vector x. These subroutines must be constructed in such a way as to take advantage of the sparsity (and/or structure) of the user-supplied A-matrix and such that these computations are done accurately.

3.2   HLEVAL: Main Program, Eigenvalue Computations

```
C-----HLEVAL  (EIGENVALUES OF HERMITIAN MATRICES)---------------------HHL00010
C  Authors:  Jane Cullum and Ralph A. Willoughby (deceased)         HHL00020
C            Los Alamos National Laboratory                         HHL00030
C            Los Alamos, New Mexico 87544                           HHL00040
C            cullumj@lanl.gov                                       HHL00045
c                                                                   HHL00050
c  These codes are copyrighted by the authors.  These codes         HHL00060
c  and modifications of them or portions of them are NOT to be      HHL00070
c  incorporated into any commercial codes without legal agreements  HHL00080
c  with the authors.  If these codes or portions of them            HHL00090
c  are used in other scientific or engineering research works       HHL00100
c  the names of the authors of these codes and appropriate          HHL00110
c  references to their written work are to be incorporated in the   HHL00120
c  derivative works.                                                HHL00130
c                                                                   HHL00140
c  This header is not to be removed from these codes.               HHL00150
C                                                                   HHL00155
C         REFERENCE: Cullum and Willoughby, Chapters 1,2,3,4        HHL00160
C         Lanczos Algorithms for Large Symmetric Eigenvalue ComputationsHHL00165
C         VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in  HHL00166
C         Applied Mathematics, 2002. SIAM Publications,             HHL00167
C         Philadelphia, PA. USA                                     HHL00168
C                                                                   HHL00169
C     CONTAINS MAIN PROGRAM FOR COMPUTING DISTINCT EIGENVALUES OF   HHL00170
C     A HERMITIAN MATRIX USING LANCZOS TRIDIAGONALIZATION WITHOUT   HHL00180
C     REORTHOGONALIZATION                                           HHL00190
C                                                                   HHL00200
C     PORTABILITY:                                                  HHL00210
C     THIS PROGRAM IS NOT PORTABLE DUE TO THE USE OF COMPLEX*16     HHL00220
C     VARIABLES.  MOREOVER, THE PFORT VERIFIER IDENTIFIED THE       HHL00230
C     FOLLOWING ADDITIONAL NONPORTABLE CONSTRUCTIONS:               HHL00240
C                                                                   HHL00250
C     1.  DATA/MACHEP/ STATEMENT                                    HHL00260
C     2.  ALL READ(5,*) STATEMENTS (FREE FORMAT)                    HHL00270
C     3.  FORMAT(20A4) USED WITH EXPLANATORY HEADER EXPLAN.         HHL00280
C     4.  HEXADECIMAL FORMAT (4Z20) USED IN ALPHA/BETA FILES 1 AND 2.  HHL00290
C                                                                   HHL00300
C-------------------------------------------------------------------HHL00310
C     SPECIFY DIMENSIONS OF ARRAYS NEEDED BY LANCZOS ROUTINES       HHL00320
C                                                                   HHL00330
C     USER SPECIFIES THE FOLLOWING:                                 HHL00340
C     OTHER ARRAY DIMENSIONS ARE COMPUTED IN PARAMETER STATEMENTS   HHL00350
C     N = DIMENSION OF THE MATRIX EIGENVALUE PROBLEM                HHL00360
C     KMAX =  MAXIMUM SIZE OF LANCZOS MATRICES TO BE USED           HHL00370
C     NSINT >= NUMBER OF SUBINTERVALS SPECIFIED IN INPUT FILE 5     HHL00380
C     NTMATS >= NUMBER OF LANCZOS MATRICES SPECIFIED IN INPUT FILE 5 HHL00390
C     BELOW WE ASSUME THAT NO MORE THAN KMAX/2 EIGENVALUES          HHL00400
C     ARE COMPUTED IN ANY ONE OF THE SUBINTERVALS (LB(J),UB(J))     HHL00410
C     SUPPLIED BY THE USER.  V2 WILL BE USED FOR BOTH UPPER AND     HHL00420
C     LOWER BOUNDS ON THE EIGENVALUES AS THEY ARE COMPUTED SO       HHL00430
C     IF MORE THAN KMAX/2 EIGENVALUES ARE TO BE COMPUTED IN ANY     HHL00440
```

```
C     ONE SUBINTERVAL, THE DIMENSION OF V2 MUST BE ADJUSTED         HHL00450
C     ACCORDINGLY.  FOR EXAMPLE IF THE USER WANTS ALL THE EIGENVALUES HHL00460
C     OF THE LANCZOS MATRIX THEN KV2 MUST BE > MAX(KMAX,N)           HHL00470
C     BECAUSE OF THE INTEGER ARITHMETIC IT IS NECESSARY TO ADD AN    HHL00480
C     EXTRA 1 TO THE EXPRESSIONS.                                    HHL00490
C                                                                    HHL00500
C     TO AVOID USING MAX(I,J) IN THE PARAMETER LISTING WE HAVE USED  HHL00510
C     THE FOLLOWING EQUIVALENT RELATIONSHIP                          HHL00520
C                                                                    HHL00530
C     MAX(I,J) = ( 2*I/(I+J))*I  + (2*J/(I+J))*J                     HHL00540
C                                                                    HHL00550
C     PARAMETER ( N = 81, KMAX = 100, NSINT = 20, NTMATS = 20)       HHL00560
      PARAMETER ( N = 625, KMAX = 1500, NSINT = 20, NTMATS = 20)     HHL00570
C                                                                    HHL00580
      PARAMETER ( KMAX1 = KMAX+1, KMAX2 = 2*KMAX, NKMAX = N+KMAX )   HHL00590
      PARAMETER ( KMAXP2 = KMAX + 2)                                 HHL00600
      PARAMETER ( N2 = 2*N, N2KMAX = N2+KMAX, NKMAX2=N+KMAX2)        HHL00610
      PARAMETER ( KMAXP02 = KMAXP2/2, KMAX102 = KMAX1/2 )            HHL00620
      PARAMETER ( NKMAX12 = N+KMAX102, NKMAXP0 = N+KMAXP02)          HHL00630
      PARAMETER (KVS = ((2*N2)/N2KMAX)*N2 + ((2*KMAX)/N2KMAX)*KMAX ) HHL00640
      PARAMETER (KV1 =((2*N)/NKMAXP0)*N+((2*KMAXP02)/NKMAXP0)*KMAXP02) HHL00650
C     PARAMETER (KV2 = ((2*N)/NKMAX12)*N +((2*KMAX102)/NKMAX12)*KMAX102)HHL00660
C BELOW GOES WITH COMPUTING ALL EIGENVALUES OF LANCZOS MATRIX        HHL00670
      PARAMETER (KV2 = ((2*N)/NKMAX)*N + ((2*KMAX)/NKMAX)*KMAX)       HHL00680
      PARAMETER (KG = ((2*KMAX2)/NKMAX2)*KMAX2 +((2*N)/NKMAX2)*N )    HHL00690
C                                                                    HHL00700
C-------------------------------------------------------------------HHL00710
C                                                                    HHL00720
      DOUBLE PRECISION  ALPHA(KMAX),BETA(KMAX1),VS(KVS)              HHL00730
      COMPLEX*16  V1(KV1),V2(KV2)                                    HHL00740
      DOUBLE PRECISION  GR(N),GC(N),LB(NSINT),UB(NSINT)              HHL00750
      DOUBLE PRECISION  BTOL,GAPTOL,TTOL,MACHEP,EPSM,RELTOL          HHL00760
      DOUBLE PRECISION  SCALE1,SCALE2,SCALE3,SCALE4,BISTOL,CONTOL,MULTOLHHL00770
      DOUBLE PRECISION  ONE,ZERO,TEMP,TKMAX,BETAM,BKMIN,T0,T1        HHL00780
      REAL  G(KG),EXPLAN(20)                                         HHL00790
      INTEGER  MP(KMAX),NMEV(NTMATS)                                 HHL00800
      INTEGER  SVSEED,RHSEED,SVSOLD                                  HHL00810
      INTEGER IABS                                                   HHL00820
      REAL  ABS                                                      HHL00830
      DOUBLE PRECISION  DABS, DSQRT, DFLOAT                          HHL00840
      EXTERNAL CMATV                                                 HHL00850
C                                                                    HHL00860
C-------------------------------------------------------------------HHL00870
      DATA MACHEP/Z3410000000000000/                                HHL00880
      EPSM = 2.0D0*MACHEP                                           HHL00890
C-------------------------------------------------------------------HHL00900
      WRITE(6,1) N,KMAX,NSINT,NTMATS                                HHL00910
    1 FORMAT(' N,KMAX,NSINT,NTMATS ='/4I10)                         HHL00920
      WRITE(6,2) KMAX1,KMAX2,N2,N2KMAX,NKMAX2                       HHL00930
    2 FORMAT(' KMAX1,KMAX2,N2,N2KMAX,NKMAX2 ='/5I10)                HHL00940
      WRITE(6,3) KMAXP02,KMAX102,NKMAXP0,NKMAX12                    HHL00950
    3 FORMAT(' KMAXP02,KMAX102,NKMAXP0,NKMAX12 ='/4I10)             HHL00960
      WRITE(6,4) KVS,KV1,KV2,KG                                     HHL00970
    4 FORMAT(' KVS,KV1,KV2,KG ='/4I10)                              HHL00980
C                                                                    HHL00990
```

```
C        THE ARRAYS V1 AND V2 ARE DEFINED AS COMPLEX*16 IN THE MAIN PROGRAMHHL01000
C        AND IN THE SUBROUTINE LANCZS.  HOWEVER, IN THE OTHER SUBROUTINES  HHL01010
C        THEY ARE DECLARED AS DOUBLE PRECISION ARRAYS. NOTE THAT THE       HHL01020
C        DIMENSION OF V2 ASSUMES THAT NO MORE THAN KMAX/2 EIGENVALUES OF   HHL01030
C        THE T-MATRICES ARE BEING COMPUTED IN ANY ONE OF THE SUB-INTERVALS HHL01040
C        BEING CONSIDERED.  V2 MUST CONTAIN UPPER AND LOWER BOUNDS         HHL01050
C        ON EACH T-EIGENVALUE COMPUTED BY BISEC IN ANY ONE GIVEN INTERVAL. HHL01060
C                                                                          HHL01070
C        ARRAYS MUST BE DIMENSIONED AS FOLLOWS:                           HHL01080
C              1. ALPHA:   >= KMAX.  BETA:  >= (KMAX+1)                    HHL01090
C              2. V1:  >= MAX(N,(KMAX+1)/2).   V2:  >= MAX(N,KMAX/2)       HHL01100
C              3. VS:  >= MAX(2*N,KMAX).                                   HHL01110
C              4. GR,GC:  >=  N                                            HHL01120
C              5. G:  >= MAX(2*KMAX,N)                                     HHL01130
C              6. MP:   >= KMAX                                            HHL01140
C              7. LB,UB:  >= NUMBER OF SUB-INTERVALS SPECIFIED             HHL01150
C              8. NMEV:  >= NUMBER OF T-MATRICES SPECIFIED                 HHL01160
C              9. EXPLAN:  DIMENSION IS 20.                                HHL01170
C                                                                          HHL01180
C                                                                          HHL01190
C        IMPORTANT TOLERANCES OR SCALES THAT ARE USED REPEATEDLY           HHL01200
C        THROUGHOUT THE PROGRAM ARE THE FOLLOWING:                         HHL01210
C        SCALED MACHINE EPSILON:  TTOL = TKMAX*EPSM WHERE                  HHL01220
C        EPSM = 2*MACHINE EPSILON AND                                      HHL01230
C        TKMAX = MAX(|ALPHA(J)|,BETA(J), J = 1,MEV)                        HHL01240
C        BISEC CONVERGENCE TOLERANCE:  BISTOL = DSQRT(1000+MEV)*TTOL       HHL01250
C        BISEC MULTIPLICITY TOLERANCE:  MULTOL = (1000+MEV)*TTOL           HHL01260
C        LANCZOS CONVERGENCE TOLERANCE:   CONTOL = BETA(MEV+1)*1.D-10      HHL01270
C-------------------------------------------------------------------------HHL01280
C        OUTPUT HEADER                                                     HHL01290
         WRITE(6,10)                                                       HHL01300
   10 FORMAT(/' LANCZOS PROCEDURE FOR HERMITIAN MATRICES'/)                HHL01310
C                                                                          HHL01320
C        SET PROGRAM PARAMETERS                                            HHL01330
C        SCALEK ARE USED IN TOLERANCES NEEDED IN SUBROUTINES LUMP,         HHL01340
C        ISOEV AND PRTEST.  USER MUST NOT MODIFY THESE SCALES.             HHL01350
         SCALE1 = 5.0D2                                                    HHL01360
         SCALE2 = 5.0D0                                                    HHL01370
         SCALE3 = 5.0D0                                                    HHL01380
         SCALE4 = 1.0D4                                                    HHL01390
         ONE  = 1.0D0                                                      HHL01400
         ZERO = 0.0D0                                                      HHL01410
         BTOL = EPSM                                                       HHL01420
C        BTOL = 1.0D-8                                                     HHL01430
         GAPTOL = 1.0D-8                                                   HHL01440
         ICONV = 0                                                         HHL01450
         MOLD = 0                                                          HHL01460
         MOLD1 = 1                                                         HHL01470
         ICT = 0                                                           HHL01480
         MMB = 0                                                           HHL01490
         IPROJ = 0                                                         HHL01500
C                                                                          HHL01510
C        READ USER-SPECIFIED PARAMETERS FROM INPUT FILE 5 (FREE FORMAT)    HHL01520
C                                                                          HHL01530
C        READ USER-PROVIDED HEADER FOR RUN                                 HHL01540
```

```
      READ(5,20) EXPLAN                                                 HHL01550
      WRITE(6,20) EXPLAN                                                HHL01560
      READ(5,20) EXPLAN                                                 HHL01570
      WRITE(6,20) EXPLAN                                                HHL01580
   20 FORMAT(20A4)                                                      HHL01590
C                                                                       HHL01600
C  MODIFIED 4/16/93, N AND KMAX SET IN PARAMETER LIST.                  HHL01610
C XXXXREAD ORDER OF MATRICES (N) , MAXIMUM ORDER OF T-MATRIX (KMAX),    HHL01620
C     NUMBER OF T-MATRICES ALLOWED (NMEVS), AND MATRIX IDENTIFICATION   HHL01630
C     NUMBERS (MATNO)                                                   HHL01640
      READ(5,20) EXPLAN                                                 HHL01650
      READ(5,*) NMEVS,MATNO                                             HHL01660
C     READ(5,*) N,KMAX,NMEVS,MATNO                                      HHL01670
C                                                                       HHL01680
C     READ SEEDS FOR LANCZS AND INVERR SUBROUTINES (SVSEED AND RHSEED)  HHL01690
C     READ MAXIMUM NUMBER OF ITERATIONS ALLOWED FOR EACH INVERSE        HHL01700
C     ITERATION (MXINIT) AND MAXIMUM NUMBER OF STURM SEQUENCES          HHL01710
C     ALLOWED (MXSTUR)                                                  HHL01720
      READ(5,20) EXPLAN                                                 HHL01730
      READ(5,*) SVSEED,RHSEED,MXINIT,MXSTUR                             HHL01740
C                                                                       HHL01750
C     ISTART = (0,1):  ISTART = 0 MEANS ALPHA/BETA FILE IS NOT          HHL01760
C     AVAILABLE.  ISTART = 1 MEANS ALPHA/BETA FILE IS AVAILABLE ON      HHL01770
C     FILE 2.                                                           HHL01780
C     ISTOP = (0,1):  ISTOP = 0 MEANS PROCEDURE GENERATES ALPHA/BETA    HHL01790
C     FILE AND THEN TERMINATES.  ISTOP = 1 MEANS PROCEDURE GENERATES    HHL01800
C     ALPHAS/BETAS IF NEEDED AND THEN COMPUTES EIGENVALUES AND ERROR    HHL01810
C     ESTIMATES AND THEN TERMINATES.                                    HHL01820
      READ(5,20) EXPLAN                                                 HHL01830
      READ(5,*) ISTART,ISTOP                                            HHL01840
C                                                                       HHL01850
C     IHIS = (0,1):  IHIS = 0 MEANS ALPHA/BETA FILE IS NOT WRITTEN      HHL01860
C     TO FILE 1.  IHIS = 1 MEANS ALPHA/BETA FILE IS WRITTEN TO FILE 1.  HHL01870
C     IDIST = (0,1):  IDIST = 0 MEANS DISTINCT T-EIGENVALUES            HHL01880
C     ARE NOT WRITTEN TO FILE 11.  IDIST = 1 MEANS DISTINCT             HHL01890
C     T-EIGENVALUES ARE WRITTEN TO FILE 11.                             HHL01900
C     IWRITE = (0,1):  IWRITE = 0 MEANS NO INTERMEDIATE OUTPUT          HHL01910
C     FROM THE COMPUTATIONS IS WRITTEN TO FILE 6.  IWRITE = 1 MEANS     HHL01920
C     T-EIGENVALUES AND ERROR ESTIMATES ARE WRITTEN TO FILE 6           HHL01930
C     AS THEY ARE COMPUTED.                                             HHL01940
      READ(5,20) EXPLAN                                                 HHL01950
      READ(5,*) IHIS,IDIST,IWRITE                                       HHL01960
C                                                                       HHL01970
C     READ IN THE RELATIVE TOLERANCE (RELTOL) FOR USE IN THE            HHL01980
C     SPURIOUS, T-MULTIPLICITY, AND PRTESTS.                            HHL01990
      READ(5,20) EXPLAN                                                 HHL02000
      READ(5,*) RELTOL                                                  HHL02010
C                                                                       HHL02020
C     READ IN THE SIZES OF THE T-MATRICES TO BE CONSIDERED.             HHL02030
      READ(5,20) EXPLAN                                                 HHL02040
      READ(5,*) (NMEV(J), J=1,NMEVS)                                    HHL02050
C                                                                       HHL02060
C     READ IN THE NUMBER OF SUBINTERVALS TO BE CONSIDERED.              HHL02070
      READ(5,20) EXPLAN                                                 HHL02080
      READ(5,*) NINT                                                    HHL02090
```

```
C                                                              HHL02100
C     READ IN THE LEFT-END POINTS OF THE SUBINTERVALS TO BE CONSIDERED. HHL02110
C     THESE MUST BE IN ALGEBRAICALLY-INCREASING ORDER          HHL02120
      READ(5,20) EXPLAN                                         HHL02130
      READ(5,*) (LB(J), J=1,NINT)                              HHL02140
C                                                              HHL02150
C     READ IN THE RIGHT-END POINTS OF THE SUBINTERVALS TO BE CONSIDERED.HHL02160
C     THESE MUST BE IN ALGEBRAICALLY-INCREASING ORDER          HHL02170
      READ(5,20) EXPLAN                                         HHL02180
      READ(5,*) (UB(J), J=1,NINT)                              HHL02190
C                                                              HHL02200
C-------------------------------------------------------------HHL02210
C                                                              HHL02220
C     INITIALIZE THE ARRAYS FOR THE USER-SPECIFIED MATRIX      HHL02230
C     AND PASS THE STORAGE LOCATIONS OF THESE ARRAYS TO THE    HHL02240
C     MATRIX-VECTOR MULTIPLY SUBROUTINE CMATV.                 HHL02250
C                                                              HHL02260
      CALL USPEC(N,MATNO)                                       HHL02270
C                                                              HHL02280
C-------------------------------------------------------------HHL02290
C                                                              HHL02300
C     MASK UNDERFLOW AND OVERFLOW                              HHL02310
C                                                              HHL02320
      CALL MASK                                                 HHL02330
C                                                              HHL02340
C-------------------------------------------------------------HHL02350
C                                                              HHL02360
C     WRITE TO FILE 6, A SUMMARY OF THE PARAMETERS FOR THIS RUN HHL02370
C                                                              HHL02380
      WRITE(6,30) MATNO,N,KMAX                                  HHL02390
   30 FORMAT(/3X,'MATRIX ID',4X,'ORDER OF A',4X,'MAX ORDER OF T'/ HHL02400
     1 I12,I14,I18/)                                           HHL02410
C                                                              HHL02420
      WRITE(6,40) ISTART,ISTOP                                  HHL02430
   40 FORMAT(/2X,'ISTART',3X,'ISTOP'/2I8/)                     HHL02440
C                                                              HHL02450
      WRITE(6,50) IHIS,IDIST,IWRITE                             HHL02460
   50 FORMAT(/4X,'IHIS',3X,'IDIST',2X,'IWRITE'/3I8/)           HHL02470
C                                                              HHL02480
      WRITE(6,60) SVSEED,RHSEED                                 HHL02490
   60 FORMAT(/' SEEDS FOR RANDOM NUMBER GENERATOR'//            HHL02500
     1 4X,'LANCZS SEED',4X,'INVERR SEED'/2I15/)                HHL02510
C                                                              HHL02520
      WRITE(6,70) (NMEV(J), J=1,NMEVS)                          HHL02530
   70 FORMAT(/' SIZES OF T-MATRICES TO BE CONSIDERED'/(6I12))  HHL02540
C                                                              HHL02550
      WRITE(6,80) RELTOL,GAPTOL,BTOL                            HHL02560
   80 FORMAT(/' RELATIVE TOLERANCE USED TO COMBINE COMPUTED T-EIGENVALUEHHL02570
     1S'/E15.3/' RELATIVE GAP TOLERANCES USED IN INVERSE ITERATION'/ HHL02580
     1E15.3/' RELATIVE TOLERANCE FOR CHECK ON SIZE OF BETAS'/E15.3/) HHL02590
C                                                              HHL02600
      WRITE(6,90) (J,LB(J),UB(J), J=1,NINT)                     HHL02610
   90 FORMAT(/' BISEC WILL BE USED ON THE FOLLOWING INTERVALS'/ HHL02620
     1 (I6,2E20.6))                                            HHL02630
C                                                              HHL02640
```

```
      IF (ISTART.EQ.0) GO TO 140                                 HHL02650
C                                                                HHL02660
C     READ IN ALPHA BETA HISTORY                                HHL02670
C                                                                HHL02680
      READ(2,100)MOLD,NOLD,SVSOLD,MATOLD                         HHL02690
  100 FORMAT(2I6,I12,I8)                                         HHL02700
C                                                                HHL02710
C  CHANGED KMAX TO PARAMETER VARIABLE SO BELOW NO LONGER ALLOWED HHL02720
C  SO DEFAULT TO TERMINATE IF HISTORY FILE IS NOT LONG ENOUGH    HHL02730
C     IF (KMAX.LT.MOLD) KMAX = MOLD                              HHL02740
C     KMAX1 = KMAX + 1                                           HHL02750
C                                                                HHL02760
      IF (KMAX.LT.MOLD) WRITE(6,115) KMAX,MOLD                   HHL02770
      IF (KMAX.LT.MOLD) GO TO 640                                HHL02780
  115 FORMAT(/' PROGRAM TERMINATES FOR USER TO RESET KMAX.  CURRENT VALUHHL02790
     1E',I6/' IS LARGER THAN THE SIZE',I6,' OF THE TRIDIAGONAL MATRIX ONHHL02800
     1FILE 2'/)                                                  HHL02810
C                                                                HHL02820
C     CHECK THAT ORDER N, MATRIX ID MATNO, AND RANDOM SEED SVSEED HHL02830
C     AGREE WITH THOSE IN THE HISTORY FILE.  IF NOT PROCEDURE STOPS. HHL02840
C                                                                HHL02850
      ITEMP = (NOLD-N)**2+(MATNO-MATOLD)**2+(SVSEED-SVSOLD)**2    HHL02860
C                                                                HHL02870
      IF (ITEMP.EQ.0) GO TO 120                                  HHL02880
C                                                                HHL02890
      WRITE(6,110)                                               HHL02900
  110 FORMAT(' PROGRAM TERMINATES'/    ' READ FROM FILE 2 CORRESPONDS TOHHL02910
     1 DIFFERENT MATRIX THAN MATRIX SPECIFIED'/)                 HHL02920
      GO TO 640                                                  HHL02930
C                                                                HHL02940
  120 CONTINUE                                                   HHL02950
      MOLD1 = MOLD+1                                             HHL02960
C                                                                HHL02970
      READ(2,130)(ALPHA(J), J=1,MOLD)                            HHL02980
      READ(2,130)(BETA(J), J=1,MOLD1)                            HHL02990
  130 FORMAT(4Z20)                                               HHL03000
C                                                                HHL03010
      IF (KMAX.EQ.MOLD) GO TO 160                                HHL03020
C                                                                HHL03030
      READ(2,130)(V1(J), J=1,N)                                  HHL03040
      READ(2,130)(V2(J), J=1,N)                                  HHL03050
C                                                                HHL03060
  140 CONTINUE                                                   HHL03070
      IIX = SVSEED                                               HHL03080
C                                                                HHL03090
C----------------------------------------------------------------HHL03100
C                                                                HHL03110
      CALL LANCZS(CMATV,V1,V2,VS,ALPHA,BETA,GR,GC,G,KMAX,MOLD1,N,IIX) HHL03120
C                                                                HHL03130
C----------------------------------------------------------------HHL03140
C                                                                HHL03150
C COMMENTED OUT BELOW BECAUSE KMAX1 IS NOW SET IN PARAMETER LIST  HHL03160
C     KMAX1 = KMAX + 1                                           HHL03170
C                                                                HHL03180
      IF (IHIS.EQ.0.AND.ISTOP.GT.0) GO TO 160                    HHL03190
```

```
C                                                                    HHL03200
      WRITE(1,150) KMAX,N,SVSEED,MATNO                                HHL03210
  150 FORMAT(2I6,I12,I8,' = KMAX,N,SVSEED,MATNO')                     HHL03220
C                                                                    HHL03230
C     TO AVOID PERTURBATIONS CAUSED BY HEX TO DECIMAL AND DECIMAL TO HEXHHL03240
C     CONVERSIONS, THE ALPHA AND BETA MUST BE WRITTEN OUT IN HEX.     HHL03250
      WRITE(1,130)(ALPHA(I), I=1,KMAX)                                HHL03260
      WRITE(1,130)(BETA(I), I=1,KMAX1)                                HHL03270
C     WRITE(1,135)(ALPHA(I), I=1,N)                                   HHL03280
C     WRITE(1,135)(BETA(I), I=1,N)                                    HHL03290
  135 FORMAT(4E20.12)                                                 HHL03300
C                                                                    HHL03310
C     WRITE(1,130)(V1(I), I=1,N)                                      HHL03320
C     WRITE(1,130)(V2(I), I=1,N)                                      HHL03330
C                                                                    HHL03340
      IF (ISTOP.EQ.0) GO TO 540                                       HHL03350
C                                                                    HHL03360
  160 CONTINUE                                                        HHL03370
      BKMIN = BTOL                                                    HHL03380
      WRITE(6,170)                                                    HHL03390
  170 FORMAT(/' T-MATRICES (ALPHA AND BETA) ARE NOW AVAILABLE'/)      HHL03400
C                                                                    HHL03410
C----------------------------------------------------------------HHL03420
C     SUBROUTINE TNORM CHECKS MIN(BETA)/(ESTIMATED NORM(A)) > BTOL .  HHL03430
C     IF THIS IS VIOLATED IB IS SET EQUAL TO THE NEGATIVE OF THE INDEX HHL03440
C     OF THE MINIMAL BETA.  IF(IB < 0) THEN SUBROUTINE TNORM IS       HHL03450
C     CALLED FOR EACH VALUE OF MEV TO DETERMINE WHETHER OR NOT THERE  HHL03460
C     IS A BETA IN THE T-MATRIX SPECIFIED THAT VIOLATES THIS TEST.    HHL03470
C     IF THERE IS SUCH A BETA THE PROGRAM TERMINATES FOR THE USER     HHL03480
C     TO DECIDE WHAT TO DO.  THIS TEST CAN BE OVER-RIDDEN BY          HHL03490
C     SIMPLY MAKING BTOL SMALLER, BUT THEN THERE IS THE POSSIBILITY   HHL03500
C     THAT LOSSES IN THE LOCAL ORTHOGONALITY MAY HURT THE COMPUTATIONS. HHL03510
C     BTOL = 1.D-8 IS HOWEVER A CONSERVATIVE CHOICE FOR BTOL.         HHL03520
C                                                                    HHL03530
C     TNORM ALSO COMPUTES TKMAX = MAX(|ALPHA(K)|,BETA(K), K=1,KMAX).  HHL03540
C     TKMAX IS USED TO SCALE THE TOLERANCES USED IN THE              HHL03550
C     T-MULTIPLICITY AND SPURIOUS TESTS IN BISEC. TKMAX IS ALSO USED IN HHL03560
C     THE PROJECTION TEST FOR HIDDEN EIGENVALUES THAT HAD 'TOO SMALL' HHL03570
C     A PROJECTION ON THE STARTING VECTOR.                            HHL03580
C                                                                    HHL03590
      CALL TNORM(ALPHA,BETA,BKMIN,TKMAX,KMAX,IB)                      HHL03600
C                                                                    HHL03610
C----------------------------------------------------------------HHL03620
C                                                                    HHL03630
      TTOL = EPSM*TKMAX                                               HHL03640
C                                                                    HHL03650
C     LOOP ON THE SIZE OF THE T-MATRIX                                HHL03660
C                                                                    HHL03670
  180 CONTINUE                                                        HHL03680
      MMB = MMB + 1                                                   HHL03690
      MEV = NMEV(MMB)                                                 HHL03700
C     IS MEV TOO LARGE ?                                              HHL03710
      IF(MEV.LE.KMAX) GO TO 200                                       HHL03720
      WRITE(6,190) MMB, MEV, KMAX                                     HHL03730
  190 FORMAT(/' TERMINATE PRIOR TO CONSIDERING THE',I6,'TH T-MATRIX'/ HHL03740
```

```
      1' BECAUSE THE SIZE REQUESTED',I6,' IS GREATER THAN THE MAXIMUM SIZHHL03750
      1E ALLOWED',I6/)                                              HHL03760
        GO TO 540                                                   HHL03770
C                                                                   HHL03780
  200 MP1 = MEV + 1                                                 HHL03790
      BETAM = BETA(MP1)                                             HHL03800
C                                                                   HHL03810
      IF (IB.GE.0) GO TO 210                                        HHL03820
C                                                                   HHL03830
      T0 = BTOL                                                     HHL03840
C                                                                   HHL03850
C-------------------------------------------------------------------HHL03860
C                                                                   HHL03870
      CALL TNORM(ALPHA,BETA,T0,T1,MEV,IBMEV)                        HHL03880
C                                                                   HHL03890
C-------------------------------------------------------------------HHL03900
C                                                                   HHL03910
      TEMP = T0/TKMAX                                               HHL03920
      IBMEV = IABS(IBMEV)                                           HHL03930
      IF (TEMP.GE.BTOL) GO TO 210                                   HHL03940
      IBMEV = -IBMEV                                                HHL03950
      GO TO 600                                                     HHL03960
C                                                                   HHL03970
  210 CONTINUE                                                      HHL03980
      IC = MXSTUR-ICT                                               HHL03990
C                                                                   HHL04000
C-------------------------------------------------------------------HHL04010
C     BISEC LOOP. THE SUBROUTINE BISEC INCORPORATES DIRECTLY THE    HHL04020
C     T-MULTIPLICITY AND SPURIOUS TESTS. T-EIGENVALUES WILL BE      HHL04030
C     CALCULATED BY BISEC SEQUENTIALLY ON INTERVALS                 HHL04040
C     (LB(J),UB(J)), J = 1,NINT).                                   HHL04050
C                                                                   HHL04060
C     ON RETURN FROM BISEC                                          HHL04070
C     NDIS = NUMBER OF DISTINCT EIGENVALUES OF T(1,MEV) ON UNION    HHL04080
C            OF THE (LB,UB) INTERVALS                               HHL04090
C     VS = DISTINCT T-EIGENVALUES IN ALGEBRAICALLY INCREASING ORDER HHL04100
C     MP = MULTIPLICITIES OF THE T-EIGENVALUES IN VS               HHL04110
C     MP(I) = (0,1,MI), MI>1, I=1,NDIS  MEANS:                      HHL04120
C        (0)  VS(I) IS SPURIOUS                                     HHL04130
C        (1)  VS(I) IS T-SIMPLE AND GOOD                            HHL04140
C        (MI) VS(I) IS MULTIPLE AND IS THEREFORE NOT ONLY GOOD BUT  HHL04150
C             ALSO A CONVERGED GOOD T-EIGENVALUE.                   HHL04160
C     WITHIN BISEC V1 AND V2 ARE DEFINED AS DOUBLE PRECISION ARRAYS HHL04170
C                                                                   HHL04180
C                                                                   HHL04190
      CALL BISEC(ALPHA,BETA,V1,V2,VS,LB,UB,EPSM,TTOL,MP,NINT,       HHL04200
     1 MEV,NDIS,IC,IWRITE)                                          HHL04210
C                                                                   HHL04220
C-------------------------------------------------------------------HHL04230
C                                                                   HHL04240
      IF (NDIS.EQ.0) GO TO 620                                      HHL04250
C                                                                   HHL04260
C     COMPUTE THE TOTAL NUMBER OF STURM SEQUENCES USED TO DATE      HHL04270
C     COMPUTE THE BISEC CONVERGENCE AND T-MULTIPLICITY TOLERANCES USED. HHL04280
C     COMPUTE THE CONVERGENCE TOLERANCE FOR EIGENVALUES OF A.       HHL04290
```

```
      ICT = ICT + IC                                          HHL04300
      TEMP = DFLOAT(MEV+1000)                                 HHL04310
      MULTOL = TEMP*TTOL                                      HHL04320
      TEMP = DSQRT(TEMP)                                      HHL04330
      BISTOL = TTOL*TEMP                                      HHL04340
      CONTOL = BETAM*1.D-10                                   HHL04350
C                                                             HHL04360
C------------------------------------------------------------HHL04370
C    SUBROUTINE LUMP 'COMBINES' T-EIGENVALUES THAT ARE 'TOO CLOSE'.   HHL04380
C    NOTE HOWEVER THAT CLOSE SPURIOUS T-EIGENVALUES ARE NOT AVERAGED  HHL04390
C    WITH GOOD ONES. HOWEVER, THEY MAY BE USED TO INCREASE THE        HHL04400
C    MULTIPLICITY OF A GOOD T-EIGENVALUE.                   HHL04410
C                                                             HHL04420
      LOOP = NDIS                                             HHL04430
      CALL LUMP(VS,RELTOL,MULTOL,SCALE2,MP,LOOP)             HHL04440
C                                                             HHL04450
C------------------------------------------------------------HHL04460
C                                                             HHL04470
      IF(NDIS.EQ.LOOP) GO TO 230                             HHL04480
C                                                             HHL04490
      WRITE(6,220) NDIS, MEV, LOOP                           HHL04500
  220 FORMAT(/I6,' DISTINCT T-EIGENVALUES WERE COMPUTED IN BISEC AT MEV HHL04510
     1',I6/ 2X,' LUMP SUBROUTINE REDUCES NUMBER OF DISTINCT EIGENVALUES HHL04520
     1TO',I6)                                                HHL04530
C                                                             HHL04540
  230 CONTINUE                                                HHL04550
      NDIS = LOOP                                             HHL04560
      BETA(MP1) = BETAM                                       HHL04570
C                                                             HHL04580
C------------------------------------------------------------HHL04590
C    THE SUBROUTINE ISOEV LABELS THOSE SIMPLE EIGENVALUES OF T(1,MEV)  HHL04600
C    WITH VERY SMALL GAPS BETWEEN NEIGHBORING EIGENVALUES OF T(1,MEV)  HHL04610
C    TO AVOID COMPUTING ERROR ESTIMATES FOR ANY SIMPLE GOOD           HHL04620
C    T-EIGENVALUE THAT IS TOO CLOSE TO A SPURIOUS T-EIGENVALUE.       HHL04630
C    ON RETURN FROM ISOEV, G CONTAINS CODED MINIMAL GAPS              HHL04640
C    BETWEEN THE DISTINCT EIGENVALUES OF T(1,MEV). (G IS REAL).       HHL04650
C    G(I) < 0 MEANS MINGAP IS DUE TO LEFT GAP G(I) > 0 MEANS DUE TO   HHL04660
C    RIGHT GAP. MP(I) = -1 MEANS THAT THE GOOD T-EIGENVALUE IS SIMPLE HHL04670
C    AND HAS A VERY SMALL MINGAP IN T(1,MEV) DUE TO A SPURIOUS        HHL04680
C    T-EIGENVALUE.   NG = NUMBER OF GOOD EIGENVALUES.                 HHL04690
C    NISO = NUMBER OF ISOLATED GOOD T-EIGENVALUES.                    HHL04700
C                                                             HHL04710
      CALL ISOEV(VS,GAPTOL,MULTOL,SCALE1,G,MP,NDIS,NG,NISO)  HHL04720
C                                                             HHL04730
C------------------------------------------------------------HHL04740
C                                                             HHL04750
      WRITE(6,240)NG,NISO,NDIS                                HHL04760
  240 FORMAT(/I6,' GOOD T-EIGENVALUES HAVE BEEN COMPUTED'/   HHL04770
     1 I6,' OF THESE ARE T-ISOLATED'/                        HHL04780
     2 I6,' = NUMBER OF DISTINCT T-EIGENVALUES COMPUTED'/)   HHL04790
C                                                             HHL04800
C    DO WE WRITE DISTINCT EIGENVALUES OF T-MATRIX TO FILE 4? HHL04810
      IF (IDIST.EQ.0) GO TO 280                              HHL04820
C                                                             HHL04830
      WRITE(11,250) NDIS,NISO,MEV,N,SVSEED,MATNO             HHL04840
```

```
  250 FORMAT(/4I6,I12,I8,' = NDIS,NISO,MEV,N,SVSEED,MATNO'/)     HHL04850
C                                                                HHL04860
      WRITE(11,260) (MP(I),VS(I),G(I), I=1,NDIS)                 HHL04870
  260 FORMAT(2(I3,E25.16,E12.3))                                 HHL04880
C                                                                HHL04890
      WRITE(11,270) NDIS, (MP(I), I=1,NDIS)                      HHL04900
  270 FORMAT(/I6,' = NDIS, T-MULTIPLICITIES (0 MEANS  SPURIOUS)'/(20I4))HHL04910
C                                                                HHL04920
  280 CONTINUE                                                   HHL04930
C                                                                HHL04940
      IF (NISO.NE.0) GO TO 310                                   HHL04950
C                                                                HHL04960
      WRITE(4,290) MEV                                           HHL04970
  290 FORMAT(/' AT MEV = ',I6,' THERE ARE NO ISOLATED T-EIGENVALUES'/  HHL04980
     1' SO NO ERROR ESTIMATES WERE COMPUTED/')                  HHL04990
C                                                                HHL05000
      WRITE(6,300)                                               HHL05010
  300 FORMAT(/' ALL COMPUTED GOOD T-EIGENVALUES ARE MULTIPLE'/   HHL05020
     1 ' THEREFORE ALL SUCH EIGENVALUES ARE ASSUMED TO HAVE CONVERGED') HHL05030
C                                                                HHL05040
      ICONV = 1                                                  HHL05050
      GO TO 350                                                  HHL05060
C                                                                HHL05070
  310 CONTINUE                                                   HHL05080
C                                                                HHL05090
C----------------------------------------------------------------HHL05100
C     SUBROUTINE INVERR COMPUTES ERROR ESTIMATES FOR ISOLATED GOOD    HHL05110
C     T-EIGENVALUES USING INVERSE ITERATION ON T(1,MEV). ON RETURN    HHL05120
C     G(J) = MINIMUM GAP IN T(1,MEV) FOR EACH VS(J), J=1,NDIS   HHL05130
C     G(MEV+I) = BETAM*|U(MEV)| = ERROR ESTIMATE FOR ISOLATED GOOD    HHL05140
C             T-EIGENVALUES, WHERE I = 1, NISO AND  BETAM = BETA(MEV+1)HHL05150
C             U(MEV) IS MEVTH COMPONENT OF THE UNIT EIGENVECTOR OF T   HHL05160
C             CORRESPONDING TO THE ITH ISOLATED GOOD T-EIGENVALUE.    HHL05170
C     A NEGATIVE ERROR ESTIMATE MEANS THAT FOR THAT PARTICULAR   HHL05180
C     EIGENVALUE THE INVERSE ITERATION DID NOT CONVERGE IN <= MXINIT  HHL05190
C     STEPS AND THAT THE CORRESPONDING ERROR ESTIMATE IS QUESTIONABLE. HHL05200
C                                                                HHL05210
C     V2 CONTAINS THE ISOLATED GOOD T-EIGENVALUES              HHL05220
C     V1 CONTAINS THE MINGAPS TO THE NEAREST DISTINCT  EIGENVALUE    HHL05230
C         OF T(1,MEV) FOR EACH ISOLATED GOOD EIGENVALUE IN V2.  HHL05240
C     VS CONTAINS THE NDIS DISTINCT EIGENVALUES OF T(1,MEV)    HHL05250
C     MP CONTAINS THE CORRESPONDING CODED T-MULTIPLICITIES      HHL05260
C     WITHIN INVERR V1 AND V2 ARE DOUBLE PRECISION ARRAYS       HHL05270
C                                                                HHL05280
      IT = MXINIT                                                HHL05290
      CALL INVERR(ALPHA,BETA,V1,V2,VS,EPSM,G,MP,MEV,MMB,NDIS,NISO,N,   HHL05300
     1 RHSEED,IT,IWRITE)                                         HHL05310
C                                                                HHL05320
C----------------------------------------------------------------HHL05330
C                                                                HHL05340
C     SIMPLE CHECK FOR CONVERGENCE. CHECKS TO SEE IF ALL OF THE ERROR  HHL05350
C     ESTIMATES ARE SMALLER THAN CONTOL.                        HHL05360
C     IF THIS TEST IS SATISFIED, THEN CONVERGENCE FLAG, ICONV IS SET   HHL05370
C     TO 1.  TYPICALLY ERROR ESTIMATES ARE VERY CONSERVATIVE.   HHL05380
C                                                                HHL05390
```

```
      WRITE(6,320) CONTOL                                      HHL05400
  320 FORMAT(/' CONVERGENCE IS TESTED USING THE CONVERGENCE TOLERANCE', HHL05410
     1E13.4/)                                                  HHL05420
C                                                              HHL05430
      II = MEV +1                                              HHL05440
      IF = MEV+NISO                                            HHL05450
      DO 330 I = II,IF                                         HHL05460
      IF (ABS(G(I)).GT.CONTOL) GO TO 350                       HHL05470
  330 CONTINUE                                                 HHL05480
      ICONV = 1                                                HHL05490
      MMB = NMEVS                                              HHL05500
C                                                              HHL05510
      WRITE(6,340) CONTOL                                      HHL05520
  340 FORMAT(' ALL COMPUTED ERROR ESTIMATES WERE LESS THAN',E15.4/ HHL05530
     1 ' THEREFORE PROCEDURE TERMINATES'/)                     HHL05540
C                                                              HHL05550
  350 CONTINUE                                                 HHL05560
C                                                              HHL05570
C     IF CONVERGENCE IS INDICATED, THAT IS ICONV = 1 ,THEN     HHL05580
C     THE SUBROUTINE PRTEST IS CALLED TO CHECK FOR ANY CONVERGED HHL05590
C     EIGENVALUES THAT HAVE BEEN MISLABELLED AS SPURIOUS BECAUSE HHL05600
C     THE PROJECTION OF THEIR EIGENVECTOR(S) ON THE STARTING   HHL05610
C     VECTOR WERE TOO SMALL.                                   HHL05620
C     NUMERICAL TESTS INDICATE THAT SUCH EIGENVALUES ARE RARE. HHL05630
C     IF FOR SOME REASON MANY OF THESE HIDDEN EIGENVALUES APPEAR HHL05640
C     ON SOME RUN, YOU CAN BE CERTAIN THAT SOMETHING IS FOULED UP. HHL05650
C                                                              HHL05660
      IF (ICONV.EQ.0) GO TO 480                                HHL05670
C                                                              HHL05680
C-------------------------------------------------------------HHL05690
C                                                              HHL05700
      CALL PRTEST (ALPHA,BETA,VS,TKMAX,EPSM,RELTOL,SCALE3,SCALE4, HHL05710
     1 MP,NDIS,MEV,IPROJ)                                      HHL05720
C                                                              HHL05730
C-------------------------------------------------------------HHL05740
C                                                              HHL05750
      IF(IPROJ.EQ.0) GO TO 470                                 HHL05760
C                                                              HHL05770
      IF(IDIST.EQ.1)  WRITE(11,360) IPROJ                      HHL05780
  360 FORMAT(' SUBROUTINE PRTEST WANTS TO RELABEL',I6,' SPURIOUS EIGENVAHHL05790
     1LUES'/' WE ACCEPT RELABELLING ONLY IF LAST COMPONENT OF T-EIGENVECHHL05800
     1TOR IS L.T. 1.D-10'/)                                    HHL05810
C                                                              HHL05820
      IIX = RHSEED                                             HHL05830
C                                                              HHL05840
C-------------------------------------------------------------HHL05850
C                                                              HHL05860
      CALL GENRAN(IIX,G,MEV)                                   HHL05870
C                                                              HHL05880
C-------------------------------------------------------------HHL05890
C                                                              HHL05900
      ITEN = -10                                               HHL05910
      NISOM = NISO + MEV                                       HHL05920
      IWRITO = IWRITE                                          HHL05930
      IWRITE = 0                                               HHL05940
```

```
C                                                                     HHL05950
      DO 390 J = 1,NDIS                                               HHL05960
      IF(MP(J).NE.ITEN) GO TO 390                                    HHL05970
      T0 = VS(J)                                                      HHL05980
C                                                                     HHL05990
C----------------------------------------------------------------HHL06000
C                                                                     HHL06010
      IT = MXINIT                                                     HHL06020
      CALL INVERM(ALPHA,BETA,V1,V2,T0,TEMP,T1,EPSM,G,MEV,IT,IWRITE)  HHL06030
C                                                                     HHL06040
C----------------------------------------------------------------HHL06050
C                                                                     HHL06060
      IF(TEMP.LE.1.D-10) GO TO 380                                   HHL06070
C     ERROR ESTIMATE WAS NOT SMALL REJECT RELABELLING OF THIS EIGENVALUEHHL06080
      IF(IDIST.EQ.1)  WRITE(11,370) J,T0,TEMP                        HHL06090
  370 FORMAT(/' LAST COMPONENT FOR',I6,'TH EIGENVALUE',E20.12/' IS TOO LHHL06100
     1ARGE = ',E15.6,' SO DO NOT ACCEPT PRTEST RELABELLING'/)        HHL06110
      MP(J) = 0                                                      HHL06120
      IPROJ = IPROJ - 1                                             HHL06130
      GO TO 390                                                      HHL06140
C     RELABELLING ACCEPTED                                          HHL06150
  380 NISOM = NISOM + 1                                              HHL06160
      G(NISOM) = BETAM*TEMP                                         HHL06170
  390 CONTINUE                                                       HHL06180
      IWRITE = IWRIT0                                                HHL06190
C                                                                     HHL06200
      IF(IPROJ.EQ.0) GO TO 430                                       HHL06210
      WRITE(6,400) IPROJ                                            HHL06220
  400 FORMAT(/I6,' T-EIGENVALUES WERE RECLASSIFIED AS GOOD.'/       HHL06230
     1' THESE ARE IDENTIFIED IN FILE 3 BY A T-MULTIPLICITY OF -10'/' USEHHL06240
     2R SHOULD INSPECT EACH TO MAKE SURE NEIGHBORS HAVE CONVERGED'/) HHL06250
C                                                                     HHL06260
      IF(IDIST.EQ.1)  WRITE(11,410) IPROJ                           HHL06270
  410 FORMAT(/I6,' T-EIGENVALUES WERE RELABELLED AS GOOD'/          HHL06280
     1' BELOW IS CORRECTED T-MULTIPLICITY PATTERN'/)                HHL06290
C                                                                     HHL06300
      WRITE(6,420) NDIS, (MP(I), I=1,NDIS)                          HHL06310
      IF(IDIST.EQ.1)  WRITE(11,420) NDIS, (MP(I), I=1,NDIS)         HHL06320
  420 FORMAT(/I6,' = NDIS, T-MULTIPLICITIES (0 MEANS  SPURIOUS)'/   HHL06330
     1 6X, ' (-10) MEANS SPURIOUS T-EIGENVALUE RELABELLED AS GOOD'/(20I4HHL06340
     1))                                                            HHL06350
C                                                                     HHL06360
C     RECALCULATE MINGAPS FOR DISTINCT T(1,MEV) EIGENVALUES.        HHL06370
  430 NM1 = NDIS - 1                                                 HHL06380
      G(NDIS) = VS(NM1)-VS(NDIS)                                    HHL06390
      G(1) = VS(2)-VS(1)                                             HHL06400
C                                                                     HHL06410
      DO 440 J = 2,NM1                                               HHL06420
      T0 = VS(J)-VS(J-1)                                             HHL06430
      T1 = VS(J+1)-VS(J)                                             HHL06440
      G(J) = T1                                                      HHL06450
      IF (T0.LT.T1) G(J) = -T0                                       HHL06460
  440 CONTINUE                                                       HHL06470
      IF(IPROJ.EQ.0) GO TO 470                                       HHL06480
C     WRITE TO FILE 4 ERROR ESTIMATES FOR THOSE T-EIGENVALUES RELABELLEDHHL06490
```

```
      NGOOD = 0                                               HHL06500
      DO 450 J = 1,NDIS                                       HHL06510
      IF(MP(J).EQ.0) GO TO 450                                HHL06520
      NGOOD = NGOOD + 1                                       HHL06530
      IF(MP(J).NE.ITEN) GO TO 450                             HHL06540
      TO = VS(J)                                              HHL06550
      NISO = NISO + 1                                         HHL06560
      NISOM = MEV + NISO                                      HHL06570
      WRITE(4,460) NGOOD,TO,G(NISOM),G(J)                     HHL06580
  450 CONTINUE                                                HHL06590
  460 FORMAT(I10,E25.16,2E14.3)                               HHL06600
C                                                             HHL06610
  470 CONTINUE                                                HHL06620
C                                                             HHL06630
C     WRITE THE GOOD T-EIGENVALUES TO FILE 3.  FIRST TRANSFER THEM    HHL06640
C     TO V2 AND THEIR T-MULTIPLICITIES TO THE CORRESPONDING POSITIONS HHL06650
C     IN MP AND COMPUTE THE A-MINGAPS, THE MINIMAL GAPS BETWEEN THE   HHL06660
C     GOOD T-EIGENVALUES.  THESE GAPS WILL BE PUT IN THE ARRAY G.     HHL06670
C     SINCE G CURRENTLY CONTAINS THE MINIMAL GAPS BETWEEN THE DISTINCT HHL06680
C     EIGENVALUES OF THE T-MATRIX, THESE GAPS WILL FIRST BE           HHL06690
C     TRANSFERRED TO GC.  NOTE THAT GC<0 MEANS THAT THAT MINIMAL GAP  HHL06700
C     IN THE T-MATRIX IS DUE TO A SPURIOUS T-EIGENVALUE.             HHL06710
C     ALL THIS INFORMATION IS PRINTED TO FILE 3                      HHL06720
C                                                             HHL06730
  480 CONTINUE                                                HHL06740
C                                                             HHL06750
      NG = 0                                                  HHL06760
      DO 490 I = 1,NDIS                                       HHL06770
      IF (MP(I).EQ.0) GO TO 490                               HHL06780
      NG = NG+1                                               HHL06790
      MP(NG) = MP(I)                                          HHL06800
      GR(NG) = VS(I)                                          HHL06810
      TEMP = G(I)                                             HHL06820
      TEMP = DABS(TEMP)                                       HHL06830
      J = I+1                                                 HHL06840
      IF (G(I).LT.ZERO) J = I-1                               HHL06850
      IF (MP(J).EQ.0) TEMP = -TEMP                            HHL06860
      GC(NG) = TEMP                                           HHL06870
  490 CONTINUE                                                HHL06880
C                                                             HHL06890
      WRITE(6,500)MEV                                         HHL06900
  500 FORMAT(//' T-EIGENVALUE CALCULATION AT MEV = ',I6,'    IS COMPLETEHHL06910
     1')                                                      HHL06920
C                                                             HHL06930
C     NG = NUMBER OF COMPUTED DISTINCT GOOD T-EIGENVALUES.  NEXT      HHL06940
C     GENERATE GAPS BETWEEN GOOD T-EIGENVALUES (AMINGAPS) AND PUT THEM HHL06950
C     IN G.  G(J) < 0 MEANS THE AMINGAP IS DUE TO THE LEFT-HAND GAP.  HHL06960
C                                                             HHL06970
      NGM1 = NG - 1                                           HHL06980
      G(NG) = GR(NGM1)-GR(NG)                                 HHL06990
      G(1) = GR(2)-GR(1)                                      HHL07000
C                                                             HHL07010
      DO 510 J = 2,NGM1                                       HHL07020
      TO = GR(J)-GR(J-1)                                      HHL07030
      T1 = GR(J+1)-GR(J)                                      HHL07040
```

```
         G(J) = T1                                                  HHL07050
         IF (T0.LT.T1) G(J) = -T0                                   HHL07060
   510 CONTINUE                                                     HHL07070
C                                                                   HHL07080
C     WRITE GOOD T-EIGENVALUES OUT TO FILE 3.                       HHL07090
C                                                                   HHL07100
       WRITE(3,520)NG,NDIS,MEV,N,SVSEED,MATNO,MULTOL,IB,BTOL        HHL07110
   520 FORMAT(4I6,I12,I8,' = NG,NDIS,MEV,N,SVEED,MATNO'/            HHL07120
      1 E20.12,I6,E13.4,' = MUTOL,INDEX MINIMAL BETA,BTOL'/         HHL07130
      1' EV NO',2X,'TMULT',7X,'GOOD T-EIGENVALUE',7X,'TMINGAP',7X,'AMINGAHHL07140
      1P')                                                         HHL07150
C                                                                   HHL07160
       WRITE(3,530)(I,MP(I),GR(I),GC(I),G(I), I=1,NG)               HHL07170
   530 FORMAT(2I6,E25.16,2E14.3)                                    HHL07180
C                                                                   HHL07190
C     IF CONVERGENCE FLAG ICONV.NE.1 AND NUMBER OF T-MATRICES       HHL07200
C     CONSIDERED TO DATE IS LESS THAN NUMBER ALLOWED, INCREMENT MEV. HHL07210
C     AND LOOP BACK TO 210 TO REPEAT COMPUTATIONS.  RESTORE BETA(MEV+1).HHL07220
C                                                                   HHL07230
       BETA(MP1) = BETAM                                            HHL07240
C                                                                   HHL07250
       IF (MMB.LT.NMEVS.AND.ICONV.NE.1) GO TO 180                   HHL07260
C                                                                   HHL07270
C     END OF LOOP ON DIFFERENT SIZE T-MATRICES ALLOWED.            HHL07280
C                                                                   HHL07290
   540 CONTINUE                                                     HHL07300
C                                                                   HHL07310
       IF(ISTOP.EQ.0)  WRITE(6,550)                                 HHL07320
   550 FORMAT(/' T-MATRICES (ALPHA AND BETA) ARE NOW AVAILABLE, TERMINATEHHL07330
      1')                                                          HHL07340
       IF (IHIS.EQ.1.AND.KMAX.NE.MOLD) WRITE(1,560)                 HHL07350
   560 FORMAT(/' ABOVE ARE THE FOLLOWING VECTORS '/                 HHL07360
      1 '   ALPHA(I), I = 1,KMAX'/                                  HHL07370
      2 '   BETA(I), I = 1,KMAX+1'/                                 HHL07380
      3 ' FINAL TWO LANCZOS VECTORS OF ORDER N FOR I = KMAX,KMAX+1'/ HHL07390
      4 ' ALL ENTRIES IN THIS FILE HAVE FORMAT 4Z20 '/             HHL07400
      5 ' ----- END OF FILE 1 NEW ALPHA, BETA HISTORY--------------'///)HHL07410
C                                                                   HHL07420
       IF (ISTOP.EQ.0) GO TO 640                                    HHL07430
C                                                                   HHL07440
       WRITE(3,570)                                                 HHL07450
   570 FORMAT(/' ABOVE ARE COMPUTED GOOD T-EIGENVALUES'/            HHL07460
      1 ' NG = NUMBER OF GOOD T-EIGENVALUES COMPUTED'/              HHL07470
      2 ' NDIS = NUMBER OF COMPUTED DISTINCT EIGENVALUES OF T(1,MEV)'/ HHL07480
      3 ' N = ORDER OF A,  MATNO = MATRIX IDENT'/                   HHL07490
      4 ' MULTOL = MULTIPLICITY TOLERANCE FOR T-EIGENVALUES IN BISEC'/ HHL07500
      4 ' TMULT IS THE T-MULTIPLICITY OF GOOD T-EIGENVALUE'/        HHL07510
      5 ' TMULT = -1 MEANS SPURIOUS T-EIGENVALUE TOO CLOSE'/        HHL07520
      6 ' DO NOT COMPUTE ERROR ESTIMATES FOR SUCH EIGENVALUES'/     HHL07530
      7 ' AMINGAP = MINIMAL GAP BETWEEN THE COMPUTED A-EIGENVALUES'/ HHL07540
      8 ' AMINGAP .LT. 0. MEANS MINIMAL GAP IS DUE TO LEFT-HAND GAP'/ HHL07550
      9 ' TMINGAP= MINIMAL GAP W.R.T.  DISTINCT EIGENVALUES IN T(1,MEV)'/HHL07560
      1 ' TMINGAP .LT. 0. MEANS MINGAP IS DUE TO SPURIOUS EIGENVALUE'/ HHL07570
      2 ' ----- END OF FILE 3 GOODEIGENVALUES---------------------'///)HHL07580
C                                                                   HHL07590
```

```
      IF (IDIST.EQ.1) WRITE(11,580)                             HHL07600
  580 FORMAT(/' ABOVE ARE THE DISTINCT EIGENVALUES OF T(1,MEV).'/   HHL07610
     2 ' THE FORMAT IS       T-MULTIPLICITY     EIGENVALUE    TMINGAP'/  HHL07620
     3 '         THIS FORMAT IS REPEATED TWICE ON EACH LINE.'/   HHL07630
     4 ' T-MULTIPLICITY = -1 MEANS THAT THE SUBROUTINE ISOEV HAS TAGGED'HHL07640
     5 /'   THIS SIMPLE T-EIGENVALUE AS HAVING A VERY CLOSE SPURIOUS'/  HHL07650
     6 '   T-EIGENVALUE SO THAT NO ERROR ESTIMATE WILL BE COMPUTED'/   HHL07660
     7 '   FOR THAT EIGENVALUE IN SUBROUTINE INVERR.'/         HHL07670
     8 ' TMINGAP .LT. 0, TMINGAP IS DUE TO LEFT GAP .GT. 0, RIGHT GAP.'/HHL07680
     9 ' EACH OF THE DISTINCT T-EIGENVALUE TABLES IS FOLLOWED'/   HHL07690
     9 ' BY THE T-MULTIPLICITY PATTERN.'/                      HHL07700
     1 ' NDIS = NUMBER OF COMPUTED DISTINCT EIGENVALUES OF T(1,MEV).'/  HHL07710
     2 ' NG = NUMBER OF GOOD T-EIGENVALUES. '/                 HHL07720
     3 ' NISO = NUMBER OF ISOLATED GOOD T-EIGENVALUES. '/      HHL07730
     4 ' NISO ALSO IS THE COUNT OF +1 ENTRIES IN T-MULTIPLICITY PATTERN.HHL07740
     5 '/' ----- END OF FILE 4 DISTINCT T-EIGENVALUES---------------'//HHL07750
     6 /)                                                      HHL07760
C                                                              HHL07770
      IF(NISO.NE.0)  WRITE(4,590)                              HHL07780
  590 FORMAT(/' ABOVE ARE THE ERROR ESTIMATES OBTAINED FOR THE ISOLATED HHL07790
     1GOOD EIGENVALUES'/                                       HHL07800
     1' OBTAINED VIA INVERSE ITERATION IN THE SUBROUTINE INVERR.'/   HHL07810
     1' ALL OTHER GOOD EIGENVALUES HAVE CONVERGED.'/           HHL07820
     2' ERROR ESTIMATE = BETAM*ABS(UM)'/                       HHL07830
     2' WHERE BETAM = BETA(MEV+1) AND UM = U(MEV).'/           HHL07840
     3' U = UNIT EIGENVECTOR OF T WHERE T*U = EV*U AND EV = ISOLATED GOOHHL07850
     3D EIGENVALUE.'/                                          HHL07860
     4' TMINGAP = GAP TO NEAREST DISTINCT EIGENVALUE OF T(1,MEV).'/   HHL07870
     5' TMINGAP .LT. 0. MEANS MINGAP IS DUE TO LEFT NEIGHBOR.'/   HHL07880
     6' ERROR ESTIMATE L.T. 0 MEANS INVERSE ITERATION DID NOT CONVERGE'/HHL07890
     7' ------ END OF FILE 7 ERRINV ----------------------------'//) HHL07900
      GO TO 640                                                HHL07910
C                                                              HHL07920
  600 CONTINUE                                                 HHL07930
C                                                              HHL07940
      IBB = IABS(IBMEV)                                        HHL07950
      IF (IBMEV.LT.0) WRITE(6,610) MEV,IBB,BETA(IBB)           HHL07960
  610 FORMAT(/' PROGRAM TERMINATES BECAUSE MEV REQUESTED = ',I6,' IS .GTHHL07970
     1',I6/' AT WHICH AN ABNORMALLY SMALL BETA = ' , E13.4,' OCCURRED'/)HHL07980
      GO TO 640                                                HHL07990
C                                                              HHL08000
  620 IF (NDIS.EQ.0.AND.ISTOP.GT.0) WRITE(6,630)               HHL08010
  630 FORMAT(/' INTERVALS SPECIFIED FOR BISECT DID NOT CONTAIN ANY EIGENHHL08020
     1VALUES'/' PROGRAM TERMINATES')                           HHL08030
C                                                              HHL08040
  640 CONTINUE                                                 HHL08050
C                                                              HHL08060
      STOP                                                     HHL08070
C-----END OF MAIN PROGRAM FOR LANCZOS HERMITIAN EIGENVALUE COMPUTATIONS-HHL08080
      END                                                      HHL08090
```

3.3 HLEVEC: Main Program, Eigenvector Computations

```
C     MODIFIED 8/16/83 ( and 4/27/93 to change array dimensioning)    HHL00010
C                                                                      HHL00020
C-----HLEVEC (EIGENVECTORS OF HERMITIAN MATRICES)---------------------HHL00030
C                                                                      HHL00040
C Authors:  Jane Cullum and Ralph A. Willoughby (deceased)            HHL00050
C           Los Alamos National Laboratory                           HHL00060
C           Los Alamos, New Mexico 87544                             HHL00070
C           E-Mail: cullumj@lanl.gov                                 HHL00075
c                                                                      HHL00080
c These codes are copyrighted by the authors.  These codes           HHL00090
c and modifications of them or portions of them are NOT to be        HHL00100
c incorporated into any commercial codes without legal agreements    HHL00110
c with the authors.  If these codes or portions of them              HHL00120
c are used in other scientific or engineering research works         HHL00130
c the names of the authors of these codes and appropriate            HHL00140
c references to their written work are to be incorporated in the      HHL00150
c derivative works.                                                   HHL00160
c                                                                      HHL00170
c This header is not to be removed from these codes.                  HHL00180
C                                                                      HHL00181
C         REFERENCE: Cullum and Willoughby, Chapters 1,2,3,4          HHL00182
C         Lanczos Algorithms for Large Symmetric Eigenvalue ComputationsHHL00183
C         VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in HHL00184
C         Applied Mathematics, 2002. SIAM Publications,              HHL00185
C         Philadelphia, PA. USA                                      HHL00186
C                                                                      HHL00187
c                                                                      HHL00190
C     CONTAINS MAIN PROGRAM FOR COMPUTING AN EIGENVECTOR CORRESPONDING HHL00200
C     TO EACH OF A SET OF EIGENVALUES THAT HAVE BEEN COMPUTED         HHL00210
C     ACCURATELY BY THE CORRESPONDING LANCZOS EIGENVALUE PROGRAM      HHL00220
C     (HLEVAL) FOR HERMITIAN MATRICES.  THIS PROGRAM COULD BE         HHL00230
C     MODIFIED TO COMPUTE ADDITIONAL EIGENVECTORS FOR ANY            HHL00240
C     MULTIPLE EIGENVALUE OF THE GIVEN A-MATRIX.  THE AMOUNT OF       HHL00250
C     ADDITIONAL COMPUTATION REQUIRED BY SUCH A MODIFICATION WOULD    HHL00260
C     DEPEND UPON THE GIVEN MATRIX AND UPON WHICH PART OF THE         HHL00270
C     SPECTRUM WAS INVOLVED.                                          HHL00280
C                                                                      HHL00290
C     THE LANCZOS EIGENVECTOR COMPUTATIONS ASSUME THAT EACH           HHL00300
C     EIGENVALUE THAT IS BEING CONSIDERED HAS CONVERGED AS AN         HHL00310
C     EIGENVALUE OF THE LANCZOS TRIDIAGONAL MATRICES.                 HHL00320
C                                                                      HHL00330
C     PORTABILITY:                                                    HHL00340
C     THIS PROGRAM IS NOT PORTABLE BECAUSE OF THE USE OF COMPLEX*16   HHL00350
C     VARIABLES.  MOREOVER, THE PFORT VERIFIER IDENTIFIED THE         HHL00360
C     FOLLOWING ADDITIONAL NONPORTABLE CONSTRUCTIONS:                 HHL00370
C                                                                      HHL00380
C     1.  DATA/MACHEP/ STATEMENT                                      HHL00390
C     2.  ALL READ(5,*) STATEMENTS (FREE FORMAT)                      HHL00400
C     3.  FORMAT(20A4) USED WITH THE EXPLANATORY HEADER, EXPLAN       HHL00410
C     4.  HEXADECIMAL FORMAT (4Z20) USED IN ALPHA/BETA FILES 1 AND 2. HHL00420
C                                                                      HHL00430
```

```
C      IMPORTANT NOTE:  PROGRAM ALLOWS ENLARGEMENT OF THE ALPHA, BETA      HHL00440
C      ARRAYS.  IN PARTICULAR, IF ANY ONE OF THE EIGENVALUES SUPPLIED      HHL00450
C      IS T-SIMPLE AND NOT CLOSE TO A SPURIOUS T-EIGENVALUE, THE PROGRAM   HHL00460
C      REQUIRES THAT KMAX BE AT LEAST 11*MEV/8 + 12.  IF KMAX IS NOT       HHL00470
C      THIS LARGE, THEN THE PROGRAM WILL RESET KMAX TO THIS SIZE           HHL00480
C      AND EXTEND THE ALPHA, BETA HISTORY IF REQUIRED.                     HHL00490
C      THUS, THE DIMENSIONS OF THE ALPHA AND BETA ARRAYS MUST BE           HHL00500
C      LARGE ENOUGH TO ALLOW FOR THIS POSSIBILITY.                         HHL00510
C      REMEMBER THAT THE BETA ARRAY, BETA(J), IS SUCH THAT                 HHL00520
C      J = 1,..., KMAX+1.  SO IF THE KMAX USED BY THE PROGRAM              HHL00530
C      IS TO BE 3000, THEN BETA MUST BE OF LENGTH AT LEAST 3001.           HHL00540
C                                                                          HHL00550
C      TO AVOID USING MAX(I,J) IN THE PARAMETER LISTING WE HAVE USED       HHL00560
C      THE FOLLOWING EQUIVALENT RELATIONSHIP                               HHL00570
C                                                                          HHL00580
C      MAX(I,J) = ( 2*I/(I+J))*I  + (2*J/(I+J))*J                          HHL00590
C                                                                          HHL00600
       parameter (n=625,mev=1500,ngood= 3,nngood=n*ngood)                  HHL00610
       parameter (kmaxn = (3*mev)/2 + 12, kmaxn1=kmaxn+1)                  HHL00620
       parameter(nkmaxn = ngood*kmaxn)                                     HHL00630
       PARAMETER ( KMAXn1 = KMAXn+1, KMAXn12 = KMAXn1/2 )                  HHL00640
       PARAMETER ( NKMAXn2 = N+KMAXn12, NPKMAXn = N+KMAXn)                 HHL00650
       PARAMETER (KVS = ((2*N)/NPKMAXn)*N + ((2*KMAXn)/NPKMAXn)*KMAXn )    HHL00660
       PARAMETER (KV2 = ((2*N)/NKMAXn2)*N +((2*KMAXn12)/NKMAXn2)*KMAXn12)  HHL00670
C-----------------------------------------------------------------------HHL00680
       COMPLEX*16  V1(kv2),V2(n),VS(n),RITVEC(nngood),ZEROC,TEMPC         HHL00690
       DOUBLE PRECISION  ALPHA(kmaxn),BETA(kmaxn1),GR(n),GC(n)            HHL00700
       DOUBLE PRECISION  TVEC(nkmaxm),GOODEV(ngood),EVNEW(ngood)          HHL00710
       DOUBLE PRECISION  EVAL,EVALN,TOLN,TTOL,ERTOL,ALFA,BATA             HHL00720
       DOUBLE PRECISION  MULTOL,SCALE0,STUTOL,BTOL,LB,UB                  HHL00730
       DOUBLE PRECISION  ONE,ZERO,MACHEP,EPSM,TEMP,SUM,ERRMIN,BKMIN       HHL00740
       DOUBLE PRECISION  RELTOL,ERROR,TERROR,TLAST(ngood)                 HHL00750
       REAL  G(kvs),AMINGP(ngood),TMINGP(ngood),EXPLAN(20)                HHL00760
       REAL  TERR(ngood),ERR(ngood),ERRDGP(ngood),RNORM(ngood)            HHL00770
       real TBETA(ngood)                                                  HHL00780
       INTEGER  MP(ngood),M1(ngood),M2(ngood),MA(ngood)                   HHL00790
       integer ML(ngood),MINT(ngood),MFIN(ngood)                          HHL00800
       INTEGER  SVSEED,SVSOLD,RHSEED,IDELTA(ngood),MULEVA(ngood)          HHL00810
       INTEGER  MBOUND,NTVCON,SVTVEC,TVSTOP,LVCONT,ERCONT,TFLAG           HHL00820
       DOUBLE PRECISION  DABS, DMAX1, DSQRT, DFLOAT                       HHL00830
       REAL ABS                                                           HHL00840
       INTEGER  IABS                                                      HHL00850
C-----------------------------------------------------------------------HHL00860
       EXTERNAL CMATV                                                     HHL00870
       DATA MACHEP/Z3410000000000000/                                     HHL00880
       EPSM = 2.D0*MACHEP                                                 HHL00890
C-----------------------------------------------------------------------HHL00900
C                                                                         HHL00910
C      ARRAYS MUST BE DIMENSIONED AS FOLLOWS:                             HHL00920
C      1.  ALPHA: >= KMAXN,  BETA: >= (KMAXN+1) WHERE KMAXN, THE          HHL00930
C                 LARGEST SIZE T-MATRIX CONSIDERED BY THE PROGRAM,        HHL00940
C                 IS THE LARGER OF THE SIZE OF THE ALPHA, BETA HISTORY    HHL00950
C                 PROVIDED ON FILE 2 (IF ANY ) AND THE SIZE WHICH THE     HHL00960
C                 PROGRAM SPECIFIES INTERNALLY, THIS LATTER IS ALWAYS     HHL00970
C                 < = 11*MEV / 8  + 12, WHERE MEV IS THE SIZE             HHL00980
```

```
C               T-MATRIX THAT WAS USED IN THE CORRESPONDING EIGENVALUE HHL00990
C               COMPUTATIONS.                                           HHL01000
C     2.  V1:  >= MAX(N,KMAX/2)                                         HHL01010
C     3.  V2, VS:   >= N                                                HHL01020
C     4.  G:  >= MAX(N,KMAX).   GR, GC:  >= N                           HHL01030
C     5.  RITVEC:  >= N*NGOOD, WHERE NGOOD IS NUMBER OF EIGENVALUES     HHL01040
C               SUPPLIED TO THIS PROGRAM.                               HHL01050
C     6.  TVEC:  >= CUMULATIVE LENGTH OF ALL THE T-EIGENVECTORS         HHL01060
C         NEEDED TO GENERATE THE DESIRED RITZ VECTORS.  AN EDUCATED     HHL01070
C         GUESS AT AN APPROPRIATE LENGTH CAN BE OBTAINED BY RUNNING THE HHL01080
C         PROGRAM WITH THE FLAG MBOUND = 1 AND MULTIPLYING THE          HHL01090
C         RESULTING SIZE BY 5/4.                                        HHL01100
C     7.  GOODEV, EVNEW, AMINGP, TMINGP, TERR, ERR, ERRGDP, RNORM, TBETAHHL01110
C         TLAST, MP, MA, M1, M2, MINT, MFIN, MULEVA, AND IDELTA ALL     HHL01120
C         MUST BE AT LEAST NGOOD.                                       HHL01130
C                                                                       HHL01140
C-----------------------------------------------------------------------HHL01150
C     OUTPUT HEADER                                                     HHL01160
      WRITE(6,10)                                                       HHL01170
   10 FORMAT(/' LANCZOS EIGENVECTOR PROCEDURE FOR HERMITIAN MATRICES'/) HHL01180
C                                                                       HHL01190
C     SET PROGRAM PARAMETERS                                            HHL01200
C     USER MUST NOT MODIFY SCALE0                                       HHL01210
      SCALE0 = 5.0D0                                                    HHL01220
      ZERO = 0.0D0                                                      HHL01230
      ZEROC = DCMPLX(ZERO,ZERO)                                         HHL01240
      ONE = 1.0D0                                                       HHL01250
      MPMIN = -1000                                                     HHL01260
C     CONVERGENCE TOLERANCE FOR T-EIGENVECTORS FOR RITZ VECTORS         HHL01270
      ERTOL = 1.D-10                                                    HHL01280
      ISREAL = 0                                                        HHL01290
C                                                                       HHL01300
C     READ USER-SPECIFIED PARAMETER FROM INPUT FILE 5 (FREE FORMAT)     HHL01310
C                                                                       HHL01320
C     READ USER-PROVIDED HEADER FOR RUN                                 HHL01330
      READ(5,20) EXPLAN                                                 HHL01340
      WRITE(6,20) EXPLAN                                                HHL01350
   20 FORMAT(20A4)                                                      HHL01360
C                                                                       HHL01370
C     READ IN THE MAXIMUM PERMISSIBLE DIMENSIONS FOR THE TVEC ARRAY     HHL01380
C     (MDIMTV), FOR THE RITVEC ARRAY (MDIMRV), AND FOR THE BETA         HHL01390
C     ARRAY (MBETA).                                                    HHL01400
C                                                                       HHL01410
      READ(5,20) EXPLAN                                                 HHL01420
      READ(5,*) MDIMTV, MDIMRV, MBETA                                   HHL01430
C                                                                       HHL01440
C     READ IN RELATIVE TOLERANCE (RELTOL) USED IN DETERMINING           HHL01450
C     APPROPRIATE SIZES FOR THE T-MATRICES USED IN THE EIGENVECTOR      HHL01460
C     COMPUTATIONS                                                      HHL01470
C                                                                       HHL01480
      READ(5,20) EXPLAN                                                 HHL01490
      READ(5,*) RELTOL                                                  HHL01500
C                                                                       HHL01510
C                                                                       HHL01520
C     SET FLAGS TO 0 OR 1:                                              HHL01530
```

```
C      MBOUND = 1:  PROGRAM TERMINATES AFTER COMPUTING 1ST GUESSES        HHL01540
C                   ON APPROPRIATE T-SIZES FOR USE IN THE RITZ VECTOR     HHL01550
C                   COMPUTATIONS                                          HHL01560
C      NTVCON = 0:  PROGRAM TERMINATES IF THE TVEC ARRAY IS NOT           HHL01570
C                   LARGE ENOUGH TO HOLD ALL THE T-EIGENVECTORS REQUIRED. HHL01580
C      SVTVEC = 0:  THE T-EIGENVECTORS ARE NOT WRITTEN TO FILE 11         HHL01590
C                   UNLESS TVSTOP = 1                                     HHL01600
C      SVTVEC = 1:  WRITE THE T-EIGENVECTORS TO FILE 11.                  HHL01610
C      TVSTOP = 1:  PROGRAM TERMINATES AFTER COMPUTING THE                HHL01620
C                   T-EIGENVECTORS                                        HHL01630
C      LVCONT = 0:  PROGRAM TERMINATES IF THE NUMBER OF T-EIGENVECTORS    HHL01640
C                   COMPUTED IS NOT EQUAL TO THE NUMBER OF RITZ           HHL01650
C                   VECTORS REQUESTED.                                    HHL01660
C      ERCONT = 0:  MEANS FOR ANY GIVEN EIGENVALUE, A RITZ VECTOR         HHL01670
C                   WILL NOT BE COMPUTED FOR THAT EIGENVALUE UNLESS       HHL01680
C                   A T-EIGENVECTOR HAS BEEN IDENTIFIED WITH A LAST       HHL01690
C                   COMPONENT WHICH SATISFIES THE SPECIFIED               HHL01700
C                   CONVERGENCE CRITERION.                                HHL01710
C      ERCONT = 1:  MEANS FOR ANY GIVEN EIGENVALUE, A RITZ VECTOR         HHL01720
C                   WILL BE COMPUTED.  IF A T-EIGENVECTOR CANNOT          HHL01730
C                   BE IDENTIFIED WHICH SATISFIES THE LAST               HHL01740
C                   COMPONENT CRITERION, THEN THE PROGRAM WILL            HHL01750
C                   USE THE T-VECTOR THAT CAME CLOSEST TO                 HHL01760
C                   SATISFYING THE CRITERION                              HHL01770
C      IWRITE = 1:  EXTENDED OUTPUT OF INTERMEDIATE COMPUTATIONS          HHL01780
C                   IS WRITTEN TO FILE 6                                  HHL01790
C      IREAD = 0:   ALPHA/BETA FILE IS REGENERATED                        HHL01800
C      IREAD = 1:   ALPHA/BETA FILE USED IN EIGENVALUE COMPUTATIONS       HHL01810
C                   IS READ IN AND EXTENDED IF NECESSARY.  IN BOTH        HHL01820
C                   CASES IREAD = 0 OR 1, THE LANCZOS VECTORS ARE         HHL01830
C                   ALWAYS REGENERATED FOR THE RITZ VECTOR                HHL01840
C                   COMPUTATIONS                                          HHL01850
C                                                                         HHL01860
       READ(5,20) EXPLAN                                                  HHL01870
       READ(5,*) MBOUND,NTVCON,SVTVEC,IREAD                               HHL01880
C                                                                         HHL01890
       READ(5,20) EXPLAN                                                  HHL01900
       READ(5,*) TVSTOP,LVCONT,ERCONT,IWRITE                             HHL01910
       IF (TVSTOP.EQ.1) SVTVEC = 1                                        HHL01920
C                                                                         HHL01930
C      READ IN SEED FOR GENERATING RANDOM STARTING VECTOR FOR THE         HHL01940
C      INVERSE ITERATION ON THE T-MATRICES.                              HHL01950
C                                                                         HHL01960
       READ(5,20) EXPLAN                                                  HHL01970
       READ(5,*) RHSEED                                                   HHL01980
C                                                                         HHL01990
C      READ IN MATNO = MATRIX/RUN IDENTIFICATION NUMBER AND               HHL02000
C      N = ORDER OF A-MATRIX                                              HHL02010
C                                                                         HHL02020
       READ(5,20) EXPLAN                                                  HHL02030
       READ(5,*) MATNO,N                                                  HHL02040
C      IF MATNO < 0, THEN MATRIX SUPPLIED IS REAL AND USER WANTS TO       HHL02050
C      CHECK ON THE T-MULTIPLICITY OF THE EIGENVALUES OF GIVEN MATRIX     HHL02060
       IF(MATNO.GT.0) GO TO 30                                            HHL02070
       ISREAL = 1                                                         HHL02080
```

```
      MATNO = - MATNO                                            HHL02090
   30 CONTINUE                                                   HHL02100
C                                                                HHL02110
C-----------------------------------------------------------------HHL02120
C     INITIALIZE THE ARRAYS FOR THE USER-SPECIFIED MATRIX        HHL02130
C     AND PASS THE STORAGE LOCATIONS OF THESE ARRAYS TO THE      HHL02140
C     MATRIX-VECTOR MULTIPLY SUBROUTINE CMATV.                   HHL02150
C                                                                HHL02160
      CALL USPEC(N,MATNO)                                        HHL02170
C                                                                HHL02180
C-----------------------------------------------------------------HHL02190
C     MASK UNDERFLOW AND OVERFLOW                                HHL02200
      CALL MASK                                                  HHL02210
C                                                                HHL02220
C-----------------------------------------------------------------HHL02230
C                                                                HHL02240
C     WRITE RUN PARAMETERS OUT TO FILE 6                         HHL02250
C                                                                HHL02260
      WRITE(6,40) MATNO,N                                        HHL02270
   40 FORMAT(/' MATRIX IDENTIFICATION NO. = ',I10,' ORDER OF A = ',I5) HHL02280
C                                                                HHL02290
      WRITE(6,50) MBOUND,NTVCON,SVTVEC,IREAD                     HHL02300
   50 FORMAT(/3X,'MBOUND',3X,'NTVCON',3X,'SVTVEC',3X,'IREAD'/3I9,I8) HHL02310
C                                                                HHL02320
      WRITE(6,60) TVSTOP,LVCONT,ERCONT,IWRITE                    HHL02330
   60 FORMAT(/3X,'TVSTOP',3X,'LVCONT',3X,'ERCONT',3X,'IWRITE'/4I9) HHL02340
C                                                                HHL02350
      WRITE(6,70) MDIMTV,MDIMRV,MBETA                            HHL02360
   70 FORMAT(/3X,'MDIMTV',3X,'MDIMRV',3X,'MBETA'/2I9,I8)         HHL02370
C                                                                HHL02380
      WRITE(6,80) RELTOL,RHSEED                                  HHL02390
   80 FORMAT(/7X,'RELTOL',3X,'RHSEED'/E13.4,I9)                  HHL02400
C                                                                HHL02410
C                                                                HHL02420
C     FROM FILE 3 READ IN THE NUMBER OF EIGENVALUES (NGOOD) FOR WHICH HHL02430
C     EIGENVECTORS ARE REQUESTED, THE ORDER (MEV) OF THE LANCZOS HHL02440
C     TRIDIAGONAL MATRIX USED IN COMPUTING THESE EIGENVALUES, THE HHL02450
C     ORDER (NOLD) OF THE USER-SPECIFIED MATRIX USED IN THE EIGENVALUE HHL02460
C     COMPUTATIONS, THE SEED (SVSEED) USED FOR GENERATING THE STARTING HHL02470
C     VECTOR THAT WAS USED IN THOSE LANCZOS EIGENVALUE COMPUTATIONS, HHL02480
C     AND THE MATRIX/RUN IDENTIFICATION NUMBER (MATOLD) USED IN THOSE HHL02490
C     COMPUTATIONS.  ALSO READ IN THE NUMBER (NDIS) OF DISTINCT  HHL02500
C     EIGENVALUES OF T(1,MEV) THAT WERE COMPUTED BUT THIS VALUE IS HHL02510
C     NOT USED IN THE EIGENVECTOR COMPUTATIONS.                  HHL02520
C                                                                HHL02530
      READ(3,90) NGOOD,NDIS,MEV,NOLD,SVSEED,MATOLD               HHL02540
   90 FORMAT(4I6,I12,I8)                                         HHL02550
C                                                                HHL02560
C     READ IN THE T-MULTIPLICITY TOLERANCE USED IN THE BISEC SUBROUTINE HHL02570
C     DURING THE COMPUTATION OF THE GIVEN EIGENVALUES.           HHL02580
C     ALSO READ IN THE FLAG IB.  IF IB < 0, THEN SOME BETA(I) IN THE HHL02590
C     T-MATRIX FILE PROVIDED ON FILE 2 FAILED THE ORTHOGONALITY  HHL02600
C     TEST IN THE TNORM SUBROUTINE.  USER SHOULD NOTE THAT THIS  HHL02610
C     VECTOR PROGRAM PROCEEDS INDEPENDENTLY OF THE SIZE OF THE BETA. HHL02620
C                                                                HHL02630
```

```
      READ(3,100) MULTOL,IB,BTOL                                    HHL02640
  100 FORMAT(E20.12,I6,E13.4)                                       HHL02650
C                                                                   HHL02660
      TEMP = DFLOAT(MEV+1000)                                       HHL02670
      TTOL = MULTOL/TEMP                                            HHL02680
      WRITE(6,110) MULTOL,TTOL                                      HHL02690
  110 FORMAT(/' T-MULTIPLICITY TOLERANCE USED IN THE EIGENVALUE COMPUTATHHL02700
     1IONS WAS',E13.4/' SCALED MACHINE EPSILON IS',E13.4)           HHL02710
C                                                                   HHL02720
C     CONTINUE WRITE TO FILE 6 OF THE PARAMETERS FOR THIS RUN       HHL02730
C                                                                   HHL02740
      WRITE(6,120)NGOOD,NDIS,MEV,NOLD,MATOLD,SVSEED,MULTOL,IB,BTOL  HHL02750
  120 FORMAT(/' EIGENVALUES SUPPLIED ARE READ IN FROM FILE 3'/' FILE 3 HHL02760
     1HEADER IS'/4X,'NG',2X,'NDIS',3X,'MEV',2X,'NOLD',2X,'MATOLD',4X, HHL02770
     1'SVSEED',6X,'MULTOL',6X,'IB',9X,'BTOL'/4I6,I8,I10,E12.3,I8,E13.4/)HHL02780
C                                                                   HHL02790
C     IS THE ARRAY RITVEC LONG ENOUGH TO HOLD ALL OF THE DESIRED    HHL02800
C     RITZ VECTORS (APPROXIMATE EIGENVECTORS)?                      HHL02810
      NMAX = NGOOD*N                                                HHL02820
      IF(MBOUND.NE.0) GO TO 130                                     HHL02830
      IF(TVSTOP.NE.1.AND.NMAX.GT.MDIMRV) GO TO 1390                 HHL02840
C                                                                   HHL02850
C     CHECK THAT THE ORDER N AND THE MATRIX IDENTIFICATION NUMBER   HHL02860
C     MATNO SPECIFIED BY THE USER AGREE WITH THOSE READ IN FROM     HHL02870
C     FILE 3.                                                       HHL02880
  130 ITEMP = (NOLD-N)**2+(MATOLD-MATNO)**2                         HHL02890
      IF (ITEMP.NE.0) GO TO 1410                                    HHL02900
C                                                                   HHL02910
C                                                                   HHL02920
C     THE LANCZOS EIGENVECTOR COMPUTATIONS ASSUME THAT EACH         HHL02930
C     EIGENVALUE THAT IS BEING CONSIDERED HAS CONVERGED AS AN       HHL02940
C     EIGENVALUE OF THE LANCZOS TRIDIAGONAL MATRICES.               HHL02950
C                                                                   HHL02960
C     READ IN FROM FILE 3, THE T-MULTIPLICITIES OF THE EIGENVALUES  HHL02970
C     WHOSE EIGENVECTORS ARE TO BE COMPUTED, THE VALUES OF THESE    HHL02980
C     EIGENVALUES AND THEIR MINIMAL GAPS AS EIGENVALUES OF THE      HHL02990
C     USER-SPECIFIED MATRIX AND AS EIGENVALUES OF THE T-MATRIX.     HHL03000
C                                                                   HHL03010
      READ(3,20) EXPLAN                                             HHL03020
      READ(3,140) (MP(J),GOODEV(J),TMINGP(J),AMINGP(J), J=1,NGOOD)  HHL03030
  140 FORMAT(6X,I6,E25.16,2E14.3)                                   HHL03040
C                                                                   HHL03050
      WRITE(6,150) (J,GOODEV(J),MP(J),TMINGP(J),AMINGP(J), J=1,NGOOD)HHL03060
  150 FORMAT(/' EIGENVALUES READ IN, T-MULTIPLICITIES, T-GAPS AND A-GAPSHHL03070
     1 '/4X,' J ',5X,'GOOD EIGENVALUE',5X,'MULT',4X,' TMINGAP ',4X, HHL03080
     1'  AMINGAP '/(I6,E25.16,I4,2E15.4))                           HHL03090
C                                                                   HHL03100
C     READ IN ERROR ESTIMATES                                       HHL03110
      WRITE(6,180) MEV,SVSEED                                       HHL03120
C     CHECK WHETHER OR NOT THERE ARE ANY ISOLATED T-EIGENVALUES IN  HHL03130
C     THE EIGENVALUES PROVIDED                                      HHL03140
      DO 160 J=1,NGOOD                                              HHL03150
      IF(MP(J).EQ.1) GO TO 170                                      HHL03160
  160 CONTINUE                                                      HHL03170
      GO TO 200                                                     HHL03180
```

```
  170 READ(4,20) EXPLAN                                         HHL03190
      READ(4,20) EXPLAN                                         HHL03200
      READ(4,20) EXPLAN                                         HHL03210
  180 FORMAT(/' THESE EIGENVALUES WERE COMPUTED USING A T-MATRIX OF  HHL03220
     1ORDER ',I5/' AND SEED FOR RANDOM NUMBER GENERATOR =',I12)  HHL03230
      READ(4,190) NISO                                          HHL03240
  190 FORMAT(18X,I6)                                            HHL03250
      READ(4,20) EXPLAN                                         HHL03260
      READ(4,20) EXPLAN                                         HHL03270
      READ(4,20) EXPLAN                                         HHL03280
  200 DO 230 J=1,NGOOD                                          HHL03290
      ERR(J) = 0.D0                                             HHL03300
      IF(MP(J).NE.1) GO TO 230                                  HHL03310
      READ(4,210) EVAL, ERR(J)                                  HHL03320
  210 FORMAT(10X,E25.16,E14.3)                                  HHL03330
      IF(DABS(EVAL - GOODEV(J)).LT.1.D-10) GO TO 230            HHL03340
      WRITE(6,220) EVAL,GOODEV(J)                               HHL03350
  220 FORMAT(' PROBLEM WITH READ IN OF ERROR ESTIMATES'/' EIGENVALUE REAHHL03360
     1D IN',E20.12,' DOES NOT MATCH GOODEV(J) ='/E20.12)        HHL03370
      GO TO 1630                                                HHL03380
C                                                               HHL03390
  230 CONTINUE                                                  HHL03400
C                                                               HHL03410
      WRITE(6,240) (J,GOODEV(J),ERR(J), J=1,NGOOD)              HHL03420
  240 FORMAT(' ERROR ESTIMATES ='/4X,' J',5X,'EIGENVALUE',10X,'ESTIMATE'HHL03430
     1 /(I6,E20.12,E14.3))                                      HHL03440
C                                                               HHL03450
      IF(IREAD.EQ.0) GO TO 340                                  HHL03460
C                                                               HHL03470
C     READ IN THE SIZE OF THE T-MATRIX PROVIDED ON FILE 2.  READ IN  HHL03480
C     THE ORDER OF THE USER-SPECIFIED MATRIX , THE SEED FOR THE  HHL03490
C     RANDOM NUMBER GENERATOR, AND THE MATRIX/TEST IDENTIFICATION  HHL03500
C     NUMBER THAT WERE USED IN THE LANCZOS EIGENVALUE COMPUTATIONS.  HHL03510
C     IF FLAG IREAD = 0, REGENERATE HISTORY.  HISTORY MUST BE    HHL03520
C     STORED IN HEXADECIMAL FORMAT TO AVOID ERRORS INCURRED IN   HHL03530
C     INPUT/OUTPUT CONVERSIONS.                                 HHL03540
C                                                               HHL03550
      READ(2,250) KMAX,NOLD,SVSOLD,MATOLD                       HHL03560
  250 FORMAT(2I6,I12,I8)                                        HHL03570
C                                                               HHL03580
      WRITE(6,260) KMAX,NOLD,SVSOLD,MATOLD                      HHL03590
  260 FORMAT(/' READ IN THE T-MATRICES STORED ON FILE 2'/' FILE 2 HEADERHHL03600
     1 IS'/2X,'KMAX',2X,'NOLD',6X,'SVSOLD',2X,'MATOLD'/2I6,I12,I8/)  HHL03610
C                                                               HHL03620
C     CHECK THAT THE ORDER, THE MATRIX/TEST IDENTIFICATION NUMBER  HHL03630
C     AND THE SEED FOR THE RANDOM NUMBER GENERATOR USED IN THE   HHL03640
C     LANCZOS COMPUTATIONS THAT GENERATED THE HISTORY FILE       HHL03650
C     BEING USED AGREE WITH WHAT THE USER HAS SPECIFIED.         HHL03660
      IF (NOLD.NE.N.OR.MATOLD.NE.MATNO.OR.SVSOLD.NE.SVSEED) GO TO 1430  HHL03670
C                                                               HHL03680
      KMAX1 = KMAX + 1                                          HHL03690
C                                                               HHL03700
C     READ IN THE T-MATRICES FROM FILE 2.  THESE ARE USED TO GENERATE  HHL03710
C     THE T-EIGENVECTORS THAT WILL BE USED IN THE RITZ VECTOR    HHL03720
C     COMPUTATIONS.  ALPHA/BETA HISTORY MUST BE STORED IN        HHL03730
```

```
C      MACHINE FORMAT, ((4Z20) FOR IBM/3081).                          HHL03740
C                                                                      HHL03750
       READ(2,270) (ALPHA(J), J=1,KMAX)                                HHL03760
       READ(2,270) (BETA(J), J=1,KMAX1)                                HHL03770
   270 FORMAT(4Z20)                                                    HHL03780
C                                                                      HHL03790
       READ(2,270) (V1(J), J=1,N)                                      HHL03800
       READ(2,270) (V2(J), J=1,N)                                      HHL03810
C                                                                      HHL03820
C      ENLARGE KMAX IF THE SIZE AT WHICH THE EIGENVALUE                HHL03830
C      COMPUTATIONS WERE PERFORMED IS ESSENTIALLY KMAX AND             HHL03840
C      THERE IS AT LEAST ONE EIGENVALUE THAT IS T-SIMPLE AND           HHL03850
C      T-ISOLATED, IN THE SENSE THAT IF ITS NEAREST T-NEIGHBOR IS TOO  HHL03860
C      CLOSE THAT NEIGHBOR IS A 'GOOD' T-EIGENVALUE.                   HHL03870
       DO 280 J = 1,NGOOD                                              HHL03880
       IF(MP(J).EQ.1) GO TO 300                                        HHL03890
   280 CONTINUE                                                        HHL03900
       WRITE(6,290)                                                    HHL03910
   290 FORMAT(/' ALL EIGENVALUES USED ARE T-MULTIPLE OR CLOSE TO SPURIOUSHHL03920
      1 T-EIGENVALUES'/' SO DO NOT CHANGE KMAX')                       HHL03930
       IF(KMAX.LT.MEV) GO TO 1450                                      HHL03940
       GO TO 320                                                       HHL03950
C                                                                      HHL03960
   300 KMAXN= 11*MEV/8 + 12                                            HHL03970
       IF(MBETA.LE.KMAXN) GO TO 1610                                   HHL03980
       IF(KMAX.GE.KMAXN )  GO TO 320                                   HHL03990
       WRITE(6,310) KMAX, KMAXN                                        HHL04000
   310 FORMAT(' ENLARGE KMAX FROM ',I6,' TO ',I6)                      HHL04010
       MOLD1 = KMAX + 1                                                HHL04020
       KMAX = KMAXN                                                    HHL04030
       GO TO 390                                                       HHL04040
C                                                                      HHL04050
   320 WRITE(6,330) KMAX                                               HHL04060
   330 FORMAT(/' T-MATRICES HAVE BEEN READ IN FROM FILE 2'/' THE LARGEST HHL04070
      1SIZE T-MATRIX ALLOWED IS',I6/)                                  HHL04080
C                                                                      HHL04090
       IF(IREAD.EQ.1) GO TO 410                                        HHL04100
C                                                                      HHL04110
C      REGENERATE THE ALPHA AND BETA                                   HHL04120
C                                                                      HHL04130
   340 MOLD1 = 1                                                       HHL04140
C                                                                      HHL04150
       DO 350 J = 1,NGOOD                                              HHL04160
       IF(MP(J).EQ.1) GO TO 370                                        HHL04170
   350 CONTINUE                                                        HHL04180
       KMAX = MEV + 12                                                 HHL04190
       WRITE(6,360) KMAX                                               HHL04200
   360 FORMAT(/' ALL EIGENVALUES FOR WHICH EIGENVECTORS ARE TO BE COMPUTEHHL04210
      1D ARE EITHER T-MULTIPLE OR CLOSE TO'/' A SPURIOUS EIGENVALUE. THERHHL04220
      1EFORE SET KMAX = MEV + 12 = ',I7)                               HHL04230
       GO TO 390                                                       HHL04240
C                                                                      HHL04250
   370 KMAXN = 11*MEV/8 + 12                                           HHL04260
       IF(MBETA.LE.KMAXN) GO TO 1610                                   HHL04270
       WRITE(6,380) KMAXN                                              HHL04280
```

```
  380 FORMAT(' SET KMAX EQUAL TO ',I6)                              HHL04290
      KMAX = KMAXN                                                  HHL04300
C                                                                   HHL04310
  390 WRITE(6,400) MOLD1,KMAX                                       HHL04320
  400 FORMAT(/' LANCZS SUBROUTINE GENERATES ALPHA(J), BETA(J+1), J =', HHL04330
     1 I6,' TO ', I6/)                                              HHL04340
C                                                                   HHL04350
C-----------------------------------------------------------------HHL04360
C                                                                   HHL04370
      CALL LANCZS(CMATV,V1,V2,VS,ALPHA,BETA,GR,GC,G,KMAX,MOLD1,N,SVSEED)HHL04380
C                                                                   HHL04390
C-----------------------------------------------------------------HHL04400
C                                                                   HHL04410
  410 CONTINUE                                                      HHL04420
C                                                                   HHL04430
C     THE SUBROUTINE STURMI DETERMINES THE SMALLEST SIZE T-MATRIX FOR  HHL04440
C     WHICH THE EIGENVALUE IN QUESTION IS AN EIGENVALUE (TO WITHIN A    HHL04450
C     GIVEN TOLERANCE) AND IF POSSIBLE THE SMALLEST SIZE T-MATRIX       HHL04460
C     FOR WHICH IT IS A DOUBLE EIGENVALUE (TO WITHIN THE SAME           HHL04470
C     TOLERANCE).  THE SIZE T-MATRIX USED IN THE EIGENVECTOR           HHL04480
C     COMPUTATIONS IS THEN DETERMINED BY LOOPING ON THE SIZES OF THE   HHL04490
C     T-EIGENVECTORS, USING THE INFORMATION FROM STURMI TO OBTAIN       HHL04500
C     STARTING GUESSES AT THE T-SIZES.                                 HHL04510
C                                                                   HHL04520
C                                                                   HHL04530
      STUTOL = SCALE0*MULTOL                                        HHL04540
      IF(IWRITE.EQ.1) WRITE(6,420)                                 HHL04550
  420 FORMAT(' FROM STURMI')                                        HHL04560
      DO 460 J = 1,NGOOD                                           HHL04570
      EVAL = GOODEV(J)                                             HHL04580
C     COMPUTE THE TOLERANCES USED BY STURMI TO DETERMINE AN INTERVAL   HHL04590
C     CONTAINING THE EIGENVALUE EVAL.                               HHL04600
      TEMP = DABS(EVAL)*RELTOL                                     HHL04610
      TOLN = DMAX1(TEMP,STUTOL)                                    HHL04620
C                                                                   HHL04630
C-----------------------------------------------------------------HHL04640
C                                                                   HHL04650
      CALL STURMI(ALPHA,BETA,EVAL,TOLN,EPSM,KMAX,MK1,MK2,IC,IWRITE) HHL04660
C                                                                   HHL04670
C-----------------------------------------------------------------HHL04680
C                                                                   HHL04690
C     STORE THE COMPUTED ORDERS OF T-MATRICES FOR LATER PRINTOUT       HHL04700
      M1(J) = MK1                                                  HHL04710
      M2(J) = MK2                                                  HHL04720
      ML(J) = (MK1 + 3*MK2)/4                                      HHL04730
      IF(MK2.EQ.KMAX)  ML(J) = KMAX                               HHL04740
C                                                                   HHL04750
      IF(IC.GT.0) GO TO 440                                        HHL04760
C     IC = 0 MEANS THERE WAS NO T-EIGENVALUE IN THE DESIGNATED INTERVAL HHL04770
C     BY T-SIZE KMAX.  THIS MEANS THAT THE T-EIGENVALUE PROVIDED HAS    HHL04780
C     NOT YET CONVERGED AS AN EIGENVALUE OF THE TRIDIAGONAL MATRICES    HHL04790
C     SO PROGRAM SHOULD NOT COMPUTE ITS EIGENVECTOR.                    HHL04800
      WRITE(6,430) J,GOODEV(J),MK1,MK2                            HHL04810
  430 FORMAT(I6,'TH EIGENVALUE',E20.12,' HAS NOT CONVERGED '/      HHL04820
     1' SO DO NOT COMPUTE ANY T-EIGENVECTOR OR RITZ VECTOR FOR IT' HHL04830
```

```
      1/' MK1 AND MK2 FOR THIS EIGENVALUE WERE',2I6)              HHL04840
          MP(J) = MPMIN                                           HHL04850
          MA(J) = -2*KMAX                                         HHL04860
          GO TO 460                                               HHL04870
C         COMPUTE AN APPROPRIATE SIZE T-MATRIX FOR THE GIVEN EIGENVALUE. HHL04880
      440 IF(M2(J).EQ.KMAX) GO TO 450                             HHL04890
C         M1 AND M2 WERE BOTH DETERMINED                          HHL04900
          MA(J) = (3*M1(J) + M2(J))/4  + 1                        HHL04910
          GO TO 460                                               HHL04920
C         M2 NOT DETERMINED                                       HHL04930
      450 MA(J) = 5*M1(J)/4  + 1                                  HHL04940
C                                                                 HHL04950
      460 CONTINUE                                                HHL04960
C                                                                 HHL04970
          IF (IWRITE.EQ.1) WRITE(6,470) (MA(JJ), JJ=1,NGOOD)      HHL04980
      470 FORMAT(/' 1ST GUESS AT APPROPRIATE SIZE T-MATRICES'/    HHL04990
        1 ' ACTUAL VALUES WILL PROBABLY BE 1/4 AGAIN AS MUCH'/(13I6)) HHL05000
C                                                                 HHL05010
C         PRINT OUT TO FILE 10 1ST GUESSES AT SIZES OF T-MATRICES TO HHL05020
C         BE USED IN THE EIGENVECTOR COMPUTATIONS.                HHL05030
C         ACTUAL SIZES MAY BE 1/4 OR MORE LARGER THAN THESE SIZES. HHL05040
          WRITE(10,480) N,KMAX                                    HHL05050
      480 FORMAT(2I8,' = ORDER OF USER MATRIX AND MAX ORDER OF T(1,MEV)') HHL05060
C                                                                 HHL05070
          WRITE(10,490)                                           HHL05080
      490 FORMAT(/' 1ST GUESS AT APPROPRIATE SIZE T-MATRICES'/    HHL05090
        1 ' ACTUAL VALUES WILL PROBABLY BE 1/4 AGAIN AS MUCH'/)   HHL05100
C                                                                 HHL05110
          WRITE(10,500)                                           HHL05120
      500 FORMAT(4X,'J',7X,'GOODEV(J)',4X,'M1(J)',1X,'M2(J)',1X,'MA(J)') HHL05130
C                                                                 HHL05140
          WRITE(10,510) (J,GOODEV(J),M1(J),M2(J), MA(J), J=1,NGOOD) HHL05150
      510 FORMAT(I5,E19.12,3I6)                                   HHL05160
C                                                                 HHL05170
          IF(MBOUND.EQ.1) WRITE(10,520)                           HHL05180
      520 FORMAT(/' GOODEV(J) IS A GOOD EIGENVALUE OF T(1,MEV)'/  HHL05190
        1 ' M1 = SMALLEST VALUE OF M SUCH THAT T(1,M) HAS AT LEAST'/ HHL05200
        1 '      ONE EIGENVALUE IN THE INTERVAL (EV-TOLN,EV+TOLN)'/ HHL05210
        1 '      TOLN(J) = DMAX1(GOODEV(J)*RELTOL, SCALEO*MULTOL)'/ HHL05220
        1 ' M2 = SMALLEST M (IF ANY) SUCH THAT IN THE ABOVE INTERVAL'/ HHL05230
        1 '      T(1,M) HAS AT LEAST TWO EIGENVALUES '/           HHL05240
        1 ' INITIAL VALUE OF MA(J) IS CHOSEN HEURISTICALLY'/      HHL05250
        1 ' PROGRAM LOOPS ON SIZE OF T-MATRIX TO GET BETTER SIZE'/ HHL05260
        1 ' END OF SIZES OF T-MATRICES FILE 10'///)              HHL05270
C                                                                 HHL05280
C                                                                 HHL05290
C         TERMINATE AFTER COMPUTING 1ST GUESSES AT  SIZES OF THE  HHL05300
C         T-MATRICES REQUIRED FOR THE GIVEN EIGENVALUES?          HHL05310
          IF(MBOUND.EQ.1) GO TO 1470                              HHL05320
C                                                                 HHL05330
C                                                                 HHL05340
C         IS THERE ROOM FOR ALL OF THE REQUESTED T-EIGENVECTORS?  HHL05350
          MTOL = 0                                                HHL05360
          DO 530 J = 1,NGOOD                                      HHL05370
          IF(MP(J).EQ.MPMIN) GO TO 530                            HHL05380
```

```
      MTOL = MTOL + IABS(MA(J))                                     HHL05390
  530 CONTINUE                                                      HHL05400
      MTOL = (5*MTOL)/4                                             HHL05410
      IF(MTOL.GT.MDIMTV.AND.NTVCON.EQ.0) GO TO 1490                 HHL05420
C                                                                   HHL05430
C-----------------------------------------------------------------HHL05440
C     GENERATE A RANDOM VECTOR TO BE USED REPEATEDLY BY            HHL05450
C     SUBROUTINE INVERM                                            HHL05460
C                                                                   HHL05470
      IIL = RHSEED                                                  HHL05480
      CALL GENRAN(IIL,G,KMAX)                                       HHL05490
C                                                                   HHL05500
C---------------------------------------------------------------- HHL05510
C                                                                   HHL05520
C     LOOP ON GIVEN EIGENVALUES TO COMPUTE THE CORRESPONDING        HHL05530
C     T-EIGENVECTOR.                                               HHL05540
C                                                                   HHL05550
      MTOL = 0                                                     HHL05560
      NTVEC = 0                                                    HHL05570
      ILBIS = 0                                                    HHL05580
      DO 720 J = 1,NGOOD                                           HHL05590
      ICOUNT = 0                                                   HHL05600
      ERRMIN = 10.D0                                               HHL05610
      MABEST = MPMIN                                               HHL05620
      IF(MP(J).EQ.MPMIN) GO TO 720                                 HHL05630
      TFLAG = 0                                                    HHL05640
      EVAL = GOODEV(J)                                             HHL05650
      TEMP = RELTOL* DABS(EVAL)                                    HHL05660
      UB = EVAL + DMAX1(STUTOL,TEMP)                               HHL05670
      LB = EVAL - DMAX1(STUTOL,TEMP)                               HHL05680
  540 KMAXU = IABS(MA(J))                                          HHL05690
C                                                                   HHL05700
C     SELECT A SUITABLE INCREMENT FOR THE ORDERS OF THE T-MATRICES  HHL05710
C     TO BE CONSIDERED IN DETERMINING APPROPRIATE SIZES FOR THE RITZ HHL05720
C     VECTOR COMPUTATIONS.                                         HHL05730
      IF(ICOUNT.GT.0) GO TO 560                                    HHL05740
C     SELECT IDELTA(J) BASED UPON THE T-MULTIPLICITY OBTAINED       HHL05750
      IF(M2(J).EQ.KMAX) GO TO 550                                  HHL05760
C     M2 DETERMINED                                                HHL05770
      IDELTA(J) = ((3*M1(J) + 5*M2(J))/8  +  1 - IABS(MA(J)))/10 + 1 HHL05780
      GO TO 560                                                    HHL05790
C     M2 NOT DETERMINED                                            HHL05800
  550 MAMAX = MIN0((11*MEV)/8 + 12, (13*M1(J))/8 + 1)              HHL05810
      IDELTA(J) = (MAMAX - IABS(MA(J)))/10  +  1                   HHL05820
  560 ICOUNT = ICOUNT + 1                                          HHL05830
C                                                                   HHL05840
C-----------------------------------------------------------------HHL05850
C     TO MIMIMIZE THE EFFECT OF THE ONE-SIDED ACCEPTANCE TEST FOR   HHL05860
C     T-EIGENVALUES IN THE BISEC SUBROUTINE, RECOMPUTE THE GIVEN    HHL05870
C     EIGENVALUE AT THE SPECIFIED KMAXU                            HHL05880
C                                                                   HHL05890
      CALL LBISEC(ALPHA,BETA,EPSM,EVAL,EVALN,LB,UB,TTOL,KMAXU,NEVT) HHL05900
C                                                                   HHL05910
C-----------------------------------------------------------------HHL05920
C                                                                   HHL05930
```

```
C     CHECK WHETHER OR NOT GIVEN T-MATRIX HAS AN EIGENVALUE IN THE       HHL05940
C     SPECIFIED INTERVAL AND IF SO WHAT ITS T-MULTIPLICITY IS.           HHL05950
C                                                                        HHL05960
      IF(NEVT.EQ.1) GO TO 600                                            HHL05970
      IF(NEVT.NE.0) GO TO 580                                            HHL05980
      ILBIS = 1                                                          HHL05990
      WRITE(6,570) EVAL,KMAXU                                            HHL06000
  570 FORMAT(/' PROBLEM ENCOUNTERED IN RECOMPUTATION OF USER-SUPPLIED EIHHL06010
     1GENVALUE',E20.12/' THE SIZE T-MATRIX SPECIFIED',I6,' DOES NOT      HHL06020
     1HAVE AN EIGENVALUE IN THE INTERVAL SPECIFIED'/' THEREFORE NO EIGENHHL06030
     1VECTOR WILL BE COMPUTED FOR THIS PARTICULAR EIGENVALUE'/)          HHL06040
      GO TO 620                                                          HHL06050
C                                                                        HHL06060
  580 IF(NEVT.GT.1)  WRITE(6,590) EVAL,KMAXU                             HHL06070
  590 FORMAT(/' PROBLEM ENCOUNTERED IN RECOMPUTATION OF USER-SUPPLIED    HHL06080
     1EIGENVALUE',E20.12/' FOR THE SIZE T-MATRIX SPECIFIED =',I6,' THE   HHL06090
     1GIVEN EIGENVALUE IS T-MULTIPLE IN THE INTERVAL SPECIFIED'/' SOMETHHHL06100
     1ING IS WRONG, THEREFORE NO EIGENVECTOR WILL BE COMPUTED FOR THIS EHHL06110
     1IGENVALUE'/)                                                       HHL06120
C                                                                        HHL06130
      MP(J) = MPMIN                                                      HHL06140
      MA(J) = -2*KMAX                                                    HHL06150
      GO TO 720                                                          HHL06160
C                                                                        HHL06170
  600 CONTINUE                                                           HHL06180
      ILBIS = 0                                                          HHL06190
C                                                                        HHL06200
      EVNEW(J) = EVALN                                                   HHL06210
      EVAL = EVALN                                                       HHL06220
      MTOL = MTOL+KMAXU                                                  HHL06230
C                                                                        HHL06240
C     IS THERE ROOM IN TVEC ARRAY FOR THE NEXT T-EIGENVECTOR?           HHL06250
C     IF NOT, SKIP TO RITZ VECTOR COMPUTATIONS.                          HHL06260
      IF (MTOL.GT.MDIMTV) GO TO 730                                      HHL06270
C                                                                        HHL06280
      IT = 3                                                             HHL06290
      KINT = MTOL - KMAXU +1                                             HHL06300
C                                                                        HHL06310
C     RECORD THE BEGINNING AND END OF THE T-EIGENVECTOR BEING COMPUTED  HHL06320
      MINT(J) = KINT                                                     HHL06330
      MFIN(J) = MTOL                                                     HHL06340
C                                                                        HHL06350
C------------------------------------------------------------------------HHL06360
C     SUBROUTINE INVERM DOES INVERSE ITERATION, I.E. SOLVES             HHL06370
C     (T(1,KMAXU) - EVAL)*U = RHS  FOR EACH EIGENVALUE TO OBTAIN THE     HHL06380
C     DESIRED T-EIGENVECTOR.                                            HHL06390
C                                                                        HHL06400
      IF(IWRITE.EQ.1)  WRITE(6,610) J                                    HHL06410
  610 FORMAT(/I6,'TH EIGENVALUE')                                        HHL06420
C                                                                        HHL06430
      CALL INVERM(ALPHA,BETA,V1,TVEC(KINT),EVAL,ERROR,TERROR,EPSM,       HHL06440
     1 G,KMAXU,IT,IWRITE)                                                HHL06450
C                                                                        HHL06460
C------------------------------------------------------------------------HHL06470
C                                                                        HHL06480
```

```
       TERR(J) = TERROR                                               HHL06490
       TLAST(J) = ERROR                                               HHL06500
       KMAXU1 = KMAXU + 1                                             HHL06510
       TBETA(J) = BETA(KMAXU1)*ERROR                                  HHL06520
C                                                                     HHL06530
C      AFTER COMPUTING EACH OF THE T-EIGENVECTORS,                    HHL06540
C      CHECK THE SIZE OF THE ERROR ESTIMATE, ERROR.                   HHL06550
C      IF THIS ESTIMATE IS NOT AS SMALL AS DESIRED AND                HHL06560
C      |MA(J)| < ML(J), ATTEMPT TO INCREASE THE SIZE OF |MA(J)|       HHL06570
C      AND REPEAT THE T-EIGENVECTOR COMPUTATIONS.                     HHL06580
C                                                                     HHL06590
       IF(ERROR.LT.ERTOL.OR.TFLAG.EQ.1)  GO TO 710                    HHL06600
C                                                                     HHL06610
       IF(ERROR.GE.ERRMIN) GO TO 620                                  HHL06620
C      LAST COMPONENT IS LESS THAN MINIMAL TO DATE                    HHL06630
       ERRMIN = ERROR                                                 HHL06640
       MABEST = MA(J)                                                 HHL06650
   620 CONTINUE                                                       HHL06660
C                                                                     HHL06670
       IF(MA(J).GT.0)  ITEST = MA(J) + IDELTA(J)                      HHL06680
       IF(MA(J).LT.0)  ITEST = -(IABS(MA(J)) + IDELTA(J))             HHL06690
       IF(IABS(ITEST).LE.ML(J).AND.ICOUNT.LE.10) GO TO 640            HHL06700
C      NEW MA(J) IS GREATER THAN MAXIMUM ALLOWED.                     HHL06710
       IF(ERCONT.EQ.0.OR.MABEST.EQ.MPMIN) GO TO 660                   HHL06720
       TFLAG = 1                                                      HHL06730
       MA(J) = MABEST                                                 HHL06740
       IF(ILBIS.EQ.0)  MTOL = MTOL - KMAXU                            HHL06750
       WRITE(6,630) MA(J)                                             HHL06760
   630 FORMAT(' 10 ORDERS WERE CONSIDERED.  NONE SATISFIED THE ERROR TESTHHL06770
      1'/' THEREFORE USE THE BEST ORDER OBTAINED FOR THE EIGENVECTORS' HHL06780
      1,I6)                                                           HHL06790
       GO TO 540                                                      HHL06800
C                                                                     HHL06810
   640 MA(J) = ITEST                                                  HHL06820
C                                                                     HHL06830
       MT = IABS(MA(J))                                               HHL06840
       IF(IWRITE.EQ.1) WRITE(6,650)  MT                               HHL06850
   650 FORMAT(/' CHANGE SIZE OF T-MATRIX TO ',I6,' RECOMPUTE T-EIGENVECTOHHL06860
      1R')                                                            HHL06870
C                                                                     HHL06880
       IF(ILBIS.EQ.0)  MTOL = MTOL - KMAXU                            HHL06890
C                                                                     HHL06900
       GO TO 540                                                      HHL06910
C                                                                     HHL06920
C         APPROPRIATE SIZE T-MATRIX WAS NOT OBTAINED                  HHL06930
   660 CONTINUE                                                       HHL06940
       WRITE(10,670) J,EVAL,MP(J)                                     HHL06950
   670 FORMAT(/' ON 10 INCREMENTS NOT ABLE TO IDENTIFY APPROPRIATE SIZE HHL06960
      1T-MATRIX FOR'/                                                 HHL06970
      1' EIGENVALUE(',I4,') = ',E20.12,' T-MULTIPLICITY =',I4/)       HHL06980
       IF(M2(J).EQ.KMAX) WRITE(10,680)                                HHL06990
       IF(M2(J).LT.KMAX) WRITE(10,690)                                HHL07000
   680 FORMAT(/' ORDERS TESTED RANGED FROM 5*M1(J)/4 TO APPROXIMATELY'/ HHL07010
      1'   MIN(11*MEV/8, 13*M1(J)/8)'/)                               HHL07020
   690 FORMAT(/' ORDERS TESTED RANGED FROM (3*M1(J)+M2(J)/4 TO APPROXIMATHHL07030
```

```
      1ELY'/'  (3*M1(J) + 5*M2(J))/8'/)                       HHL07040
         WRITE(10,700)                                        HHL07050
     700 FORMAT(' ALLOWING LARGER ORDERS FOR THE T-MATRICES MAY RESULT IN  HHL07060
      1 SUCCESS'/' BUT PROBABLY WILL NOT.  PROBLEM IS PROBABLY DUE TO'  HHL07070
      1 /' LACK OF CONVERGENCE OF GIVEN EIGENVALUE, CHECK THE ERROR ESTIMHHL07080
      1ATE')                                                  HHL07090
         MP(J) = MPMIN                                        HHL07100
         IF(ILBIS.EQ.0) MTOL = MTOL - KMAXU                   HHL07110
         GO TO 720                                            HHL07120
     710 NTVEC = NTVEC + 1                                    HHL07130
C                                                             HHL07140
     720 CONTINUE                                             HHL07150
         NGOODC = NGOOD                                       HHL07160
         GO TO 750                                            HHL07170
C                                                             HHL07180
C     COME HERE IF THERE IS NOT ENOUGH ROOM FOR ALL OF T-EIGENVECTORS  HHL07190
     730 NGOODC = J-1                                         HHL07200
         WRITE(6,740) J,MTOL,MDIMTV                           HHL07210
     740 FORMAT(/' NOT ENOUGH ROOM IN TVEC ARRAY FOR ',I4,'TH T-EIGENVECTORHHL07220
      1'/'  TVEC DIMENSION REQUESTED = ',I6,' BUT TVEC HAS DIMENSION ',I6HHL07230
      1/)                                                     HHL07240
         IF(NGOODC.EQ.0)  GO TO 1510                          HHL07250
         MTOL = MTOL-KMAXU                                    HHL07260
C                                                             HHL07270
     750 CONTINUE                                             HHL07280
C                                                             HHL07290
C     THE LOOP ON T-EIGENVECTOR COMPUTATIONS IS COMPLETE.     HHL07300
C     WRITE OUT THE SIZE T-MATRICES THAT WILL BE USED FOR     HHL07310
C     THE RITZ VECTOR COMPUTATIONS.                           HHL07320
C                                                             HHL07330
         WRITE(10,760)                                        HHL07340
     760 FORMAT(/' SIZES OF T-MATRICES THAT WILL BE USED IN THE RITZ COMPUTHHL07350
      1ATIONS'/5X,'J',16X,'GOODEV(J)',1X,'MA(J)')             HHL07360
C                                                             HHL07370
         WRITE(10,770)   (J,GOODEV(J),MA(J), J=1,NGOOD)       HHL07380
     770 FORMAT(I6,E25.14,I6)                                 HHL07390
         WRITE(10,520)                                        HHL07400
C                                                             HHL07410
         WRITE(6,780) MTOL                                    HHL07420
     780 FORMAT(/' THE CUMULATIVE LENGTH OF THE T-EIGENVECTORS IS',I18)  HHL07430
C                                                             HHL07440
         WRITE(6,790) NTVEC,NGOOD                             HHL07450
     790 FORMAT(/I6,' T-EIGENVECTORS OUT OF',I6,' REQUESTED WERE COMPUTED')HHL07460
C                                                             HHL07470
C     SAVE THE T-EIGENVECTORS ON FILE 11?                     HHL07480
         IF(TVSTOP.NE.1.AND.SVTVEC.EQ.0) GO TO 850            HHL07490
C                                                             HHL07500
         WRITE(11,800) NTVEC,MTOL,MATNO,SVSEED                HHL07510
     800 FORMAT(I6,3I12,' = NTVEC,MTOL,MATNO,SVSEED')          HHL07520
C                                                             HHL07530
         DO 830 J=1,NGOODC                                    HHL07540
C     IF MP(J) = MPMIN THEN NO SUITABLE T-EIGENVECTOR IS AVAILABLE  HHL07550
C     FOR THAT EIGENVALUE.                                    HHL07560
         IF(MP(J).EQ.MPMIN) WRITE(11,810) J,MA(J),GOODEV(J),MP(J)  HHL07570
     810 FORMAT(2I6,E20.12,I6/' TH EIGVAL,T-SIZE,EVALUE,FLAG,NO EIGVEC')  HHL07580
```

```
       IF(MP(J).NE.MPMIN) WRITE(11,820) J,MA(J),GOODEV(J),MP(J)        HHL07590
   820 FORMAT(I6,I6,E20.12,I6/' T-EIGVEC,SIZE T,EVALUE OF A,MP(J)')     HHL07600
       IF(MP(J).EQ.MPMIN) GO TO 830                                     HHL07610
       KI = MINT(J)                                                     HHL07620
       KF = MFIN(J)                                                     HHL07630
C                                                                       HHL07640
       WRITE(11,270) (TVEC(K), K=KI,KF)                                 HHL07650
C                                                                       HHL07660
   830 CONTINUE                                                         HHL07670
C                                                                       HHL07680
       IF(TVSTOP.NE.1) GO TO 850                                        HHL07690
C                                                                       HHL07700
       WRITE(6,840) TVSTOP, NTVEC,NGOOD                                 HHL07710
   840 FORMAT(/' USER SET TVSTOP = ',I1/                               HHL07720
      1' THEREFORE PROGRAM TERMINATES AFTER T-EIGENVECTOR COMPUTATIONS'/ HHL07730
      1' T-EIGENVECTORS THAT WERE COMPUTED ARE SAVED ON FILE 11'/       HHL07740
      1I8,' T-EIGENVECTORS WERE COMPUTED OUT OF',I7,' REQUESTED'/)      HHL07750
C                                                                       HHL07760
       GO TO 1630                                                       HHL07770
C                                                                       HHL07780
   850 CONTINUE                                                         HHL07790
C     IF NOT ABLE TO COMPUTE ALL THE REQUESTED T-EIGENVECTORS          HHL07800
C     CONTINUE WITH THE LANCZOS VECTOR COMPUTATIONS ANYWAY?            HHL07810
       IF(NTVEC.NE.NGOOD.AND.LVCONT.EQ.0) GO TO 1530                    HHL07820
C                                                                       HHL07830
C     COMPUTE THE MAXIMUM SIZE OF THE T-MATRIX USED FOR THOSE          HHL07840
C     EIGENVALUES WITH GOOD ERROR ESTIMATES.                          HHL07850
C                                                                       HHL07860
       KMAXU = 0                                                        HHL07870
       DO 860 J = 1,NGOODC                                              HHL07880
       MT = IABS(MA(J))                                                 HHL07890
       IF(MT.LT.KMAXU.OR.MP(J).EQ.MPMIN) GO TO 860                      HHL07900
       KMAXU = MT                                                       HHL07910
   860 CONTINUE                                                         HHL07920
C                                                                       HHL07930
       IF(KMAXU.EQ.0) GO TO 1570                                        HHL07940
C                                                                       HHL07950
       WRITE(6,870) KMAXU                                               HHL07960
   870 FORMAT(/I6,' = LARGEST SIZE T-MATRIX TO BE USED IN THE RITZ VECTORHHL07970
      1 COMPUTATIONS')                                                  HHL07980
C                                                                       HHL07990
C     COUNT THE NUMBER OF RITZ VECTORS NOT BEING COMPUTED             HHL08000
       MREJEC = 0                                                       HHL08010
       DO 880 J=1,NGOODC                                                HHL08020
   880 IF(MP(J).EQ.MPMIN)  MREJEC = MREJEC + 1                          HHL08030
       MREJET = MREJEC + (NGOOD-NGOODC)                                 HHL08040
       IF(MREJET.NE.0) WRITE(6,890) MREJET                             HHL08050
   890 FORMAT(/' RITZ VECTORS ARE NOT COMPUTED FOR',I6,' OF THE EIGENVALUHHL08060
      1ES'/)                                                            HHL08070
       NACT = NGOODC - MREJEC                                           HHL08080
       WRITE(6,900) NGOOD,NTVEC,NACT                                    HHL08090
   900 FORMAT(/I6,' RITZ VECTORS WERE REQUESTED'/I6,' T-EIGENVECTORS WEREHHL08100
      1 COMPUTED'/I6,' RITZ VECTORS WILL BE COMPUTED'/)                 HHL08110
C     CHECK IF THERE ARE ANY RITZ VECTORS TO COMPUTE                  HHL08120
       IF(MREJEC.EQ.NGOODC) GO TO 1550                                  HHL08130
```

```
C                                                                   HHL08140
C     CONTINUE WITH THE LANCZOS VECTOR COMPUTATIONS?                HHL08150
      IF(LVCONT.EQ.0.AND.MREJEC.NE.0) GO TO 1530                    HHL08160
C                                                                   HHL08170
C     NOW COMPUTE THE RITZ VECTORS.  REGENERATE THE                 HHL08180
C     LANCZOS VECTORS.                                              HHL08190
C                                                                   HHL08200
      DO 910 I = 1,NMAX                                             HHL08210
  910 RITVEC(I) = ZEROC                                             HHL08220
C                                                                   HHL08230
C     REGENERATE THE STARTING VECTOR. THIS MUST BE GENERATED AND    HHL08240
C     NORMALIZED PRECISELY THE WAY IT WAS DONE IN THE EIGENVALUE    HHL08250
C     COMPUTATIONS, OTHERWISE THERE WILL BE A MISMATCH BETWEEN      HHL08260
C     THE T-EIGENVECTORS THAT HAVE BEEN COMPUTED FROM THE T-MATRICES HHL08270
C     READ IN FROM FILE 2 AND THE LANCZOS VECTORS THAT ARE         HHL08280
C     BEING REGENERATED.                                            HHL08290
C                                                                   HHL08300
C-------------------------------------------------------------------HHL08310
C                                                                   HHL08320
      IIL = SVSEED                                                  HHL08330
      CALL GENRAN(IIL,G,N)                                          HHL08340
C                                                                   HHL08350
C-------------------------------------------------------------------HHL08360
C                                                                   HHL08370
      DO 920 I = 1,N                                                HHL08380
  920 GR(I) = G(I)                                                  HHL08390
C                                                                   HHL08400
C-------------------------------------------------------------------HHL08410
C                                                                   HHL08420
      CALL GENRAN(IIL,G,N)                                          HHL08430
C                                                                   HHL08440
C-------------------------------------------------------------------HHL08450
C                                                                   HHL08460
      DO 930 I = 1,N                                                HHL08470
  930 GC(I) = G(I)                                                  HHL08480
C                                                                   HHL08490
      DO 940 I = 1,N                                                HHL08500
  940 V2(I) = DCMPLX(GR(I),GC(I))                                   HHL08510
C                                                                   HHL08520
C-------------------------------------------------------------------HHL08530
      CALL CINPRD(V2,V2,SUM,N)                                      HHL08540
C-------------------------------------------------------------------HHL08550
C                                                                   HHL08560
      SUM = ONE/DSQRT(SUM)                                          HHL08570
      DO 950 I = 1,N                                                HHL08580
      V1(I) = ZEROC                                                 HHL08590
  950 V2(I) = V2(I)*SUM                                             HHL08600
C                                                                   HHL08610
C     LOOP FOR GENERATING REQUIRED RITZ VECTORS (IVEC = 1,KMAXU)    HHL08620
C     USES GRAM-SCHMIDT ORTHOGONALIZATION WITHOUT MODIFICATION      HHL08630
C                                                                   HHL08640
      IVEC = 1                                                      HHL08650
      BATA = ZERO                                                   HHL08660
C                                                                   HHL08670
      GO TO 1010                                                    HHL08680
```

```
C                                                              HHL08690
  960 CONTINUE                                                 HHL08700
C                                                              HHL08710
C------------------------------------------------------------- HHL08720
C    CMATV(V2,VS,SUM) CALCULATES  VS = A*V2 - SUM*VS           HHL08730
     SUM = ZERO                                                HHL08740
     CALL CMATV(V2,VS,SUM)                                     HHL08750
     CALL CINPRD(V2,VS,ALFA,N)                                 HHL08760
C                                                              HHL08770
C------------------------------------------------------------- HHL08780
C                                                              HHL08790
     DO 970 J=1,N                                              HHL08800
  970 V1(J) = (VS(J) - BATA*V1(J)) - ALFA*V2(J)                HHL08810
C                                                              HHL08820
C------------------------------------------------------------- HHL08830
     CALL CINPRD(V1,V1,BATA,N)                                 HHL08840
C------------------------------------------------------------- HHL08850
C                                                              HHL08860
     BATA = DSQRT(BATA)                                        HHL08870
     SUM = ONE/BATA                                            HHL08880
C                                                              HHL08890
     TEMP = BETA(IVEC)                                         HHL08900
     TEMP = DABS(BATA - TEMP)/TEMP                             HHL08910
     IF (TEMP.LT.1.0D-10)GO TO 990                             HHL08920
C                                                              HHL08930
C    THE BETA BEING REGENERATED DO NOT MATCH THE HISTORY FILE  HHL08940
C    SOMETHING IS WRONG IN THE LANCZOS VECTOR GENERATION       HHL08950
C    PROGRAM TERMINATES FOR USER TO CORRECT THE PROBLEM        HHL08960
C    WHICH MUST BE IN THE STARTING VECTOR GENERATION OR IN     HHL08970
C    THE MATRIX-VECTOR MULTIPLY SUBROUTINE CMATV SUPPLIED.     HHL08980
C    THIS SUBROUTINE MUST BE THE SAME ONE USED IN THE          HHL08990
C    EIGENVALUE COMPUTATIONS OR A MISMATCH WILL ENSUE.         HHL09000
C                                                              HHL09010
     WRITE(6,980) IVEC,BATA,BETA(IVEC),TEMP                    HHL09020
  980 FORMAT(/2X,'IVEC',16X,'BATA',10X,'BETA(IVEC)',14X,'RELDIF'/I6, HHL09030
    13E20.12/' IN LANCZOS VECTOR REGENERATION THE ENTRIES OF THE TRIDIAHHL09040
    1GONAL MATRICES BEING'/' GENERATED ARE NOT THE SAME AS THOSE IN THEHHL09050
    1 MATRIX SUPPLIED ON FILE 2.'/' THEREFORE SOMETHING IS BEING INITIAHHL09060
    1LIZED OR COMPUTED DIFFERENTLY FROM THE WAY'/' IT WAS COMPUTED IN THHL09070
    1HE EIGENVALUE COMPUTATIONS'/' THE PROGRAM TERMINATES FOR THE USER HHL09080
    1TO DETERMINE WHAT THE PROBLEM IS'/)                       HHL09090
     GO TO 1630                                                HHL09100
C                                                              HHL09110
C                                                              HHL09120
  990 CONTINUE                                                 HHL09130
     DO 1000 J = 1,N                                           HHL09140
     TEMPC = SUM*V1(J)                                         HHL09150
     V1(J) = V2(J)                                             HHL09160
 1000 V2(J) = TEMPC                                            HHL09170
C                                                              HHL09180
 1010 CONTINUE                                                 HHL09190
C                                                              HHL09200
     LFIN = 0                                                  HHL09210
     DO 1030 J = 1,NGOODC                                      HHL09220
     LL = LFIN                                                 HHL09230
```

```
      LFIN = LFIN + N                                           HHL09240
C                                                               HHL09250
      IF(IABS(MA(J)).LT.IVEC.OR.MP(J).EQ.MPMIN) GO TO 1030      HHL09260
      II = IVEC + MINT(J) - 1                                   HHL09270
      TEMP = TVEC(II)                                           HHL09280
C     II IS THE (IVEC)TH COMPONENT OF THE T-EIGENVECTOR CONTAINED HHL09290
C     IN TVEC(MINT(J)).                                         HHL09300
C                                                               HHL09310
      DO 1020 K = 1,N                                           HHL09320
      LL = LL + 1                                               HHL09330
 1020 RITVEC(LL) = TEMP*V2(K) + RITVEC(LL)                      HHL09340
C                                                               HHL09350
 1030 CONTINUE                                                  HHL09360
C                                                               HHL09370
      IVEC = IVEC + 1                                           HHL09380
      IF (IVEC.LE.KMAXU) GO TO 960                              HHL09390
C                                                               HHL09400
C                                                               HHL09410
C     RITZVECTOR GENERATION IS COMPLETE. NORMALIZE EACH RITZVECTOR. HHL09420
C     NOTE THAT IF CERTAIN RITZ VECTORS WERE NOT COMPUTED THEN THAT  HHL09430
C     PORTION OF THE RITVEC ARRAY WAS NOT UTILIZED.             HHL09440
C                                                               HHL09450
      LFIN = 0                                                  HHL09460
      DO 1130 J = 1,NGOODC                                      HHL09470
C                                                               HHL09480
      KK = LFIN                                                 HHL09490
      LFIN = LFIN + N                                           HHL09500
      IF(MP(J).EQ.MPMIN) GO TO 1130                             HHL09510
C                                                               HHL09520
      DO 1040 K = 1,N                                           HHL09530
      KK = KK + 1                                               HHL09540
 1040 V2(K) = RITVEC(KK)                                        HHL09550
C                                                               HHL09560
C---------------------------------------------------------------HHL09570
      CALL CINPRD(V2,V2,SUM,N)                                  HHL09580
C---------------------------------------------------------------HHL09590
C                                                               HHL09600
      SUM = DSQRT(SUM)                                          HHL09610
      RNORM(J) = SUM                                            HHL09620
      TEMP = DABS(ONE-SUM)                                      HHL09630
      SUM = ONE/SUM                                             HHL09640
C                                                               HHL09650
      KK = LFIN - N                                             HHL09660
      DO 1050 K = 1,N                                           HHL09670
      KK = KK + 1                                               HHL09680
      V2(K) = SUM*V2(K)                                         HHL09690
 1050 RITVEC(KK) = V2(K)                                        HHL09700
C                                                               HHL09710
C     ONLY ENTER NEXT PORTION IF GIVEN MATRIX IS REAL.          HHL09720
      IF(ISREAL.NE.1) GO TO 1100                                HHL09730
C                                                               HHL09740
C     AT THIS POINT RITZ VECTOR IS IN V2.                       HHL09750
C     THIS PROGRAM CAN BE USED ON REAL MATRICES TO DETERMINE    HHL09760
C     WHICH IF ANY EIGENVALUES ARE A-MULTIPLE AND IF SO TO COMPUTE HHL09770
C     TWO EIGENVECTORS FOR THOSE EIGENVALUES THAT ARE MULTIPLE AND ONE HHL09780
```

```
C     FOR THOSE THAT ARE NOT MULTIPLE. HERE ONLY IDENTIFIES WHETHER    HHL09790
C     EIGENVALUE IS AT LEAST DOUBLE.  THIS IS DONE BY CHECKING THE      HHL09800
C     RATIOS OF SUCCEEDING REAL AND IMAGINARY PARTS OF THE COMPUTED     HHL09810
C     RITZ VECTORS.                                                     HHL09820
C                                                                       HHL09830
      SUM = DIMAG(V2(1))/DREAL(V2(1))                                   HHL09840
      DO 1060 K=2,N                                                     HHL09850
      TEMP = DREAL(V2(K))                                               HHL09860
      IF(DABS(TEMP).LT.1.D-9) GO TO 1060                                HHL09870
      TEMP = DIMAG(V2(K))/DREAL(V2(K))                                  HHL09880
      IF(DABS(TEMP - SUM).LE.1.D-6) GO TO 1060                          HHL09890
      MULEVA(J) = 2                                                     HHL09900
      GO TO 1070                                                        HHL09910
 1060 CONTINUE                                                          HHL09920
      MULEVA(J) = 1                                                     HHL09930
 1070 IF(MULEVA(J).EQ.2) WRITE(6,1090) J,GOODEV(J)                      HHL09940
      IF(MULEVA(J).EQ.1) WRITE(6,1080) J,GOODEV(J)                      HHL09950
 1080 FORMAT(I6,'TH EIGENVALUE CONSIDERED =',E20.12,' IS SIMPLE')       HHL09960
 1090 FORMAT(I6,'TH EIGENVALUE CONSIDERED =',E20.12,' IS MULTIPLE')     HHL09970
C                                                                       HHL09980
 1100 CONTINUE                                                          HHL09990
C                                                                       HHL10000
      IF (IWRITE.NE.0) WRITE(6,1110) J,GOODEV(J)                        HHL10010
 1110 FORMAT(/I5,' TH EIGENVALUE CONSIDERED = ',E20.12/)                HHL10020
C                                                                       HHL10030
      IF (IWRITE.NE.0) WRITE(6,1120) TERR(J),TBETA(J),TEMP              HHL10040
 1120 FORMAT(' NORM OF ERROR IN T-EIGENVECTOR = ',E14.3/               HHL10050
     1 ' BETA(MA(J)+1)*U(MA(J))  = ',E14.3/                             HHL10060
     1 ' ABS(NORM(RITVEC) - 1.0)  = ',E14.3/)                           HHL10070
C                                                                       HHL10080
      LINT = LFIN - N + 1                                               HHL10090
      EVAL = EVNEW(J)                                                   HHL10100
C                                                                       HHL10110
C----------------------------------------------------------------------HHL10120
C                                                                       HHL10130
      CALL CMATV(RITVEC(LINT),V2,EVAL)                                  HHL10140
C                                                                       HHL10150
C----------------------------------------------------------------------HHL10160
C                                                                       HHL10170
C     COMPUTE ERROR IN RITZ VECTOR CONSIDERED AS A EIGENVECTOR OF A.    HHL10180
C     V2 = A*RITVEC - EVAL*RITVEC                                       HHL10190
C                                                                       HHL10200
C----------------------------------------------------------------------HHL10210
      CALL CINPRD(V2,V2,SUM,N)                                          HHL10220
C----------------------------------------------------------------------HHL10230
C                                                                       HHL10240
      SUM = DSQRT(SUM)                                                  HHL10250
      ERR(J) = SUM                                                      HHL10260
      GAP = ABS(AMINGP(J))                                              HHL10270
      ERRDGP(J) = SUM/GAP                                               HHL10280
C                                                                       HHL10290
 1130 CONTINUE                                                          HHL10300
C                                                                       HHL10310
C                                                                       HHL10320
C     RITZVECTORS ARE NORMALIZED AND ERROR ESTIMATES ARE IN ERR ARRAY   HHL10330
```

```
C     AND IN ERRDGP ARRAY. STORE EVERYTHING                     HHL10340
C                                                               HHL10350
C                                                               HHL10360
      WRITE(9,1140)                                             HHL10370
 1140 FORMAT(6X,'GOODEV(J)',1X,'MA(J)',4X,'A MINGAP',6X,'AERROR',2X,  HHL10380
     1 'AERROR/GAP',6X,'TERROR')                                HHL10390
C                                                               HHL10400
      WRITE(13,1150)                                            HHL10410
 1150 FORMAT(16X,'GOODEV(J)',5X,'RITZNORM',6X,'AMINGAP',5X,     HHL10420
     1 'TBETA(J)',5X,'TLAST(J)')                                HHL10430
C                                                               HHL10440
      DO 1180 J=1,NGOODC                                        HHL10450
C                                                               HHL10460
      IF(MP(J).EQ.MPMIN) GO TO 1180                             HHL10470
C                                                               HHL10480
      WRITE(9,1160)EVNEW(J),MA(J),AMINGP(J),ERR(J),ERRDGP(J),TERR(J)  HHL10490
 1160 FORMAT(E15.8,I6,4E12.4)                                   HHL10500
C                                                               HHL10510
      WRITE(13,1170) EVNEW(J),RNORM(J),AMINGP(J),TBETA(J),TLAST(J)  HHL10520
 1170 FORMAT(E25.14,4E13.5)                                     HHL10530
C                                                               HHL10540
 1180 CONTINUE                                                  HHL10550
C                                                               HHL10560
      IF(MREJEC.EQ.0) GO TO 1260                                HHL10570
      WRITE(9,1190)                                             HHL10580
 1190 FORMAT(/' RITZ VECTORS WERE NOT COMPUTED FOR THE FOLLOWING EIGENVAHHL10590
     1LUES'/' EITHER BECAUSE THEY HAD NOT CONVERGED OR BECAUSE THE ERRORHHL10600
     1 ESTIMATE'/'  WAS NOT AS SMALL AS DESIRED'/)              HHL10610
C                                                               HHL10620
      DO 1250 J = 1,NGOODC                                      HHL10630
      IF(MP(J).NE.MPMIN) GO TO 1250                             HHL10640
C     WRITE OUT MESSAGE FOR EACH EIGENVALUE FOR WHICH NO EIGENVECTOR  HHL10650
C     WAS COMPUTED.                                             HHL10660
C                                                               HHL10670
      WRITE(9,1200)                                             HHL10680
 1200 FORMAT(6X,'GOODEV(J)',3X,'MA(J)',5X,'AMINGP(J)',6X,'TLAST(J)',3X, HHL10690
     1'MP(J)')                                                  HHL10700
      WRITE(9,1210) GOODEV(J),MA(J),AMINGP(J),TBETA(J),MP(J)    HHL10710
 1210 FORMAT(E15.8,I8,2E14.4,I8)                                HHL10720
C                                                               HHL10730
      WRITE(13,1220)                                            HHL10740
 1220 FORMAT(/' RITZ VECTORS WERE NOT COMPUTED FOR THE FOLLOWING EIGENVAHHL10750
     1LUES'/' EITHER BECAUSE THEY HAD NOT CONVERGED OR BECAUSE'/' THE ERHHL10760
     1ROR ESTIMATE WAS NOT AS SMALL AS DESIRED'/)              HHL10770
C                                                               HHL10780
      WRITE(13,1230)                                            HHL10790
 1230 FORMAT(6X,'GOODEV(J)',3X,'MA(J)',3X,'M1(J)',3X,'M2(J)',3X,'MP(J)' HHL10800
     1/)                                                        HHL10810
      WRITE(13,1240) GOODEV(J),MA(J),M1(J),M2(J),MP(J)          HHL10820
 1240 FORMAT(E15.8,4I8)                                         HHL10830
C                                                               HHL10840
 1250 CONTINUE                                                  HHL10850
 1260 CONTINUE                                                  HHL10860
C                                                               HHL10870
      WRITE(9,1270)                                             HHL10880
```

```
 1270 FORMAT(/' ABOVE ARE ERROR ESTIMATES FOR THE A AND T EIGENVECTORS'/HHL10890
     1 ' ASSOCIATED WITH THE GOODEV LISTED IN COLUMN 1'/          HHL10900
     1 ' AERROR = NORM(A*X - EV*X)  TERROR = NORM(T*Y - EV*Y) '/   HHL10910
     1 ' WHERE T = T(1,MA(J))   X = RITZ VECTOR = V*Y  V = SUCCESSIVE'/HHL10920
     1 ' LANCZOS VECTORS. A MINGAP = GAP TO NEAREST A-EIGENVALUE'//) HHL10930
C                                                                 HHL10940
      WRITE(13,1280)                                              HHL10950
 1280 FORMAT(/' ABOVE ARE ERROR ESTIMATES ASSOCIATED WITH THE GOODEV'/ HHL10960
     1 ' RITZNORM = NORM(RITZ VECTOR)'/                           HHL10970
     1 ' TBETA(J) = CDABS(BETA(MA(J)+1)*Y(MA(J))),  T*Y = GOODEV*Y'/ HHL10980
     1 ' TLAST(J) = CDABS(Y(MA(J)))'/                             HHL10990
     1 ' AMINGAP = DISTANCE TO CLOSEST COMPUTED GOOD T-EIGENVALUE'/) HHL11000
C                                                                 HHL11010
C     NUMBER OF RITZ VECTORS COMPUTED                             HHL11020
      NCOMPU = NGOODC - MREJEC                                    HHL11030
      WRITE(12,1290) N,NCOMPU,NGOODC,MATNO                        HHL11040
 1290 FORMAT(3I6,I12,' SIZE A, NO.RITZVECS, NO.EVALUES,MATNO')    HHL11050
C                                                                 HHL11060
      LFIN = 0                                                    HHL11070
      DO 1350 J = 1,NGOODC                                        HHL11080
      LINT = LFIN + 1                                             HHL11090
      LFIN = LFIN + N                                             HHL11100
C                                                                 HHL11110
      IF(MP(J).EQ.MPMIN) GO TO 1330                               HHL11120
C     RITZ VECTOR WAS COMPUTED                                    HHL11130
      WRITE(12,1300) J, GOODEV(J), MP(J)                          HHL11140
 1300 FORMAT(I6,4X,E20.12,I6,' J, EIGENVAL, MP(J)')               HHL11150
C                                                                 HHL11160
      WRITE(12,1310) ERR(J),ERRDGP(J)                             HHL11170
 1310 FORMAT(2E15.5,' = NORM(A*Z-EVAL*Z) AND  NORM(A*Z-EVAL*Z)/MINGAP') HHL11180
C                                                                 HHL11190
      WRITE(12,1320) (RITVEC(LL), LL=LINT,LFIN)                   HHL11200
 1320 FORMAT(4E20.12)                                             HHL11210
      GO TO 1350                                                  HHL11220
C     NO RITZ VECTOR WAS COMPUTED FOR THIS EIGENVALUE             HHL11230
 1330 WRITE(12,1340) J,GOODEV(J),MP(J)                            HHL11240
 1340 FORMAT(I6,4X,E20.12,I6,' J,EIGVALUE,NO RITZ VECTOR COMPUTED') HHL11250
C                                                                 HHL11260
 1350 CONTINUE                                                    HHL11270
C                                                                 HHL11280
C     DID ANY T-MATRICES INCLUDE OFF-DIAGONAL ENTRIES SMALLER THAN HHL11290
C     DESIRED, AS SPECIFIED BY BTOL?                              HHL11300
C                                                                 HHL11310
      IF(IB.GT.0) GO TO 1380                                      HHL11320
      WRITE(6,1360) KMAXU                                         HHL11330
 1360 FORMAT(/' FOR LARGEST T-MATRIX CONSIDERED',I7,' CHECK THE SIZE OF HHL11340
     1BETAS')                                                     HHL11350
C                                                                 HHL11360
C-----------------------------------------------------------------HHL11370
C                                                                 HHL11380
      CALL TNORM(ALPHA,BETA,BKMIN,TEMP,KMAXU,IBMT)                HHL11390
C                                                                 HHL11400
C-----------------------------------------------------------------HHL11410
C                                                                 HHL11420
      IF(IBMT.LT.0) WRITE (6,1370)                                HHL11430
```

```
 1370 FORMAT(/' WARNING THE T-MATRICES FOR ONE OR MORE OF THE EIGENVALUEHHL11440
     1S CONSIDERED'/' HAD AN OFF-DIAGONAL ENTRY THAT WAS SMALLER THAN THHHL11450
     1E BETA TOLERANCE THAT WAS SPECIFIED'/)                        HHL11460
 1380 CONTINUE                                                      HHL11470
C                                                                   HHL11480
      GO TO 1630                                                    HHL11490
C                                                                   HHL11500
 1390 WRITE(6,1400) NGOOD,NMAX,MDIMRV                               HHL11510
 1400 FORMAT(/I4,' RITZ VECTORS WERE REQUESTED BUT THE REQUIRED DIMENSIOHHL11520
     1N',I6/' IS LARGER THAN THE USER-SPECIFIED DIMENSION OF RITVEC',I6 HHL11530
     1/' THEREFORE, THE EIGENVECTOR PROCEDURE TERMINATES FOR THE USER TOHHL11540
     1 INTERVENE')                                                  HHL11550
C                                                                   HHL11560
      GO TO 1630                                                    HHL11570
C                                                                   HHL11580
 1410 WRITE(6,1420) NOLD,N,MATOLD,MATNO                             HHL11590
 1420 FORMAT(/' PARAMETERS READ FROM FILE 3 DO NOT AGREE WITH THOSE SPECHHL11600
     1 IFIED'/' BY THE USER.   NOLD,N,MATOLD,MATNO = '/2I6,2I12/     HHL11610
     1' THEREFORE, PROGRAM TERMINATES FOR USER TO RESOLVE THE DIFFERENCEHHL11620
     1S'/)                                                          HHL11630
C                                                                   HHL11640
      GO TO 1630                                                    HHL11650
C                                                                   HHL11660
 1430 WRITE(6,1440)                                                 HHL11670
 1440 FORMAT(/' PARAMETERS IN ALPHA,BETA FILE READ IN DO NOT AGREE WITH HHL11680
     1 THOSE'/' SPECIFIED BY THE USER.  THEREFORE, THE PROCEDURE TERMINAHHL11690
     1TES'/' FOR THE USER TO RESOLVE THE DIFFERENCES.'/)            HHL11700
C                                                                   HHL11710
      GO TO 1630                                                    HHL11720
C                                                                   HHL11730
 1450 WRITE(6,1460) KMAX,MEV                                        HHL11740
 1460 FORMAT(/' ON ALPHA,BETA HEADER KMAX = ',I6/                   HHL11750
     1' BUT EIGENVALUES WERE COMPUTED AT MEV = ',I6,' PROGRAM STOPS'/) HHL11760
C                                                                   HHL11770
      GO TO 1630                                                    HHL11780
C                                                                   HHL11790
 1470 WRITE(6,1480)                                                 HHL11800
 1480 FORMAT(/' PROGRAM COMPUTED 1ST GUESSES ON T-MATRIX SIZES, READ THEHHL11810
     1M TO FILE 10'/' THEN TERMINATED AS REQUESTED.')               HHL11820
      GO TO 1630                                                    HHL11830
C                                                                   HHL11840
 1490 WRITE(6,1500) MTOL, MDIMTV                                    HHL11850
 1500 FORMAT(/' PROGRAM TERMINATES BECAUSE THE TVEC DIMENSION ANTICIPATEHHL11860
     1D',I7/'  IS LARGER THAN THE TVEC DIMENSION',I7,' SPECIFIED BY THE HHL11870
     1USER.'/'  USER MAY RESET THE TVEC DIMENSION AND RESTART THE PROGRAHHL11880
     1M')                                                           HHL11890
      GO TO 1630                                                    HHL11900
C                                                                   HHL11910
 1510 WRITE(6,1520)                                                 HHL11920
 1520 FORMAT(/' PROGRAM TERMINATES BECAUSE NO SUITABLE T-EIGENVECTORS WEHHL11930
     1RE IDENTIFIED'/'  FOR ANY OF THE EIGENVALUES SUPPLIED.  PROBLEM COHHL11940
     1ULD BE CAUSED'/'  BY TOO SMALL A TVEC DIMENSION OR SIMPLY BE THAT HHL11950
     1IT WAS NOT POSSIBLE'/'  TO IDENTIFY T-VECTORS.  USER SHOULD CHECK HHL11960
     1OUTPUT'/)                                                     HHL11970
      GO TO 1630                                                    HHL11980
```

```
C                                                               HHL11990
 1530 WRITE(6,1540) LVCONT,NTVEC,NGOOD                          HHL12000
 1540 FORMAT(/' LVCONT FLAG =',I2,' AND NUMBER ',I5,' OF T-EIGENVECTORS HHL12010
     1 COMPUTED N.E.'/' NUMBER',I5,' REQUESTED SO PROGRAM TERMINATES'/) HHL12020
      GO TO 1630                                               HHL12030
 1550 WRITE(6,1560)                                            HHL12040
 1560 FORMAT(/' PROGRAM TERMINATES WITHOUT COMPUTING ANY RITZ VECTORS'/ HHL12050
     1 ' BECAUSE ALL T-EIGENVECTORS WERE REJECTED AS NOT SUITABLE'/    HHL12060
     1 ' PROBABLE CAUSE IS LACK OF CONVERGENCE OF THE EIGENVALUES'/)   HHL12070
      GO TO 1630                                               HHL12080
C                                                               HHL12090
 1570 WRITE(6,1580)                                            HHL12100
 1580 FORMAT(/' PROGRAM INDICATES THAT IT IS NOT POSSIBLE TO COMPUTE ANYHHL12110
     1 OF THE'/' REQUESTED EIGENVECTORS. THEREFORE PROGRAM TERMINATES') HHL12120
      DO 1590 J=1,NGOODC                                       HHL12130
 1590 WRITE(6,1600)  J,GOODEV(J),MP(J)                         HHL12140
 1600 FORMAT(/4X,' J',11X,'GOODEV(J)',4X,'MP(J)'/I6,E20.12,I9) HHL12150
      GO TO 1630                                               HHL12160
C                                                               HHL12170
 1610 WRITE(6,1620) MBETA,KMAXN                                HHL12180
 1620 FORMAT(/' PROGRAM TERMINATES BECAUSE THE STORAGE ALLOTTED FOR THE HHL12190
     1BETA ARRAY',I8/' IS NOT SUFFICIENT FOR THE ENLARGED KMAX =',I8,' THHL12200
     1HAT THE PROGRAM WANTS'/' USER CAN ENLARGE THE ALPHA AND BETA ARRAYHHL12210
     1S AND RERUN THE PROGRAM.'/)                              HHL12220
C                                                               HHL12230
 1630 CONTINUE                                                 HHL12240
C                                                               HHL12250
      STOP                                                     HHL12260
C-----END OF MAIN PROGRAM FOR LANCZOS HERMITIAN EIGENVECTOR COMPUTATIONSHHL12270
      END                                                      HHL12280
```

3.4   HLEMULT: LANCZS and Sample Matrix-Vector Multiply Subroutines

```
C-----HLEMULT----HERMITIAN MATRICES-------------------------------------HHL00005
C  Authors:   Jane Cullum and Ralph A. Willoughby (deceased)      HHL00006
C             Los Alamos National Laboratory                      HHL00007
C             Los Alamos, New Mexico 87544                         HHL00008
C             cullumj@lanl.gov                                     HHL00009
c                                                                  HHL00010
c  These codes are copyrighted by the authors.  These codes        HHL00011
c  and modifications of them or portions of them are NOT to be     HHL00012
c  incorporated into any commercial codes without legal agreements HHL00013
c  with the authors.  If these codes or portions of them           HHL00014
c  are used in other scientific or engineering research works      HHL00015
c  the names of the authors of these codes and appropriate         HHL00016
c  references to their written work are to be incorporated in the  HHL00017
c  derivative works.                                               HHL00018
c                                                                  HHL00019
c  This header is not to be removed from these codes.              HHL00020
C                                                                  HHL00021
C         REFERENCE: Cullum and Willoughby, Chapters 1,2,3,4       HHL00022
C         Lanczos Algorithms for Large Symmetric Eigenvalue ComputationsHHL00023
C         VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in  HHL00024
C         Applied Mathematics, 2002. SIAM Publications,            HHL00025
C         Philadelphia, PA. USA                                    HHL00026
C                                                                  HHL00027
C     CONTAINS SUBROUTINE LANCZS AND SAMPLE USPEC, CMATV           HHL00030
C     USED BY THE HERMITIAN VERSION OF THE LANCZOS ALGORITHMS      HHL00040
C                                                                  HHL00050
C     PORTABILITY:                                                 HHL00060
C     THESE PROGRAMS ARE NOT PORTABLE DUE TO THE USE OF COMPLEX*16 HHL00070
C     VARIABLES.  MOREOVER, THE PFORT VERIFIER IDENTIFIED THE      HHL00080
C     FOLLOWING ADDITIONAL NONPORTABLE CONSTRUCTIONS:              HHL00090
C     1.   THE ENTRY MECHANISM USED TO PASS THE STORAGE            HHL00100
C          LOCATIONS OF THE USER-SPECIFIED MATRIX FROM THE         HHL00110
C          SUBROUTINE USPEC TO THE MATRIX-VECTOR SUBROUTINE CMATV. HHL00120
C     2.   IN THE PROGRAMS PROVIDED FOR 'HERMITIAN POISSON' TEST MATRICESHHL00130
C          USPEC CONTAINS FREE FORMAT (8,*), AND FORMAT (20A4); AND HHL00140
C          EXACT ERROR SUBROUTINE CONTAINS DATA/MACHEP DEFINITION. HHL00150
C                                                                  HHL00160
C                                                                  HHL00170
C-----LANCZS-COMPUTE THE LANCZOS TRIDIAGONAL MATRICES------------------HHL00180
C                                                                  HHL00190
C     GRAM-SCHMIDT ORTHOGONALIZATION WITHOUT MODIFICATION          HHL00200
C     REQUIRES EXTRA VECTOR VS IN LANCZS.  MODIFICATION IS NOT     HHL00210
C     PERMISSIBLE IN THE HERMITIAN CASE BECAUSE COMPLEX PORTION    HHL00220
C     OF THE MODIFICATION COULD NOT BE INCORPORATED.              HHL00230
C                                                                  HHL00240
      SUBROUTINE LANCZS(MATVEC,V1,V2,VS,ALPHA,BETA,GR,GC,G,KMAX,MOLD1,N,HHL00250
     1 IIX)                                                        HHL00260
C                                                                  HHL00270
C--------------------------------------------------------------------HHL00280
      COMPLEX*16 V1(1), V2(1), VS(1), ZEROC, TEMP                  HHL00290
```

```
      DOUBLE PRECISION  ALPHA(1), BETA(1), BATA, SUM, ONE, ZERO      HHL00300
      DOUBLE PRECISION GR(1),GC(1)                                   HHL00310
      REAL  G(1)                                                     HHL00320
      EXTERNAL MATVEC                                                HHL00330
      DOUBLE PRECISION  DSQRT                                        HHL00340
C------------------------------------------------------------------HHL00350
C                                                                    HHL00360
      ZERO = 0.D0                                                    HHL00370
      ONE = 1.D0                                                     HHL00380
      ZEROC = DCMPLX(ZERO,ZERO)                                      HHL00390
C                                                                    HHL00400
      IF(MOLD1.GT.1)GO TO 50                                         HHL00410
C                                                                    HHL00420
C     ALPHA/BETA GENERATION STARTS AT I = 1                          HHL00430
C     MOLD1 = 1 SET V1 = 0. AND V2 = RANDOM UNIT VECTOR              HHL00440
      IIL=IIX                                                        HHL00450
C                                                                    HHL00460
C------------------------------------------------------------------HHL00470
      CALL GENRAN(IIL,G,N)                                           HHL00480
C------------------------------------------------------------------HHL00490
C                                                                    HHL00500
      DO 10 I = 1,N                                                  HHL00510
   10 GR(I) = G(I)                                                   HHL00520
C                                                                    HHL00530
C------------------------------------------------------------------HHL00540
      CALL GENRAN(IIL,G,N)                                           HHL00550
C------------------------------------------------------------------HHL00560
C                                                                    HHL00570
      DO 20 I = 1,N                                                  HHL00580
   20 GC(I) = G(I)                                                   HHL00590
C                                                                    HHL00600
      DO 30 I = 1,N                                                  HHL00610
   30 V2(I) = DCMPLX(GR(I),GC(I))                                    HHL00620
C                                                                    HHL00630
C------------------------------------------------------------------HHL00640
      CALL CINPRD(V2,V2,SUM,N)                                       HHL00650
C------------------------------------------------------------------HHL00660
C                                                                    HHL00670
      SUM = ONE/DSQRT(SUM)                                           HHL00680
      DO 40 I = 1,N                                                  HHL00690
      V1(I) = ZEROC                                                  HHL00700
   40 V2(I) = V2(I)*SUM                                              HHL00710
      BETA(1) = ZERO                                                 HHL00720
C                                                                    HHL00730
C     ALPHA BETA GENERATION LOOP                                     HHL00740
   50 CONTINUE                                                       HHL00750
C                                                                    HHL00760
      DO 80 I=MOLD1,KMAX                                             HHL00770
      SUM = ZERO                                                     HHL00780
C                                                                    HHL00790
C------------------------------------------------------------------HHL00800
C     MATVEC(V2,VS,SUM) CALCULATES  VS = A*V2 - SUM*VS               HHL00810
      CALL MATVEC(V2,VS,SUM)                                         HHL00820
      CALL CINPRD(V2,VS,SUM,N)                                       HHL00830
C------------------------------------------------------------------HHL00840
```

```
C                                                                       HHL00850
      ALPHA(I) = SUM                                                    HHL00860
      BATA = BETA(I)                                                    HHL00870
      DO 60 J=1,N                                                       HHL00880
   60 V1(J) = (VS(J)-BATA*V1(J)) - SUM*V2(J)                            HHL00890
C                                                                       HHL00900
C-----------------------------------------------------------------HHL00910
      CALL CINPRD(V1,V1,SUM,N)                                          HHL00920
C-----------------------------------------------------------------HHL00930
C                                                                       HHL00940
      IN = I+1                                                          HHL00950
      BETA(IN) = DSQRT(SUM)                                             HHL00960
      SUM = ONE/BETA(IN)                                                HHL00970
      DO 70 J=1,N                                                       HHL00980
      TEMP = SUM*V1(J)                                                  HHL00990
      V1(J) = V2(J)                                                     HHL01000
   70 V2(J) = TEMP                                                      HHL01010
   80 CONTINUE                                                          HHL01020
C     END ALPHA, BETA GENERATION LOOP                                   HHL01030
C                                                                       HHL01040
C-----END OF LANCZS----------------------------------------------HHL01050
C                                                                       HHL01060
      RETURN                                                            HHL01070
      END                                                               HHL01080
C                                                                       HHL01090
C-----USPEC-GENERAL SPARSE, HERMITIAN MATRIX---------------------HHL01100
C                                                                       HHL01110
C     SUBROUTINE USPEC(N,MATNO)                                         HHL01120
      SUBROUTINE GUSPEC(N,MATNO)                                        HHL01130
C                                                                       HHL01140
C-----------------------------------------------------------------HHL01150
      COMPLEX*16 A(3000)                                                HHL01160
      DOUBLE PRECISION  AD(1000)                                        HHL01170
      INTEGER  IROW(3000),ICOL(1000)                                    HHL01180
C-----------------------------------------------------------------HHL01190
C     DIMENSION ARRAYS NEEDED TO DEFINE MATRIX, READ IN VALUES FOR      HHL01200
C     ARRAYS AND THEN PASS THE STORAGE LOCATIONS OF THESE ARRAYS TO     HHL01210
C     THE MATRIX-VECTOR MULTIPLY SUBROUTINE CMATV.                      HHL01220
C                                                                       HHL01230
C     USER-SUPPLIED MATRIX IS STORED IN FOLLOWING SPARSE FORMAT:        HHL01240
C     N = ORDER OF A-MATRIX                                             HHL01250
C     NZS = NUMBER OF NONZERO SUBDIAGONAL ENTRIES IN A                  HHL01260
C     NZL = INDEX OF LAST COLUMN CONTAINING NONZERO SUBDIAGONAL ENTRIES HHL01270
C     ICOL(J), J=1,NZL IS THE NUMBER OF NONZERO SUBDIAGONAL ELEMENTS    HHL01280
C             IN COLUMN J.                                              HHL01290
C     IROW(K), K = 1,NZS, IS THE ROW INDEX FOR CORRESPONDING A(K).      HHL01300
C     AD(I), I=1,N ARE DIAGONAL ENTRIES (INCLUDING ANY 0 DIAGONAL       HHL01310
C          ENTRIES)                                                     HHL01320
C     A(K), K=1,NZS ARE NONZERO SUBDIAGONAL ENTRIES, LISTED BY COLUMN.  HHL01330
C     FOR J > NZL THERE ARE NO NONZERO SUBDIAGONAL ELEMENTS IN COLUMN J. HHL01340
C     ICOL(J) = 0 IS ALLOWED                                            HHL01350
C                                                                       HHL01360
C-----------------------------------------------------------------HHL01370
C     IN THIS SAMPLE SUBROUTINE THE ARRAYS ARE READ IN FROM FILE 8      HHL01380
C                                                                       HHL01390
```

```
      READ(8,10) NZS,NOLD,NZL,MATOLD                            HHL01400
   10 FORMAT(I10,2I6,I8)                                        HHL01410
C                                                               HHL01420
      WRITE(6,20) NZS,NOLD,NZL,MATOLD                           HHL01430
   20 FORMAT(I10,2I6,I8,' = NZS,NOLD,NZL,MATOLD'/)              HHL01440
C                                                               HHL01450
C     TEST OF PARAMETER CORRECTNESS                             HHL01460
      ITEMP = (NOLD-N)**2 + (MATNO-MATOLD)**2                   HHL01470
C                                                               HHL01480
      IF(ITEMP.EQ.0) GO TO 40                                   HHL01490
C                                                               HHL01500
      WRITE(6,30)                                               HHL01510
   30 FORMAT(/' PROGRAM TERMINATES BECAUSE EITHER ORDERS OF OR LABELS FOHHL01520
     1R MATRIX DISAGREE'/)                                      HHL01530
      GO TO 80                                                  HHL01540
C                                                               HHL01550
   40 CONTINUE                                                  HHL01560
C                                                               HHL01570
C     NUMBER OF NONZERO SUBDIAGONAL ENTRIES IN EACH COLUMN IS READ  HHL01580
C     THEN THE CORRESPONDING ROW INDEX FOR EACH SUCH ENTRY IS READ  HHL01590
      READ(8,50) (ICOL(K), K=1,NZL)                             HHL01600
      READ(8,50) (IROW(K), K=1,NZS)                             HHL01610
   50 FORMAT(13I6)                                              HHL01620
C     DIAGONAL IS READ FIRST, THEN NONZERO BELOW DIAGONAL ENTRIES   HHL01630
      READ(8,60) (AD(K), K=1,N)                                 HHL01640
   60 FORMAT(4E20.12)                                           HHL01650
      READ(8,70) (A(K), K=1,NZS)                                HHL01660
C  50 FORMAT(4Z20)                                              HHL01670
   70 FORMAT(4E20.12)                                           HHL01680
C                                                               HHL01690
C------------------------------------------------------------------HHL01700
C     PASS STORAGE LOCATIONS OF ARRAYS THAT DEFINE THE MATRIX TO HHL01710
C     THE MATRIX-VECTOR MULTIPLY SUBROUTINE CMATV               HHL01720
      CALL CMATVE(A,AD,ICOL,IROW,N,NZL)                         HHL01730
C------------------------------------------------------------------HHL01740
C                                                               HHL01750
      RETURN                                                    HHL01760
   80 STOP                                                      HHL01770
C                                                               HHL01780
C-----END OF USPEC FOR GENERAL, SPARSE HERMITIAN MATRICES-----------HHL01790
      END                                                       HHL01800
C                                                               HHL01810
C-----START OF MATRIX-VECTOR MULTIPLY-GENERAL SPARSE HERMITIAN-------- HHL01820
C                                                               HHL01830
C     SUBROUTINE CMATV(W,U,SUM)                                 HHL01840
      SUBROUTINE GCMATV(W,U,SUM)                                HHL01850
C                                                               HHL01860
C------------------------------------------------------------------HHL01870
      COMPLEX*16  U(1),W(1),A(1)                                HHL01880
      DOUBLE PRECISION  AD(1),SUM                               HHL01890
      INTEGER  IROW(1),ICOL(1)                                  HHL01900
C------------------------------------------------------------------HHL01910
C     SPARSE MATRIX-VECTOR MULTIPLY FOR LANCZS  U = A*W - SUM*U  HHL01920
C     SEE USPEC SUBROUTINE FOR DESCRIPTION OF THE ARRAYS THAT DEFINE HHL01930
C     THE MATRIX                                                HHL01940
```

```
C                                                         HHL01950
      GO TO 3                                             HHL01960
C                                                         HHL01970
C-------------------------------------------------------HHL01980
C     STORAGE LOCATIONS OF ARRAYS ARE PASSED TO CMATV FROM USPEC  HHL01990
      ENTRY CMATVE(A,AD,ICOL,IROW,N,NZL)                  HHL02000
C-------------------------------------------------------HHL02010
C                                                         HHL02020
      GO TO 4                                             HHL02030
    3 CONTINUE                                            HHL02040
C                                                         HHL02050
C     COMPUTE THE DIAGONAL TERMS                          HHL02060
      DO 10 I = 1,N                                       HHL02070
   10 U(I) = AD(I)*W(I)-SUM*U(I)                          HHL02080
C                                                         HHL02090
C     COMPUTE BY COLUMN                                   HHL02100
      LLAST = 0                                           HHL02110
      DO 30 J = 1,NZL                                     HHL02120
C                                                         HHL02130
      IF (ICOL(J).EQ.0) GO TO 30                          HHL02140
      LFIRST = LLAST + 1                                  HHL02150
      LLAST = LLAST + ICOL(J)                             HHL02160
C                                                         HHL02170
      DO 20 L = LFIRST,LLAST                              HHL02180
      I = IROW(L)                                         HHL02190
C                                                         HHL02200
      U(I) = U(I) + A(L)*W(J)                             HHL02210
      U(J) = U(J) + DCONJG(A(L))*W(I)                     HHL02220
C                                                         HHL02230
   20 CONTINUE                                            HHL02240
C                                                         HHL02250
   30 CONTINUE                                            HHL02260
C                                                         HHL02270
    4 RETURN                                              HHL02280
C                                                         HHL02290
C-----END OF CMATV-GENERAL, SPARSE, HERMITIAN MATRICES ----------------HHL02300
      END                                                 HHL02310
C                                                         HHL02320
C-----USPEC, CMATV, EXEVG, AND HEXVEC FOR HERMITIAN 'POISSON' MATRICES--HHL02330
C                                                         HHL02340
C-----USPEC (HERMITIAN POISSON MATRICES)-----------------------------HHL02350
C                                                         HHL02360
      SUBROUTINE HUSPEC(N,MATNO)                          HHL02370
C     SUBROUTINE USPEC(N,MATNO)                           HHL02380
C                                                         HHL02390
C-------------------------------------------------------HHL02400
      DOUBLE PRECISION  C0,C1,C2,HALF,ONE,SCR,SCI,ANGLE,TEMP  HHL02410
      COMPLEX*16 SC,TC,CL0,CL1,CL3,CL4                    HHL02420
      REAL EXPLAN(20)                                     HHL02430
      DOUBLE PRECISION EIGVAL(1000)                       HHL02440
      REAL GAPS(1000)                                     HHL02450
      INTEGER MULTS(1000)                                 HHL02460
C-------------------------------------------------------HHL02470
      HALF = 0.5D0                                        HHL02480
      ONE  = 1.0D0                                        HHL02490
```

```
C                                                               HHL02500
C     READ IN PARAMETERS TO DEFINE MATRIX                       HHL02510
C     MATRIX IS COMPLEX DIAGONAL SIMILITARY TRANSFORM OF REAL SYMMETRIC HHL02520
C     POISSON MATRIX WHICH HAS SYMMETRIC TOEPLITZ BLOCKS ALONG  HHL02530
C     THE DIAGONAL, EACH ONE OF WHICH HAS THE PARAMETER C2 ALONG THE HHL02540
C     DIAGONAL AND -C0 ABOVE AND BELOW THE DIAGONAL, AND OFF-DIAGONAL HHL02550
C     BLOCKS THAT ARE DIAGONAL WITH DIAGONAL ENTRIES -C1.  EACH BLOCK HHL02560
C     IS KX*KX AND THERE ARE KY BLOCKS.  THE HERMITIAN VERSION IS HHL02570
C     OBTAINED BY APPLYING A DIAGONAL SIMILARITY TRANSFORM TO THE HHL02580
C     REAL MATRIX WHERE THIS TRANSFORMATION IS SUCH THAT ITS     HHL02590
C     DIAGONAL ENTRIES ARE (SC)**(K-1), K = 1,...,N, WHERE SC   HHL02600
C     HAS MODULUS 1.                                            HHL02610
C                                                               HHL02620
      READ(8,10) EXPLAN                                         HHL02630
      WRITE(6,10) EXPLAN                                        HHL02640
      READ(8,10) EXPLAN                                         HHL02650
   10 FORMAT(20A4)                                              HHL02660
C     IF MTYPE = 0 WE HAVE ZERO BOUNDARY CONDITIONS             HHL02670
C     IF MTYPE = 1 WE HAVE NORMAL DERIVATIVE BOUNDARY CONDITIONS HHL02680
C     NOTE THAT SUBROUTINES EXEVG AND HEXVEC ARE VALID ONLY FOR HHL02690
C     MTYPE = 0.                                                HHL02700
      READ(8,*) NOLD,MATOLD,IVEC,MTYPE                          HHL02710
      WRITE(6,20) NOLD,MATOLD                                   HHL02720
   20 FORMAT(' ORDER OF MATRIX READ FROM FILE =',I6/' MATRIX NUMBER =', HHL02730
     1I8/)                                                      HHL02740
      IF(MTYPE.EQ.0) WRITE(6,30)                                HHL02750
   30 FORMAT(/' HERMITIAN POISSON CORRESPONDING TO ZERO BOUNDARY CONDITIHHL02760
     1ONS'/)                                                    HHL02770
      IF(MTYPE.EQ.1) WRITE(6,40)                                HHL02780
   40 FORMAT(/' HERMITIAN POISSON CORRESPONDING TO NORMAL DERIVATIVE BOUHHL02790
     1NDARY CONDITIONS'/)                                       HHL02800
      IF(IVEC.NE.0.AND.MTYPE.EQ.0) WRITE(6,50)                  HHL02810
   50 FORMAT(' COMPUTE THE TRUE EIGENVALUES AND PUT IN FN TRUEEVAL'/) HHL02820
C                                                               HHL02830
C     TEST OF PARAMETER CORRECTNESS                             HHL02840
      ITEMP = (NOLD-N)**2 + (MATNO-MATOLD)**2                   HHL02850
C                                                               HHL02860
      IF(ITEMP.EQ.0) GO TO 70                                   HHL02870
C                                                               HHL02880
      WRITE(6,60)                                               HHL02890
   60 FORMAT(' PROGRAM TERMINATES BECAUSE EITHER ORDERS OF OR LABELS FORHHL02900
     1 MATRIX DISAGREE')                                        HHL02910
      GO TO 150                                                 HHL02920
C                                                               HHL02930
   70 CONTINUE                                                  HHL02940
C                                                               HHL02950
      READ(8,10) EXPLAN                                         HHL02960
      READ(8,*)  C0,KX,KY                                       HHL02970
      IF (KX.GT.4.AND.KY.GT.4) GO TO 90                         HHL02980
      WRITE(6,80) KX,KY                                         HHL02990
   80 FORMAT(2I6,' = KX KY  ONE OR BOTH OF KX KY TOO SMALL SO STOP'/) HHL03000
      GO TO 150                                                 HHL03010
   90 CONTINUE                                                  HHL03020
      READ(8,10) EXPLAN                                         HHL03030
C     BELOW SC = COS(ANGLE) + I SIN(ANGLE)                      HHL03040
```

```
C       READ IN DESIRED COSINE, COMPUTE ANGLE, THEN SINE              HHL03050
        READ(8,*) SCR                                                 HHL03060
        ANGLE = DACOS(SCR)                                            HHL03070
        SCI = DSIN(ANGLE)                                             HHL03080
        SC = DCMPLX(SCR,SCI)                                          HHL03090
        WRITE(6,100) SC                                               HHL03100
C       IF (IVEC.NE.0.AND.MTYPE.EQ.0) WRITE(9,7) SC                   HHL03110
  100 FORMAT(' GENERATOR OF DIAGONAL TRANSFORMATION ='/2E20.12)       HHL03120
C                                                                     HHL03130
        TC = SC                                                       HHL03140
        DO 110 J=2,KX                                                 HHL03150
  110 TC = SC*TC                                                      HHL03160
        WRITE(6,120) TC                                               HHL03170
  120 FORMAT(' TC = ',2E20.12)                                        HHL03180
C                                                                     HHL03190
        N = KX*KY                                                     HHL03200
        C2 = ONE                                                      HHL03210
        C1 = HALF-C0                                                  HHL03220
        TEMP = DSQRT(2.0D0)                                           HHL03230
        IF (MTYPE.EQ.0) TEMP = ONE                                    HHL03240
        CL0 = -SC*C0                                                  HHL03250
        CL1 = -TC*C1                                                  HHL03260
        CL3 = -SC*C0*TEMP                                             HHL03270
        CL4 = -TC*C1*TEMP                                             HHL03280
C                                                                     HHL03290
        WRITE(6,130) N,MTYPE,KX,KY,C2,C0,C1                           HHL03300
  130 FORMAT(/5X,'N',1X,'MTYPE',4X,'KX',4X,'KY',7X,'DIAGONAL',        HHL03310
      1 3X,'X-CODIAGONAL',3X,'Y-CODIAGONAL'/4I6,3E15.8/)              HHL03320
C                                                                     HHL03330
C-------------------------------------------------------------------HHL03340
        CALL HMATVE(C2,CL0,CL1,CL3,CL4,KX,KY)                         HHL03350
C-------------------------------------------------------------------HHL03360
C                                                                     HHL03370
        IF(IVEC.EQ.0.OR.MTYPE.NE.0) GO TO 140                         HHL03380
C                                                                     HHL03390
C       COMPUTE THE EXACT EIGENVALUES                                 HHL03400
C                                                                     HHL03410
C-------------------------------------------------------------------HHL03420
        CALL EXEVG(EIGVAL,C0,C1,C2,GAPS,MULTS,KX,KY)                  HHL03430
C-------------------------------------------------------------------HHL03440
C                                                                     HHL03450
        IF(IVEC.LT.0) GO TO 150                                       HHL03460
C                                                                     HHL03470
  140 CONTINUE                                                        HHL03480
        RETURN                                                        HHL03490
C                                                                     HHL03500
C-----END OF USPEC--------------------------------------------------HHL03510
  150 STOP                                                            HHL03520
        END                                                           HHL03530
C                                                                     HHL03540
C-----START OF CMATV FOR HERMITIAN POISSON MATRICES----------------HHL03550
C                                                                     HHL03560
        SUBROUTINE HMATV(W,U,SUM)                                     HHL03570
C       SUBROUTINE CMATV(W,U,SUM)                                     HHL03580
C                                                                     HHL03590
```

```
C---------------------------------------------------------------HHL03600
      DOUBLE PRECISION  C2,SUM                                   HHL03610
      COMPLEX*16 U(1),W(1)                                       HHL03620
      COMPLEX*16 CL0,CL1,CL3,CL4,CR0,CR1,CR3,CR4                 HHL03630
C---------------------------------------------------------------HHL03640
C     CALCULATES U = A*W - SUM*U                                 HHL03650
C                                                                HHL03660
      GO TO 3                                                    HHL03670
C                                                                HHL03680
      ENTRY HMATVE(C2,CL0,CL1,CL3,CL4,KK,LL)                     HHL03690
C                                                                HHL03700
      GO TO 4                                                    HHL03710
C                                                                HHL03720
    3 CONTINUE                                                   HHL03730
C                                                                HHL03740
      N = KK*LL                                                  HHL03750
      CR0 = DCONJG(CL0)                                          HHL03760
      CR1 = DCONJG(CL1)                                          HHL03770
      CR3 = DCONJG(CL3)                                          HHL03780
      CR4 = DCONJG(CL4)                                          HHL03790
C                                                                HHL03800
C---------------------------------------------------------------HHL03810
C     FIRST AND LAST BLOCKS                                      HHL03820
      J = 1                                                      HHL03830
      U(J)=(C2*W(J)+CR3*W(J+1)+CR1*W(J+KK)) - SUM*U(J)           HHL03840
      J = 2                                                      HHL03850
      U(J)=(C2*W(J)+CL3*W(J-1)+CR0*W(J+1)+CR1*W(J+KK))-SUM*U(J)  HHL03860
      J = KK                                                     HHL03870
      U(J)=(C2*W(J)+CL3*W(J-1)+CR1*W(J+KK))-SUM*U(J)             HHL03880
      J = KK - 1                                                 HHL03890
      U(J)=(C2*W(J)+CR3*W(J+1)+CL0*W(J-1)+CR1*W(J+KK))-SUM*U(J)  HHL03900
      J = N - KK + 1                                             HHL03910
      U(J)=(C2*W(J)+CR3*W(J+1)+CL4*W(J-KK))-SUM*U(J)             HHL03920
      J = N - KK + 2                                             HHL03930
      U(J)=(C2*W(J)+CL3*W(J-1)+CR0*W(J+1)+CL4*W(J-KK))-SUM*U(J)  HHL03940
      J = N                                                      HHL03950
      U(J)=(C2*W(J)+CL3*W(J-1)+CL4*W(J-KK))-SUM*U(J)             HHL03960
      J = N - 1                                                  HHL03970
      U(J)=(C2*W(J)+CL0*W(J-1)+CR3*W(J+1)+CL4*W(J-KK))-SUM*U(J)  HHL03980
C                                                                HHL03990
      KK2 = KK - 2                                               HHL04000
      DO 10 JJ = 3,KK2                                           HHL04010
      J = JJ                                                     HHL04020
      U(J)=(C2*W(J)+CL0*W(J-1)+CR0*W(J+1)+CR1*W(J+KK))-SUM*U(J)  HHL04030
      J = N - KK + JJ                                            HHL04040
   10 U(J)=(C2*W(J)+CL0*W(J-1)+CR0*W(J+1)+CL4*W(J-KK))-SUM*U(J)  HHL04050
C                                                                HHL04060
C     START BLOCKS 2 AND LL-1                                    HHL04070
      J = KK + 1                                                 HHL04080
      U(J)=(C2*W(J)+CR3*W(J+1)+CL1*W(J-KK)+CR1*W(J+KK))-SUM*U(J) HHL04090
      J = KK + 2                                                 HHL04100
      U(J)=(C2*W(J)+CL3*W(J-1)+CR0*W(J+1)+CL1*W(J-KK)+CR1*W(J+KK))HHL04110
    1 -SUM*U(J)                                                 HHL04120
      J = KK + KK                                                HHL04130
      U(J)=(C2*W(J)+CL3*W(J-1)+CL1*W(J-KK)+CR1*W(J+KK))-SUM*U(J) HHL04140
```

```
       J = KK + KK - 1                                               HHL04150
       U(J)=(C2*W(J)+CR3*W(J+1)+CL0*W(J-1)+CL1*W(J-KK)+CR1*W(J+KK))   HHL04160
      1  -SUM*U(J)                                                    HHL04170
       J = N - 2*KK + 1                                               HHL04180
       U(J)=(C2*W(J)+CR3*W(J+1)+CR4*W(J+KK)+CL1*W(J-KK))              HHL04190
      1 -SUM*U(J)                                                     HHL04200
       J = N - 2*KK + 2                                               HHL04210
       U(J)=(C2*W(J)+CL3*W(J-1)+CR0*W(J+1)+CR4*W(J+KK)+CL1*W(J-KK))   HHL04220
      1 -SUM*U(J)                                                     HHL04230
       J = N - KK                                                     HHL04240
       U(J)=(C2*W(J)+CL3*W(J-1)+CR4*W(J+KK)+CL1*W(J-KK))-SUM*U(J)     HHL04250
       J = N - KK - 1                                                 HHL04260
       U(J)=(C2*W(J)+CR3*W(J+1)+CL0*W(J-1)+CR4*W(J+KK)+CL1*W(J-KK))   HHL04270
      1 -SUM*U(J)                                                     HHL04280
C                                                                     HHL04290
       DO 20 JJ = 3,KK2                                               HHL04300
       J = KK + JJ                                                    HHL04310
       U(J)=(C2*W(J)+CL0*W(J-1)+CR0*W(J+1)+CL1*W(J-KK)+CR1*W(J+KK))   HHL04320
      1 -SUM*U(J)                                                     HHL04330
       J = N - 2*KK + JJ                                              HHL04340
       U(J)=(C2*W(J)+CL0*W(J-1)+CR0*W(J+1)+CR4*W(J+KK)+CL1*W(J-KK))   HHL04350
      1 -SUM*U(J)                                                     HHL04360
   20 CONTINUE                                                        HHL04370
C                                                                     HHL04380
C    MIDDLE BLOCKS                                                    HHL04390
       LL2 = LL - 2                                                   HHL04400
       JP = KK                                                        HHL04410
       DO 40 JJ = 3,LL2                                               HHL04420
       JP = JP + KK                                                   HHL04430
C      JP = (JJ-1)*KK                                                 HHL04440
       J = JP + 1                                                     HHL04450
       U(J)=(C2*W(J)+CR3*W(J+1)+CL1*W(J-KK)+CR1*W(J+KK))-SUM*U(J)     HHL04460
       J = J + 1                                                      HHL04470
       U(J)=(C2*W(J)+CL3*W(J-1)+CR0*W(J+1)+CL1*W(J-KK)+              HHL04480
      1 CR1*W(J+KK))-SUM*U(J)                                         HHL04490
       J = J + KK - 2                                                 HHL04500
       U(J) = (C2*W(J)+CL3*W(J-1)+CL1*W(J-KK)+CR1*W(J+KK))-SUM*U(J)   HHL04510
       J = J - 1                                                      HHL04520
       U(J)=(C2*W(J)+CR3*W(J+1)+CL0*W(J-1)+CL1*W(J-KK)+              HHL04530
      1 CR1*W(J+KK))-SUM*U(J)                                         HHL04540
C                                                                     HHL04550
       DO 30 II = 3,KK2                                               HHL04560
       J = JP + II                                                    HHL04570
       U(J)=(C2*W(J)+CL0*W(J-1)+CR0*W(J+1)+CL1*W(J-KK)+CR1*W(J+KK))   HHL04580
      1 -SUM*U(J)                                                     HHL04590
   30 CONTINUE                                                        HHL04600
C                                                                     HHL04610
   40 CONTINUE                                                        HHL04620
C                                                                     HHL04630
    4 RETURN                                                          HHL04640
C                                                                     HHL04650
C-----END OF HMATV-------------------------------------------------HHL04660
       END                                                           HHL04670
C                                                                     HHL04680
C-----START OF EXEVG-----------------------------------------------HHL04690
```

```
C                                                                      HHL04700
C     FOR MTYPE = 0, ZERO BOUNDARY CONDITIONS:                         HHL04710
C     COMPUTES EXACT EIGENVALUES OF HERMITIAN POISSON MATRIX,          HHL04720
C     THEIR MULTIPLICITIES, AND THE GAPS BETWEEN THE EIGENVALUES AND   HHL04730
C     PUTS THEM RESPECTIVELY INTO VECTORS U, MP, AND G.  THESE         HHL04740
C     QUANTITIES ARE ALL WRITTEN TO FILE 9.                            HHL04750
C                                                                      HHL04760
      SUBROUTINE EXEVG(U,C0,C1,C2,G,MP,KX,KY)                          HHL04770
C                                                                      HHL04780
C----------------------------------------------------------------------HHL04790
      DOUBLE PRECISION  U(*),MACHEP                                    HHL04800
      DOUBLE PRECISION  EPSM,C0,C1,C2,T0,T1,PIK,PIL,ONE,TWO,ATOLN,EE   HHL04810
      REAL G(1)                                                        HHL04820
      INTEGER MP(1)                                                    HHL04830
C----------------------------------------------------------------------HHL04840
      DATA MACHEP/Z3410000000000000/                                  HHL04850
      EPSM = 2.0D0*MACHEP                                             HHL04860
C----------------------------------------------------------------------HHL04870
      N = KX*KY                                                        HHL04880
      ONE  = 1.0D0                                                     HHL04890
      TWO  = 2.0D0                                                     HHL04900
      T0 = DACOS(-ONE)                                                 HHL04910
      T1 = DFLOAT(KX+1)                                                HHL04920
      PIK = T0/T1                                                      HHL04930
      T1 = DFLOAT(KY+1)                                                HHL04940
      PIL = T0/T1                                                      HHL04950
C     GENERATE EXACT EIGENVALUES                                      HHL04960
      KP = 0                                                           HHL04970
      DO 20 J = 1,KY                                                   HHL04980
      T1 = PIL*DFLOAT(J)                                               HHL04990
      T0 = C2 - TWO*C1*DCOS(T1)                                        HHL05000
      DO 10 I = 1,KX                                                   HHL05010
      KP = KP+1                                                        HHL05020
      T1 = PIK*DFLOAT(I)                                               HHL05030
   10 U(KP) = T0 - TWO*C0*DCOS(T1)                                     HHL05040
   20 CONTINUE                                                         HHL05050
C                                                                      HHL05060
C     ORDER U VECTOR BY INCREASING ALGEBRAIC SIZE                      HHL05070
      DO 40 K = 2,N                                                    HHL05080
      KM1 = K-1                                                        HHL05090
      DO 30 L = 1,KM1                                                  HHL05100
      JJ = K-L                                                         HHL05110
      IF (U(JJ+1).GE.U(JJ)) GO TO 40                                   HHL05120
      T0 = U(JJ)                                                       HHL05130
      U(JJ) = U(JJ+1)                                                  HHL05140
   30 U(JJ+1) = T0                                                     HHL05150
   40 CONTINUE                                                         HHL05160
      ATOLN = DMAX1(DABS(U(1)),DABS(U(N)))*EPSM                        HHL05170
C                                                                      HHL05180
      WRITE(9,50)                                                      HHL05190
   50 FORMAT(' TRUE EIGENVALUES FOR HERMITIAN POISSON')                HHL05200
C                                                                      HHL05210
      WRITE(9,60)N,KX,KY,C2,C0,C1,ATOLN                                HHL05220
      WRITE(6,60) N,KX,KY,C2,C0,C1,ATOLN                               HHL05230
   60 FORMAT(1X,'A-SIZE',2X,'X-DIM',2X,'Y-DIM'/3I7/                    HHL05240
```

```
     1 5X,'A-DIAGONAL',3X,'X-CODIAGONAL',3X,'Y-CODIAGONAL',10X,'ATOLN'/   HHL05250
     2 4E15.8)                                                            HHL05260
C                                                                         HHL05270
C     DETERMINE TRUE MULTIPLICITIES FOR EXACT EIGENVALUES                 HHL05280
      I = 1                                                               HHL05290
      IDEX = 1                                                            HHL05300
      J = 1                                                               HHL05310
      NEXACT = 0                                                          HHL05320
   70 J = J+1                                                             HHL05330
      IF (J.GT.N) GO TO 80                                                HHL05340
      EE = DABS(U(J)-U(I))                                                HHL05350
      IF (EE.GT.ATOLN) GO TO 80                                           HHL05360
      IDEX = IDEX+1                                                       HHL05370
      GO TO 70                                                            HHL05380
   80 NEXACT = NEXACT+1                                                   HHL05390
      U(NEXACT) = U(I)                                                    HHL05400
      MP(NEXACT) = IDEX                                                   HHL05410
C     MP(K) = MULTIPLICITY OF KTH EIGENVALUE CLUSTER FOR A                HHL05420
      IDEX = 1                                                            HHL05430
      I = J                                                               HHL05440
      IF (I.GT.N) GO TO 90                                                HHL05450
      GO TO 70                                                            HHL05460
   90 CONTINUE                                                            HHL05470
C                                                                         HHL05480
C     MULTIPLICITIES HAVE BEEN DETERMINED                                 HHL05490
C     NEXACT = NUMBER OF DISTINCT A-EIGENVALUES                           HHL05500
C                                                                         HHL05510
      WRITE(9,100)NEXACT                                                  HHL05520
      WRITE(6,100)NEXACT                                                  HHL05530
  100 FORMAT(I6,' = NUMBER OF TRUE A-EIGENVALUES WHICH ARE DISTINCT'/)    HHL05540
C                                                                         HHL05550
C     MINGAP CALCULATION FOR DISTINCT A-EIGENVALUES                       HHL05560
      NM1 = NEXACT - 1                                                    HHL05570
      G(NEXACT) = U(NM1)-U(NEXACT)                                        HHL05580
      G(1) = U(2)-U(1)                                                    HHL05590
C                                                                         HHL05600
      DO 110 J = 2,NM1                                                    HHL05610
      T0 = U(J)-U(J-1)                                                    HHL05620
      T1 = U(J+1)-U(J)                                                    HHL05630
      G(J) = T1                                                           HHL05640
      IF (T0.LT.T1) G(J) = -T0                                            HHL05650
  110 CONTINUE                                                            HHL05660
C                                                                         HHL05670
C     NEXACT DISTINCT A-EIGENVALUES ARE IN U IN ASCENDING ORDER           HHL05680
C     MP = MULTIPLICITIES OF THE DISTINCT EIGENVALUES OF A                HHL05690
C     G = TRUE MINIMUM GAP IN A FOR EACH OF THESE EIGENVALUES             HHL05700
C     G < 0 INDICATES THE LEFT-HAND GAP WAS MINIMAL.                      HHL05710
C     OUTPUT MULTIPLICITIES, DISTINCT EVS, AND MINGAPS TO FILE 11         HHL05720
C                                                                         HHL05730
      WRITE(9,120)                                                        HHL05740
  120 FORMAT(5X,'I',1X,'AMULT',5X,'TRUE A-EIGENVALUE(I)',                 HHL05750
     1 3X,'A-MINGAP(I)')                                                  HHL05760
C                                                                         HHL05770
      WRITE(9,130)(J,MP(J),U(J),G(J), J=1,NEXACT)                         HHL05780
  130 FORMAT(2I6,E25.16,E14.3)                                            HHL05790
```

```
C                                                                  HHL05800
      WRITE(9,140)                                                  HHL05810
  140 FORMAT(' NEXACT DISTINCT A-EIGENVALUES ARE IN ASCENDING ORDER'/   HHL05820
     1 ' AMULT = MULTIPLICITIES OF THE DISTINCT EIGENVALUES OF A.'/     HHL05830
     2 ' A-MINGAP(I) = TRUE MINIMUM GAP IN A FOR EACH EIGENVALUE.'/     HHL05840
     3 ' A-MINGAP(I) LT 0 INDICATES THE LEFT-HAND GAP WAS MINIMAL.'//)  HHL05850
C                                                                  HHL05860
C     WE ORDER U VECTOR BY INCREASING SIZE OF THE GAPS              HHL05870
C                                                                  HHL05880
      DO 150 K = 1,N                                                HHL05890
  150 MP(K) = K                                                     HHL05900
C                                                                  HHL05910
      DO 170 K = 2,N                                                HHL05920
      KM1 = K-1                                                     HHL05930
C                                                                  HHL05940
      DO 160 L = 1,KM1                                              HHL05950
      JJ = K - L                                                    HHL05960
      IF (ABS(G(JJ+1)).GE.ABS(G(JJ))) GO TO 170                     HHL05970
      EE = U(JJ)                                                    HHL05980
      U(JJ) = U(JJ+1)                                               HHL05990
      U(JJ+1) = EE                                                  HHL06000
      GG = G(JJ)                                                    HHL06010
      G(JJ) = G(JJ+1)                                               HHL06020
      G(JJ+1) = GG                                                  HHL06030
      IEE = MP(JJ)                                                  HHL06040
      MP(JJ) = MP(JJ+1)                                             HHL06050
  160 MP(JJ+1) = IEE                                                HHL06060
C                                                                  HHL06070
  170 CONTINUE                                                      HHL06080
C                                                                  HHL06090
      WRITE(9,180)                                                  HHL06100
  180 FORMAT(5X,'K',6X,'A-MINGAP',5X,'TRUE A-EIGENVALUE(I)',2X,'A-EVNO')HHL06110
C                                                                  HHL06120
      WRITE(9,190)(J,G(J),U(J),MP(J), J=1,NEXACT)                   HHL06130
  190 FORMAT(I6,E14.3,E25.16,I8)                                    HHL06140
C                                                                  HHL06150
      WRITE(9,200)                                                  HHL06160
  200 FORMAT(' NEXACT DISTINCT A-EIGENVALUES. GAPS IN ASCENDING ORDER'/ HHL06170
     2 ' A-MINGAP(I) = TRUE MINIMUM GAP IN A FOR EACH EIGENVALUE.'/     HHL06180
     3 ' A-MINGAP(I) LT 0 INDICATES THE LEFT-HAND GAP WAS MINIMAL.'/    HHL06190
     3 ' A-MATRIX IS BLOCK TRIDIAGONAL AND EACH DIAGONAL BLOCK IS OF ORDHHL06200
     3ER NX.'/                                                     HHL06210
     4 ' NX = NUMBER OF POINTS ON EACH X-LINE. THERE ARE NY DIAGONAL BLOHHL06220
     4CKS.'/                                                       HHL06230
     5 ' NY = NUMBER OF POINTS ON EACH Y-LINE.'/                   HHL06240
     5 ' A-DIAGONAL   = A(K,K)'/                                   HHL06250
     6 ' X-CODIAGONAL = A(I,I+1)'/                                 HHL06260
     7 ' Y-CODIAGONAL = A(I,I+NX)'/                                HHL06270
     8 ' ----- END OF FILE 9 EXACTEV---------------------------'//) HHL06280
C                                                                  HHL06290
C-----END OF EXEVG-------------------------------------------------HHL06300
C                                                                  HHL06310
      RETURN                                                        HHL06320
      END                                                           HHL06330
C                                                                  HHL06340
```

```
C-----START OF HEXVEC----------------------------------------------------HHL06350
C                                                                        HHL06360
C     FOR THE HERMITIAN POISSON TEST CASES WITH MTYPE = 0 ONLY:          HHL06370
C     FOR A GIVEN RITZ VECTOR V AND EIGENVALUE X1, COMPUTES              HHL06380
C     THE CLOSEST TRUE EIGENVALUE Y1 AND CORRESPONDING TRUE              HHL06390
C     EIGENVECTOR Z, CALCULATES THE NORM OF V-Z AND THE MAXIMAL          HHL06400
C     DIFFERENCE OF THE COMPONENTS.  USER WOULD HAVE TO                  HHL06410
C     INCORPORATE ENTRY AND CALL TO THIS SUBROUTINE INTO                 HHL06420
C     HLEVEC PROGRAM IF THESE QUANTITIES ARE DESIRED.                    HHL06430
C     U CONTAINS THE COMPUTED TRUE EIGENVALUES.                          HHL06440
C     W CONTAINS THE TRUE EIGENVECTOR FOR THE REAL POISSON MATRIX        HHL06450
C                                                                        HHL06460
      SUBROUTINE HEXVEC(Z,V,U,W,X1,Y1,MP,JNUM)                           HHL06470
C                                                                        HHL06480
C----------------------------------------------------------------------- HHL06490
      DOUBLE PRECISION  U(*),W(*)                                        HHL06500
      DOUBLE PRECISION WI(110),WJ(110),WII(110)                          HHL06510
      DOUBLE PRECISION X1,Y1,EV,EE,WS,PIK,PIL,SUM,TEMP                   HHL06520
      DOUBLE PRECISION ATOLN,EPSM,ZERO,HALF,ONE,TWO,MACHEP               HHL06530
      DOUBLE PRECISION C0,C1,C2,T0,T1,T2                                 HHL06540
      COMPLEX*16 CONE,S,SB,STEMP,V(1),Z(1)                              HHL06550
      INTEGER MP(1)                                                      HHL06560
C----------------------------------------------------------------------- HHL06570
      DATA MACHEP/Z3410000000000000/                                    HHL06580
      EPSM = 2.0D0*MACHEP                                                HHL06590
C----------------------------------------------------------------------- HHL06600
C     THIS PROGRAM CALCULATES THE EXACT EIGENVALUES AND EIGENVECTORS     HHL06610
C     OF THE HERMITIAN POISSON MATRIX A OF ORDER  N = KX BY KY           HHL06620
C     A CONSISTS OF KY TRIDIAGONAL BLOCKS OF ORDER KX                    HHL06630
C     KX = X-DIMENSION     KY = Y-DIMENSION.                             HHL06640
C                                                                        HHL06650
C     C2 = DIAGONAL OF KX BY KX MATRIX                                   HHL06660
C    -C0 = CO-DIAGONAL OF THE KX BY KX MATRIX.                           HHL06670
C    -C1 = Y-CODIAGONAL.                                                 HHL06680
C                                                                        HHL06690
C     NOTE THAT THE VECTORS WI,WJ,WII ARE DIMENSIONED INTERNALLY         HHL06700
C     THEY ARE USED JUST TO KEEP FROM REGENERATING INFORMATION.          HHL06710
C     WI,WII = REAL*8 ARRAYS OF DIMENSION AT LEAST KX                    HHL06720
C     WJ     = REAL*8 ARRAY  OF DIMENSION AT LEAST KY.                   HHL06730
C                                                                        HHL06740
C     NOTATION USED IN PROGRAM                                           HHL06750
C                                                                        HHL06760
C     PIK = ARCOS(-1)/(KX+1)    PIL = ARCOS(-1)/(KY+1)                   HHL06770
C     WI(I) = PIK*I        WJ(J) = PIL*J                                 HHL06780
C                                                                        HHL06790
C     T0 = C2 - 2*C1*COS(PIL*J)   EV(I,J) = T0 - 2*C0*COS(PIK*I)         HHL06800
C     I = 1,KX    J = 1,KY    KP = (J-1)*KX + I                          HHL06810
C                                                                        HHL06820
C     W(KV) = SIN(PIK*I*IK)*SIN(PIL*J*JK)                                HHL06830
C     IK = 1,KX    JK = 1,KY    KV = (JK-1)*KX + IK                      HHL06840
C     W IS UNSCALED EIGENVECTOR FOR EV(I,J)                              HHL06850
C     WS = 1/||W||: ||W|| = .5*DSQRT(T2*T3)  T2 = KX+1  T3 = KY+1        HHL06860
C     U(K) IS A-EV ORDERED BY INCREASING SIZE, K = 1,N                   HHL06870
C                                                                        HHL06880
C     GIVEN X1 FIND Y1 AND KVEC SUCH THAT                                HHL06890
```

```
C           Y1 = EV(KVEC) AND |X1-Y1| = MIN                    HHL06900
C           ALSO GIVEN UNIT RITZ VECTOR ASSOCIATED WITH X1     HHL06910
C           CALCULATE UNIT EIGENVECTOR W, A*W = Y1*W           HHL06920
C           T2 = ||V-W||  T1 = MAX(|V(K)-W(K)|, K= 1,N)        HHL06930
C           MAX OCCURS FIRST AT K = KK                         HHL06940
C                                                              HHL06950
C--------------------------------------------------------------HHL06960
C     C2 = A(K,K)                                              HHL06970
C     C0 = A(K,K+1) = A(K+1,K)                                 HHL06980
C     C1 = A(K,K+KX) = A(K+KX,K)                               HHL06990
C     C0 + C1 = HALF                                           HHL07000
C                                                              HHL07010
      GO TO 3                                                  HHL07020
C                                                              HHL07030
C--------------------------------------------------------------HHL07040
      ENTRY EXVECP(SB,C0,C1,C2,KX,KY)                          HHL07050
C--------------------------------------------------------------HHL07060
      GO TO 4                                                  HHL07070
C                                                              HHL07080
    3 CONTINUE                                                 HHL07090
C                                                              HHL07100
C     SPECIFY PARAMETERS                                       HHL07110
      N = KX*KY                                                HHL07120
      ZERO = 0.0D0                                             HHL07130
      HALF = 0.5D0                                             HHL07140
      ONE  = 1.0D0                                             HHL07150
      TWO  = 2.0D0                                             HHL07160
      T0 = DACOS(-ONE)                                         HHL07170
      T1 = DFLOAT(KX+1)                                        HHL07180
      PIK = T0/T1                                              HHL07190
      T2 = DFLOAT(KY+1)                                        HHL07200
      PIL = T0/T2                                              HHL07210
      WS = TWO/DSQRT(T1*T2)                                    HHL07220
C                                                              HHL07230
C     GENERATE WI WJ VECTORS                                   HHL07240
      KP = 0                                                   HHL07250
      DO 20 J = 1,KY                                           HHL07260
      T1 = PIL*DFLOAT(J)                                       HHL07270
      WJ(J) = T1                                               HHL07280
      T0 = C2 - TWO*C1*DCOS(T1)                                HHL07290
      DO 10 I = 1,KX                                           HHL07300
      KP = KP+1                                                HHL07310
      T1 = PIK*DFLOAT(I)                                       HHL07320
      WI(I) = T1                                               HHL07330
   10 U(KP) = T0 - TWO*C0*DCOS(T1)                             HHL07340
   20 CONTINUE                                                 HHL07350
C     U(KP) = EV(I,J) = C2 - 2*C1*COS(PIL*J) - 2*C0*COS(PIK*I) HHL07360
C                                                              HHL07370
C     INITIALIZE MP VECTOR                                     HHL07380
      DO 30 K = 1,N                                            HHL07390
   30 MP(K) = K                                                HHL07400
C                                                              HHL07410
C     WE ORDER U VECTOR BY INCREASING SIZE OF THE EVS          HHL07420
      DO 50 K = 2,N                                            HHL07430
      KM1 = K-1                                                HHL07440
```

```
C                                                             HHL07450
      DO 40 L = 1,KM1                                         HHL07460
      JJ = K - L                                             HHL07470
      IF (U(JJ+1).GE.U(JJ)) GO TO 50                         HHL07480
      EE = U(JJ)                                             HHL07490
      U(JJ) = U(JJ+1)                                        HHL07500
      U(JJ+1) = EE                                           HHL07510
      IEE = MP(JJ)                                           HHL07520
      MP(JJ) = MP(JJ+1)                                      HHL07530
   40 MP(JJ+1) = IEE                                         HHL07540
C                                                             HHL07550
   50 CONTINUE                                                HHL07560
C                                                             HHL07570
      ATOLN = DMAX1(DABS(U(1)),DABS(U(N)))*EPSM              HHL07580
C                                                             HHL07590
      WRITE(6,60) N,KX,KY,C2,C0,C1,ATOLN                    HHL07600
   60 FORMAT(/' EXACT ERRORS FOR CONVERGED GOODEV'/          HHL07610
     1 4I6,' = N KX KY'//                                    HHL07620
     1 4E12.5,' = C2 C0 C1 ATOLN'//)                         HHL07630
C                                                             HHL07640
C     KP = MP(K) MEANS EIGENVALUE U(K) CORRESPONDS TO EIGENVECTOR W(KP) HHL07650
C     COMPUTE TOLERANCE USED IN COMPUTING TRUE MULTIPLICITIES HHL07660
C                                                             HHL07670
C     X1 IS AN INPUT PARAMETER. WE CALCULATE EXACT          HHL07680
C     A-EIGENVALUE WHICH IS CLOSEST TO X1, LABEL IT Y1 AND CALCULATE HHL07690
C     UNIT EIGENVECTOR OF A ASSOCIATED WITH Y1. A*W = Y1*W, ||W|| = 1. HHL07700
C     Y1 = U(KEV). EIGENVALUES OF A ARE ORDERED BY INCREASING SIZE. HHL07710
C     V = COMPLEX RITZ VECTOR ASSOCIATED WITH GOODEV X1     HHL07720
C     WE SHOULD HAVE V = D*W WHERE D = DIAG(D(1),D(2),..,D(N)) HHL07730
C     D(1) = ONE, D(K+1)/D(K) = SB, |SB| = ONE              HHL07740
C                                                             HHL07750
      KX1 = 0                                                HHL07760
      IF (X1.LE.U(1)) KX1 = 1                                HHL07770
      IF (X1.GE.U(N)) KX1 = N                                HHL07780
      NM1 = N-1                                              HHL07790
      IF (KX1.NE.0) GO TO 80                                 HHL07800
C                                                             HHL07810
      DO 70 KVEC = 2,N                                       HHL07820
      IF (X1.GE.U(KVEC)) GO TO 70                            HHL07830
C     U(KVEC-1).LE.X1.LT.U(KVEC)                            HHL07840
      T1 = X1 - U(KVEC-1)                                    HHL07850
      T2 = U(KVEC) - X1                                      HHL07860
      KX1 = KVEC - 1                                         HHL07870
      IF (T1.GT.T2) KX1 = KVEC                               HHL07880
      GO TO 80                                               HHL07890
   70 CONTINUE                                                HHL07900
C                                                             HHL07910
   80 Y1 = U(KX1)                                            HHL07920
C                                                             HHL07930
      IF (KX1.EQ.1) EE = U(2) - U(1)                         HHL07940
      IF (KX1.EQ.N) EE = U(N) - U(NM1)                       HHL07950
      IF (KX1.EQ.1.OR.KX1.EQ.N) GO TO 90                     HHL07960
      EE = DMIN1(U(KX1+1)-U(KX1),U(KX1)-U(KX1-1))            HHL07970
   90 CONTINUE                                                HHL07980
C                                                             HHL07990
```

```
      T0 = DABS(ONE - X1/Y1)                                    HHL08000
C                                                               HHL08010
      WRITE(6,100) N,KX1,JNUM,Y1,X1,T0,EE                       HHL08020
  100 FORMAT(3I8,' = N, A-EV NUMBER,GOODEV NO'//                HHL08030
     1 18X,'EXACTEV',19X,'GOODEV',4X,'RELERROR',4X,'A-MINGAP'/  HHL08040
     1 2E25.16,2E12.3/)                                         HHL08050
C                                                               HHL08060
      IF (EE.GT.ATOLN) GO TO 120                                HHL08070
C                                                               HHL08080
      WRITE(6,110)                                              HHL08090
  110 FORMAT(' Y1 IS A MULTIPLE EIGENVALUE OF A SO WE EXIT'/)   HHL08100
C                                                               HHL08110
      GO TO 200                                                 HHL08120
C                                                               HHL08130
C     Y1 IS TOEPLITZ EIGENVALUE CLOSEST TO X1.                  HHL08140
C     CALCULATION OF EIGENVECTOR ASSOCIATED WITH EIGENVALUE Y1  HHL08150
C     A*W = Y1*W                                                HHL08160
C                                                               HHL08170
C     DETERMINE I J FROM K: MP(K) = KP = (J-1)*KX+I             HHL08180
  120 CONTINUE                                                  HHL08190
      K = KX1                                                   HHL08200
      KP = MP(K)                                                HHL08210
      I = MOD(KP,KX)                                            HHL08220
      IF (I.EQ.0) I = KX                                        HHL08230
      T1 = WI(I)                                                HHL08240
      J = 1 + (KP-1)/KX                                         HHL08250
      T2 = WJ(J)                                                HHL08260
C                                                               HHL08270
      DO 130 II = 1,KX                                          HHL08280
      T0 = T1*DFLOAT(II)                                        HHL08290
  130 WII(II) = WS*DSIN(T0)                                     HHL08300
C                                                               HHL08310
      KV = 0                                                    HHL08320
      DO 150 JJ = 1,KY                                          HHL08330
      T0 = T2*DFLOAT(JJ)                                        HHL08340
      T0 = DSIN(T0)                                             HHL08350
C                                                               HHL08360
      DO 140 II = 1,KX                                          HHL08370
      KV = KV + 1                                               HHL08380
  140 W(KV) = T0*WII(II)                                        HHL08390
C                                                               HHL08400
  150 CONTINUE                                                  HHL08410
C                                                               HHL08420
C     W IS UNIT EXACT EIGENVECTOR OF A ASSOCIATED WITH Y1       HHL08430
C     V IS UNIT COMPLEX RITZVECTOR OF B ASSOCIATED WITH X1      HHL08440
C                                                               HHL08450
      CONE = DCMPLX(ONE,ZERO)                                   HHL08460
      STEMP = CONE                                              HHL08470
      DO 160 K = 1,N                                            HHL08480
      Z(K) = STEMP*W(K)                                         HHL08490
  160 STEMP = STEMP*SB                                          HHL08500
C                                                               HHL08510
      T1 = ZERO                                                 HHL08520
      S = ONE                                                   HHL08530
      KK = 0                                                    HHL08540
```

```
      DO 170 K = 1,N                                              HHL08550
      IF (CDABS(Z(K)).LE.T1) GO TO 170                            HHL08560
      T1 = CDABS(Z(K))                                            HHL08570
      KK = K                                                      HHL08580
  170 CONTINUE                                                    HHL08590
C                                                                 HHL08600
      S = V(KK)/Z(KK)                                             HHL08610
C                                                                 HHL08620
      KK = 0                                                      HHL08630
      T1 = ZERO                                                   HHL08640
      T2 = ZERO                                                   HHL08650
      DO 180 K = 1,N                                              HHL08660
      TEMP = CDABS(S*Z(K) - V(K))                                 HHL08670
      T2 = T2 + TEMP**2                                           HHL08680
      IF (TEMP.LE.T1) GO TO 180                                   HHL08690
      KK = K                                                      HHL08700
      T1 = TEMP                                                   HHL08710
  180 CONTINUE                                                    HHL08720
C                                                                 HHL08730
      T2 = DSQRT(T2)                                              HHL08740
      WRITE(6,190) KK,T1,T2                                       HHL08750
  190 FORMAT(' EIGENVECTOR ERROR. MAX ERROR AT COMPONENT = ',I6/  HHL08760
     1 ' MAX CDABS(EXACTVEC(K)-RITZVEC(K)) = ',E12.5/             HHL08770
     1 ' NORM(EXACTVEC-RITZVEC) = ',E12.5/)                       HHL08780
C                                                                 HHL08790
  200 CONTINUE                                                    HHL08800
C                                                                 HHL08810
C-----END OF HEXVEC-----------------------------------------------HHL08820
    4 RETURN                                                      HHL08830
      END                                                         HHL08840
C                                                                 HHL08850
C-----USPEC (TRIDIAGONAL HERMITIAN MATRICES)---------------------HHL08860
C                                                                 HHL08870
C     SUBROUTINE USPEC(N,MATNO)                                   HHL08880
      SUBROUTINE TSPEC(N,MATNO)                                   HHL08890
C                                                                 HHL08900
C---------------------------------------------------------------HHL08910
      DOUBLE PRECISION  D(100), DAR(100),DAI(100), PI, EIGVAL(100) HHL08920
      DOUBLE PRECISION SPACE                                      HHL08930
      COMPLEX*16   DA(100),DB(100)                                HHL08940
      REAL EXPLAN(20)                                             HHL08950
C---------------------------------------------------------------HHL08960
C     DIMENSION ARRAYS NEEDED TO DEFINE MATRIX.  THEN             HHL08970
C     PASS THE STORAGE LOCATIONS OF THESE ARRAYS TO THE MATRIX-VECTOR HHL08980
C     MULTIPLY SUBROUTINE CMATV.                                  HHL08990
C                                                                 HHL09000
C     DIAGONAL ENTRY = D, ABOVE DIAGONAL ENTRY = DA, BELOW DIAGONAL = DB.HHL09010
C                                                                 HHL09020
      READ(8,10) EXPLAN                                           HHL09030
   10 FORMAT(20A4)                                                HHL09040
      READ(8,*) NOLD,MATOLD                                       HHL09050
C                                                                 HHL09060
      WRITE(6,20) N,MATOLD                                        HHL09070
   20 FORMAT(I10,2I6,I8,' = N,MATOLD'/)                           HHL09080
C                                                                 HHL09090
```

```
C     TEST OF PARAMETER CORRECTNESS                         HHL09100
      ITEMP = (NOLD-N)**2 + (MATNO-MATOLD)**2               HHL09110
C                                                           HHL09120
      IF(ITEMP.EQ.0) GO TO 40                               HHL09130
C                                                           HHL09140
      WRITE(6,30)                                           HHL09150
   30 FORMAT(' PROGRAM TERMINATES BECAUSE EITHER ORDERS OF OR LABELS FORHHL09160
     1 MATRIX DISAGREE')                                    HHL09170
      GO TO 250                                             HHL09180
C                                                           HHL09190
   40 CONTINUE                                              HHL09200
C                                                           HHL09210
C     IF ITOEP = 1 THEN MATRIX IS TOEPLITZ AND WE PRINT OUT TRUE  HHL09220
C     EIGENVALUES                                           HHL09230
      READ(8,10) EXPLAN                                     HHL09240
      READ(8,*) ITOEP                                       HHL09250
      READ(8,10) EXPLAN                                     HHL09260
C                                                           HHL09270
      IF(ITOEP.EQ.1) WRITE(6,50)                            HHL09280
   50 FORMAT(/' TEST MATRIX IS HERMITIAN TOEPLITZ'/)        HHL09290
      IF(ITOEP.NE.1) GO TO 110                              HHL09300
C                                                           HHL09310
      READ(8,*) DAR(1),DAI(1),D(1)                          HHL09320
      DA(1) = DCMPLX(DAR(1),DAI(1))                         HHL09330
      DB(1) = DCONJG(DA(1))                                 HHL09340
      DO 60 J=2,N                                           HHL09350
      D(J) = D(1)                                           HHL09360
      DA(J) = DA(1)                                         HHL09370
   60 DB(J) = DB(1)                                         HHL09380
      WRITE(6,70) DB(1),D(1),DA(1)                          HHL09390
      WRITE(9,70) DB(1),D(1),DA(1)                          HHL09400
   70 FORMAT(' HERMITIAN TOEPLITZ MATRIX IS USED.'/' BELOW DIAGONAL ENTRHHL09410
     1Y = ',2E12.3/' DIAGONAL ENTRY = ',E12.3/' ABOVE DIAGONAL ENTRY =' HHL09420
     1,2E12.3)                                              HHL09430
C                                                           HHL09440
C     COMPUTE THE TRUE EIGENVALUES.  FORMULA IS CORRECT ONLY FOR THOSE  HHL09450
C     MATRICES WHOSE DIAGONAL = 2., ABOVE DIAGONAL = A, BELOW DIAGONAL  HHL09460
C     = A-CONJUGATE, AND A HAS NORM 1.                      HHL09470
C                                                           HHL09480
      PI = DACOS(-1.D0)                                     HHL09490
      DO 80 J=1,N                                           HHL09500
   80 EIGVAL(J) = 2.D0 * (1.D0 -DCOS(PI*DFLOAT(J)/DFLOAT(N+1)))  HHL09510
      WRITE(9,90) N                                         HHL09520
   90 FORMAT(I6, ' = ORDER OF MATRIX'/' TRUE EIGENVALUES ARE'/)  HHL09530
      WRITE(9,100) (J, EIGVAL(J), J=1,N)                    HHL09540
  100 FORMAT(I5,4X,E25.16,6X,I5,4X,E25.16)                  HHL09550
      GO TO 240                                             HHL09560
C                                                           HHL09570
C     NONTOEPLITZ HERMITIAN.  DIAGONAL ENTRIES ARE EQUALLY-SPACED.  HHL09580
C     ABOVE DIAGONAL ENTRIES ARE GENERATED BY GENERATING EQUALLY-SPACED HHL09590
C     REAL PARTS, AND EQUALLY-SPACED IMAGINARY PARTS.  THE BELOW    HHL09600
C     DIAGONAL ENTRIES ARE THEN OBTAINED BY TAKING THE COMPLEX CONJUGATEHHL09610
C     OF THE ABOVE DIAGONAL ENTRIES                         HHL09620
C                                                           HHL09630
  110 READ(8,*) D(1), SPACE                                 HHL09640
```

```
      WRITE(6,120) D(1),SPACE                                    HHL09650
  120 FORMAT(' 1ST DIAGONAL ENTRY =',E20.12,' SPACING =',E20.12)  HHL09660
      DO 130 J=2,N                                               HHL09670
  130 D(J) = D(J-1) + SPACE                                      HHL09680
      WRITE(6,140) (D(J), J=1,3)                                 HHL09690
  140 FORMAT(' 1ST THREE DIAGONAL ENTRIES ='/(2E20.12))          HHL09700
      READ(8,10) EXPLAN                                          HHL09710
      READ(8,*) DAR(1), SPACE                                    HHL09720
      WRITE(6,150) DAR(1),SPACE                                  HHL09730
  150 FORMAT(' REAL PART OF 1ST ABOVE DIAGONAL ENTRY =',E20.12,/  HHL09740
     1' SPACING = ',E20.12)                                      HHL09750
      DO 160 J=2,N                                               HHL09760
  160 DAR(J) = DAR(J-1) + SPACE                                  HHL09770
      WRITE(6,170) (DAR(J), J=1,3)                               HHL09780
  170 FORMAT(' REAL PARTS OF 1ST THREE ABOVE DIAGONAL ENTRIES ='/  HHL09790
     1(2E20.12))                                                 HHL09800
      READ(8,10) EXPLAN                                          HHL09810
      READ(8,*) DAI(1), SPACE                                    HHL09820
      WRITE(6,180) DAI(1),SPACE                                  HHL09830
  180 FORMAT(' IMAGINARY PART OF 1ST ABOVE =',E20.12,/' SPACING =',  HHL09840
     1 E20.12)                                                   HHL09850
      DO 190 J=2,N                                               HHL09860
  190 DAI(J) = DAI(J-1) + SPACE                                  HHL09870
      WRITE(6,200) (DAI(J), J = 1,3)                             HHL09880
  200 FORMAT(' IMAGINARY PARTS OF 1ST THREE ABOVE DIAGONAL ENTRIES ='/  HHL09890
     1 (2E20.12))                                                HHL09900
      DO 210 J=1,N                                               HHL09910
      DA(J) = DCMPLX(DAR(J),DAI(J))                              HHL09920
  210 DB(J) = DCONJG(DA(J))                                      HHL09930
C                                                                HHL09940
      WRITE(9,220) (D(J), J=1,N)                                 HHL09950
  220 FORMAT(' DIAGONAL ENTRIES ='/(4E20.12))                    HHL09960
      WRITE(9,230) (DA(J), J=1,N)                                HHL09970
  230 FORMAT(' ABOVE DIAGONAL ENTRIES'/(4E20.12))                HHL09980
C                                                                HHL09990
C     PASS STORAGE LOCATIONS OF ARRAYS THAT DEFINE THE MATRIX TO  HHL10000
C     THE MATRIX-VECTOR MULTIPLY SUBROUTINE CMATV                 HHL10010
C                                                                HHL10020
  240 CONTINUE                                                   HHL10030
C                                                                HHL10040
C---------------------------------------------------------------HHL10050
      CALL TMATVE(DA,DB,D,N)                                     HHL10060
C---------------------------------------------------------------HHL10070
C                                                                HHL10080
      RETURN                                                     HHL10090
  250 STOP                                                       HHL10100
C                                                                HHL10110
C-----END OF USPEC----------------------------------------------HHL10120
      END                                                        HHL10130
C                                                                HHL10140
C-----START OF MATRIX-VECTOR MULTIPLY (HERMITIAN TRIDIAGONAL)----HHL10150
C                                                                HHL10160
C     SUBROUTINE CMATV(W,U,SUM)                                  HHL10170
      SUBROUTINE TMATV(W,U,SUM)                                  HHL10180
C                                                                HHL10190
```

```
C-----------------------------------------------------------------HHL10200
      COMPLEX*16   U(1),W(*),DA(1),DB(1)                           HHL10210
      DOUBLE PRECISION  D(1),SUM                                   HHL10220
C-----------------------------------------------------------------HHL10230
C    HERMITIAN MATRIX-VECTOR MULTIPLY FOR LANCZS  U = A*W - SUM*U  HHL10240
C    MATRIX IS TRIDIAGONAL HERMITIAN TOEPLITZ                      HHL10250
C-----------------------------------------------------------------HHL10260
C                                                                  HHL10270
C    COMPUTE A*W - SUM*U                                           HHL10280
C                                                                  HHL10290
      GO TO 3                                                      HHL10300
C-----------------------------------------------------------------HHL10310
C    STORAGE LOCATIONS ARE PASSED TO CMATV FROM USPEC              HHL10320
      ENTRY TMATVE(DA,DB,D,N)                                      HHL10330
      GO TO 4                                                      HHL10340
C-----------------------------------------------------------------HHL10350
    3 CONTINUE                                                     HHL10360
C                                                                  HHL10370
      U(1) = D(1)*W(1) + DA(1)*W(2) - SUM*U(1)                     HHL10380
      N1 = N-1                                                     HHL10390
      DO 10 I = 2,N1                                               HHL10400
   10 U(I) = DB(I-1)*W(I-1)+D(I)*W(I) + DA(I)*W(I+1) -SUM*U(I)     HHL10410
      U(N) = DB(N-1)*W(N-1) + D(N)*W(N) - SUM*U(N)                 HHL10420
C                                                                  HHL10430
    4 RETURN                                                       HHL10440
C                                                                  HHL10450
C-----END OF CMATV------------------------------------------------HHL10460
      END                                                          HHL10470
C-----DUMMY USPEC DOES NOTHING------------------------------------HHL10480
C                                                                  HHL10490
      SUBROUTINE USPEC(N,MATNO)                                    HHL10500
C     SUBROUTINE CUSPEC(N,MATNO)                                   HHL10510
C                                                                  HHL10520
C-----------------------------------------------------------------HHL10530
      RETURN                                                       HHL10540
      END                                                          HHL10550
```

3.5   HLEVAL: HLEVEC: File Definitions, Sample Input Files

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

```
Sample Specifications of the input/output files for HLEVAL
-----------------------------------------------------------------
 HLEVAL EXEC HERMITIAN EIGENVALUE CALCULATION
FI 06 TERM
FILEDEF  1 DISK &1        NHISTORY  A (RECFM F LRECL 80 BLOCK 80
FILEDEF  2 DISK &1        HISTORY   A (RECFM F LRECL 80 BLOCK 80
FILEDEF  3 DISK &1        GOODEV    A (RECFM F LRECL 80 BLOCK 80
FILEDEF  4 DISK &1        ERRINV    A (RECFM F LRECL 80 BLOCK 80
FILEDEF  5 DISK HLEVAL    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   HLEVAL    LESUB   HLEMULT
-----------------------------------------------------------------
```

```
Sample Input File for HLEVAL
-----------------------------------------------------------------
 HLEVAL INPUT EIGENVALUE COMPUTATION, NO REORTHOGONALIZATION
 HERMITIAN TEST MATRIX
LINE 1     N      KMAX     NMEVS      MATNO
          528    1600        3     721830
LINE 2   SVSEED    RHSEED       MXINIT    MXSTUR
        49302312  5731029          5    100000
LINE 3   ISTART     ISTOP
             0         1
LINE 4    IHIS      IDIST   IWRITE
             1         0        1
LINE 5   RELTOL  (RELATIVE TOLERANCE IN 'COMBINING' GOODEV)
       .0000000001
LINE 6   MB(1)    MB(2)    MB(3)    MB(4)     (ORDERS OF T(1,MEV) )
           528    1056     1584
LINE 7    NINT     (NUMBER OF SUB-INTERVALS FOR BISEC)
             1
LINE 8    LB(1)    LB(2)    LB(3)    LB(4)  (INTERVAL LOWER BOUNDS)
           1.0
LINE 9    UB(1)    UB(2)    UB(3)    UB(4)  (INTERVAL UPPER BOUNDS)
           2.0
-----------------------------------------------------------------
```

Below is a listing of the input/output files definitions which are accessed by the Hermitian Lanczos eigenvector program, HLEVEC. Included also is a sample of the input file which HLEVEC requires on file 5. The parameters are supplied in free format. HLEVEC computes eigenvectors for each of a user-specified subset of the eigenvalues computed by the companion code HLEVEC. Eigenvector approximations will be computed only for eigenvalue approximations which have converged.

```
Sample Specifications of the Input/Output Files for HLEVEC
--------------------------------------------------------------------
 HLEVEC EXEC TO RUN LANCZOS EIGENVECTOR PROGRAM, HERMITIAN MATRICES
FI 06 TERM
FILEDEF  2 DISK &1       HISTORY   A (RECFM F LRECL 80 BLOCK 80
FILEDEF  3 DISK &1       GOODEV    A (RECFM F LRECL 80 BLOCK 80
FILEDEF  4 DISK &1       ERRINV    A (RECFM F LRECL 80 BLOCK 80
FILEDEF  5 DISK HLEVEC   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  HLEVEC  LESUB  HLEMULT
--------------------------------------------------------------------
```

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

Chapter 4

Factored Inverses of Real Symmetric Matrices

4.1 Introduction

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

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

$$B^{-1}x = \lambda x, \qquad (4.1.1)$$

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

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

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

reasonable size.

The single-vector Lanczos codes in this chapter can be used to compute either a very few or very many of the distinct eigenvalues of the given real symmetric matrix. The documentation for these codes is contained in Chapter 2, Section 2.2. As in the direct real symmetric case (Chapter 2, Section 2.1), the A-multiplicity of a given computed eigenvalue can be obtained only with additional computation, and the modifications required to do this additional computation are not included in these versions of the codes. This implementation uses the basic Lanczos recursion contained in Eqns (1.2.1) and (1.2.2) to generate a family of real symmetric tridiagonal matrices (T-matrices) for the matrix B^{-1} , whose sizes are specified by the user. Specifically, for $i = 1, 2, ..., m$ and a randomly-generated starting vector v_1 with $\|v_1\| = 1$, generate Lanczos vectors v_i using the following recursion and Eqn(1.2.2) applied to the matrix B^{-1} .

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

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

LIVAL, the main program for the factored inverse computations, calls the subroutine BISEC to compute eigenvalues of the specified Lanczos tridiagonal matrices on the user-specified intervals. BISEC simultaneously computes these T-eigenvalues with their T-multiplicities and sorts the computed T-eigenvalues into two classes, the 'good' T-eigenvalues and the 'spurious' T-eigenvalues. The 'good' T-eigenvalues are accepted as approximations to eigenvalues of the B^{-1} matrix associated with the user-specified matrix A, scalars $SCALE$ and $SHIFT$, and the permutation matrix P (if any). The accuracy of these 'good' T-eigenvalues as eigenvalues of B^{-1} is then estimated using error estimates computed by subroutine INVERR. Error estimates are computed only for isolated 'good' T-eigenvalues. All other 'good' T-eigenvalues are assumed to have converged.

Convergence is then checked. If convergence has not yet occurred and a larger T-matrix has been specified by the user, the program will continue on to the larger T-matrix, repeating the above procedure on this larger matrix. After each T-matrix eigenvalue computation, the corresponding approximations to the eigenvalues of the user-specified matrix A are computed and included in the output.

Once the eigenvalues of B^{-1} have been computed accurately enough, the user can select a subset of the 'converged' eigenvalues for which eigenvectors are to be computed. The main program LIVEC, for computing eigenvectors of the inverse of a real symmetric matrix, given a factorization, is used to compute the desired eigenvectors. If the matrix B is a permutation of the matrix C, then LIVEC unwinds the permutation to obtain the corresponding eigenvectors of the user-supplied A-matrix.

All of the computations are done in double precision real arithmetic. Once the Lanczos T-matrices have been computed, the remaining computations use the same subroutines that are used in the real symmetric case discussed in Chapter 2. In addition to the programs and subroutines provided here, the user must supply a subroutine USPEC which defines and initializes the factorization of the scaled, shifted, and permuted version B of the original matrix A, and a subroutine BSOLV which computes matrix-vector multiplies $B^{-1}x$ for any given vector x. These subroutines must be constructed in such a way as to take advantage of the sparsity (and/or structure) of the user-supplied A-matrix and such that these computations are done accurately.

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

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

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

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

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

4.2   LIVAL: Main Program, Eigenvalue Computations

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

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

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

```
C                                                                      LIV01570
C     READ IN THE NUMBER OF SUBINTERVALS TO BE CONSIDERED.             LIV01580
      READ(5,20) EXPLAN                                                LIV01590
      READ(5,*) NINT                                                   LIV01600
C                                                                      LIV01610
C     READ IN THE LEFT-END POINTS OF THE SUBINTERVALS TO BE CONSIDERED. LIV01620
C     THESE MUST BE IN ALGEBRAICALLY-INCREASING ORDER                 LIV01630
      READ(5,20) EXPLAN                                                LIV01640
      READ(5,*) (LB(J), J=1,NINT)                                      LIV01650
C                                                                      LIV01660
C     READ IN THE RIGHT-END POINTS OF THE SUBINTERVALS TO BE CONSIDERED.LIV01670
C     THESE MUST BE IN ALGEBRAICALLY-INCREASING ORDER                 LIV01680
      READ(5,20) EXPLAN                                                LIV01690
      READ(5,*) (UB(J), J=1,NINT)                                      LIV01700
C                                                                      LIV01710
C----------------------------------------------------------------------LIV01720
C     INITIALIZE THE ARRAYS FOR THE FACTORIZATION OF THE ASSOCIATED    LIV01730
C     SCALED, SHIFTED AND PERMUTED VERSION OF THE A-MATRIX.            LIV01740
C     THE STORAGE LOCATIONS OF THESE ARRAYS ARE PASSED TO THE BSOLV    LIV01750
C     SUBROUTINE WHICH WILL BE CALLED FROM LANCZS FOR THE T-MATRIX     LIV01760
C     GENERATION.                                                      LIV01770
C                                                                      LIV01780
      CALL USPEC(N,MATNO)                                              LIV01790
C                                                                      LIV01800
C----------------------------------------------------------------------LIV01810
C                                                                      LIV01820
C     MASKS UNDERFLOW AND OVERFLOW, USER MUST SUPPLY OR COMMENT OUT.   LIV01830
      CALL MASK                                                        LIV01840
C                                                                      LIV01850
C----------------------------------------------------------------------LIV01860
C                                                                      LIV01870
C     WRITE TO FILE 6, A SUMMARY OF THE PARAMETERS FOR THIS RUN        LIV01880
C                                                                      LIV01890
      WRITE(6,30) MATNO,N,KMAX,SHIFT,S0                                LIV01900
   30 FORMAT(/3X,'MATRIX ID',4X,'ORDER OF A',4X,'MAX ORDER OF T'//     LIV01910
     1 I12,I14,I18//8X,' SHIFT',8X,'SCALE'/2E15.6//                    LIV01920
     1 ' C = SCALE*A + SHIFT*I '/                                      LIV01930
     1 ' B = P*C*P-TRANSPOSE WHERE P IS A REORDERING OF C'/            LIV01940
     1 ' LANCZOS PROCEDURE USES THE FACTORIZATION OF B'/)              LIV01950
C                                                                      LIV01960
      WRITE(6,40) ISTART,ISTOP                                         LIV01970
   40 FORMAT(/2X,'ISTART',3X,'ISTOP'/2I8/)                             LIV01980
C                                                                      LIV01990
      WRITE(6,50) IHIS,IDIST,IWRITE                                    LIV02000
   50 FORMAT(/4X,'IHIS',3X,'IDIST',2X,'IWRITE'/3I8/)                   LIV02010
C                                                                      LIV02020
      WRITE(6,60) SVSEED,RHSEED                                        LIV02030
   60 FORMAT(/' SEEDS FOR RANDOM NUMBER GENERATOR'//                   LIV02040
     1 4X,'LANCZS SEED',4X,'INVERR SEED'/2I15/)                        LIV02050
C                                                                      LIV02060
      WRITE(6,70) (NMEV(J), J=1,NMEVS)                                 LIV02070
   70 FORMAT(/' SIZES OF T-MATRICES TO BE CONSIDERED'/(6I12))          LIV02080
C                                                                      LIV02090
      WRITE(6,80) RELTOL,GAPTOL,BTOL                                   LIV02100
   80 FORMAT(/' RELATIVE TOLERANCE USED TO COMBINE COMPUTED T-EIGENVALUELIV02110
```

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

```
C                                                                   LIV02670
      IF(IHIS.EQ.0.AND.ISTOP.GT.0) GO TO 170                         LIV02680
C                                                                   LIV02690
      WRITE(1,160) KMAX,N,SVSEED,MATNO,SHIFT                         LIV02700
  160 FORMAT(2I6,I12,I8,E13.4,' = KMAX,N,SVSEED,MATNO,SHIFT')        LIV02710
C                                                                   LIV02720
      WRITE(1,130)(ALPHA(I), I=1,KMAX)                              LIV02730
      WRITE(1,130)(BETA(I), I=1,KMAX1)                              LIV02740
C                                                                   LIV02750
      WRITE(1,130)(V1(I), I=1,N)                                    LIV02760
      WRITE(1,130)(V2(I), I=1,N)                                    LIV02770
C                                                                   LIV02780
      IF (ISTOP.EQ.0) GO TO 600                                     LIV02790
C                                                                   LIV02800
  170 CONTINUE                                                      LIV02810
      KMAX1 = KMAX + 1                                              LIV02820
      BKMIN = BTOL                                                  LIV02830
C                                                                   LIV02840
      WRITE(6,180)                                                  LIV02850
  180 FORMAT(/' T-MATRICES (ALPHA AND BETA) ARE NOW AVAILABLE'/)    LIV02860
C                                                                   LIV02870
C-------------------------------------------------------------------LIV02880
C     SUBROUTINE TNORM CHECKS MIN(BETA)/(ESTIMATED NORM(A)) > BTOL . LIV02890
C     IF THIS IS VIOLATED IB IS SET EQUAL TO THE NEGATIVE OF THE INDEXLIV02900
C     OF THE MINIMAL BETA.  IF(IB < 0) THEN SUBROUTINE TNORM IS      LIV02910
C     CALLED FOR EACH VALUE OF MEV TO DETERMINE WHETHER OR NOT THERE LIV02920
C     IS A BETA IN THE T-MATRIX SPECIFIED THAT VIOLATES THIS TEST.   LIV02930
C     IF THERE IS SUCH A BETA THE PROGRAM TERMINATES FOR THE USER    LIV02940
C     TO DECIDE WHAT TO DO.  THIS TEST CAN BE OVER-RIDDEN BY         LIV02950
C     SIMPLY MAKING BTOL SMALLER, BUT THEN THERE IS THE POSSIBILITY  LIV02960
C     THAT LOSSES IN THE LOCAL ORTHOGONALITY MAY HURT THE COMPUTATIONS.LIV02970
C     BTOL = 1.D-8 IS HOWEVER A CONSERVATIVE CHOICE FOR BTOL.        LIV02980
C                                                                   LIV02990
C     TNORM ALSO COMPUTES TKMAX = MAX(|ALPHA(K)|,BETA(K), K=1,KMAX). LIV03000
C     TKMAX IS USED TO SCALE THE TOLERANCES USED IN THE             LIV03010
C     T-MULTIPLICITY AND SPURIOUS TESTS IN BISEC. TKMAX IS ALSO USED INLIV03020
C     THE PROJECTION TEST FOR HIDDEN EIGENVALUES THAT HAD 'TOO SMALL' LIV03030
C     A PROJECTION ON THE STARTING VECTOR.                          LIV03040
C                                                                   LIV03050
      CALL TNORM(ALPHA,BETA,BKMIN,TKMAX,KMAX,IB)                    LIV03060
C                                                                   LIV03070
C-------------------------------------------------------------------LIV03080
      TTOL = EPSM*TKMAX                                             LIV03090
C                                                                   LIV03100
C     LOOP ON THE SIZE OF THE T-MATRIX                             LIV03110
  190 CONTINUE                                                      LIV03120
      MMB = MMB + 1                                                 LIV03130
      MEV = NMEV(MMB)                                               LIV03140
C     IS MEV TOO LARGE ?                                           LIV03150
      IF(MEV.LE.KMAX) GO TO 210                                    LIV03160
C                                                                   LIV03170
      WRITE(6,200) MMB, MEV, KMAX                                   LIV03180
  200 FORMAT(/' TERMINATE PRIOR TO CONSIDERING THE',I6,'TH T-MATRIX'/LIV03190
     1' BECAUSE THE SIZE REQUESTED',I6,' IS GREATER THAN THE MAXIMUM SIZLIV03200
     1E ALLOWED',I6/)                                               LIV03210
```

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

```
C------------------------------------------------------------------LIV03770
C     SUBROUTINE LUMP 'COMBINES' T-EIGENVALUES THAT ARE 'TOO CLOSE'.   LIV03780
C     NOTE HOWEVER THAT CLOSE SPURIOUS T-EIGENVALUES ARE NOT AVERAGED  LIV03790
C     WITH GOOD ONES. HOWEVER, THEY MAY BE USED TO INCREASE THE        LIV03800
C     T-MULTIPLICITY OF A GOOD T-EIGENVALUE.                           LIV03810
C                                                                      LIV03820
      LOOP = NDIS                                                      LIV03830
      CALL LUMP(VS,RELTOL,MULTOL,SCALE2,MP,LOOP)                       LIV03840
C                                                                      LIV03850
C------------------------------------------------------------------LIV03860
      IF(NDIS.EQ.LOOP) GO TO 250                                      LIV03870
C                                                                      LIV03880
      WRITE(6,240) NDIS, MEV, LOOP                                    LIV03890
  240 FORMAT(/I6,' DISTINCT T-EIGENVALUES WERE COMPUTED IN BISEC AT MEVLIV03900
     1',I6/ 2X,' LUMP SUBROUTINE REDUCES NUMBER OF DISTINCT T-EIGENVALUELIV03910
     1S TO',I6)                                                       LIV03920
C                                                                      LIV03930
  250 CONTINUE                                                        LIV03940
      NDIS = LOOP                                                     LIV03950
      BETA(MP1) = BETAM                                               LIV03960
C------------------------------------------------------------------LIV03970
C     THE SUBROUTINE ISOEV LABELS THOSE SIMPLE EIGENVALUES OF T(1,MEV) LIV03980
C     WITH VERY SMALL GAPS BETWEEN NEIGHBORING EIGENVALUES OF T(1,MEV) LIV03990
C     TO AVOID COMPUTING ERROR ESTIMATES FOR ANY SIMPLE GOOD          LIV04000
C     T-EIGENVALUE THAT IS TOO CLOSE TO A SPURIOUS EIGENVALUE.        LIV04010
C     ON RETURN FROM ISOEV, G CONTAINS CODED MINIMAL GAPS            LIV04020
C     BETWEEN THE DISTINCT EIGENVALUES OF T(1,MEV). (G IS REAL).      LIV04030
C     G(I) < 0 MEANS MINGAP IS DUE TO LEFT GAP G(I) > 0 MEANS DUE TO  LIV04040
C     RIGHT GAP. MP(I) = -1 MEANS THAT THE GOOD T-EIGENVALUE IS SIMPLE LIV04050
C     AND HAS A VERY SMALL MINGAP IN T(1,MEV) DUE TO A SPURIOUS       LIV04060
C     T-EIGENVALUE.  NG = NUMBER OF GOOD T-EIGENVALUES.               LIV04070
C     NISO = NUMBER OF ISOLATED GOOD T-EIGENVALUES.                   LIV04080
C                                                                      LIV04090
      CALL ISOEV(VS,GAPTOL,MULTOL,SCALE1,G,MP,NDIS,NG,NISO)           LIV04100
C                                                                      LIV04110
C------------------------------------------------------------------LIV04120
C                                                                      LIV04130
      WRITE(6,260)NG,NISO,NDIS                                        LIV04140
  260 FORMAT(/I6,' GOOD T-EIGENVALUES HAVE BEEN COMPUTED'/            LIV04150
     1 I6,' OF THESE ARE T-ISOLATED'/                                 LIV04160
     2 I6,' = NUMBER OF DISTINCT T-EIGENVALUES COMPUTED'/)            LIV04170
C                                                                      LIV04180
C     DO WE WRITE DISTINCT EIGENVALUES OF T-MATRIX TO FILE 11?        LIV04190
      IF (IDIST.EQ.0) GO TO 310                                       LIV04200
C                                                                      LIV04210
      WRITE(11,270) NDIS,NISO,MEV,N,SVSEED,MATNO                      LIV04220
  270 FORMAT(/4I6,I12,I8,' = NDIS,NISO,MEV,N,SVSEED,MATNO'/)          LIV04230
C                                                                      LIV04240
      WRITE(11,280)                                                   LIV04250
  280 FORMAT(/1X,'MP',21X,'EVBI',5X,'TMINGAP',1X,'MP',21X,'EVBI',5X,  LIV04260
     1'TMINGAP'/)                                                     LIV04270
C                                                                      LIV04280
      WRITE(11,290) (MP(I),VS(I),G(I), I=1,NDIS)                      LIV04290
  290 FORMAT(2(I3,E25.16,E12.3))                                      LIV04300
C                                                                      LIV04310
```

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

```
      MMB = NMEVS                                               LIV04870
C                                                               LIV04880
      WRITE(6,370) CONTROL                                      LIV04890
  370 FORMAT(' ALL COMPUTED ERROR ESTIMATES WERE LESS THAN',E15.4/  LIV04900
     1 ' THEREFORE PROCEDURE TERMINATES'/)                     LIV04910
C                                                               LIV04920
  380 CONTINUE                                                  LIV04930
C                                                               LIV04940
      IF (ICONV.EQ.0) GO TO 510                                 LIV04950
C                                                               LIV04960
C--------------------------------------------------------------- LIV04970
C     IF CONVERGENCE IS INDICATED, THAT IS ICONV = 1 ,THEN      LIV04980
C     THE SUBROUTINE PRTEST IS CALLED TO CHECK FOR ANY CONVERGED LIV04990
C     T-EIGENVALUES THAT HAVE BEEN MISLABELLED AS SPURIOUS BECAUSE LIV05000
C     THE PROJECTION OF THEIR EIGENVECTOR(S) ON THE STARTING    LIV05010
C     VECTOR WAS(WERE) TOO SMALL.                               LIV05020
C     NUMERICAL TESTS INDICATE THAT SUCH EIGENVALUES ARE RARE.  LIV05030
C     IF FOR SOME REASON MANY OF THESE HIDDEN EIGENVALUES APPEAR LIV05040
C     ON SOME RUN, YOU CAN BE CERTAIN THAT SOMETHING IS FOULED UP. LIV05050
C                                                               LIV05060
      CALL PRTEST(ALPHA,BETA,VS,TKMAX,EPSM,RELTOL,SCALE3,SCALE4, LIV05070
     1 MP,NDIS,MEV,IPROJ)                                       LIV05080
C                                                               LIV05090
C---------------------------------------------------------------LIV05100
C                                                               LIV05110
      IF(IPROJ.EQ.0) GO TO 500                                  LIV05120
C                                                               LIV05130
      IF(IDIST.EQ.1)  WRITE(11,390) IPROJ                       LIV05140
  390 FORMAT(' SUBROUTINE PRTEST WANTS TO RELABEL',I6,' SPURIOUS T-EIGENLIV05150
     1VALUES'/' WE ACCEPT RELABELLING ONLY IF LAST COMPONENT OF T-EIGENVLIV05160
     1ECTOR IS L.T.  1.D-10'/)                                  LIV05170
C                                                               LIV05180
      IIX = RHSEED                                              LIV05190
C                                                               LIV05200
C---------------------------------------------------------------LIV05210
C                                                               LIV05220
      CALL GENRAN(IIX,G,MEV)                                    LIV05230
C                                                               LIV05240
C---------------------------------------------------------------LIV05250
C                                                               LIV05260
      ITEN = -10                                                LIV05270
      NISOM = NISO + MEV                                        LIV05280
      IWRITO = IWRITE                                           LIV05290
      IWRITE = 0                                                LIV05300
C                                                               LIV05310
      DO 420 J = 1,NDIS                                         LIV05320
      IF(MP(J).NE.ITEN) GO TO 420                               LIV05330
      T0 = VS(J)                                                LIV05340
C                                                               LIV05350
C---------------------------------------------------------------LIV05360
C                                                               LIV05370
      IT = MXINIT                                               LIV05380
      CALL INVERM(ALPHA,BETA,V1,V2,T0,TEMP,T1,EPSM,G,MEV,IT,IWRITE) LIV05390
C                                                               LIV05400
C---------------------------------------------------------------LIV05410
```

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

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

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

```
      3 ' GOOD EIGENVALUE OF A WHERE EVA = (SHIFT-ONE/EVBI)(-ONE/S0)'/   LIV07070
      4 ' MULTOL = T-MULTIPLICITY TOLERANCE FOR T-EIGENVALUES IN BISEC'/ LIV07080
      4 ' TMULT IS THE T-MULTIPLICITY OF GOOD T-EIGENVALUE'/             LIV07090
      5 ' TMULT = -1 MEANS SPURIOUS T-EIGENVALUE TOO CLOSE'/             LIV07100
      6 ' DO NOT COMPUTE ERROR ESTIMATES FOR SUCH EIGENVALUES'/          LIV07110
      7 ' AMINGAP = MINIMAL GAP BETWEEN THE COMPUTED A-EIGENVALUES'/     LIV07120
      8 ' AMINGAP .LT. 0. MEANS MINIMAL GAP IS DUE TO LEFT-HAND GAP'/    LIV07130
      9 ' TMINGAP= MINIMAL GAP W.R.T.  DISTINCT EIGENVALUES IN T(1,MEV)'/LIV07140
      1 ' TMINGAP .LT. 0. MEANS MINGAP IS DUE TO SPURIOUS T-EIGENVALUE'/ LIV07150
      2 ' ----- END OF FILE 3 GOODEIGENVALUES----------------------'///)LIV07160
C                                                                       LIV07170
      IF (IDIST.EQ.1) WRITE(11,640)                                     LIV07180
  640 FORMAT(/' ABOVE ARE THE DISTINCT EIGENVALUES OF T(1,MEV).'/       LIV07190
      2 ' THE FORMAT IS      T-MULTIPLICITY    T-EIGENVALUE   TMINGAP'/  LIV07200
      3 '          THIS FORMAT IS REPEATED TWICE ON EACH LINE.'/         LIV07210
      4 ' T-MULTIPLICITY = -1 MEANS THAT THE SUBROUTINE ISOEV HAS TAGGED'LIV07220
      5/'   THIS SIMPLE T-EIGENVALUE AS HAVING A VERY CLOSE SPURIOUS'/   LIV07230
      6 '   T-EIGENVALUE SO THAT NO ERROR ESTIMATE WILL BE COMPUTED'/    LIV07240
      7 '   FOR THAT EIGENVALUE IN SUBROUTINE INVERR.'/                  LIV07250
      8 ' TMINGAP .LT. 0, TMINGAP IS DUE TO LEFT GAP .GT. 0, RIGHT GAP.'/LIV07260
      9 ' EACH OF THE DISTINCT T-EIGENVALUE TABLES IS FOLLOWED'/         LIV07270
      9 ' BY THE T-MULTIPLICITY PATTERN.'/                               LIV07280
      1 ' NDIS = NUMBER OF COMPUTED DISTINCT EIGENVALUES OF T(1,MEV).'/  LIV07290
      2 ' NG = NUMBER OF GOOD T-EIGENVALUES. '/                          LIV07300
      3 ' NISO = NUMBER OF ISOLATED GOOD T-EIGENVALUES. '/               LIV07310
      4 ' NISO ALSO IS THE COUNT OF +1 ENTRIES IN T-MULTIPLICITY PATTERN.LIV07320
      5 '/' ----- END OF FILE 11 DISTINCT T-EIGENVALUES-------------'///LIV07330
      6)                                                                 LIV07340
C                                                                       LIV07350
      IF(NIOS.NE.0)  WRITE(4,650)                                       LIV07360
  650 FORMAT(/' ABOVE ARE THE ERROR ESTIMATES OBTAINED FOR THE ISOLATED LIV07370
     1GOOD T-EIGENVALUES'/                                              LIV07380
     1' OBTAINED VIA INVERSE ITERATION IN THE SUBROUTINE INVERR.'/      LIV07390
     1' ALL OTHER GOOD T-EIGENVALUES HAVE CONVERGED.'/                  LIV07400
     2' ERROR ESTIMATE = BETAM*ABS(UM)'/                                LIV07410
     2' WHERE BETAM = BETA(MEV+1) AND UM = U(MEV).'/                    LIV07420
     3' U = UNIT EIGENVECTOR OF T WHERE T*U = EV*U AND EV = ISOLATED GOOLIV07430
     3D T-EIGENVALUE.'/                                                 LIV07440
     4' TMINGAP = GAP TO NEAREST DISTINCT EIGENVALUE OF T(1,MEV).'/     LIV07450
     5' TMINGAP .LT. 0. MEANS MINGAP IS DUE TO LEFT NEIGHBOR'/          LIV07460
     6' ERROR ESTIMATE L.T. 0 MEANS INVERSE ITERATION DID NOT CONVERGE'/LIV07470
     7' ------ END OF FILE 4 ERRINV ----------------------------'//)    LIV07480
      GO TO 700                                                         LIV07490
C                                                                       LIV07500
  660 CONTINUE                                                          LIV07510
C                                                                       LIV07520
      IBB = IABS(IBMEV)                                                 LIV07530
      IF (IBMEV.LT.0) WRITE(6,670) MEV,IBB,BETA(IBB)                    LIV07540
  670 FORMAT(/' PROGRAM TERMINATES BECAUSE MEV REQUESTED = ',I6,' IS .GTLIV07550
     1',I6/' AT WHICH AN ABNORMALLY SMALL BETA = ' , E13.4,' OCCURRED')/)LIV07560
      GO TO 700                                                         LIV07570
C                                                                       LIV07580
  680 IF (NDIS.EQ.0.AND.ISTOP.GT.0) WRITE(6,690)                        LIV07590
  690 FORMAT(/' INTERVALS SPECIFIED FOR BISECT DID NOT CONTAIN ANY T-EIGLIV07600
     1ENVALUES'/' PROGRAM TERMINATES')                                  LIV07610
```

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

4.3 LIVEC: Main Program, Eigenvector Computations

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

```
C       THIS LARGE, THEN THE PROGRAM RESETS KMAX TO THIS SIZE        LIV00470
C       AND EXTENDS THE ALPHA, BETA HISTORY IF REQUIRED.             LIV00480
C       THUS, THE DIMENSIONS OF THE ALPHA AND BETA ARRAYS MUST BE    LIV00490
C       LARGE ENOUGH TO ALLOW FOR THIS POSSIBILITY.                  LIV00500
C       REMEMBER THAT THE BETA ARRAY, BETA(J), IS SUCH THAT          LIV00510
C       J = 1,..., KMAX+1,  SO IF THE KMAX USED BY THE PROGRAM       LIV00520
C       IS TO BE 3000, THEN BETA MUST BE OF LENGTH AT LEAST 3001.    LIV00530
C                                                                    LIV00540
C-------------------------------------------------------------------LIV00550
        DOUBLE PRECISION  ALPHA(1000),BETA(1001)                     LIV00560
        DOUBLE PRECISION  V1(2200),V2(2200),VS(2200)                 LIV00570
        DOUBLE PRECISION  RITVEC(40000),TVEC(5000)                   LIV00580
        DOUBLE PRECISION  GOODA(50),GOODBI(50),EVNEW(50),TLAST(50)   LIV00590
        DOUBLE PRECISION  EVAL,EVALN,TOLN,TTOL,ERTOL,ALFA,BATA       LIV00600
        DOUBLE PRECISION  MULTOL,SCALE0,STUTOL,BTOL,LB,UB,S0,RNORME  LIV00610
        DOUBLE PRECISION  ONE,ZERO,MACHEP,EPSM,TEMP,SUM,SHIFT,SHIFT0 LIV00620
        DOUBLE PRECISION  RELTOL,ERROR,TERROR,BKMIN,ERRMIN           LIV00630
        REAL G(5000),AMINGP(50),TMINGP(50),BIERR(50),BIEVER(50),BIERRG(50)LIV00640
        REAL TERR(50),RNORM(50),TBETA(50),BIMING(50)                 LIV00650
        REAL EXPLAN(20)                                              LIV00660
        INTEGER  MP(50),IDELTA(50)                                   LIV00670
        INTEGER  M1(50),M2(50),MA(50),ML(50),MINT(50),MFIN(50)       LIV00680
        INTEGER  SVSEED,SVSOLD,RHSEED                                LIV00690
        INTEGER  MBOUND,NTVCON,SVTVEC,TVSTOP,LVCONT,ERCONT,TFLAG     LIV00700
        DOUBLE PRECISION  FINPRO                                     LIV00710
        DOUBLE PRECISION  DABS, DMAX1, DSQRT, DFLOAT                 LIV00720
        REAL ABS                                                     LIV00730
        INTEGER  IABS                                                LIV00740
        EXTERNAL BSOLV                                               LIV00750
C-------------------------------------------------------------------LIV00760
        DATA MACHEP/Z3410000000000000/                              LIV00770
        EPSM = 2.D0*MACHEP                                           LIV00780
C-------------------------------------------------------------------LIV00790
C       ARRAYS MUST BE DIMENSIONED AS FOLLOWS:                       LIV00800
C       1.  ALPHA:  >= KMAXN,  BETA:  >= (KMAXN+1) WHERE KMAXN, THE  LIV00810
C                   LARGEST SIZE T-MATRIX CONSIDERED BY THE PROGRAM, LIV00820
C                   IS THE LARGER OF THE SIZE OF THE ALPHA, BETA HISTORY LIV00830
C                   PROVIDED ON FILE 2 (IF ANY ) AND THE SIZE WHICH THE LIV00840
C                   PROGRAM SPECIFIES INTERNALLY, THIS LATTER IS ALWAYS LIV00850
C                   < = 11*MEV / 8  +  12, WHERE MEV IS THE SIZE     LIV00860
C                   T-MATRIX THAT WAS USED IN THE CORRESPONDING EIGENVALUE LIV00870
C                   COMPUTATIONS.                                    LIV00880
C       2.  V1:  >= MAX(N,KMAX)                                      LIV00890
C       3.  V2, VS:   >= N                                           LIV00900
C       4.  G:  >= MAX(N,KMAX)                                       LIV00910
C       5.  RITVEC:  >= N*NGOOD, WHERE NGOOD IS NUMBER OF EIGENVALUES LIV00920
C                    SUPPLIED TO THIS PROGRAM.                       LIV00930
C       6.  TVEC:  >= CUMULATIVE LENGTH OF ALL THE T-EIGENVECTORS    LIV00940
C           NEEDED TO GENERATE THE DESIRED RITZ VECTORS.  AN EDUCATED LIV00950
C           GUESS AT AN APPROPRIATE LENGTH CAN BE OBTAINED BY RUNNING THE LIV00960
C           PROGRAM WITH THE FLAG MBOUND = 1 AND MULTIPLYING THE     LIV00970
C           RESULTING SIZE BY 5/4.                                   LIV00980
C       7.  GOODA, GOODBI, EVNEW, AMINGP, TMINGP, TERR, RNORM,       LIV00990
C           TBETA, TLAST, BIERR, BIERRG, MP, MA, M1, M2, MINT,       LIV01000
C           MFIN AND IDELTA MUST BE OF DIMENSION AT LEAST NGOOD.     LIV01010
```

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

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

```
      WRITE(6,40) S0,SHIFT                                     LIV02120
   40 FORMAT(/4X,'SCALE APPLIED TO MATRIX',4X,'SHIFT APPLIED TO MATRIX'/LIV02130
     1E27.4,E27.4)                                             LIV02140
C                                                              LIV02150
      WRITE(6,50) MBOUND,NTVCON,SVTVEC,IREAD                   LIV02160
   50 FORMAT(/3X,'MBOUND',3X,'NTVCON',3X,'SVTVEC',3X,'IREAD'/3I9,I8/) LIV02170
C                                                              LIV02180
      WRITE(6,60) TVSTOP,LVCONT,ERCONT,IWRITE                  LIV02190
   60 FORMAT(3X,'TVSTOP',3X,'LVCONT',3X,'ERCONT',3X,'IWRITE'/4I9) LIV02200
C                                                              LIV02210
      WRITE(6,70) MDIMTV,MDIMRV,MBETA                          LIV02220
   70 FORMAT(/3X,'MDIMTV',3X,'MDIMRV',3X,'MBETA'/2I9,I8)       LIV02230
C                                                              LIV02240
      WRITE(6,80) RELTOL,RHSEED                                LIV02250
   80 FORMAT(/7X,'RELTOL',3X,'RHSEED'/E13.4,I9)                LIV02260
C                                                              LIV02270
C     FROM FILE 3 READ IN THE NUMBER OF EIGENVALUES (NGOOD) FOR WHICH LIV02280
C     EIGENVECTORS ARE REQUESTED, THE ORDER (MEV) OF THE LANCZOS LIV02290
C     TRIDIAGONAL MATRIX USED IN COMPUTING THESE EIGENVALUES, THE LIV02300
C     ORDER (NOLD) OF THE USER-SPECIFIED MATRIX USED IN THE EIGENVALUE LIV02310
C     COMPUTATIONS, THE SEED (SVSEED) USED FOR GENERATING THE STARTING LIV02320
C     VECTOR THAT WAS USED IN THOSE LANCZOS EIGENVALUE COMPUTATIONS, LIV02330
C     AND THE MATRIX/RUN IDENTIFICATION NUMBER (MATOLD) USED IN THOSE LIV02340
C     COMPUTATIONS.  ALSO READ IN THE NUMBER (NDIS) OF DISTINCT LIV02350
C     EIGENVALUES OF T(1,MEV) THAT WERE COMPUTED BUT THIS VALUE IS LIV02360
C     NOT USED IN THE EIGENVECTOR COMPUTATIONS.                LIV02370
C                                                              LIV02380
      READ(3,90) NGOOD,NDIS,MEV,NOLD,SVSEED,MATOLD             LIV02390
   90 FORMAT(4I6,I12,I8)                                       LIV02400
C                                                              LIV02410
C     READ IN THE MULTIPLICITY TOLERANCE USED IN THE BISEC SUBROUTINE LIV02420
C     DURING THE COMPUTATION OF THE GIVEN EIGENVALUES.         LIV02430
C     ALSO READ IN THE FLAG IB.  IF IB < 0, THEN SOME BETA(I) IN THE LIV02440
C     T-MATRIX FILE PROVIDED ON FILE 2 FAILED THE ORTHOGONALITY LIV02450
C     TEST IN THE TNORM SUBROUTINE.  USER SHOULD NOTE THAT THIS VECTOR LIV02460
C     PROGRAM PROCEEDS INDEPENDENTLY OF THE SIZE OF THE BETA USED. LIV02470
C                                                              LIV02480
      READ(3,100) MULTOL,IB,BTOL,SHIFTO                        LIV02490
  100 FORMAT(E20.12,I6,2E10.3)                                 LIV02500
C                                                              LIV02510
      TEMP = DFLOAT(MEV+1000)                                  LIV02520
      TTOL = MULTOL/TEMP                                       LIV02530
C                                                              LIV02540
      WRITE(6,110) MULTOL,TTOL                                 LIV02550
  110 FORMAT(/' T-MULTIPLICITY TOLERANCE USED IN THE EIGENVALUE COMPUTATLIV02560
     1IONS WAS',E13.4/' SCALED MACHINE EPSILON TTOL IS',E13.4) LIV02570
C                                                              LIV02580
C     CONTINUE WRITE TO FILE 6 OF THE PARAMETERS FOR THIS RUN  LIV02590
C                                                              LIV02600
      NG = NGOOD                                               LIV02610
      WRITE(6,120)NG,NDIS,MEV,NOLD,MATOLD,SVSEED,IB,MULTOL,BTOL,SHIFTO LIV02620
  120 FORMAT(/' EIGENVALUES ARE READ IN FROM FILE 3.  THE HEADER IS'/ LIV02630
     1 4X,'NG',2X,'NDIS',3X,'MEV',2X,'NOLD',2X,'MATOLD',6X,'SVSEED'/ LIV02640
     1 4I6,I8,I12/                                             LIV02650
     1 6X,'IB',6X,'MULTOL',8X,'BTOL',6X,'SHIFTO'/              LIV02660
```

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

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

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

```
C                                                                      LIV04320
C----------------------------------------------------------------------LIV04330
C                                                                      LIV04340
      CALL LANCZS(BSOLV,ALPHA,BETA,V1,V2,VS,G,KMAX,MOLD1,N,SVSEED)      LIV04350
C                                                                      LIV04360
C----------------------------------------------------------------------LIV04370
C                                                                      LIV04380
  440 CONTINUE                                                         LIV04390
C                                                                      LIV04400
C     THE SUBROUTINE STURMI DETERMINES THE SMALLEST SIZE T-MATRIX FOR  LIV04410
C     WHICH THE EIGENVALUE IN QUESTION IS AN EIGENVALUE (TO WITHIN A   LIV04420
C     GIVEN TOLERANCE) AND IF POSSIBLE THE SMALLEST SIZE T-MATRIX      LIV04430
C     FOR WHICH IT IS A DOUBLE EIGENVALUE (TO WITHIN THE SAME          LIV04440
C     TOLERANCE).  THE SIZE T-MATRIX USED IN THE EIGENVECTOR           LIV04450
C     COMPUTATIONS IS THEN DETERMINED BY LOOPING ON SIZE OF THE        LIV04460
C     T-EIGENVECTORS, USING THE VALUES FROM STURMI TO DETERMINE        LIV04470
C     FIRST GUESSES AT THE APPROPRIATE T-SIZES.                        LIV04480
C                                                                      LIV04490
C                                                                      LIV04500
      STUTOL = SCALE0*MULTOL                                           LIV04510
      IF(IWRITE.EQ.1) WRITE(6,450)                                     LIV04520
  450 FORMAT(' FROM STURMI')                                           LIV04530
      DO 490 J = 1,NGOOD                                               LIV04540
      EVAL = GOODBI(J)                                                 LIV04550
C     COMPUTE THE TOLERANCES USED BY STURMI TO DETERMINE AN INTERVAL   LIV04560
C     CONTAINING THE EIGENVALUE EVAL.                                  LIV04570
      TEMP = DABS(EVAL)*RELTOL                                         LIV04580
      TOLN = DMAX1(TEMP,STUTOL)                                        LIV04590
C                                                                      LIV04600
C----------------------------------------------------------------------LIV04610
C                                                                      LIV04620
      CALL STURMI(ALPHA,BETA,EVAL,TOLN,EPSM,KMAX,MK1,MK2,IC,IWRITE)    LIV04630
C                                                                      LIV04640
C----------------------------------------------------------------------LIV04650
C                                                                      LIV04660
C     STORE THE COMPUTED ORDERS OF T-MATRICES FOR LATER PRINTOUT       LIV04670
      M1(J) = MK1                                                      LIV04680
      M2(J) = MK2                                                      LIV04690
      ML(J) = (MK1 + 3*MK2)/4                                          LIV04700
      IF(MK2.EQ.KMAX)  ML(J) = KMAX                                    LIV04710
C                                                                      LIV04720
      IF(IC.GT.0) GO TO 470                                            LIV04730
C     IC = 0 MEANS THERE WAS NO T-EIGENVALUE IN THE DESIGNATED INTERVALLIV04740
C     BY T-SIZE KMAX.  THIS MEANS THAT THE EIGENVALUE PROVIDED HAS     LIV04750
C     NOT YET CONVERGED SO ITS EIGENVECTOR SHOULD NOT BE COMPUTED.     LIV04760
      WRITE(6,460) J,GOODBI(J),MK1,MK2                                 LIV04770
  460 FORMAT(I6,'TH EIGENVALUE',E20.12,' HAS NOT CONVERGED '/          LIV04780
     1' SO DO NOT COMPUTE ANY T-EIGENVECTOR OR RITZ VECTOR FOR IT'     LIV04790
     1/' MK1 AND MK2 FOR THIS EIGENVALUE WERE',2I6)                    LIV04800
      MP(J) = MPMIN                                                    LIV04810
      MA(J) = -2*KMAX                                                  LIV04820
      GO TO 490                                                        LIV04830
C     COMPUTE AN APPROPRIATE SIZE T-MATRIX FOR THE GIVEN EIGENVALUE.   LIV04840
  470 IF(M2(J).EQ.KMAX) GO TO 480                                      LIV04850
C     M1 AND M2 WERE BOTH DETERMINED                                   LIV04860
```

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

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

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

```
C     CHECK THE SIZE OF THE ERROR ESTIMATE, ERROR.              LIVO6520
C     IF THIS ESTIMATE IS NOT AS SMALL AS DESIRED AND           LIVO6530
C     |MA(J)| < ML(J), ATTEMPT TO INCREASE THE SIZE OF |MA(J)|  LIVO6540
C     AND REPEAT THE T-EIGENVECTOR COMPUTATIONS.                LIVO6550
C                                                               LIVO6560
      IF(ERROR.LT.ERTOL.OR.TFLAG.EQ.1)  GO TO 740               LIVO6570
C                                                               LIVO6580
      IF(ERROR.GE.ERRMIN) GO TO 650                             LIVO6590
C     LAST COMPONENT IS LESS THAN MINIMAL TO DATE               LIVO6600
      ERRMIN = ERROR                                            LIVO6610
      MABEST = MA(J)                                            LIVO6620
  650 CONTINUE                                                  LIVO6630
C                                                               LIVO6640
      IF(MA(J).GT.0)  ITEST = MA(J) + IDELTA(J)                 LIVO6650
      IF(MA(J).LT.0)  ITEST = -(IABS(MA(J)) + IDELTA(J))        LIVO6660
      IF(IABS(ITEST).LE.ML(J).AND.ICOUNT.LE.10) GO TO 670       LIVO6670
C     NEW MA(J) IS GREATER THAN MAXIMUM ALLOWED.                LIVO6680
      IF(ERCONT.EQ.0.OR.MABEST.EQ.MPMIN) GO TO 690              LIVO6690
      TFLAG = 1                                                 LIVO6700
      MA(J) = MABEST                                            LIVO6710
      IF(ILBIS.EQ.0) MTOL = MTOL - KMAXU                        LIVO6720
      WRITE(6,660) MA(J)                                        LIVO6730
  660 FORMAT(' 10 ORDERS WERE CONSIDERED.  NONE SATISFIED THE ERROR TESTLIVO6740
     1'/' THEREFORE USE THE BEST ORDER OBTAINED FOR THE EIGENVECTORS'   LIVO6750
     1,I6)                                                      LIVO6760
      GO TO 570                                                 LIVO6770
C                                                               LIVO6780
  670 MA(J) = ITEST                                             LIVO6790
C                                                               LIVO6800
      MT = IABS(MA(J))                                          LIVO6810
      IF(IWRITE.EQ.1) WRITE(6,680)  MT                          LIVO6820
  680 FORMAT(/' CHANGE SIZE OF T-MATRIX TO ',I6,' RECOMPUTE T-EIGENVECTOLIVO6830
     1R')                                                       LIVO6840
C                                                               LIVO6850
      IF(ILBIS.EQ.0)  MTOL = MTOL - KMAXU                       LIVO6860
C                                                               LIVO6870
      GO TO 570                                                 LIVO6880
C                                                               LIVO6890
C     APPROPRIATE SIZE T-MATRIX WAS NOT OBTAINED                LIVO6900
  690 CONTINUE                                                  LIVO6910
      WRITE(10,700) J,EVAL,MP(J)                                LIVO6920
  700 FORMAT(/' ON 10 INCREMENTS NOT ABLE TO IDENTIFY APPROPRIATE SIZE  LIVO6930
     1T-MATRIX FOR'/                                            LIVO6940
     1' EIGENVALUE(',I4,') = ',E20.12,' T-MULTIPLICITY =',I4/)  LIVO6950
      IF(M2(J).EQ.KMAX) WRITE(10,710)                           LIVO6960
      IF(M2(J).LT.KMAX) WRITE(10,720)                           LIVO6970
  710 FORMAT(' ORDERS TESTED RANGED FROM 5*M1(J)/4 TO APPROXIMATELY     LIVO6980
     1 MIN(11*MEV/8,13*M1(J)/8)')                               LIVO6990
  720 FORMAT(' ORDERS TESTED RANGED FROM APPROX. (3*M1(J)+M2(J))/4 TO (3LIVO7000
     1*M1(J)+5*M2(J))/8')                                       LIVO7010
      WRITE(10,730)                                             LIVO7020
  730 FORMAT(' ALLOWING LARGER ORDERS FOR THE T-MATRICES MAY RESULT IN  LIVO7030
     1 SUCCESS'/' BUT PROBABLY WILL NOT.  PROBLEM IS PROBABLY DUE TO'    LIVO7040
     1 /' LACK OF CONVERGENCE OF GIVEN EIGENVALUE, CHECK THE ERROR ESTIMLIVO7050
     1ATE')                                                     LIVO7060
```

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

```
C                                                                  LIV07620
      WRITE(11,300) (TVEC(K), K=KI,KF)                             LIV07630
C                                                                  LIV07640
  860 CONTINUE                                                     LIV07650
C                                                                  LIV07660
      IF(TVSTOP.NE.1) GO TO 880                                    LIV07670
C                                                                  LIV07680
      WRITE(6,870) TVSTOP, NTVEC,NGOOD                             LIV07690
  870 FORMAT(/' USER SET TVSTOP = ',I1/                            LIV07700
     1' THEREFORE PROGRAM TERMINATES AFTER T-EIGENVECTOR COMPUTATIONS'/ LIV07710
     1' T-EIGENVECTORS THAT WERE COMPUTED ARE SAVED ON FILE 11'/   LIV07720
     1I8,' T-EIGENVECTORS WERE COMPUTED OUT OF',I7,' REQUESTED'/)  LIV07730
C                                                                  LIV07740
      GO TO 1670                                                   LIV07750
C                                                                  LIV07760
  880 CONTINUE                                                     LIV07770
C     IF NOT ABLE TO COMPUTE ALL THE REQUESTED T-EIGENVECTORS      LIV07780
C     CONTINUE WITH THE LANCZOS VECTOR COMPUTATIONS ANYWAY?        LIV07790
      IF(NTVEC.NE.NGOOD.AND.LVCONT.EQ.0) GO TO 1570                LIV07800
C                                                                  LIV07810
C     COMPUTE THE MAXIMUM SIZE OF THE T-MATRIX USED FOR THOSE      LIV07820
C     EIGENVALUES WITH GOOD ERROR ESTIMATES.                       LIV07830
C                                                                  LIV07840
      KMAXU = 0                                                    LIV07850
      DO 890 J = 1,NGOODC                                          LIV07860
      MT = IABS(MA(J))                                             LIV07870
      IF(MT.LT.KMAXU.OR.MP(J).EQ.MPMIN) GO TO 890                  LIV07880
      KMAXU = MT                                                   LIV07890
  890 CONTINUE                                                     LIV07900
C                                                                  LIV07910
      IF(KMAXU.EQ.0) GO TO 1610                                    LIV07920
C                                                                  LIV07930
      WRITE(6,900) KMAXU                                           LIV07940
  900 FORMAT(/I6,' = LARGEST SIZE T-MATRIX TO BE USED IN THE RITZ VECTOR LIV07950
     1 COMPUTATIONS')                                              LIV07960
C                                                                  LIV07970
C     COUNT THE NUMBER OF RITZ VECTORS NOT BEING COMPUTED          LIV07980
      MREJEC = 0                                                   LIV07990
      DO 910 J=1,NGOODC                                            LIV08000
  910 IF(MP(J).EQ.MPMIN)  MREJEC = MREJEC + 1                      LIV08010
      MREJET = MREJEC + (NGOOD-NGOODC)                             LIV08020
      IF(MREJET.NE.0) WRITE(6,920) MREJET                          LIV08030
  920 FORMAT(/' RITZ VECTORS ARE NOT COMPUTED FOR',I6,' OF THE EIGNEVALU LIV08040
     1ES'/)                                                        LIV08050
      NACT = NGOODC - MREJEC                                       LIV08060
      WRITE(6,930) NGOOD,NTVEC,NACT                                LIV08070
  930 FORMAT(/I6,' RITZ VECTORS WERE REQUESTED'/I6,' T-EIGENVECTORS WERE LIV08080
     1 COMPUTED'/I6,' RITZ VECTORS WILL BE COMPUTED'/)             LIV08090
C     CHECK IF THERE ARE ANY RITZ VECTORS TO COMPUTE               LIV08100
      IF(MREJEC.EQ.NGOODC) GO TO 1590                              LIV08110
C                                                                  LIV08120
C     CONTINUE WITH THE LANCZOS VECTOR COMPUTATIONS?               LIV08130
      IF(LVCONT.EQ.0.AND.MREJEC.NE.0) GO TO 1570                   LIV08140
C                                                                  LIV08150
C     NOW COMPUTE THE RITZ VECTORS.  REGENERATE THE                LIV08160
```

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

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

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

```
C     BIEVER = |(VS-TRANS)*BI*VS - GOODBI(J)|                       LIV09820
C                                                                   LIV09830
C-------------------------------------------------------------------LIV09840
C     V1 = (B-INVERSE)*VS                                           LIV09850
      JBSOLV = 2                                                    LIV09860
      CALL BSOLV(VS,V1,JBSOLV)                                      LIV09870
C-------------------------------------------------------------------LIV09880
C                                                                   LIV09890
      EVALN = EVNEW(J)                                              LIV09900
C                                                                   LIV09910
C-------------------------------------------------------------------LIV09920
      TEMP = FINPRO(N,V1(1),1,VS(1),1)                              LIV09930
C-------------------------------------------------------------------LIV09940
C                                                                   LIV09950
      TEMP = DABS(TEMP - EVALN)                                     LIV09960
      BIEVER(J) = TEMP                                              LIV09970
      DO 1110 K = 1,N                                               LIV09980
 1110 V1(K) = V1(K) - EVALN*VS(K)                                   LIV09990
C                                                                   LIV10000
C-------------------------------------------------------------------LIV10010
      SUM = FINPRO(N,V1(1),1,V1(1),1)                               LIV10020
C-------------------------------------------------------------------LIV10030
C                                                                   LIV10040
      SUM = DSQRT(SUM)                                              LIV10050
      BIERR(J) = SUM                                                LIV10060
      BIERRG(J) = SUM/ABS(BIMING(J))                               LIV10070
C                                                                   LIV10080
      LINT = LFIN - N + 1                                           LIV10090
      EVAL = (ONE/EVALN - SHIFT)/S0                                 LIV10100
      GOODA(J) = EVAL                                               LIV10110
      TEMP = BIEVER(J)                                              LIV10120
C                                                                   LIV10130
      IF(IWRITE.EQ.0)  GO TO 1140                                   LIV10140
      WRITE(6,1120) J,GOODBI(J)                                     LIV10150
 1120 FORMAT(/I5,' TH B-INVERSE EIGENVALUE COMPUTED = ',E20.12/)    LIV10160
C                                                                   LIV10170
      WRITE(6,1130)  TERR(J),TBETA(J),RNORME                        LIV10180
 1130 FORMAT(' NORM OF ERROR IN T-EIGENVECTOR = ',E14.3/            LIV10190
     1' BETA(MA(J)+1)*U(MA(J)) = ',E14.3/                           LIV10200
     1' ABS(NORM(RITVEC) - 1.0) = ',E14.3/)                         LIV10210
C                                                                   LIV10220
 1140 CONTINUE                                                      LIV10230
C                                                                   LIV10240
C     RITZVECTORS ARE NORMALIZED AND ERROR ESTIMATES ARE IN BIERR   LIV10250
C     AND BIERRG ARRAYS. STORE EVERYTHING                           LIV10260
C                                                                   LIV10270
      WRITE(13,1150)                                                LIV10280
 1150 FORMAT(6X,'BIEIGENVALUE',6X,'RITZNORM',7X,'TBETA',7X,'TLAST',5X,LIV10290
     1 'BIERROR',6X,'BIEVER')                                       LIV10300
C                                                                   LIV10310
      WRITE(9,1160)                                                 LIV10320
 1160 FORMAT(5X,'BIEIGENVALUE',4X,'MA(J)',4X,'BIMINGAP',5X,'BIERROR',3X LIV10330
     1 ,'BIERR/GAP',6X,'TERROR')                                    LIV10340
C                                                                   LIV10350
      DO 1190 J=1,NGOODC                                            LIV10360
```

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

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

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

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

4.4   LIMULT: LANCZS and Sample Matrix-Vector Multiply Subroutines

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

        END                                                    LOR02690
```

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

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

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

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

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

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

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

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

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

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

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

4.6 LIVAL: LIVEC: File Definitions, Sample Input Files

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

```
Sample Specification of the Input/Output Files for LIVAL
-------------------------------------------------------------------
 LIVAL EXEC LANCZOS EIGENVALUE CALCULATION USING FACTORIZATION
FI 06 TERM
FILEDEF  1 DISK &1       NHISTORY  A (RECFM F LRECL 80 BLOCK 80
FILEDEF  2 DISK &1       HISTORY   A (RECFM F LRECL 80 BLOCK 80
FILEDEF  3 DISK &1       GOODEV    A (RECFM F LRECL 80 BLOCK 80
FILEDEF  4 DISK &1       ERRINV    A (RECFM F LRECL 80 BLOCK 80
FILEDEF  5 DISK LIVAL    INPUT     A (RECFM F LRECL 80 BLOCK 80
FILEDEF  7 DISK &1       LDATA     A (RECFM F LRECL 80 BLOCK 80
FILEDEF 11 DISK &1       DISTINCT  A (RECFM F LRECL 80 BLOCK 80
LOAD   LIVAL   LESUB   LIMULT
-------------------------------------------------------------------


Sample Input File for LIVAL
----------------------------------------------------------------------
 LIVAL EIGENVALUE COMPUTATION, NO REORTHOGONALIZATION
 USING INVERSE OF REAL SYMMETRIC MATRIX VIA FACTORIZATION
LINE 1    N     KMAX    NMEVS    MATNO     SO     SHIFT
         528    2640       2    721830    1.0       0.
LINE 2   SVSEED    RHSEED      MXINIT    MXSTUR
        49302312  5731029         5     100000
LINE 3   ISTART     ISTOP
             0         1
LINE 4    IHIS      IDIST  IWRITE
             1         0       1
LINE 5   RELTOL (RELATIVE TOLERANCE IN 'COMBINING' GOODEV)
     .0000000001
LINE 6   MB(1)   MB(2)   MB(3)   MB(4)    (ORDERS OF T(1,MEV) )
          100     125
LINE 7   NINT     (NUMBER OF SUB-INTERVALS FOR BISEC)
            1
LINE 8   LB(1)    LB(2)   LB(3)   LB(4)   (INTERVAL LOWER BOUNDS)
           1.0
LINE 9   UB(1)    UB(2)   UB(3)   UB(4)   (INTERVAL UPPER BOUNDS)
           100.0
----------------------------------------------------------------------
```

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

```
Sample Specifications of the Input/Output Files for LIVEC
---------------------------------------------------------------
 LIVEC EXEC, EIGENVECTORS FOR INVERSE OF REAL SYMMETRIC MATRIX
FI 06 TERM
FILEDEF  2 DISK &1        HISTORY   A (RECFM F LRECL 80 BLOCK 80
FILEDEF  3 DISK &1        GOODEV    A (RECFM F LRECL 80 BLOCK 80
FILEDEF  4 DISK &1        ERRINV    A (RECFM F LRECL 80 BLOCK 80
FILEDEF  5 DISK LIVEC     INPUT     A (RECFM F LRECL 80 BLOCK 80
FILEDEF  7 DISK &1        LDATA     A (RECFM F LRECL 80 BLOCK 80
FILEDEF  9 DISK &1        ERREST    A (RECFM F LRECL 80 BLOCK 80
FILEDEF 10 DISK &1        BOUNDS    A (RECFM F LRECL 80 BLOCK 80
FILEDEF 11 DISK &1        TEIGVECS  A (RECFM F LRECL 80 BLOCK 80
FILEDEF 12 DISK &1        RITZVECS  A (RECFM F LRECL 80 BLOCK 80
FILEDEF 13 DISK &1        PAIGE     A (RECFM F LRECL 80 BLOCK 80
LOAD  LIVEC  LESUB  LIMULT
---------------------------------------------------------------
```

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

Chapter 5

Real Symmetric Generalized Problems

5.1   Introduction

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

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

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

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

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

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

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

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

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

We use the following 'generalized' Lanczos recursion. For $i = 1, 2, \ldots m$ and a randomly-generated starting

vector v_1 with $\|v_1\|_B = 1$, generate Lanczos vectors v_i using the following recursion.

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

where

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

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

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

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

Once the eigenvalues have been computed accurately enough, the user can select a subset of the 'converged' eigenvalues for which eigenvectors are to be computed. The main program LGVEC, for computing eigenvectors of the real symmetric generalized problem using a factorization of B, is used to compute the desired eigenvectors.

All of the computations are done in double precision arithmetic. Once the Lanczos matrices have been computed, the remaining computations use the same subroutines which are used in the real symmetric case discussed in Chapter 2. In addition to the programs and subroutines provided here, the user must supply a subroutine USPECA which defines and initializes the A-matrix and a subroutine USPECB which defines and initializes the factors of the B-matrix. A subroutine AMATV which computes matrix-vector multiplies Ax for the A-matrix, and a subroutine BSOLV which solves the system of equations $Bz = v$ must also be supplied. These subroutines must be constructed in such a way as to take advantage of the sparsity (and/or structure) of the two user-supplied matrices A and B and such that they are accurate.

The optional preprocessing programs PERMUT, LORDER, LFACT, and LTEST listed in Chapter 4 can also be used with the codes in this chapter. PERMUT calls the SPARSPAK Library [9] to attempt to identify a reordering or permutation P of the given matrix B for which the sparseness of B is preserved under the factorization of the permuted matrix. LORDER takes a given matrix C and permutation P and computes the sparse format for the permuted matrix, PCP^T . LFACT computes the Cholesky factors of a given positive definite matrix. LTEST performs a very crude check on the numerical condition of the matrix supplied to it, by solving a system of equations with and without iterative refinement, LINPACK [7]. Obviously, if the B-matrix is permuted then the A-matrix must be subjected to the same permutation. These codes assume that the Cholesky factor supplied in the subroutine USPECB

corresponds to the permuted B-matrix and that the AMATV subroutine supplied corresponds to the corresponding permuted A-matrix. Thus, the Lanczos codes compute the eigenvalues and eigenvectors of the permuted problem. The permutation (if any) is then unwrapped in the eigenvector program LGVEC.

5.2   LGVAL: Main Program, Eigenvalue Computations

```
C-----LGVAL   (EIGENVALUES, GENERALIZED SYMMETRIC PROBLEM)---------------LGV00010
C  Authors:  Jane Cullum and Ralph A. Willoughby (Deceased)       LGV00020
C            Los Alamos National Laboratory                       LGV00030
C            Los Alamos, New Mexico 87544                         LGV00040
C                                                                 LGV00050
C            E-mail:  cullumj@lanl.gov                            LGV00060
C                                                                 LGV00070
C These codes are copyrighted by the authors.  These codes       LGV00080
C and modifications of them or portions of them are NOT to be     LGV00090
C incorporated into any commercial codes or used for any other    LGV00100
C commercial purposes such as consulting for other companies,     LGV00110
C without legal agreements with the authors of these Codes.       LGV00120
C If these Codes or portions of them are used in other scientific or LGV00130
C engineering research works the names of the authors of these codes LGV00140
C and appropriate references to their written work are to be      LGV00150
C incorporated in the derivative works.                          LGV00160
C                                                                 LGV00170
C This header is not to be removed from these codes.             LGV00180
C                                                                 LGV00190
C        REFERENCE: Cullum and Willoughby, Chapters 1,2,3,4      LGV00191
C        Lanczos Algorithms for Large Symmetric Eigenvalue ComputationsLGV00192
C        VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in LGV00193
C        Applied Mathematics, 2002. SIAM Publications,           LGV00194
C        Philadelphia, PA. USA                                   LGV00195
C                                                                 LGV00196
C                                                                 LGV00200
C    CONTAINS MAIN PROGRAM FOR COMPUTING DISTINCT EIGENVALUES OF  LGV00210
C    A*X = EVAL*B*X WHERE A AND B ARE REAL SYMMETRIC MATRICES,    LGV00220
C    B IS POSITIVE DEFINITE, AND THE CHOLESKY FACTORS OF B        LGV00230
C    ARE AVAILABLE FOR USE IN THE PROCEDURE.  PROCEDURE USES      LGV00240
C    GENERALIZATION OF LANCZOS TRIDIAGONALIZATION WITHOUT ANY     LGV00250
C    REORTHGONALIZATION.                                         LGV00260
C                                                                 LGV00270
C    PFORT VERIFIER IDENTIFIED THE FOLLOWING NONPORTABLE         LGV00280
C    CONSTRUCTIONS                                               LGV00290
C                                                                 LGV00300
C    1.  DATA/MACHEP/ STATEMENT                                  LGV00310
C    2.  ALL READ(5,*) STATEMENTS (FREE FORMAT)                  LGV00320
C    3.  FORMAT(20A4) USED WITH EXPLANATORY HEADER EXPLAN.       LGV00330
C    4.  HEXADECIMAL FORMAT (4Z20) USED IN ALPHA/BETA FILES 1 AND 2. LGV00340
C                                                                 LGV00350
C----------------------------------------------------------------LGV00360
C                                                                 LGV00370
      DOUBLE PRECISION   ALPHA(5000),BETA(5001)                   LGV00380
      DOUBLE PRECISION   V1(5000),V2(5000),VS(5000)              LGV00390
      DOUBLE PRECISION   LB(20),UB(20)                            LGV00400
      DOUBLE PRECISION   BTOL,GAPTOL,TTOL,MACHEP,EPSM,RELTOL      LGV00410
      DOUBLE PRECISION   SCALE1,SCALE2,SCALE3,SCALE4,BISTOL,CONTOL,MULTOLLGV00420
      DOUBLE PRECISION   ONE,ZERO,TEMP,TKMAX,BETAM,BKMIN,T0,T1    LGV00430
      REAL  G(5000),EXPLAN(20)                                    LGV00440
      INTEGER  MP(5000),NMEV(20)                                  LGV00450
```

```
      INTEGER  SVSEED,RHSEED,SVSOLD                              LGV00460
      INTEGER IABS                                               LGV00470
      REAL  ABS                                                  LGV00480
      DOUBLE PRECISION  DABS, DSQRT, DFLOAT                       LGV00490
      EXTERNAL LSOLV, AMATV                                      LGV00500
C                                                                LGV00510
C------------------------------------------------------------------LGV00520
      DATA MACHEP/Z3410000000000000/                             LGV00530
      EPSM = 2.0D0*MACHEP                                        LGV00540
C------------------------------------------------------------------LGV00550
C                                                                LGV00560
C     ARRAYS MUST BE DIMENSIONED AS FOLLOWS:                     LGV00570
C     DIMENSION OF V2 ASSUMES THAT NO MORE THAN KMAX/2 EIGENVALUES LGV00580
C     OF THE LANCZOS T-MATRICES ARE BEING COMPUTED IN ANY ONE OF THE LGV00590
C     SUB-INTERVALS BEING CONSIDERED.  V2 CONTAINS THE UPPER AND LOWER LGV00600
C     BOUNDS FOR EACH T-EIGENVALUE BEING COMPUTED BY BISEC IN ANY ONE LGV00610
C     GIVEN INTERVAL.                                            LGV00620
C                                                                LGV00630
C     1.   ALPHA: >= KMAX,   BETA: >= (KMAX+1)                   LGV00640
C     2.   V1:  >= MAX(N,KMAX+1)                                 LGV00650
C     3.   V2,VS:  >= MAX(N,KMAX)                                LGV00660
C     4.   G:  >= MAX(N,2*KMAX)                                  LGV00670
C     5.   MP:   >= KMAX                                         LGV00680
C     6.   LB,UB:  >= NUMBER OF SUBINTERVALS SUPPLIED TO BISEC.  LGV00690
C     7.   NMEV:  >= NUMBER OF T-MATRICES ALLOWED.               LGV00700
C     8.   EXPLAN:  DIMENSION IS 20.                             LGV00710
C                                                                LGV00720
C                                                                LGV00730
C     IMPORTANT TOLERANCES OR SCALES THAT ARE USED REPEATEDLY     LGV00740
C     THROUGHOUT THE PROGRAM ARE THE FOLLOWING:                  LGV00750
C     SCALED MACHINE EPSILON:  TTOL = TKMAX*EPSM WHERE           LGV00760
C     EPSM = 2*MACHINE EPSILON AND                               LGV00770
C     TKMAX = MAX(|ALPHA(J)|,BETA(J), J = 1,MEV)                 LGV00780
C     BISEC CONVERGENCE TOLERANCE:  BISTOL = DSQRT(1000+MEV)*TTOL LGV00790
C     BISEC T-MULTIPLICITY TOLERANCE:  MULTOL = (1000+MEV)*TTOL  LGV00800
C     LANCZOS CONVERGENCE TOLERANCE:   CONTOL = BETA(MEV+1)*1.D-10 LGV00810
C------------------------------------------------------------------LGV00820
C     OUTPUT HEADER                                              LGV00830
      WRITE(6,10)                                                LGV00840
   10 FORMAT(/' LANCZOS EIGENVALUE PROCEDURE FOR REAL SYMMETRIC GENERALILGV00850
     1ZED PROBLEMS,'/' A*X = EVAL*B*X, B POSITIVE DEFINITE WITH CHOLESKYLGV00860
     1 FACTORS AVAILABLE'/)                                      LGV00870
C                                                                LGV00880
C     SET PROGRAM PARAMETERS                                     LGV00890
C     SCALEK ARE USED IN TOLERANCES NEEDED IN SUBROUTINES LUMP,  LGV00900
C     ISOEV AND PRTEST.  USER MUST NOT MODIFY THEM.              LGV00910
      SCALE1 = 5.0D2                                             LGV00920
      SCALE2 = 5.0D0                                             LGV00930
      SCALE3 = 5.0D0                                             LGV00940
      SCALE4 = 1.0D4                                             LGV00950
      ONE  = 1.0D0                                               LGV00960
      ZERO = 0.0D0                                               LGV00970
      BTOL = 1.0D-8                                              LGV00980
C     BTOL = EPSM                                                LGV00990
      GAPTOL = 1.0D-8                                            LGV01000
```

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

```
C                                                              LGV01560
C     READ IN THE SIZES OF THE T-MATRICES TO BE CONSIDERED.    LGV01570
      READ(5,20) EXPLAN                                        LGV01580
      READ(5,*) (NMEV(J), J=1,NMEVS)                           LGV01590
C                                                              LGV01600
C     READ IN THE NUMBER OF SUBINTERVALS TO BE CONSIDERED.     LGV01610
      READ(5,20) EXPLAN                                        LGV01620
      READ(5,*) NINT                                           LGV01630
C                                                              LGV01640
C     READ IN THE LEFT-END POINTS OF THE SUBINTERVALS TO BE CONSIDERED. LGV01650
C     THESE MUST BE IN ALGEBRAICALLY-INCREASING ORDER          LGV01660
      READ(5,20) EXPLAN                                        LGV01670
      READ(5,*) (LB(J), J=1,NINT)                              LGV01680
C                                                              LGV01690
C     READ IN THE RIGHT-END POINTS OF THE SUBINTERVALS TO BE CONSIDERED.LGV01700
C     THESE MUST BE IN ALGEBRAICALLY-INCREASING ORDER          LGV01710
      READ(5,20) EXPLAN                                        LGV01720
      READ(5,*) (UB(J), J=1,NINT)                              LGV01730
C                                                              LGV01740
C----------------------------------------------------------------LGV01750
C     INITIALIZE THE ARRAYS FOR THE USER-SPECIFIED MATRICES     LGV01760
C     AND PASS THE STORAGE LOCATIONS OF THESE ARRAYS TO THE     LGV01770
C     MATRIX-VECTOR MULTIPLY SUBROUTINE AMATV AND THE SOLVE     LGV01780
C     SUBROUTINE LSOLV.                                        LGV01790
C                                                              LGV01800
      CALL USPECA(N,MATNOA)                                    LGV01810
      CALL USPECB(N,MATNOB)                                    LGV01820
C                                                              LGV01830
C----------------------------------------------------------------LGV01840
C                                                              LGV01850
C     MASK UNDERFLOW AND OVERFLOW                              LGV01860
      CALL MASK                                                LGV01870
C                                                              LGV01880
C----------------------------------------------------------------LGV01890
C                                                              LGV01900
C     WRITE TO FILE 6, A SUMMARY OF THE PARAMETERS FOR THIS RUN LGV01910
C                                                              LGV01920
      WRITE(6,30) MATNOA,MATNOB,N,KMAX                         LGV01930
   30 FORMAT(/3X,'A-MATRIX ID',3X,'B-MATRIX ID',4X,'ORDER OF A',4X, LGV01940
     1'MAX ORDER OF T'/I14,I14,I14,I18/)                       LGV01950
C                                                              LGV01960
      WRITE(6,40) ISTART,ISTOP                                 LGV01970
   40 FORMAT(/2X,'ISTART',3X,'ISTOP'/2I8/)                     LGV01980
C                                                              LGV01990
      WRITE(6,50) IHIS,IDIST,IWRITE                            LGV02000
   50 FORMAT(/4X,'IHIS',3X,'IDIST',2X,'IWRITE'/3I8/)           LGV02010
C                                                              LGV02020
      WRITE(6,60) SVSEED,RHSEED                                LGV02030
   60 FORMAT(/' SEEDS FOR RANDOM NUMBER GENERATOR'//           LGV02040
     1 4X,'LANCZS SEED',4X,'INVERR SEED'/2I15/)                LGV02050
C                                                              LGV02060
      WRITE(6,70) (NMEV(J), J=1,NMEVS)                         LGV02070
   70 FORMAT(/' SIZES OF T-MATRICES TO BE CONSIDERED'/(6I12))  LGV02080
C                                                              LGV02090
      WRITE(6,80) RELTOL,GAPTOL,BTOL                           LGV02100
```

```
   80 FORMAT(/' RELATIVE TOLERANCE USED TO COMBINE COMPUTED T-EIGENVALUELGV02110
      1S'/E15.3/' RELATIVE GAP TOLERANCES USED IN INVERSE ITERATION'/    LGV02120
      1E15.3/' RELATIVE TOLERANCE FOR CHECK ON SIZE OF BETAS'/E15.3/)    LGV02130
C                                                                        LGV02140
      WRITE(6,90) (J,LB(J),UB(J), J=1,NINT)                              LGV02150
   90 FORMAT(/' BISEC WILL BE USED ON THE FOLLOWING INTERVALS'/          LGV02160
      1 (I6,2E20.6))                                                     LGV02170
C                                                                        LGV02180
      IF (ISTART.EQ.0) GO TO 140                                         LGV02190
C                                                                        LGV02200
C     READ IN ALPHA BETA HISTORY                                         LGV02210
C                                                                        LGV02220
      READ(2,100)MOLD,NOLD,SVSOLD,MATAO,MATBO                            LGV02230
  100 FORMAT(2I6,I12,2I8)                                                LGV02240
C                                                                        LGV02250
      IF (KMAX.LT.MOLD) KMAX = MOLD                                      LGV02260
      KMAX1 = KMAX + 1                                                   LGV02270
C                                                                        LGV02280
C     CHECK THAT ORDER N, MATRIX IDS (MATNOA AND MATNOB), AND RANDOM     LGV02290
C     SEED (SVSEED) AGREE WITH THOSE IN THE HISTORY FILE.  IF NOT        LGV02300
C     PROCEDURE STOPS.                                                   LGV02310
C                                                                        LGV02320
      ITEMP = (NOLD-N)**2 + (MATNOA-MATAO)**2 + (SVSEED-SVSOLD)**2       LGV02330
      1 + (MATNOB-MATBO)**2                                              LGV02340
C                                                                        LGV02350
      IF (ITEMP.EQ.0) GO TO 120                                          LGV02360
C                                                                        LGV02370
      WRITE(6,110)                                                       LGV02380
  110 FORMAT(' PROGRAM TERMINATES'/    ' READ FROM FILE 2 CORRESPONDS TOLGV02390
      1 DIFFERENT MATRIX THAN MATRIX SPECIFIED'/)                        LGV02400
      GO TO 640                                                          LGV02410
C                                                                        LGV02420
  120 CONTINUE                                                           LGV02430
      MOLD1 = MOLD+1                                                     LGV02440
C                                                                        LGV02450
      READ(2,130)(ALPHA(J), J=1,MOLD)                                    LGV02460
      READ(2,130)(BETA(J), J=1,MOLD1)                                    LGV02470
  130 FORMAT(4Z20)                                                       LGV02480
C                                                                        LGV02490
      IF (KMAX.EQ.MOLD) GO TO 160                                        LGV02500
C                                                                        LGV02510
C     SAVE V1 = B*V(KMAX), VS = B*V(KMAX+1), V2 = V(KMAX+1)              LGV02520
      READ(2,130) (V1(J), J=1,N)                                         LGV02530
      READ(2,130) (VS(J), J=1,N)                                         LGV02540
      READ(2,130) (V2(J), J=1,N)                                         LGV02550
C                                                                        LGV02560
  140 CONTINUE                                                           LGV02570
      IIX = SVSEED                                                       LGV02580
C                                                                        LGV02590
C-----------------------------------------------------------------------LGV02600
C                                                                        LGV02610
      CALL LANCZS(LSOLV,AMATV,ALPHA,BETA,V1,V2,VS,G,KMAX,MOLD1,N,IIX)    LGV02620
C                                                                        LGV02630
C-----------------------------------------------------------------------LGV02640
C                                                                        LGV02650
```

```
      KMAX1 = KMAX + 1                                          LGV02660
C                                                               LGV02670
      IF (IHIS.EQ.0.AND.ISTOP.GT.0) GO TO 160                   LGV02680
C                                                               LGV02690
      WRITE(1,150) KMAX,N,SVSEED,MATNOA,MATNOB                  LGV02700
  150 FORMAT(2I6,I12,2I8,' = KMAX,N,SVSEED,MATNOA,MATNOB')      LGV02710
C                                                               LGV02720
      WRITE(1,130)(ALPHA(I), I=1,KMAX)                         LGV02730
      WRITE(1,130)(BETA(I), I=1,KMAX1)                         LGV02740
C                                                               LGV02750
C     SAVE V1 = B*V(KMAX), VS = B*V(KMAX+1), V2 = V(KMAX+1)     LGV02760
      WRITE(1,130) (V1(I), I=1,N)                              LGV02770
      WRITE(1,130) (VS(I), I=1,N)                              LGV02780
      WRITE(1,130) (V2(I), I=1,N)                              LGV02790
C                                                               LGV02800
      IF (ISTOP.EQ.0) GO TO 540                                 LGV02810
C                                                               LGV02820
  160 CONTINUE                                                  LGV02830
      BKMIN = BTOL                                              LGV02840
      WRITE(6,170)                                              LGV02850
  170 FORMAT(/' T-MATRICES (ALPHA AND BETA) ARE NOW AVAILABLE'/) LGV02860
C                                                               LGV02870
C-----------------------------------------------------------------LGV02880
C     SUBROUTINE TNORM CHECKS MIN(BETA)/(ESTIMATED NORM(A)) > BTOL . LGV02890
C     IF THIS IS VIOLATED IB IS SET EQUAL TO THE NEGATIVE OF THE INDEX LGV02900
C     OF THE MINIMAL BETA.  IF(IB < 0) THEN SUBROUTINE TNORM IS  LGV02910
C     CALLED FOR EACH VALUE OF MEV TO DETERMINE WHETHER OR NOT THERE LGV02920
C     IS A BETA IN THE T-MATRIX SPECIFIED THAT VIOLATES THIS TEST. LGV02930
C     IF THERE IS SUCH A BETA THE PROGRAM TERMINATES FOR THE USER LGV02940
C     TO DECIDE WHAT TO DO.  THIS TEST CAN BE OVER-RIDDEN BY     LGV02950
C     SIMPLY MAKING BTOL SMALLER, BUT THEN THERE IS THE POSSIBILITY LGV02960
C     THAT LOSSES IN THE LOCAL ORTHOGONALITY MAY HURT THE COMPUTATIONS. LGV02970
C     BTOL = 1.D-8 IS HOWEVER A CONSERVATIVE CHOICE FOR BTOL.    LGV02980
C                                                               LGV02990
C     TNORM ALSO COMPUTES TKMAX = MAX(|ALPHA(K)|,BETA(K), K=1,KMAX). LGV03000
C     TKMAX IS USED TO SCALE THE TOLERANCES USED IN THE         LGV03010
C     T-MULTIPLICITY AND SPURIOUS TESTS IN BISEC. TKMAX IS ALSO USED IN LGV03020
C     THE PROJECTION TEST FOR HIDDEN EIGENVALUES THAT HAD 'TOO SMALL' LGV03030
C     A PROJECTION ON THE STARTING VECTOR.                      LGV03040
C                                                               LGV03050
      CALL TNORM(ALPHA,BETA,BKMIN,TKMAX,KMAX,IB)               LGV03060
C                                                               LGV03070
C-----------------------------------------------------------------LGV03080
C                                                               LGV03090
      TTOL = EPSM*TKMAX                                         LGV03100
C                                                               LGV03110
C     LOOP ON THE SIZE OF THE T-MATRIX                          LGV03120
C                                                               LGV03130
  180 CONTINUE                                                  LGV03140
      MMB = MMB + 1                                             LGV03150
      MEV = NMEV(MMB)                                           LGV03160
C     IS MEV TOO LARGE ?                                        LGV03170
      IF(MEV.LE.KMAX) GO TO 200                                 LGV03180
      WRITE(6,190) MMB, MEV, KMAX                              LGV03190
  190 FORMAT(/' TERMINATE PRIOR TO CONSIDERING THE',I6,'TH T-MATRIX'/ LGV03200
```

```
      1' BECAUSE THE SIZE REQUESTED',I6,' IS GREATER THAN THE MAXIMUM SIZLGV03210
      1E ALLOWED',I6/)                                              LGV03220
         GO TO 540                                                  LGV03230
C                                                                   LGV03240
  200 MP1 = MEV + 1                                                 LGV03250
      BETAM = BETA(MP1)                                             LGV03260
C                                                                   LGV03270
      IF (IB.GE.0) GO TO 210                                        LGV03280
C                                                                   LGV03290
      T0 = BTOL                                                     LGV03300
C                                                                   LGV03310
C-------------------------------------------------------------------LGV03320
C                                                                   LGV03330
      CALL TNORM(ALPHA,BETA,T0,T1,MEV,IBMEV)                        LGV03340
C                                                                   LGV03350
C-------------------------------------------------------------------LGV03360
C                                                                   LGV03370
      TEMP = T0/TKMAX                                               LGV03380
      IBMEV = IABS(IBMEV)                                           LGV03390
      IF (TEMP.GE.BTOL) GO TO 210                                   LGV03400
      IBMEV = -IBMEV                                                LGV03410
      GO TO 600                                                     LGV03420
C                                                                   LGV03430
  210 CONTINUE                                                      LGV03440
      IC = MXSTUR-ICT                                               LGV03450
C                                                                   LGV03460
C-------------------------------------------------------------------LGV03470
C     BISEC LOOP. THE SUBROUTINE BISEC INCORPORATES DIRECTLY THE    LGV03480
C     T-MULTIPLICITY AND SPURIOUS TESTS. T-EIGENVALUES WILL BE      LGV03490
C     CALCULATED BY BISEC SEQUENTIALLY ON INTERVALS                 LGV03500
C     (LB(J),UB(J)), J = 1,NINT).                                   LGV03510
C                                                                   LGV03520
C     ON RETURN FROM BISEC                                          LGV03530
C     NDIS = NUMBER OF DISTINCT EIGENVALUES OF T(1,MEV) ON UNION    LGV03540
C            OF THE (LB,UB) INTERVALS                               LGV03550
C     VS = DISTINCT T-EIGENVALUES IN ALGEBRAICALLY INCREASING ORDER LGV03560
C     MP = MULTIPLICITIES OF THE T-EIGENVALUES IN VS               LGV03570
C     MP(I) = (0,1,MI), MI>1, I=1,NDIS  MEANS:                      LGV03580
C        (0)  VS(I) IS SPURIOUS                                     LGV03590
C        (1)  VS(I) IS T-SIMPLE AND GOOD                            LGV03600
C        (MI) VS(I) IS MULTIPLE AND IS THEREFORE NOT ONLY GOOD BUT  LGV03610
C             ALSO A CONVERGED GOOD T-EIGENVALUE.                   LGV03620
C                                                                   LGV03630
C                                                                   LGV03640
      CALL BISEC(ALPHA,BETA,V1,V2,VS,LB,UB,EPSM,TTOL,MP,NINT,       LGV03650
     1 MEV,NDIS,IC,IWRITE)                                          LGV03660
C                                                                   LGV03670
C-------------------------------------------------------------------LGV03680
C                                                                   LGV03690
      IF (NDIS.EQ.0) GO TO 620                                      LGV03700
C                                                                   LGV03710
C     COMPUTE THE TOTAL NUMBER OF STURM SEQUENCES USED TO DATE      LGV03720
C     COMPUTE THE BISEC CONVERGENCE AND T-MULTIPLICITY TOLERANCES USED. LGV03730
C     COMPUTE THE CONVERGENCE TOLERANCE FOR EIGENVALUES OF A.       LGV03740
      ICT = ICT + IC                                                LGV03750
```

```
      TEMP = DFLOAT(MEV+1000)                                    LGV03760
      MULTOL = TEMP*TTOL                                         LGV03770
      TEMP = DSQRT(TEMP)                                         LGV03780
      BISTOL = TTOL*TEMP                                         LGV03790
      CONTOL = BETAM*1.D-10                                      LGV03800
C                                                                LGV03810
C------------------------------------------------------------LGV03820
C    SUBROUTINE LUMP 'COMBINES' T-EIGENVALUES THAT ARE 'TOO CLOSE'. LGV03830
C    NOTE HOWEVER THAT CLOSE SPURIOUS T-EIGENVALUES ARE NOT AVERAGED LGV03840
C    WITH GOOD ONES. HOWEVER, THEY MAY BE USED TO INCREASE THE   LGV03850
C    MULTIPLICITY OF A GOOD T-EIGENVALUE.                        LGV03860
C                                                                LGV03870
      LOOP = NDIS                                                LGV03880
      CALL LUMP(VS,RELTOL,MULTOL,SCALE2,MP,LOOP)                 LGV03890
C                                                                LGV03900
C------------------------------------------------------------LGV03910
C                                                                LGV03920
      IF(NDIS.EQ.LOOP) GO TO 230                                 LGV03930
C                                                                LGV03940
      WRITE(6,220) NDIS, MEV, LOOP                               LGV03950
  220 FORMAT(/I6,' DISTINCT T-EIGENVALUES WERE COMPUTED IN BISEC AT MEV LGV03960
     1',I6/ 2X,' LUMP SUBROUTINE REDUCES NUMBER OF DISTINCT EIGENVALUES LGV03970
     10',I6)                                                     LGV03980
C                                                                LGV03990
  230 CONTINUE                                                   LGV04000
      NDIS = LOOP                                                LGV04010
      BETA(MP1) = BETAM                                          LGV04020
C                                                                LGV04030
C------------------------------------------------------------LGV04040
C    THE SUBROUTINE ISOEV LABELS THOSE SIMPLE EIGENVALUES OF T(1,MEV) LGV04050
C    WITH VERY SMALL GAPS BETWEEN NEIGHBORING EIGENVALUES OF T(1,MEV) LGV04060
C    TO AVOID COMPUTING ERROR ESTIMATES FOR ANY SIMPLE GOOD     LGV04070
C    T-EIGENVALUE THAT IS TOO CLOSE TO A SPURIOUS EIGENVALUE.   LGV04080
C    ON RETURN FROM ISOEV, G CONTAINS CODED MINIMAL GAPS        LGV04090
C    BETWEEN THE DISTINCT EIGENVALUES OF T(1,MEV). (G IS REAL). LGV04100
C    G(I) < 0 MEANS MINGAP IS DUE TO LEFT GAP G(I) > 0 MEANS DUE TO LGV04110
C    RIGHT GAP. MP(I) = -1 MEANS THAT THE GOOD T-EIGENVALUE IS SIMPLE LGV04120
C    AND HAS A VERY SMALL MINGAP IN T(1,MEV) DUE TO A SPURIOUS  LGV04130
C    EIGENVALUE.  NG = NUMBER OF GOOD T-EIGENVALUES.            LGV04140
C    NISO = NUMBER OF ISOLATED GOOD T-EIGENVALUES.              LGV04150
C                                                                LGV04160
      CALL ISOEV(VS,GAPTOL,MULTOL,SCALE1,G,MP,NDIS,NG,NISO)      LGV04170
C                                                                LGV04180
C------------------------------------------------------------LGV04190
C                                                                LGV04200
      WRITE(6,240)NG,NISO,NDIS                                   LGV04210
  240 FORMAT(/I6,' GOOD T-EIGENVALUES HAVE BEEN COMPUTED'/       LGV04220
     1 I6,' OF THESE ARE T-ISOLATED'/                            LGV04230
     2 I6,' = NUMBER OF DISTINCT T-EIGENVALUES COMPUTED'/)       LGV04240
C                                                                LGV04250
C    DO WE WRITE DISTINCT EIGENVALUES OF T-MATRIX TO FILE 11?    LGV04260
      IF (IDIST.EQ.0) GO TO 280                                  LGV04270
C                                                                LGV04280
      WRITE(11,250) NDIS,NISO,MEV,N,SVSEED,MATNOA,MATNOB         LGV04290
  250 FORMAT(/4I6,I12,I8,'=ND,NIS,MEV,N,SEED,MNA,MNB'/)          LGV04300
```

```
C                                                                    LGV04310
      WRITE(11,260) (MP(I),VS(I),G(I), I=1,NDIS)                     LGV04320
  260 FORMAT(2(I3,E25.16,E12.3))                                     LGV04330
C                                                                    LGV04340
      WRITE(11,270) NDIS, (MP(I), I=1,NDIS)                          LGV04350
  270 FORMAT(/I6,' = NDIS, T-MULTIPLICITIES (0 MEANS  SPURIOUS)'/(20I4))LGV04360
C                                                                    LGV04370
  280 CONTINUE                                                       LGV04380
C                                                                    LGV04390
      IF (NISO.NE.0) GO TO 310                                       LGV04400
C                                                                    LGV04410
      WRITE(4,290) MEV                                               LGV04420
  290 FORMAT(/' AT MEV = ',I6,' THERE ARE NO ISOLATED T-EIGENVALUES'/  LGV04430
     1' SO NO ERROR ESTIMATES WERE COMPUTED/')                      LGV04440
C                                                                    LGV04450
      WRITE(6,300)                                                   LGV04460
  300 FORMAT(/' ALL COMPUTED GOOD T-EIGENVALUES ARE MULTIPLE'/      LGV04470
     1 ' THEREFORE ALL SUCH EIGENVALUES ARE ASSUMED TO HAVE CONVERGED') LGV04480
C                                                                    LGV04490
      ICONV = 1                                                      LGV04500
      GO TO 350                                                      LGV04510
C                                                                    LGV04520
  310 CONTINUE                                                       LGV04530
C                                                                    LGV04540
C--------------------------------------------------------------------LGV04550
C     SUBROUTINE INVERR COMPUTES ERROR ESTIMATES FOR ISOLATED GOOD  LGV04560
C     T-EIGENVALUES USING INVERSE ITERATION ON T(1,MEV). ON RETURN  LGV04570
C     G(J) = MINIMUM GAP IN T(1,MEV) FOR EACH VS(J), J=1,NDIS       LGV04580
C     G(MEV+I) = BETAM*|U(MEV)| = ERROR ESTIMATE FOR ISOLATED GOOD  LGV04590
C             T-EIGENVALUES, WHERE I = 1, NISO AND  BETAM = BETA(MEV+1)LGV04600
C             U(MEV) IS MEVTH COMPONENT OF THE UNIT EIGENVECTOR OF T  LGV04610
C             CORRESPONDING TO THE ITH ISOLATED GOOD T-EIGENVALUE.  LGV04620
C     A NEGATIVE ERROR ESTIMATE MEANS THAT FOR THAT PARTICULAR      LGV04630
C     EIGENVALUE THE INVERSE ITERATION DID NOT CONVERGE IN <= MXINIT  LGV04640
C     STEPS AND THAT THE CORRESPONDING ERROR ESTIMATE IS QUESTIONABLE.  LGV04650
C                                                                    LGV04660
C     V2 CONTAINS THE ISOLATED GOOD T-EIGENVALUES                   LGV04670
C     V1 CONTAINS THE MINGAPS TO THE NEAREST DISTINCT  EIGENVALUE   LGV04680
C        OF T(1,MEV) FOR EACH ISOLATED GOOD T-EIGENVALUE IN V2.     LGV04690
C     VS CONTAINS THE NDIS DISTINCT EIGENVALUES OF T(1,MEV)         LGV04700
C     MP CONTAINS THE CORRESPONDING CODED T-MULTIPLICITIES          LGV04710
C                                                                    LGV04720
      IT = MXINIT                                                    LGV04730
      CALL INVERR(ALPHA,BETA,V1,V2,VS,EPSM,G,MP,MEV,MMB,NDIS,NISO,N,  LGV04740
     1 RHSEED,IT,IWRITE)                                            LGV04750
C                                                                    LGV04760
C--------------------------------------------------------------------LGV04770
C                                                                    LGV04780
C     SIMPLE CHECK FOR CONVERGENCE. CHECKS TO SEE IF ALL OF THE ERROR  LGV04790
C     ESTIMATES ARE SMALLER THAN CONTOL = BETAM*1.D-10.             LGV04800
C     IF THIS TEST IS SATISFIED, THEN CONVERGENCE FLAG, ICONV IS SET  LGV04810
C     TO 1.  TYPICALLY ERROR ESTIMATES ARE VERY CONSERVATIVE.       LGV04820
C                                                                    LGV04830
      WRITE(6,320) CONTOL                                            LGV04840
  320 FORMAT(/' CONVERGENCE IS TESTED USING THE CONVERGENCE TOLERANCE', LGV04850
```

```
      1E13.4/)                                                   LGV04860
C                                                                LGV04870
      II = MEV +1                                                LGV04880
      IF = MEV+NISO                                              LGV04890
      DO 330 I = II,IF                                           LGV04900
      IF (ABS(G(I)).GT.CONTOL) GO TO 350                         LGV04910
  330 CONTINUE                                                   LGV04920
      ICONV = 1                                                  LGV04930
      MMB = NMEVS                                                LGV04940
C                                                                LGV04950
      WRITE(6,340) CONTOL                                        LGV04960
  340 FORMAT(' ALL COMPUTED ERROR ESTIMATES WERE LESS THAN',E15.4/  LGV04970
     1 ' THEREFORE PROCEDURE TERMINATES'/)                       LGV04980
C                                                                LGV04990
  350 CONTINUE                                                   LGV05000
C                                                                LGV05010
C     IF CONVERGENCE IS INDICATED, THAT IS ICONV = 1 ,THEN       LGV05020
C     THE SUBROUTINE PRTEST IS CALLED TO CHECK FOR ANY CONVERGED  LGV05030
C     T-EIGENVALUES THAT HAVE BEEN MISLABELLED AS SPURIOUS BECAUSE LGV05040
C     THE PROJECTION OF THEIR EIGENVECTOR(S) ON THE STARTING      LGV05050
C     VECTOR WERE TOO SMALL.                                     LGV05060
C     NUMERICAL TESTS INDICATE THAT SUCH EIGENVALUES ARE RARE.   LGV05070
C     IF FOR SOME REASON MANY OF THESE HIDDEN EIGENVALUES APPEAR  LGV05080
C     ON SOME RUN, YOU CAN BE CERTAIN THAT SOMETHING IS FOULED UP. LGV05090
C                                                                LGV05100
      IF (ICONV.EQ.0) GO TO 480                                  LGV05110
C                                                                LGV05120
C----------------------------------------------------------------LGV05130
C                                                                LGV05140
      CALL PRTEST (ALPHA,BETA,VS,TKMAX,EPSM,RELTOL,SCALE3,SCALE4, LGV05150
     1 MP,NDIS,MEV,IPROJ)                                        LGV05160
C                                                                LGV05170
C----------------------------------------------------------------LGV05180
C                                                                LGV05190
      IF(IPROJ.EQ.0) GO TO 470                                   LGV05200
C                                                                LGV05210
      IF(IDIST.EQ.1)  WRITE(11,360) IPROJ                        LGV05220
  360 FORMAT(' SUBROUTINE PRTEST WANTS TO RELABEL',I6,' SPURIOUS T-EIGENLGV05230
     1VALUES'/' WE ACCEPT RELABELLING ONLY IF LAST COMPONENT OF T-EIGENVLGV05240
     1ECTOR IS L.T.  1.D-10'/)                                   LGV05250
C                                                                LGV05260
      IIX = RHSEED                                               LGV05270
C                                                                LGV05280
C----------------------------------------------------------------LGV05290
C                                                                LGV05300
      CALL GENRAN(IIX,G,MEV)                                     LGV05310
C                                                                LGV05320
C----------------------------------------------------------------LGV05330
C                                                                LGV05340
      ITEN = -10                                                 LGV05350
      NISOM = NISO + MEV                                         LGV05360
      IWRITO = IWRITE                                            LGV05370
      IWRITE = 0                                                 LGV05380
C                                                                LGV05390
      DO 390 J = 1,NDIS                                          LGV05400
```

```
      IF(MP(J).NE.ITEN) GO TO 390                              LGV05410
      T0 = VS(J)                                               LGV05420
C                                                              LGV05430
C------------------------------------------------------------LGV05440
C                                                              LGV05450
      IT = MXINIT                                              LGV05460
      CALL INVERM(ALPHA,BETA,V1,V2,T0,TEMP,T1,EPSM,G,MEV,IT,IWRITE)  LGV05470
C                                                              LGV05480
C------------------------------------------------------------LGV05490
C                                                              LGV05500
      IF(TEMP.LE.1.D-10) GO TO 380                             LGV05510
C     ERROR ESTIMATE WAS NOT SMALL REJECT RELABELLING OF THIS EIGENVALUELGV05520
      IF(IDIST.EQ.1)  WRITE(11,370) J,T0,TEMP                  LGV05530
  370 FORMAT(/' LAST COMPONENT FOR',I6,'TH EIGENVALUE',E20.12/' IS TOO LLGV05540
     1ARGE = ',E15.6,' SO DO NOT ACCEPT PRTEST RELABELLING'/)  LGV05550
      MP(J) = 0                                                LGV05560
      IPROJ = IPROJ - 1                                        LGV05570
      GO TO 390                                                LGV05580
C     RELABELLING ACCEPTED                                     LGV05590
  380 NISOM = NISOM + 1                                        LGV05600
      G(NISOM) = BETAM*TEMP                                    LGV05610
  390 CONTINUE                                                 LGV05620
      IWRITE = IWRIT0                                          LGV05630
C                                                              LGV05640
      IF(IPROJ.EQ.0) GO TO 430                                 LGV05650
      WRITE(6,400) IPROJ                                       LGV05660
  400 FORMAT(/I6,' T-EIGENVALUES WERE RECLASSIFIED AS GOOD.'/  LGV05670
     1' THESE ARE IDENTIFIED IN FILE 3 BY A T-MULTIPLICITY OF -10'/' USELGV05680
     2R SHOULD INSPECT EACH TO MAKE SURE NEIGHBORS HAVE CONVERGED'/)  LGV05690
C                                                              LGV05700
      IF(IDIST.EQ.1)  WRITE(11,410) IPROJ                      LGV05710
  410 FORMAT(/I6,' T-EIGENVALUES WERE RELABELLED AS GOOD'/     LGV05720
     1' BELOW IS CORRECTED T-MULTIPLICITY PATTERN'/)           LGV05730
C                                                              LGV05740
      WRITE(6,420) NDIS, (MP(I), I=1,NDIS)                     LGV05750
      IF(IDIST.EQ.1)  WRITE(11,420) NDIS, (MP(I), I=1,NDIS)    LGV05760
  420 FORMAT(/I6,' = NDIS, T-MULTIPLICITIES (0 MEANS  SPURIOUS)'/  LGV05770
     1 6X, ' (-10) MEANS SPURIOUS T-EIGENVALUE RELABELLED AS GOOD'/(20I4LGV05780
     1))                                                       LGV05790
C                                                              LGV05800
C     RECALCULATE MINGAPS FOR DISTINCT T(1,MEV) EIGENVALUES.   LGV05810
  430 NM1 = NDIS - 1                                           LGV05820
      G(NDIS) = VS(NM1)-VS(NDIS)                               LGV05830
      G(1) = VS(2)-VS(1)                                       LGV05840
C                                                              LGV05850
      DO 440 J = 2,NM1                                         LGV05860
      T0 = VS(J)-VS(J-1)                                       LGV05870
      T1 = VS(J+1)-VS(J)                                       LGV05880
      G(J) = T1                                                LGV05890
      IF (T0.LT.T1) G(J) = -T0                                 LGV05900
  440 CONTINUE                                                 LGV05910
      IF(IPROJ.EQ.0) GO TO 470                                 LGV05920
C     WRITE TO FILE 4 ERROR ESTIMATES FOR THOSE T-EIGENVALUES RELABELLEDLGV05930
      NGOOD = 0                                                LGV05940
      DO 450 J = 1,NDIS                                        LGV05950
```

```
      IF(MP(J).EQ.0) GO TO 450                              LGV05960
      NGOOD = NGOOD + 1                                     LGV05970
      IF(MP(J).NE.ITEN) GO TO 450                           LGV05980
      TO = VS(J)                                            LGV05990
      NISO = NISO + 1                                       LGV06000
      NISOM = MEV + NISO                                    LGV06010
      WRITE(4,460) NGOOD,TO,G(NISOM),G(J)                   LGV06020
  450 CONTINUE                                              LGV06030
  460 FORMAT(I10,E25.16,2E14.3)                             LGV06040
C                                                           LGV06050
  470 CONTINUE                                              LGV06060
C                                                           LGV06070
C     WRITE THE GOOD T-EIGENVALUES TO FILE 3.  FIRST TRANSFER THEM    LGV06080
C     TO V2 AND THEIR T-MULTIPLICITIES TO THE CORRESPONDING POSITIONS LGV06090
C     IN MP AND COMPUTE THE AB-MINGAPS, THE MINIMAL GAPS BETWEEN THE  LGV06100
C     GOOD T-EIGENVALUES.  THESE GAPS WILL BE PUT IN THE ARRAY G.     LGV06110
C     SINCE G CURRENTLY CONTAINS THE MINIMAL GAPS BETWEEN THE DISTINCT LGV06120
C     EIGENVALUES OF THE T-MATRIX, THESE GAPS WILL FIRST BE           LGV06130
C     TRANSFERRED TO V1.  NOTE THAT V1<0 MEANS THAT THAT MINIMAL GAP  LGV06140
C     IN THE T-MATRIX IS DUE TO A SPURIOUS T-EIGENVALUE.             LGV06150
C     ALL THIS INFORMATION IS PRINTED TO FILE 3                       LGV06160
C                                                           LGV06170
  480 CONTINUE                                              LGV06180
C                                                           LGV06190
      NG = 0                                                LGV06200
      DO 490 I = 1,NDIS                                     LGV06210
      IF (MP(I).EQ.0) GO TO 490                             LGV06220
      NG = NG+1                                             LGV06230
      MP(NG) = MP(I)                                        LGV06240
      V2(NG) = VS(I)                                        LGV06250
      TEMP = G(I)                                           LGV06260
      TEMP = DABS(TEMP)                                     LGV06270
      J = I+1                                               LGV06280
      IF (G(I).LT.ZERO) J = I-1                             LGV06290
      IF (MP(J).EQ.0) TEMP = -TEMP                          LGV06300
      V1(NG) = TEMP                                         LGV06310
  490 CONTINUE                                              LGV06320
C                                                           LGV06330
      WRITE(6,500)MEV                                       LGV06340
  500 FORMAT(//' T-EIGENVALUE CALCULATION AT MEV = ',I6,'    IS COMPLETELGV06350
     1')                                                    LGV06360
C                                                           LGV06370
C     NG = NUMBER OF COMPUTED DISTINCT GOOD T-EIGENVALUES.  NEXT      LGV06380
C     GENERATE GAPS BETWEEN GOOD T-EIGENVALUES (ABMINGAPS) AND PUT THEM LGV06390
C     IN G.  G(J) < 0 MEANS THE ABMINGAP IS DUE TO THE LEFT-HAND GAP. LGV06400
C                                                           LGV06410
      NGM1 = NG - 1                                         LGV06420
      G(NG) = V2(NGM1)-V2(NG)                               LGV06430
      G(1) = V2(2)-V2(1)                                    LGV06440
C                                                           LGV06450
      DO 510 J = 2,NGM1                                     LGV06460
      TO = V2(J)-V2(J-1)                                    LGV06470
      T1 = V2(J+1)-V2(J)                                    LGV06480
      G(J) = T1                                             LGV06490
      IF (TO.LT.T1) G(J) = -TO                              LGV06500
```

```
  510 CONTINUE                                                       LGV06510
C                                                                    LGV06520
C     WRITE GOOD T-EIGENVALUES OUT TO FILE 3.                        LGV06530
C                                                                    LGV06540
      WRITE(3,520)NG,NDIS,MEV,N,SVSEED,MATNOA,MATNOB,MULTOL,IB,BTOL  LGV06550
  520 FORMAT(4I6,I12,2I8,'=NG,ND,MEV,N,SEED,MNA,MNB'/                LGV06560
     1 E20.12,I6,E13.4,' = MUTOL,INDEX MINIMAL BETA,BTOL'/           LGV06570
     1' EV NO',1X,'TMULT',10X,'GOOD EIGENVALUE',7X,'TMINGAP',6X,'ABMINGALGV06580
     1P')                                                           LGV06590
C                                                                    LGV06600
      WRITE(3,530)(I,MP(I),V2(I),V1(I),G(I), I=1,NG)                 LGV06610
  530 FORMAT(2I6,E25.16,2E14.3)                                      LGV06620
C                                                                    LGV06630
C     IF CONVERGENCE FLAG ICONV.NE.1 AND NUMBER OF T-MATRICES        LGV06640
C     CONSIDERED TO DATE IS LESS THAN NUMBER ALLOWED, INCREMENT MEV. LGV06650
C     AND LOOP BACK TO 210 TO REPEAT COMPUTATIONS.  RESTORE BETA(MEV+1).LGV06660
C                                                                    LGV06670
      BETA(MP1) = BETAM                                              LGV06680
C                                                                    LGV06690
      IF (MMB.LT.NMEVS.AND.ICONV.NE.1) GO TO 180                     LGV06700
C                                                                    LGV06710
C     END OF LOOP ON DIFFERENT SIZE T-MATRICES ALLOWED.             LGV06720
C                                                                    LGV06730
  540 CONTINUE                                                       LGV06740
C                                                                    LGV06750
      IF(ISTOP.EQ.0)  WRITE(6,550)                                   LGV06760
  550 FORMAT(/' T-MATRICES (ALPHA AND BETA) ARE NOW AVAILABLE, TERMINATELGV06770
     1')                                                            LGV06780
      IF (IHIS.EQ.1.AND.KMAX.NE.MOLD) WRITE(1,560)                   LGV06790
  560 FORMAT(/' ABOVE ARE THE FOLLOWING VECTORS '/                   LGV06800
     1 '   ALPHA(I), I = 1,KMAX'/                                    LGV06810
     2 '   BETA(I), I = 1,KMAX+1'/                                   LGV06820
     3 ' FINAL THREE VECTORS USED IN LANCZS SUBROUTINE'/             LGV06830
     3 ' V1 = B*V(KMAX), VS = B*V(KMAX+1), V2 = V(KMAX+1)'/          LGV06840
     4 ' ALL VECTORS IN THIS FILE HAVE HEX FORMAT 4Z20'/             LGV06850
     5 ' ----- END OF FILE 1 NEW ALPHA, BETA HISTORY---------------'////)LGV06860
C                                                                    LGV06870
      IF (ISTOP.EQ.0) GO TO 640                                      LGV06880
C                                                                    LGV06890
      WRITE(3,570)                                                   LGV06900
  570 FORMAT(/' ABOVE ARE COMPUTED GOOD T-EIGENVALUES'/              LGV06910
     1 ' NG = NUMBER OF GOOD T-EIGENVALUES COMPUTED'/                LGV06920
     2 ' ND = NUMBER OF COMPUTED DISTINCT EIGENVALUES OF T(1,MEV)'/  LGV06930
     3 ' N = ORDER OF A AND B-MATRIX,  MNA, MNB = MATRIX IDENTS'/    LGV06940
     4 ' MULTOL = T-MULTIPLICITY TOLERANCE FOR T-EIGENVALUES IN BISEC'/ LGV06950
     4 ' TMULT IS THE T-MULTIPLICITY OF GOOD T-EIGENVALUE'/          LGV06960
     5 ' TMULT = -1 MEANS SPURIOUS T-EIGENVALUE TOO CLOSE'/          LGV06970
     6 ' DO NOT COMPUTE ERROR ESTIMATES FOR SUCH EIGENVALUES'/       LGV06980
     7 ' ABMINGAP = MINIMAL GAP BETWEEN THE COMPUTED EIGENVALUES'/   LGV06990
     8 ' ABMINGAP .LT. 0. MEANS MINIMAL GAP IS DUE TO LEFT-HAND GAP'/ LGV07000
     9 ' TMINGAP= MINIMAL GAP W.R.T.  DISTINCT EIGENVALUES IN T(1,MEV)'/LGV07010
     1 ' TMINGAP .LT. 0. MEANS MINGAP IS DUE TO SPURIOUS T-EIGENVALUE'/ LGV07020
     2 ' ----- END OF FILE 3 GOODEIGENVALUES----------------------'////)LGV07030
C                                                                    LGV07040
      IF (IDIST.EQ.1) WRITE(11,580)                                  LGV07050
```

```
  580 FORMAT(/' ABOVE ARE THE DISTINCT EIGENVALUES OF T(1,MEV).'/        LGV07060
     2 ' THE FORMAT IS        T-MULTIPLICITY   T-EIGENVALUE   TMINGAP'/   LGV07070
     3 '          THIS FORMAT IS REPEATED TWICE ON EACH LINE.'/           LGV07080
     4 ' T-MULTIPLICITY = -1 MEANS THAT THE SUBROUTINE ISOEV HAS TAGGED'  LGV07090
     5/'  THIS SIMPLE T-EIGENVALUE AS HAVING A VERY CLOSE SPURIOUS'/      LGV07100
     6 '  T-EIGENVALUE SO THAT NO ERROR ESTIMATE WILL BE COMPUTED'/       LGV07110
     7 '  FOR THAT EIGENVALUE IN SUBROUTINE INVERR.'/                     LGV07120
     8 ' TMINGAP .LT. 0, TMINGAP IS DUE TO LEFT GAP .GT. 0, RIGHT GAP.'/  LGV07130
     9 ' EACH OF THE DISTINCT T-EIGENVALUE TABLES IS FOLLOWED'/           LGV07140
     9 ' BY THE T-MULTIPLICITY PATTERN.'/                                 LGV07150
     1 ' NDIS = NUMBER OF COMPUTED DISTINCT EIGENVALUES OF T(1,MEV).'/    LGV07160
     2 ' NG = NUMBER OF GOOD T-EIGENVALUES. '/                            LGV07170
     3 ' NISO = NUMBER OF ISOLATED GOOD T-EIGENVALUES. '/                 LGV07180
     4 ' NISO ALSO IS THE COUNT OF +1 ENTRIES IN T-MULTIPLICITY PATTERN. LGV07190
     5 '/' ----- END OF FILE 11 DISTINCT T-EIGENVALUES--------------'///  LGV07200
     6 )                                                                  LGV07210
C                                                                         LGV07220
      IF(NISO.NE.0)  WRITE(4,590)                                        LGV07230
  590 FORMAT(/' ABOVE ARE THE ERROR ESTIMATES OBTAINED FOR THE ISOLATED  LGV07240
     1GOOD T-EIGENVALUES'/                                                LGV07250
     1' OBTAINED VIA INVERSE ITERATION IN THE SUBROUTINE INVERR.'/        LGV07260
     1' ALL OTHER GOOD T-EIGENVALUES HAVE CONVERGED.'/                    LGV07270
     2' ERROR ESTIMATE = BETAM*ABS(UM)'/                                  LGV07280
     2' WHERE BETAM = BETA(MEV+1) AND UM = U(MEV).'/                      LGV07290
     3' U = UNIT EIGENVECTOR OF T WHERE T*U = EV*U AND EV = ISOLATED GOO  LGV07300
     3D T-EIGENVALUE.'/                                                   LGV07310
     4' TMINGAP = GAP TO NEAREST DISTINCT EIGENVALUE OF T(1,MEV).'/       LGV07320
     5' TMINGAP .LT. 0. MEANS MINGAP IS DUE TO A LEFT NEIGHBOR.'/         LGV07330
     6' ERROR ESTIMATE L.T. 0 MEANS INVERSE ITERATION DID NOT CONVERGE'/  LGV07340
     7' ------ END OF FILE 4 ERRINV ----------------------------'//)      LGV07350
      GO TO 640                                                          LGV07360
C                                                                         LGV07370
  600 CONTINUE                                                            LGV07380
C                                                                         LGV07390
      IBB = IABS(IBMEV)                                                  LGV07400
      IF (IBMEV.LT.0) WRITE(6,610) MEV,IBB,BETA(IBB)                     LGV07410
  610 FORMAT(/' PROGRAM TERMINATES BECAUSE MEV REQUESTED = ',I6,' IS .GT  LGV07420
     1',I6/' AT WHICH AN ABNORMALLY SMALL BETA = ' , E13.4,' OCCURRED'/)  LGV07430
      GO TO 640                                                          LGV07440
C                                                                         LGV07450
  620 IF (NDIS.EQ.0.AND.ISTOP.GT.0) WRITE(6,630)                         LGV07460
  630 FORMAT(/' INTERVALS SPECIFIED FOR BISECT DID NOT CONTAIN ANY T-EIG  LGV07470
     1ENVALUES'/' PROGRAM TERMINATES')                                   LGV07480
C                                                                         LGV07490
  640 CONTINUE                                                            LGV07500
C                                                                         LGV07510
      STOP                                                               LGV07520
C-----END OF MAIN PROGRAM FOR LANCZOS EIGENVALUE COMPUTATIONS----------  LGV07530
      END                                                               LGV07540
```

5.3   LGVEC: Main Program, Eigenvector Computations

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

```
C                                                                      LGV00460
C    1.  DATA/MACHEP/ STATEMENT                                        LGV00470
C    2.  ALL READ(5,*) STATEMENTS (FREE FORMAT)                        LGV00480
C    3.  FORMAT(20A4) USED WITH THE EXPLANATORY HEADER, EXPLAN         LGV00490
C    4.  HEXADECIMAL FORMAT (4Z20) USED IN ALPHA/BETA FILES 1 AND 2.   LGV00500
C                                                                      LGV00510
C    IMPORTANT NOTE:  PROGRAM ALLOWS ENLARGEMENT OF THE ALPHA, BETA    LGV00520
C    ARRAYS.  IN PARTICULAR, IF ANY ONE OF THE EIGENVALUES SUPPLIED    LGV00530
C    IS T-SIMPLE AND NOT CLOSE TO A SPURIOUS EIGENVALUE, THE PROGRAM   LGV00540
C    REQUIRES THAT KMAX BE AT LEAST 11*MEV/8 + 12.  IF KMAX IS NOT     LGV00550
C    THIS LARGE, THEN THE PROGRAM RESETS KMAX TO THIS SIZE             LGV00560
C    AND EXTENDS THE ALPHA, BETA HISTORY IF REQUIRED.                  LGV00570
C    THUS, THE DIMENSIONS OF THE ALPHA AND BETA ARRAYS MUST BE         LGV00580
C    LARGE ENOUGH TO ALLOW FOR THIS POSSIBILITY.                       LGV00590
C    REMEMBER THAT THE BETA ARRAY, BETA(J), IS SUCH THAT              LGV00600
C    J = 1,..., KMAX+1.  SO IF THE KMAX USED BY THE PROGRAM            LGV00610
C    IS TO BE 3000, THEN BETA MUST BE OF LENGTH AT LEAST 3001.         LGV00620
C                                                                      LGV00630
C----------------------------------------------------------------------LGV00640
      DOUBLE PRECISION  ALPHA(5000),BETA(5001)                         LGV00650
      DOUBLE PRECISION  V1(5000),V2(5000),VS(5000)                     LGV00660
      DOUBLE PRECISION  RITVEC(30000),TVEC(30000),GOODEV(50),EVNEW(50) LGV00670
      DOUBLE PRECISION  EVAL,EVALN,TOLN,TTOL,ERTOL,ALFA,BATA           LGV00680
      DOUBLE PRECISION  MULTOL,SCALE0,STUTOL,BTOL,LB,UB                LGV00690
      DOUBLE PRECISION  ONE,ZERO,MACHEP,EPSM,TEMP,SUM,ERRMIN,BKMIN     LGV00700
      DOUBLE PRECISION  RELTOL,ERROR,TERROR,TLAST(50)                  LGV00710
      REAL  G(10000),AMINGP(50),TMINGP(50),EXPLAN(20)                  LGV00720
      REAL  TERR(50),ERR(50),ERRDGP(50),RNORM(50),TBETA(50)           LGV00730
      INTEGER  MP(50),M1(50),M2(50),MA(50),ML(50),MINT(50),MFIN(50)    LGV00740
      INTEGER  SVSEED,SVSOLD,RHSEED,IDELTA(50)                         LGV00750
      INTEGER  MBOUND,NTVCON,SVTVEC,TVSTOP,LVCONT,ERCONT,TFLAG         LGV00760
      DOUBLE PRECISION  FINPRO                                         LGV00770
      DOUBLE PRECISION  DABS, DMAX1, DSQRT, DFLOAT                     LGV00780
      REAL ABS                                                         LGV00790
      INTEGER  IABS                                                    LGV00800
      EXTERNAL LSOLV, AMATV                                            LGV00810
C----------------------------------------------------------------------LGV00820
      DATA MACHEP/Z3410000000000000/                                  LGV00830
      EPSM = 2.D0*MACHEP                                               LGV00840
C----------------------------------------------------------------------LGV00850
C                                                                      LGV00860
C    ARRAYS MUST BE DIMENSIONED AS FOLLOWS:                            LGV00870
C    1.  ALPHA:  >= KMAXN,  BETA: >= (KMAXN+1) WHERE KMAXN, THE        LGV00880
C                LARGEST SIZE T-MATRIX CONSIDERED BY THE PROGRAM,      LGV00890
C                IS THE LARGER OF THE SIZE OF THE ALPHA, BETA HISTORY  LGV00900
C                PROVIDED ON FILE 2 (IF ANY ) AND THE SIZE WHICH THE   LGV00910
C                PROGRAM SPECIFIES INTERNALLY, THIS LATTER IS ALWAYS   LGV00920
C                < = 11*MEV / 8  +  12, WHERE MEV IS THE SIZE          LGV00930
C                T-MATRIX THAT WAS USED IN THE CORRESPONDING EIGENVALUE LGV00940
C                COMPUTATIONS.                                         LGV00950
C    2.  V1:  >= MAX(N,KMAX)                                           LGV00960
C    3.  V2,VS:  >= N                                                  LGV00970
C    4.  G:  >= MAX(N,KMAX)                                            LGV00980
C    5.  RITVEC:  >= N*NGOOD, WHERE NGOOD IS NUMBER OF EIGENVALUES     LGV00990
C                SUPPLIED TO THIS PROGRAM.                             LGV01000
```

```
C      6.  TVEC:  >= CUMULATIVE LENGTH OF ALL THE T-EIGENVECTORS         LGV01010
C          NEEDED TO GENERATE THE DESIRED RITZ VECTORS.  AN EDUCATED     LGV01020
C          GUESS AT AN APPROPRIATE LENGTH CAN BE OBTAINED BY RUNNING THE LGV01030
C          PROGRAM WITH THE FLAG MBOUND = 1 AND MULTIPLYING THE          LGV01040
C          RESULTING SIZE BY 5/4.                                        LGV01050
C      7.  GOODEV, AMINGP, TMINGP, TERR, ERR, ERRGDP, RNORM, TBETA,      LGV01060
C          TLAST, EVNEW, MP, MA, M1, M2, MINT, MFIN AND IDELTA ALL MUST  LGV01070
C          BE >= NGOOD.                                                  LGV01080
C                                                                        LGV01090
C------------------------------------------------------------------------LGV01100
C      OUTPUT HEADER                                                     LGV01110
       WRITE(6,10)                                                       LGV01120
    10 FORMAT(/' LANCZOS EIGENVECTOR PROCEDURE FOR REAL SYMMETRIC MATRICELGV01130
      1S'/)                                                              LGV01140
C                                                                        LGV01150
C      SET PROGRAM PARAMETERS                                            LGV01160
C      USER MUST NOT MODIFY SCALE0                                       LGV01170
       SCALE0 = 5.0D0                                                    LGV01180
       ZERO = 0.0D0                                                      LGV01190
       ONE = 1.0D0                                                       LGV01200
       MPMIN = -1000                                                     LGV01210
C      SET CONVERGENCE CRITERION FOR T-EIGENVECTORS.                     LGV01220
       ERTOL = 1.D-10                                                    LGV01230
C                                                                        LGV01240
C      READ USER-SPECIFIED PARAMETER FROM INPUT FILE 5 (FREE FORMAT)     LGV01250
C                                                                        LGV01260
C      READ USER-PROVIDED HEADER FOR RUN                                 LGV01270
       READ(5,20) EXPLAN                                                 LGV01280
       WRITE(6,20) EXPLAN                                                LGV01290
    20 FORMAT(20A4)                                                      LGV01300
C                                                                        LGV01310
C      READ IN THE MAXIMUM PERMISSIBLE DIMENSIONS FOR THE TVEC ARRAY     LGV01320
C      (MDIMTV), FOR THE RITVEC ARRAY (MDIMRV), AND FOR THE BETA         LGV01330
C      ARRAY (MBETA).                                                    LGV01340
       READ(5,20) EXPLAN                                                 LGV01350
       READ(5,*) MDIMTV, MDIMRV, MBETA                                   LGV01360
C                                                                        LGV01370
C      READ IN RELATIVE TOLERANCE (RELTOL) USED IN DETERMINING           LGV01380
C      APPROPRIATE SIZES FOR THE T-MATRICES TO BE USED IN THE RITZ       LGV01390
C      VECTOR COMPUTATIONS.                                              LGV01400
       READ(5,20) EXPLAN                                                 LGV01410
       READ(5,*) RELTOL                                                  LGV01420
C                                                                        LGV01430
C      SET FLAGS TO 0 OR 1:                                              LGV01440
C      MBOUND = 1:  PROGRAM TERMINATES AFTER COMPUTING 1ST GUESSES       LGV01450
C                   ON APPROPRIATE T-SIZES FOR USE IN THE RITZ VECTOR    LGV01460
C                   COMPUTATIONS                                         LGV01470
C      NTVCON = 0:  PROGRAM TERMINATES IF THE TVEC ARRAY IS NOT          LGV01480
C                   LARGE ENOUGH TO HOLD ALL THE T-EIGENVECTORS REQUIRED.LGV01490
C      SVTVEC = 0:  THE T-EIGENVECTORS ARE NOT WRITTEN TO FILE 11        LGV01500
C                   UNLESS TVSTOP = 1                                    LGV01510
C      SVTVEC = 1:  WRITE THE T-EIGENVECTORS TO FILE 11.                 LGV01520
C      TVSTOP = 1:  PROGRAM TERMINATES AFTER COMPUTING THE               LGV01530
C                   T-EIGENVECTORS                                       LGV01540
C      LVCONT = 0:  PROGRAM TERMINATES IF THE NUMBER OF T-EIGENVECTORS   LGV01550
```

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

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

```
      1I12,E12.3,I8,E13.4/)                                         LGV02660
C                                                                   LGV02670
C     IS THE ARRAY RITVEC LONG ENOUGH TO HOLD ALL OF THE DESIRED    LGV02680
C     RITZ VECTORS (APPROXIMATE EIGENVECTORS)?                      LGV02690
      NMAX = NGOOD*N                                                LGV02700
      IF(MBOUND.NE.0) GO TO 120                                     LGV02710
      IF(TVSTOP.NE.1.AND.NMAX.GT.MDIMRV) GO TO 1350                 LGV02720
C                                                                   LGV02730
C     CHECK THAT THE ORDER N AND THE MATRIX IDENTIFICATION NUMBERS  LGV02740
C     MATNOA AND MATNOB SPECIFIED BY THE USER AGREE WITH THOSE READ LGV02750
C     IN FROM FILE 3.                                               LGV02760
  120 ITEMP = (NOLD-N)**2 + (MATA-MATNOA)**2 + (MATB-MATNOB)**2     LGV02770
      IF (ITEMP.NE.0) GO TO 1370                                    LGV02780
C                                                                   LGV02790
C     READ IN FROM FILE 3, THE T-MULTIPLICITIES OF THE EIGENVALUES  LGV02800
C     WHOSE EIGENVECTORS ARE TO BE COMPUTED, THE VALUES OF THESE    LGV02810
C     EIGENVALUES AND THEIR MINIMAL GAPS AS EIGENVALUES OF THE      LGV02820
C     USER-SPECIFIED MATRIX AND AS EIGENVALUES OF THE T-MATRIX.     LGV02830
C                                                                   LGV02840
      READ(3,20) EXPLAN                                             LGV02850
      READ(3,130) (MP(J),GOODEV(J),TMINGP(J),AMINGP(J), J=1,NGOOD)  LGV02860
  130 FORMAT(6X,I6,E25.16,2E14.3)                                   LGV02870
C                                                                   LGV02880
      WRITE(6,140) (J,GOODEV(J),MP(J),TMINGP(J),AMINGP(J), J=1,NGOOD) LGV02890
  140 FORMAT(/' EIGENVALUES READ IN, T-MULTIPLICITIES, T-GAPS AND A-GAPSLGV02900
     1 '/4X,' J ',5X,'GOOD EIGENVALUE',5X,'MULT',4X,'  TMINGAP ',4X, LGV02910
     1'  ABMINGAP '/(I6,E25.16,I4,2E15.4))                          LGV02920
C                                                                   LGV02930
C     READ IN ERROR ESTIMATES                                       LGV02940
      WRITE(6,150) MEV,SVSEED                                       LGV02950
  150 FORMAT(/' THESE EIGENVALUES WERE COMPUTED USING A T-MATRIX OF LGV02960
     1ORDER ',I5/' AND SEED FOR RANDOM NUMBER GENERATOR =',I12)     LGV02970
C     CHECK WHETHER OR NOT THERE ARE ANY T-ISOLATED EIGENVALUES IN  LGV02980
C     THE EIGENVALUES PROVIDED                                      LGV02990
      DO 160 J=1,NGOOD                                              LGV03000
      IF(MP(J).EQ.1) GO TO 170                                      LGV03010
  160 CONTINUE                                                      LGV03020
      GO TO 190                                                     LGV03030
  170 READ(4,20) EXPLAN                                             LGV03040
      READ(4,20) EXPLAN                                             LGV03050
      READ(4,20) EXPLAN                                             LGV03060
      READ(4,180) NISO                                              LGV03070
  180 FORMAT(18X,I6)                                                LGV03080
      READ(4,20) EXPLAN                                             LGV03090
      READ(4,20) EXPLAN                                             LGV03100
      READ(4,20) EXPLAN                                             LGV03110
  190 DO 220 J=1,NGOOD                                              LGV03120
      ERR(J) = 0.D0                                                 LGV03130
      IF(MP(J).NE.1) GO TO 220                                      LGV03140
      READ(4,200) EVAL, ERR(J)                                      LGV03150
  200 FORMAT(10X,E25.16,E14.3)                                      LGV03160
      IF(DABS(EVAL - GOODEV(J)).LT.1.D-10) GO TO 220                LGV03170
      WRITE(6,210) EVAL,GOODEV(J)                                   LGV03180
  210 FORMAT(' PROBLEM WITH READ IN OF ERROR ESTIMATES'/' EIGENVALUE REALGV03190
     1D IN',E20.12,' DOES NOT MATCH GOODEV(J) ='/E20.12)            LGV03200
```

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

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

```
C      WHICH THE EIGENVALUE IN QUESTION IS A T-EIGENVALUE (TO WITHIN A    LGV04310
C      GIVEN TOLERANCE) AND IF POSSIBLE THE SMALLEST SIZE T-MATRIX        LGV04320
C      FOR WHICH IT IS A DOUBLE T-EIGENVALUE (TO WITHIN THE SAME          LGV04330
C      TOLERANCE).  THE SIZE T-MATRIX USED IN THE RITZ VECTOR            LGV04340
C      COMPUTATIONS IS THEN DETERMINED BY LOOPING ON SIZE OF THE         LGV04350
C      T-EIGENVECTORS, STARTING WITH A T-SIZE DETERMINED BY STURMI.      LGV04360
C                                                                        LGV04370
C                                                                        LGV04380
       STUTOL = SCALE0*MULTOL                                           LGV04390
       IF(IWRITE.EQ.1) WRITE(6,410)                                     LGV04400
   410 FORMAT(' FROM STURMI')                                           LGV04410
       DO 450 J = 1,NGOOD                                               LGV04420
       EVAL = GOODEV(J)                                                 LGV04430
C      COMPUTE THE TOLERANCES USED BY STURMI TO DETERMINE AN INTERVAL   LGV04440
C      CONTAINING THE EIGENVALUE EVAL.                                  LGV04450
       TEMP = DABS(EVAL)*RELTOL                                         LGV04460
       TOLN = DMAX1(TEMP,STUTOL)                                        LGV04470
C                                                                        LGV04480
C--------------------------------------------------------------------LGV04490
C                                                                        LGV04500
       CALL STURMI(ALPHA,BETA,EVAL,TOLN,EPSM,KMAX,MK1,MK2,IC,IWRITE)    LGV04510
C                                                                        LGV04520
C--------------------------------------------------------------------LGV04530
C                                                                        LGV04540
C      STORE THE COMPUTED ORDERS OF T-MATRICES FOR LATER PRINTOUT       LGV04550
       M1(J) = MK1                                                      LGV04560
       M2(J) = MK2                                                      LGV04570
       ML(J) = (MK1 + 3*MK2)/4                                          LGV04580
       IF(MK2.EQ.KMAX)  ML(J) = KMAX                                    LGV04590
C                                                                        LGV04600
       IF(IC.GT.0) GO TO 430                                            LGV04610
C      IC = 0 MEANS THERE WAS NO EIGENVALUE IN THE DESIGNATED INTERVAL  LGV04620
C      BY T-SIZE KMAX.  THIS MEANS THAT THE EIGENVALUE PROVIDED HAS     LGV04630
C      NOT YET CONVERGED SO ITS EIGENVECTOR IS NOT COMPUTED.           LGV04640
       WRITE(6,420) J,GOODEV(J),MK1,MK2                                LGV04650
   420 FORMAT(I6,'TH EIGENVALUE',E20.12,' HAS NOT CONVERGED '/         LGV04660
      1' SO DO NOT COMPUTE ANY T-EIGENVECTOR OR RITZ VECTOR FOR IT'    LGV04670
      1/' MK1 AND MK2 FOR THIS EIGENVALUE WERE',2I6)                   LGV04680
       MP(J) = MPMIN                                                   LGV04690
       MA(J) = -2*KMAX                                                 LGV04700
       GO TO 450                                                       LGV04710
C      COMPUTE AN APPROPRIATE SIZE T-MATRIX FOR THE GIVEN EIGENVALUE.  LGV04720
   430 IF(M2(J).EQ.KMAX) GO TO 440                                     LGV04730
C      M1 AND M2 WERE BOTH DETERMINED                                  LGV04740
       MA(J) = (3*M1(J) + M2(J))/4  + 1                                LGV04750
       GO TO 450                                                       LGV04760
C      M2 NOT DETERMINED                                               LGV04770
   440 MA(J) = (5*M1(J))/4  + 1                                        LGV04780
C                                                                        LGV04790
   450 CONTINUE                                                        LGV04800
C                                                                        LGV04810
       IF (IWRITE.EQ.1) WRITE(6,460) (MA(JJ), JJ=1,NGOOD)             LGV04820
   460 FORMAT(/' 1ST GUESS AT APPROPRIATE SIZE T-MATRICES'/           LGV04830
      1 ' ACTUAL VALUES WILL PROBABLY BE 1/4 AGAIN AS MUCH'/(13I6))   LGV04840
C                                                                        LGV04850
```

```
C     PRINT OUT TO FILE 10 1ST GUESSES AT SIZES OF THE T-MATRICES TO    LGV04860
C     BE USED IN THE EIGENVECTOR COMPUTATIONS.                          LGV04870
C     PROGRAM LOOPS ON T-SIZE TO DETERMINE APPROPRIATE SIZE T-MATRIX.   LGV04880
      WRITE(10,470) N,KMAX                                              LGV04890
  470 FORMAT(2I8,' = ORDER OF USER MATRIX AND MAX ORDER OF T(1,MEV)')   LGV04900
C                                                                       LGV04910
      WRITE(10,480)                                                     LGV04920
  480 FORMAT(/' 1ST GUESS AT APPROPRIATE SIZE T-MATRICES'/              LGV04930
     1 ' ACTUAL VALUES WILL PROBABLY BE 1/4 AGAIN AS MUCH'/)            LGV04940
C                                                                       LGV04950
      WRITE(10,490)                                                     LGV04960
  490 FORMAT(4X,'J',3X,'AB-EIGENVALUE',4X,'M1(J)',1X,'M2(J)',1X,'MA(J)')LGV04970
C                                                                       LGV04980
      WRITE(10,500) (J,GOODEV(J),M1(J),M2(J), MA(J), J=1,NGOOD)         LGV04990
  500 FORMAT(I5,E19.12,3I6)                                             LGV05000
C                                                                       LGV05010
      IF(MBOUND.EQ.1) WRITE(10,510)                                     LGV05020
  510 FORMAT(/'  EV = AB-EIGENVALUE IS A GOOD EIGENVALUE OF T(1,MEV)'/  LGV05030
     1' M1 = SMALLEST VALUE OF M SUCH THAT T(1,M) HAS AT LEAST'/        LGV05040
     1 '    ONE EIGENVALUE IN THE INTERVAL (EV-TOLN,EV+TOLN)'/          LGV05050
     1 '     TOLN(J) = DMAX1(EV(J)*RELTOL, SCALE0*MULTOL)'/             LGV05060
     1 ' M2 = SMALLEST M (IF ANY) SUCH THAT IN THE ABOVE INTERVAL'/     LGV05070
     1 '     T(1,M) HAS AT LEAST TWO EIGENVALUES '/                     LGV05080
     1 ' IABS(MA(J)) = APPROPRIATE SIZE T-MATRIX FOR EV(J)'/            LGV05090
     1 ' INITIAL VALUE OF MA(J) IS CHOSEN HEURISTICALLY'/               LGV05100
     1 ' PROGRAM LOOPS ON SIZE OF T-MATRIX TO GET BETTER SIZE'/         LGV05110
     1 ' END OF SIZES OF T-MATRICES FILE 10'///)                       LGV05120
C                                                                       LGV05130
C                                                                       LGV05140
C     TERMINATE AFTER COMPUTING 1ST GUESSES AT  SIZES OF THE            LGV05150
C     T-MATRICES REQUIRED FOR THE GIVEN EIGENVALUES?                    LGV05160
      IF(MBOUND.EQ.1) GO TO 1430                                        LGV05170
C                                                                       LGV05180
C                                                                       LGV05190
C     IS THERE ROOM FOR ALL OF THE REQUESTED T-EIGENVECTORS?            LGV05200
      MTOL = 0                                                          LGV05210
      DO 520 J = 1,NGOOD                                                LGV05220
      IF(MP(J).EQ.MPMIN) GO TO 520                                      LGV05230
      MTOL = MTOL + IABS(MA(J))                                         LGV05240
  520 CONTINUE                                                          LGV05250
      MTOL = (5*MTOL)/4                                                 LGV05260
      IF(MTOL.GT.MDIMTV.AND.NTVCON.EQ.0) GO TO 1450                     LGV05270
C                                                                       LGV05280
C----------------------------------------------------------------------LGV05290
C     GENERATE A RANDOM VECTOR TO BE USED REPEATEDLY BY                 LGV05300
C     SUBROUTINE INVERM                                                 LGV05310
C                                                                       LGV05320
      IIL = RHSEED                                                      LGV05330
      CALL GENRAN(IIL,G,KMAX)                                           LGV05340
C                                                                       LGV05350
C--------------------------------------------------------------------- LGV05360
C                                                                       LGV05370
C     LOOP ON GIVEN EIGENVALUES TO COMPUTE THE CORRESPONDING            LGV05380
C     T-EIGENVECTOR.                                                    LGV05390
C                                                                       LGV05400
```

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

```
      1ING IS WRONG, THEREFORE NO EIGENVECTOR WILL BE COMPUTED FOR THIS ELGV05960
      1IGENVALUE'/)                                                      LGV05970
C                                                                        LGV05980
      MP(J) = MPMIN                                                      LGV05990
      MA(J) = -2*KMAX                                                    LGV06000
      GO TO 710                                                          LGV06010
C                                                                        LGV06020
  590 CONTINUE                                                           LGV06030
      ILBIS = 0                                                          LGV06040
C                                                                        LGV06050
      EVNEW(J) = EVALN                                                   LGV06060
      EVAL = EVALN                                                       LGV06070
      MTOL = MTOL+KMAXU                                                  LGV06080
C                                                                        LGV06090
C     IS THERE ROOM IN TVEC ARRAY FOR THE NEXT T-EIGENVECTOR?            LGV06100
C     IF NOT, SKIP TO RITZ VECTOR COMPUTATIONS.                          LGV06110
      IF (MTOL.GT.MDIMTV) GO TO 720                                      LGV06120
C                                                                        LGV06130
      IT = 3                                                             LGV06140
      KINT = MTOL - KMAXU +1                                             LGV06150
C                                                                        LGV06160
C     RECORD THE BEGINNING AND END OF THE T-EIGENVECTOR BEING COMPUTED   LGV06170
      MINT(J) = KINT                                                     LGV06180
      MFIN(J) = MTOL                                                     LGV06190
C                                                                        LGV06200
C------------------------------------------------------------------------LGV06210
C     SUBROUTINE INVERM DOES INVERSE ITERATION, I.E. SOLVES              LGV06220
C     (T(1,KMAXU) - EVAL)*U = RHS  FOR EACH EIGENVALUE TO OBTAIN THE     LGV06230
C     DESIRED T-EIGENVECTOR.                                             LGV06240
C                                                                        LGV06250
      IF(IWRITE.EQ.1)  WRITE(6,600) J                                    LGV06260
  600 FORMAT(/I6,'TH EIGENVALUE')                                        LGV06270
C                                                                        LGV06280
      CALL INVERM(ALPHA,BETA,V1,TVEC(KINT),EVAL,ERROR,TERROR,EPSM,       LGV06290
     1 G,KMAXU,IT,IWRITE)                                                LGV06300
C                                                                        LGV06310
C------------------------------------------------------------------------LGV06320
C                                                                        LGV06330
      TERR(J) = TERROR                                                   LGV06340
      TLAST(J) = ERROR                                                   LGV06350
      KMAXU1 = KMAXU + 1                                                 LGV06360
      TBETA(J) = BETA(KMAXU1)*ERROR                                      LGV06370
C                                                                        LGV06380
C     AFTER COMPUTING EACH OF THE T-EIGENVECTORS,                        LGV06390
C     CHECK THE SIZE OF THE ERROR ESTIMATE, ERROR.                       LGV06400
C     IF THIS ESTIMATE IS NOT AS SMALL AS DESIRED AND                    LGV06410
C     |MA(J)| < ML(J), ATTEMPT TO INCREASE THE SIZE OF |MA(J)|           LGV06420
C     AND REPEAT THE T-EIGENVECTOR COMPUTATIONS.                         LGV06430
C                                                                        LGV06440
      IF(ERROR.LT.ERTOL.OR.TFLAG.EQ.1)  GO TO 700                        LGV06450
C                                                                        LGV06460
      IF(ERROR.GE.ERRMIN) GO TO 610                                      LGV06470
C     LAST COMPONENT IS LESS THAN MINIMAL TO DATE                        LGV06480
      ERRMIN = ERROR                                                     LGV06490
      MABEST = MA(J)                                                     LGV06500
```

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

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

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

```
      IIL = SVSEED                                              LGV08160
      CALL GENRAN(IIL,G,N)                                      LGV08170
C                                                               LGV08180
C--------------------------------------------------------------LGV08190
C                                                               LGV08200
      DO 910 K = 1,N                                            LGV08210
  910 V2(K) = G(K)                                              LGV08220
C                                                               LGV08230
C--------------------------------------------------------------LGV08240
C     COMPUTE L-TRANSPOSE*V2 AND ITS NORM                       LGV08250
      ISOLV = 2                                                 LGV08260
      CALL LSOLV(V2,VS,ISOLV)                                   LGV08270
      SUM = FINPRO(N,VS(1),1,VS(1),1)                           LGV08280
C--------------------------------------------------------------LGV08290
C                                                               LGV08300
C     NORMALIZE STARTING VECTORS:  (V2-TRANSPOSE*B*V2) = 1      LGV08310
      SUM = ONE/DSQRT(SUM)                                      LGV08320
      DO 920 K = 1,N                                            LGV08330
      VS(K) = SUM*VS(K)                                         LGV08340
  920 V2(K) = SUM*V2(K)                                         LGV08350
C                                                               LGV08360
C--------------------------------------------------------------LGV08370
C     INITIALIZE V1 = B*V2 = L*VS                               LGV08380
      ISOLV = 1                                                 LGV08390
      CALL LSOLV(VS,V1,ISOLV)                                   LGV08400
C--------------------------------------------------------------LGV08410
C                                                               LGV08420
      DO 930 K = 1,N                                            LGV08430
      VS(K) = V1(K)                                             LGV08440
  930 V1(K) = ZERO                                              LGV08450
C                                                               LGV08460
      IVEC = 1                                                  LGV08470
      BATA = ZERO                                               LGV08480
C                                                               LGV08490
      GO TO 1000                                                LGV08500
C                                                               LGV08510
C     VS = B*V(I),  V1 = B*V(I-1),  V2 = V(I)                   LGV08520
  940 CONTINUE                                                  LGV08530
      SUM = BATA                                                LGV08540
C                                                               LGV08550
C--------------------------------------------------------------LGV08560
C     COMPUTE V1 = A*V2 - SUM*V1                                LGV08570
      CALL MATVEC(V2,V1,SUM)                                    LGV08580
C     COMPUTE ALFA                                              LGV08590
      ALFA = FINPRO(N,V1(1),1,V2(1),1)                          LGV08600
C--------------------------------------------------------------LGV08610
C                                                               LGV08620
      DO 950 K = 1,N                                            LGV08630
  950 V1(K) = V1(K)-ALFA*VS(K)                                  LGV08640
C                                                               LGV08650
C     SET V1 = B*V(IVEC) AND VS = (NEW BATA)*B*V(IVEC+1)        LGV08660
      DO 960 K = 1,N                                            LGV08670
      TEMP = V1(K)                                              LGV08680
      V1(K) = VS(K)                                             LGV08690
  960 VS(K) = TEMP                                              LGV08700
```

```
C                                                               LGV08710
C---------------------------------------------------------------LGV08720
C     COMPUTE V2 = (L-INVERSE)*VS                               LGV08730
      ISOLV = 3                                                 LGV08740
      CALL LSOLV(VS,V2,ISOLV)                                   LGV08750
C     COMPUTE NEXT BATA                                         LGV08760
      SUM = FINPRO(N,V2(1),1,V2(1),1)                           LGV08770
C---------------------------------------------------------------LGV08780
C                                                               LGV08790
      BATA = DSQRT(SUM)                                         LGV08800
      TEMP = BETA(IVEC)                                         LGV08810
      TEMP = DABS(BATA - TEMP)/TEMP                             LGV08820
      IF (TEMP.LT.1.0D-10)GO TO 980                             LGV08830
C                                                               LGV08840
C     THE BETA BEING REGENERATED DO NOT MATCH THE BETA IN FILE 2. LGV08850
C     SOMETHING IS WRONG IN THE LANCZOS VECTOR GENERATION.      LGV08860
C     PROGRAM TERMINATES FOR USER TO CORRECT THE PROBLEM        LGV08870
C     WHICH MUST BE IN THE STARTING VECTOR GENERATION OR IN     LGV08880
C     THE SUBROUTINES AMATV AND LSOLV SUPPLIED.                 LGV08890
C     THESE SUBROUTINES MUST BE THE SAME ONES USED IN THE       LGV08900
C     EIGENVALUE COMPUTATIONS OR A MISMATCH WILL ENSUE.         LGV08910
C                                                               LGV08920
      WRITE(6,970) IVEC,BATA,BETA(IVEC),TEMP                    LGV08930
  970 FORMAT(/2X,'IVEC',16X,'BATA',10X,'BETA(IVEC)',14X,'RELDIF'/I6, LGV08940
     13E20.12/' IN LANCZOS VECTOR REGENERATION THE ENTRIES OF THE TRIDIALGV08950
     1GONAL MATRICES BEING'/' GENERATED ARE NOT THE SAME AS THOSE IN THELGV08960
     1 MATRIX SUPPLIED ON FILE 2.'/' THEREFORE SOMETHING IS BEING INITIALGV08970
     1LIZED OR COMPUTED DIFFERENTLY FROM THE WAY'/' IT WAS COMPUTED IN TLGV08980
     1HE EIGENVALUE COMPUTATIONS'/' THE PROGRAM TERMINATES FOR THE USER LGV08990
     1TO DETERMINE WHAT THE PROBLEM IS'/)                       LGV09000
      GO TO 1590                                                LGV09010
  980 CONTINUE                                                  LGV09020
C                                                               LGV09030
C---------------------------------------------------------------LGV09040
      ISOLV = 4                                                 LGV09050
      CALL LSOLV(V2,V2,ISOLV)                                   LGV09060
C---------------------------------------------------------------LGV09070
C                                                               LGV09080
      SUM = ONE/BATA                                            LGV09090
      DO 990 K = 1,N                                            LGV09100
      V2(K) = SUM*V2(K)                                         LGV09110
  990 VS(K) = SUM*VS(K)                                         LGV09120
C                                                               LGV09130
 1000 CONTINUE                                                  LGV09140
C                                                               LGV09150
      LFIN = 0                                                  LGV09160
      DO 1020 J = 1,NGOODC                                      LGV09170
      LL = LFIN                                                 LGV09180
      LFIN = LFIN + N                                           LGV09190
C                                                               LGV09200
      IF(IABS(MA(J)).LT.IVEC.OR.MP(J).EQ.MPMIN) GO TO 1020      LGV09210
      II = IVEC + MINT(J) - 1                                   LGV09220
      TEMP = TVEC(II)                                           LGV09230
C     II IS THE (IVEC)TH COMPONENT OF THE T-EIGENVECTOR CONTAINED LGV09240
C     IN TVEC(MINT(J)).                                         LGV09250
```

```
C                                                              LGV09260
      DO 1010 K = 1,N                                          LGV09270
      LL = LL + 1                                              LGV09280
 1010 RITVEC(LL) = TEMP*V2(K) + RITVEC(LL)                     LGV09290
C                                                              LGV09300
 1020 CONTINUE                                                 LGV09310
C                                                              LGV09320
      IVEC = IVEC + 1                                          LGV09330
      IF (IVEC.LE.KMAXU) GO TO 940                             LGV09340
C                                                              LGV09350
C     RITZVECTOR GENERATION IS COMPLETE. B-NORMALIZE EACH RITZVECTOR. LGV09360
C     NOTE THAT IF CERTAIN RITZ VECTORS WERE NOT COMPUTED THEN THAT    LGV09370
C     PORTION OF THE RITVEC ARRAY WAS NOT UTILIZED.           LGV09380
C                                                              LGV09390
      LFIN = 0                                                 LGV09400
      DO 1090 J = 1,NGOODC                                     LGV09410
C                                                              LGV09420
      KK = LFIN                                                LGV09430
      LFIN = LFIN + N                                          LGV09440
      IF(MP(J).EQ.MPMIN) GO TO 1090                            LGV09450
C                                                              LGV09460
      DO 1030 K = 1,N                                          LGV09470
      KK = KK + 1                                              LGV09480
 1030 V2(K) = RITVEC(KK)                                       LGV09490
C                                                              LGV09500
C------------------------------------------------------------ LGV09510
      ISOLV = 2                                                LGV09520
      CALL LSOLV(V2,VS,ISOLV)                                  LGV09530
      SUM = FINPRO(N,VS(1),1,VS(1),1)                          LGV09540
C------------------------------------------------------------ LGV09550
C                                                              LGV09560
      SUM = DSQRT(SUM)                                         LGV09570
      RNORM(J) = SUM                                           LGV09580
      TEMP = DABS(ONE-SUM)                                     LGV09590
      SUM = ONE/SUM                                            LGV09600
C                                                              LGV09610
      DO 1040 K = 1,N                                          LGV09620
      VS(K) = SUM*VS(K)                                        LGV09630
      V2(K) = SUM*V2(K)                                        LGV09640
 1040 CONTINUE                                                 LGV09650
C                                                              LGV09660
C------------------------------------------------------------ LGV09670
      ISOLV = 1                                                LGV09680
      CALL LSOLV(VS,V1,ISOLV)                                  LGV09690
C------------------------------------------------------------ LGV09700
C                                                              LGV09710
C     V1 = B*V2                                                LGV09720
      EVAL = EVNEW(J)                                          LGV09730
C                                                              LGV09740
C     COMPUTE ERROR IN RITZ VECTOR CONSIDERED AS A EIGENVECTOR OF A.   LGV09750
C     V1 = A*RITVEC - EVAL*B*RITVEC                            LGV09760
C                                                              LGV09770
C------------------------------------------------------------ LGV09780
      CALL AMATV(V2,V1,EVAL)                                   LGV09790
      SUM = FINPRO(N,V1(1),1,V1(1),1)                          LGV09800
```

```
C---------------------------------------------------------------------LGV09810
C                                                                     LGV09820
      SUM = DSQRT(SUM)                                                LGV09830
      ERR(J) = SUM                                                    LGV09840
      GAP = ABS(AMINGP(J))                                            LGV09850
      ERRDGP(J) = SUM/GAP                                             LGV09860
C                                                                     LGV09870
C                                                                     LGV09880
      IF (JPERM.EQ.0) GO TO 1050                                      LGV09890
C                                                                     LGV09900
C---------------------------------------------------------------------LGV09910
C     ON RETURN V2 = P(TRANSPOSE)*V2                                  LGV09920
      IPERM = 2                                                       LGV09930
      CALL LPERM(V2,V1,IPERM)                                         LGV09940
C---------------------------------------------------------------------LGV09950
C                                                                     LGV09960
 1050 CONTINUE                                                        LGV09970
      KK = LFIN - N                                                   LGV09980
      DO 1060 K = 1,N                                                 LGV09990
      KK = KK + 1                                                     LGV10000
 1060 RITVEC(KK) = V2(K)                                              LGV10010
C                                                                     LGV10020
      IF (IWRITE.NE.0) WRITE(6,1070) J,GOODEV(J)                      LGV10030
 1070 FORMAT(/I5,' TH EIGENVALUE CONSIDERED = ',E20.12/)              LGV10040
C                                                                     LGV10050
      IF (IWRITE.NE.0) WRITE(6,1080) TERR(J),TBETA(J),TEMP            LGV10060
 1080 FORMAT(' NORM OF ERROR IN T-EIGENVECTOR = ',E14.3/              LGV10070
     1 ' BETA(MA(J)+1)*U(MA(J)) = ',E14.3/                            LGV10080
     1 ' ABS(NORM(RITVEC) - 1.0)  = ',E14.3/)                         LGV10090
C                                                                     LGV10100
 1090 CONTINUE                                                        LGV10110
C                                                                     LGV10120
C     RITZVECTORS ARE NORMALIZED AND ERROR ESTIMATES ARE IN ERR ARRAY LGV10130
C     AND IN ERRDGP ARRAY. STORE EVERYTHING                           LGV10140
C                                                                     LGV10150
C                                                                     LGV10160
      WRITE(9,1100)                                                   LGV10170
 1100 FORMAT(2X,'AB-EIGENVALUE',2X,'MA(J)',2X,'AB-MINGAP',5X,'ABERROR',1LGV10180
     1X, 'ABERROR/GAP',6X,'TERROR')                                   LGV10190
C                                                                     LGV10200
      WRITE(13,1110)                                                  LGV10210
 1110 FORMAT(12X,'AB-EIGENVALUE',5X,'RITZNORM',5X,'ABMINGAP',5X,       LGV10220
     1 'TBETA(J)',5X,'TLAST(J)')                                      LGV10230
C                                                                     LGV10240
      DO 1140 J=1,NGOODC                                              LGV10250
C                                                                     LGV10260
      IF(MP(J).EQ.MPMIN) GO TO 1140                                   LGV10270
C                                                                     LGV10280
      WRITE(9,1120)EVNEW(J),MA(J),AMINGP(J),ERR(J),ERRDGP(J),TERR(J)   LGV10290
 1120 FORMAT(E15.8,I6,4E12.4)                                         LGV10300
C                                                                     LGV10310
      WRITE(13,1130) EVNEW(J),RNORM(J),AMINGP(J),TBETA(J),TLAST(J)     LGV10320
 1130 FORMAT(E25.14,4E13.5)                                           LGV10330
C                                                                     LGV10340
 1140 CONTINUE                                                        LGV10350
```

```
C                                                                   LGV10360
      IF(MREJEC.EQ.0) GO TO 1220                                    LGV10370
      WRITE(9,1150)                                                 LGV10380
 1150 FORMAT(/' RITZ VECTORS WERE NOT COMPUTED FOR THE FOLLOWING EIGENVALGV10390
     1LUES'/' EITHER BECAUSE THEY HAD NOT CONVERGED OR BECAUSE THE ERRORLGV10400
     1 ESTIMATE'/'  WAS NOT AS SMALL AS DESIRED'/)                  LGV10410
C                                                                   LGV10420
      DO 1210 J = 1,NGOODC                                          LGV10430
      IF(MP(J).NE.MPMIN) GO TO 1210                                 LGV10440
C     WRITE OUT MESSAGE FOR EACH EIGENVALUE FOR WHICH NO EIGENVECTOR LGV10450
C     WAS COMPUTED.                                                 LGV10460
C                                                                   LGV10470
      WRITE(9,1160)                                                 LGV10480
 1160 FORMAT(2X,'AB-EIGENVALUE',3X,'MA(J)',5X,'AMINGP(J)',6X,'TLAST(J)',LGV10490
     13X,'MP(J)')                                                   LGV10500
      WRITE(9,1170) GOODEV(J),MA(J),AMINGP(J),TBETA(J),MP(J)        LGV10510
 1170 FORMAT(E15.8,I8,2E14.4,I8)                                    LGV10520
C                                                                   LGV10530
      WRITE(13,1180)                                                LGV10540
 1180 FORMAT(/' RITZ VECTORS WERE NOT COMPUTED FOR THE FOLLOWING EIGENVALGV10550
     1LUES'/' BECAUSE THEY HAD NOT CONVERGED'/)                     LGV10560
C                                                                   LGV10570
      WRITE(13,1190)                                                LGV10580
 1190 FORMAT(2X,'AB-EIGENVALUE',3X,'MA(J)',3X,'M1(J)',3X,'M2(J)',3X,'MP(LGV10590
     1J)'/)                                                         LGV10600
      WRITE(13,1200) GOODEV(J),MA(J),M1(J),M2(J),MP(J)             LGV10610
 1200 FORMAT(E15.8,4I8)                                             LGV10620
C                                                                   LGV10630
 1210 CONTINUE                                                      LGV10640
 1220 CONTINUE                                                      LGV10650
C                                                                   LGV10660
      WRITE(9,1230)                                                 LGV10670
 1230 FORMAT(/' ABOVE ARE ERROR ESTIMATES FOR THE AB AND T EIGENVECTORS'LGV10680
     1 /' ASSOCIATED WITH THE AB-EIGENVALUES LISTED IN COLUMN 1'/   LGV10690
     1 ' ABERROR = NORM(A*X - EV*B*X)   TERROR = NORM(T*Y - EV*Y)   LGV10700
     1 '/' WHERE T = T(1,MA(J))    X = RITZ VECTOR = V*Y  V = SUCCESSIVELGV10710
     1 '/' LANCZOS VECTORS. ABMINGAP = GAP TO NEAREST AB-EIGENVALUE'//) LGV10720
C                                                                   LGV10730
      WRITE(13,1240)                                                LGV10740
 1240 FORMAT(/' ABOVE ARE ERROR ESTIMATES ASSOCIATED WITH THE AB-EIGVALSLGV10750
     1 '/' RITZNORM = NORM(COMPUTED RITZ VECTOR)'/                  LGV10760
     1 ' TBETA(J) = BETA(MA(J)+1)*Y(MA(J)),  T*Y = EVAL*Y'/         LGV10770
     1 ' TLAST(J) = Y(MA(J))'/                                      LGV10780
     1 ' ABMINGAP = GAP TO NEAREST AB-EIGENVALUE'/)                 LGV10790
C                                                                   LGV10800
C     NUMBER OF RITZ VECTORS COMPUTED                               LGV10810
      NCOMPU = NGOODC - MREJEC                                      LGV10820
      WRITE(12,1250) N,NCOMPU,NGOODC,MATNOA,MATNOB                  LGV10830
 1250 FORMAT(3I6,2I8,' SIZE A, NO.RITZVECS, NO.EVALS,MATNOA,MATNOB') LGV10840
C                                                                   LGV10850
      LFIN = 0                                                      LGV10860
      DO 1310 J = 1,NGOODC                                          LGV10870
      LINT = LFIN + 1                                               LGV10880
      LFIN = LFIN + N                                               LGV10890
C                                                                   LGV10900
```

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

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

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

5.4   LGMULT: LANCZS and Sample Matrix-Vector Multiply Subnroutines

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

```
C------------------------------------------------------------------LGM00450
C     ALPHA, BETA, AND LANCZOS VECTOR GENERATION                    LGM00460
C     ALPHA BETA GENERATION STARTS WITH IVEC = 1, BETA(1) = ZERO,   LGM00470
C     V2 =  RANDOM VECTOR WITH UNIT B-NORM, VS = B*V2, AND V1 = 0.; LGM00480
C     OR STARTS WITH AN EXISTING ALPHA/BETA FILE AND THE MOST       LGM00490
C     RECENTLY GENERATED V2, VS, AND V1.                            LGM00500
C                                                                   LGM00510
      ZERO = 0.0D0                                                  LGM00520
      ONE  = 1.0D0                                                  LGM00530
      IF (MOLD1.GT.1) GO TO 40                                      LGM00540
      BETA(1) = ZERO                                                LGM00550
      IIL = IIX                                                     LGM00560
C                                                                   LGM00570
C------------------------------------------------------------------LGM00580
      CALL GENRAN(IIL,G,N)                                          LGM00590
C------------------------------------------------------------------LGM00600
C                                                                   LGM00610
      DO 10 K = 1,N                                                 LGM00620
   10 V2(K) = G(K)                                                  LGM00630
C                                                                   LGM00640
C------------------------------------------------------------------LGM00650
C     COMPUTE L-TRANSPOSE*V2 AND ITS NORM                           LGM00660
      ISOLV = 2                                                     LGM00670
      CALL LSOLV(V2,VS,ISOLV)                                       LGM00680
      SUM = FINPRO(N,VS(1),1,VS(1),1)                               LGM00690
C------------------------------------------------------------------LGM00700
C                                                                   LGM00710
C     NORMALIZE STARTING VECTORS:  (V2-TRANSPOSE*B*V2) = 1          LGM00720
      SUM = ONE/DSQRT(SUM)                                          LGM00730
      DO 20 K = 1,N                                                 LGM00740
      VS(K) = SUM*VS(K)                                             LGM00750
   20 V2(K) = SUM*V2(K)                                             LGM00760
C                                                                   LGM00770
C------------------------------------------------------------------LGM00780
C     INITIALIZE V1 = B*V2 = L*VS                                   LGM00790
      ISOLV = 1                                                     LGM00800
      CALL LSOLV(VS,V1,ISOLV)                                       LGM00810
C------------------------------------------------------------------LGM00820
C                                                                   LGM00830
      DO 30 K = 1,N                                                 LGM00840
      VS(K) = V1(K)                                                 LGM00850
   30 V1(K) = 0.D0                                                  LGM00860
   40 CONTINUE                                                      LGM00870
C                                                                   LGM00880
C     INITIALIZATIONS ARE:  VS = B*V(I),  V1 = B*V(I-1),  V2 = V(I) LGM00890
C                                                                   LGM00900
      DO 80 IVEC = MOLD1,KMAX                                       LGM00910
      SUM = BETA(IVEC)                                              LGM00920
C                                                                   LGM00930
C------------------------------------------------------------------LGM00940
C     COMPUTE V1 = A*V2 - SUM*V1                                    LGM00950
      CALL MATVEC(V2,V1,SUM)                                        LGM00960
C     COMPUTE ALPHA(I)                                              LGM00970
      SUM = FINPRO(N,V1(1),1,V2(1),1)                               LGM00980
C------------------------------------------------------------------LGM00990
```

```
C                                                                       LGM01000
      ALPHA(IVEC) = SUM                                                 LGM01010
      DO 50 K = 1,N                                                     LGM01020
   50 V1(K) = V1(K)-SUM*VS(K)                                           LGM01030
C                                                                       LGM01040
C     SET V1 = B*V(IVEC) AND VS = BETA(IVEC+1)*B*V(IVEC+1)              LGM01050
      DO 60 K = 1,N                                                     LGM01060
      TEMP = V1(K)                                                      LGM01070
      V1(K) = VS(K)                                                     LGM01080
   60 VS(K) = TEMP                                                      LGM01090
C                                                                       LGM01100
C-----------------------------------------------------------------------LGM01110
C     COMPUTE V2 = (L-INVERSE)*VS                                       LGM01120
      ISOLV = 3                                                         LGM01130
      CALL LSOLV(VS,V2,ISOLV)                                           LGM01140
C     COMPUTE BETA(IVEC+1)                                              LGM01150
      SUM = FINPRO(N,V2(1),1,V2(1),1)                                   LGM01160
C-----------------------------------------------------------------------LGM01170
C                                                                       LGM01180
      IN = IVEC+1                                                       LGM01190
      BETA(IN) = DSQRT(SUM)                                             LGM01200
C                                                                       LGM01210
C-----------------------------------------------------------------------LGM01220
      ISOLV = 4                                                         LGM01230
      CALL LSOLV(V2,V2,ISOLV)                                           LGM01240
C-----------------------------------------------------------------------LGM01250
C                                                                       LGM01260
      SUM = ONE/BETA(IN)                                                LGM01270
      DO 70 K = 1,N                                                     LGM01280
      V2(K) = SUM*V2(K)                                                 LGM01290
   70 VS(K) = SUM*VS(K)                                                 LGM01300
C                                                                       LGM01310
   80 CONTINUE                                                          LGM01320
C                                                                       LGM01330
      RETURN                                                            LGM01340
C-----END LANCZS--------------------------------------------------------LGM01350
      END                                                               LGM01360
C                                                                       LGM01370
C-----USPEC (GENERAL SYMMETRIC SPARSE MATRICES)------------------------LGM01380
C                                                                       LGM01390
C     SUBROUTINE USPECA(N,MATNOA)                                       LGM01400
      SUBROUTINE GUSPEC(N,MATNOA)                                       LGM01410
C                                                                       LGM01420
C-----------------------------------------------------------------------LGM01430
      DOUBLE PRECISION  ASD(10000),AD(5010)                            LGM01440
      INTEGER  IROW(10000),ICOL(5010)                                  LGM01450
C-----------------------------------------------------------------------LGM01460
C   USPEC DIMENSIONS AND INITIALIZES THE ARRAYS NEEDED TO DEFINE        LGM01470
C   THE USER-SPECIFIED A-MATRIX AND THEN PASSES THE STORAGE LOCATIONS   LGM01480
C   OF THESE ARRAYS TO THE MULTIPLY SUBROUTINE AMATV.                   LGM01490
C                                                                       LGM01500
C   MATRIX IS STORED IN FOLLOWING SPARSE MATRIX FORMAT:                 LGM01510
C   N = ORDER OF A-MATRIX,                                              LGM01520
C   NZS = NUMBER OF NONZERO SUBDIAGONAL ENTRIES,                        LGM01530
C   NZL = INDEX OF LAST COLUMN CONTAINING NONZERO SUBDIAGONAL ENTRIES,  LGM01540
```

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

```
C-----END OF USPECA-------------------------------------------------LGM02100
      END                                                           LGM02110
C                                                                   LGM02120
C-----USPECB FOR CHOLESKY FACTORS OF GENERAL SPARSE SYMMETRIC MATRIX----LGM02130
C                                                                   LGM02140
C     SUBROUTINE USPECB(N,MATNOB)                                   LGM02150
      SUBROUTINE CUSPEC(N,MATNOB)                                   LGM02160
C                                                                   LGM02170
C------------------------------------------------------------------LGM02180
      DOUBLE PRECISION  BD(2200),BSD(10000)                         LGM02190
      INTEGER  KCOL(2200),KROW(10000),IPR(2200),IPT(2200)           LGM02200
C------------------------------------------------------------------LGM02210
C     DIMENSIONS ARRAYS NEEDED TO DEFINE CHOLESKY FACTOR OF B-MATRIX,   LGM02220
C     READS CHOLESKY FACTOR FROM FILE 7, AND THEN PASSES STORAGE    LGM02230
C     LOCATIONS OF THESE ARRAYS TO THE MATRIX SOLVE SUBROUTINE LSOLV   LGM02240
C                                                                   LGM02250
C     THE LANCZOS PROCEDURE LGVAL WILL USE THE CHOLESKY FACTORS ON  LGM02260
C     FILE 7.  THESE FACTORS MAY CORRESPOND TO A PERMUTED VERSION OF   LGM02270
C     THE GIVEN B-MATRIX IN WHICH CASE THIS PERMUTATION WILL BE STORED   LGM02280
C     IN IPR. THE ITH ROW OF THE PERMUTED B WILL CORRESPOND TO THE  LGM02290
C     JTH ROW OF B WHERE J = IPR(I) AND I = IPT(J).  IF B IS        LGM02300
C     PERMUTED, THE LANCZOS PROCEDURE ASSUMES THAT THE USER-PROVIDED   LGM02310
C     A-MATRIX IS IN FACT, THE CORRESPONDING PERMUTED VERSION OF THE   LGM02320
C     ORIGINAL A-MATRIX.                                            LGM02330
C                                                                   LGM02340
C     THE CHOLESKY FACTOR IS STORED IN THE FOLLOWING SPARSE FORMAT:  LGM02350
C     N = ORDER OF THE B-MATRIX.                                    LGM02360
C     NZT = NUMBER OF NONZERO SUBDIAGONAL ENTRIES IN THE CHOLESKY   LGM02370
C           FACTOR, L.                                              LGM02380
C     KCOL(J), J=1,N IS THE NUMBER OF NONZERO SUBDIAGONAL ELEMENTS IN   LGM02390
C             COLUMN J OF L.                                        LGM02400
C     KROW(K), K=1,NZT IS THE ROW INDEX FOR CORRESPONDING ENTRY BSD(K).  LGM02410
C     BD(J), J = 1,N CONTAINS THE DIAGONAL ENTRIES OF L.            LGM02420
C     BSD(K), K =1,NZT CONTAINS THE NONZERO SUBDIAGONAL ENTRIES OF L   LGM02430
C           BY COLUMN.                                              LGM02440
C------------------------------------------------------------------LGM02450
C                                                                   LGM02460
C     READ CHOLESKY FACTOR FROM FILE 7.  MUST BE STORED             LGM02470
C     IN SPARSE MATRIX FORMAT.                                      LGM02480
      READ(7,10) NZT,NOLD,NZL,MATOLD,JPERM                          LGM02490
   10 FORMAT(I10,2I6,I8,I6)                                         LGM02500
C                                                                   LGM02510
      WRITE(6,20) NZT,NZL,N,NOLD,MATOLD,JPERM                       LGM02520
   20 FORMAT(' HEADER, CHOLESKY FACTOR FILE'/                       LGM02530
     1 3X,'NZT',3X,'NZL',5X,'N',2X,'NOLD',2X,'MATOLD',1X,'JPERM'/   LGM02540
     1 4I6,I8,I6/)                                                  LGM02550
C                                                                   LGM02560
      IF (N.NE.NOLD.OR.MATNOB.NE.MATOLD) GO TO 70                   LGM02570
C                                                                   LGM02580
      READ(7,30) (KCOL(K), K = 1,NZL)                               LGM02590
      READ(7,30) (KROW(K), K = 1,NZT)                               LGM02600
   30 FORMAT(13I6)                                                  LGM02610
      READ(7,40) (BD(K), K = 1,N)                                   LGM02620
      READ(7,40) (BSD(K), K = 1,NZT)                                LGM02630
   40 FORMAT(4Z20)                                                  LGM02640
```

```
C  20 FORMAT(3E25.16)                                              LGM02650
C                                                                  LGM02660
      IF(JPERM.EQ.0)  GO TO 60                                     LGM02670
C                                                                  LGM02680
      READ(7,30) (IPR(K), K = 1,N)                                 LGM02690
      DO 50 K = 1,N                                                LGM02700
      J = IPR(K)                                                   LGM02710
   50 IPT(J) = K                                                   LGM02720
C                                                                  LGM02730
C-----------------------------------------------------------------LGM02740
      CALL LPERME(IPR,IPT,N)                                       LGM02750
C-----------------------------------------------------------------LGM02760
C                                                                  LGM02770
   60 CONTINUE                                                     LGM02780
C                                                                  LGM02790
C-----------------------------------------------------------------LGM02800
C     PASS STORAGE LOCATIONS OF FACTORS TO SUBROUTINE LSOLV        LGM02810
      CALL LSOLVE(BSD,BD,KCOL,KROW,N,NZT,NZL)                      LGM02820
C-----------------------------------------------------------------LGM02830
C                                                                  LGM02840
      GO TO 90                                                     LGM02850
C                                                                  LGM02860
   70 CONTINUE                                                     LGM02870
C     DEFAULT EXIT                                                 LGM02880
      WRITE(6,80) MATNOB,MATOLD                                    LGM02890
   80 FORMAT(' TERMINATE.  PARAMETERS IN CHOLESKY FACTOR FILE'/    LGM02900
     1' DO NOT AGREE WITH THOSE SPECIFIED BY THE USER'/            LGM02910
     1' MATNOB = ',I8,' MATOLD = ',I8/)                            LGM02920
      STOP                                                         LGM02930
C                                                                  LGM02940
   90 CONTINUE                                                     LGM02950
C-----END OF USPECB-----------------------------------------------LGM02960
      RETURN                                                       LGM02970
      END                                                          LGM02980
C                                                                  LGM02990
C-----MATRIX-VECTOR MULTIPLY FOR REAL SPARSE SYMMETRIC MATRICES---------LGM03000
C                                                                  LGM03010
C     SUBROUTINE AMATV(W,U,SUM)                                    LGM03020
      SUBROUTINE GCMATV(W,U,SUM)                                   LGM03030
C                                                                  LGM03040
C-----------------------------------------------------------------LGM03050
      DOUBLE PRECISION   U(1),W(1),ASD(1),AD(1),SUM                LGM03060
      INTEGER  IROW(1),ICOL(1)                                     LGM03070
C-----------------------------------------------------------------LGM03080
C   SPARSE MATRIX-VECTOR MULTIPLY FOR LANCZS  U = A*W - SUM*U      LGM03090
C   SEE USPECA SUBROUTINE FOR DESCRIPTION OF THE ARRAYS            LGM03100
C   THAT DEFINE THE A-MATRIX                                       LGM03110
C-----------------------------------------------------------------LGM03120
      GO TO 3                                                      LGM03130
C     STORAGE LOCATIONS OF ARRAYS ARE PASSED TO AMATV FROM USPECA  LGM03140
      ENTRY AMATVE(ASD,AD,ICOL,IROW,N,NZL)                         LGM03150
      GO TO 4                                                      LGM03160
C-----------------------------------------------------------------LGM03170
C                                                                  LGM03180
C     COMPUTE THE DIAGONAL TERMS                                   LGM03190
```

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

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

```
   150 CONTINUE                                                    LGM04300
C                                                                  LGM04310
     4 RETURN                                                      LGM04320
C                                                                  LGM04330
C-----END OF LSOLV-------------------------------------------      LGM04340
      END                                                          LGM04350
C                                                                  LGM04360
C-----START OF USPEC FOR DIAGONAL TEST A-MATRIX------------------------LGM04370
C                                                                  LGM04380
      SUBROUTINE USPECA(N,MATNO)                                   LGM04390
C     SUBROUTINE DUSPEC(N,MATNO)                                   LGM04400
C                                                                  LGM04410
C-----------------------------------------------------------------LGM04420
      DOUBLE PRECISION  D(1000), SPACE, SHIFT                      LGM04430
      DOUBLE PRECISION  DABS, DFLOAT                               LGM04440
      REAL  EXPLAN(20)                                             LGM04450
C-----------------------------------------------------------------LGM04460
C                                                                  LGM04470
      READ(8,10) EXPLAN                                            LGM04480
   10 FORMAT(20A4)                                                 LGM04490
      READ(8,*) NOLD,NUNIF,SPACE,D(1),SHIFT                        LGM04500
      NNUNIF = NOLD - NUNIF                                        LGM04510
      WRITE(6,20) NOLD,SPACE,NNUNIF,D(1),SHIFT                     LGM04520
   20 FORMAT(/' DIAGONAL TEST A-MATRIX, SIZE = ',I4/' MOST ENTRIES ARE 'LGM04530
     1,E10.3,' UNITS APART.',I3,' ENTRIES'/' ARE IRREGULARLY SPACED. FIRLGM04540
     1ST ENTRY IS ',E10.3,' SHIFT = ',E10.3/)                     LGM04550
C                                                                  LGM04560
      IF(N.NE.NOLD) GO TO 90                                       LGM04570
C     COMPUTE THE UNIFORM PORTION OF THE SPECTRUM                  LGM04580
      DO 30 J=2,NUNIF                                              LGM04590
   30 D(J) = D(1) - DFLOAT(J-1)*SPACE                              LGM04600
      NUNIF1=NUNIF + 1                                             LGM04610
      READ(8,10)  EXPLAN                                           LGM04620
      DO 40 J=NUNIF1,N                                             LGM04630
   40 READ(8,*) D(J)                                               LGM04640
      NB = NUNIF - 2                                               LGM04650
C                                                                  LGM04660
      IF(SHIFT.EQ.0.) GO TO 60                                     LGM04670
      DO 50 J=1,N                                                  LGM04680
   50 D(J) = D(J) + SHIFT                                          LGM04690
C                                                                  LGM04700
C     PRINT OUT A-MATRIX                                           LGM04710
   60 WRITE(6,70) (D(I), I=1,10 )                                  LGM04720
      WRITE(6,80) (D(I), I = NB,N)                                 LGM04730
   70 FORMAT(/' GENERALIZED LANCZOS TEST, 1ST 10 ENTRIES OF DIAGONAL A-MLGM04740
     1ATRIX = '/(3E22.14))                                        LGM04750
   80 FORMAT(/' MIDDLE UNIFORM PORTION OF MATRIX IS NOT PRINTED OUT'/   LGM04760
     1' END OF UNIFORM PLUS NONUNIFORM SECTION = '/(3E25.16))     LGM04770
C                                                                  LGM04780
C     DIAGONAL GENERATION COMPLETE                                LGM04790
C                                                                  LGM04800
C-----------------------------------------------------------------LGM04810
C   CALL ENTRY TO MATRIX-VECTOR MULTIPLY SUBROUTINE  TO PASS      LGM04820
C   STORAGE LOCATION OF D-ARRAY AND ORDER OF A-MATRIX.            LGM04830
      CALL MVDIAE(D,N)                                            LGM04840
```

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

```
   90 DS(K) = DSQRT(D(K))                                          LGM05400
C                                                                  LGM05410
C------------------------------------------------------------------LGM05420
C    PASS STORAGE LOCATION OF THE L-FACTOR (THE DS-ARRAY) AND ORDER OF  LGM05430
C    B-MATRIX TO LSOLV SUBROUTINE.                                 LGM05440
      CALL DSOLVE(DS,N)                                            LGM05450
C------------------------------------------------------------------LGM05460
C                                                                  LGM05470
      RETURN                                                       LGM05480
  100 WRITE(6,110) NOLD,N                                          LGM05490
  110 FORMAT(' PROGRAM TERMINATES BECAUSE NOLD = ',I5,'DOES NOT EQUAL N LGM05500
     1 =',I5)                                                      LGM05510
C-----END OF USPECB SUBROUTINE FOR DIAGONAL TEST MATRICES--------------LGM05520
      STOP                                                         LGM05530
      END                                                          LGM05540
C                                                                  LGM05550
C-----MATRIX-VECTOR MULTIPLY FOR DIAGONAL TEST MATRICES---------------LGM05560
C                                                                  LGM05570
      SUBROUTINE AMATV(W,U,SUM)                                    LGM05580
C     SUBROUTINE DCMATV(W,U,SUM)                                   LGM05590
C                                                                  LGM05600
C     AMATV COMPUTES  U = (DIAGONAL MATRIX) * W - SUM * U          LGM05610
C------------------------------------------------------------------LGM05620
      DOUBLE PRECISION  W(1),U(1),D(1),SUM                         LGM05630
C------------------------------------------------------------------LGM05640
      GO TO 3                                                      LGM05650
      ENTRY MVDIAE(D,N)                                            LGM05660
      GO TO 4                                                      LGM05670
C------------------------------------------------------------------LGM05680
C                                                                  LGM05690
    3 DO 10 I=1,N                                                  LGM05700
   10 U(I)= D(I)*W(I) - SUM*U(I)                                   LGM05710
C                                                                  LGM05720
    4 RETURN                                                       LGM05730
C                                                                  LGM05740
C-----END OF DIAGONAL TEST MATRIX  MULTIPLY-------------------------LGM05750
      END                                                          LGM05760
C                                                                  LGM05770
C-----LSOLV FOR DIAGONAL MATRIX------------------------------------LGM05780
C                                                                  LGM05790
      SUBROUTINE LSOLV(W,U,ISOLV)                                  LGM05800
C     SUBROUTINE DSOLV(W,U,ISOLV)                                  LGM05810
C                                                                  LGM05820
C------------------------------------------------------------------LGM05830
      DOUBLE PRECISION U(1), W(1),  DS(1)                          LGM05840
C------------------------------------------------------------------LGM05850
      GO TO 3                                                      LGM05860
      ENTRY DSOLVE(DS,N)                                           LGM05870
      GO TO 4                                                      LGM05880
C------------------------------------------------------------------LGM05890
    3 GO TO (10,30,50,70), ISOLV                                   LGM05900
C                                                                  LGM05910
C     ISOLV = 1                                                    LGM05920
   10 CONTINUE                                                     LGM05930
      DO 20 K = 1,N                                                LGM05940
```

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

5.5 LGVAL: LGVEC: File Definitions, Sample Input Files

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

```
Sample Specification of Input/Output Files for LGVAL
-----------------------------------------------------------------
 LGVAL EXEC LANCZOS EIGENVALUE CALCULATION AX = EV*BX CASE
FI 06 TERM
FILEDEF  1 DISK &1        NHISTORY  A (RECFM F LRECL 80 BLOCK 80
FILEDEF  2 DISK &1        HISTORY   A (RECFM F LRECL 80 BLOCK 80
FILEDEF  3 DISK &1        GOODEV    A (RECFM F LRECL 80 BLOCK 80
FILEDEF  4 DISK &1        ERRINV    A (RECFM F LRECL 80 BLOCK 80
FILEDEF  5 DISK LGVAL     INPUT     A (RECFM F LRECL 80 BLOCK 80
FILEDEF  7 DISK &1        LDATA     A (RECFM F LRECL 80 BLOCK 80
FILEDEF  8 DISK &1        ADATA     A (RECFM F LRECL 80 BLOCK 80
FILEDEF 11 DISK &1        DISTINCT  A (RECFM F LRECL 80 BLOCK 80
LOAD  LGVAL    LESUB  LGMULT
-----------------------------------------------------------------



Sample Input File for LGVAL
-----------------------------------------------------------------
 LGVAL INPUT LANCZOS EIGENVALUE COMPUTATION, NO REORTHOGONALIZATION
 AX = EV*BX GENERALIZED EIGENVALUE PROBLEM
LINE 1     N     KMAX    NMEVS     MATNOA    MATNOB
         100     300       1         100        100
LINE 2   SVSEED    RHSEED      MXINIT   MXSTUR
       49302312   5731029          5   100000
LINE 3   ISTART      ISTOP
              0         1
LINE 4     IHIS      IDIST  IWRITE
              1          0       1
LINE 5   RELTOL (RELATIVE TOLERANCE IN 'COMBINING' GOODEV)
      .0000000001
LINE 6   MB(1)   MB(2)   MB(3)   MB(4)     (ORDERS OF T(1,MEV) )
           300
LINE 7   NINT      (NUMBER OF SUB-INTERVALS FOR BISEC)
             1
LINE 8   LB(1)    LB(2)   LB(3)   LB(4)   (INTERVAL LOWER BOUNDS)
           1.5
LINE 9   UB(1)    UB(2)   UB(3)   UB(4)   (INTERVAL UPPER BOUNDS)
          2100.
-----------------------------------------------------------------
```

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

```
Sample Specifications for the Input/Output Files for LGVEC
------------------------------------------------------------------------
 LGVEC EXEC TO RUN LANCZOS EIGENVECTOR PROGRAM, REAL SYMMETRIC MATRICES
FI 06 TERM
FILEDEF  2 DISK &1        HISTORY   A (RECFM F LRECL 80 BLOCK 80
FILEDEF  3 DISK &1        GOODEV    A (RECFM F LRECL 80 BLOCK 80
FILEDEF  4 DISK &1        ERRINV    A (RECFM F LRECL 80 BLOCK 80
FILEDEF  5 DISK LGVEC     INPUT     A (RECFM F LRECL 80 BLOCK 80
FILEDEF  7 DISK &1        LDATA     A (RECFM F LRECL 80 BLOCK 80
FILEDEF  8 DISK &1        ADATA     A (RECFM F LRECL 80 BLOCK 80
FILEDEF  9 DISK &1        ERREST    A (RECFM F LRECL 80 BLOCK 80
FILEDEF 10 DISK &1        BOUNDS    A (RECFM F LRECL 80 BLOCK 80
FILEDEF 11 DISK &1        TEIGVECS  A (RECFM F LRECL 80 BLOCK 80
FILEDEF 12 DISK &1        RITZVECS  A (RECFM F LRECL 80 BLOCK 80
FILEDEF 13 DISK &1        PAIGE     A (RECFM F LRECL 80 BLOCK 80
LOAD  LGVEC  LESUB  LGMULT
------------------------------------------------------------------------




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

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 $\Sigma_1 = \mathrm{diag}\{\sigma_1, ..., \sigma_n\}$ 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 $A^T A$ 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 $A u_i$ or $A^T v_i$. 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 $A^T y$ 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--------------------------------------------------   LSV00010
C  Authors:  Jane Cullum and Ralph A. Willoughby (Deceased)        LSV00020
C            Los Alamos National Laboratory                        LSV00030
C            Los Alamos, New Mexico 87544                          LSV00040
C                                                                  LSV00050
C            E-mail:  cullumj@lanl.gov                             LSV00060
C                                                                  LSV00070
C  These codes are copyrighted by the authors.  These codes        LSV00080
C  and modifications of them or portions of them are NOT to be     LSV00090
C  incorporated into any commercial codes or used for any other    LSV00100
C  commercial purposes such as consulting for other companies,     LSV00110
C  without legal agreements with the authors of these Codes.       LSV00120
C  If these Codes or portions of them are used in other scientific or LSV00130
C  engineering research works the names of the authors of these codes LSV00140
C  and appropriate references to their written work are to be      LSV00150
C  incorporated in the derivative works.                           LSV00160
C                                                                  LSV00170
C  This header is not to be removed from these codes.              LSV00180
C                                                                  LSV00190
C                                                                  LSV00200
C                                                                  LSV00210
C         REFERENCE: Cullum and Willoughby, Chapter 5             LSV00220
C         Lanczos Algorithms for Large Symmetric Eigenvalue ComputationsLSV00230
C         VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in LSV00240
C         Applied Mathematics, 2002. SIAM Publications,           LSV00250
C         Philadelphia, PA. USA                                   LSV00260
C                                                                  LSV00270
C                                                                  LSV00280
C                                                                  LSV00290
C     DOCUMENTATION FOR THE SINGLE-VECTOR                         LSV00300
C     LANCZOS SINGULAR VALUE/VECTOR PROGRAMS                      LSV00310
C     FOR REAL, RECTANGULAR MATRICES                             LSV00320
C                                                                  LSV00330
C------------------------------------------------------------------LSV00340
C                                                                  LSV00350
C     GIVEN A REAL RECTANGULAR MATRIX A OF ORDER M X N THE THREE  LSV00360
C     SETS OF FORTRAN FILES LABELLED LSVAL, LSSUB, AND LSMULT     LSV00370
C     CAN BE USED TO COMPUTE DISTINCT SINGULAR VALUES OF A IN     LSV00380
C     USER-SPECIFIED INTERVALS.                                  LSV00390
C                                                                  LSV00400
C     CORRESPONDING SINGULAR VECTORS FOR SELECTED, COMPUTED       LSV00410
C     SINGULAR VALUES CAN BE COMPUTED USING THE SETS OF FILES     LSV00420
C     LABELLED LSVEC, LSSUB AND LSMULT.                          LSV00430
C                                                                  LSV00440
C     THESE PROGRAMS USE LANCZOS TRIDIAGONALIZATION WITHOUT       LSV00450
C     REORTHOGONALIZATION ON THE ASSOCIATED REAL SYMMETRIC MATRIX LSV00460
C                                                                  LSV00470
C                               ----              ----            LSV00480
C                              |  0               A  |            LSV00490
C              B  =            |                     |            LSV00500
C                              |  A-TRANSPOSE     0  |            LSV00510
C                               ----              ----            LSV00520
```

```
C                                                                   LSV00530
C      OF ORDER M + N  TO GENERATE REAL SYMMETRIC TRIDIAGONAL        LSV00540
C      MATRICES, T(1,MEV), OF ORDER MEV.  SUBSETS OF THE EIGENVALUES OF  LSV00550
C      THESE T-MATRICES, LABELLED AS THE 'GOOD EIGENVALUES' OF T(1,MEV), LSV00560
C      ARE APPROXIMATIONS TO THE DESIRED SINGULAR VALUES OF A.      LSV00570
C      CORRESPONDING RITZ VECTORS FOR B ARE APPROXIMATIONS TO       LSV00580
C      EIGENVECTORS OF B WHICH IN TURN CONTAIN APPROXIMATIONS TO     LSV00590
C      THE DESIRED LEFT AND RIGHT SINGULAR VECTORS OF A.  THIS       LSV00600
C      PROCEDURE USES A SPECIAL STARTING VECTOR SUGGESTED BY GOLUB   LSV00610
C      AND KAHAN.  THUS, THE STARTING LANCZOS VECTOR IS EITHER OF    LSV00620
C      THE FORM (V1,0) OR (0,V2) WHERE V1 IS MX1 AND V2 IS NX1 AND   LSV00630
C      ALL SUCCEEDING LANCZOS VECTORS GENERATED ALTERNATE BETWEEN    LSV00640
C      THESE 2 FORMS.  THIS SPECIAL CHOICE OF STARTING VECTOR RESULTS  LSV00650
C      IN SIGNIFICANT GAINS IN STORAGE AND OPERATION COUNTS AND      LSV00660
C      ALSO IN CONVERGENCE RELATIVE TO A 'BRUTE FORCE' APPLICATION   LSV00670
C      OF THE REAL SYMMETRIC LANCZOS PROCEDURE DIRECTLY TO THE       LSV00680
C      MATRIX B ABOVE.  FOR MORE DETAILS SEE REFERENCE 1 BELOW.      LSV00690
C      IN THE DISCUSSIONS T(1,MEV) DENOTES THE LANCZOS T-MATRIX      LSV00700
C      OF SIZE MEV.                                                  LSV00710
C                                                                   LSV00720
C      THE IDEAS USED IN THESE PROGRAMS ARE DISCUSSED IN THE FOLLOWING  LSV00730
C      REFERENCES.                                                  LSV00740
C                                                                   LSV00750
C      1.  JANE CULLUM, RALPH A. WILLOUGHBY AND MARK LAKE, A LANCZOS  LSV00760
C          ALGORITHM FOR COMPUTING SINGULAR VALUES AND VECTORS OF LARGE  LSV00770
C          MATRICES, SIAM J. SCIENTIFIC AND STATISTICAL COMPUTING,  LSV00780
C          VOL. 4, JUNE 1983, PP. 197-215.                          LSV00790
C                                                                   LSV00800
C      2.  JANE CULLUM AND RALPH A. WILLOUGHBY, LANCZOS ALGORITHMS   LSV00810
C          FOR LARGE SYMMETRIC MATRICES,  PROGRESS IN               LSV00820
C          SCIENTIFIC COMPUTING, EDITORS, G. GOLUB, H.O. KREISS,     LSV00830
C          S. ARBARBANEL, AND R. GLOWINSKI,  BIRKHAUSER BOSTON INC.,  LSV00840
C          CAMBRIDGE, MASSACHUSETTS, 1984.                          LSV00850
C                                                                   LSV00860
C      3.  JANE CULLUM AND RALPH A. WILLOUGHBY, COMPUTING EIGENVECTORS  LSV00870
C          (AND EIGENVALUES) OF LARGE, SYMMETRIC MATRICES USING      LSV00880
C          LANCZOS TRIDIAGONALIZATION, LECTURE NOTES IN MATHEMATICS,  LSV00890
C          773, NUMERICAL ANALYSIS PROCEEDINGS, DUNDEE 1979, EDITED BY  LSV00900
C          G. A. WATSON, SPRINGER-VERLAG, (1980), BERLIN, PP.46-63.  LSV00910
C                                                                   LSV00920
C      4. IBID, LANCZOS AND THE COMPUTATION IN SPECIFIED INTERVALS OF  LSV00930
C         THE SPECTRUM OF LARGE SPARSE, REAL SYMMETRIC MATRICES, SPARSE  LSV00940
C         MATRIX PROCEEDINGS 1978, ED. I.S. DUFF AND G. W. STEWART,  LSV00950
C         SIAM, PHILADELPHIA, PP.220-255, 1979.                     LSV00960
C                                                                   LSV00970
C      5. IBID, COMPUTING EIGENVALUES OF VERY LARGE SYMMETRIC MATRICES-  LSV00980
C         AN IMPLEMENTATION OF A LANCZOS ALGORITHM WITHOUT           LSV00990
C         REORTHOGONALIZATION, J. COMPUT. PHYS. 44(1981), 329-358.   LSV01000
C                                                                   LSV01010
C                                                                   LSV01020
C-----PORTABILITY-------------------------------------------------------LSV01030
C                                                                   LSV01040
C                                                                   LSV01050
C      PROGRAMS WERE TESTED FOR PORTABILITY USING THE PFORT VERIFIER.  LSV01060
C      FOR DETAILS OF THE VERIFIER SEE FOR EXAMPLE, B. G. RYDER AND   LSV01070
```

```
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    LSV01630
C       ARE INCLUDED IN THE FORTRAN FILE LSMULT.                        LSV01640
C                                                                       LSV01650
C       IMPORTANT NOTE:                                                 LSV01660
C       THE SAMPLE MATRIX-VECTOR MULTIPLY SUBROUTINES IN LSMULT         LSV01670
C       ASSUME THAT M >= N.  THEY ALSO ASSUME THAT THE USER-SUPPLIED    LSV01680
C       INFORMATION ABOUT THE GIVEN MATRIX IS STORED ON FILE 8.         LSV01690
C       THE USER SHOULD SEE THE LSMULT PROGRAMS FOR MORE DETAILS.       LSV01700
C                                                                       LSV01710
C       SUBROUTINE SVMAT HAS THE CALLING SEQUENCE                       LSV01720
C                                                                       LSV01730
C            CALL SVMAT(W,U,SUM)                                        LSV01740
C                                                                       LSV01750
C       WHERE U AND W ARE REAL*8 VECTORS AND SUM IS A REAL*8            LSV01760
C       SCALAR.  SVMAT CALCULATES U = A*W - SUM*U FOR THE               LSV01770
C       USER-SPECIFIED A-MATRIX.  SUBROUTINE STRAN HAS THE              LSV01780
C       CALLING SEQUENCE                                                LSV01790
C                                                                       LSV01800
C            CALL STRAN(W,U,SUM)                                        LSV01810
C                                                                       LSV01820
C       STRAN CALCULATES U = (A-TRANSPOSE)*W - SUM*U FOR THE            LSV01830
C       TRANSPOSE OF THE USER-SUPPLIED A-MATRIX.  THE ARRAY AND PARAMETER LSV01840
C       INFORMATION NEEDED TO PERFORM THE MATRIX-VECTOR MULTIPLIES      LSV01850
C       IS PASSED TO THE SVMAT AND THE STRAN SUBROUTINES FROM THE       LSV01860
C       USPEC SUBROUTINE VIA ENTRIES.  ONE SET OF THE SAMPLE SVMAT      LSV01870
C       AND STRAN SUBROUTINES INCLUDED IN LSMULT COMPUTES              LSV01880
C       MATRIX-VECTOR MULTIPLIES FOR AN ARBITRARY SPARSE,               LSV01890
C       RECTANGULAR MATRIX STORED IN THE SPARSE FORMAT SPECIFIED        LSV01900
C       IN THE CORRESPONDING SAMPLE USPEC SUBROUTINE.  THE LANCZS       LSV01910
C       SUBROUTINE CALLS SVMAT AND STRAN IN THE GENERATION OF           LSV01920
C       THE LANCZOS T-MATRICES FOR THE B MATRIX.                        LSV01930
C                                                                       LSV01940
C       THE DATA FOR THE A-MATRIX IS ASSUMED TO BE ON FILE 8 AND        LSV01950
C       IN THE FOLLOWING SPARSE FORMAT:                                 LSV01960
C       NZ = NUMBER OF NONZERO ELEMENTS OF A                            LSV01970
C       ICOL(K), K = 1,N, NUMBER OF NONZEROS OF A IN COLUMN K.          LSV01980
C       IROW(K), K = 1,NZ, ROW INDEX OF A(K).                           LSV01990
C       A(K), K=1,NZ  CONTAINS THE ELEMENTS OF A BY COLUMN.             LSV02000
C                                                                       LSV02010
C                                                                       LSV02020
C       SVMATV AND STRAN ARE CALLED FROM THE SUBROUTINE LANCZS          LSV02030
C       WHICH GENERATES THE LANCZOS TRIDIAGONAL MATRICES, THE           LSV02040
C       BETA HISTORY.  SIMILARLY, THSE SUBROUTINES ARE CALLED FROM      LSV02050
C       THE CORRESPONDING SINGULAR VECTOR PROGRAM, LSVEC.               LSV02060
C       SVMAT AND STRAN ARE DECLARED AS EXTERNAL VARIABLES.             LSV02070
C       EACH IS AN ARGUMENT FOR THE LANCZS SUBROUTINE.                  LSV02080
C                                                                       LSV02090
C       USPEC, SVMAT, AND STRAN SUBROUTINES SUITABLE FOR THE            LSV02100
C       USER-SPECIFIED MATRIX MUST BE SUPPLIED BY THE USER.             LSV02110
C                                                                       LSV02120
C       THE MAIN PROGRAMS FOR THE SINGULAR VALUE AND SINGULAR VECTOR    LSV02130
C       CALCULATIONS ASSUME THAT INPUT FILE 5 CONTAINS THE ROW ORDER    LSV02140
C       M AND THE COLUMN ORDER N OF THE GIVEN A-MATRIX AND MATNO,       LSV02150
C       AN IDENTIFICATION NUMBER OF <= 8 DIGITS FOR THE GIVEN MATRIX.   LSV02160
C                                                                       LSV02170
```

```
C                                                              LSV02180
C-----MACHEP---------------------------------------------------LSV02190
C                                                              LSV02200
C                                                              LSV02210
C     MACHEP IS A MACHINE DEPENDENT PARAMETER SPECIFYING THE RELATIVE  LSV02220
C     PRECISION OF THE FLOATING POINT ARITHMETIC USED.         LSV02230
C     MACHEP = 2.2 * 10**-16 FOR DOUBLE PRECISION ARITHMETIC ON   LSV02240
C     IBM 370-3081.                                            LSV02250
C                                                              LSV02260
C     THE USER WILL HAVE TO RESET THIS PARAMETER TO            LSV02270
C     THE CORRESPONDING VALUE FOR THE MACHINE BEING USED.  NOTE THAT   LSV02280
C     IF A MACHINE WITH A MACHINE EPSILON THAT IS MUCH LARGER THAN THE  LSV02290
C     VALUE GIVEN HERE IS BEING USED, THEN THERE COULD BE      LSV02300
C     PROBLEMS WITH THE TOLERANCES.                            LSV02310
C                                                              LSV02320
C                                                              LSV02330
C-----SUBROUTINES AND FUNCTIONS USER MUST SUPPLY----------------------LSV02340
C                                                              LSV02350
C                                                              LSV02360
C     GENRAN,  FINPRO,  MASK,  USPEC,  SVMAT AND  STRAN        LSV02370
C                                                              LSV02380
C     GENRAN = COMPUTES K PSEUDO-RANDOM NUMBERS AND STORES THEM IN   LSV02390
C              THE REAL ARRAY, G.  THIS SUBROUTINE IS USED TO  LSV02400
C              GENERATE A STARTING VECTOR FOR THE LANCZOS PROCEDURE   LSV02410
C              IN THE SUBROUTINE LANCZS AND A STARTING RIGHT-HAND SIDE  LSV02420
C              FOR INVERSE ITERATION IN THE SUBROUTINE INVERR. LSV02430
C                                                              LSV02440
C              TESTS REPORTED IN THE REFERENCES USED EITHER GGL1 OR   LSV02450
C              GGL2 FROM THE IBM LIBRARY  SLMATH.              LSV02460
C              THE EXISTING CALLING SEQUENCE IS:               LSV02470
C                                                              LSV02480
C                      CALL GENRAN(IIX,G,K).                   LSV02490
C                                                              LSV02500
C              WHERE IIX =INTEGER SEED, G = REAL*4 ARRAY WHOSE LSV02510
C              DIMENSION MUST BE >= K.  K RANDOM NUMBERS ARE GENERATED  LSV02520
C              AND PLACED IN G.                                LSV02530
C                                                              LSV02540
C     FINPRO = DOUBLE PRECISION FUNCTION WHICH COMPUTES THE INNER   LSV02550
C              PRODUCT OF 2 DOUBLE PRECISION VECTORS OF DIMENSION  K.   LSV02560
C              TESTS REPORTED IN THE REFERENCES USED THE HARWELL   LSV02570
C              LIBRARY SUBROUTINE FM02AD.                      LSV02580
C              EXISTING CALLING SEQUENCE IS                    LSV02590
C                                                              LSV02600
C                      CALL FINPRO(N,V,J,W,K).                 LSV02610
C                                                              LSV02620
C              COMPUTES THE INNER PRODUCT OF DIMENSION N OF THE VECTORS LSV02630
C              V AND W.  SUCCESSIVE COMPONENTS OF V AND OF W ARE STORED LSV02640
C              AT LOCATIONS THAT ARE ,RESPECTIVELY, J AND K UNITS APART.LSV02650
C                                                              LSV02660
C     MASK = MASKS OVERFLOW AND UNDERFLOW.                     LSV02670
C            USER MUST SUPPLY OR COMMENT OUT CALL.             LSV02680
C                                                              LSV02690
C     USPEC = DIMENSIONS AND INITIALIZES ARRAYS NEEDED TO SPECIFY   LSV02700
C             USER-SUPPLIED A-MATRIX.  SEE A-MATRIX SPECIFICATION SECTIONLSV02710
C                                                              LSV02720
```

```
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          LSV03830
C                    (BETA VECTOR) AND LAST 2 LANCZOS VECTORS USED      LSV03840
C                    IN THE T-MATRIX GENERATION.                        LSV03850
C                    IF IHIS = 0 AND ISTOP = 1, FILE 1 IS NOT WRITTEN.  LSV03860
C                                                                       LSV03870
C      (2)     INPUT FILE:                                              LSV03880
C                    SAME AS FILE 1 EXCEPT THAT IT CONTAINS A           LSV03890
C                    PREVIOUSLY-GENERATED T-MATRIX (IF ANY). IF ISTART = 1, LSV03900
C                    PROGRAM ASSUMES THAT THERE IS A HISTORY FILE OF    LSV03910
C                    BETAS ON FILE 2.  THESE BETAS AND THE LAST TWO LANCZOS LSV03920
C                    VECTORS USED IN THE T-MATRIX GENERATION ARE READ IN. LSV03930
C                                                                       LSV03940
C      (3)     OUTPUT FILE:                                             LSV03950
C                    COMPUTED GOOD EIGENVALUES OF THE T-MATRICES CONSIDERED. LSV03960
C                    ALSO CONTAINS T-MULTIPLICITIES OF THESE T-EIGENVALUES AS LSV03970
C                    EIGENVALUES OF THE T-MATRIX, AND THEIR GAPS AS     LSV03980
C                    EIGENVALUES IN THE B MATRIX AND IN THE T-MATRIX.   LSV03990
C                    NOTE THAT THESE GOOD T-EIGENVALUES ARE THE COMPUTED LSV04000
C                    SINGULAR VALUES OF THE A-MATRIX AND THAT THE GAPS  LSV04010
C                    OF THESE EIGENVALUES AS EIGENVALUES OF THE B-MATRIX LSV04020
C                    ARE EQUAL TO THEIR GAPS AS SINGULAR VALUES OF A.  FILE LSV04030
C                    3 IS ALWAYS WRITTEN.                               LSV04040
C                                                                       LSV04050
C      (4)     OUTPUT FILE:                                             LSV04060
C                    ERROR ESTIMATES FOR THE ISOLATED COMPUTED SINGULAR LSV04070
C                    SINGULAR VALUES (ISOLATED GOOD EIGENVALUES OF T(1,MEV)) LSV04080
C                    THESE ARE OBTAINED USING THE SUBROUTINE INVERR. THESE LSV04090
C                    ESTIMATES USE THE LAST COMPONENTS OF THE ASSOCIATED LSV04100
C                    T-EIGENVECTORS WHICH ARE COMPUTED USING INVERSE    LSV04110
C                    ITERATION.  FILE 4 IS ALWAYS WRITTEN.              LSV04120
C                                                                       LSV04130
C                                                                       LSV04140
C      (8)     INPUT FILE:                                              LSV04150
C                    SAMPLE USPEC SUBROUTINE ASSUMES THAT THE ARRAYS    LSV04160
C                    REQUIRED TO SPECIFY THE USER'S MATRIX ARE STORED ON LSV04170
C                    FILE 8.  USERS MUST MAKE WHATEVER DEFINITIONS ARE  LSV04180
C                    APPROPRIATE FOR THEIR MATRICES.                    LSV04190
C                                                                       LSV04200
C      (9)     OUTPUT FILE:  OPTIONAL                                   LSV04210
C                    CAN BE USED TO STORE THE TRUE SINGULAR VALUES OF   LSV04220
C                    A GIVEN TEST MATRIX, WHEN THE SINGULAR VALUE PROCEDURE LSV04230
C                    IS BEING EXERCISED ON A TEST MATRIX.               LSV04240
C                                                                       LSV04250
C      (11)    OUTPUT FILE:                                             LSV04260
C                    COMPUTED DISTINCT EIGENVALUES OF T-MATRICES USED.  LSV04270
C                    ALSO CONTAINS THEIR T-MULTIPLICITIES AND T-GAPS TO LSV04280
C                    NEAREST DISTINCT T-EIGENVALUES, AND THE T-MULTIPLICITY LSV04290
C                    PATTERN OF THE GOOD AND THE SPURIOUS T-EIGENVALUES. LSV04300
C                    FILE 11 IS WRITTEN ONLY IF IDIST = 1.             LSV04310
C                                                                       LSV04320
C                                                                       LSV04330
C-----PARAMETERS SET BY THE SINGULAR VALUE PROGRAMS--------------------LSV04340
C                                                                       LSV04350
C                                                                       LSV04360
C    THESE PARAMETERS ARE SET INTERNALLY IN THE PROGRAM                LSV04370
```

```
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      LSV04930
C               OF THE NORM OF THE T-MATRIX, T(1,KMAX).                      LSV04940
C                                                                           LSV04950
C     GAPTOL = RELATIVE TOLERANCE USED IN THE SUBROUTINE ISOEV              LSV04960
C                TO DETERMINE FOR WHICH OF THE GOOD T-EIGENVALUES,           LSV04970
C                THE COMPUTED SINGULAR VALUES, ERROR ESTIMATES SHOULD        LSV04980
C                BE COMPUTED.  THE PROGRAM SETS GAPTOL = 1.D-8.              LSV04990
C                IF FOR A GIVEN 'GOOD' T-EIGENVALUE OF THE GIVEN             LSV05000
C                T-MATRIX  THE COMPUTED GAP IN THE T-MATRIX IS TOO           LSV05010
C                SMALL AND IS DUE TO A 'SPURIOUS' EIGENVALUE OF              LSV05020
C                THE T-MATRIX, THEN THE 'GOOD' T-EIGENVALUE IS ASSUMED       LSV05030
C                TO HAVE CONVERGED AND AN ERROR ESTIMATE IS NOT             LSV05040
C                COMPUTED.                                                   LSV05050
C                                                                           LSV05060
C                                                                           LSV05070
C-----USER-SPECIFIED PARAMETERS FOR SINGULAR VALUE PROGRAMS------------LSV05080
C                                                                           LSV05090
C                                                                           LSV05100
C     RELTOL = RELATIVE TOLERANCE USED IN 'COMBINING' COMPUTED              LSV05110
C                EIGENVALUES OF T(1,MEV) PRIOR TO COMPUTING ERROR            LSV05120
C                ESTIMATES.                                                  LSV05130
C                                                                           LSV05140
C     THE LUMPING OF T-EIGENVALUES OCCURS IN SUBROUTINE LUMP.               LSV05150
C     LUMPING IS NECESSARY BECAUSE IT IS IMPOSSIBLE TO ACCURATELY           LSV05160
C     PREDICT THE ACCURACY OF THE BISEC SUBROUTINE.  LUMP 'COMBINES'        LSV05170
C     T-EIGENVALUES THAT HAVE SLIPPED BY THE TOLERANCE THAT WAS USED        LSV05180
C     IN THE T-MULTIPLICITY TESTS.  IN PARTICULAR IF FOR SOME J,            LSV05190
C                                                                           LSV05200
C     |EVALUE(J)-EVALUE(J-1)| < DMAX1(RELTOL*|EVALUE(J)|,SCALE2*MULTOL)     LSV05210
C                                                                           LSV05220
C     THEN THESE T-EIGENVALUES ARE 'COMBINED'.  MULTOL IS THE TOLERANCE LSV05230
C     THAT WAS USED IN THE T-MULTIPLICITY TEST IN BISEC.  SEE THE HEADERLSV05240
C     ON THE LUMP SUBROUTINE FOR MORE DETAILS.                             LSV05250
C                                                                           LSV05260
C     RELTOL IS SET TO 1.D-10.                                             LSV05270
C                                                                           LSV05280
C     MXINIT = MAXIMUM NUMBER OF INVERSE ITERATIONS ALLOWED IN             LSV05290
C                SUBROUTINE INVERR FOR EACH ISOLATED GOOD T-EIGENVALUE       LSV05300
C                CONSIDERED.  TYPICALLY ONLY ONE IS REQUIRED.               LSV05310
C                                                                           LSV05320
C     SEEDS FOR RANDOM NUMBER GENERATORS = INTEGER*4 SCALARS.              LSV05330
C                                                                           LSV05340
C               (1) SVSEED = SEED FOR STARTING VECTOR USED IN              LSV05350
C                    T-MATRIX GENERATION IN LANCZS SUBROUTINE              LSV05360
C                                                                           LSV05370
C               (2) RHSEED = SEED FOR RIGHT-HAND SIDE USED IN              LSV05380
C                    INVERSE ITERATION COMPUTATIONS IN INVERR.             LSV05390
C                                                                           LSV05400
C     BISEC DATA                                                           LSV05410
C                                                                           LSV05420
C     (1) NINT  =  NUMBER OF SUBINTERVALS ON WHICH SINGULAR VALUES         LSV05430
C                ARE TO BE COMPUTED.                                        LSV05440
C                                                                           LSV05450
C     (2) LB(J) = (J = 1,NINT)  =  LEFT END POINTS OF THESE INTERVALS.     LSV05460
C                MUST BE PROVIDED IN INCREASING ORDER.  THAT IS,           LSV05470
```

```
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    LSV07680
C                                                                        LSV07690
C           2.5*KMAX + MAX(KMAX,M) + MAX(KMAX,N) + .5* MAX(2*KMAX,M,N)   LSV07700
C                                                                        LSV07710
C       PLUS WHATEVER IS NEEDED TO GENERATE A*X FOR THE GIVEN MATRIX A.   LSV07720
C       THE ARRAYS BETA, VS AND MP CONSUME 2.5*KMAX*8 BYTES.             LSV07730
C       THE ARRAY V1 CONSUMES MAXIMUM(KMAX+1,M)*8 BYTES,  THE            LSV07740
C       ARRAY V2 CONSUMES MAXIMUM(KMAX,N)*8 BYTES, WITH THE              LSV07750
C       QUALIFICATION STATED ABOVE WHERE V2 IS DEFINED.  THE G-ARRAY     LSV07760
C       CONSUMES .5*MAX(2*KMAX,M,N)*8 BYTES.                             LSV07770
C                                                                        LSV07780
C                                                                        LSV07790
C------------------------------------------------------------------------LSV07800
C                                                                        LSV07810
C       COMMENTS FOR SINGULAR VECTOR COMPUTATIONS                        LSV07820
C                                                                        LSV07830
C------------------------------------------------------------------------LSV07840
C                                                                        LSV07850
C                                                                        LSV07860
C       THE SINGULAR VALUES WHOSE SINGULAR VECTORS ARE TO BE COMPUTED    LSV07870
C       MUST HAVE BEEN COMPUTED USING THE CORRESPONDING LANCZOS          LSV07880
C       SINGULAR VALUE PROGRAMS FOR REAL RECTANGULAR MATRICES BECAUSE    LSV07890
C       THESE SINGULAR VECTOR PROGRAMS USE THE SAME FAMILY OF LANCZOS    LSV07900
C       TRIDIAGONAL MATRICES THAT WAS USED IN THE CORRESPONDING          LSV07910
C       SINGULAR VALUE COMPUTATIONS.                                     LSV07920
C                                                                        LSV07930
C       THESE PROGRAMS ASSUME THAT THE SINGULAR VALUES SUPPLIED TO IT    LSV07940
C       HAVE BEEN COMPUTED ACCURATELY, AS MEASURED BY THE                LSV07950
C       ERROR ESTIMATES COMPUTED IN THE CORRESPONDING LANCZOS           LSV07960
C       SINGULAR VALUE COMPUTATIONS, ALTHOUGH THESE ESTIMATES            LSV07970
C       ARE TYPICALLY CONSERVATIVE.   THE SINGULAR VALUES SUPPLIED       LSV07980
C       ARE STORED IN THE ARRAY GOODSV(J), J=1,NGOOD.                    LSV07990
C                                                                        LSV08000
C       FOR EACH GOODSV(J), THE SUBROUTINE STURMI COMPUTES THE           LSV08010
C       SMALLEST SIZE LANCZOS TRIDIAGONAL MATRIX, T(1,M1(J)), FOR        LSV08020
C       WHICH  GOODSV(J) IS A T-EIGENVALUE TO WITHIN A SPECIFIED         LSV08030
C       TOLERANCE.  IT ALSO ATTEMPTS TO COMPUTE THE SIZE, M2(J),         LSV08040
C       BY WHICH THE GIVEN SINGULAR VALUE BECOMES A DOUBLE               LSV08050
C       T-EIGENVALUE TO WITHIN THE GIVEN TOLERANCE.  THESE SIZES ARE     LSV08060
C       USED TO DETERMINE 1ST GUESSES AT SIZES FOR THE T-EIGENVECTORS    LSV08070
C       THAT WILL BE USED IN THE SINGULAR VECTOR COMPUTATIONS.           LSV08080
C       SUBROUTINE INVERM SUCCESSIVELY COMPUTES CORRESPONDING            LSV08090
C       T-EIGENVECTORS OF ENLARGED T-MATRICES UNTIL A SUITABLE           LSV08100
C       SIZE T-MATRIX IS DETERMINED FOR EACH J.  UP TO 10 SUCH           LSV08110
C       T-EIGENVECTOR COMPUTATIONS ARE ALLOWED FOR EACH SINGULAR         LSV08120
C       VALUE SUPPLIED.                                                  LSV08130
C                                                                        LSV08140
C       AFTER APPROPRIATE T-EIGENVECTORS HAVE BEEN COMPUTED,             LSV08150
C       RITZ VECTORS FOR THE MATRIX B CORRESPONDING TO THESE             LSV08160
C       T-EIGENVECTORS ARE THEN COMPUTED.  SECTIONS OF THESE             LSV08170
C       RITZ VECTORS ARE THEN TAKEN AS APPROXIMATE LEFT AND              LSV08180
C       RIGHT SINGULAR VECTORS CORRESPONDING TO THE GIVEN                LSV08190
C       SINGULAR VALUES GOODSV(J),  J = 1,...,NGOOD.                     LSV08200
C                                                                        LSV08210
C       THIS IMPLEMENTATION FIRST COMPUTES ALL OF THE RELEVANT           LSV08220
```

```
C       T-EIGENVECTORS OF THE SYMMETRIC TRIDIAGONAL MATRICES          LSV08230
C       IN THE VECTOR, TVEC.                                          LSV08240
C                                                                     LSV08250
C       THEN, AS EACH OF THE LANCZOS VECTORS IS REGENERATED, ALL      LSV08260
C       OF THE B-MATRIX RITZ VECTORS CORRESPONDING TO THESE           LSV08270
C       T-EIGENVECTORS ARE UPDATED USING THE CURRENTLY-GENERATED      LSV08280
C       LANCZOS VECTOR.  LANCZOS VECTORS ARE GENERATED (NOTE          LSV08290
C       THAT THEY ARE NOT BEING KEPT), UNTIL ENOUGH HAVE              LSV08300
C       BEEN GENERATED TO MAP THE LONGEST T-EIGENVECTOR INTO ITS      LSV08310
C       CORRESPONDING B-MATRIX  RITZ VECTOR.  THE ARRAY RITVEC        LSV08320
C       CONTAINS THE SUCCESSIVE RITZ VECTORS WHICH ARE THEN           LSV08330
C       SPLIT INTO APPROXIMATIONS TO THE LEFT AND RIGHT SINGULAR      LSV08340
C       VECTORS OF THE USER-SUPPLIED MATRIX A.                        LSV08350
C                                                                     LSV08360
C                                                                     LSV08370
C-----PARAMETER CONTROLS FOR SINGULAR VECTOR PROGRAMS-----------------LSV08380
C                                                                     LSV08390
C                                                                     LSV08400
C       PARAMETER CONTROLS ARE INTRODUCED TO ALLOW SEGMENTATION OF THE LSV08410
C       SINGULAR VECTOR COMPUTATIONS AND TO ALLOW VARIOUS COMBINATIONS LSV08420
C       OF READ/WRITES.                                               LSV08430
C                                                                     LSV08440
C       THE FLAG MBOUND ALLOWS THE USER TO DETERMINE A FIRST GUESS ON THE LSV08450
C       STORAGE THAT WILL BE REQUIRED BY THE T-EIGENVECTORS FOR THE   LSV08460
C       SINGULAR VALUES WHOSE SINGULAR VECTORS ARE TO BE COMPUTED.    LSV08470
C       THIS CAN BE USED TO ESTIMATE THE REQUIRED SIZE OF THE TVEC ARRAY. LSV08480
C                                                                     LSV08490
C       MBOUND = (0,1) MEANS                                          LSV08500
C                                                                     LSV08510
C               (0)  PROGRAM COMPUTES FIRST GUESSES AT THE SIZES      LSV08520
C                    OF THE T-MATRICES REQUIRED BY EACH OF THE        LSV08530
C                    SINGULAR VALUES SUPPLIED AND THEN CONTINUES      LSV08540
C                    WITH THE CORRESPONDING T-EIGENVECTOR             LSV08550
C                    COMPUTATIONS.                                    LSV08560
C                                                                     LSV08570
C               (1)  PROGRAM COMPUTES FIRST GUESSES AT THE SIZES      LSV08580
C                    OF THE T-MATRICES REQUIRED BY EACH OF THE        LSV08590
C                    SINGULAR VALUES SUPPLIED, STORES THESE IN FILE   LSV08600
C                    10 AND THEN TERMINATES.  THE USER CAN USE THESE  LSV08610
C                    SIZES TO ESTIMATE THE SIZE TVEC ARRAY NEEDED     LSV08620
C                    FOR THE DESIRED T-EIGENVECTOR COMPUTATIONS.      LSV08630
C                                                                     LSV08640
C       THE FLAGS NTVCON, TVSTOP, LVCONT, AND ERCONT CONTROL THE STOPPING LSV08650
C       CRITERIA FOR INTERMEDIATE POINTS IN THE LANCZOS PROCEDURE.  THEY LSV08660
C       TERMINATE THE PROCEDURE IF VARIOUS SPECIFIED QUANTITIES COULD LSV08670
C       NOT BE COMPUTED AS DESIRED.                                   LSV08680
C                                                                     LSV08690
C       NTVCON = (0,1) MEANS                                          LSV08700
C                                                                     LSV08710
C               (0)  IF THE ESTIMATED STORAGE FOR THE T-EIGENVECTORS  LSV08720
C                    EXCEEDS THE USER-SPECIFIED DIMENSION OF THE      LSV08730
C                    TVEC ARRAY PROGRAM DOES NOT CONTINUE WITH THE    LSV08740
C                    T-EIGENVECTOR COMPUTATIONS. TERMINATION OCCURS.  LSV08750
C                                                                     LSV08760
C               (1)  CONTINUE WITH THE T-EIGENVECTOR COMPUTATIONS     LSV08770
```

```
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                                                              LSV09330
C     FILE 6 WAS USED AS THE INTERACTIVE TERMINAL OUTPUT FILE.  LSV09340
C     THIS FILE PROVIDES A RUNNING ACCOUNT OF THE PROGRESS OF THE LSV09350
C     COMPUTATIONS.  ADDITIONAL PRINTOUT IS GENERATED WHEN       LSV09360
C     THE FLAG IWRITE = 1.                                       LSV09370
C                                                              LSV09380
C                                                              LSV09390
C DESCRIPTION OF OTHER I/O FILES                               LSV09400
C                                                              LSV09410
C FILE (K)     CONTAINS:                                       LSV09420
C                                                              LSV09430
C     (2)      INPUT FILE:                                     LSV09440
C              PREVIOUSLY-GENERATED T-MATRICES (BETA ARRAY)    LSV09450
C              AND THE FINAL TWO LANCZOS VECTORS USED ON THAT  LSV09460
C              COMPUTATION.  THIS PROGRAM ALLOWS ENLARGEMENT   LSV09470
C              OF ANY T-MATRICES PROVIDED ON FILE 2.           LSV09480
C                                                              LSV09490
C     (3)      INPUT FILE:                                     LSV09500
C              THE SINGULAR VALUES FOR WHICH CORRESPONDING     LSV09510
C              SINGULAR VECTORS ARE REQUESTED.  FILE 3 ALSO    LSV09520
C              CONTAINS THE T-MULTIPLICITIES OF THESE SINGULAR LSV09530
C              VALUES (AS T-EIGENVALUES) AND THEIR COMPUTED GAPS LSV09540
C              BOTH THE T-MATRICES AND IN THE USER-SUPPLIED MATRIX. LSV09550
C              THIS FILE IS CREATED IN THE LANCZOS SINGULAR    LSV09560
C              VALUE COMPUTATIONS.                             LSV09570
C                                                              LSV09580
C     (4)      INPUT FILE:                                     LSV09590
C              ERROR ESTIMATES FOR THE ISOLATED SINGULAR VALUES LSV09600
C              OF FILE 3.  THIS FILE IS CREATED DURING THE LANCZOS LSV09610
C              SINGULAR VALUE COMPUTATIONS.                    LSV09620
C                                                              LSV09630
C     (8)      INPUT FILE:                                     LSV09640
C              USPEC SUBROUTINE ASSUMES THAT THE USER-         LSV09650
C              SUPPLIED MATRIX IS ON FILE 8.                   LSV09660
C                                                              LSV09670
C     (9)      OUTPUT FILE:                                    LSV09680
C              ERROR ESTIMATES FOR THE COMPUTED RITZ VECTORS CONSIDERED LSV09690
C              AS EIGENVECTORS OF THE B-MATRIX. THESE ESTIMATES LSV09700
C              ARE OF THE FORM                                 LSV09710
C                  BERROR = || B*RITVEC - SVAL*RITVEC ||       LSV09720
C              WHERE B DENOTES THE M+N ORDER SYMMETRIC MATRIX  LSV09730
C              ASSOCIATED WITH THE USER-SUPPLIED MATRIX A, SVAL LSV09740
C              DENOTES THE SINGULAR VALUE BEING CONSIDERED AND LSV09750
C              RITVEC DENOTES THE ASSOCIATED COMPUTED RITZ VECTOR. LSV09760
C                                                              LSV09770
C     (10)     OUTPUT FILE:                                    LSV09780
C              GUESSES AT APPROPRIATE SIZE T-MATRICES FOR THE  LSV09790
C              T-EIGENVECTORS FOR EACH SUPPLIED SINGULAR VALUE LSV09800
C              IN THE ARRAY GOODSV(J),  J = 1,...,NGOOD.       LSV09810
C                                                              LSV09820
C     (11)     OUTPUT FILE:                                    LSV09830
C              COMPUTED T-EIGENVECTORS CORRESPONDING TO SINGULAR LSV09840
C              VALUES IN THE GOODSV ARRAY.  NOTE THAT IT IS POSSIBLE LSV09850
C              IN CERTAIN SITUATIONS THAT FOR SOME SINGULAR VALUES LSV09860
C              SUPPLIED IN THE GOODSV ARRAY A T-EIGENVECTOR WILL LSV09870
```

```
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                                                                       LSV10980
C     GOODSV(J),  = REAL*8 ARRAYS EACH OF DIMENSION AT LEAST NGOOD.      LSV10990
C     SVNEW(J)      CONTAIN THE SINGULAR VALUES FOR WHICH                LSV11000
C                   SINGULAR VECTORS ARE REQUESTED.  SINGULAR VALUES     LSV11010
C                   IN GOODSV ARE READ IN FROM FILE 3.                   LSV11020
C                                                                       LSV11030
C     BMINGP(J), = REAL*4 ARRAYS OF DIMENSION AT LEAST NGOOD.            LSV11040
C     TMINGP(J)    CONTAIN, RESPECTIVELY, THE MINIMAL GAPS FOR           LSV11050
C                  CORRESPONDING SINGULAR VALUES IN GOODSV ARRAY IN      LSV11060
C                  B-MATRIX AND IN T-MATRIX.                             LSV11070
C                                                                       LSV11080
C     TERR(J), ERR(J),     = REAL*4 ARRAYS (EXCEPT TLAST WHICH IS        LSV11090
C     ERRDGP(J), TLAST(J)    REAL*8).  EACH MUST BE OF DIMENSION         LSV11100
C     RNORM(J), TBETA(J)     AT LEAST NGOOD.  USED TO STORE QUANTITIES   LSV11110
C                            GENERATED DURING THE COMPUTATIONS FOR       LSV11120
C                            LATER PRINTOUT.                             LSV11130
C                                                                       LSV11140
C     G(J)     = REAL*4 ARRAY WHOSE DIMENSION MUST BE AT LEAST           LSV11150
C                MAX(KMAX,M,N).  USED IN SUBROUTINE GENRAN TO HOLD       LSV11160
C                RANDOM NUMBERS NEEDED FOR THE LANCZOS VECTOR            LSV11170
C                REGENERATION AND FOR THE INVERSE ITERATION             LSV11180
C                COMPUTATIONS IN THE SUBROUTINE INVERM.                  LSV11190
C                                                                       LSV11200
C     MP(J) = INTEGER*4 ARRAY WHOSE DIMENSION IS AT LEAST NGOOD.         LSV11210
C             INITIALLY CONTAINS THE T-MULTIPLICITY OF THE SINGULAR      LSV11220
C             VALUE GOODSV(J) AS AN EIGENVALUE OF THE T-MATRIX.          LSV11230
C             USED TO FLAG SINGULAR VALUES FOR WHICH NO T-EIGENVECTOR    LSV11240
C             OR NO RITZ VECTOR IS TO BE COMPUTED.                       LSV11250
C                                                                       LSV11260
C     MA(J)   = INTEGER*4 ARRAYS EACH OF WHOSE DIMENSIONS                LSV11270
C               IS AT LEAST NGOOD.  USED IN DETERMINING                  LSV11280
C               AN APPROPRIATE T-MATRIX FOR EACH SINGULAR VALUE          LSV11290
C               IN GOODSV ARRAY.                                         LSV11300
C                                                                       LSV11310
C     MINT(J),MFIN(J) = INTEGER*4 ARRAYS WHOSE DIMENSIONS MUST BE AT     LSV11320
C                       LEAST NGOOD.  USED TO POINT TO THE BEGINNINGS    LSV11330
C                       AND THE ENDS OF THE COMPUTED EIGENVECTOR         LSV11340
C                       OF THE T-MATRIX, T(1,|MA(J)|).                   LSV11350
C                                                                       LSV11360
C     IDELTA(J) = INTEGER*4 ARRAY WHOSE DIMENSION MUST BE AT             LSV11370
C                 LEAST NGOOD.  CONTAINS INCREMENTS USED IN LOOPS        LSV11380
C                 ON APPROPRIATE SIZE T-MATRIX FOR THE T-EIGENVECTOR     LSV11390
C                 COMPUTATIONS.                                          LSV11400
C                                                                       LSV11410
C                                                                       LSV11420
C-----SUBROUTINES INCLUDED FOR THE SINGULAR VECTOR COMPUTATIONS---------LSV11430
C                                                                       LSV11440
C                                                                       LSV11450
C     STURMI = FOR EACH GIVEN SINGULAR VALUE GOODSV(J) DETERMINES        LSV11460
C              THE SMALLEST SIZE T-MATRIX FOR WHICH GOODSV(J) IS         LSV11470
C              A T-EIGENVALUE (TO WITHIN A GIVEN TOLERANCE) AND IF       LSV11480
C              POSSIBLE THE SMALLEST SIZE T-MATRIX FOR WHICH             LSV11490
C              IT IS A DOUBLE T-EIGENVALUE (TO WITHIN THE SAME           LSV11500
C              TOLERANCE).  THE SIZE T-MATRIX USED IN THE                LSV11510
C              T-EIGENVECTOR COMPUTATIONS IS THEN DETERMINED BY          LSV11520
```

```
C                     STARTING WITH AN INITIAL GUESS BASED ON THE          LSV11530
C                     INFORMATION FROM STURMI, AND THEN LOOPING ON THE      LSV11540
C                     SIZE OF THE T-EIGENVECTOR COMPUTATIONS.               LSV11550
C                                                                          LSV11560
C      LBISEC = RECOMPUTES THE VALUE OF THE GIVEN SINGULAR VALUE            LSV11570
C                     AT THE T-SIZE SPECIFIED FOR THE T-EIGENVECTOR         LSV11580
C                     COMPUTATION.  LBISEC IS A SIMPLIFICATION OF THE       LSV11590
C                     BISEC SUBROUTINE USED IN THE LANCZOS SINGULAR         LSV11600
C                     VALUE COMPUTATIONS.                                   LSV11610
C                                                                          LSV11620
C      INVERM = FOR THE T-SIZES CONSIDERED BY THE PROGRAM COMPUTES          LSV11630
C                     THE CORRESPONDING EIGENVECTORS OF THESE T-MATRICES    LSV11640
C                     CORRESPONDING TO THE USER-SUPPLIED SINGULAR VALUES    LSV11650
C                     IN THE GOODSV ARRAY.                                  LSV11660
C                                                                          LSV11670
C      LANCZS AND TNORM  SUBROUTINES ARE ALSO USED HERE AS WELL AS         LSV11680
C      IN THE CORRESPONDING SINGULAR VALUE COMPUTATIONS.                    LSV11690
C                                                                          LSV11700
C                                                                          LSV11710
C---------------------------------------------------------------------LSV11720
```

6.3   LSVAL: Main Program, Eigenvalue Computations

```
C-----LSVAL   (SINGULAR VALUES OF REAL, RECTANGULAR MATRICES------------LSV00010
C  Authors:   Jane Cullum and Ralph A. Willoughby (Deceased)          LSV00020
C             Los Alamos National Laboratory                          LSV00030
C             Los Alamos, New Mexico 87544                            LSV00040
C                                                                     LSV00050
C             E-mail:  cullumj@lanl.gov                               LSV00060
C                                                                     LSV00070
C  These codes are copyrighted by the authors.  These codes           LSV00080
C  and modifications of them or portions of them are NOT to be        LSV00090
C  incorporated into any commercial codes or used for any other       LSV00100
C  commercial purposes such as consulting for other companies,        LSV00110
C  without legal agreements with the authors of these Codes.          LSV00120
C  If these Codes or portions of them are used in other scientific or LSV00130
C  engineering research works the names of the authors of these codes LSV00140
C  and appropriate references to their written work are to be         LSV00150
C  incorporated in the derivative works.                              LSV00160
C                                                                     LSV00170
C  This header is not to be removed from these codes.                 LSV00180
C                                                                     LSV00190
C         REFERENCE: Cullum and Willoughby, Chapter 5                 LSV00191
C         Lanczos Algorithms for Large Symmetric Eigenvalue ComputationsLSV00192
C         VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in LSV00193
C         Applied Mathematics, 2002. SIAM Publications,               LSV00194
C         Philadelphia, PA. USA                                       LSV00195
C                                                                     LSV00196
C                                                                     LSV00197
C                                                                     LSV00200
C     CONTAINS MAIN PROGRAM FOR COMPUTING DISTINCT SINGULAR VALUES OF LSV00210
C     A REAL M X N MATRIX USING LANCZOS TRIDIAGONALIZATION WITHOUT     LSV00220
C     REORTHOGONALIZATION AND WITH SPECIAL STARTING VECTORS.          LSV00230
C                                                                     LSV00240
C     FOR A GIVEN REAL MATRIX A OF ORDER M X N THE LANCZOS RECURSION   LSV00250
C     IS APPLIED TO THE ASSOCIATED REAL SYMMETRIC MATRIX B OF ORDER   LSV00260
C     MN = M + N                                                      LSV00270
C                                                                     LSV00280
C                                    ----              ----           LSV00290
C                                   |  0               A  |           LSV00300
C                       B  =        |                     |           LSV00310
C                                   |  A-TRANSPOSE     0  |           LSV00320
C                                    ----              ----           LSV00330
C                                                                     LSV00340
C     USING SPECIAL STARTING VECTORS.  PLEASE NOTE: ONLY EVEN ORDER   LSV00350
C     LANCZOS TRIDIAGONAL MATRICES AND ONLY NONNEGATIVE SUBINTERVALS  LSV00360
C     ARE PERMISSIBLE.                                                LSV00370
C                                                                     LSV00380
C     PFORT VERIFIER IDENTIFIED THE FOLLOWING NONPORTABLE             LSV00390
C     CONSTRUCTIONS                                                   LSV00400
C                                                                     LSV00410
C     1.  DATA/MACHEP/ STATEMENT                                      LSV00420
C     2.  ALL READ(5,*) STATEMENTS (FREE FORMAT)                      LSV00430
C     3.  FORMAT(20A4) USED WITH EXPLANATORY HEADER EXPLAN.           LSV00440
```

```
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
      IPAR0 = 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/)                                             LSV02100
C                                                              LSV02110
     WRITE(6,50) ISTART,ISTOP                                  LSV02120
  50 FORMAT(/2X,'ISTART',3X,'ISTOP'/2I8/)                      LSV02130
C                                                              LSV02140
     WRITE(6,60) IHIS,IDIST,IWRITE,IPAR                        LSV02150
  60 FORMAT(/4X,'IHIS',3X,'IDIST',2X,'IWRITE',4X,'IPAR'/4I8/)  LSV02160
C                                                              LSV02170
     WRITE(6,70) SVSEED,RHSEED                                 LSV02180
  70 FORMAT(/' SEEDS FOR RANDOM NUMBER GENERATOR'//            LSV02190
   1 4X,'LANCZS SEED',4X,'INVERR SEED'/2I15/)                  LSV02200
C                                                              LSV02210
     WRITE(6,80) (NMEV(J), J=1,NMEVS)                          LSV02220
  80 FORMAT(/' SIZES OF T-MATRICES TO BE CONSIDERED'/(6I12))   LSV02230
C                                                              LSV02240
     WRITE(6,90) RELTOL,GAPTOL,BTOL                            LSV02250
  90 FORMAT(/' RELATIVE TOLERANCE USED TO COMBINE COMPUTED T-EIGENVALUELSV02260
   1S'/E15.3//' RELATIVE GAP TOLERANCES USED IN INVERSE ITERATION'/ LSV02270
   1E15.3/' RELATIVE TOLERANCE FOR CHECK ON SIZE OF BETAS'/E15.3/) LSV02280
C                                                              LSV02290
     WRITE(6,100) (J,LB(J),UB(J), J=1,NINT)                    LSV02300
 100 FORMAT(/' BISEC WILL BE USED ON THE FOLLOWING INTERVALS'/ LSV02310
   1 (I6,2E20.6)/)                                             LSV02320
C                                                              LSV02330
     IF (ISTART.EQ.0.AND.IPAR.EQ.1) WRITE(6,110)               LSV02340
     IF (ISTART.EQ.0.AND.IPAR.EQ.2) WRITE(6,120)               LSV02350
 110 FORMAT(/' STARTING VECTOR IS OF FORM (0,V2)'/)            LSV02360
 120 FORMAT(/' STARTING VECTOR IS OF FORM (V1,0)'/)            LSV02370
C                                                              LSV02380
     IF (ISTART.EQ.0) GO TO 170                                LSV02390
C                                                              LSV02400
C    READ IN BETA HISTORY FROM FILE 2                          LSV02410
C                                                              LSV02420
     READ(2,130)MOLD,MO,NO,IPARO,IPAR,SVSOLD,MATOLD            LSV02430
 130 FORMAT(3I6,2I3,I12,I8)                                    LSV02440
C                                                              LSV02450
     IF (KMAX.LT.MOLD) KMAX = MOLD                             LSV02460
     KMAX1 = KMAX + 1                                          LSV02470
C                                                              LSV02480
C    CHECK THAT M, N, MATRIX ID MATNO, AND RANDOM SEED SVSEED  LSV02490
C    AGREE WITH THOSE IN THE HISTORY FILE.  IF NOT PROCEDURE STOPS. LSV02500
C                                                              LSV02510
     ITEMP = (MO-M)**2+(NO-N)**2+(MATNO-MATOLD)**2+(SVSEED-SVSOLD)**2 LSV02520
C                                                              LSV02530
     IF (ITEMP.EQ.0) GO TO 150                                 LSV02540
C                                                              LSV02550
     WRITE(6,140)                                              LSV02560
 140 FORMAT(' PROGRAM TERMINATES'/    ' READ FROM FILE 2 CORRESPONDS TOLSV02570
   1 DIFFERENT MATRIX THAN MATRIX SPECIFIED'/)                 LSV02580
     GO TO 690                                                 LSV02590
C                                                              LSV02600
 150 CONTINUE                                                  LSV02610
     MOLD1 = MOLD+1                                            LSV02620
C                                                              LSV02630
     READ(2,160)(BETA(J), J=1,MOLD1)                           LSV02640
```

```
  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-----------------------------------------------------------------LSV03200
C                                                                  LSV03210
      TTOL = EPSM*TKMAX                                            LSV03220
C                                                                  LSV03230
C     LOOP ON THE SIZE OF THE T-MATRIX                             LSV03240
C                                                                  LSV03250
  210 CONTINUE                                                     LSV03260
      MMB = MMB + 1                                                LSV03270
C     NOTE THAT ONLY EVEN ORDER T-SIZES ARE PERMISSIBLE.          LSV03280
      MEV = NMEV(MMB)                                              LSV03290
C     IS MEV TOO LARGE ?                                           LSV03300
      IF(MEV.LE.KMAX) GO TO 230                                    LSV03310
      WRITE(6,220) MMB, MEV, KMAX                                  LSV03320
  220 FORMAT(/' TERMINATE PRIOR TO CONSIDERING THE',I6,'TH T-MATRIX'/  LSV03330
     1' BECAUSE THE SIZE REQUESTED',I6,' IS GREATER THAN THE MAXIMUM SIZLSV03340
     1E ALLOWED',I6/)                                              LSV03350
      GO TO 570                                                    LSV03360
C                                                                  LSV03370
  230 MP1 = MEV + 1                                                LSV03380
      BETAM = BETA(MP1)                                            LSV03390
C                                                                  LSV03400
      IF (IB.GE.0) GO TO 240                                       LSV03410
C                                                                  LSV03420
      T0 = BTOL                                                    LSV03430
C                                                                  LSV03440
C-----------------------------------------------------------------LSV03450
C                                                                  LSV03460
      CALL TNORM(BETA,T0,T1,MEV,IBMEV)                             LSV03470
C                                                                  LSV03480
C-----------------------------------------------------------------LSV03490
C                                                                  LSV03500
      TEMP = T0/TKMAX                                              LSV03510
      IBMEV = IABS(IBMEV)                                          LSV03520
      IF (TEMP.GE.BTOL) GO TO 240                                  LSV03530
      IBMEV = -IBMEV                                               LSV03540
      GO TO 630                                                    LSV03550
C                                                                  LSV03560
  240 CONTINUE                                                     LSV03570
      IC = MXSTUR-ICT                                              LSV03580
C                                                                  LSV03590
C-----------------------------------------------------------------LSV03600
C     BISEC LOOP. THE SUBROUTINE BISEC INCORPORATES DIRECTLY THE   LSV03610
C     T-MULTIPLICITY AND SPURIOUS TESTS. T-EIGENVALUES WILL BE     LSV03620
C     CALCULATED BY BISEC SEQUENTIALLY ON INTERVALS                LSV03630
C     (LB(J),UB(J)), J = 1,NINT).                                  LSV03640
C                                                                  LSV03650
C     ON RETURN FROM BISEC                                         LSV03660
C     NDIS = NUMBER OF DISTINCT EIGENVALUES OF T(1,MEV) ON UNION   LSV03670
C            OF THE (LB,UB) INTERVALS                              LSV03680
C     VS = DISTINCT T-EIGENVALUES IN ALGEBRAICALLY INCREASING ORDER LSV03690
C     MP = T-MULTIPLICITIES OF THE T-EIGENVALUES STORED IN VS      LSV03700
C     MP(I) = (0,1,MI), MI>1, I=1,NDIS  MEANS:                     LSV03710
C         (0)  VS(I) IS SPURIOUS                                   LSV03720
C         (1)  VS(I) IS T-SIMPLE AND GOOD                          LSV03730
C         (MI) VS(I) IS T-MULTIPLE AND IS THEREFORE NOT ONLY GOOD BUT LSV03740
```

```
C             ALSO A CONVERGED GOOD T-EIGENVALUE.                   LSV03750
C                                                                   LSV03760
C                                                                   LSV03770
      CALL BISEC(BETA,V1,V2,VS,LB,UB,EPSM,TTOL,MP,NINT,             LSV03780
     1 MEV,NDIS,IC,IWRITE)                                          LSV03790
C                                                                   LSV03800
C-------------------------------------------------------------------LSV03810
C                                                                   LSV03820
      IF (NDIS.EQ.0) GO TO 650                                      LSV03830
C                                                                   LSV03840
C     COMPUTE THE TOTAL NUMBER OF STURM SEQUENCES USED TO DATE      LSV03850
C     COMPUTE THE BISEC CONVERGENCE AND T-MULTIPLICITY TOLERANCES USED. LSV03860
C     COMPUTE THE CONVERGENCE TOLERANCE FOR T-EIGENVALUES.          LSV03870
      ICT = ICT + IC                                                LSV03880
      TEMP = DFLOAT(MEV+1000)                                       LSV03890
      MULTOL = TEMP*TTOL                                            LSV03900
      TEMP = DSQRT(TEMP)                                            LSV03910
      BISTOL = TTOL*TEMP                                            LSV03920
      CONTOL = BETAM*1.D-10                                         LSV03930
C                                                                   LSV03940
C-------------------------------------------------------------------LSV03950
C     SUBROUTINE LUMP 'COMBINES' T-EIGENVALUES THAT ARE 'TOO CLOSE'. LSV03960
C     NOTE HOWEVER THAT CLOSE SPURIOUS T-EIGENVALUES ARE NOT AVERAGED LSV03970
C     WITH GOOD ONES. HOWEVER, THEY MAY BE USED TO INCREASE THE     LSV03980
C     T-MULTIPLICITY OF A GOOD T-EIGENVALUE.                        LSV03990
C                                                                   LSV04000
      LOOP = NDIS                                                   LSV04010
      CALL LUMP(VS,RELTOL,MULTOL,SCALE2,MP,LOOP)                    LSV04020
C                                                                   LSV04030
C-------------------------------------------------------------------LSV04040
C                                                                   LSV04050
      IF(NDIS.EQ.LOOP) GO TO 260                                    LSV04060
C                                                                   LSV04070
      WRITE(6,250) NDIS, MEV, LOOP                                  LSV04080
  250 FORMAT(/I6,' DISTINCT T-EIGENVALUES WERE COMPUTED IN BISEC AT MEV LSV04090
     1=',I6/ 2X,' LUMP SUBROUTINE REDUCES NUMBER OF DISTINCT T-EIGENVALULSV04100
     1ES TO',I6)                                                    LSV04110
C                                                                   LSV04120
  260 CONTINUE                                                      LSV04130
      NDIS = LOOP                                                   LSV04140
      BETA(MP1) = BETAM                                             LSV04150
C                                                                   LSV04160
C-------------------------------------------------------------------LSV04170
C     THE SUBROUTINE ISOEV LABELS THOSE SIMPLE T-EIGENVALUES OF T(1,MEV)LSV04180
C     WITH VERY SMALL GAPS BETWEEN NEIGHBORING T-EIGENVALUES OF T(1,MEV)LSV04190
C     TO AVOID COMPUTING ERROR ESTIMATES FOR ANY SIMPLE GOOD        LSV04200
C     T-EIGENVALUE THAT IS TOO CLOSE TO A SPURIOUS T-EIGENVALUE.    LSV04210
C     ON RETURN FROM ISOEV, G CONTAINS CODED MINIMAL GAPS           LSV04220
C     BETWEEN THE DISTINCT EIGENVALUES OF T(1,MEV). (G IS REAL).    LSV04230
C     G(I) < 0 MEANS MINGAP IS DUE TO LEFT GAP G(I) > 0 MEANS DUE TO LSV04240
C     RIGHT GAP. MP(I) = -1 MEANS THAT THE GOOD T-EIGENVALUE IS SIMPLE LSV04250
C     AND HAS A VERY SMALL MINGAP IN T(1,MEV) DUE TO A SPURIOUS     LSV04260
C     T-EIGENVALUE.                                                 LSV04270
C     NG = NUMBER OF GOOD T-EIGENVALUES.                            LSV04280
C     NISO = NUMBER OF ISOLATED, GOOD T-EIGENVALUES.                LSV04290
```

```
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)        LSV04850
C      MP CONTAINS THE CORRESPONDING CODED T-MULTIPLICITIES         LSV04860
C                                                                   LSV04870
       IT = MXINIT                                                  LSV04880
       CALL INVERR(BETA,V1,V2,VS,EPSM,G,MP,MEV,MMB,NDIS,NISO,NM,    LSV04890
      1 RHSEED,IT,IWRITE)                                           LSV04900
C                                                                   LSV04910
C-------------------------------------------------------------------LSV04920
C                                                                   LSV04930
C      SIMPLE CHECK FOR CONVERGENCE. CHECKS TO SEE IF ALL OF THE ERROR  LSV04940
C      ESTIMATES ARE SMALLER THAN CONTOL.                          LSV04950
C      IF THIS TEST IS SATISFIED, THEN CONVERGENCE FLAG, ICONV IS SET   LSV04960
C      TO 1.  TYPICALLY ERROR ESTIMATES ARE VERY CONSERVATIVE.     LSV04970
C                                                                   LSV04980
       WRITE(6,350) CONTOL                                          LSV04990
  350 FORMAT(/' CONVERGENCE IS TESTED USING THE CONVERGENCE TOLERANCE', LSV05000
      1E13.4/)                                                      LSV05010
C                                                                   LSV05020
       II = MEV +1                                                  LSV05030
       IF = MEV+NISO                                                LSV05040
       DO 360 I = II,IF                                             LSV05050
       IF (ABS(G(I)).GT.CONTOL) GO TO 380                          LSV05060
  360 CONTINUE                                                      LSV05070
       ICONV = 1                                                    LSV05080
       MMB = NMEVS                                                  LSV05090
C                                                                   LSV05100
       WRITE(6,370) CONTOL                                          LSV05110
  370 FORMAT(' ALL COMPUTED ERROR ESTIMATES WERE LESS THAN',E15.4/ LSV05120
      1 ' THEREFORE PROCEDURE TERMINATES'/)                        LSV05130
C                                                                   LSV05140
  380 CONTINUE                                                      LSV05150
C                                                                   LSV05160
C      IF CONVERGENCE IS INDICATED, THAT IS ICONV = 1 ,THEN        LSV05170
C      THE SUBROUTINE PRTEST IS CALLED TO CHECK FOR ANY CONVERGED  LSV05180
C      T-EIGENVALUES THAT HAVE BEEN MISLABELLED AS SPURIOUS BECAUSE LSV05190
C      THE PROJECTION OF THEIR SINGULAR VECTOR ON THE STARTING     LSV05200
C      VECTOR WAS TOO SMALL.  NUMERICAL TESTS INDICATE THAT        LSV05210
C      SUCH SINGULAR VALUES ARE RARE.  THEREFORE, IF MANY OF       LSV05220
C      THESE HIDDEN SINGULAR VALUES APPEAR ON SOME RUN, THE USER   LSV05230
C      CAN BE CERTAIN THAT SOMETHING IS FOULED UP.                 LSV05240
C                                                                   LSV05250
       IF (ICONV.EQ.0) GO TO 510                                   LSV05260
C                                                                   LSV05270
C-------------------------------------------------------------------LSV05280
C                                                                   LSV05290
       CALL PRTEST (BETA,VS,TKMAX,EPSM,RELTOL,SCALE3,SCALE4,       LSV05300
      1 MP,NDIS,MEV,IPROJ)                                          LSV05310
C                                                                   LSV05320
C-------------------------------------------------------------------LSV05330
C                                                                   LSV05340
       IF(IPROJ.EQ.0) GO TO 500                                    LSV05350
C                                                                   LSV05360
       IF(IDIST.EQ.1)  WRITE(11,390) IPROJ                         LSV05370
  390 FORMAT(' SUBROUTINE PRTEST WANTS TO RELABEL',I6,' SPURIOUS T-EIGENLSV05380
      1VALUES'/' WE ACCEPT RELABELLING ONLY IF LAST COMPONENT OF T-EIGENVLSV05390
```

```
      1ECTOR IS L.T. 1.D-10'/)                                      LSV05400
C                                                                   LSV05410
      IIX = RHSEED                                                  LSV05420
C                                                                   LSV05430
C-----------------------------------------------------------------LSV05440
C                                                                   LSV05450
      CALL GENRAN(IIX,G,MEV)                                        LSV05460
C                                                                   LSV05470
C-----------------------------------------------------------------LSV05480
C                                                                   LSV05490
      ITEN = -10                                                    LSV05500
      NISOM = NISO + MEV                                            LSV05510
      IWRITO = IWRITE                                               LSV05520
      IWRITE = 0                                                    LSV05530
C                                                                   LSV05540
      DO 420 J = 1,NDIS                                             LSV05550
      IF(MP(J).NE.ITEN) GO TO 420                                   LSV05560
      TO = VS(J)                                                    LSV05570
C                                                                   LSV05580
C-----------------------------------------------------------------LSV05590
C                                                                   LSV05600
      IT = MXINIT                                                   LSV05610
      CALL INVERM(BETA,V1,V2,TO,TEMP,T1,EPSM,G,MEV,IT,IWRITE)       LSV05620
C                                                                   LSV05630
C-----------------------------------------------------------------LSV05640
C                                                                   LSV05650
      IF(TEMP.LE.1.D-10) GO TO 410                                  LSV05660
C     ERROR ESTIMATE WAS NOT SMALL REJECT RELABELLING OF THIS       LSV05670
C     T-EIGENVALUE.                                                 LSV05680
      IF(IDIST.EQ.1) WRITE(11,400) J,TO,TEMP                        LSV05690
  400 FORMAT(/' LAST COMPONENT FOR',I6,'TH T-EIGENVALUE',E20.12/' IS TOOLSV05700
     1 LARGE = ',E15.6,' SO DO NOT ACCEPT PRTEST RELABELLING'/)     LSV05710
      MP(J) = 0                                                     LSV05720
      IPROJ = IPROJ - 1                                             LSV05730
      GO TO 420                                                     LSV05740
C     RELABELLING ACCEPTED                                          LSV05750
  410 NISOM = NISOM + 1                                             LSV05760
      G(NISOM) = BETAM*TEMP                                         LSV05770
  420 CONTINUE                                                      LSV05780
      IWRITE = IWRITO                                               LSV05790
C                                                                   LSV05800
      IF(IPROJ.EQ.0) GO TO 460                                      LSV05810
      WRITE(6,430) IPROJ                                            LSV05820
  430 FORMAT(/I6,' T-EIGENVALUES WERE RECLASSIFIED AS GOOD.'/       LSV05830
     1' THESE ARE IDENTIFIED IN FILE 3 BY A T-MULTIPLICITY OF -10'/' USELSV05840
     2R SHOULD INSPECT EACH TO MAKE SURE NEIGHBORS HAVE CONVERGED'/) LSV05850
C                                                                   LSV05860
      IF(IDIST.EQ.1)  WRITE(11,440) IPROJ                           LSV05870
  440 FORMAT(/I6,' T-EIGENVALUES WERE RELABELLED AS GOOD'/          LSV05880
     1' BELOW IS CORRECTED T-MULTIPLICITY PATTERN'/)                LSV05890
C                                                                   LSV05900
      WRITE(6,450) NDIS, (MP(I), I=1,NDIS)                          LSV05910
      IF(IDIST.EQ.1) WRITE(11,450) NDIS, (MP(I), I=1,NDIS)          LSV05920
  450 FORMAT(/I6,' = NDIS, T-MULTIPLICITIES (0 MEANS  SPURIOUS)'/   LSV05930
     1 6X, ' (-10) MEANS SPURIOUS T-EIGENVALUE RELABELLED AS GOOD'/(20I4LSV05940
```

```
      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                                                               LSV06500
      WRITE(6,530)MEV                                           LSV06510
  530 FORMAT(//' SINGULAR VALUE CALCULATION AT MEV = ',I6,'    IS COMPLELSV06520
     1TE'//)                                                    LSV06530
C                                                               LSV06540
C     NG = NUMBER OF COMPUTED DISTINCT GOOD T-EIGENVALUES.  NEXT LSV06550
C     GENERATE GAPS BETWEEN GOOD T-EIGENVALUES (BMINGAPS) AND PUT THEM LSV06560
C     IN G.  G(J) < 0 MEANS THE BMINGAP IS DUE TO THE LEFT-HAND GAP. LSV06570
C                                                               LSV06580
      NGM1 = NG - 1                                             LSV06590
      G(NG) = V2(NGM1)-V2(NG)                                   LSV06600
      G(1) = V2(2)-V2(1)                                        LSV06610
C                                                               LSV06620
      DO 540 J = 2,NGM1                                         LSV06630
      T0 = V2(J)-V2(J-1)                                        LSV06640
      T1 = V2(J+1)-V2(J)                                        LSV06650
      G(J) = T1                                                 LSV06660
      IF (T0.LT.T1) G(J) = -T0                                  LSV06670
  540 CONTINUE                                                  LSV06680
C                                                               LSV06690
C     WRITE GOOD T-EIGENVALUES (COMPUTED SINGULAR VALUES) OUT TO FILE 3.LSV06700
C                                                               LSV06710
      WRITE(3,550)NG,NDIS,MEV,M,N,SVSEED,MATNO,IPARO,MULTOL,IB,BTOL LSV06720
  550 FORMAT(5I6,I12,I8,I2,'=NG,ND,MEV,M,N,SEED,MN,IPARO'/      LSV06730
     1 E20.12,I6,E13.4,' = MUTOL,INDEX MINIMAL BETA,BTOL'/      LSV06740
     1' SV NO',2X,'T-MULT',10X,'SINGULAR VALUE',7X,'BMINGAP',7X,'TMINGAPLSV06750
     1')                                                        LSV06760
C                                                               LSV06770
      WRITE(3,560)(I,MP(I),V2(I),G(I),V1(I),  I=1,NG)           LSV06780
  560 FORMAT(I6,I8,E25.16,2E14.3)                               LSV06790
C                                                               LSV06800
C     IF CONVERGENCE FLAG ICONV.NE.1 AND NUMBER OF T-MATRICES    LSV06810
C     CONSIDERED TO DATE IS LESS THAN NUMBER ALLOWED, INCREMENT MEV. LSV06820
C     AND LOOP BACK TO 210 TO REPEAT COMPUTATIONS.  RESTORE BETA(MEV+1).LSV06830
C                                                               LSV06840
      BETA(MP1) = BETAM                                         LSV06850
C                                                               LSV06860
      IF (MMB.LT.NMEVS.AND.ICONV.NE.1) GO TO 210                LSV06870
C                                                               LSV06880
C     END OF LOOP ON DIFFERENT SIZE T-MATRICES ALLOWED.         LSV06890
C                                                               LSV06900
  570 CONTINUE                                                  LSV06910
C                                                               LSV06920
      IF(ISTOP.EQ.0)  WRITE(6,580)                              LSV06930
  580 FORMAT(/' T-MATRICES (BETA) ARE NOW AVAILABLE, TERMINATE'/)LSV06940
      IF (IHIS.EQ.1.AND.KMAX.NE.MOLD) WRITE(1,590)              LSV06950
  590 FORMAT(/' ABOVE ARE THE FOLLOWING VECTORS '/              LSV06960
     2 '   BETA(I), I = 1,KMAX+1'/                              LSV06970
     3 ' FINAL TWO LANCZOS VECTORS OF ORDERS M,N FOR I = KMAX,KMAX+1'/ LSV06980
     4 ' ALL VECTORS IN THIS FILE HAVE FORMAT 4Z20'/            LSV06990
     5 ' ----- END OF FILE 1 NEW BETA HISTORY---------------'///)LSV07000
C                                                               LSV07010
      IF (ISTOP.EQ.0) GO TO 690                                 LSV07020
C                                                               LSV07030
      WRITE(3,600)                                              LSV07040
```

```
    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 SPURIOUS'LSV07240
       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)-------------LSV00010
C  Authors:  Jane Cullum and Ralph A. Willoughby (Deceased)         LSV00020
C            Los Alamos National Laboratory                          LSV00030
C            Los Alamos, New Mexico 87544                            LSV00040
C                                                                    LSV00050
C            E-mail:  cullumj@lanl.gov                               LSV00060
C                                                                    LSV00070
C  These codes are copyrighted by the authors.  These codes         LSV00080
C  and modifications of them or portions of them are NOT to be       LSV00090
C  incorporated into any commercial codes or used for any other      LSV00100
C  commercial purposes such as consulting for other companies,       LSV00110
C  without legal agreements with the authors of these Codes.         LSV00120
C  If these Codes or portions of them are used in other scientific or LSV00130
C  engineering research works the names of the authors of these codes LSV00140
C  and appropriate references to their written work are to be        LSV00150
C  incorporated in the derivative works.                             LSV00160
C                                                                    LSV00170
C  This header is not to be removed from these codes.                LSV00180
C                                                                    LSV00190
C        REFERENCE: Cullum and Willoughby, Chapter 5                 LSV00191
C        Lanczos Algorithms for Large Symmetric Eigenvalue ComputationsLSV00192
C        VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in LSV00193
C        Applied Mathematics, 2002. SIAM Publications,               LSV00194
C        Philadelphia, PA. USA                                       LSV00195
C                                                                    LSV00196
C                                                                    LSV00197
C                                                                    LSV00200
C     CONTAINS MAIN PROGRAM FOR COMPUTING A LEFT AND A               LSV00210
C     RIGHT SINGULAR VECTOR CORRESPONDING TO EACH OF A SET           LSV00220
C     OF SINGULAR VALUES WHICH HAVE BEEN COMPUTED ACCURATELY BY THE  LSV00230
C     CORRESPONDING LANCZOS SINGULAR VALUE PROGRAM (LSVAL)           LSV00240
C     FOR REAL RECTANGULAR MATRICES.  THIS PROGRAM COULD BE          LSV00250
C     MODIFIED TO COMPUTE ADDITIONAL SINGULAR VECTORS FOR ANY        LSV00260
C     SINGULAR VALUE THAT IS A MULTIPLE SINGULAR VALUE OF A.         LSV00270
C     THE AMOUNT OF ADDITIONAL COMPUTATION REQUIRED BY SUCH A        LSV00280
C     MODIFICATION DEPENDS UPON THE GIVEN A-MATRIX AND UPON          LSV00290
C     THE PART OF THE SPECTRUM INVOLVED.                             LSV00300
C                                                                    LSV00310
C     FOR A GIVEN REAL MATRIX A OF ORDER M X N THE LANCZOS RECURSION LSV00320
C     IS APPLIED TO THE ASSOCIATED REAL SYMMETRIC MATRIX B OF ORDER  LSV00330
C     MN = M+N                                                       LSV00340
C                                                                    LSV00350
C                               ----            ----                 LSV00360
C                              | 0            A |                    LSV00370
C                  B  =        |                |                    LSV00380
C                              | A-TRANSPOSE  0 |                    LSV00390
C                               ----            ----                 LSV00400
C     USING SPECIAL STARTING VECTORS.                                LSV00410
C                                                                    LSV00420
C     THESE SINGULAR VECTOR COMPUTATIONS ASSUME THAT EACH            LSV00430
C     SINGULAR VALUE THAT IS BEING CONSIDERED HAS CONVERGED AS       LSV00440
```

```
C       AN EIGENVALUE OF THE LANCZOS TRIDIAGONAL MATRICES GENERATED.      LSV00450
C                                                                         LSV00460
C       THE EIGENVALUES OF EACH EVEN-ORDERED LANCZOS MATRIX OCCUR         LSV00470
C       IN + AND - PAIRS, AND THE RITZ VECTOR COMPUTATION RESTS ON        LSV00480
C       AN INVERSE ITERATION COMPUTATION FOR A LANCZOS MATRIX.            LSV00490
C       THIS CAUSES AN ANOMALY IN THE SINGULAR VECTOR COMPUTATIONS        LSV00500
C       FOR VERY SMALL SINGULAR VALUES.  IN PRACTICE WE SEE THAT          LSV00510
C       FOR ANY SUCH SINGULAR VALUE THAT ONE MEMBER OF EACH PAIR OF       LSV00520
C       APPROXIMATE SINGULAR VECTORS WILL BE MORE ACCURATE THAN THE       LSV00530
C       OTHER MEMBER OF THAT PAIR IS.  IF IPAR = 1 (STARTING LANCZOS      LSV00540
C       VECTOR IS OF FORM (0,V2) WHERE V2 IS NX1) THEN THE RIGHT          LSV00550
C       SINGULAR VECTOR WILL BE OBTAINED MORE ACCURATELY THAN THE         LSV00560
C       LEFT SINGULAR VECTOR.  IF IPAR = 2 (STARTING LANCZOS VECTOR       LSV00570
C       IS OF FORM (V1,0) WHERE V1 IS MX1) THEN THE LEFT SINGULAR         LSV00580
C       VECTOR WILL BE MORE ACCURATE THAN THE RIGHT SINGULAR VECTOR.      LSV00590
C       PRIOR TO NORMALIZATION THE SIZES OF THESE INACCURATE VECTORS      LSV00600
C       WILL BE THE SAME AS THE SIZE OF THE ASSOCIATED VERY SMALL         LSV00610
C       SINGULAR VALUE.  IN FACT IN THE LIMIT, FOR A ZERO SINGULAR VALUE  LSV00620
C       AND IPAR = 1, THE VECTOR COMPUTED AS THE APPROXIMATION TO THE     LSV00630
C       LEFT SINGULAR VECTOR WILL BE THE 0 VECTOR.  (IF IPAR = 2 THEN     LSV00640
C       THIS WOULD BE THE RIGHT SINGULAR VECTOR).  THE CORRESPONDING      LSV00650
C       ERROR ESTIMATES WILL REFLECT THE INACCURACY OF THE ONE MEMBER     LSV00660
C       OF EACH SUCH PAIR, SINCE THESE ESTIMATES ARE A SUM OF ESTIMATES   LSV00670
C       FOR THE INDIVIDUAL MEMBERS OF THE PAIR.  THEREFORE, FOR ANY VERY  LSV00680
C       SMALL SINGULAR VALUE A CORRESPONDING SINGULAR VECTOR WILL BE      LSV00690
C       COMPUTED ONLY IF THE USER HAS SET THE FLAG ERCONT TO 1.           LSV00700
C                                                                         LSV00710
C-------------------------------------------------------------------------LSV00720
C                                                                         LSV00730
C       PFORT VERIFIER IDENTIFIED THE FOLLOWING NONPORTABLE               LSV00740
C       CONSTRUCTIONS                                                     LSV00750
C                                                                         LSV00760
C       1.  DATA/MACHEP/ STATEMENT                                        LSV00770
C       2.  ALL READ(5,*) STATEMENTS (FREE FORMAT)                        LSV00780
C       3.  FORMAT(20A4) USED WITH THE EXPLANATORY HEADER, EXPLAN         LSV00790
C       4.  HEXADECIMAL FORMAT (4Z20) USED FOR BETA HISTORY.              LSV00800
C                                                                         LSV00810
C       IMPORTANT NOTE:  THIS PROGRAM ALLOWS ENLARGEMENT OF THE           LSV00820
C       BETA ARRAY.  IN PARTICULAR, IF ANY ONE OF THE SINGULAR VALUES     LSV00830
C       SUPPLIED IS T-SIMPLE AND AS AN EIGENVALUE OF THE ASSOCIATED       LSV00840
C       LANCZOS TRIDIAGONAL MATRIX IS NOT CLOSE TO A SPURIOUS             LSV00850
C       EIGENVALUE OF THAT MATRIX, THIS PROGRAM WILL REQUIRE              LSV00860
C       THAT KMAX BE AT LEAST THE LARGEST EVEN NUMBER LESS                LSV00870
C       THAN OR EQUAL TO  (11*MEV)/8 + 13.  IF KMAX IS NOT THAT           LSV00880
C       LARGE, THEN THIS PROGRAM WILL RESET KMAX TO THIS SIZE             LSV00890
C       AND EXTEND THE BETA HISTORY IF REQUIRED.                          LSV00900
C       THUS, THE DIMENSION OF THE BETA ARRAY MUST BE                     LSV00910
C       LARGE ENOUGH TO ALLOW FOR THIS POSSIBILITY.                       LSV00920
C       REMEMBER THAT THE BETA ARRAY, BETA(J), IS SUCH THAT               LSV00930
C       J = 1,..., KMAX+1.  SO IF THE KMAX USED BY THE PROGRAM            LSV00940
C       IS TO BE 3000, THEN BETA MUST BE OF LENGTH AT LEAST 3001.         LSV00950
C                                                                         LSV00960
C-------------------------------------------------------------------------LSV00970
        DOUBLE PRECISION  BETA(5001),V1(5000),V2(5000),RITVEC(30000)      LSV00980
        DOUBLE PRECISION  TVEC(30000),GOODSV(50),SVNEW(50),TLAST(50)      LSV00990
```

```
      DOUBLE PRECISION  SVAL,SVALN,TOLN,TTOL,ERTOL,BATA        LSV01000
      DOUBLE PRECISION  MULTOL,SCALE0,STUTOL,BTOL,LB,UB        LSV01010
      DOUBLE PRECISION  ONE,ZERO,MACHEP,EPSM,TEMP,SUM          LSV01020
      DOUBLE PRECISION  RELTOL,ERROR,TERROR,ERRMIN,BKMIN       LSV01030
      REAL      G(10000),BMINGP(50),TMINGP(50),EXPLAN(20)      LSV01040
      REAL      TERR(50),BERR(50),BERRGP(50),RNORM(50),TBETA(50) LSV01050
      INTEGER  MP(50),M1(50),M2(50),MA(50),ML(50),MINT(50),MFIN(50) LSV01060
      INTEGER  SVSEED,SVSOLD,RHSEED,IDELTA(50)                 LSV01070
      INTEGER  MBOUND,NTVCON,SVTVEC,TVSTOP,LVCONT,ERCONT,TFLAG LSV01080
      DOUBLE PRECISION  FINPRO                                 LSV01090
      DOUBLE PRECISION  DABS, DMAX1, DSQRT, DFLOAT             LSV01100
      REAL ABS                                                 LSV01110
      INTEGER  IABS                                            LSV01120
C----------------------------------------------------------------LSV01130
      EXTERNAL SVMAT, STRAN                                    LSV01140
      DATA MACHEP/Z3410000000000000/                           LSV01150
      EPSM = 2.D0*MACHEP                                       LSV01160
C----------------------------------------------------------------LSV01170
C                                                              LSV01180
C     ARRAYS MUST BE DIMENSIONED AS FOLLOWS:                   LSV01190
C     1.  BETA: >= (KMAX+1) WHERE KMAX, THE LARGEST SIZE       LSV01200
C                 T-MATRIX CONSIDERED BY THE PROGRAM, IS THE   LSV01210
C                 LARGER OF THE SIZE OF THE BETA HISTORY PROVIDED LSV01220
C                 ON FILE 2 (IF ANY ) AND THE SIZE WHICH THE PROGRAM LSV01230
C                 SPECIFIES INTERNALLY, THIS LATTER IS ALWAYS  LSV01240
C                 < = (11*MEV)/8 + 13, WHERE MEV IS THE SIZE   LSV01250
C                 T-MATRIX THAT WAS USED IN THE CORRESPONDING  LSV01260
C                 SINGULAR VALUE COMPUTATIONS.  NOTE THAT ALL  LSV01270
C                 T-MATRICES CONSIDERED MUST HAVE EVEN ORDER.  LSV01280
C     2.  V1:  >= MAX(M,KMAX)                                  LSV01290
C     3.  V2:   >= N                                           LSV01300
C     4.  G:  >= MAX(M,N,KMAX)                                 LSV01310
C     5.  RITVEC:  >= (N+M)*NGOOD, WHERE NGOOD IS THE NUMBER OF LSV01320
C                   SINGULAR VALUES SUPPLIED TO THIS PROGRAM.  LSV01330
C     6.  TVEC:  >= CUMULATIVE LENGTH OF ALL THE T-EIGENVECTORS LSV01340
C                   NEEDED TO GENERATE THE DESIRED RITZ VECTORS.  AN LSV01350
C                   EDUCATED GUESS AT AN APPROPRIATE LENGTH CAN BE LSV01360
C                   OBTAINED BY RUNNING THE PROGRAM WITH THE FLAG LSV01370
C                   MBOUND = 1 AND MULTIPLYING THE RESULTING SIZE BY 5/4. LSV01380
C     7.  GOODSV, TMINGP, BMINGP,TERR, BERR, BERRGP, RNORM,    LSV01390
C         TBETA, TLAST, SVNEW, MP, MA, M1, M2, MINT, MFIN AND  LSV01400
C         IDELTA MUST ALL BE >= NGOOD.                         LSV01410
C                                                              LSV01420
C----------------------------------------------------------------LSV01430
C     OUTPUT HEADER                                            LSV01440
      WRITE(6,10)                                              LSV01450
   10 FORMAT(/' LANCZOS PROCEDURE FOR REAL, RECTANGULAR MATRICES'/ LSV01460
     1'      COMPUTE SINGULAR VECTORS'/)                       LSV01470
C                                                              LSV01480
C     SET PROGRAM PARAMETERS                                   LSV01490
C     USER MUST NOT MODIFY SCALE0                              LSV01500
      SCALE0 = 5.0D0                                           LSV01510
      ZERO = 0.0D0                                             LSV01520
      ONE = 1.0D0                                              LSV01530
      MPMIN = -1000                                            LSV01540
```

```
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        LSV02100
C                   IS WRITTEN TO FILE 6                               LSV02110
C      IREAD = 0:   BETA FILE IS REGENERATED.                          LSV02120
C      IREAD = 1:   BETA FILE USED IN SINGULAR VALUE COMPUTATIONS      LSV02130
C                   IS READ IN AND EXTENDED IF NECESSARY.  IN BOTH     LSV02140
C                   CASES IREAD = 0 OR 1, THE LANCZOS VECTORS ARE      LSV02150
C                   ALWAYS REGENERATED FOR THE RITZ VECTOR            LSV02160
C                   COMPUTATIONS                                       LSV02170
C                                                                      LSV02180
       READ(5,20) EXPLAN                                               LSV02190
       READ(5,*) MBOUND,NTVCON,SVTVEC,IREAD                            LSV02200
C                                                                      LSV02210
       READ(5,20) EXPLAN                                               LSV02220
       READ(5,*) TVSTOP,LVCONT,ERCONT,IWRITE                          LSV02230
       IF (TVSTOP.EQ.1) SVTVEC = 1                                     LSV02240
C                                                                      LSV02250
C      READ IN SEED (RHSEED) FOR GENERATING RANDOM STARTING VECTOR     LSV02260
C      FOR THE INVERSE ITERATION ON THE T-MATRICES.                    LSV02270
C                                                                      LSV02280
       READ(5,20) EXPLAN                                               LSV02290
       READ(5,*) RHSEED                                                LSV02300
C                                                                      LSV02310
C-----------------------------------------------------------------------LSV02320
C      INITIALIZE THE ARRAYS FOR THE USER-SPECIFIED MATRIX AND         LSV02330
C      PASS THE STORAGE LOCATIONS OF THESE ARRAYS TO THE MATRIX-VECTOR LSV02340
C      MULTIPLY SUBROUTINES SVMAT AND STRAN.                           LSV02350
C                                                                      LSV02360
       CALL USPEC(M,N,MATNO)                                           LSV02370
C                                                                      LSV02380
C-----------------------------------------------------------------------LSV02390
C      MASK UNDERFLOW AND OVERFLOW                                     LSV02400
       CALL MASK                                                       LSV02410
C                                                                      LSV02420
C-----------------------------------------------------------------------LSV02430
C      WRITE RUN PARAMETERS OUT TO FILE 6                              LSV02440
C                                                                      LSV02450
       WRITE(6,30) M,N,MATNO                                           LSV02460
   30 FORMAT(/' MATRIX ORDER =',I5,' BY ',I5/                          LSV02470
      1 ' A-MATRIX AND CASE IDENTIFIER = ',I10/)                       LSV02480
C                                                                      LSV02490
       WRITE(6,40) MBOUND,NTVCON,SVTVEC,IREAD                          LSV02500
   40 FORMAT(/3X,'MBOUND',3X,'NTVCON',3X,'SVTVEC',3X,'IREAD'/3I9,I8/)   LSV02510
C                                                                      LSV02520
       WRITE(6,50) TVSTOP,LVCONT,ERCONT,IWRITE                        LSV02530
   50 FORMAT(/3X,'TVSTOP',3X,'LVCONT',3X,'ERCONT',3X,'IWRITE'/4I9)     LSV02540
C                                                                      LSV02550
       WRITE(6,60) MDIMTV,MDIMRV,MBETA                                 LSV02560
   60 FORMAT(/3X,'MDIMTV',3X,'MDIMRV',3X,'MBETA'/2I9,I8)               LSV02570
C                                                                      LSV02580
       WRITE(6,70) RELTOL,RHSEED                                       LSV02590
   70 FORMAT(/7X,'RELTOL',3X,'RHSEED'/E13.4,I9)                        LSV02600
C                                                                      LSV02610
C      FROM FILE 3 READ IN THE NUMBER OF SINGULAR VALUES (NGOOD)       LSV02620
C      FOR WHICH SINGULAR VECTORS ARE REQUESTED, THE ORDER (MEV) OF    LSV02630
C      THE LANCZOS TRIDIAGONAL MATRIX USED IN COMPUTING THESE          LSV02640
```

```
C      SINGULAR VALUES, THE ORDER MOLD X NOLD OF THE USER-SPECIFIED      LSV02650
C      MATRIX USED IN THOSE COMPUTATIONS, THE SEED (SVSEED) USED FOR     LSV02660
C      GENERATING THE STARTING VECTOR THAT WAS USED IN THOSE            LSV02670
C      COMPUTATIONS, AND THE MATRIX/RUN IDENTIFICATION NUMBER (MATOLD)   LSV02680
C      USED IN THOSE COMPUTATIONS.  ALSO READ IN THE NUMBER (NDIS) OF    LSV02690
C      DISTINCT EIGENVALUES OF THE MATRIX T(1,MEV) THAT WERE COMPUTED    LSV02700
C      BUT THIS VALUE IS NOT USED IN THE SINGULAR VECTOR                LSV02710
C      COMPUTATIONS.                                                    LSV02720
C                                                                       LSV02730
       READ(3,80) NGOOD,NDIS,MEV,MOLD,NOLD,SVSEED,MATOLD,IPARO          LSV02740
    80 FORMAT(5I6,I12,I8,I2)                                            LSV02750
C                                                                       LSV02760
C      READ IN THE T-MULTIPLICITY TOLERANCE USED IN THE BISEC SUBROUTINE LSV02770
C      DURING THE COMPUTATION OF THE GIVEN SINGULAR VALUES.             LSV02780
C      ALSO READ IN THE FLAG IB.  IF IB < 0, THEN SOME BETA(I) IN THE   LSV02790
C      T-MATRIX FILE PROVIDED ON FILE 2 FAILED THE ORTHOGONALITY        LSV02800
C      TEST IN THE TNORM SUBROUTINE.  USER SHOULD NOTE THAT THIS        LSV02810
C      PROGRAM PROCEEDS INDEPENDENTLY OF THE SIZE OF THE BETA USED.     LSV02820
C                                                                       LSV02830
       READ(3,90) MULTOL,IB,BTOL                                       LSV02840
    90 FORMAT(E20.12,I6,E13.4)                                         LSV02850
C                                                                       LSV02860
       TEMP = DFLOAT(MEV+1000)                                         LSV02870
       TTOL = MULTOL/TEMP                                              LSV02880
       WRITE(6,100) MULTOL,TTOL                                        LSV02890
   100 FORMAT(/' T-MULTIPLICITY TOLERANCE USED IN THE SINGULAR VALUE COMPLSV02900
      1UTATIONS WAS',E13.4/' SCALED MACHINE EPSILON IS',E13.4)         LSV02910
C                                                                       LSV02920
C      CONTINUE WRITE TO FILE 6 OF THE PARAMETERS FOR THIS RUN          LSV02930
C                                                                       LSV02940
       WRITE(6,110)NGOOD,NDIS,MEV,MOLD,NOLD,MATOLD,SVSEED,MULTOL,IB,    LSV02950
      1BTOL,IPARO                                                      LSV02960
   110 FORMAT(/' SINGULAR VALUES SUPPLIED ARE READ IN FROM FILE 3'/     LSV02970
      1 6X,'NG',2X,'NDIS',3X,'MEV',2X,'MOLD',2X,'NOLD',2X,'MATOLD',4X/  LSV02980
      1I8,4I6,I8//6X,'SVSEED',6X,'MULTOL',9X,'IB',8X,'BTOL',4X,'IPARO'/ LSV02990
      1I12,E12.3,I11,E12.4,I9/)                                        LSV03000
C                                                                       LSV03010
C      IS THE ARRAY RITVEC LONG ENOUGH TO HOLD ALL OF THE DESIRED       LSV03020
C      RITZ VECTORS (APPROXIMATE EIGENVECTORS OF B)?                    LSV03030
       MNMAX = NGOOD*MN                                                LSV03040
       IF(MBOUND.EQ.1) GO TO 120                                       LSV03050
       IF(TVSTOP.NE.1.AND.MNMAX.GT.MDIMRV) GO TO 1600                  LSV03060
C                                                                       LSV03070
C      CHECK THAT THE ORDERS M,N AND THE MATRIX IDENTIFICATION NUMBER   LSV03080
C      MATNO SPECIFIED BY THE USER AGREE WITH THOSE READ IN FROM        LSV03090
C      FILE 3.                                                         LSV03100
   120 ITEMP = (MOLD-M)**2+(NOLD-N)**2+(MATOLD-MATNO)**2               LSV03110
       IF (ITEMP.NE.0) GO TO 1620                                      LSV03120
C                                                                       LSV03130
C      READ IN FROM FILE 3, THE T-MULTIPLICITIES OF THE SINGULAR VALUES LSV03140
C      WHOSE SINGULAR VECTORS ARE TO BE COMPUTED, THE VALUES OF THESE   LSV03150
C      SINGULAR VALUES AND THEIR MINIMAL GAPS AS SINGULAR VALUES OF THE LSV03160
C      USER-SPECIFIED MATRIX AND OF THE RELATED T-MATRIX.              LSV03170
C                                                                       LSV03180
       READ(3,20) EXPLAN                                              LSV03190
```

```
      READ(3,130) (MP(J),GOODSV(J),BMINGP(J),TMINGP(J), J=1,NGOOD)       LSV03200
  130 FORMAT(6X,I8,E25.16,2E14.3)                                        LSV03210
C                                                                        LSV03220
      WRITE(6,140) (J,GOODSV(J),MP(J),BMINGP(J),  J=1,NGOOD)             LSV03230
  140 FORMAT(/' SINGULAR VALUES READ IN FROM FILE 3 AND THEIR T-MULTIPLILSV03240
     1CITIES'/4X,' J ',4X,' SINGULAR VALUE',5X,'TMULT',4X,'BMINGP'/      LSV03250
     1(I6,E20.12,I6,E13.4))                                             LSV03260
C                                                                        LSV03270
      WRITE(6,150) MEV,SVSEED                                            LSV03280
  150 FORMAT(/' THESE SINGULAR VALUES WERE COMPUTED USING A T-MATRIX OF LSV03290
     1ORDER ',I5/' AND SEED FOR RANDOM NUMBER GENERATOR =',I12)          LSV03300
C                                                                        LSV03310
C     READ IN THE ERROR ESTIMATES                                        LSV03320
C                                                                        LSV03330
C     CHECK WHETHER OR NOT THERE ARE ANY ISOLATED T-EIGENVALUES IN       LSV03340
C     THE T-EIGENVALUES PROVIDED (HERE THE SINGULAR VALUES ARE           LSV03350
C     CONSIDERED AS EIGENVALUES OF THE ASSOCIATED LANCZOS TRIDIAGONAL    LSV03360
C     MATRICES.)                                                         LSV03370
      DO 160 J=1,NGOOD                                                   LSV03380
      IF(MP(J).EQ.1) GO TO 170                                           LSV03390
  160 CONTINUE                                                           LSV03400
      GO TO 190                                                          LSV03410
  170 READ(4,20) EXPLAN                                                  LSV03420
      READ(4,20) EXPLAN                                                  LSV03430
      READ(4,20) EXPLAN                                                  LSV03440
      READ(4,180) NISO                                                   LSV03450
  180 FORMAT(18X,I6)                                                     LSV03460
      READ(4,20) EXPLAN                                                  LSV03470
      READ(4,20) EXPLAN                                                  LSV03480
      READ(4,20) EXPLAN                                                  LSV03490
  190 DO 220 J=1,NGOOD                                                   LSV03500
      BERR(J) = 0.D0                                                     LSV03510
      IF(MP(J).NE.1) GO TO 220                                           LSV03520
      READ(4,200) SVAL, BERR(J)                                          LSV03530
  200 FORMAT(10X,E25.16,E14.3)                                           LSV03540
      IF(DABS(SVAL - GOODSV(J)).LT.1.D-10) GO TO 220                     LSV03550
      WRITE(6,210) SVAL,GOODSV(J)                                        LSV03560
  210 FORMAT(' PROBLEM WITH READ IN OF ERROR ESTIMATES'/' SINGULAR VALUELSV03570
     1READ IN',E20.12,' DOES NOT MATCH GOODSV(J) ='/E20.12)              LSV03580
      GO TO 1860                                                         LSV03590
C                                                                        LSV03600
  220 CONTINUE                                                           LSV03610
C                                                                        LSV03620
      WRITE(6,230) (J,GOODSV(J),BERR(J), J=1,NGOOD)                      LSV03630
  230 FORMAT(' ERROR ESTIMATES ='/4X,' J',3X,'SINGULAR VALUE',8X,        LSV03640
     1'ESTIMATE'/(I6,E20.12,E14.3))                                      LSV03650
C                                                                        LSV03660
      IF(IREAD.EQ.0) IPAR = IPARO                                        LSV03670
      IF(IREAD.EQ.0)  GO TO 350                                          LSV03680
C                                                                        LSV03690
C     READ IN THE SIZE OF THE T-MATRIX PROVIDED ON FILE 2.  READ IN      LSV03700
C     THE ORDER OF THE USER-SPECIFIED MATRIX , THE FLAGS IPARO           LSV03710
C     AND IPAR WHICH INDICATE RESPECTIVELY THE PARITY OF THE             LSV03720
C     STARTING VECTOR USED IN THE GENERATION OF THE EXISTING             LSV03730
C     BETA AND THE PARITY OF THE NEXT LANCZOS VECTOR THAT                LSV03740
```

```
C      HAS TO BE GENERATED IF THE BETA HISTORY IS EXTENDED,          LSV03750
C      THE SEED USED BY THE RANDOM NUMBER GENERATOR WHEN             LSV03760
C      GENERATING THE STARTING VECTOR THAT WAS USED, AND THE         LSV03770
C      MATRIX/TEST IDENTIFICATION NUMBER THAT WERE USED IN           LSV03780
C      THE LANCZOS SINGULAR VALUE COMPUTATIONS.  IF THE FLAG         LSV03790
C      IREAD = 0, REGENERATE HISTORY AND DO NOT READ ANYTHING        LSV03800
C      FROM FILE 2.  HISTORY MUST BE STORED IN MACHINE FORMAT,       LSV03810
C      ((4Z20) FOR IBM 3081).                                        LSV03820
C                                                                    LSV03830
       READ(2,240) KMAX,MOLD,NOLD,IPARO,IPAR,SVSOLD,MATOLD           LSV03840
   240 FORMAT(3I6,2I3,I12,I8)                                        LSV03850
C                                                                    LSV03860
       WRITE(6,250) KMAX,MOLD,NOLD,IPARO,IPAR,SVSOLD,MATOLD          LSV03870
   250 FORMAT(/' READ IN HEADER FROM BETA FILE 2'/                   LSV03880
      1 2X,'KMAX',2X,'MOLD',2X,'NOLD',2X,'IPARO',2X,'IPAR',6X,'SVSOLD LSV03890
      1 ',2X,'MATOLD'/3I6,I7,I6,I12,I12)                             LSV03900
C                                                                    LSV03910
C      CHECK THAT THE PARAMETERS READ IN AGREE WITH WHAT THE USER    LSV03920
C      HAS SPECIFIED                                                 LSV03930
       IF(MOLD.NE.M.OR.NOLD.NE.N.OR.MATOLD.NE.MATNO.OR.SVSOLD.NE.SVSEED) LSV03940
      1 GO TO 1640                                                   LSV03950
C                                                                    LSV03960
       IF(IPARO.EQ.1) WRITE(6,260)                                   LSV03970
       IF(IPARO.EQ.2) WRITE(6,270)                                   LSV03980
   260 FORMAT(/' STARTING VECTOR USED IN EXISTING SINGULAR VALUE HISTORY LSV03990
      1WAS'/' OF THE FORM (0,V2)')                                   LSV04000
   270 FORMAT(/' STARTING VECTOR USED IN EXISTING SINGULAR VALUE HISTORY LSV04010
      1WAS'/' OF THE FORM (V1,0)')                                   LSV04020
C                                                                    LSV04030
       KMAX1 = KMAX + 1                                              LSV04040
C                                                                    LSV04050
C      READ IN THE T-MATRICES FROM FILE 2.  THESE ARE USED TO GENERATE  LSV04060
C      THE T-EIGENVECTORS THAT WILL BE USED IN THE RITZ VECTOR       LSV04070
C      COMPUTATIONS.  HISTORY MUST BE STORED IN 4Z20 FORMAT.         LSV04080
C                                                                    LSV04090
       READ(2,280) (BETA(J), J=1,KMAX1)                              LSV04100
   280 FORMAT(4Z20)                                                  LSV04110
C                                                                    LSV04120
       READ(2,280) (V1(J), J=1,M)                                    LSV04130
       READ(2,280) (V2(J), J=1,N)                                    LSV04140
C                                                                    LSV04150
C      KMAX MAY BE ENLARGED IF THE SIZE AT WHICH THE SINGULAR VALUE  LSV04160
C      COMPUTATIONS WERE PERFORMED IS ESSENTIALLY KMAX AND           LSV04170
C      THERE IS AT LEAST ONE SINGULAR VALUE THAT IS SIMPLE AS AN     LSV04180
C      EIGENVALUE OF T(1,MEV), AND IF ITS NEAREST NEIGHBOR IN THE    LSV04190
C      T-MATRIX IS TOO CLOSE, THAT NEIGHBOR IS A 'GOOD' T-EIGENVALUE. LSV04200
       DO 290 J = 1,NGOOD                                            LSV04210
       IF(MP(J).EQ.1) GO TO 310                                      LSV04220
   290 CONTINUE                                                      LSV04230
       WRITE(6,300)                                                  LSV04240
   300 FORMAT(/' ALL SINGULAR VALUES USED ARE T-MULTIPLE OR CLOSE TO SPURLSV04250
      1IOUS EIGENVALUES'/' (AS EIGENVALUES OF T(1,MEV)) SO KMAX IS NOT CHLSV04260
      1ANGED'/)                                                      LSV04270
       IF(KMAX.LT.MEV) GO TO 1660                                    LSV04280
       GO TO 330                                                     LSV04290
```

```
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)')                                                 LSV04850
C                                                              LSV04860
C-----------------------------------------------------------------LSV04870
C                                                              LSV04880
      CALL LANCZS(SVMAT,STRAN,BETA,V1,V2,G,KMAX,MOLD1,M,N,IPAR,SVSEED) LSV04890
C                                                              LSV04900
C-----------------------------------------------------------------LSV04910
C                                                              LSV04920
  460 CONTINUE                                                 LSV04930
C                                                              LSV04940
C     THE SUBROUTINE STURMI DETERMINES THE SMALLEST SIZE T-MATRIX FOR LSV04950
C     WHICH THE SINGULAR VALUE IN QUESTION IS AN EIGENVALUE (TO  LSV04960
C     WITHIN A SPECIFIED TOLERANCE) AND IF POSSIBLE THE SMALLEST LSV04970
C     SIZE T-MATRIX FOR WHICH THE SINGULAR VALUE IS A DOUBLE   LSV04980
C     EIGENVALUE (TO WITHIN THE SAME TOLERANCE).  THE SIZE     LSV04990
C     T-MATRIX THAT WILL BE USED IN EACH OF THE RITZ VECTOR COMPUTATIONSLSV05000
C     IS THEN DETERMINED BY LOOPING ON THE SIZE OF THE T-EIGENVECTOR LSV05010
C     COMPUTATIONS, STARTING WITH A SIZE DETERMINED FROM THE   LSV05020
C     INFORMATION OBTAINED FROM STURMI.                        LSV05030
C                                                              LSV05040
      STUTOL = SCALE0*MULTOL                                   LSV05050
      IF(IWRITE.EQ.1) WRITE(6,470)                             LSV05060
  470 FORMAT(' FROM STURMI')                                   LSV05070
      DO 510 J = 1,NGOOD                                       LSV05080
      SVAL = GOODSV(J)                                         LSV05090
C     COMPUTE THE TOLERANCES USED BY STURMI TO DETERMINE AN INTERVAL LSV05100
C     CONTAINING THE SINGULAR VALUE SVAL.                      LSV05110
      TEMP = DABS(SVAL)*RELTOL                                 LSV05120
      TOLN = DMAX1(TEMP,STUTOL)                                LSV05130
C                                                              LSV05140
C-----------------------------------------------------------------LSV05150
C                                                              LSV05160
      CALL STURMI(BETA,SVAL,TOLN,EPSM,KMAX,MK1,MK2,IC,IWRITE)  LSV05170
C                                                              LSV05180
C-----------------------------------------------------------------LSV05190
C                                                              LSV05200
C     STORE THE COMPUTED ORDERS OF T-MATRICES FOR LATER PRINTOUT LSV05210
      IF(MK1.GT.1) GO TO 475                                   LSV05220
C     SVAL IS VERY SMALL SINGULAR VALUE, RESET MK1 TO CORRECT VALUE LSV05230
      MK1 = MK2                                                LSV05240
      MK2 = MIN0(2*MK1,KMAX)                                   LSV05250
      M1(J) = MK1                                              LSV05260
      M2(J) = MK2                                              LSV05270
      ML(J) = MK2                                              LSV05280
      GO TO 476                                                LSV05290
  475 M1(J) = MK1                                              LSV05300
      M2(J) = MK2                                              LSV05310
      ML(J) = (MK1 + 3*MK2)/4                                  LSV05320
      IF(MK2.EQ.KMAX)  ML(J) = KMAX                            LSV05330
C                                                              LSV05340
  476 IF(IC.GT.0) GO TO 490                                    LSV05350
C     IC = 0 MEANS THERE WAS NO T-EIGENVALUE IN THE DESIGNATED INTERVAL LSV05360
C     EVEN BY T-SIZE KMAX.  THIS MEANS THAT THE SINGULAR VALUE LSV05370
C     PROVIDED HAS NOT YET CONVERGED SO PROGRAM DOES NOT COMPUTE LSV05380
C     A SINGULAR VECTOR FOR IT.                                LSV05390
```

```
      WRITE(6,480) J,GOODSV(J),MK1,MK2                              LSV05400
  480 FORMAT(I6,'TH SINGULAR VALUE',E20.12,' HAS NOT CONVERGED '/   LSV05410
     1' SO DO NOT COMPUTE ANY T-EIGENVECTOR OR RITZ VECTOR FOR IT'  LSV05420
     1/' MK1 AND MK2 FOR THIS SINGULAR VALUE WERE',2I6)            LSV05430
      MP(J) = MPMIN                                                 LSV05440
      MA(J) = -2*KMAX                                               LSV05450
      GO TO 510                                                     LSV05460
C     COMPUTE AN APPROPRIATE SIZE T-MATRIX FOR THE GIVEN SINGULAR   LSV05470
C     VALUE.                                                        LSV05480
  490 IF(M2(J).EQ.KMAX) GO TO 500                                   LSV05490
C     M1 AND M2 WERE BOTH DETERMINED                                LSV05500
      MAJ = (3*M1(J) + M2(J))/4  + 1                                LSV05510
      IF((MAJ/2)*2.NE.MAJ)  MAJ = MAJ + 1                           LSV05520
      MA(J) = MAJ                                                   LSV05530
      GO TO 510                                                     LSV05540
C     M2 NOT DETERMINED                                             LSV05550
  500 MAJ = (5*M1(J))/4  + 1                                        LSV05560
      IF((MAJ/2)*2.NE.MAJ)  MAJ = MAJ + 1                           LSV05570
      MA(J) = MAJ                                                   LSV05580
C                                                                   LSV05590
  510 CONTINUE                                                      LSV05600
C                                                                   LSV05610
      IF (IWRITE.EQ.1) WRITE(6,520) (MA(JJ), JJ=1,NGOOD)            LSV05620
  520 FORMAT(/' 1ST GUESS AT APPROPRIATE SIZE T-MATRICES'/          LSV05630
     1 ' ACTUAL VALUES WILL PROBABLY BE 1/4 AGAIN AS MUCH'/(13I6))  LSV05640
C                                                                   LSV05650
C     PRINT OUT TO FILE 10 1ST GUESSES AT SIZES OF THE T-MATRICES TO LSV05660
C     BE USED IN THE SINGULAR VECTOR COMPUTATIONS.                  LSV05670
C     PROGRAM LOOPS ON T-SIZE TO DETERMINE APPROPRIATE SIZE T-MATRIX. LSV05680
      WRITE(10,530) N,KMAX                                          LSV05690
  530 FORMAT(2I8,' = ORDER OF USER MATRIX AND MAX ORDER OF T(1,MEV)') LSV05700
C                                                                   LSV05710
      WRITE(10,540)                                                 LSV05720
  540 FORMAT(/' 1ST GUESS AT APPROPRIATE SIZE T-MATRICES'/          LSV05730
     1 ' ACTUAL VALUES WILL PROBABLY BE 1/4 AGAIN AS MUCH'/)        LSV05740
C                                                                   LSV05750
      WRITE(10,550)                                                 LSV05760
  550 FORMAT(4X,'J',7X,'GOODSV(J)',4X,'M1(J)',1X,'M2(J)',1X,'MA(J)') LSV05770
C                                                                   LSV05780
      WRITE(10,560) (J,GOODSV(J),M1(J),M2(J), MA(J), J=1,NGOOD)     LSV05790
  560 FORMAT(I5,E19.12,3I6)                                         LSV05800
C                                                                   LSV05810
      IF(MBOUND.EQ.1) WRITE(10,570)                                 LSV05820
  570 FORMAT(/' GOODSV(J) IS A GOOD EIGENVALUE OF T(1,MEV)'/        LSV05830
     1 ' M1 = SMALLEST VALUE OF M SUCH THAT T(1,M) HAS AT LEAST'/   LSV05840
     1 '      ONE EIGENVALUE IN THE INTERVAL (SV-TOLN,SV+TOLN)'/    LSV05850
     1 '      TOLN(J) = DMAX1(GOODSV(J)*RELTOL, SCALE0*MULTOL)'/    LSV05860
     1 ' M2 = SMALLEST M (IF ANY) SUCH THAT IN THE ABOVE INTERVAL'/ LSV05870
     1 '      T(1,M) HAS AT LEAST TWO EIGENVALUES '/                LSV05880
     1 ' INITIAL VALUE OF MA(J) IS CHOSEN HEURISTICALLY'/           LSV05890
     1 ' PROGRAM LOOPS ON SIZE OF T-MATRIX TO GET APPROPRIATE SIZE'/ LSV05900
     1 ' END OF SIZES OF T-MATRICES FILE 10'///)                    LSV05910
C                                                                   LSV05920
C                                                                   LSV05930
C     TERMINATE AFTER COMPUTING 1ST GUESSES AT  SIZES OF THE        LSV05940
```

```
C     T-MATRICES REQUIRED FOR THE GIVEN SINGULAR VALUES?              LSV05950
      IF(MBOUND.EQ.1) GO TO 1700                                      LSV05960
C                                                                     LSV05970
C                                                                     LSV05980
C     IS THERE ROOM FOR ALL OF THE REQUESTED T-EIGENVECTORS?          LSV05990
      MTOL = 0                                                        LSV06000
      DO 580 J = 1,NGOOD                                              LSV06010
      IF(MP(J).EQ.MPMIN) GO TO 580                                    LSV06020
      MTOL = MTOL + IABS(MA(J))                                       LSV06030
  580 CONTINUE                                                        LSV06040
      MTOL = (5*MTOL)/4                                               LSV06050
      IF(MTOL.GT.MDIMTV.AND.NTVCON.EQ.0) GO TO 1720                   LSV06060
C                                                                     LSV06070
C---------------------------------------------------------------------LSV06080
C     GENERATE A RANDOM VECTOR TO BE USED REPEATEDLY BY              LSV06090
C     SUBROUTINE INVERM                                              LSV06100
C                                                                     LSV06110
      IIL = RHSEED                                                    LSV06120
      CALL GENRAN(IIL,G,KMAX)                                         LSV06130
C                                                                     LSV06140
C---------------------------------------------------------------------LSV06150
C                                                                     LSV06160
C     FOR EACH SINGULAR VALUE LOOP ON T-EIGENVECTOR COMPUTATIONS      LSV06170
C     TO COMPUTE AN APPROPRIATE T-EIGENVECTOR TO USE IN THE          LSV06180
C     RITZ VECTOR COMPUTATIONS.                                      LSV06190
C                                                                     LSV06200
      MTOL = 0                                                        LSV06210
      NTVEC = 0                                                       LSV06220
      ILBIS = 0                                                       LSV06230
      DO 770 J = 1,NGOOD                                              LSV06240
      ICOUNT = 0                                                      LSV06250
      ERRMIN = 10.D0                                                  LSV06260
      MABEST = MPMIN                                                  LSV06270
      IF(MP(J).EQ.MPMIN) GO TO 770                                    LSV06280
      TFLAG = 0                                                       LSV06290
      SVAL = GOODSV(J)                                                LSV06300
      TEMP = RELTOL*DABS(SVAL)                                        LSV06310
      UB = SVAL + DMAX1(STUTOL,TEMP)                                  LSV06320
      LB = SVAL - DMAX1(STUTOL,TEMP)                                  LSV06330
      LB = DMAX1(LB,ZERO)                                             LSV06340
  590 KMAXU = IABS(MA(J))                                             LSV06350
C                                                                     LSV06360
C     SELECT A SUITABLE INCREMENT FOR THE ORDERS OF THE T-MATRICES    LSV06370
C     TO BE CONSIDERED IN DETERMINING APPROPRIATE SIZES FOR THE RITZ  LSV06380
C     VECTOR COMPUTATIONS.  ALL ORDERS CONSIDERED MUST BE EVEN.       LSV06390
      IF(ICOUNT.GT.0) GO TO 610                                       LSV06400
C     SELECT IDELTA(J) BASED UPON THE T-MULTIPLICITY OBTAINED         LSV06410
      IF(M2(J).EQ.KMAX) GO TO 600                                     LSV06420
C     M2 DETERMINED                                                   LSV06430
      IDEL = ((3*M1(J) + 5*M2(J))/8  +  1 - IABS(MA(J)))/10  +  1     LSV06440
      IF((IDEL/2)*2.NE.IDEL) IDEL = IDEL + 1                          LSV06450
      IDELTA(J) = IDEL                                                LSV06460
      GO TO 610                                                       LSV06470
C     M2 NOT DETERMINED                                               LSV06480
  600 MAMAX = MINO((11*MEV)/8 + 12, (13*M1(J))/8 + 1)                 LSV06490
```

```
      IDEL = (MAMAX - IABS(MA(J)))/10  +  1                     LSV06500
      IF((IDEL/2)*2.NE.IDEL)  IDEL = IDEL + 1                   LSV06510
      IDELTA(J) = IDEL                                          LSV06520
  610 ICOUNT = ICOUNT + 1                                       LSV06530
C                                                               LSV06540
C-----------------------------------------------------------------LSV06550
C     TO MIMIMIZE THE EFFECT OF THE ONE-SIDED ACCEPTANCE TEST FOR LSV06560
C     EIGENVALUES IN THE BISEC SUBROUTINE, RECOMPUTE THE GIVEN   LSV06570
C     SINGULAR VALUE AT THE SPECIFIED KMAXU                      LSV06580
C                                                               LSV06590
      CALL LBISEC(BETA,EPSM,SVAL,SVALN,LB,UB,TTOL,KMAXU,NEVT)   LSV06600
C                                                               LSV06610
C-----------------------------------------------------------------LSV06620
C                                                               LSV06630
C     CHECK WHETHER OR NOT GIVEN T-MATRIX HAS AN EIGENVALUE IN THE LSV06640
C     SPECIFIED INTERVAL AND IF SO WHAT ITS T-MULTIPLICITY IS.   LSV06650
C                                                               LSV06660
      IF(NEVT.EQ.1) GO TO 650                                   LSV06670
      IF(NEVT.NE.0)  GO TO 630                                  LSV06680
      ILBIS = 1                                                 LSV06690
      WRITE(6,620) SVAL,KMAXU                                   LSV06700
  620 FORMAT(/' PROBLEM ENCOUNTERED IN RECOMPUTATION OF USER-SUPPLIED SILSV06710
     1NGULAR VALUE',E20.12/' THE SIZE T-MATRIX SPECIFIED',I6,' DOES NOT LSV06720
     1HAVE A SINGULAR VALUE IN THE INTERVAL SPECIFIED'/' INCREASE SIZE ALSV06730
     1ND TRY AGAIN'/)                                           LSV06740
      GO TO 670                                                 LSV06750
C                                                               LSV06760
  630 IF(NEVT.GT.1)  WRITE(6,640) SVAL,KMAXU                    LSV06770
  640 FORMAT(/' PROBLEM ENCOUNTERED IN RECOMPUTATION OF USER-SUPPLIED   LSV06780
     1SINGULAR VALUE',E20.12/' FOR THE SIZE T-MATRIX SPECIFIED =',I6,' TLSV06790
     1HE GIVEN SINGULAR VALUE IS T-MULTIPLE IN THE INTERVAL SPECIFIED'/'LSV06800
     1SOMETHING IS WRONG, THEREFORE NO SINGULAR VECTORS WILL BE COMPUTEDLSV06810
     1 FOR THIS SINGULAR VALUE'/)                               LSV06820
C                                                               LSV06830
      MP(J) = MPMIN                                             LSV06840
      MA(J) = -2*KMAX                                           LSV06850
      GO TO 770                                                 LSV06860
C                                                               LSV06870
  650 CONTINUE                                                  LSV06880
      ILBIS = 0                                                 LSV06890
C                                                               LSV06900
C                                                               LSV06910
      SVNEW(J) = SVALN                                          LSV06920
      SVAL = SVALN                                              LSV06930
      MTOL = MTOL+KMAXU                                         LSV06940
C                                                               LSV06950
C     IS THERE ROOM IN TVEC ARRAY FOR THE NEXT T-EIGENVECTOR?   LSV06960
C     IF NOT, SKIP TO RITZ VECTOR COMPUTATIONS.                 LSV06970
      IF (MTOL.GT.MDIMTV) GO TO 780                             LSV06980
C                                                               LSV06990
      IT = 3                                                    LSV07000
      KINT = MTOL - KMAXU +1                                    LSV07010
C                                                               LSV07020
C     RECORD THE BEGINNING AND END OF THE T-EIGENVECTOR BEING COMPUTED LSV07030
      MINT(J) = KINT                                            LSV07040
```

```
      MFIN(J) = MTOL                                            LSV07050
C                                                               LSV07060
C-------------------------------------------------------------LSV07070
C     SUBROUTINE INVERM DOES INVERSE ITERATION, I.E. SOLVES     LSV07080
C     (T(1,KMAXU) - SVAL)*U = RHS  FOR EACH SINGULAR VALUE TO    LSV07090
C     OBTAIN THE DESIRED T-EIGENVECTOR.                         LSV07100
C                                                               LSV07110
      IF(IWRITE.EQ.1)  WRITE(6,660) J                           LSV07120
  660 FORMAT(/I6,'TH SINGULAR VALUE ')                          LSV07130
C                                                               LSV07140
      CALL INVERM(BETA,V1,TVEC(KINT),SVAL,ERROR,TERROR,EPSM,G,KMAXU, LSV07150
     1 IT,IWRITE)                                               LSV07160
C                                                               LSV07170
C-------------------------------------------------------------LSV07180
C                                                               LSV07190
      TERR(J) = TERROR                                          LSV07200
      TLAST(J) = ERROR                                          LSV07210
      KMAXU1 = KMAXU + 1                                        LSV07220
      TBETA(J) = BETA(KMAXU1)*ERROR                             LSV07230
C                                                               LSV07240
C     AFTER COMPUTING EACH OF THE T-EIGENVECTORS,               LSV07250
C     CHECK THE SIZE OF THE ERROR ESTIMATE, ERROR.              LSV07260
C     IF THIS ESTIMATE IS NOT AS SMALL AS DESIRED AND           LSV07270
C     |MA(J)| < ML(J), ATTEMPT TO INCREASE THE SIZE OF |MA(J)|  LSV07280
C     AND REPEAT THE T-EIGENVECTOR COMPUTATIONS.                LSV07290
C                                                               LSV07300
      IF(ERROR.LT.ERTOL.OR.TFLAG.EQ.1)  GO TO 760               LSV07310
C                                                               LSV07320
      IF(ERROR.GE.ERRMIN) GO TO 670                             LSV07330
C     LAST COMPONENT IS LESS THAN MINIMAL TO DATE               LSV07340
      ERRMIN = ERROR                                            LSV07350
      MABEST = MA(J)                                            LSV07360
  670 CONTINUE                                                  LSV07370
C                                                               LSV07380
      IF(MA(J).GT.0)  ITEST = MA(J) + IDELTA(J)                 LSV07390
      IF(MA(J).LT.0)  ITEST = -(IABS(MA(J)) + IDELTA(J))        LSV07400
      IF(IABS(ITEST).LE.ML(J).AND.ICOUNT.LE.10) GO TO 690       LSV07410
C     NEW MA(J) IS GREATER THAN MAXIMUM ALLOWED.                LSV07420
      IF(ERCONT.EQ.0.OR.MABEST.EQ.MPMIN) GO TO 710              LSV07430
      TFLAG = 1                                                 LSV07440
      MA(J) = MABEST                                            LSV07450
      IF(ILBIS.EQ.0)   MTOL = MTOL - KMAXU                      LSV07460
      WRITE(6,680) MA(J)                                        LSV07470
  680 FORMAT(' 10 ORDERS WERE CONSIDERED.  NONE SATISFIED THE ERROR TESTLSV07480
     1'/' THEREFORE USE THE BEST ORDER OBTAINED FOR THE T-EIGENVECTORS' LSV07490
     1,I6)                                                      LSV07500
      GO TO 590                                                 LSV07510
C                                                               LSV07520
  690 MA(J) = ITEST                                             LSV07530
C                                                               LSV07540
      MT = IABS(MA(J))                                          LSV07550
      IF(IWRITE.EQ.1.AND.ILBIS.EQ.0) WRITE(6,700) MT            LSV07560
  700 FORMAT(/' CHANGE SIZE OF T-MATRIX TO ',I6,' RECOMPUTE T-EIGENVECTOLSV07570
     1R')                                                       LSV07580
C                                                               LSV07590
```

```
      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')                                          LSV08700
C                                                               LSV08710
C     COUNT THE NUMBER OF RITZ VECTORS NOT BEING COMPUTED       LSV08720
      MREJEC = 0                                                LSV08730
      DO 930 J=1,NGOODC                                         LSV08740
  930 IF(MP(J).EQ.MPMIN)  MREJEC = MREJEC + 1                   LSV08750
      MREJET = MREJEC + (NGOOD-NGOODC)                          LSV08760
      IF(MREJET.NE.0) WRITE(6,940) MREJET                       LSV08770
  940 FORMAT(/' RITZ VECTORS ARE NOT COMPUTED FOR',I6,' OF THE SINGULAR LSV08780
     1VALUES'/)                                                 LSV08790
      NACT = NGOODC - MREJEC                                    LSV08800
      WRITE(6,950) NGOOD,NTVEC,NACT                             LSV08810
  950 FORMAT(/I6,' RITZ VECTORS WERE REQUESTED'/I6,' T-EIGENVECTORS WERELSV08820
     1 COMPUTED'/I6,' RITZ VECTORS WILL BE COMPUTED'/)          LSV08830
C     CHECK IF THERE ARE ANY RITZ VECTORS TO COMPUTE            LSV08840
      IF(MREJEC.EQ.NGOODC) GO TO 1780                           LSV08850
C                                                               LSV08860
C     CONTINUE WITH THE LANCZOS VECTOR COMPUTATIONS?            LSV08870
      IF(LVCONT.EQ.0.AND.MREJEC.NE.0) GO TO 1760                LSV08880
C                                                               LSV08890
C     NOW COMPUTE THE RITZ VECTORS.  REGENERATE THE             LSV08900
C     LANCZOS VECTORS.                                          LSV08910
C                                                               LSV08920
      DO 960 I = 1,MNMAX                                        LSV08930
  960 RITVEC(I) = ZERO                                          LSV08940
C                                                               LSV08950
C     REGENERATE THE STARTING VECTOR. THIS MUST BE GENERATED AND LSV08960
C     NORMALIZED PRECISELY THE WAY IT WAS DONE IN THE CORRESPONDING LSV08970
C     SINGULAR VALUE COMPUTATIONS, OTHERWISE THERE WILL BE A    LSV08980
C     MISMATCH BETWEEN THE T-EIGENVECTORS THAT HAVE BEEN COMPUTED LSV08990
C     FROM THE T-MATRICES READ IN FROM FILE 2 (IF THEY WERE READ IN) LSV09000
C     AND THE LANCZOS TRIDIAGONAL MATRICES  THAT ARE BEING REGENERATED. LSV09010
C                                                               LSV09020
C     STARTING VECTORS ARE OF THE FORM (V1,0) OR (0,V2) WHERE V1 IS LSV09030
C     OF LENGTH M AND V2 IS OF LENGTH N.  SUCCEEDING LANCZOS VECTORS LSV09040
C     ALTERNATE BETWEEN THESE TWO FORMS AND THE DIAGONAL ENTRIES OF THE LSV09050
C     T-MATRICES ALL VANISH.  THE PARAMETER IPARO DETERMINES THE SHAPE LSV09060
C     OF THE STARTING VECTOR.  IF IPARO=1, THEN STARTING VECTOR WAS LSV09070
C     OF THE FORM (0,V2).   IF IPARO=2, THEN STARTING VECTOR WAS OF LSV09080
C     THE FORM (V1,0).                                          LSV09090
C     REGENERATE STARTING VECTOR                                LSV09100
      BATA = ZERO                                               LSV09110
      IPAR = IPARO                                              LSV09120
      ITNUM = 1                                                 LSV09130
      IF (IPAR.EQ.2) GO TO 1020                                 LSV09140
C                                                               LSV09150
C----------------------------------------------------------------LSV09160
C     IPAR = 1 SO SET V2 TO RANDOM UNIT VECTOR AND SET V1 = 0.  LSV09170
C                                                               LSV09180
      IIL = SVSEED                                              LSV09190
      CALL GENRAN(IIL,G,N)                                      LSV09200
C                                                               LSV09210
C----------------------------------------------------------------LSV09220
C                                                               LSV09230
      DO 970 J = 1,N                                            LSV09240
```

```
   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                                                                    LSV10350
C       DO ONE ITERATION OF LANCZOS WHERE NEW LANCZOS VECTOR WILL HAVE  LSV10360
C       THE FORM  (V1,0).                                            LSV10370
C                                                                    LSV10380
C-------------------------------------------------------------------LSV10390
C                                                                    LSV10400
        CALL SVMAT(V2,V1,BATA)                                      LSV10410
C                                                                    LSV10420
C-------------------------------------------------------------------LSV10430
C                                                                    LSV10440
C-------------------------------------------------------------------LSV10450
        BATA = DSQRT(FINPRO(M,V1,1,V1,1))                           LSV10460
C-------------------------------------------------------------------LSV10470
        SUM = ONE/BATA                                              LSV10480
        ITNUM = ITNUM + 1                                           LSV10490
        IPAR = 1                                                    LSV10500
C                                                                    LSV10510
        TEMP = BETA(ITNUM)                                          LSV10520
        TEMP = DABS(BATA - TEMP)/TEMP                               LSV10530
        IF (TEMP.GE.1.0D-10) GO TO 1090                             LSV10540
C                                                                    LSV10550
C       NORMALIZE LANCZOS VECTOR                                    LSV10560
        DO 1160 J = 1,M                                             LSV10570
 1160 V1(J) = V1(J)*SUM                                             LSV10580
C                                                                    LSV10590
C       UPDATE RITZ VECTORS                                         LSV10600
        DO 1180 J = 1,NGOODC                                        LSV10610
        IF (IABS(MA(J)).LT.ITNUM.OR.MP(J).EQ.MPMIN) GO TO 1180      LSV10620
        LL = MN*(J-1)                                               LSV10630
        II = MINT(J) + ITNUM - 1                                    LSV10640
        TEMP = TVEC(II)                                             LSV10650
C                                                                    LSV10660
        DO 1170 K = 1,M                                             LSV10670
        LL = LL + 1                                                 LSV10680
 1170 RITVEC(LL) = TEMP*V1(K) + RITVEC(LL)                          LSV10690
C                                                                    LSV10700
 1180 CONTINUE                                                      LSV10710
C  HAVE ALL REQUIRED LANCZOS VECTORS BEEN COMPUTED ?                LSV10720
        IF (ITNUM.LT.KMAXU) GO TO 1080                              LSV10730
C                                                                    LSV10740
 1190 CONTINUE                                                      LSV10750
C                                                                    LSV10760
C       RITZVECTOR GENERATION IS COMPLETE. NORMALIZE EACH RITZVECTOR  LSV10770
C       AS AN EIGENVECTOR OF THE ASSOCIATED SYMMETRIC MATRIX B.     LSV10780
C       THEN COMPUTE THE ERRORS IN THESE VECTORS AS EIGENVECTORS    LSV10790
C       OF B AND WRITE THESE OUT TO FILE 9.   THEN INDIVIDUALLY     LSV10800
C       NORMALIZE THE FIRST M AND THE LAST N COMPONENTS OF EACH OF  LSV10810
C       THESE RITZ VECTORS AND TAKE THESE NORMALIZED VECTORS AS     LSV10820
C       RESPECTIVELY APPROXIMATIONS TO THE LEFT AND TO THE RIGHT    LSV10830
C       SINGULAR VECTORS OF THE CORRESPONDING SINGULAR VALUE OF     LSV10840
C       THE ORIGINAL MATRIX.                                        LSV10850
C                                                                    LSV10860
C                                                                    LSV10870
C       NORMALIZE THE RITZ VECTORS AS EIGENVECTORS OF B             LSV10880
        DO 1280 J = 1,NGOODC                                        LSV10890
```

```
         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        LSV11450
C      VECTORS.                                                         LSV11460
C                                                                       LSV11470
       WRITE(9,1290)                                                    LSV11480
 1290 FORMAT(11X,'GOODSV(J)',3X,'MA(J)',2X,'  BMINGAP',6X,' BERROR',2X, LSV11490
     1 'BERROR/BGAP',4X,' TERROR')                                      LSV11500
C                                                                       LSV11510
       WRITE(13,1300)                                                   LSV11520
 1300 FORMAT(11X,'GOODSV(J)',5X,'RITZNORM',5X,' BMINGAP',5X,'TBETA(J)', LSV11530
     1 5X,'TLAST(J)')                                                   LSV11540
C                                                                       LSV11550
       DO 1330 J=1,NGOODC                                               LSV11560
C                                                                       LSV11570
       IF(MP(J).EQ.MPMIN) GO TO 1330                                    LSV11580
C                                                                       LSV11590
       WRITE(9,1310)SVNEW(J),MA(J),BMINGP(J),BERR(J),BERRGP(J),TERR(J)  LSV11600
 1310 FORMAT(E20.12,I6,4E13.5)                                          LSV11610
C                                                                       LSV11620
       WRITE(13,1320) SVNEW(J),RNORM(J),BMINGP(J),TBETA(J),TLAST(J)     LSV11630
 1320 FORMAT(E20.12,4E13.5)                                             LSV11640
C                                                                       LSV11650
 1330 CONTINUE                                                          LSV11660
C                                                                       LSV11670
       IF (MREJEC.EQ.0) GO TO 1410                                      LSV11680
C                                                                       LSV11690
       WRITE(9,1340)                                                    LSV11700
 1340 FORMAT(/' RITZ VECTORS WERE NOT COMPUTED FOR THE FOLLOWING SINGULALSV11710
     1R VALUES'/' EITHER BECAUSE THEY HAD NOT CONVERGED OR BECAUSE THE ELSV11720
     1RROR ESTIMATE'/'  WAS NOT AS SMALL AS DESIRED'/)                  LSV11730
C                                                                       LSV11740
       WRITE(13,1350)                                                   LSV11750
 1350 FORMAT(/' RITZ VECTORS WERE NOT COMPUTED FOR THE FOLLOWING SINGULALSV11760
     1R VALUES'/' EITHER BECAUSE THEY HAD NOT CONVERGED OR BECAUSE THE ELSV11770
     1ERROR ESTIMATE'/' WAS NOT AS SMALL AS DESIRED'/)                  LSV11780
C                                                                       LSV11790
       DO 1400 J = 1,NGOODC                                             LSV11800
       IF(MP(J).NE.MPMIN) GO TO 1400                                    LSV11810
C      EACH SINGULAR VALUE FOR WHICH NO SINGULAR VECTOR WAS CALCULATED  LSV11820
C      HAS INFORMATION OUTPUTTED TO FILES 4 AND 13                      LSV11830
C                                                                       LSV11840
       WRITE(9,1360)                                                    LSV11850
 1360 FORMAT(6X,'GOODSV(J)',3X,'MA(J)',5X,'BMINGP(J)',6X,'TLAST(J)',    LSV11860
     1 6X,'TBETA(J)',3X,'MP(J)')                                        LSV11870
       WRITE(9,1370) GOODSV(J),MA(J),BMINGP(J),TLAST(J),TBETA(J),MP(J)  LSV11880
 1370 FORMAT(E15.8,I8,3E14.4,I8)                                        LSV11890
C                                                                       LSV11900
       WRITE(13,1380)                                                   LSV11910
 1380 FORMAT(6X,'GOODSV(J)',3X,'MA(J)',3X,'M1(J)',3X,'M2(J)',3X,'MP(J)' LSV11920
     1/)                                                                LSV11930
       WRITE(13,1390) GOODSV(J),MA(J),M1(J),M2(J),MP(J)                 LSV11940
 1390 FORMAT(E15.8,4I8)                                                 LSV11950
C                                                                       LSV11960
 1400 CONTINUE                                                          LSV11970
C                                                                       LSV11980
 1410 CONTINUE                                                          LSV11990
```

```
C                                                              LSV12000
      WRITE(9,1420)                                            LSV12010
 1420 FORMAT(/' ABOVE ARE ERROR ESTIMATES FOR THE B AND T EIGENVECTORS'/LSV12020
     1 ' ASSOCIATED WITH THE GOODSV LISTED IN COLUMN 1'/        LSV12030
     1 ' BERROR = NORM(B*X - SV*X)   TERROR = NORM(T*Y - SV*Y) '/LSV12040
     1 ' WHERE T = T(1,MA(J))    X = RITZ VECTOR = V*Y  V = SUCCESSIVE'/LSV12050
     1 ' LANCZOS VECTORS. BMINGAP = GAP TO NEAREST B-EIGENVALUE'//)  LSV12060
C                                                              LSV12070
      WRITE(13,1430)                                           LSV12080
 1430 FORMAT(/' ABOVE ARE ERROR ESTIMATES FOR THE GOODSVS'/    LSV12090
     1 ' RITZNORM = NORM(COMPUTED RITZ VECTOR FOR B-MATRIX'/   LSV12100
     1 ' TBETA(J) = BETA(MA(J)+1)*Y(MA(J)),  T*Y = SV*Y '/     LSV12110
     1 ' TLAST(J) = DABS(Y(MA(J)) '/)                          LSV12120
C                                                              LSV12130
C     NUMBER OF RITZ VECTORS COMPUTED                          LSV12140
      NCOMPU = NGOODC - MREJEC                                 LSV12150
      WRITE(12,1440) N,NCOMPU,NGOODC,MATNO                     LSV12160
 1440 FORMAT(3I6,I12,' SIZE A, NO.RITZVECS, NO.SVALUES,MATNO') LSV12170
C                                                              LSV12180
C     INDIVIDUALLY NORMALIZE THE FIRST M AND THE LAST N COMPONENTS OF  LSV12190
C     EACH RITZ VECTOR.                                        LSV12200
C                                                              LSV12210
      LFIN = 0                                                 LSV12220
      DO 1560 J = 1,NGOODC                                     LSV12230
C                                                              LSV12240
      IF(MP(J).EQ.MPMIN) GO TO 1540                            LSV12250
C                                                              LSV12260
C     RITZ VECTOR WAS COMPUTED                                 LSV12270
      LINT = MN*(J-1) + 1                                      LSV12280
      LFIN = MN*J                                              LSV12290
      LFIN1 = LINT + M - 1                                     LSV12300
      LINT1 = LFIN1 + 1                                        LSV12310
C                                                              LSV12320
      SUM = 0.D0                                               LSV12330
      TEMP = 0.D0                                              LSV12340
      DO 1450 I = LINT,LFIN1                                   LSV12350
 1450 SUM = SUM + RITVEC(I)*RITVEC(I)                          LSV12360
      SUM = ONE/DSQRT(SUM)                                     LSV12370
      DO 1460 I = LINT,LFIN1                                   LSV12380
 1460 RITVEC(I) = SUM*RITVEC(I)                                LSV12390
      DO 1470 I = LINT1,LFIN                                   LSV12400
 1470 TEMP = TEMP + RITVEC(I)*RITVEC(I)                        LSV12410
      TEMP = ONE/DSQRT(TEMP)                                   LSV12420
      DO 1480 I = LINT1,LFIN                                   LSV12430
 1480 RITVEC(I) = TEMP*RITVEC(I)                               LSV12440
C                                                              LSV12450
      WRITE(12,1490) J, GOODSV(J), MP(J)                       LSV12460
 1490 FORMAT(/I6,4X,E20.12,I6,' J, SINGULAR VALUE, MP(J)')     LSV12470
C                                                              LSV12480
      WRITE(12,1500) BERR(J),BERRGP(J)                         LSV12490
 1500 FORMAT(2E15.5,' = NORM(B*Z-SVAL*Z) AND  NORM(B*Z-SVAL*Z)/BMINGAP')LSV12500
C                                                              LSV12510
      WRITE(12,1510) J                                         LSV12520
 1510 FORMAT(/I6,'TH LEFT SINGULAR VECTOR'/)                   LSV12530
C     WRITE(12,170) (RITVEC(LL), LL=LINT,LFIN1)                LSV12540
```

```
      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)                                                  LSV13100
 1650 FORMAT(/' PARAMETERS IN BETA FILE DO NOT AGREE WITH THOSE SPECIFIELSV13110
     1D BY THE USER.'/' THEREFORE, THE PROGRAM TERMINATES FOR THE USER TLSV13120
     10 RESOLVE THE DIFFERENCES'/)                                   LSV13130
C                                                                    LSV13140
      GO TO 1860                                                     LSV13150
C                                                                    LSV13160
 1660 WRITE(6,1670) KMAX,MEV                                         LSV13170
 1670 FORMAT(/' IN BETA HISTORY HEADER KMAX =',I6/                   LSV13180
     1' BUT SINGULAR VALUES WERE COMPUTED AT MEV = ',I6,' PROGRAM STOPS'LSV13190
     1)                                                              LSV13200
C                                                                    LSV13210
      GO TO 1860                                                     LSV13220
C                                                                    LSV13230
 1680 WRITE(6,1690) MEV                                              LSV13240
 1690 FORMAT(/' SOMETHING IS WRONG.'/' HEADER SAYS THAT SIZE T-MATRIX USLSV13250
     1ED IN THE SINGULAR VALUE COMPUTATIONS WAS = ',I6/' BUT THIS IS AN LSV13260
     10DD ORDER AND THAT IS NOT ALLOWED.  PROGRAM STOPS'/)           LSV13270
C                                                                    LSV13280
      GO TO 1860                                                     LSV13290
C                                                                    LSV13300
 1700 WRITE(6,1710)                                                  LSV13310
 1710 FORMAT(/' PROGRAM COMPUTED 1ST GUESSES AT T-MATRIX SIZES, READ THELSV13320
     1M TO FILE 10'/' THEN TERMINATED AS REQUESTED.')                LSV13330
      GO TO 1860                                                     LSV13340
C                                                                    LSV13350
 1720 WRITE(6,1730) MTOL, MDIMTV                                     LSV13360
 1730 FORMAT(/' PROGRAM TERMINATES BECAUSE THE TVEC DIMENSION ANTICIPATELSV13370
     1D',I7/'  IS LARGER THAN THE TVEC DIMENSION',I7,' SPECIFIED BY THE LSV13380
     1USER.'/'  USER MAY RESET THE TVEC DIMENSION AND RESTART THE PROGRALSV13390
     1M'/)                                                           LSV13400
      GO TO 1860                                                     LSV13410
C                                                                    LSV13420
 1740 WRITE(6,1750)                                                  LSV13430
 1750 FORMAT(/' PROGRAM TERMINATES BECAUSE NO SUITABLE T-EIGENVECTORS WELSV13440
     1RE IDENTIFIED'/' FOR ANY OF THE SINGULAR VALUES SUPPLIED.  PROBLEMLSV13450
     1 COULD BE CAUSED BY'/' TOO SMALL A TVEC DIMENSION OR SIMPLY BE THALSV13460
     1T NO SUITABLE T-VECTORS'/' WERE IDENTIFIED.  USER SHOULD CHECK OUTLSV13470
     1PUT'/)                                                         LSV13480
      GO TO 1860                                                     LSV13490
C                                                                    LSV13500
 1760 WRITE(6,1770) LVCONT,NTVEC,NGOOD                               LSV13510
 1770 FORMAT(/' LVCONT FLAG =',I2,' AND NUMBER ',I5,' OF T-EIGENVECTORS LSV13520
     1 COMPUTED N.E.'/' NUMBER',I5,' REQUESTED SO PROGRAM TERMINATES'/) LSV13530
      GO TO 1860                                                     LSV13540
 1780 WRITE(6,1790)                                                  LSV13550
 1790 FORMAT(/' PROGRAM TERMINATES WITHOUT COMPUTING ANY RITZ VECTORS'/ LSV13560
     1/' BECAUSE ALL OF THE T-EIGENVECTORS WERE REJECTED AS NOT SUITABLELSV13570
     1 FOR'/' THE RITZ VECTOR COMPUTATIONS.  PROBABLE CAUSE WAS LACK OF LSV13580
     1CONVERGENCE'/' OF THE SINGULAR VALUES'/)                       LSV13590
      GO TO 1860                                                     LSV13600
C                                                                    LSV13610
 1800 WRITE(6,1810)                                                  LSV13620
 1810 FORMAT(/' PROGRAM INDICATES THAT IT IS NOT POSSIBLE TO COMPUTE ANYLSV13630
     1 OF THE'/' REQUESTED EIGENVECTORS. THEREFORE PROGRAM TERMINATES') LSV13640
```

```
      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----------------------------------------------------LSM00010
C  Authors:  Jane Cullum and Ralph A. Willoughby (Deceased)      LSM00020
C            Los Alamos National Laboratory                      LSM00030
C            Los Alamos, New Mexico 87544                        LSM00040
C                                                                LSM00050
C            E-mail:  cullumj@lanl.gov                           LSM00060
C                                                                LSM00070
C  These codes are copyrighted by the authors.  These codes      LSM00080
C  and modifications of them or portions of them are NOT to be   LSM00090
C  incorporated into any commercial codes or used for any other  LSM00100
C  commercial purposes such as consulting for other companies,   LSM00110
C  without legal agreements with the authors of these Codes.     LSM00120
C  If these Codes or portions of them are used in other scientific or LSM00130
C  engineering research works the names of the authors of these codes LSM00140
C  and appropriate references to their written work are to be    LSM00150
C  incorporated in the derivative works.                         LSM00160
C                                                                LSM00170
C  This header is not to be removed from these codes.            LSM00180
C                                                                LSM00190
C         REFERENCE: Cullum and Willoughby, Chapter 5            LSM00191
C         Lanczos Algorithms for Large Symmetric Eigenvalue ComputationsLSM00192
C         VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in LSM00193
C         Applied Mathematics, 2002. SIAM Publications,          LSM00194
C         Philadelphia, PA. USA                                  LSM00195
C                                                                LSM00196
C                                                                LSM00197
C                                                                LSM00200
C     CONTAINS SUBROUTINES LANCZS, USPECS, STRAN, AND SVMAT      LSM00210
C     FOR USE WITH THE LANCZOS SINGULAR VALUE/VECTOR PROGRAMS    LSM00220
C                                                                LSM00230
C     NONPORTABLE CONSTRUCTIONS:                                 LSM00240
C     1.  THE ENTRY MECHANISM USED TO PASS THE STORAGE LOCATIONS LSM00250
C         OF THE USER-SPECIFIED MATRIX FROM THE SUBROUTINE USPEC LSM00260
C         TO THE MATRIX-VECTOR MULTIPLY SUBROUTINES SVMAT AND    LSM00270
C         STRAN.                                                 LSM00280
C     2.  IN THE SAMPLE USPEC PROVIDED:  THE FREE FORMAT (8,*),  LSM00290
C         AND THE FORMAT (20A4).                                 LSM00300
C                                                                LSM00310
C-----START OF LANCZS-------------------------------------------LSM00320
C                                                                LSM00330
      SUBROUTINE LANCZS(MATVEC,MTRAN,BETA,V1,V2,G,KMAX,MOLD1,     LSM00340
     1               M,N,IPAR,IIX)                                LSM00350
C                                                                LSM00360
C--------------------------------------------------------------LSM00370
      DOUBLE PRECISION BETA(1),V1(1),V2(1),SUM,TEMP,ONE,ZERO     LSM00380
      REAL G(1)                                                  LSM00390
      DOUBLE PRECISION FINPRO                                    LSM00400
      INTEGER IPAR                                               LSM00410
      EXTERNAL MATVEC,MTRAN                                      LSM00420
C--------------------------------------------------------------LSM00430
```

```
C     COMPUTE T(1,MEV) FOR SYMMETRIZED VERSION OF GIVEN A-MATRIX.      LSM00440
C                                                                      LSM00450
C                                        ----            ----          LSM00460
C                                      |  0             A  |           LSM00470
C                       B    =         |                   |           LSM00480
C                                      |  A-TRANSPOSE    0  |           LSM00490
C                                        ----            ----          LSM00500
C                                                                      LSM00510
C     WHERE A IS AN M BY N REAL SPARSE MATRIX, USING STARTING          LSM00520
C     VECTORS OF THE FORM (V1,0) WHEN THE FLAG IPAR = 2 AND            LSM00530
C     OF THE FORM (0,V2) WHEN THE FLAG IPAR = 1.  V1 IS OF             LSM00540
C     DIMENSION M, THE ROW DIMENSION OF A, AND V2 IS OF DIMENSION      LSM00550
C     N, THE COLUMN DIMENSION OF A.                                    LSM00560
C                                                                      LSM00570
C     WITH STARTING VECTORS OF THESE FORMS, THE LANCZOS VECTORS        LSM00580
C     GENERATED ALTERNATE BETWEEN THESE 2 FORMS AND ALL OF THE         LSM00590
C     DIAGONAL ENTRIES OF THE LANCZOS TRIDIAGONAL MATRICES T(1,MEV)    LSM00600
C     GENERATED ARE 0.                                                 LSM00610
C                                                                      LSM00620
C     LANCZS USES 2 USER-SUPPLIED SUBROUTINES MATVEC AND MTRAN.        LSM00630
C     MAIN PROGRAM CALLS THESE SVMAT AND STRAN, RESPECTIVELY.          LSM00640
C     CALLING SEQUENCES ARE                                            LSM00650
C                                                                      LSM00660
C          CALL MATVEC(V2,V1,SUM)                                      LSM00670
C          CALL MTRAN(V1,V2,SUM)                                       LSM00680
C                                                                      LSM00690
C     MATVEC  COMPUTES V1 = A*V2 - SUM*V1.                             LSM00700
C     MTRAN COMPUTES V2 = (A-TRANSPOSE)*V1 - SUM*V2.                   LSM00710
C                                                                      LSM00720
C     ON EXIT V1 AND V2 CONTAIN THE NONZERO PARTS OF THE               LSM00730
C     LAST TWO LANCZOS VECTORS.                                        LSM00740
C                                                                      LSM00750
C     IF MOLD1 = 1 THEN T(1,KMAX) IS GENERATED FROM SCRATCH.          LSM00760
C     IF MOLD1 > 1 THEN A PREVIOUSLY-GENERATED T-MATRIX OF SIZE        LSM00770
C     (MOLD1-1) IS EXTENDED TO ONE OF SIZE KMAX. SINGULAR VALUE        LSM00780
C     PRGORAMS CAN ONLY UTILIZE T-MATRICES OF EVEN ORDER.              LSM00790
C     BETA(KMAX+1) IS ALSO COMPUTED FOR USE IN THE ERROR ESTIMATES.    LSM00800
C                                                                      LSM00810
C----------------------------------------------------------------------LSM00820
      ONE  = 1.0D0                                                     LSM00830
      ZERO = 0.0D0                                                     LSM00840
      ITNUM = MOLD1                                                    LSM00850
C                                                                      LSM00860
      IF (ITNUM .GT. 1) GO TO (80,100), IPAR                          LSM00870
C                                                                      LSM00880
C     NO PREVIOUS BETA HISTORY                                         LSM00890
      BETA(1) = ZERO                                                   LSM00900
      IIL = IIX                                                        LSM00910
      IF (IPAR .EQ. 2) GO TO 40                                       LSM00920
C                                                                      LSM00930
C----------------------------------------------------------------------LSM00940
C     IPAR = 1 SO SET V2 EQUAL TO A UNIT RANDOM VECTOR AND SET V1 = 0. LSM00950
      CALL GENRAN(IIL,G,N)                                             LSM00960
C----------------------------------------------------------------------LSM00970
C                                                                      LSM00980
```

```
      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                                                                       LSM01540
      IPAR = 2                                                          LSM01550
      IF (ITNUM .GT. KMAX) GO TO 120                                    LSM01560
C                                                                       LSM01570
C     BELOW IS START FOR MOLD1 > 1 AND IPAR = 2                         LSM01580
C     DO ONE ITERATION OF LANCZOS TO OBTAIN (V1,0)                      LSM01590
C                                                                       LSM01600
  100 CONTINUE                                                          LSM01610
      SUM = BETA(ITNUM)                                                 LSM01620
C                                                                       LSM01630
C-----------------------------------------------------------------------LSM01640
      CALL MATVEC(V2,V1,SUM)                                            LSM01650
C-----------------------------------------------------------------------LSM01660
C                                                                       LSM01670
C-----------------------------------------------------------------------LSM01680
      SUM = FINPRO(M,V1(1),1,V1(1),1)                                   LSM01690
C-----------------------------------------------------------------------LSM01700
C                                                                       LSM01710
      ITNUM = ITNUM + 1                                                 LSM01720
      BETA(ITNUM) = DSQRT(SUM)                                          LSM01730
      SUM = ONE/BETA(ITNUM)                                            LSM01740
C                                                                       LSM01750
      DO 110 J = 1,M                                                    LSM01760
  110 V1(J)= V1(J) * SUM                                                LSM01770
C                                                                       LSM01780
      IPAR = 1                                                          LSM01790
      IF (ITNUM .GT. KMAX) GO TO 120                                    LSM01800
      GO TO 80                                                          LSM01810
C                                                                       LSM01820
  120 CONTINUE                                                          LSM01830
C                                                                       LSM01840
      RETURN                                                            LSM01850
C-----END OF LANCZS-----------------------------------------------------LSM01860
      END                                                               LSM01870
C                                                                       LSM01880
C-----START OF USPEC (GENERAL SPARSE, RECTANGULAR MATRIX)---------------LSM01890
C                                                                       LSM01900
C     SUBROUTINE USPEC(M,N,MATNO)                                       LSM01910
      SUBROUTINE SUSPEC(M,N,MATNO)                                      LSM01920
C                                                                       LSM01930
C----------------------------------------------------------------       LSM01940
      DOUBLE PRECISION A(10000)                                         LSM01950
      INTEGER IROW(10000),ICOL(3010)                                    LSM01960
C----------------------------------------------------------------       LSM01970
C     DIMENSIONS ARRAYS NEEDED TO DEFINE THE USER-SUPPLIED              LSM01980
C     M X N RECTANGULAR A-MATRIX, READS IN VALUES OF THESE              LSM01990
C     ARRAYS AND THEN PASSES THE STORAGE LOCATIONS OF THESE             LSM02000
C     ARRAYS TO THE CORRESPONDING MATRIX-VECTOR MULTIPLY                LSM02010
C     SUBROUTINES SVMAT AND STRAN.                                      LSM02020
C                                                                       LSM02030
C     THE A-MATRIX IS STORED IN THE FOLLOWING SPARSE FORMAT:            LSM02040
C     M = NUMBER OF ROWS IN A.                                          LSM02050
C     N = NUMBER OF COLUMNS IN A.                                       LSM02060
C     NZ = NUMBER OF NONZERO ENTRIES IN A-MATRIX.                       LSM02070
C     ICOL(J), J=1,N IS NUMBER OF NONZERO ENTRIES IN COLUMN J.          LSM02080
```

```
C     IROW(K), K = 1,NZ IS THE ROW INDEX FOR CORRESPONDING A(K).      LSM02090
C     A(K), K=1,NZ IS NONZERO ENTRIES IN A, COLUMN BY COLUMN.         LSM02100
C     IT IS ASSUMED THAT ICOL(J) > 0 FOR ALL J                        LSM02110
C                                                                     LSM02120
C     NOTE:  ASSOCIATED SUBROUTINES SVMAT AND STRAN ASSUME THAT       LSM02130
C            M >= N.                                                  LSM02140
C                                                                     LSM02150
C---------------------------------------------------------------------LSM02160
C     READ IN  MATRIX FROM  FILE 8                                    LSM02170
C                                                                     LSM02180
      READ(8,10) NZ,MOLD,NOLD,MATOLD                                  LSM02190
   10 FORMAT(I10,2I6,I8)                                              LSM02200
C                                                                     LSM02210
      WRITE(6,20) NZ,MOLD,NOLD,MATOLD                                 LSM02220
   20 FORMAT(6X,'NZ',4X,'MOLD',4X,'NOLD',4X,'MATOLD'/I10,2I6,I10/)    LSM02230
C                                                                     LSM02240
C     TEST OF PARAMETER CORRECTNESS                                   LSM02250
      ITEMP = (MOLD-M)**2 + (NOLD-N)**2 + (MATOLD-MATNO)**2           LSM02260
C                                                                     LSM02270
      IF (ITEMP.EQ.0) GO TO 40                                        LSM02280
C                                                                     LSM02290
      WRITE(6,30)                                                     LSM02300
   30 FORMAT(' PROGRAM TERMINATES BECAUSE EITHER ORDERS OF OR LABELS FORLSM02310
     1 MATRIX DISAGREE')                                              LSM02320
      GO TO 70                                                        LSM02330
C                                                                     LSM02340
   40 CONTINUE                                                        LSM02350
C                                                                     LSM02360
C     NUMBER OF NONZERO ENTRIES IN EACH COLUMN IS READ IN             LSM02370
C     THEN THE CORRESPONDING ROW INDEX FOR EACH SUCH ENTRY IS READ    LSM02380
      READ(8,50) (ICOL(K), K=1,N)                                     LSM02390
      READ(8,50) (IROW(K), K=1,NZ)                                    LSM02400
   50 FORMAT(13I6)                                                    LSM02410
C                                                                     LSM02420
C     READ IN THE NONZERO ENTRIES IN THE MATRIX                       LSM02430
      READ(8,60) (A(K), K=1,NZ)                                       LSM02440
   60 FORMAT(3E25.16)                                                 LSM02450
C  50 FORMAT(4E19.10)                                                 LSM02460
C                                                                     LSM02470
C---------------------------------------------------------------------LSM02480
C     PASS STORAGE LOCATIONS OF ARRAYS THAT DEFINE THE MATRIX TO      LSM02490
C     THE MATRIX-VECTOR MULTIPLY SUBROUTINES SVMAT AND STRAN          LSM02500
      CALL SMATVE(A,ICOL,IROW,M,N)                                    LSM02510
      CALL STRANE(A,ICOL,IROW,M,N)                                    LSM02520
C---------------------------------------------------------------------LSM02530
C                                                                     LSM02540
C-----END OF USPEC----------------------------------------------------LSM02550
      RETURN                                                          LSM02560
   70 STOP                                                            LSM02570
      END                                                             LSM02580
C                                                                     LSM02590
C-----STRAN (GENERAL SPARSE MATRIX)-----------------------------------LSM02600
C                                                                     LSM02610
C     SUBROUTINE STRAN(W,U,SUM)                                       LSM02620
      SUBROUTINE SSTRAN(W,U,SUM)                                      LSM02630
```

```
C                                                               LSM02640
C-----------------------------------------------------------------LSM02650
      DOUBLE PRECISION W(1),U(1),A(1),SUM,TEMP                   LSM02660
      INTEGER IROW(1),ICOL(1)                                    LSM02670
C-----------------------------------------------------------------LSM02680
C     SUBROUTINE TO COMPUTE U = (A-TRANSPOSE)*W - SUM*U WHERE A IS LSM02690
C     A GENERAL, SPARSE M X N MATRIX WITH M >= N.                LSM02700
C                                                               LSM02710
C     ASSUMES MATRIX IS STORED IN SPARSE FORMAT GIVEN IN         LSM02720
C     CORRESPONDING USPEC SUBROUTINE.                            LSM02730
C-----------------------------------------------------------------LSM02740
      JLAST = 0                                                  LSM02750
      DO 20 J = 1,N                                              LSM02760
      JFIRST = JLAST + 1                                         LSM02770
      JLAST  = JLAST + ICOL(J)                                   LSM02780
      TEMP   = -SUM*U(J)                                         LSM02790
C                                                               LSM02800
      DO 10 K = JFIRST,JLAST                                     LSM02810
      IK = IROW(K)                                               LSM02820
   10 TEMP = A(K)*W(IK) + TEMP                                   LSM02830
C                                                               LSM02840
   20 U(J) = TEMP                                                LSM02850
C                                                               LSM02860
      RETURN                                                     LSM02870
C                                                               LSM02880
C-----------------------------------------------------------------LSM02890
      ENTRY STRANE(A,ICOL,IROW,M,N)                              LSM02900
C-----------------------------------------------------------------LSM02910
C                                                               LSM02920
C-----END OF STRAN FOR GENERAL SPARSE MATRIX----------------------LSM02930
      RETURN                                                     LSM02940
      END                                                        LSM02950
C                                                               LSM02960
C-----SVMAT (GENERAL SPARSE MATRIX)-------------------------------LSM02970
C                                                               LSM02980
C     SUBROUTINE SVMAT(W,U,SUM)                                  LSM02990
      SUBROUTINE SSVMAT(W,U,SUM)                                 LSM03000
C                                                               LSM03010
C-----------------------------------------------------------------LSM03020
      DOUBLE PRECISION  W(1),U(1),A(1),SUM,TEMP                  LSM03030
      INTEGER IROW(1),ICOL(1)                                    LSM03040
C-----------------------------------------------------------------LSM03050
C     SUBROUTINE TO COMPUTE U = A*W - SUM*U WHERE A IS A         LSM03060
C     GENERAL, SPARSE M X N MATRIX WITH M >= N.                  LSM03070
C                                                               LSM03080
C     ASSUMES THAT THE MATRIX IS STORED IN THE SPARSE FORMAT     LSM03090
C     GIVEN IN THE CORRESPONDING USPEC SUBROUTINE.              LSM03100
C-----------------------------------------------------------------LSM03110
      DO 10 I = 1,M                                              LSM03120
   10 U(I) = -SUM*U(I)                                           LSM03130
C                                                               LSM03140
C MAIN LOOP.  PROCESSING PROCEEDS COL BY COL.  JFIRST AND JLAST ARE LSM03150
C POINTERS TO THE FIRST AND LAST NONZEROS IN COLUMN J.          LSM03160
C                                                               LSM03170
      JLAST = 0                                                  LSM03180
```

```
      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          LSS00990
      BETAM = BETA(MP1)                                         LSS01000
      BETA(MP1) = ZERO                                          LSS01010
C                                                               LSS01020
      DO 10 I = 1,MP1                                           LSS01030
   10 BETA2(I) = BETA(I)*BETA(I)                                LSS01040
C                                                               LSS01050
C     EP0 IS USED IN T-MULTIPLICITY AND SPURIOUS TESTS          LSS01060
C     EP1 AND EPS ARE USED IN THE BISEC CONVERGENCE TEST        LSS01070
C                                                               LSS01080
      TEMP = DFLOAT(MEV+1000)                                   LSS01090
      EP0  = TEMP*TTOL                                          LSS01100
      EP1  = DSQRT(TEMP)*TTOL                                   LSS01110
C                                                               LSS01120
      WRITE(6,20)MEV,NINT                                       LSS01130
   20 FORMAT(/' BISEC CALCULATION'/' ORDER OF T IS',I6/         LSS01140
     1' NUMBER OF INTERVALS IS',I6/)                            LSS01150
C                                                               LSS01160
      WRITE(6,30) EP0,EP1                                       LSS01170
   30 FORMAT(/' MULTOL, TOLERANCE USED IN T-MULTIPLICITY AND SPURIOUS TELSS01180
     1STS = ',E10.3/' BISTOL, TOLERANCE USED IN BISEC CONVERGENCE TEST =LSS01190
     1',E10.3/)                                                 LSS01200
C                                                               LSS01210
C     LOOP ON THE NINT INTERVALS  (LB(J),UB(J)), J=1,NINT       LSS01220
      DO 430 JIND = 1,NINT                                      LSS01230
      LB = LBD(JIND)                                            LSS01240
      UB = UBD(JIND)                                            LSS01250
C                                                               LSS01260
      WRITE(6,40)JIND,LB,UB                                     LSS01270
   40 FORMAT(//1X,'BISEC INTERVAL NO',2X,'LOWER BOUND',2X,'UPPER BOUND'/LSS01280
     1I18,2E13.5/)                                              LSS01290
C                                                               LSS01300
C     INITIALIZATION AND PARAMETER SPECIFICATION                LSS01310
C     ICT IS TOTAL STURM COUNT ON (LB,UB)                       LSS01320
C                                                               LSS01330
      NA = 0                                                    LSS01340
      MD = 0                                                    LSS01350
      NG = 0                                                    LSS01360
      ICT = 0                                                   LSS01370
C                                                               LSS01380
C     START OF T-EIGENVALUE CALCULATIONS                        LSS01390
      X1 = UB                                                   LSS01400
      ISTURM = 1                                                LSS01410
      GO TO 330                                                 LSS01420
C     FORWARD STURM CALCULATION TO DETERMINE NA = NO. T-EIGENVALUES > UBLSS01430
   50 NA = NEV                                                  LSS01440
C                                                               LSS01450
      X1 = LB                                                   LSS01460
      ISTURM = 2                                                LSS01470
      GO TO 330                                                 LSS01480
C     FORWARD STURM CALC TO DETERMINE MT = NO. T-EIGENVALUES ON (LB,UB) LSS01490
   60 CONTINUE                                                  LSS01500
      MT=NEV                                                    LSS01510
      ICT = ICT +2                                              LSS01520
C                                                               LSS01530
```

```
      WRITE(6,70)MT,NA                                           LSS01540
   70 FORMAT(/2I6,' = NO. TMEV ON (LB,UB) AND NO. .GT. UB'/)     LSS01550
C                                                                LSS01560
C     DEFAULT TEST: IS ESTIMATED NUMBER OF STURMS > MXSTUR?      LSS01570
      IEST = 30*MT                                               LSS01580
      IF (IEST.LT.MXSTUR) GO TO 90                               LSS01590
C                                                                LSS01600
      WRITE(6,80)                                                LSS01610
   80 FORMAT(//' ESTIMATED NUMBER OF STURMS REQUIRED EXCEEDS USER LIMIT'LSS01620
     1/' SKIP THIS SUBINTERVAL')                                 LSS01630
      GO TO 110                                                  LSS01640
C                                                                LSS01650
   90 CONTINUE                                                   LSS01660
C                                                                LSS01670
      IF (MT.GE.1) GO TO 120                                     LSS01680
C                                                                LSS01690
      WRITE(6,100)                                               LSS01700
  100 FORMAT(//' THERE ARE NO T-EIGENVALUES ON THIS INTERVAL)'/) LSS01710
C                                                                LSS01720
  110 ISKIP = ISKIP+1                                            LSS01730
      IDEF(ISKIP) = JIND                                         LSS01740
      GO TO 430                                                  LSS01750
C                                                                LSS01760
C     REGULAR CASE.                                              LSS01770
  120 CONTINUE                                                   LSS01780
C                                                                LSS01790
      IF (IWRITE.NE.0) WRITE(6,130)                              LSS01800
  130 FORMAT(/' DISTINCT T-EIGENVALUES COMPUTED USING BISEC'/    LSS01810
     1 13X,'T-EIGENVALUE',2X,'TMULT',3X,'MD',4X,'NG')            LSS01820
C                                                                LSS01830
C     SET UP INITIAL UPPER AND LOWER BOUNDS FOR T-EIGENVALUES    LSS01840
      DO 140 I=1,MT                                              LSS01850
      VB(I) = LB                                                 LSS01860
      MTI = MT + I                                               LSS01870
  140 VB(MTI) = UB                                               LSS01880
C                                                                LSS01890
C     CALCULATE T-EIGENVALUES FROM LB UP TO UB  K = MT,...,1     LSS01900
C     MAIN LOOP FOR FINDING KTH T-EIGENVALUE                     LSS01910
C                                                                LSS01920
      K = MT                                                     LSS01930
  150 CONTINUE                                                   LSS01940
      ICO = 0                                                    LSS01950
      XL = VB(K)                                                 LSS01960
      MTK = MT+K                                                 LSS01970
      XU = VB(MTK)                                               LSS01980
C                                                                LSS01990
      ISTURM = 3                                                 LSS02000
      X1 = XU                                                    LSS02010
      ICO = ICO + 1                                              LSS02020
      GO TO 330                                                  LSS02030
C     FORWARD STURM CALCULATION AT XU                            LSS02040
  160 NU=NEV                                                     LSS02050
C                                                                LSS02060
C     BISECTION LOOP FOR KTH T-EIGENVALUE. TEST X1=MIDPOINT OF (XL,XU) LSS02070
      ISTURM = 4                                                 LSS02080
```

```
  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                            LSS02640
  240 KL = KEV                                              LSS02650
      JL = JEV                                              LSS02660
C                                                           LSS02670
      JSTURM = 2                                            LSS02680
      IC0 = IC0 + 2                                         LSS02690
      X1 = XU+EP0                                           LSS02700
      GO TO 370                                             LSS02710
C     BACKWARD STURM CALCULATION                            LSS02720
  250 JU = JEV                                              LSS02730
      KU = KEV                                              LSS02740
C                                                           LSS02750
C     FOR T(1,MEV)                                          LSS02760
C     NU - KU = NO. T-EIGENVALUES ON (XU, XU + EP0)         LSS02770
C     KL - KU = NO. T-EIGENVALUES ON (XL - EP0, XU + EP0)   LSS02780
C                                                           LSS02790
C     FOR T(2,MEV)                                          LSS02800
C     JL -JU = NO. T-EIGENVALUES ON (XL - EP0, XU + EP0)    LSS02810
C                                                           LSS02820
C     IS THIS A SIMPLE T-EIGENVALUE?                        LSS02830
C                                                           LSS02840
      IF (KL-KU-1.EQ.0) GO TO 290                           LSS02850
C                                                           LSS02860
C     VS(NDIS) = KTH-T-EIGENVALUE OF (LB,UB) IS T-MULTIPLE AND HENCE  LSS02870
C     GOOD                                                  LSS02880
      IF (KU.EQ.NU) GO TO 280                               LSS02890
C     CONTINUE TO CHECK FOR T-MULTIPLICITY                  LSS02900
  260 CONTINUE                                              LSS02910
      ISTURM = 5                                            LSS02920
      X1 = X1+EP0                                           LSS02930
      IC0 = IC0 + 1                                         LSS02940
      GO TO 330                                             LSS02950
C     FORWARD STURM CALCULATION                             LSS02960
  270 KNE = KU-NEV                                          LSS02970
      KU = NEV                                              LSS02980
      IF (KNE.NE.0) GO TO 260                               LSS02990
C     SPECIFY T-MULTIPLICITY = MP(NDIS)                     LSS03000
  280 MPEV = KL-KU                                          LSS03010
      KNEW = KU                                             LSS03020
      GO TO 300                                             LSS03030
C     END T-MULTIPLE CASE                                   LSS03040
C                                                           LSS03050
C     T-EIGENVALUE IS SIMPLE   CHECK IF IT IS SPURIOUS      LSS03060
  290 CONTINUE                                              LSS03070
      MPEV = 1                                              LSS03080
      IF (JU.LT.JL) MPEV=0                                  LSS03090
      KNEW = K-1                                            LSS03100
C                                                           LSS03110
C     X1 >= XU+EP0                                          LSS03120
C     SPURIOUS TEST AND SIMPLE CASE COMPLETED               LSS03130
C     START OF NEXT T-EIGENVALUE COMPUTATION                LSS03140
C                                                           LSS03150
  300 K = KNEW                                              LSS03160
      MP(NDIS) = MPEV                                       LSS03170
      IF (MPEV.GE.1) NG = NG + 1                            LSS03180
```

```
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. ICO = STURM COUNT FOR KTH T-EIGENVALUE     LSS03230
      ICT = ICT + ICO                                                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      LSS03740
C     JEV = -NA + (NUMBER OF T(2,MEV) T-EIGENVALUES) > X1      LSS03750
C                                                              LSS03760
C     SET PARAMETERS FOR NEXT INTERVAL                         LSS03770
  410 CONTINUE                                                 LSS03780
      IC = ICT+IC                                              LSS03790
      MXSTUR = MXSTUR-ICT                                      LSS03800
C                                                              LSS03810
      WRITE(6,420) JIND,NG,MD,ICT                              LSS03820
  420 FORMAT(/' T-EIGENVALUE CALCULATION ON INTERVAL',I6,'  IS COMPLETE'LSS03830
     1 /3X,'NO. GOOD',3X,'NO. DISTINCT',4X,'STURMS'/I10,I13,I10)   LSS03840
C                                                              LSS03850
  430 CONTINUE                                                 LSS03860
C                                                              LSS03870
C     END LOOP ON THE SUBINTERVALS (LB(J),UB(J)), J=1,NINT     LSS03880
C     ISKIP OUTPUT                                             LSS03890
C                                                              LSS03900
      IF (ISKIP.GT.0) WRITE(6,440)ISKIP                        LSS03910
  440 FORMAT(' BISEC DEFAULTED ON',I3,3X,'INTERVALS'/          LSS03920
     1 ' DEFAULTS OCCUR IF AN INTERVAL HAS NO T-EIGENVALUES'/  LSS03930
     2 ' OR THE STURM ESTIMATE EXCEEDS THE USER-SPECIFIED LIMIT'/)  LSS03940
C                                                              LSS03950
      IF (ISKIP.GT.0) WRITE(6,450)(IDEF(I), I=1,ISKIP)         LSS03960
  450 FORMAT(' BISEC DEFAULTED ON INTERVALS'/(10I8))           LSS03970
C                                                              LSS03980
C     RESET BETA AT I = MP1                                    LSS03990
      BETA(MP1) = BETAM                                        LSS04000
C-----END OF BISEC----------------------------------------------------LSS04010
      RETURN                                                   LSS04020
      END                                                      LSS04030
C                                                              LSS04040
C-----INVERSE ITERATION ON T(1,MEV)----------------------------------LSS04050
C                                                              LSS04060
      SUBROUTINE INVERR(BETA,V1,V2,VS,EPS,G,MP,MEV,MMB,NDIS,NISO,  LSS04070
     1 NM,IKL,IT,IWRITE)                                       LSS04080
C                                                              LSS04090
C-------------------------------------------------------------------LSS04100
      DOUBLE PRECISION BETA(1),V1(1),V2(1),VS(1)               LSS04110
      DOUBLE PRECISION X1,U,Z,EST,TEMP,T0,T1,RATIO,SUM,XU,NORM,TSUM  LSS04120
      DOUBLE PRECISION  BETAM,EPS,EPS3,EPS4,ZERO,ONE           LSS04130
      REAL G(1)                                                LSS04140
      INTEGER  MP(1)                                           LSS04150
C-------------------------------------------------------------------LSS04160
      DOUBLE PRECISION FINPRO                                  LSS04170
      REAL  ABS                                                LSS04180
      DOUBLE PRECISION  DABS, DMIN1, DSQRT, DFLOAT             LSS04190
C-------------------------------------------------------------------LSS04200
C     COMPUTES ERROR ESTIMATES FOR COMPUTED ISOLATED GOOD T-EIGENVALUES LSS04210
C     IN VS AND WRITES THESE T-EIGENVALUES AND ESTIMATES TO FILE 4.   LSS04220
C     BY DEFINITION A GOOD T-EIGENVALUE IS ISOLATED IF ITS     LSS04230
C     CLOSEST NEIGHBOR IS ALSO GOOD, OR IF ONE OF ITS NEIGHBORS IS  LSS04240
C     SPURIOUS BUT THAT NEIGHBOR IS FAR ENOUGH AWAY.  SO       LSS04250
C     IN PARTICULAR, WE COMPUTE ESTIMATES FOR GOOD T-EIGENVALUES  LSS04260
C     THAT ARE IN CLUSTERS OF GOOD T-EIGENVALUES.              LSS04270
C                                                              LSS04280
```

```
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                                      LSS05940
         I = MEV-II                                             LSS05950
  120 NORM = NORM+DABS(V2(I))                                   LSS05960
C                                                               LSS05970
         IF (NORM.GE.ONE) GO TO 140                             LSS05980
         IT = IT+1                                              LSS05990
         IF (IT.GT.ITER) GO TO 140                              LSS06000
         XU = EPS4/NORM                                         LSS06010
C                                                               LSS06020
         DO 130 I = 1,MEV                                       LSS06030
  130 V2(I) = V2(I)*XU                                          LSS06040
C                                                               LSS06050
         GO TO 70                                               LSS06060
C     ANOTHER INVERSE ITERATION STEP                            LSS06070
C                                                               LSS06080
C     INVERSE ITERATION FINISHED                                LSS06090
C     NORMALIZE COMPUTED T-EIGENVECTOR : V2 = V2/||V2||         LSS06100
  140 CONTINUE                                                  LSS06110
         SUM = FINPRO(MEV,V2(1),1,V2(1),1)                      LSS06120
         SUM = ONE/DSQRT(SUM)                                   LSS06130
C                                                               LSS06140
         DO 150 II = 1,MEV                                      LSS06150
  150 V2(II) = SUM*V2(II)                                       LSS06160
C                                                               LSS06170
C     SAVE ERROR ESTIMATE FOR LATER OUTPUT                      LSS06180
         EST = BETAM*DABS(V2(MEV))                              LSS06190
         IF (IT.GT.ITER) EST = -EST                             LSS06200
         MEVPNI = MEV + NISO                                    LSS06210
         G(MEVPNI) = EST                                        LSS06220
         IF (IWRITE.EQ.0) GO TO 180                             LSS06230
C                                                               LSS06240
C     FILE 6 (TERMINAL) OUTPUT OF ERROR ESTIMATES.             LSS06250
         IF (JEV.EQ.1) GAP = VS(2) - VS(1)                      LSS06260
         IF (JEV.EQ.MEV) GAP = VS(MEV) - VS(MEV-1)              LSS06270
         IF (JEV.EQ.MEV.OR.JEV.EQ.1) GO TO 160                  LSS06280
         TEMP = DMIN1(VS(JEV+1)-VS(JEV),VS(JEV)-VS(JEV-1))      LSS06290
         GAP = TEMP                                             LSS06300
  160 CONTINUE                                                  LSS06310
C                                                               LSS06320
         WRITE(6,170) NISO,JEV,X1,EST,GAP                       LSS06330
  170 FORMAT(2I6,E25.16,2E12.3)                                 LSS06340
C                                                               LSS06350
  180 CONTINUE                                                  LSS06360
C                                                               LSS06370
C     END ERROR ESTIMATE LOOP ON ISOLATED GOOD T-EIGENVALUES.   LSS06380
C     GENERATE DISTINCT MINGAPS FOR T(1,MEV).  THIS IS USEFUL AS AN  LSS06390
C     INDICATOR OF THE GOODNESS OF THE INVERSE ITERATION ESTIMATES.  LSS06400
C     TRANSFER ISOLATED GOOD T-EIGENVALUES AND CORRESPONDING TMINGAPS LSS06410
C     TO V2 AND V1 FOR OUTPUT PURPOSES ONLY.                    LSS06420
C                                                               LSS06430
         NM1 = NDIS - 1                                         LSS06440
         G(NDIS) = VS(NM1)-VS(NDIS)                             LSS06450
         G(1) = VS(2)-VS(1)                                     LSS06460
C                                                               LSS06470
         DO 190 J = 2,NM1                                       LSS06480
```

```
      T0 = VS(J)-VS(J-1)                                       LSS06490
      T1 = VS(J+1)-VS(J)                                       LSS06500
      G(J) = T1                                                LSS06510
      IF (T0.LT.T1) G(J)=-T0                                   LSS06520
  190 CONTINUE                                                 LSS06530
      ISO = 0                                                  LSS06540
      DO 200 J = 1,NDIS                                        LSS06550
      IF (MP(J).NE.1) GO TO 200                                LSS06560
      ISO = ISO+1                                              LSS06570
      V1(ISO) = G(J)                                           LSS06580
      V2(ISO) = VS(J)                                          LSS06590
  200 CONTINUE                                                 LSS06600
C                                                              LSS06610
      IF(NISO.EQ.0) GO TO 250                                  LSS06620
C                                                              LSS06630
C     ERROR ESTIMATES ARE WRITTEN TO FILE 4                    LSS06640
      WRITE(4,210)MEV,NDIS,NG,NISO,NM,IKL,ITER,BETAM           LSS06650
  210 FORMAT(1X,'TSIZE',2X,'NDIS',1X,'NGOOD',2X,'NISO',3X,'M+N'/5I6/  LSS06660
     1 4X,'RHSEED',2X,'MXINIT',5X,'BETAM'/I10,I8,E10.3/        LSS06670
     2 2X,'GOODEVNO',8X,'GOOD T-EIGENVALUE',6X,'BETAM*UM',7X,'TMINGAP') LSS06680
C                                                              LSS06690
      ISPUR = 0                                                LSS06700
      I = 0                                                    LSS06710
      DO 240 J = 1,NDIS                                        LSS06720
      IF(MP(J).NE.0) GO TO 220                                 LSS06730
      ISPUR = ISPUR + 1                                        LSS06740
      GO TO 240                                                LSS06750
  220 IF(MP(J).NE.1) GO TO 240                                 LSS06760
      I = I + 1                                                LSS06770
      MEVI = MEV + I                                           LSS06780
      IGOOD = J - ISPUR                                        LSS06790
      WRITE(4,230) IGOOD,V2(I),G(MEVI),V1(I)                   LSS06800
  230 FORMAT(I10,E25.16,2E14.3)                                LSS06810
  240 CONTINUE                                                 LSS06820
      GO TO 270                                                LSS06830
C                                                              LSS06840
  250 WRITE(4,260)                                             LSS06850
  260 FORMAT(/' THERE ARE NO ISOLATED T-EIGENVALUES SO NO ERROR ESTIMATELSS06860
     1S WERE COMPUTED')                                        LSS06870
C     RESTORE BETA(MEV+1) = BETAM                              LSS06880
  270 BETA(MP1) = BETAM                                        LSS06890
C-----END OF INVERR-------------------------------------------------LSS06900
      RETURN                                                   LSS06910
      END                                                      LSS06920
C                                                              LSS06930
C-----START OF TNORM-----------------------------------------------LSS06940
C                                                              LSS06950
      SUBROUTINE TNORM(BETA,BMIN,TMAX,MEV,IB)                  LSS06960
C                                                              LSS06970
C-----------------------------------------------------------------LSS06980
      DOUBLE PRECISION  BETA(1)                                LSS06990
      DOUBLE PRECISION  TMAX,BMIN,BSIZE,BTOL                   LSS07000
      DOUBLE PRECISION  DABS, DMAX1                            LSS07010
C-----------------------------------------------------------------LSS07020
C     COMPUTE SCALING FACTOR USED IN THE T-MULTIPLICITY, SPURIOUS AND  LSS07030
```

```
C      PRTESTS.   CHECK RELATIVE SIZE OF THE BETA(K), K=1,MEV        LSS07040
C      AS A TEST ON THE LOCAL ORTHOGONALITY OF THE LANCZOS VECTORS.  LSS07050
C                                                                    LSS07060
C             TMAX = MAX (BETA(I),   I=1,MEV)                        LSS07070
C             BMIN = MIN (BETA(I) I=2,MEV)                           LSS07080
C             BSIZE = BMIN/TMAX                                      LSS07090
C             |IB| = INDEX OF MINIMAL(BETA)                          LSS07100
C             IB < 0 IF BMIN/TMAX < BTOL                             LSS07110
C--------------------------------------------------------------------LSS07120
C     SPECIFY PARAMETERS                                             LSS07130
      IB = 2                                                         LSS07140
      BTOL = BMIN                                                    LSS07150
      BMIN = BETA(2)                                                 LSS07160
      TMAX = BETA(2)                                                 LSS07170
C                                                                    LSS07180
      DO 20 I = 2,MEV                                                LSS07190
      IF (BETA(I).GE.BMIN) GO TO 10                                  LSS07200
      IB = I                                                         LSS07210
      BMIN = BETA(I)                                                 LSS07220
   10 TMAX = DMAX1(TMAX,BETA(I))                                     LSS07230
   20 CONTINUE                                                       LSS07240
C                                                                    LSS07250
C     TEST OF LOCAL ORTHOGONALITY USING SCALED BETAS                 LSS07260
      BSIZE = BMIN/TMAX                                              LSS07270
      IF (BSIZE.GE.BTOL) GO TO 40                                    LSS07280
C                                                                    LSS07290
C     DEFAULT.  BSIZE IS SMALLER THAN TOLERANCE BTOL SPECIFIED IN MAIN LSS07300
C     PROGRAM.  PROGRAM TERMINATES FOR USER TO DECIDE WHAT TO DO     LSS07310
C     BECAUSE LOCAL ORTHOGONALITY OF THE LANCZOS VECTORS COULD BE    LSS07320
C     LOST.                                                          LSS07330
C                                                                    LSS07340
      IB = -IB                                                       LSS07350
      WRITE(6,30) MEV                                                LSS07360
   30 FORMAT(/' BETA TEST INDICATES POSSIBLE LOSS OF LOCAL ORTHOGONALITYLSS07370
     10VER 1ST',I6,' LANCZOS VECTORS'/)                             LSS07380
C                                                                    LSS07390
   40 CONTINUE                                                       LSS07400
C                                                                    LSS07410
      WRITE(6,50) IB                                                 LSS07420
   50 FORMAT(/' MINIMUM BETA RATIO OCCURS AT',I6,' TH BETA'/)        LSS07430
C                                                                    LSS07440
      WRITE(6,60) MEV,BMIN,TMAX,BSIZE                                LSS07450
   60 FORMAT(/1X,'TSIZE',6X,'MIN BETA',5X,'TKMAX',6X,'MIN RATIO'/    LSS07460
     1 I6,E14.3,E10.3,E15.3/)                                        LSS07470
C                                                                    LSS07480
C-----END OF TNORM---------------------------------------------------LSS07490
      RETURN                                                         LSS07500
      END                                                            LSS07510
C                                                                    LSS07520
C                                                                    LSS07530
C-----START OF LUMP-------------------------------------------------LSS07540
C                                                                    LSS07550
      SUBROUTINE LUMP(V1,RELTOL,MULTOL,SCALE2,LINDEX,LOOP)           LSS07560
C                                                                    LSS07570
C--------------------------------------------------------------------LSS07580
```

```
      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                                                                 LSS08690
C     ON EXIT                                                     LSS08700
C     G CONTAINS THE TMINGAPS.                                   LSS08710
C     G(I) < 0 MEANS MINGAP IS DUE TO LEFT GAP                   LSS08720
C     MP(I) IS NOT CHANGED EXCEPT THAT  MP(I)=-1, IF MP(I)=1,    LSS08730
C     TMINGAP WAS TOO SMALL AND DUE TO A SPURIOUS T-EIGENVALUE.  LSS08740
C                                                                 LSS08750
C     IF MP(I)=-1 THAT SIMPLE GOOD T-EIGENVALUE WILL BE SKIPPED  LSS08760
C     IN THE SUBSEQUENT ERROR ESTIMATE COMPUTATIONS IN INVERR    LSS08770
C     THAT IS, WE COMPUTE ERROR ESTIMATES ONLY FOR THOSE GOOD    LSS08780
C     T-EIGENVALUES WITH MP(I)=1.                                LSS08790
C-----------------------------------------------------------------LSS08800
C     CALCULATE MINGAPS FOR DISTINCT T(1,MEV) EIGENVALUES.       LSS08810
      NM1 = NDIS - 1                                             LSS08820
      G(NDIS) = VS(NM1)-VS(NDIS)                                 LSS08830
      G(1) = VS(2)-VS(1)                                         LSS08840
C                                                                 LSS08850
      DO 10 J = 2,NM1                                            LSS08860
      T0 = VS(J)-VS(J-1)                                         LSS08870
      T1 = VS(J+1)-VS(J)                                         LSS08880
      G(J) = T1                                                  LSS08890
      IF (T0.LT.T1) G(J) = -T0                                   LSS08900
   10 CONTINUE                                                   LSS08910
C                                                                 LSS08920
C     SET MP(I)=-1 FOR SIMPLE GOOD T-EIGENVALUES WHOSE MINGAPS  ARE LSS08930
C     'TOO SMALL' AND DUE TO SPURIOUS T-EIGENVALUES.            LSS08940
C                                                                 LSS08950
      NISO = 0                                                   LSS08960
      NG = 0                                                     LSS08970
      DO 20 J = 1,NDIS                                           LSS08980
      IF (MP(J).EQ.0) GO TO 20                                   LSS08990
      NG = NG+1                                                  LSS09000
      IF (MP(J).NE.1) GO TO 20                                   LSS09010
C     VS(J) IS NEXT TO SIMPLE GOOD T-EIGENVALUE                  LSS09020
      NISO = NISO + 1                                            LSS09030
      I = J+1                                                    LSS09040
      IF (G(J).LT.0.0) I = J-1                                   LSS09050
      IF (MP(I).NE.0) GO TO 20                                   LSS09060
      GAP = ABS(G(J))                                            LSS09070
      T0 = DMAX1(SCALE1*MULTOL,GAPTOL*DABS(VS(J)))               LSS09080
C     T0 = DMAX1(GAPTOL,GAPTOL*DABS(VS(J)))                      LSS09090
      TEMP = T0                                                  LSS09100
      IF (GAP.GT.TEMP) GO TO 20                                  LSS09110
      MP(J) = -MP(J)                                             LSS09120
      NISO = NISO-1                                              LSS09130
   20 CONTINUE                                                   LSS09140
C                                                                 LSS09150
C-----END OF ISOEV----------------------------------------------- LSS09160
      RETURN                                                     LSS09170
      END                                                        LSS09180
C                                                                 LSS09190
C-----START OF PRTEST-------------------------------------------- LSS09200
C                                                                 LSS09210
      SUBROUTINE PRTEST(BETA,TEIG,TKMAX,EPSM,RELTOL,SCALE3,SCALE4, LSS09220
     1 TMULT,NDIST,MEV,IPROJ)                                    LSS09230
```

```
C                                                               LSS09240
C-----------------------------------------------------------------LSS09250
      DOUBLE PRECISION  BETA(1),TEIG(1),SIGMA(4)                LSS09260
      DOUBLE PRECISION  EPSM,RELTOL,PRTOL,TKMAX,LRATIO,URATIO   LSS09270
      DOUBLE PRECISION  EPS,EPS1,BETAM,LBD,UBD,SIG,YU,YV,LRATS,URATS LSS09280
      DOUBLE PRECISION  ZERO,ONE,TEN,BISTOL,SCALE3,SCALE4,AEV,TEMP LSS09290
      INTEGER  TMULT(1),ISIGMA(4)                               LSS09300
      DOUBLE PRECISION  DABS, DMAX1, DSQRT, DFLOAT              LSS09310
C-----------------------------------------------------------------LSS09320
C     AFTER CONVERGENCE HAS BEEN ESTABLISHED, SUBROUTINE PRTEST LSS09330
C     TESTS COMPUTED EIGENVALUES OF T(1,MEV) THAT HAVE BEEN LABELLED LSS09340
C     SPURIOUS TO DETERMINE IF ANY SINGULAR VALUES OF A HAVE BEEN LSS09350
C     MISSED BY LANCZOS PROCEDURE.  A SINGULAR VALUE WHOSE     LSS09360
C     SINGULAR VECTOR(S) HAS A VERY SMALL PROJECTION ON THE    LSS09370
C     STARTING VECTOR (< SINGLE PRECISION) CAN BE MISSED BECAUSE LSS09380
C     IT WILL THEN ALSO BE AN EIGENVALUE OF T(2,MEV) TO WITHIN LSS09390
C     THE SQUARE OF THIS ORIGINAL PROJECTION.  HOWEVER,        LSS09400
C     OUR EXPERIENCE IS THAT SUCH SMALL PROJECTIONS OCCUR ONLY LSS09410
C     VERY INFREQUENTLY.                                       LSS09420
C                                                               LSS09430
C     THIS SUBROUTINE IS CALLED ONLY AFTER CONVERGENCE HAS BEEN LSS09440
C     ESTABLISHED. ONCE CONVERGENCE HAS BEEN OBSERVED ON THE   LSS09450
C     OTHER SINGULAR VALUES, THEN ONE CAN EXPECT TO ALSO HAVE  LSS09460
C     CONVERGENCE ON ANY SUCH 'HIDDEN' SINGULAR VALUES. (IF THERE LSS09470
C     ARE ANY).  PROCEDURE CONSIDERS ONLY SPURIOUS T-EIGENVALUES AND LSS09480
C     ONLY THOSE SPURIOUS T-EIGENVALUES THAT ARE ISOLATED FROM GOOD LSS09490
C     T-EIGENVALUES.  FOR EACH SUCH T-EIGENVALUE IT DOES 2 STURM LSS09500
C     SEQUENCES AND A FEW SCALAR MULTIPLICATIONS.  UPON RETURN TO MAIN LSS09510
C     PROGRAM ERROR ESTIMATES WILL BE COMPUTED FOR ANY T-EIGENVALUES LSS09520
C     THAT HAVE BEEN LABELLED AS 'HIDDEN'.  SUCH T-EIGENVALUES  LSS09530
C     WILL BE RELABELLED AS 'GOOD' ONLY IF THESE ERROR ESTIMATES LSS09540
C     ARE SUFFICIENTLY SMALL.                                  LSS09550
C-----------------------------------------------------------------LSS09560
      ZERO = 0.0D0                                              LSS09570
      ONE  = 1.0D0                                              LSS09580
      TEN  = 10.0D0                                             LSS09590
      PRTOL = 1.D-6                                             LSS09600
      TEMP = DFLOAT(MEV+1000)                                   LSS09610
      TEMP = DSQRT(TEMP)                                        LSS09620
      BISTOL = TKMAX*EPSM*TEMP                                  LSS09630
      NSIGMA = 4                                                LSS09640
      SIGMA(1) = TEN*TKMAX                                      LSS09650
C                                                               LSS09660
      DO 10 J = 2,NSIGMA                                        LSS09670
   10 SIGMA(J) = TEN*SIGMA(J-1)                                 LSS09680
C                                                               LSS09690
      IFIN = 0                                                  LSS09700
      MF = 1                                                    LSS09710
      ML = MEV                                                  LSS09720
      BETAM = BETA(MF)                                          LSS09730
      BETA(MF) = ZERO                                           LSS09740
      IPROJ = 0                                                 LSS09750
      J = 1                                                     LSS09760
C                                                               LSS09770
      IF (TMULT(1).NE.0) GO TO 110                              LSS09780
```

```
C                                                                 LSS09790
      AEV = DABS(TEIG(1))                                         LSS09800
      TEMP = PRTOL*AEV                                            LSS09810
      EPS1 = DMAX1(TEMP,SCALE4*BISTOL)                            LSS09820
C     EPS1 = DMAX1(TEMP,PRTOL)                                    LSS09830
      TEMP = RELTOL*AEV                                           LSS09840
      EPS  = DMAX1(TEMP,SCALE3*BISTOL)                            LSS09850
C     EPS  = DMAX1(TEMP,RELTOL)                                   LSS09860
C                                                                 LSS09870
      IF (TEIG(2)-TEIG(1).LT.EPS1.AND.TMULT(2).NE.0) GO TO 110    LSS09880
C                                                                 LSS09890
   20 LBD = TEIG(J) - EPS                                         LSS09900
      UBD = TEIG(J) + EPS                                         LSS09910
      MEVL = 0                                                    LSS09920
      IL = 0                                                      LSS09930
      YU = ONE                                                    LSS09940
C                                                                 LSS09950
      DO 50 I=MF,ML                                               LSS09960
      IF (YU.NE.ZERO) GO TO 30                                    LSS09970
      YV = BETA(I)/EPSM                                           LSS09980
      GO TO 40                                                    LSS09990
   30 YV = BETA(I)*BETA(I)/YU                                     LSS10000
   40 YU = -LBD-YV                                                LSS10010
      IF (YU.GE.ZERO) GO TO 50                                    LSS10020
C     MEVL INCREMENTED                                            LSS10030
      MEVL = MEVL + 1                                             LSS10040
      IL = I                                                      LSS10050
   50 CONTINUE                                                    LSS10060
C                                                                 LSS10070
      LRATIO = YU                                                 LSS10080
      MEV1L = MEVL                                                LSS10090
      IF (IL.EQ.ML) MEV1L=MEVL-1                                  LSS10100
C                                                                 LSS10110
C     MEVL = NUMBER OF EVS OF T(1,MEV) WHICH ARE < LBD            LSS10120
C     MEV1L = NUMBER OF EVS OF T(1,MEV-1) WHICH ARE < LBD         LSS10130
C     LRATIO = DET(T(1,MEV)-LBD)/DET(T(1,MEV-1)-LBD):            LSS10140
C                                                                 LSS10150
      MEVU = 0                                                    LSS10160
      IL = 0                                                      LSS10170
      YU = ONE                                                    LSS10180
C                                                                 LSS10190
      DO 80 I=MF,ML                                               LSS10200
      IF (YU.NE.ZERO) GO TO 60                                    LSS10210
      YV = BETA(I)/EPSM                                           LSS10220
      GO TO 70                                                    LSS10230
   60 YV = BETA(I)*BETA(I)/YU                                     LSS10240
   70 YU = -UBD-YV                                                LSS10250
      IF (YU.GE.ZERO) GO TO 80                                    LSS10260
C     MEVU INCREMENTED                                            LSS10270
      MEVU = MEVU + 1                                             LSS10280
      IL = I                                                      LSS10290
   80 CONTINUE                                                    LSS10300
C                                                                 LSS10310
      URATIO = YU                                                 LSS10320
      MEV1U = MEVU                                                LSS10330
```

```
      IF (IL.EQ.ML) MEV1U=MEVU-1                               LSS10340
C                                                              LSS10350
C     MEVU = NUMBER OF EVS OF T(MEV) WHICH ARE < UBD           LSS10360
C     MEV1U = NUMBER OF EVS OF T(MEV-1) WHICH ARE < UBD        LSS10370
C     URATIO = DET(TM-UBD)/DET(T(M-1)-UBD): TM=T(MF,ML)        LSS10380
C                                                              LSS10390
      NEV1 = MEV1U-MEV1L                                       LSS10400
C                                                              LSS10410
      DO 90 K=1,NSIGMA                                         LSS10420
      SIG = SIGMA(K)                                           LSS10430
      LRATS = LRATIO-SIG                                       LSS10440
      URATS = URATIO-SIG                                       LSS10450
C     NOTE THE INCREMENT IS ON NUMBER OF EVALUES OF T(M-1)     LSS10460
      MEVLS = MEV1L                                            LSS10470
      IF (LRATS.LT.0.) MEVLS=MEV1L+1                           LSS10480
      MEVUS = MEV1U                                            LSS10490
      IF (URATS.LT.0.) MEVUS=MEV1U+1                           LSS10500
      ISIGMA(K) = MEVUS - MEVLS                                LSS10510
   90 CONTINUE                                                 LSS10520
C                                                              LSS10530
      ICOUNT = 0                                               LSS10540
      DO 100 K=1,NSIGMA                                        LSS10550
  100 IF (ISIGMA(K).EQ.1) ICOUNT=ICOUNT + 1                    LSS10560
C                                                              LSS10570
      IF (ICOUNT.LT.2.OR.NEV1.EQ.0) GO TO 110                  LSS10580
      TMULT(J) = -10                                           LSS10590
      IPROJ=IPROJ+1                                            LSS10600
C                                                              LSS10610
  110 J=J+1                                                    LSS10620
C                                                              LSS10630
      IF (J.GE.NDIST) GO TO 120                                LSS10640
      IF (TMULT(J).NE.0) GO TO 110                             LSS10650
C                                                              LSS10660
      AEV = DABS(TEIG(J))                                      LSS10670
      TEMP = PRTOL*AEV                                         LSS10680
      EPS1 = DMAX1(TEMP,SCALE4*BISTOL)                         LSS10690
C     EPS1 = DMAX1(TEMP,PRTOL)                                 LSS10700
      TEMP = RELTOL*AEV                                        LSS10710
      EPS  = DMAX1(TEMP,SCALE3*BISTOL)                         LSS10720
C     EPS  = DMAX1(TEMP,RELTOL)                                LSS10730
C                                                              LSS10740
      IF (TEIG(J)-TEIG(J-1).LT.EPS1.AND.TMULT(J-1).NE.0) GO TO 110  LSS10750
      IF (TEIG(J+1)-TEIG(J).LT.EPS1.AND.TMULT(J+1).NE.0) GO TO 110  LSS10760
C                                                              LSS10770
      GO TO 20                                                 LSS10780
C                                                              LSS10790
  120 IF (IFIN.EQ.1) GO TO 130                                 LSS10800
      IF (TMULT(NDIST).NE.0) GO TO 130                         LSS10810
C                                                              LSS10820
      AEV = DABS(TEIG(NDIST))                                  LSS10830
      TEMP = PRTOL*AEV                                         LSS10840
      EPS1 = DMAX1(TEMP,SCALE4*BISTOL)                         LSS10850
C     EPS1 = DMAX1(TEMP,PRTOL)                                 LSS10860
      TEMP = RELTOL*AEV                                        LSS10870
      EPS  = DMAX1(TEMP,SCALE3*BISTOL)                         LSS10880
```

```
C     EPS  = DMAX1(TEMP,RELTOL)                                  LSS10890
C                                                                LSS10900
      NDIST1=NDIST -1                                            LSS10910
      TEMP = TEIG(NDIST)-TEIG(NDIST1)                            LSS10920
      IF (TEMP.LT.EPS1.AND.TMULT(NDIST1).NE.0) GO TO 130         LSS10930
      IFIN = 1                                                   LSS10940
C                                                                LSS10950
      GO TO 20                                                   LSS10960
C                                                                LSS10970
  130 BETA(MF) = BETAM                                           LSS10980
C                                                                LSS10990
C-----END OF PRTEST----------------------------------------------LSS11000
      RETURN                                                     LSS11010
      END                                                        LSS11020
C                                                                LSS11030
C------START OF STURMI-------------------------------------------LSS11040
C                                                                LSS11050
      SUBROUTINE STURMI(BETA,X1,TOLN,EPSM,MMAX,MK1,MK2,IC,IWRITE) LSS11060
C                                                                LSS11070
C---------------------------------------------------------------LSS11080
      DOUBLE PRECISION  BETA(1)                                  LSS11090
      DOUBLE PRECISION  EPSM,X1,TOLN,EVL,EVU,BETA2               LSS11100
      DOUBLE PRECISION  U1,U2,V1,V2,ZERO,ONE                     LSS11110
      INTEGER I,IC,ICD,IC0,IC1,IC2,MK1,MK2,MMAX                  LSS11120
C---------------------------------------------------------------LSS11130
C                                                                LSS11140
C     FOR ANY GOOD T-EIGENVALUE THAT HAS CONVERGED AS AN EIGENVALUE LSS11150
C     OF THE T-MATRICES THIS SUBROUTINE CALCULATES              LSS11160
C     THE SMALLEST SIZE OF THE T-MATRIX, T(1,MK1) DEFINED       LSS11170
C     BY THE BETA ARRAY SUCH THAT MK1.LE.MMAX                   LSS11180
C     AND THE INTERVAL (X1-TOLN,X1+TOLN) CONTAINS AT LEAST ONE  LSS11190
C     EIGENVALUE OF T(1,MK1). IT ALSO CALCULATES MK2 <= MMAX    LSS11200
C     AS THE SMALLEST SIZE T-MATRIX (IF ANY) SUCH THAT THIS INTERVAL LSS11210
C     CONTAINS AT LEAST TWO EIGENVALUES OF T(1,MK2).            LSS11220
C     IF NO T-MATRIX OF ORDER < MMAX SATISFIES THIS REQUIREMENT LSS11230
C     THEN MK2 IS SET EQUAL TO MMAX.  THE SINGULAR VECTOR PROGRAM LSS11240
C     USES THESE VALUES TO DETERMINE A 1ST GUESS AT AN APPROPRIATE LSS11250
C     SIZE T-MATRIX FOR THE SINGULAR VALUE  X1.                 LSS11260
C                                                                LSS11270
C     ON EXIT IC = NUMBER OF EIGENVALUES OF T(1,MK2) IN THIS INTERVAL LSS11280
C                                                                LSS11290
C     STURMI REGENERATES THE QUANTITIES BETA(I)**2 EACH TIME IT IS LSS11300
C     CALLED, OBVIOUSLY FOR THE PRICE OF ANOTHER VECTOR OF LENGTH LSS11310
C     MMAX THIS GENERATION COULD BE DONE ONCE IN THE MAIN       LSS11320
C     PROGRAM BEFORE THE LOOP ON THE CALLS TO SUBROUTINE STURMI. LSS11330
C                                                                LSS11340
C     IF ANY OF THE GOOD T-EIGENVALUES BEING CONSIDERED WERE MULTIPLE LSS11350
C     AS SINGULAR VALUES OF THE USER-SPECIFIED MATRIX, THEN     LSS11360
C     THIS SUBROUTINE COULD BE MODIFIED TO COMPUTE ADDITIONAL   LSS11370
C     SIZES MKJ, J = 3, ...  WHICH COULD THEN BE USED IN THE    LSS11380
C     MAIN LANCZOS SINGULAR VECTOR PROGRAM TO COMPUTE ADDITIONAL LSS11390
C     SINGULAR VECTORS CORRESPONDING TO THESE MULTIPLE SINGULAR LSS11400
C     VALUES. THE MAIN PROGRAM LSVEC PROVIDED DOES NOT INCLUDE  LSS11410
C     THIS OPTION.                                              LSS11420
C                                                                LSS11430
```

```
C------------------------------------------------------------------------LSS11440
C       INITIALIZATION OF PARAMETERS                                      LSS11450
        MK1 = 0                                                           LSS11460
        MK2 = 0                                                           LSS11470
        ZERO = 0.0D0                                                      LSS11480
        ONE  = 1.0D0                                                      LSS11490
        BETA(1) = ZERO                                                    LSS11500
        EVL = X1-TOLN                                                     LSS11510
        EVU = X1+TOLN                                                     LSS11520
        U1 = ONE                                                          LSS11530
        U2 = ONE                                                          LSS11540
        IC0 = 0                                                           LSS11550
        IC1 = 0                                                           LSS11560
        IC2 = 0                                                           LSS11570
C                                                                         LSS11580
C       MAIN LOOP FOR CALCULATING THE SIZES MK1,MK2                       LSS11590
        DO 60 I = 1,MMAX                                                  LSS11600
        BETA2 = BETA(I)*BETA(I)                                           LSS11610
        IF (U1.NE.ZERO) GO TO 10                                          LSS11620
        V1 = BETA(I)/EPSM                                                 LSS11630
        GO TO 20                                                          LSS11640
     10 V1 = BETA2/U1                                                     LSS11650
     20 U1 = EVL - V1                                                     LSS11660
        IF (U1.LT.ZERO) IC1 = IC1+1                                       LSS11670
        IF (U2.NE.ZERO) GO TO 30                                          LSS11680
        V2 = BETA(I)/EPSM                                                 LSS11690
        GO TO 40                                                          LSS11700
     30 V2 = BETA2/U2                                                     LSS11710
     40 U2 = EVU - V2                                                     LSS11720
        IF (U2.LT.ZERO) IC2 = IC2+1                                       LSS11730
C       TEST FOR CHANGE IN NUMBER OF T-EIGENVALUES ON (EVL,EVU)           LSS11740
        ICD = IC1-IC2                                                     LSS11750
        IC = ICD-IC0                                                      LSS11760
        IF (IC.GE.1) GO TO 50                                             LSS11770
        GO TO 60                                                          LSS11780
     50 CONTINUE                                                          LSS11790
        IF (IC0.EQ.0) MK1 = I                                             LSS11800
        IC0 = IC0+1                                                       LSS11810
        IF (IC0.GT.1) GO TO 70                                            LSS11820
     60 CONTINUE                                                          LSS11830
C                                                                         LSS11840
        I = I-1                                                           LSS11850
        IF (IC0.EQ.0) MK1 = MMAX                                          LSS11860
     70 MK2 = I                                                           LSS11870
        IC = ICD                                                          LSS11880
C                                                                         LSS11890
        IF (IWRITE.EQ.1) WRITE(6,80) X1,MK1,MK2,IC                        LSS11900
     80 FORMAT(' EVAL =',E20.12,' MK1 =',I6,' MK2 =',I6,' IC =',I3/)      LSS11910
C                                                                         LSS11920
        RETURN                                                           LSS11930
C-----END OF STURMI------------------------------------------------------LSS11940
        END                                                              LSS11950
C                                                                         LSS11960
C                                                                         LSS11970
C-----START OF INVERM----------------------------------------------------LSS11980
```

```
C                                                              LSS11990
      SUBROUTINE INVERM(BETA,V1,V2,X1,ERROR,ERRORV,EPS,G,MEV,IT, LSS12000
     1 IWRITE)                                                  LSS12010
C                                                              LSS12020
C-------------------------------------------------------------LSS12030
      DOUBLE PRECISION  BETA(1),V1(1),V2(1)                    LSS12040
      DOUBLE PRECISION  X1,U,Z,TEMP,RATIO,SUM,XU,NORM,TSUM,BETAM LSS12050
      DOUBLE PRECISION  EPS,EPS3,EPS4,ERROR,ERRORV,ZERO,ONE    LSS12060
      REAL  G(1)                                               LSS12070
      DOUBLE PRECISION  DABS, DSQRT, DFLOAT                     LSS12080
      DOUBLE PRECISION FINPRO                                   LSS12090
      REAL  ABS                                                 LSS12100
C-------------------------------------------------------------LSS12110
C                                                              LSS12120
C     COMPUTES T-EIGENVECTORS FOR ISOLATED GOOD T-EIGENVALUES X1 LSS12130
C     USING INVERSE ITERATION ON T(1,MEV(X1)) SOLVING EQUATION LSS12140
C     (T - X1*I)V2 = RIGHT-HAND SIDE (RANDOMLY-GENERATED) .     LSS12150
C     PROGRAM REFACTORS T- X1*I ON EACH ITERATION OF INVERSE ITERATION. LSS12160
C     TYPICALLY ONLY ONE ITERATION IS NEEDED PER T-EIGENVALUE X1. LSS12170
C                                                              LSS12180
C     IF IWRITE = 1 THEN THERE ARE EXTENDED WRITES TO FILE 6 (TERMINAL) LSS12190
C                                                              LSS12200
C     ON ENTRY G CONTAINS A REAL*4 RANDOM VECTOR WHICH WAS GENERATED LSS12210
C     IN MAIN PROGRAM.                                         LSS12220
C                                                              LSS12230
C     ON ENTRY AND EXIT                                        LSS12240
C     MEV = ORDER OF T                                         LSS12250
C     BETA CONTAINS THE OFFDIAGONAL ENTRIES OF T.              LSS12260
C     EPS = 2. * MACHINE EPSILON                               LSS12270
C                                                              LSS12280
C     IN PROGRAM:                                              LSS12290
C     ITER = MAXIMUM NUMBER STEPS ALLOWED FOR INVERSE ITERATION LSS12300
C     ITER = IT ON ENTRY.                                      LSS12310
C     V1,V2 = WORK SPACES USED IN THE FACTORIZATION OF T(1,MEV). LSS12320
C     V1 AND V2 MUST BE OF DIMENSION AT LEAST MEV.             LSS12330
C                                                              LSS12340
C     ON EXIT                                                  LSS12350
C     V2 = THE UNIT EIGENVECTOR OF T(1,MEV) CORRESPONDING TO X1. LSS12360
C     ERROR = |V2(MEV)| = ERROR ESTIMATE FOR CORRESPONDING     LSS12370
C             RITZ VECTOR FOR X1.                              LSS12380
C                                                              LSS12390
C     ERRORV = || T*V2 - X1*V2 || = ERROR ESTIMATE ON T-EIGENVECTOR. LSS12400
C     IF IT.GT.ITER THEN ERRORV = -ERRORV                      LSS12410
C     IT = NUMBER OF ITERATIONS ACTUALLY REQUIRED              LSS12420
C-------------------------------------------------------------LSS12430
C     INITIALIZATION AND PARAMETER SPECIFICATION               LSS12440
      ONE  = 1.0D0                                             LSS12450
      ZERO = 0.0D0                                             LSS12460
      ITER = IT                                                LSS12470
      MP1 = MEV+1                                              LSS12480
      MM1 = MEV-1                                              LSS12490
      BETAM = BETA(MP1)                                        LSS12500
      BETA(MP1) = ZERO                                         LSS12510
C                                                              LSS12520
C     CALCULATE SCALE AND TOLERANCES                           LSS12530
```

```
        TSUM = ZERO                                             LSS12540
        DO 10 I = 2,MEV                                         LSS12550
     10 TSUM = TSUM + BETA(I)                                   LSS12560
C                                                               LSS12570
        EPS3 = EPS*TSUM                                         LSS12580
        EPS4 = DFLOAT(MEV)*EPS3                                 LSS12590
C                                                               LSS12600
C       GENERATE SCALED RANDOM RIGHT-HAND SIDE                  LSS12610
        GSUM = ZERO                                             LSS12620
        DO 20 I = 1,MEV                                         LSS12630
     20 GSUM = GSUM+ABS(G(I))                                   LSS12640
        GSUM = EPS4/GSUM                                        LSS12650
C                                                               LSS12660
C       INITIALIZE RIGHT HAND SIDE FOR INVERSE ITERATION        LSS12670
        DO 30 I = 1,MEV                                         LSS12680
     30 V2(I) = GSUM*G(I)                                       LSS12690
        IT = 1                                                  LSS12700
C                                                               LSS12710
C       CALCULATE UNIT EIGENVECTOR OF T(1,MEV) FOR ISOLATED GOOD  LSS12720
C       T-EIGENVALUE X1.                                        LSS12730
C                                                               LSS12740
C       TRIANGULAR FACTORIZATION WITH NEAREST NEIGHBOR PIVOT    LSS12750
C       STRATEGY. INTERCHANGES ARE LABELLED BY SETTING BETA < 0. LSS12760
C                                                               LSS12770
     40 CONTINUE                                                LSS12780
        U = -X1                                                 LSS12790
        Z = BETA(2)                                             LSS12800
C                                                               LSS12810
        DO 60 I=2,MEV                                           LSS12820
        IF (BETA(I).GT.DABS(U)) GO TO 50                        LSS12830
C       NO PIVOT INTERCHANGE                                    LSS12840
        V1(I-1) = Z/U                                           LSS12850
        V2(I-1) = V2(I-1)/U                                     LSS12860
        V2(I) = V2(I)-BETA(I)*V2(I-1)                           LSS12870
        RATIO = BETA(I)/U                                       LSS12880
        U = -X1-Z*RATIO                                         LSS12890
        Z = BETA(I+1)                                           LSS12900
        GO TO 60                                                LSS12910
C       PIVOT INTERCHANGE                                       LSS12920
     50 CONTINUE                                                LSS12930
        RATIO = U/BETA(I)                                       LSS12940
        BETA(I) = -BETA(I)                                      LSS12950
        V1(I-1) = -X1                                           LSS12960
        U = Z-RATIO*V1(I-1)                                     LSS12970
        Z = -RATIO*BETA(I+1)                                    LSS12980
        TEMP = V2(I-1)                                          LSS12990
        V2(I-1) = V2(I)                                         LSS13000
        V2(I) = TEMP-RATIO*V2(I)                                LSS13010
     60 CONTINUE                                                LSS13020
C                                                               LSS13030
        IF (U.EQ.ZERO) U=EPS3                                   LSS13040
C                                                               LSS13050
C       SMALLNESS TEST AND DEFAULT VALUE FOR LAST COMPONENT     LSS13060
C       PIVOT(I-1) = |BETA(I)| FOR INTERCHANGE CASE             LSS13070
C       (I-1,I+1) ELEMENT IN RIGHT FACTOR = BETA(I+1)           LSS13080
```

```
C     END OF FACTORIZATION AND FORWARD SUBSTITUTION                  LSS13090
C                                                                    LSS13100
C     BACK SUBSTITUTION                                              LSS13110
      V2(MEV) = V2(MEV)/U                                            LSS13120
      DO 80 II = 1,MM1                                               LSS13130
      I = MEV-II                                                     LSS13140
      IF (BETA(I+1).LT.ZERO) GO TO 70                                LSS13150
C     NO PIVOT INTERCHANGE                                           LSS13160
      V2(I) = V2(I)-V1(I)*V2(I+1)                                    LSS13170
      GO TO 80                                                       LSS13180
C     PIVOT INTERCHANGE                                              LSS13190
   70 BETA(I+1) = -BETA(I+1)                                         LSS13200
      V2(I) = (V2(I)-V1(I)*V2(I+1)-BETA(I+2)*V2(I+2))/BETA(I+1)      LSS13210
   80 CONTINUE                                                       LSS13220
C                                                                    LSS13230
C                                                                    LSS13240
C     TESTS FOR CONVERGENCE OF INVERSE ITERATION                     LSS13250
C     IF SUM |V2| COMPS. LE. 1 AND IT. LE. ITER DO ANOTHER INVIT STEP LSS13260
C                                                                    LSS13270
      NORM = DABS(V2(MEV))                                           LSS13280
      DO 90 II = 1,MM1                                               LSS13290
      I = MEV-II                                                     LSS13300
   90 NORM = NORM+DABS(V2(I))                                        LSS13310
C                                                                    LSS13320
C     IS DESIRED GROWTH IN VECTOR ACHIEVED ?                         LSS13330
C     IF NOT, DO ANOTHER INVERSE ITERATION STEP UNLESS NUMBER ALLOWED ISLSS13340
C     EXCEEDED.                                                      LSS13350
      IF (NORM.GE.ONE) GO TO 110                                     LSS13360
C                                                                    LSS13370
      IT=IT+1                                                        LSS13380
      IF (IT.GT.ITER) GO TO 110                                      LSS13390
C                                                                    LSS13400
      XU = EPS4/NORM                                                 LSS13410
      DO 100 I=1,MEV                                                 LSS13420
  100 V2(I) = V2(I)*XU                                               LSS13430
C                                                                    LSS13440
      GO TO 40                                                       LSS13450
C                                                                    LSS13460
C     NORMALIZE COMPUTED T-EIGENVECTOR : V2 = V2/||V2||              LSS13470
C                                                                    LSS13480
  110 CONTINUE                                                       LSS13490
C                                                                    LSS13500
      SUM = FINPRO(MEV,V2(1),1,V2(1),1)                             LSS13510
      SUM = ONE/DSQRT(SUM)                                           LSS13520
      DO 120 II = 1,MEV                                              LSS13530
  120 V2(II) = SUM*V2(II)                                            LSS13540
C                                                                    LSS13550
C     SAVE ERROR ESTIMATE FOR LATER OUTPUT                           LSS13560
      ERROR = DABS(V2(MEV))                                          LSS13570
C                                                                    LSS13580
C     GENERATE ERRORV = ||T*V2 - X1*V2||.                           LSS13590
      V1(MEV) = BETA(MEV)*V2(MEV-1)-X1*V2(MEV)                       LSS13600
      DO 130 J = 2,MM1                                               LSS13610
      JM = MP1 - J                                                   LSS13620
      V1(JM) = BETA(JM)*V2(JM-1) + BETA(JM+1)*V2(JM+1                LSS13630
```

```
      1) - X1*V2(JM)                                          LSS13640
  130 CONTINUE                                                LSS13650
C                                                             LSS13660
      V1(1) = BETA(2)*V2(2) - X1*V2(1)                        LSS13670
      ERRORV = FINPRO(MEV,V1(1),1,V1(1),1)                    LSS13680
      ERRORV = DSQRT(ERRORV)                                  LSS13690
      IF (IT.GT.ITER) ERRORV = -ERRORV                        LSS13700
      IF (IWRITE.EQ.0) GO TO 150                              LSS13710
C                                                             LSS13720
C     FILE 6 (TERMINAL) OUTPUT OF ERROR ESTIMATES.            LSS13730
      WRITE(6,140) MEV,X1,ERROR,ERRORV                        LSS13740
  140 FORMAT(' INVERSE ITERATION OUTPUT'/                     LSS13750
     1 2X,'TSIZE',13X,'T-EIGENVALUE',11X,'U(M)',9X,'ERRORV'/  LSS13760
     1 I6,E25.16,2E15.5)                                      LSS13770
C                                                             LSS13780
C     RESTORE BETA(MEV+1) = BETAM                             LSS13790
  150 CONTINUE                                                LSS13800
      BETA(MP1) = BETAM                                       LSS13810
C-----END OF INVERM-------------------------------------------LSS13820
      RETURN                                                  LSS13830
      END                                                     LSS13840
C                                                             LSS13850
C-----START OF LBISEC-----------------------------------------LSS13860
C                                                             LSS13870
      SUBROUTINE LBISEC(BETA,EPSM,EVAL,EVALN,LB,UB,TTOL,M,NEVT) LSS13880
C                                                             LSS13890
C------------------------------------------------------------LSS13900
      DOUBLE PRECISION  BETA(1),X0,X1,XL,XU,YU,YV,LB,UB       LSS13910
      DOUBLE PRECISION  EPSM,EP1,EVAL,EVALN,EVD,EPT           LSS13920
      DOUBLE PRECISION  ZERO,ONE,HALF,TTOL,TEMP               LSS13930
      DOUBLE PRECISION  DABS,DSQRT,DFLOAT                     LSS13940
C------------------------------------------------------------LSS13950
C     SPECIFY PARAMETERS                                      LSS13960
      ZERO = 0.0D0                                            LSS13970
      HALF = 0.5D0                                            LSS13980
      ONE  = 1.0D0                                            LSS13990
      XL = LB                                                 LSS14000
      XU = UB                                                 LSS14010
C                                                             LSS14020
C     EP1 = DSQRT(1000+M)*TTOL     TTOL = EPSM*TKMAX          LSS14030
C     TKMAX = MAX(BETA(K), K= 1,KMAX)                         LSS14040
C                                                             LSS14050
      TEMP = DFLOAT(1000+M)                                   LSS14060
      EP1 = DSQRT(TEMP)*TTOL                                  LSS14070
C                                                             LSS14080
      NA = 0                                                  LSS14090
      X1 = XU                                                 LSS14100
      JSTURM = 1                                              LSS14110
      GO TO 60                                                LSS14120
C     FORWARD STURM CALCULATION                               LSS14130
   10 NA = NEV                                                LSS14140
      X1 = XL                                                 LSS14150
      JSTURM = 2                                              LSS14160
      GO TO 60                                                LSS14170
C     FORWARD STURM CALCULATION                               LSS14180
```

```
   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
-----------------------------------------------------------------------
```

Chapter 7

Nondefective Complex Symmetric Matrices

7.1   Introduction

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

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

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

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

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

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

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

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

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

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

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

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

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

As stated earlier, all computations are in double precision complex arithmetic. The user must supply a subroutine USPEC which defines and initializes the user-specified matrix A and a subroutine CMATV which computes matrix-vector multiplies Ax for any given vector x. These subroutines must be constructed

in such a way as to take advantage of the sparsity (and/or structure) of the user-supplied A-matrix and such that these computations are done accurately.

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

7.2   Documentation for the Codes in Chapter 7

```
C-----CSLEVALD-------------------------------------------------   CSL00010
C                                                                 CSL00020
C     DOCUMENTATION FOR SINGLE-VECTOR                             CSL00030
C     LANCZOS EIGENVALUE/EIGENVECTOR PROGRAMS FOR                 CSL00040
C     NONDEFECTIVE COMPLEX SYMMETRIC MATRICES                     CSL00050
C                                                                 CSL00060
C----------------------------------------------------------------CSL00070
C          REFERENCE:  Cullum and Willoughby, Chapter 6,         CSL00080
C          Lanczos Algorithms for Large Symmetric Eigenvalue ComputationsCSL00090
C          VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in   CSL00100
C          Applied Mathematics, 2002. SIAM Publications,          CSL00110
C          Philadelphia, PA. USA                                 CSL00120
C                                                                 CSL00130
C                                                                 CSL00140
C----------------------------------------------------------------CSL00150
C  Authors:  Jane Cullum and Ralph A. Willoughby (Deceased)      CSL00160
C            Los Alamos National Laboratory                      CSL00170
C            Los Alamos, New Mexico 87544                         CSL00180
C                                                                 CSL00190
C            E-mail:  cullumj@lanl.gov                            CSL00200
C                                                                 CSL00210
C These codes are copyrighted by the authors.  These codes       CSL00220
C and modifications of them or portions of them are NOT to be    CSL00230
C incorporated into any commercial codes or used for any other   CSL00240
C commercial purposes such as consulting for other companies,    CSL00250
C without legal agreements with the authors of these Codes.      CSL00260
C If these Codes or portions of them                             CSL00270
C are used in other scientific or engineering research works     CSL00280
C the names of the authors of these codes and appropriate        CSL00290
C references to their written work are to be incorporated in the CSL00300
C derivative works.                                              CSL00310
C                                                                 CSL00320
C This header is not to be removed from these codes.             CSL00330
C                                                                 CSL00340
C     GIVEN A NONDEFECTIVE COMPLEX SYMMETRIC MATRIX A OF ORDER N  CSL00350
C     THE THREE SETS OF FORTRAN FILES LABELLED CSLEVAL, CSLESUB,  CSL00360
C     AND CSLEMULT CAN BE USED TO COMPUTE DISTINCT EIGENVALUES OF CSL00370
C     A.  NOTE THAT THESE PROGRAMS DIFFER FROM THE REAL SYMMETRIC CSL00380
C     AND HERMITIAN PROGRAMS IN THAT IT IS NOT POSSIBLE TO        CSL00390
C     COMPUTE THE EIGENVALUES OF THE LANCZOS TRIDIAGONAL MATRICES CSL00400
C     ONLY IN SPECIFIED INTERVALS.  THUS, ON ANY GIVEN           CSL00410
C     ITERATION ALL OF THE EIGENVALUES OF THESE TRIDIAGONAL MATRICES CSL00420
C     MUST BE COMPUTED.  IN FACT TWO COMPLETE TRIDIAGONAL EIGENVALUE CSL00430
C     COMPUTATIONS ARE USED.                                     CSL00440
C                                                                 CSL00450
C     CORRESPONDING EIGENVECTORS FOR SELECTED, COMPUTED EIGENVALUES CAN CSL00460
C     BE COMPUTED USING THE CORRESPONDING SETS OF FILES LABELLED  CSL00470
C     CSLEVEC, CSLESUB AND CSLEMULT.                             CSL00480
C                                                                 CSL00490
C     THESE PROGRAMS ALL USE A GENERALIZATION OF LANCZOS          CSL00500
C     TRIDIAGONALIZATION TO COMPLEX SYMMETRIC MATRICES TO         CSL00510
```

```
C      GENERATE COMPLEX SYMMETRIC TRIDIAGONAL MATRICES, T(1,MEV)        CSL00520
C      OF ORDER MEV.  NO REORTHOGONALIZATION IS USED.  SUBSETS OF       CSL00530
C      THE EIGENVALUES OF THESE T-MATRICES, LABELLED AS THE             CSL00540
C      'GOOD EIGENVALUES', YIELD APPROXIMATIONS TO THE DESIRED          CSL00550
C      EIGENVALUES OF A.  CORRESPONDING RITZ VECTORS ARE APPROXIMATIONS CSL00560
C      TO THE DESIRED EIGENVECTORS OF A.  NOTE THAT IN THE DISCUSSION   CSL00570
C      T(1,MEV) DENOTES THE LANCZOS MATRIX OF ORDER MEV AND T(2,MEV)    CSL00580
C      DENOTES THE MATRIX OF SIZE MEV-1 OBTAINED FROM T(1,MEV) BY       CSL00590
C      DELETING THE FIRST ROW AND COLUMN OF T(1,MEV).                   CSL00600
C                                                                       CSL00610
C      THE IDEAS USED IN THESE PROGRAMS ARE DISCUSSED IN THE FOLLOWING  CSL00620
C      REFERENCES.                                                      CSL00630
C                                                                       CSL00640
C      1.   JANE CULLUM AND RALPH A. WILLOUGHBY, LANCZOS ALGORITHMS     CSL00650
C           FOR LARGE SYMMETRIC MATRICES,  PROGRESS IN                  CSL00660
C           SCIENTIFIC COMPUTING, EDITORS, G. GOLUB, H.O. KREISS,       CSL00670
C           S. ARBARBANEL, AND R. GLOWINSKI,  BIRKHAUSER BOSTON INC.,   CSL00680
C           CAMBRIDGE, MASSACHUSETTS, 1985.                             CSL00690
C                                                                       CSL00700
C      2.   JANE CULLUM AND RALPH A. WILLOUGHBY, COMPUTING EIGENVECTORS CSL00710
C           (AND EIGENVALUES) OF LARGE, SYMMETRIC MATRICES USING        CSL00720
C           LANCZOS TRIDIAGONALIZATION, LECTURE NOTES IN MATHEMATICS,   CSL00730
C           773, NUMERICAL ANALYSIS PROCEEDINGS, DUNDEE 1979, EDITED BY CSL00740
C           G. A. WATSON, SPRINGER-VERLAG, (1980), BERLIN, PP.46-63.    CSL00750
C                                                                       CSL00760
C      3. IBID, LANCZOS AND THE COMPUTATION IN SPECIFIED INTERVALS OF   CSL00770
C           THE SPECTRUM OF LARGE SPARSE, REAL SYMMETRIC MATRICES, SPARSE CSL00780
C           MATRIX PROCEEDINGS 1978, ED. I.S. DUFF AND G. W. STEWART,   CSL00790
C           SIAM, PHILADELPHIA, PP.220-255, 1979.                       CSL00800
C                                                                       CSL00810
C      4. IBID, COMPUTING EIGENVALUES OF VERY LARGE SYMMETRIC MATRICES- CSL00820
C           AN IMPLEMENTATION OF A LANCZOS ALGORITHM WITHOUT            CSL00830
C           REORTHOGONALIZATION, J. COMPUT. PHYS. 44(1981), 329-358.    CSL00840
C                                                                       CSL00850
C      5. IBID, A LANCZOS ALGORITHM FOR NONDEFECTIVE COMPLEX SYMMETRIC  CSL00860
C           MATRICES, IBM RESEARCH REPORT, 1984.                        CSL00870
C                                                                       CSL00880
C-----PORTABILITY-------------------------------------------------------CSL00890
C                                                                       CSL00900
C      PROGRAMS WERE TESTED FOR PORTABILITY USING THE PFORT VERIFIER.   CSL00910
C      FOR DETAILS OF THE VERIFIER SEE FOR EXAMPLE, B. G. RYDER AND     CSL00920
C      A. D. HALL, 'THE PFORT VERIFIER', COMPUTING SCIENCE TECHNICAL    CSL00930
C      REPORT 12, BELL LABORATORIES, MURRAY HILL, NEW JERSEY 07974,     CSL00940
C      (REVISED), JANUARY 1981.                                         CSL00950
C                                                                       CSL00960
C      PORTABILITY:                                                     CSL00970
C      THESE PROGRAMS ARE NOT PORTABLE DUE TO THE USE OF COMPLEX*16     CSL00980
C      VARIABLES AND CORRESPONDING COMPLEX FUNCTIONS.  IN ADDITION, THE CSL00990
C      PFORT VERIFIER IDENTIFIED THE FOLLOWING NONPORTABLE             CSL01000
C      CONSTRUCTIONS.                                                   CSL01010
C      IN CSLEVAL AND IN CSLEVEC                                        CSL01020
C           1.   DATA/MACHEP STATEMENT                                  CSL01030
C           2.   ALL READ(5,*) STATEMENTS (FREE FORMAT)                 CSL01040
C           3.   FORMAT(20A4) USED FOR THE EXPLANATORY HEADER ARRAY, EXPLANCSL01050
C           4.   HEXADECIMAL FORMAT (4Z20) FOR ALPHA/BETA FILES 1 AND 2. CSL01060
```

```
C     IN CSLEMULT                                              CSL01070
C        1.  IN CMATV AND  USPEC THE ENTRY THAT PASSES THE STORAGE  CSL01080
C            LOCATIONS OF THE ARRAYS DEFINING THE USER-SPECIFIED  CSL01090
C            MATRIX.                                            CSL01100
C        2.  IN SAMPLE USPEC PROVIDED : FREE FORMAT (8,*), THE  CSL01110
C            FORMAT (20A4), AND THE DATA/MACHEP STATEMENT.      CSL01120
C                                                               CSL01130
C     IN THE COMMENTS BELOW :                                   CSL01140
C     REAL*16 = COMPLEX VARIABLE, 16 BYTES OF STORAGE           CSL01150
C     REAL*8 = REAL VARIABLE, 8 BYTES OF STORAGE                CSL01160
C     REAL*4 = REAL VARIABLE, 4 BYTES OF STORAGE                CSL01170
C     INTEGER*4 = INTEGER VARIABLE, 4 BYTES OF STORAGE          CSL01180
C                                                               CSL01190
C-----A-MATRIX SPECIFICATION-------------------------------------CSL01200
C                                                               CSL01210
C     SUBROUTINE USPEC IS USED TO SPECIFY THE USER-SUPPLIED MATRIX.  CSL01220
C     SUBROUTINE CMATV IS A CORRESPONDING MATRIX-VECTOR MULTIPLY  CSL01230
C     SUBROUTINE WHICH SHOULD BE DESIGNED TO TAKE ADVANTAGE OF   CSL01240
C     ANY SPECIAL PROPERTIES OF THE USER-SUPPLIED MATRIX.  THE   CSL01250
C     MATRIX-VECTOR MULTIPLIES REQUIRED BY THE LANCZOS PROCEDURES  CSL01260
C     MUST BE COMPUTED RAPIDLY AND ACCURATELY.                  CSL01270
C                                                               CSL01280
C     SUBROUTINE USPEC HAS THE CALLING SEQUENCE                 CSL01290
C                                                               CSL01300
C          CALL USPEC(N,MATNO)                                  CSL01310
C                                                               CSL01320
C     WHERE N IS THE ORDER OF THE USER-SUPPLIED MATRIX A AND    CSL01330
C     MATNO IS A <= 8 DIGIT INTEGER USED AS A MATRIX AND        CSL01340
C     TEST IDENTIFICATION NUMBER.  THIS SUBROUTINE DEFINES (DIMENSIONS) CSL01350
C     THE ARRAYS REQUIRED TO SPECIFY THE USER-SUPPLIED MATRIX AND  CSL01360
C     INITIALIZES THESE ARRAYS AND ANY OTHER PARAMETERS NEEDED TO  CSL01370
C     DEFINE THE MATRIX.  THE STORAGE LOCATIONS OF THESE PARAMETERS  CSL01380
C     AND ARRAYS ARE THEN PASSED TO THE MATRIX-VECTOR MULTIPLY   CSL01390
C     SUBROUTINE CMATV VIA AN ENTRY.  A SAMPLE USPEC SUBROUTINE  CSL01400
C     IS INCLUDED.  THIS SAMPLE SUBROUTINE ASSUMES THAT THE MATRIX  CSL01410
C     IS STORED ON FILE 8 IN A TYPICAL SPARSE MATRIX FORMAT.    CSL01420
C     SEE THE HEADER ON THE SUBROUTINE USPEC FOR DETAILS ON THIS  CSL01430
C     PARTICULAR STORAGE FORMAT.                                CSL01440
C                                                               CSL01450
C     SUBROUTINE CMATV HAS THE CALLING SEQUENCE                 CSL01460
C                                                               CSL01470
C          CALL CMATV(W,U,SUM)                                  CSL01480
C                                                               CSL01490
C     IN THE COMPLEX SYMMETRIC CASE, U AND W ARE                CSL01500
C     COMPLEX*16 VECTORS AND SUM IS A COMPLEX*16                CSL01510
C     SCALAR.  CMATV CALCULATES U = A*W - SUM*U FOR THE         CSL01520
C     USER-SPECIFIED MATRIX A.  THE ARRAY AND PARAMETER INFORMATION  CSL01530
C     NEEDED TO PERFORM THE MATRIX-VECTOR MULTIPLIES IS PASSED TO  CSL01540
C     THE CMATV SUBROUTINE FROM THE USPEC SUBROUTINE VIA THE CMATVE  CSL01550
C     ENTRY IN CMATV.  A SAMPLE CMATV SUBROUTINE IS INCLUDED WHICH  CSL01560
C     COMPUTES MATRIX-VECTOR MULTIPLIES FOR AN ARBITRARY SPARSE,  CSL01570
C     COMPLEX SYMMETRIC MATRIX STORED IN THE SPARSE FORMAT      CSL01580
C     SPECIFIED IN THE SAMPLE USPEC SUBROUTINE.                 CSL01590
C                                                               CSL01600
C     CMATV IS CALLED FROM THE SUBROUTINE LANCZS WHICH GENERATES  CSL01610
```

```
C       THE T-MATRICES IN THE ALPHA, BETA ARRAYS.  IT IS ALSO CALLED    CSL01620
C       FROM THE MAIN PROGRAM CSLEVEC FOR THE EIGENVECTOR COMPUTATIONS.  CSL01630
C       CMATV IS DECLARED AS AN EXTERNAL VARIABLE AND IS AN ARGUMENT     CSL01640
C       FOR THE SUBROUTINE LANCZS.                                       CSL01650
C                                                                        CSL01660
C       THE USPEC AND CMATV SUBROUTINES MUST BE MODIFIED BY THE USER     CSL01670
C       TO ACCOMODATE THE USER'S SPECIFIED MATRIX.                       CSL01680
C                                                                        CSL01690
C       THE MAIN PROGRAMS FOR THE EIGENVALUE AND EIGENVECTOR             CSL01700
C       CALCULATIONS ASSUME THAT INPUT FILE 5 CONTAINS N = ORDER OF      CSL01710
C       THE MATRIX AND MATNO = AN IDENTIFICATION NUMBER OF <= 8 DIGITS   CSL01720
C       FOR THE MATRIX AND THE RUN.                                      CSL01730
C                                                                        CSL01740
C                                                                        CSL01750
C-----MACHEP-------------------------------------------------------------CSL01760
C                                                                        CSL01770
C                                                                        CSL01780
C       MACHEP IS A MACHINE DEPENDENT PARAMETER SPECIFYING THE RELATIVE  CSL01790
C       PRECISION OF THE FLOATING POINT ARITHMETIC USED.                 CSL01800
C       MACHEP = 2.2 * 10**-16 FOR DOUBLE PRECISION ARITHMETIC ON        CSL01810
C       IBM 370-3081.                                                    CSL01820
C                                                                        CSL01830
C       THE USER WILL HAVE TO RESET THIS PARAMETER TO                    CSL01840
C       THE CORRESPONDING VALUE FOR THE MACHINE BEING USED.  NOTE THAT   CSL01850
C       IF A MACHINE WITH A MACHINE EPSILON THAT IS MUCH LARGER THAN THE CSL01860
C       VALUE GIVEN HERE IS BEING USED, THEN THERE COULD BE              CSL01870
C       PROBLEMS WITH THE TOLERANCES.                                    CSL01880
C                                                                        CSL01890
C                                                                        CSL01900
C-----SUBROUTINES AND FUNCTIONS USER MUST SUPPLY------------------------CSL01910
C                                                                        CSL01920
C    GENRAN,  MASK,  USPEC,  AND  CMATV                                  CSL01930
C                                                                        CSL01940
C    GENRAN = COMPUTES K PSEUDO-RANDOM NUMBERS AND STORES THEM IN        CSL01950
C             THE REAL ARRAY, G.  THIS SUBROUTINE IS USED TO            CSL01960
C             GENERATE A STARTING VECTOR FOR THE LANCZOS PROCEDURE      CSL01970
C             IN THE SUBROUTINE LANCZS AND A STARTING RIGHT-HAND SIDE   CSL01980
C             FOR INVERSE ITERATION IN THE SUBROUTINE INVERR.           CSL01990
C                                                                        CSL02000
C             TESTS REPORTED IN THE REFERENCES USED EITHER GGL1 OR       CSL02010
C             GGL2 FROM THE IBM LIBRARY  SLMATH.                         CSL02020
C             THE EXISTING CALLING SEQUENCE IS:                          CSL02030
C                                                                        CSL02040
C                   CALL GENRAN(IIX,G,K).                                CSL02050
C                                                                        CSL02060
C             WHERE IIX =INTEGER SEED, G = REAL*4 ARRAY WHOSE            CSL02070
C             DIMENSION MUST BE >= K.  K RANDOM NUMBERS ARE GENERATED    CSL02080
C             AND PLACED IN G.                                           CSL02090
C                                                                        CSL02100
C    MASK = MASKS OVERFLOW AND UNDERFLOW.                                CSL02110
C           USER MUST SUPPLY OR COMMENT OUT CALL.                        CSL02120
C                                                                        CSL02130
C    USPEC = DIMENSIONS AND INITIALIZES ARRAYS NEEDED TO SPECIFY         CSL02140
C            USER-SUPPLIED MATRIX.  SEE A-MATRIX SPECIFICATION SECTION.CSL02150
C                                                                        CSL02160
```

```
C     CMATV = MATRIX-VECTOR MULTIPLY FOR USER-SUPPLIED MATRIX.        CSL02170
C           SEE A-MATRIX SPECIFICATION SECTION.                       CSL02180
C                                                                     CSL02190
C                                                                     CSL02200
C---------------------------------------------------------------------CSL02210
C                                                                     CSL02220
C     COMMENTS FOR EIGENVALUE COMPUTATIONS                            CSL02230
C                                                                     CSL02240
C---------------------------------------------------------------------CSL02250
C                                                                     CSL02260
C                                                                     CSL02270
C-----PARAMETER CONTROLS FOR EIGENVALUE PROGRAMS----------------------CSL02280
C                                                                     CSL02290
C     PARAMETER CONTROLS ARE INTRODUCED TO ALLOW SEGMENTATION OF THE  CSL02300
C     EIGENVALUE COMPUTATIONS AND TO ALLOW VARIOUS COMBINATIONS OF    CSL02310
C     READ/WRITES.                                                    CSL02320
C                                                                     CSL02330
C     THE FLAG ISTART CONTROLS THE T-MATRIX (ALPHA/BETA HISTORY)      CSL02340
C     GENERATION.                                                     CSL02350
C                                                                     CSL02360
C     ISTART = (0,1)  MEANS                                           CSL02370
C                                                                     CSL02380
C             (0) THERE IS NO EXISTING ALPHA/BETA HISTORY AND ONE     CSL02390
C                 MUST BE GENERATED.                                  CSL02400
C                                                                     CSL02410
C             (1) THERE IS AN EXISTING ALPHA/BETA HISTORY AND IT IS   CSL02420
C                 TO BE READ IN FROM FILE 2 AND EXTENDED IF NECESSARY. CSL02430
C                                                                     CSL02440
C     THE FLAG ISTOP CAN BE USED IN CONJUNCTION WITH THE FLAG ISTART TO CSL02450
C     ALLOW SEGMENTATION OF THE EIGENVALUE COMPUTATIONS.             CSL02460
C                                                                     CSL02470
C     ISTOP  = (0,1)  MEANS                                           CSL02480
C                                                                     CSL02490
C             (0) PROGRAM COMPUTES ONLY THE REQUESTED ALPHAS/BETAS,   CSL02500
C                 STORES THEM AND THE LAST 2 LANCZOS VECTORS GENERATED CSL02510
C                 IN FILE 1 AND THEN TERMINATES.                      CSL02520
C                                                                     CSL02530
C             (1) PROGRAM COMPUTES REQUESTED ALPHAS/BETAS AND THEN    CSL02540
C                 USES THE CMTQL1 SUBROUTINE TO CALCULATE EIGENVALUES CSL02550
C                 OF THE TRIDIAGONAL MATRICES GENERATED FOR THE ORDERS CSL02560
C                 SPECIFIED BY THE USER.  PROGRAM THEN USES THE       CSL02570
C                 SUBROUTINE INVERR TO COMPUTE ERROR ESTIMATES FOR    CSL02580
C                 THE ISOLATED GOOD T-EIGENVALUES WHICH ARE USED TO   CSL02590
C                 CHECK THE CONVERGENCE OF THESE GOOD T-EIGENVALUES.  CSL02600
C                                                                     CSL02610
C     CONTROL PARAMETERS FOR WRITES                                   CSL02620
C                                                                     CSL02630
C     IHIS  = (0,1)  MEANS                                            CSL02640
C                                                                     CSL02650
C             (0) IF ISTOP .GT. 0 THEN ALPHAS/BETAS ARE NOT SAVED     CSL02660
C                 ON FILE 1.                                          CSL02670
C                                                                     CSL02680
C             (1) PROGRAM WRITES ALPHAS/BETAS AND LAST 2 LANCZOS      CSL02690
C                 VECTORS TO FILE 1 SO THAT THE T-MATRIX GENERATION   CSL02700
C                 MAY BE REUSED OR CONTINUED LATER IF NECESSARY.      CSL02710
```

```
C                    TYPICALLY ONE WOULD ALWAYS DO THIS ON ANY RUN WHERE    CSL02720
C                    A HISTORY FILE IS BEING GENERATED.  HISTORY MUST       CSL02730
C                    BE SAVED IN MACHINE FORMAT ((4Z20) FOR IBM/3081)       CSL02740
C                    SO THAT NO ERRORS ARE INTRODUCED DUE TO FORMAT         CSL02750
C                    CONVERSIONS.                                           CSL02760
C                                                                           CSL02770
C     IDIST  = (0,1)  MEANS                                                 CSL02780
C                                                                           CSL02790
C                 (0) DISTINCT EIGENVALUES OF T-MATRICES ARE NOT SAVED.     CSL02800
C                                                                           CSL02810
C                 (1) PROGRAM WRITES COMPUTED DISTINCT EIGENVALUES OF       CSL02820
C                     T-MATRICES ALONG WITH THEIR T-MULTIPLICITIES          CSL02830
C                     TO FILE 11.                                           CSL02840
C                                                                           CSL02850
C     IWRITE = (0,1)   MEANS                                                CSL02860
C                                                                           CSL02870
C                 (0) NO EXTENDED OUTPUT FROM SUBROUTINES COMPEV AND INVERRCSL02880
C                     IS SENT TO FILE 6.                                    CSL02890
C                                                                           CSL02900
C                 (1) INDIVIDUAL COMPUTED EIGENVALUES AND CORRESPONDING     CSL02910
C                     ERROR ESTIMATES FROM THE SUBROUTINES COMPEV AND       CSL02920
C                     INVERR ARE PRINTED OUT TO FILE 6 AS THEY ARE COMPUTEDCSL02930
C                                                                           CSL02940
C     SAVTEV = (-1,0,1)   MEANS                                             CSL02950
C                                                                           CSL02960
C                 (-1) NO T-EIGENVALUE COMPUTATIONS.  PREVIOUSLY-COMPUTED   CSL02970
C                      EIGENVALUES OF T(1,MEV) AND T(2,MEV) ARE TO          CSL02980
C                      BE READ IN FROM FILE 10.                             CSL02990
C                                                                           CSL03000
C                 (0) COMPUTED EIGENVALUES OF T(1,MEV) AND OF T(2,MEV)      CSL03010
C                     ARE NOT TO BE SAVED ON FILE 10.  THIS IS NOT          CSL03020
C                     RECOMMENDED IF THE T-MATRICES BEING USED ARE VERY     CSL03030
C                     LARGE BECAUSE IN THAT CASE THE TRIDIAGONAL            CSL03040
C                     EIGENVALUE COMPUTATIONS ARE VERY EXPENSIVE.           CSL03050
C                                                                           CSL03060
C                 (1) COMPUTED EIGENVALUES OF T(1,MEV) AND OF T(2,MEV)      CSL03070
C                     WILL BE SAVED ON FILE 10.  THIS IS RECOMMENDED        CSL03080
C                     BECAUSE ONCE THESE T-EIGENVALUES ARE COMPUTED THE     CSL03090
C                     LATTER PORTION OF THE EIGENVALUE PROGRAM IS EASILY     CSL03100
C                     RESTARTED FROM THE POINT OF THESE EIGENVALUE          CSL03110
C                     COMPUTATIONS.                                         CSL03120
C                                                                           CSL03130
C     THE PROGRAM ALWAYS MAKES A SEPARATE LIST OF THE COMPUTED GOOD         CSL03140
C     T-EIGENVALUES ALONG WITH THEIR MINIMAL GAPS AND WRITES THEM OUT       CSL03150
C     TO FILE 3.  CORRESPONDING ERROR ESTIMATES FOR ANY ISOLATED           CSL03160
C     GOOD T-EIGENVALUES ARE ALWAYS WRITTEN TO FILE 4.                      CSL03170
C                                                                           CSL03180
C                                                                           CSL03190
C-----INPUT/OUTPUT FILES FOR EIGENVALUE PROGRAMS----------------------CSL03200
C                                                                           CSL03210
C     ANY INPUT DATA OTHER THAN THE ALPHA/BETA HISTORY OR PREVIOUSLY-       CSL03220
C     COMPUTED EIGENVALUES OF T(1,MEV) AND T(2,MEV) SHOULD BE STORED        CSL03230
C     ON FILE 5. SEE SAMPLE INPUT/OUTPUT FROM TYPICAL RUN.                  CSL03240
C     THE READ STATEMENTS IN THE GIVEN FORTRAN PROGRAM ASSUME THAT          CSL03250
C     THE DATA STORED ON FILE 5 IS IN FREE FORMAT.  USER SHOULD NOTE        CSL03260
```

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

```
C              NEAREST DISTINCT EIGENVALUES, AND THE T-MULTIPLICITY    CSL03820
C              PATTERN OF THE GOOD AND THE SPURIOUS T-EIGENVALUES.     CSL03830
C              FILE 11 IS WRITTEN ONLY IF IDIST = 1.                   CSL03840
C                                                                      CSL03850
C                                                                      CSL03860
C-----PARAMETERS SET BY THE EIGENVALUE PROGRAM-------------------------CSL03870
C                                                                      CSL03880
C     THESE PARAMETERS ARE SET INTERNALLY IN THE PROGRAM              CSL03890
C                                                                      CSL03900
C     SCALEK    K = 1,2,3,4                                            CSL03910
C                                                                      CSL03920
C              THE SCALING FACTORS SCALEK HAVE BEEN INTRODUCED IN AN   CSL03930
C              ATTEMPT TO MAKE THE TOLERANCES USED IN THE              CSL03940
C              T-MULTIPLICITY, SPURIOUS, AND ISOLATION TESTS ADJUST    CSL03950
C              TO THE SCALE OF THE GIVEN MATRIX.  THESE FACTORS MUST   CSL03960
C              NOT BE MODIFIED.                                        CSL03970
C                                                                      CSL03980
C     BTOL = RELATIVE TOLERANCE USED TO ESTIMATE ANY LOSS OF LOCAL     CSL03990
C            ORTHOGONALITY OF THE LANCZOS VECTORS AFTER THE T-MATRIX   CSL04000
C            HAS BEEN GENERATED.  THE LANCZOS PROCEDURE WORKS WELL     CSL04010
C            ONLY IF LOCAL ORTHOGONALITY BETWEEN SUCCESSIVE LANCZOS    CSL04020
C            VECTORS IS MAINTAINED.  THE TNORM SUBROUTINE TESTS        CSL04030
C            WHETHER OR NOT                                            CSL04040
C                                                                      CSL04050
C                MINIMUM  |BETA(I)|/||A|| > BTOL.                      CSL04060
C                I=2,KMAX                                              CSL04070
C                                                                      CSL04080
C            IF THIS TEST IS VIOLATED BY SOME BETA AND A T-MATRIX THAT CSL04090
C            WOULD INCLUDE SUCH A BETA IS REQUESTED, THEN THE LANCZOS  CSL04100
C            PROCEDURE WILL TERMINATE FOR THE USER TO DECIDE WHAT TO   CSL04110
C            DO.  THE USER CAN OVER-RIDE THIS TEST BY SIMPLY DECREASING CSL04120
C            THE SIZE OF BTOL, BUT THEN CONVERGENCE IS NOT AS CERTAIN. CSL04130
C            THE PROGRAM SETS BTOL = 1.D-8 WHICH IS A VERY CONSERVATIVE CSL04140
C            CHOICE. THE || A || IS ESTIMATED BY USING                 CSL04150
C            AN ESTIMATE OF THE NORM OF THE T-MATRIX, T(1,KMAX).       CSL04160
C                                                                      CSL04170
C     GAPTOL = RELATIVE TOLERANCE USED IN THE SUBROUTINE ISOEV         CSL04180
C              TO DETERMINE WHICH OF THE GOOD T-EIGENVALUES NEED       CSL04190
C              ERROR ESTIMATES.  THE PROGRAM SETS GAPTOL = 1.D-7.      CSL04200
C              IF FOR A GIVEN 'GOOD' T-EIGENVALUE THE COMPUTED GAP     CSL04210
C              IS TOO SMALL AND IS DUE TO A 'SPURIOUS' T-EIGENVALUE    CSL04220
C              THEN THE 'GOOD' T-EIGENVALUE IS ASSUMED TO HAVE CONVERGEDCSL04230
C              AND NO ERROR ESTIMATES ARE COMPUTED.                    CSL04240
C                                                                      CSL04250
C-----USER-SPECIFIED PARAMETERS FOR EIGENVALUE PROGRAMS----------------CSL04260
C                                                                      CSL04270
C     RELTOL = RELATIVE TOLERANCE USED IN 'COMBINING' COMPUTED         CSL04280
C              EIGENVALUES OF T(1,MEV) PRIOR TO COMPUTING ERROR        CSL04290
C              ESTIMATES.                                              CSL04300
C                                                                      CSL04310
C     THE LUMPING OF T-EIGENVALUES OCCURS IN SUBROUTINE LUMP.          CSL04320
C     LUMPING IS NECESSARY BECAUSE IT IS IMPOSSIBLE TO ACCURATELY      CSL04330
C     PREDICT THE ACCURACY OF THE CMTQL1 SUBROUTINE.  LUMP 'COMBINES'  CSL04340
C     T-EIGENVALUES THAT HAVE SLIPPED BY THE TOLERANCE THAT WAS USED   CSL04350
C     IN THE T-MULTIPLICITY TESTS.  IN PARTICULAR IF FOR SOME J,       CSL04360
```

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

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

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

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

```
C                                                                     CSL06570
C     THEN, AS EACH OF THE LANCZOS VECTORS IS REGENERATED, ALL        CSL06580
C     OF THE RITZ VECTORS CORRESPONDING TO THESE                      CSL06590
C     T-EIGENVECTORS ARE UPDATED USING THE CURRENTLY-GENERATED        CSL06600
C     LANCZOS VECTOR.  LANCZOS VECTORS ARE GENERATED (NOTE            CSL06610
C     THAT THEY ARE NOT BEING KEPT), UNTIL ENOUGH HAVE                CSL06620
C     BEEN GENERATED TO MAP THE LONGEST T-EIGENVECTOR INTO ITS        CSL06630
C     CORRESPONDING RITZ VECTOR.  THE ARRAY RITVEC CONTAINS THE       CSL06640
C     SUCCESSIVE RITZ VECTORS WHICH ARE THE APPROXIMATE               CSL06650
C     EIGENVECTORS OF A.                                              CSL06660
C                                                                     CSL06670
C                                                                     CSL06680
C-----PARAMETER CONTROLS FOR EIGENVECTOR PROGRAMS---------------------CSL06690
C                                                                     CSL06700
C                                                                     CSL06710
C     PARAMETER CONTROLS ARE INTRODUCED TO ALLOW SEGMENTATION OF THE  CSL06720
C     EIGENVECTOR COMPUTATIONS AND TO ALLOW VARIOUS COMBINATIONS OF   CSL06730
C     READ/WRITES.                                                    CSL06740
C                                                                     CSL06750
C     THE FLAG MBOUND ALLOWS THE USER TO DETERMINE A FIRST GUESS ON THE CSL06760
C     STORAGE THAT WILL BE REQUIRED BY THE T-EIGENVECTORS FOR THE     CSL06770
C     EIGENVALUES WHOSE EIGENVECTORS ARE TO BE COMPUTED.              CSL06780
C     THIS CAN BE USED TO ESTIMATE THE REQUIRED SIZE OF THE TVEC ARRAY. CSL06790
C                                                                     CSL06800
C     MBOUND = (0,1) MEANS                                            CSL06810
C                                                                     CSL06820
C            (0)  PROGRAM COMPUTES FIRST GUESSES AT THE SIZES         CSL06830
C                 OF THE T-MATRICES REQUIRED BY EACH OF THE           CSL06840
C                 EIGENVALUES SUPPLIED AND THEN CONTINUES WITH        CSL06850
C                 THE CORRESPONDING T-EIGENVECTOR COMPUTATIONS.       CSL06860
C                                                                     CSL06870
C            (1)  PROGRAM COMPUTES FIRST GUESSES AT THE SIZES         CSL06880
C                 OF THE T-MATRICES REQUIRED BY EACH OF THE           CSL06890
C                 EIGENVALUES SUPPLIED, STORES THESE IN FILE 10       CSL06900
C                 AND THEN TERMINATES.  THE USER CAN USE THESE        CSL06910
C                 SIZES TO ESTIMATE THE SIZE TVEC ARRAY NEEDED        CSL06920
C                 FOR THE DESIRED T-EIGENVECTOR COMPUTATIONS.         CSL06930
C                                                                     CSL06940
C     THE FLAGS NTVCON, TVSTOP, LVCONT, AND ERCONT CONTROL THE STOPPING CSL06950
C     CRITIERIA FOR INTERMEDIATE POINTS IN THE LANCZOS PROCEDURE.     CSL06960
C     THEY CAUSE TERMINATION OF THE LANCZOS PROCEDURE IF VARIOUS      CSL06970
C     QUANTITIES CANNOT BE COMPUTED AS DESIRED.                       CSL06980
C                                                                     CSL06990
C     NTVCON = (0,1) MEANS                                            CSL07000
C                                                                     CSL07010
C            (0)  IF THE ESTIMATED STORAGE FOR THE T-EIGENVECTORS     CSL07020
C                 EXCEEDS THE USER-SPECIFIED DIMENSION OF THE         CSL07030
C                 TVEC ARRAY PROGRAM DOES NOT CONTINUE WITH THE       CSL07040
C                 T-EIGENVECTOR COMPUTATIONS. TERMINATION OCCURS.     CSL07050
C                                                                     CSL07060
C            (1)  CONTINUE WITH THE T-EIGENVECTOR COMPUTATIONS        CSL07070
C                 EVEN IF THE ESTIMATED STORAGE FOR TVEC EXCEEDS      CSL07080
C                 THE USER-SPECIFIED DIMENSION OF THE TVEC ARRAY.     CSL07090
C                 IN THIS SITUATION THE PROGRAM COMPUTES AS MANY      CSL07100
C                 T-EIGENVECTORS AS IT HAS ROOM FOR, IN THE SAME      CSL07110
```

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

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

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

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

```
C                    ARE REQUESTED. THESE EIGENVALUES ARE READ IN        CSL09320
C                    FROM FILE 3.                                        CSL09330
C                                                                        CSL09340
C     GR(J),GC(J)    = REAL*8 ARRAYS WHOSE DIMENSION MUST BE AT          CSL09350
C                       LEAST MAX(N,KMAX).  USED TO HOLD RANDOMLY-       CSL09360
C                       GENERATED STARTING VECTORS FOR LANCZS            CSL09370
C                       COMPUTATIONS AND FOR THE INVERM SUBROUTINE.      CSL09380
C                                                                        CSL09390
C     AMINGP(J), = REAL*4 ARRAYS OF DIMENSION AT LEAST NGOOD.            CSL09400
C     TMINGP(J)    CONTAIN, RESPECTIVELY, THE MINIMAL GAPS FOR           CSL09410
C                  CORRESPONDING EIGENVALUES IN GOODEV ARRAY IN          CSL09420
C                  A-MATRIX AND IN T-MATRIX.                             CSL09430
C                                                                        CSL09440
C     TERR(J), ERR(J),    = REAL*4 ARRAYS (EXCEPT TLAST WHICH IS         CSL09450
C     ERRDGP(J), TLAST(J)   REAL*8) EACH OF WHOSE DIMENSIONS MUST BE     CSL09460
C     RNORM(J), TBETA(J)    AT LEAST NGOOD.  USED TO STORE QUANTITIES    CSL09470
C                           GENERATED DURING THE COMPUTATIONS FOR        CSL09480
C                           LATER PRINTOUT.                              CSL09490
C                                                                        CSL09500
C     G(J)     = REAL*4 ARRAY WHOSE DIMENSION MUST BE AT LEAST           CSL09510
C                MAX(KMAX,N).  USED IN SUBROUTINE GENRAN TO HOLD         CSL09520
C                RANDOM NUMBERS NEEDED FOR THE LANCZOS VECTORS           CSL09530
C                REGENERATION AND FOR THE INVERSE ITERATION              CSL09540
C                COMPUTATIONS IN THE SUBROUTINE INVERM.                  CSL09550
C                                                                        CSL09560
C     MP(J) = INTEGER*4 ARRAY WHOSE DIMENSION IS AT LEAST NGOOD.         CSL09570
C             INITIALLY CONTAINS THE T-MULTIPLICITY OF THE EIGENVALUE    CSL09580
C             GOODEV(J) AS AN EIGENVALUE OF THE T-MATRIX T(1,MEV).       CSL09590
C             USED TO FLAG EIGENVALUES FOR WHICH NO T-EIGENVECTOR        CSL09600
C             OR NO RITZ VECTOR IS TO BE COMPUTED.                       CSL09610
C                                                                        CSL09620
C     MA(J)   = INTEGER*4 ARRAYS EACH OF WHOSE DIMENSIONS                CSL09630
C               IS AT LEAST NGOOD.  USED IN DETERMINING                  CSL09640
C               AN APPROPRIATE T-MATRIX FOR EACH EIGENVALUE              CSL09650
C               IN GOODEV ARRAY.                                         CSL09660
C                                                                        CSL09670
C     MINT(J),MFIN(J) = INTEGER*4 ARRAYS WHOSE DIMENSIONS MUST BE AT     CSL09680
C                       LEAST NGOOD.  USED TO POINT TO THE BEGINNINGS    CSL09690
C                       AND THE ENDS OF THE COMPUTED EIGENVECTOR         CSL09700
C                       OF THE T-MATRIX, T(1,|MA(J)|).                   CSL09710
C                                                                        CSL09720
C     IDELTA(J)  = INTEGER*4 ARRAY WHOSE DIMENSION MUST BE AT            CSL09730
C                  LEAST NGOOD.  CONTAINS INCREMENTS USED IN LOOPS       CSL09740
C                  ON APPROPRIATE SIZE T-MATRIX FOR THE T-EIGENVECTOR    CSL09750
C                  COMPUTATIONS.                                         CSL09760
C                                                                        CSL09770
C                                                                        CSL09780
C     INTERC(J)  = INTEGER*4 ARRAY WHOSE DIMENSION MUST BE AT            CSL09790
C                  LEAST KMAX.  WORK SPACE USED IN INVERM.               CSL09800
C                                                                        CSL09810
C-----SUBROUTINES INCLUDED FOR THE EIGENVECTOR COMPUTATIONS------------CSL09820
C                                                                        CSL09830
C                                                                        CSL09840
C     INVERM = FOR THE T-SIZES CONSIDERED BY THE PROGRAM COMPUTES        CSL09850
C              THE CORRESPONDING EIGENVECTORS OF THESE T-MATRICES        CSL09860
```

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

7.3   CSLEVAL: Main Program, Eigenvalue Computations

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

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

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

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

```
C                                                                   CSL02100
C     READ IN ALPHA BETA HISTORY                                    CSL02110
C     HISTORY MUST BE STORED IN MACHINE FORMAT TO PREVENT           CSL02120
C     ERRORS CAUSED BY INPUT/OUTPUT CONVERSIONS.                    CSL02130
C                                                                   CSL02140
      READ(2,100)MOLD,NOLD,SVSOLD,MATOLD                            CSL02150
  100 FORMAT(2I6,I12,I8)                                            CSL02160
C                                                                   CSL02170
      IF (KMAX.LT.MOLD) KMAX = MOLD                                 CSL02180
      KMAX1 = KMAX + 1                                              CSL02190
C                                                                   CSL02200
C     CHECK THAT ORDER N, MATRIX ID MATNO, AND RANDOM SEED SVSEED   CSL02210
C     AGREE WITH THOSE IN THE HISTORY FILE.  IF NOT PROCEDURE STOPS.CSL02220
C                                                                   CSL02230
      ITEMP = (NOLD-N)**2+(MATNO-MATOLD)**2+(SVSEED-SVSOLD)**2       CSL02240
C                                                                   CSL02250
      IF (ITEMP.EQ.0) GO TO 120                                     CSL02260
C                                                                   CSL02270
      WRITE(6,110)                                                  CSL02280
  110 FORMAT(' PROGRAM TERMINATES'/    ' READ FROM FILE 2 CORRESPONDS TOCSL02290
     1 DIFFERENT MATRIX THAN MATRIX SPECIFIED'/)                    CSL02300
      GO TO 650                                                     CSL02310
C                                                                   CSL02320
  120 CONTINUE                                                      CSL02330
      MOLD1 = MOLD+1                                                CSL02340
C                                                                   CSL02350
      READ(2,130)(ALPHA(J), J=1,MOLD)                               CSL02360
      READ(2,130)(BETA(J), J=1,MOLD1)                               CSL02370
  130 FORMAT(4Z20)                                                  CSL02380
C                                                                   CSL02390
      IF (KMAX.EQ.MOLD) GO TO 160                                   CSL02400
C                                                                   CSL02410
      READ(2,130)(V1(J), J=1,N)                                     CSL02420
      READ(2,130)(V2(J), J=1,N)                                     CSL02430
C                                                                   CSL02440
  140 CONTINUE                                                      CSL02450
      IIX = SVSEED                                                  CSL02460
C                                                                   CSL02470
C-------------------------------------------------------------------CSL02480
C                                                                   CSL02490
      CALL LANCZS(CMATV,V1,V2,ALPHA,BETA,GR,GC,G,KMAX,MOLD1,N,IIX)  CSL02500
C                                                                   CSL02510
C-------------------------------------------------------------------CSL02520
C                                                                   CSL02530
      KMAX1 = KMAX + 1                                              CSL02540
C                                                                   CSL02550
      IF (IHIS.EQ.0.AND.ISTOP.GT.0) GO TO 160                       CSL02560
C                                                                   CSL02570
      WRITE(1,150) KMAX,N,SVSEED,MATNO                              CSL02580
  150 FORMAT(2I6,I12,I8,' = KMAX,N,SVSEED,MATNO')                   CSL02590
C                                                                   CSL02600
      WRITE(1,130)(ALPHA(I), I=1,KMAX)                              CSL02610
      WRITE(1,130)(BETA(I), I=1,KMAX1)                              CSL02620
C                                                                   CSL02630
      WRITE(1,130)(V1(I), I=1,N)                                    CSL02640
```

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

```
C                                                                  CSL03200
  210 TEMP = T0/TKMAX                                              CSL03210
      IBMEV = IABS(IBMEV)                                          CSL03220
      IF (TEMP.GE.BTOL) GO TO 220                                  CSL03230
      IBMEV = -IBMEV                                               CSL03240
      GO TO 590                                                    CSL03250
  220 CONTINUE                                                     CSL03260
C                                                                  CSL03270
C------------------------------------------------------------------CSL03280
C     SUBROUTINE COMPEV CALLS SUBROUTINE CMTQL1 TO COMPUTE THE     CSL03290
C     T-EIGENVALUES.  COMPEV THEN APPLIES THE T-MULTIPLICITY AND   CSL03300
C     SPURIOUS TESTS TO THE COMPUTED T-EIGENVALUES.  HERE INITIALIZECSL03310
C     THE TOLERANCES USED IN THE T-MULTIPLICITY AND THE SPURIOUS   CSL03320
C     TESTS.  THE MAX(|LAMBDA(T(1,MEV)|) WILL BE INCORPORATED      CSL03330
C     INSIDE THE SUBROUTINE COMPEV.  NOTE THAT THE OBSERVED ACCURACYCSL03340
C     OF THE COMPUTED T-EIGENVALUES FOR THE COMPLEX SYMMETRIC CASE CSL03350
C     IS APPROXIMATELY 3 DIGITS LESS THAN THAT ACHIEVED IN THE REAL CSL03360
C     CASE. THUS, A FACTOR OF 500 HAS BEEN INTRODUCED.  THIS HOWEVERCSL03370
C     MEANS THAT THIS TEST IS NOT AS SHARP AS IT WAS IN THE        CSL03380
C     REAL SYMMETRIC AND HERMITIAN CASES.  THUS, IT HAS LOWER      CSL03390
C     RESOLUTION AND CAN OCCASIONALLY MAKE A MISTAKE.             CSL03400
C                                                                  CSL03410
      MULTOL = 500.D0 * DFLOAT(MEV+1000) * EPSM                    CSL03420
      SPUTOL =   MULTOL                                            CSL03430
C                                                                  CSL03440
C     ON RETURN FROM COMPEV                                        CSL03450
C     NDIS = NUMBER OF DISTINCT EIGENVALUES OF T(1,MEV)           CSL03460
C     VS = DISTINCT T-EIGENVALUES IN INCREASING ORDER OF MAGNITUDE CSL03470
C     GR(K) = |VS(K)|, K = 1,NDIS,  GR(K).LE.GR(K+1)             CSL03480
C     MP = T-MULTIPLICITIES OF THE T-EIGENVALUES IN VS            CSL03490
C     MP(I) = (0,1,MI), MI>1, I=1,NDIS  MEANS:                    CSL03500
C         (0)  VS(I) IS SPURIOUS                                  CSL03510
C         (1)  VS(I) IS SIMPLE AND GOOD                           CSL03520
C         (MI) VS(I) IS T-MULTIPLE AND IS THEREFORE NOT ONLY GOOD BUTCSL03530
C              ALSO A CONVERGED GOOD T-EIGENVALUE.                CSL03540
C                                                                  CSL03550
C                                                                  CSL03560
      CALL COMPEV(ALPHA,BETA,V1,V2,VS,GR,MULTOL,SPUTOL,MP,MP2,     CSL03570
     1MEV,NDIS,SAVTEV)                                             CSL03580
C                                                                  CSL03590
C------------------------------------------------------------------CSL03600
C                                                                  CSL03610
      IF (NDIS.EQ.0) GO TO 630                                     CSL03620
C                                                                  CSL03630
C     ON EXIT FROM COMPEV MULTOL AND SPUTOL SHOULD BE SCALED       CSL03640
C     BY THE SIZES OF THE T-EIGENVALUES                           CSL03650
      EVMAX = GR(NDIS)                                             CSL03660
      LOOP = NDIS                                                  CSL03670
C                                                                  CSL03680
C------------------------------------------------------------------CSL03690
C                                                                  CSL03700
      CALL LUMP(VS,V1,GR,RELTOL,SPUTOL,SCALE2,MP,MP2,LOOP)         CSL03710
C                                                                  CSL03720
C------------------------------------------------------------------CSL03730
C                                                                  CSL03740
```

```
      IF (LOOP.LT.0) GO TO 650                                    CSL03750
C                                                                 CSL03760
      IF (NDIS.EQ.LOOP) GO TO 240                                 CSL03770
C                                                                 CSL03780
      WRITE(6,230) NDIS,LOOP,MEV                                  CSL03790
  230 FORMAT(/' AFTER LUMP NDIS,LOOP,MEV = ',3I6/)                CSL03800
C                                                                 CSL03810
  240 CONTINUE                                                    CSL03820
      NDIS = LOOP                                                 CSL03830
C                                                                 CSL03840
C-----------------------------------------------------------------CSL03850
C     CALCULATE MINGAPS FOR DISTINCT T(1,MEV) EIGENVALUES.        CSL03860
C     ON EXIT |GG(K)| = MIN(J.NE.K,|VS(K)-VS(J)|), MP2(K)=J INDEX CSL03870
C     FOR MINIMUM.  GG(K)< 0 MEANS NEAREST NEIGHBOR IS SPURIOUS.  CSL03880
      IGAP = 0                                                    CSL03890
      ITAG = 1                                                    CSL03900
C                                                                 CSL03910
      CALL COMGAP(VS,GR,GG,MP,MP2,NDIS,IGAP,ITAG)                 CSL03920
C                                                                 CSL03930
C-----------------------------------------------------------------CSL03940
C                                                                 CSL03950
C     SET CONVERGENCE CRITIERION                                  CSL03960
      TTOL = EPSM * EVMAX                                         CSL03970
      CONTOL = CDABS(BETAM)*1.D-10                                CSL03980
C                                                                 CSL03990
  250 CONTINUE                                                    CSL04000
      BETA(MP1) = BETAM                                           CSL04010
C                                                                 CSL04020
C-----------------------------------------------------------------CSL04030
C     THE SUBROUTINE ISOEV LABELS THOSE SIMPLE EIGENVALUES OF T(1,MEV) CSL04040
C     WITH VERY SMALL GAPS BETWEEN NEIGHBORING EIGENVALUES OF T(1,MEV) CSL04050
C     TO AVOID COMPUTING ERROR ESTIMATES FOR ANY SIMPLE GOOD      CSL04060
C     T-EIGENVALUE THAT IS TOO CLOSE TO A SPURIOUS T-EIGENVALUE.  CSL04070
C     MP(I) = -1 MEANS THAT THE GOOD T-EIGENVALUE IS SIMPLE AND   CSL04080
C     IS TOO CLOSE TO A SPURIOUS T-EIGENVALUE.                    CSL04090
C                                                                 CSL04100
C     NG = NUMBER OF GOOD T-EIGENVALUES.                          CSL04110
C     NISO = NUMBER OF ISOLATED GOOD T-EIGENVALUES.               CSL04120
C     GG = MINIMAL GAPS IN T(1,MEV)                               CSL04130
C     GR(K) = |VS(K)|, K=1,NDIS                                   CSL04140
C                                                                 CSL04150
      CALL ISOEV(VS,GR,GG,GAPTOL,SPUTOL,SCALE1,MP,NDIS,NG,NISO)   CSL04160
C                                                                 CSL04170
C-----------------------------------------------------------------CSL04180
C                                                                 CSL04190
      WRITE(6,260)NG,NISO,NDIS                                    CSL04200
  260 FORMAT(/I6,' GOOD T-EIGENVALUES HAVE BEEN COMPUTED'/        CSL04210
     1 I6,' OF THESE ARE ISOLATED'/                               CSL04220
     2 I6,' = NUMBER OF DISTINCT T-EIGENVALUES COMPUTED'/)        CSL04230
C                                                                 CSL04240
C     DO WE WRITE DISTINCT EIGENVALUES OF T-MATRIX TO FILE 11?    CSL04250
      IF (IDIST.EQ.0) GO TO 300                                   CSL04260
C                                                                 CSL04270
      WRITE(11,270) NDIS,NISO,MEV,N,SVSEED,MATNO                  CSL04280
  270 FORMAT(/4I6,I12,I8,' = NDIS,NISO,MEV,N,SVSEED,MATNO'/)      CSL04290
```

```
C                                                                 CSL04300
      WRITE(11,280) (I,MP(I),VS(I),GG(I),MP2(I), I=1,NDIS)        CSL04310
  280 FORMAT(I4,I4,2E20.12,E12.3,I6)                              CSL04320
C                                                                 CSL04330
      WRITE(11,290) NDIS, (MP(I), I=1,NDIS)                       CSL04340
  290 FORMAT(/I6,' = NDIS, T-MULTIPLICITIES (0 MEANS  SPURIOUS)'/(20I4))CSL04350
C                                                                 CSL04360
  300 CONTINUE                                                    CSL04370
C                                                                 CSL04380
      IF (NISO.NE.0) GO TO 330                                    CSL04390
C                                                                 CSL04400
      WRITE(4,310) MEV                                            CSL04410
  310 FORMAT(/' AT MEV = ',I6,' THERE ARE NO ISOLATED T-EIGENVALUES'/ CSL04420
     1' SO NO ERROR ESTIMATES WERE COMPUTED/')                   CSL04430
C                                                                 CSL04440
      WRITE(6,320)                                                CSL04450
  320 FORMAT(/' ALL COMPUTED GOOD T-EIGENVALUES ARE T-MULTIPLE'/ CSL04460
     1 ' THEREFORE THESE EIGENVALUES ARE ASSUMED TO HAVE CONVERGED')CSL04470
C                                                                 CSL04480
      ICONV = 1                                                   CSL04490
      GO TO 370                                                   CSL04500
C                                                                 CSL04510
  330 CONTINUE                                                    CSL04520
C                                                                 CSL04530
C-----------------------------------------------------------------CSL04540
C    SUBROUTINE INVERR COMPUTES ERROR ESTIMATES FOR ISOLATED GOOD CSL04550
C    T-EIGENVALUES USING INVERSE ITERATION ON T(1,MEV). ON RETURN CSL04560
C    GG(J) = MINIMUM GAP IN T(1,MEV) FOR EACH VS(J), J=1,NDIS     CSL04570
C    G(I) = |BETAM|*|U(MEV)| = ERROR ESTIMATE FOR ISOLATED GOOD   CSL04580
C             T-EIGENVALUES, WHERE I = 1, NISO AND  BETAM = BETA(MEV+1)CSL04590
C             U(MEV) IS MEVTH COMPONENT OF THE UNIT EIGENVECTOR OF T CSL04600
C             CORRESPONDING TO THE ITH ISOLATED GOOD T-EIGENVALUE. CSL04610
C    A NEGATIVE ERROR ESTIMATE MEANS THAT FOR THAT PARTICULAR     CSL04620
C    T-EIGENVALUE THE INVERSE ITERATION DID NOT CONVERGE IN <= MXINIT CSL04630
C    STEPS AND THAT THE CORRESPONDING ERROR ESTIMATE IS QUESTIONABLE. CSL04640
C                                                                 CSL04650
C    ON EXIT                                                      CSL04660
C    V2 CONTAINS THE ISOLATED GOOD T-EIGENVALUES                  CSL04670
C    GR CONTAINS THE MINGAPS TO THE NEAREST DISTINCT EIGENVALUE   CSL04680
C        OF T(1,MEV) FOR EACH ISOLATED GOOD T-EIGENVALUE IN V2.   CSL04690
C    VS CONTAINS THE NDIS DISTINCT EIGENVALUES OF T(1,MEV)        CSL04700
C    MP CONTAINS THE CORRESPONDING CODED T-MULTIPLICITIES         CSL04710
C                                                                 CSL04720
      IT = MXINIT                                                 CSL04730
C                                                                 CSL04740
      CALL INVERR(ALPHA,BETA,V1,V2,VS,EPSM,GR,GC,G,GG,MP,MP2,MEV,MMB, CSL04750
     1NDIS,NISO,N,RHSEED,IT,IWRITE)                              CSL04760
C                                                                 CSL04770
C-----------------------------------------------------------------CSL04780
C                                                                 CSL04790
C    SIMPLE CHECK FOR CONVERGENCE. CHECKS TO SEE IF ALL OF THE ERROR CSL04800
C    ESTIMATES ARE SMALLER THAN CONTOL = CDABS(BETA(MEV+1)*1.D-10 CSL04810
C    IF THIS TEST IS SATISFIED, THEN CONVERGENCE FLAG, ICONV IS SET CSL04820
C    TO 1.  TYPICALLY ERROR ESTIMATES ARE VERY CONSERVATIVE.     CSL04830
C                                                                 CSL04840
```

```
      WRITE(6,340) CONTOL                                        CSL04850
  340 FORMAT(/' CONVERGENCE IS TESTED USING THE CONVERGENCE TOLERANCE', CSL04860
     1E13.4/)                                                    CSL04870
C                                                                CSL04880
      DO 350 I = 1,NISO                                          CSL04890
      IF (ABS(G(I)).GT.CONTOL) GO TO 370                         CSL04900
  350 CONTINUE                                                   CSL04910
      ICONV = 1                                                  CSL04920
      MMB = NMEVS                                                CSL04930
C                                                                CSL04940
      WRITE(6,360) CONTOL                                        CSL04950
  360 FORMAT(' ALL COMPUTED ERROR ESTIMATES WERE LESS THAN',E15.4/ CSL04960
     1 ' THEREFORE PROCEDURE TERMINATES'/)                       CSL04970
C                                                                CSL04980
  370 CONTINUE                                                   CSL04990
C                                                                CSL05000
C     IN REAL SYMMETRIC AND HERMITIAN LANCZOS PROGRAMS           CSL05010
C     AT THIS CORRESPONDING POINT THE SUBROUTINE PRTEST IS CALLED CSL05020
C     TO IDENTIFY ANY T-EIGENVALUES THAT MAY HAVE BEEN MISLABELLED CSL05030
C     AS SPURIOUS BECAUSE THEIR PROJECTIONS ON THE STARTING VECTOR CSL05040
C     WERE TOO SMALL.  THIS CHECK WAS MADE ONLY AFTER CONVERGENCE CSL05050
C     HAD OCCURRED.  HOWEVER, THE PRTEST SUBROUTINE IS BASED UPON CSL05060
C     STURM SEQUENCING AND THAT IS NOT VALID FOR COMPLEX SYMMETRIC CSL05070
C     MATRICES.  PERHAPS THERE IS SOME RECTANGLE ANALOG OF THE    CSL05080
C     PRTEST BUT WE HAVE NOT ATTEMPTED TO IDENTIFY AND INCLUDE    CSL05090
C     SUCH A TEST BECAUSE WE EXPECT, AS IN THE REAL SYMMETRIC AND CSL05100
C     HERMITIAN CASES THAT HIDDEN EIGENVALUES WILL BE RARE.       CSL05110
C                                                                CSL05120
C     WRITE THE GOOD T-EIGENVALUES TO FILE 3.  FIRST TRANSFER THEM CSL05130
C     TO V2 AND THEIR T-MULTIPLICITIES TO THE CORRESPONDING POSITIONS CSL05140
C     IN MP AND COMPUTE THE A-MINGAPS, THE MINIMAL GAPS BETWEEN THE CSL05150
C     GOOD T-EIGENVALUES.  THESE GAPS WILL BE PUT IN THE ARRAY GG. CSL05160
C     NOTE THAT AFTER THE SECOND CALL TO COMGAP THE ARRAY GC      CSL05170
C     WILL CONTAIN THE CORRESPONDING MINIMAL GAPS IN THE         CSL05180
C     T-MATRIX, T(1,MEV).                                        CSL05190
C                                                                CSL05200
  380 CONTINUE                                                   CSL05210
C                                                                CSL05220
      NG = 0                                                     CSL05230
      DO 390 I = 1,NDIS                                          CSL05240
      IF (MP(I).EQ.0) GO TO 390                                  CSL05250
      NG = NG+1                                                  CSL05260
      MP(NG) = MP(I)                                             CSL05270
      V2(NG) = VS(I)                                             CSL05280
      GC(NG) = GG(I)                                             CSL05290
  390 CONTINUE                                                   CSL05300
C                                                                CSL05310
      DO 400 I = 1,NG                                            CSL05320
  400 GR(I) = CDABS(V2(I))                                       CSL05330
C                                                                CSL05340
C----------------------------------------------------------------CSL05350
C     CALCULATE MINGAPS FOR GOODEV                               CSL05360
C     ON EXIT GG(K) = MIN(J.NE.K,|V2(K)-V2(J)|), MP2(K)=J INDEX FOR MIN CSL05370
C     NG = NUMBER OF COMPUTED DISTINCT GOOD T-EIGENVALUES.       CSL05380
      IGAP = 0                                                   CSL05390
```

```
      ITAG = 0                                                  CSL05400
C                                                               CSL05410
      CALL COMGAP(V2,GR,GG,MP,MP2,NG,IGAP,ITAG)                 CSL05420
C                                                               CSL05430
C---------------------------------------------------------------CSL05440
C                                                               CSL05450
C     WRITE GOOD T-EIGENVALUES OUT TO FILE 3.                   CSL05460
C                                                               CSL05470
      WRITE(6,410)MEV                                           CSL05480
  410 FORMAT(//' EIGENVALUE CALCULATION AT MEV = ',I6,'  IS COMPLETE'/) CSL05490
C                                                               CSL05500
      WRITE(3,420)NG,NDIS,MEV,N,SVSEED,MATNO,MULTOL,SPUTOL,IB,BTOL CSL05510
  420 FORMAT(4I6,I12,I8,' = NG,NDIS,MEV,N,SVEED,MATNO'/         CSL05520
     1 2E15.5,I6,E13.4,' = MULTOL,SPUTOL,IB,BTOL'/             CSL05530
     1' EVNO',1X,'MULT',13X,'R(GOODEV)',13X,'I(GOODEV)',        CSL05540
     1 3X,'TMINGAP',3X,'AMINGAP',1X,'NEIGH')                    CSL05550
C                                                               CSL05560
      WRITE(3,430)(I,MP(I),V2(I),GC(I),GG(I),MP2(I), I=1,NG)    CSL05570
  430 FORMAT(2I5,2E22.14,2E10.3,I6)                             CSL05580
C                                                               CSL05590
C     ORDER GOODEV BY INCREASING GAP SIZE                       CSL05600
      DO 440 I = 1,NG                                           CSL05610
      MP(I) = I                                                 CSL05620
      V1(I) = V2(I)                                             CSL05630
      G(I) = GG(I)                                              CSL05640
  440 CONTINUE                                                  CSL05650
C                                                               CSL05660
C     WRITE(12,436)                                             CSL05670
  450 FORMAT(' MINGAPS FOR GOOD T-EIGENVALUES'/                 CSL05680
     1 1X,'EVNUM',1X,'NEIGH',15X,'R(EV)',15X,'I(EV)',4X,'MINGAP') CSL05690
C                                                               CSL05700
C     WRITE(12,439) (K,MP2(K),V2(K),G(K), K = 1,NG)             CSL05710
  460 FORMAT(2I6,2E20.12,E10.3)                                 CSL05720
C                                                               CSL05730
      DO 480 K = 2,NG                                           CSL05740
      KM1 = K-1                                                 CSL05750
      DO 470 L = 1,KM1                                          CSL05760
      KK = K-L                                                  CSL05770
      KP1 = KK+1                                                CSL05780
      IF (G(KP1).GE.G(KK)) GO TO 480                            CSL05790
      Z = V1(KK)                                                CSL05800
      V1(KK) = V1(KP1)                                          CSL05810
      V1(KP1) = Z                                               CSL05820
      GTEMP = G(KK)                                             CSL05830
      G(KK) = G(KP1)                                            CSL05840
      G(KP1) = GTEMP                                            CSL05850
      ITEMP = MP(KK)                                            CSL05860
      MP(KK) = MP(KP1)                                          CSL05870
      MP(KP1) = ITEMP                                           CSL05880
  470 CONTINUE                                                  CSL05890
  480 CONTINUE                                                  CSL05900
C                                                               CSL05910
C     WRITE(12,441)                                             CSL05920
      WRITE(3,490)                                              CSL05930
  490 FORMAT(' T-EIGENVALUES ORDERED BY INCREASING MINGAP'/     CSL05940
```

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

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

7.4 CSLEVEC: Main Program, Eigenvector Computations

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

```
C      IMPORTANT NOTE:  PROGRAM ALLOWS ENLARGEMENT OF THE ALPHA,BETA      CSL00450
C      ARRAYS.  IN PARTICULAR, IF ANY ONE OF THE EIGENVALUES SUPPLIED     CSL00460
C      IS T-SIMPLE AND NOT CLOSE TO A SPURIOUS T-EIGENVALUE, THE PROGRAM  CSL00470
C      REQUIRES THAT KMAX BE AT LEAST 11*MEV/8 + 12.  IF KMAX IS NOT      CSL00480
C      THIS LARGE, THEN THE PROGRAM WILL RESET KMAX TO THIS SIZE          CSL00490
C      AND EXTEND THE ALPHA, BETA HISTORY IF REQUIRED.                    CSL00500
C      THUS, THE DIMENSIONS OF THE ALPHA AND BETA ARRAYS MUST BE          CSL00510
C      LARGE ENOUGH TO ALLOW FOR THIS POSSIBILITY.                        CSL00520
C      REMEMBER THAT THE BETA ARRAY, BETA(J), IS SUCH THAT                CSL00530
C      J = 1,..., KMAX+1.  SO IF THE KMAX USED BY THE PROGRAM             CSL00540
C      IS TO BE 3000, THEN BETA MUST BE OF LENGTH AT LEAST 3001.          CSL00550
C                                                                         CSL00560
C-------------------------------------------------------------------CSL00570
       COMPLEX*16  V1(1600),V2(1600),RITVEC(10000),ZEROC,TEMPC            CSL00580
       COMPLEX*16  ALPHA(1600),BETA(1601),GOODEV(50),TVEC(20000)          CSL00590
       COMPLEX*16  EVAL,ALFA,BATA,SUMC                                    CSL00600
       DOUBLE PRECISION  GR(1600),GC(1600)                                CSL00610
       DOUBLE PRECISION  ERTOL,SUM,TEMP,BKMIN                             CSL00620
       DOUBLE PRECISION  MULTOL,SPUTOL,SCALE0,BTOL                        CSL00630
       DOUBLE PRECISION  ONE,ZERO,MACHEP,EPSM                             CSL00640
       DOUBLE PRECISION  RELTOL,ERROR,ERRMIN,TERROR,TLAST(50)             CSL00650
       REAL  G(1600),AMINGP(50),TMINGP(50),EXPLAN(20)                     CSL00660
       REAL  TERR(50),ERR(50),ERRDGP(50),RNORM(50),TBETA(50)              CSL00670
       INTEGER  MP(50),MA(50),ML(50),MINT(50),MFIN(50),IDELTA(50)         CSL00680
       INTEGER  SVSEED,SVSOLD,RHSEED                                      CSL00690
       INTEGER INTERC(1600)                                               CSL00700
       INTEGER  MBOUND,NTVCON,SVTVEC,TVSTOP,LVCONT,ERCONT,TFLAG           CSL00710
       DOUBLE PRECISION  DABS, DMAX1, DSQRT                               CSL00720
       REAL ABS                                                           CSL00730
       INTEGER  IABS                                                      CSL00740
C-------------------------------------------------------------------CSL00750
       EXTERNAL CMATV                                                     CSL00760
       DATA MACHEP/Z3410000000000000/                                     CSL00770
       EPSM = 2.D0*MACHEP                                                 CSL00780
C-------------------------------------------------------------------CSL00790
C                                                                         CSL00800
C      ARRAYS MUST BE DIMENSIONED AS FOLLOWS:                             CSL00810
C      1.  ALPHA:  >= KMAXN,  BETA:  >= (KMAXN+1) WHERE KMAXN, THE        CSL00820
C                  LARGEST SIZE T-MATRIX CONSIDERED BY THE PROGRAM,       CSL00830
C                  IS THE LARGER OF THE SIZE OF THE ALPHA, BETA HISTORY   CSL00840
C                  PROVIDED ON FILE 2 (IF ANY ) AND THE SIZE WHICH THE    CSL00850
C                  PROGRAM SPECIFIES INTERNALLY, THIS LATTER IS ALWAYS    CSL00860
C                  < = 11*MEV / 8  +  12, WHERE MEV IS THE SIZE           CSL00870
C                  T-MATRIX THAT WAS USED IN THE CORRESPONDING EIGENVALUE CSL00880
C                  COMPUTATIONS.                                          CSL00890
C      2.  V1:  >= MAX(N,KMAX)                                            CSL00900
C      3.  V2:  >= N                                                      CSL00910
C      4.  G, GR, GC:  >= MAX(N,KMAX)                                     CSL00920
C      5.  RITVEC:  >= N*NGOOD, WHERE NGOOD IS THE NUMBER OF EIGENVALUES  CSL00930
C                   SUPPLIED TO THIS PROGRAM.                             CSL00940
C      6.  TVEC:  >= CUMULATIVE LENGTH OF ALL THE T-EIGENVECTORS NEEDED   CSL00950
C                    TO GENERATE THE DESIRED RITZ VECTORS.  AN EDUCATED   CSL00960
C                    GUESS AT AN APPROPRIATE LENGTH CAN BE OBTAINED       CSL00970
C                    BY RUNNING THE PROGRAM WITH THE FLAG MBOUND = 1      CSL00980
C                    AND MULTIPLYING THE RESULTING SIZE BY 5/4.           CSL00990
```

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

```
C                       VECTORS REQUESTED.                           CSL01550
C        ERCONT = 0:  MEANS FOR ANY GIVEN EIGENVALUE, A RITZ VECTOR   CSL01560
C                       WILL NOT BE COMPUTED FOR THAT EIGENVALUE UNLESS CSL01570
C                       A T-EIGENVECTOR HAS BEEN IDENTIFIED WITH A LAST CSL01580
C                       COMPONENT WHICH SATISFIES THE SPECIFIED        CSL01590
C                       CONVERGENCE CRITERION.                         CSL01600
C        ERCONT = 1:  MEANS FOR ANY GIVEN EIGENVALUE, A RITZ VECTOR   CSL01610
C                       WILL BE COMPUTED.  IF A T-EIGENVECTOR CANNOT   CSL01620
C                       BE IDENTIFIED WHICH SATISFIES THE LAST         CSL01630
C                       COMPONENT CRITERION, THEN THE PROGRAM WILL     CSL01640
C                       USE THE T-VECTOR THAT CAME CLOSEST TO          CSL01650
C                       SATISFYING THE CRITERION                       CSL01660
C        IWRITE = 1:  EXTENDED OUTPUT OF INTERMEDIATE COMPUTATIONS     CSL01670
C                       IS WRITTEN TO FILE 6                           CSL01680
C        IREAD = 0:   ALPHA/BETA FILE IS REGENERATED.                 CSL01690
C        IREAD = 1:   ALPHA/BETA FILE USED IN EIGENVALUE COMPUTATIONS CSL01700
C                       IS READ IN AND EXTENDED IF NECESSARY.  IN BOTH CSL01710
C                       CASES IREAD = 0 OR 1, THE LANCZOS VECTORS ARE  CSL01720
C                       ALWAYS REGENERATED FOR THE RITZ VECTOR         CSL01730
C                       COMPUTATIONS                                   CSL01740
C                                                                      CSL01750
      READ(5,20) EXPLAN                                                CSL01760
      READ(5,*) MBOUND,NTVCON,SVTVEC,IREAD                             CSL01770
C                                                                      CSL01780
      READ(5,20) EXPLAN                                                CSL01790
      READ(5,*) TVSTOP,LVCONT,ERCONT,IWRITE                           CSL01800
      IF (TVSTOP.EQ.1) SVTVEC = 1                                      CSL01810
C                                                                      CSL01820
C     READ IN SEED (RHSEED) FOR GENERATING RANDOM STARTING VECTOR     CSL01830
C     FOR INVERSE ITERATION ON THE T-MATRICES.                        CSL01840
C                                                                      CSL01850
      READ(5,20) EXPLAN                                                CSL01860
      READ(5,*) RHSEED                                                 CSL01870
C                                                                      CSL01880
C     READ IN MATNO = MATRIX/RUN IDENTIFICATION NUMBER AND            CSL01890
C     N = ORDER OF A-MATRIX                                           CSL01900
C                                                                      CSL01910
      READ(5,20) EXPLAN                                                CSL01920
      READ(5,*) MATNO,N                                                CSL01930
C                                                                      CSL01940
C----------------------------------------------------------------------CSL01950
C     INITIALIZE THE ARRAYS FOR THE USER-SPECIFIED MATRIX             CSL01960
C     AND PASS THE STORAGE LOCATIONS OF THESE ARRAYS TO THE           CSL01970
C     MATRIX-VECTOR MULTIPLY SUBROUTINE CMATV.                        CSL01980
C                                                                      CSL01990
      CALL USPEC(N,MATNO)                                             CSL02000
C                                                                      CSL02010
C----------------------------------------------------------------------CSL02020
C                                                                      CSL02030
C     MASK UNDERFLOW AND OVERFLOW                                     CSL02040
      CALL MASK                                                       CSL02050
C                                                                      CSL02060
C----------------------------------------------------------------------CSL02070
C     WRITE RUN PARAMETERS OUT TO FILE 6                             CSL02080
C                                                                      CSL02090
```

```
      WRITE(6,30) MATNO,N                                         CSL02100
   30 FORMAT(/' MATRIX IDENTIFICATION NO. = ',I10,' ORDER OF A = ',I5)  CSL02110
C                                                                 CSL02120
      WRITE(6,40) MBOUND,NTVCON,SVTVEC,IREAD                      CSL02130
   40 FORMAT(/3X,'MBOUND',3X,'NTVCON',3X,'SVTVEC',3X,'IREAD'/3I9,I8)  CSL02140
C                                                                 CSL02150
      WRITE(6,50) TVSTOP,LVCONT,ERCONT,IWRITE                     CSL02160
   50 FORMAT(/3X,'TVSTOP',3X,'LVCONT',3X,'ERCONT',3X,'IWRITE'/4I9)  CSL02170
C                                                                 CSL02180
      WRITE(6,60) MDIMTV,MDIMRV,MBETA                             CSL02190
   60 FORMAT(/3X,'MDIMTV',3X,'MDIMRV',3X,'MBETA'/2I9,I8)          CSL02200
C                                                                 CSL02210
      WRITE(6,70) RELTOL,RHSEED                                   CSL02220
   70 FORMAT(/7X,'RELTOL',3X,'RHSEED'/E13.4,I9)                   CSL02230
C                                                                 CSL02240
C                                                                 CSL02250
C     FROM FILE 3 READ IN THE NUMBER OF EIGENVALUES (NGOOD) FOR WHICH  CSL02260
C     EIGENVECTORS ARE REQUESTED, THE ORDER (MEV) OF THE LANCZOS  CSL02270
C     TRIDIAGONAL MATRIX USED IN COMPUTING THESE EIGENVALUES, THE  CSL02280
C     ORDER (NOLD) OF THE USER-SPECIFIED MATRIX USED IN THE EIGENVALUE  CSL02290
C     COMPUTATIONS, THE SEED (SVSEED) USED FOR GENERATING THE STARTING  CSL02300
C     VECTOR THAT WAS USED IN THOSE LANCZOS EIGENVALUE COMPUTATIONS,  CSL02310
C     AND THE MATRIX/RUN IDENTIFICATION NUMBER (MATOLD) USED IN THOSE  CSL02320
C     COMPUTATIONS.  ALSO READ IN THE NUMBER (NDIS) OF DISTINCT   CSL02330
C     EIGENVALUES OF T(1,MEV) THAT WERE COMPUTED BUT THIS VALUE IS  CSL02340
C     NOT USED IN THE EIGENVECTOR COMPUTATIONS.                  CSL02350
C                                                                 CSL02360
      READ(3,80) NGOOD,NDIS,MEV,NOLD,SVSEED,MATOLD               CSL02370
   80 FORMAT(4I6,I12,I8)                                          CSL02380
C                                                                 CSL02390
C     READ IN THE TOLERANCES USED IN THE T-MULTIPLICITY AND SPURIOUS  CSL02400
C     TESTS DURING THE EIGENVALUE COMPUTATIONS.                  CSL02410
C     ALSO READ IN THE FLAG IB.  IF IB < 0, THEN SOME BETA(I) IN THE  CSL02420
C     T-MATRIX FILE PROVIDED ON FILE 2 FAILED THE ORTHOGONALITY   CSL02430
C     TEST IN THE TNORM SUBROUTINE.  USER SHOULD NOTE THAT THIS   CSL02440
C     PROGRAM PROCEEDS INDEPENDENTLY OF THE SIZES OF THE BETA USED.  CSL02450
C                                                                 CSL02460
      READ(3,90) MULTOL,SPUTOL,IB,BTOL                           CSL02470
   90 FORMAT(2E15.5,I6,E13.4)                                     CSL02480
C                                                                 CSL02490
      WRITE(6,100) MULTOL,SPUTOL                                  CSL02500
  100 FORMAT(/' MULTIPLICITY TOLERANCE USED IN THE T-EIGENVALUE COMPUTAT  CSL02510
     1IONS WAS',E13.4/' TOLERANCE USED IN SPURIOUS CHECK',E13.4)  CSL02520
C                                                                 CSL02530
C     CONTINUE WRITE TO FILE 6 OF THE PARAMETERS FOR THIS RUN     CSL02540
C                                                                 CSL02550
      WRITE(6,110)NGOOD,NDIS,MEV,NOLD,MATOLD,SVSEED,MULTOL,SPUTOL,IB,  CSL02560
     1BTOL                                                        CSL02570
  110 FORMAT(/' EIGENVALUES SUPPLIED ARE READ IN FROM FILE 3'/' FILE 3  CSL02580
     1HEADER IS'/4X,'NG',2X,'NDIS',3X,'MEV',2X,'NOLD',2X,'MATOLD',4X,  CSL02590
     1'SVSEED'/4I6,I8,I10/7X,'MULTOL',7X,'SPUTOL',6X,'IB',9X,'BTOL'/  CSL02600
     12E13.4,I8,E13.4)                                            CSL02610
C                                                                 CSL02620
C     IS THE ARRAY RITVEC LONG ENOUGH TO HOLD ALL OF THE DESIRED  CSL02630
C     RITZ VECTORS (APPROXIMATE EIGENVECTORS)?                    CSL02640
```

```
      NMAX = NGOOD*N                                            CSL02650
      IF(MBOUND.EQ.1) GO TO 120                                 CSL02660
      IF(TVSTOP.NE.1.AND.NMAX.GT.MDIMRV) GO TO 1310             CSL02670
C                                                               CSL02680
C     CHECK THAT THE ORDER N AND THE MATRIX IDENTIFICATION NUMBER CSL02690
C     MATNO SPECIFIED BY THE USER AGREE WITH THOSE READ IN FROM CSL02700
C     FILE 3.                                                   CSL02710
  120 ITEMP = (NOLD-N)**2+(MATOLD-MATNO)**2                     CSL02720
      IF (ITEMP.NE.0) GO TO 1330                                CSL02730
C                                                               CSL02740
C     READ IN FROM FILE 3, THE T(1,MEV)-MULTIPLICITIES OF THE   CSL02750
C     EIGENVALUES WHOSE EIGENVECTORS ARE TO BE COMPUTED, THE VALUES CSL02760
C     OF THESE EIGENVALUES AND THEIR MINIMAL GAPS AS EIGENVALUES CSL02770
C     OF THE USER-SPECIFIED MATRIX AND AS EIGENVALUES OF THE T-MATRIX. CSL02780
C                                                               CSL02790
      READ(3,20) EXPLAN                                         CSL02800
      READ(3,130) (MP(J),GOODEV(J),TMINGP(J),AMINGP(J), J=1,NGOOD) CSL02810
  130 FORMAT(5X,I5,2E22.14,2E10.3)                              CSL02820
C                                                               CSL02830
      WRITE(6,140) (J,GOODEV(J),MP(J),TMINGP(J),AMINGP(J), J=1,NGOOD) CSL02840
  140 FORMAT(/' EIGENVALUES READ IN, T-MULTIPLICITIES, T-GAPS AND A-GAPSCSL02850
     1 '/4X,' J ',15X,' EIGENVALUE',14X,'TMULT',4X,'  TMINGAP ',4X, CSL02860
     1'  AMINGAP '/(I6,2E20.12,I4,2E15.4))                      CSL02870
C                                                               CSL02880
C     READ IN ERROR ESTIMATES                                  CSL02890
      WRITE(6,170) MEV,SVSEED                                   CSL02900
C     CHECK WHETHER OR NOT THERE ARE ANY T-ISOLATED EIGENVALUES IN CSL02910
C     THE EIGENVALUES PROVIDED                                 CSL02920
      DO 150 J=1,NGOOD                                          CSL02930
      IF(MP(J).EQ.1) GO TO 160                                  CSL02940
  150 CONTINUE                                                  CSL02950
      GO TO 190                                                 CSL02960
  160 READ(4,20) EXPLAN                                         CSL02970
      READ(4,20) EXPLAN                                         CSL02980
      READ(4,20) EXPLAN                                         CSL02990
  170 FORMAT(/' THESE EIGENVALUES WERE COMPUTED USING A T-MATRIX OF CSL03000
     1ORDER ',I5/' AND SEED FOR RANDOM NUMBER GENERATOR =',I12) CSL03010
      READ(4,180) NISO                                          CSL03020
  180 FORMAT(18X,I6)                                            CSL03030
      READ(4,20) EXPLAN                                         CSL03040
      READ(4,20) EXPLAN                                         CSL03050
      READ(4,20) EXPLAN                                         CSL03060
  190 DO 220 J=1,NGOOD                                          CSL03070
      ERR(J) = 0.D0                                             CSL03080
      IF(MP(J).NE.1) GO TO 220                                  CSL03090
      READ(4,200) EVAL, ERR(J)                                  CSL03100
  200 FORMAT(10X,2E20.12,E14.3)                                 CSL03110
      IF(CDABS(EVAL - GOODEV(J)).LT.1.D-10) GO TO 220           CSL03120
      WRITE(6,210) EVAL,GOODEV(J)                               CSL03130
  210 FORMAT(' PROBLEM WITH READ IN OF ERROR ESTIMATES'/' EIGENVALUE REACSL03140
     1D IN',2E20.12,' DOES NOT MATCH GOODEV(J) ='/2E20.12)      CSL03150
      GO TO 1550                                                CSL03160
C                                                               CSL03170
  220 CONTINUE                                                  CSL03180
C                                                               CSL03190
```

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

```
         IF(MBETA.LE.KMAXN) GO TO 1530                              CSL03750
         IF(KMAX.GE.KMAXN )  GO TO 310                              CSL03760
         WRITE(6,300) KMAX, KMAXN                                   CSL03770
  300 FORMAT(' ENLARGE KMAX FROM ',I6,' TO ',I6)                    CSL03780
         MOLD1 = KMAX + 1                                           CSL03790
         KMAX = KMAXN                                               CSL03800
         GO TO 380                                                  CSL03810
C                                                                   CSL03820
  310 WRITE(6,320) KMAX                                             CSL03830
  320 FORMAT(/' T-MATRICES HAVE BEEN READ IN FROM FILE 2'/' THE LARGEST CSL03840
     1SIZE T-MATRIX ALLOWED IS',I6/)                               CSL03850
C                                                                   CSL03860
         IF(IREAD.EQ.1) GO TO 400                                  CSL03870
C                                                                   CSL03880
C     REGENERATE THE ALPHA AND BETA                                CSL03890
C                                                                   CSL03900
  330 MOLD1 = 1                                                     CSL03910
C                                                                   CSL03920
C     SET KMAX                                                      CSL03930
         DO 340 J = 1,NGOOD                                         CSL03940
         IF(MP(J).EQ.1) GO TO 360                                   CSL03950
  340 CONTINUE                                                      CSL03960
         KMAX = MEV + 12                                            CSL03970
         WRITE(6,350) KMAX                                          CSL03980
  350 FORMAT(/' ALL EIGENVALUES FOR WHICH EIGENVECTORS ARE TO BE COMPUTECSL03990
     1D ARE EITHER T-MULTIPLE OR CLOSE TO'/' A SPURIOUS EIGENVALUE. THERCSL04000
     1EFORE SET KMAX = MEV + 12 = ',I7)                            CSL04010
         GO TO 380                                                  CSL04020
C                                                                   CSL04030
  360 KMAXN = 11*MEV/8 + 12                                         CSL04040
         IF(MBETA.LE.KMAXN) GO TO 1530                              CSL04050
         WRITE(6,370) KMAXN                                         CSL04060
  370 FORMAT(' SET KMAX EQUAL TO ',I6)                              CSL04070
         KMAX = KMAXN                                               CSL04080
C                                                                   CSL04090
  380 WRITE(6,390) MOLD1,KMAX                                       CSL04100
  390 FORMAT(/' LANCZS SUBROUTINE GENRATES ALPHA(J), BETA(J+1), J =', CSL04110
     1 I6,' TO ', I6/)                                             CSL04120
C                                                                   CSL04130
C-------------------------------------------------------------------CSL04140
C                                                                   CSL04150
         CALL LANCZS(CMATV,V1,V2,ALPHA,BETA,GR,GC,G,KMAX,MOLD1,N,SVSEED) CSL04160
C                                                                   CSL04170
C-------------------------------------------------------------------CSL04180
C                                                                   CSL04190
  400 CONTINUE                                                      CSL04200
C                                                                   CSL04210
C     SIMPLE STURM SEQUENCING IS NOT VALID FOR COMPLEX SYMMETRIC    CSL04220
C     MATRICES.  THUS, THE STRATEGY USED HERE FOR SELECTING         CSL04230
C     APPROPRIATE SIZE T-MATRICES FOR THE EIGENVECTOR COMPUTATIONS  CSL04240
C     MUST BE DIFFERENT FROM THAT USED IN THE REAL SYMMETRIC,       CSL04250
C     HERMITIAN, AND SINGULAR VALUE CASES.  AS IN THOSE CASES,      CSL04260
C     FOR EACH EIGENVALUE, A FIRST GUESS IS SELECTED AND THEN       CSL04270
C     LOOPING ON THE SIZE OF THE T-EIGENVECTOR COMPUTATIONS         CSL04280
C     DETERMINES APPROPRIATE SIZES FOR THE EIGENVECTOR COMPUTATIONS. CSL04290
```

```
C     FIRST GUESSES AT APPROPRIATE SIZES ARE SPECIFIED BELOW.      CSL04300
C                                                                   CSL04310
      DO 430 J = 1,NGOOD                                            CSL04320
      EVAL = GOODEV(J)                                              CSL04330
C     COMPUTE A FIRST GUESS ON AN APPROPRIATE SIZE T-MATRIX EACH    CSL04340
C     EIGENVALUE.                                                   CSL04350
      IF(MP(J).GT.1) GO TO 410                                      CSL04360
C     EIGENVALUE IS T-SIMPLE                                        CSL04370
      IF(MP(J).EQ.MONE) GO TO 420                                   CSL04380
C     EIGENVALUE IS T-SIMPLE AND T-ISOLATED                         CSL04390
      MA(J) = (8*MEV)/9  +  1                                       CSL04400
      ML(J) = ((11*MEV)/8 + 12)                                     CSL04410
      GO TO 430                                                     CSL04420
C     EIGENVALUE IS T-MULTIPLE                                      CSL04430
  410 MA(J) = (5*MEV)/(4*MP(J))  +  1                               CSL04440
      ML(J) = (7*MEV)/(4*MP(J)) + 1                                 CSL04450
      GO TO 430                                                     CSL04460
C     EIGENVALUE IS T-SIMPLE AND NOT T-ISOLATED                     CSL04470
  420 MA(J) = (5*MEV)/8 + 1                                         CSL04480
      ML(J) = MEV                                                   CSL04490
  430 CONTINUE                                                      CSL04500
C                                                                   CSL04510
      IF (IWRITE.EQ.1) WRITE(6,440) (MA(JJ), JJ=1,NGOOD)           CSL04520
  440 FORMAT(/' 1ST GUESS AT APPROPRIATE SIZES FOR T-MATRICES '/    CSL04530
     1' ACTUAL VALUES WILL PROBABLY BE 1/4 AGAIN AS MUCH'/(13I6))   CSL04540
C                                                                   CSL04550
      WRITE(10,450) N,KMAX                                          CSL04560
  450 FORMAT(2I8,' = ORDER OF USER MATRIX AND MAX ORDER OF T(1,MEV)') CSL04570
C                                                                   CSL04580
      WRITE(10,460)                                                 CSL04590
  460 FORMAT(/' 1ST GUESS AT APPROPRIATE SIZES FOR T-MATRICES '/    CSL04600
     1' ACTUAL VALUES WILL PROBABLY BE 1/4 AGAIN AS MUCH'/)         CSL04610
      WRITE(10,470)                                                 CSL04620
  470 FORMAT(5X,'J',8X,'REAL(GOODEV)',8X,'IMAG(GOODEV)',7X,'MA(J)', CSL04630
     17X,'MP(J)')                                                   CSL04640
C                                                                   CSL04650
      WRITE(10,480) (J,GOODEV(J), MA(J), MP(J), J=1,NGOOD)         CSL04660
  480 FORMAT(I6,2E20.12,I12,I12)                                    CSL04670
C                                                                   CSL04680
      IF(MBOUND.EQ.1) WRITE(10,490)                                 CSL04690
  490 FORMAT(/' GOODEV(J) IS A GOOD EIGENVALUE OF T(1,MEV)'/        CSL04700
     1 ' IABS(MA(J)) = APPROPRIATE SIZE T-MATRIX FOR GOODEV(J)'/    CSL04710
     1 ' INITIAL VALUE OF MA(J) IS CHOSEN HEURISTICALLY'/           CSL04720
     1 ' PROGRAM LOOPS ON SIZE OF T-MATRIX TO GET BETTER SIZE'/     CSL04730
     1 ' END OF SIZES OF T-MATRICES FILE 10'///)                    CSL04740
C                                                                   CSL04750
C                                                                   CSL04760
C     TERMINATE AFTER COMPUTING 1ST GUESSES ON SIZES OF T-MATRICES  CSL04770
C     REQUIRED FOR THE GIVEN EIGENVALUES?                           CSL04780
      IF(MBOUND.EQ.1) GO TO 1390                                    CSL04790
C                                                                   CSL04800
C                                                                   CSL04810
C     IS THERE ROOM FOR ALL OF THE REQUESTED T-EIGENVECTORS?        CSL04820
      MTOL = 0                                                      CSL04830
      DO 500 J = 1,NGOOD                                            CSL04840
```

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

```
  560 ICOUNT = ICOUNT + 1                                      CSL05400
      MTOL = MTOL+KMAXU                                         CSL05410
C                                                              CSL05420
C     IS THERE ROOM IN TVEC ARRAY FOR THE NEXT T-EIGENVECTOR?  CSL05430
C     IF NOT, SKIP TO RITZ VECTOR COMPUTATIONS.                CSL05440
      IF (MTOL.GT.MDIMTV) GO TO 700                            CSL05450
C                                                              CSL05460
      IT = 3                                                   CSL05470
      KINT = MTOL - KMAXU +1                                   CSL05480
C                                                              CSL05490
C     RECORD THE BEGINNING AND END OF THE T-EIGENVECTOR BEING COMPUTED  CSL05500
      MINT(J) = KINT                                           CSL05510
      MFIN(J) = MTOL                                           CSL05520
C                                                              CSL05530
C--------------------------------------------------------------CSL05540
C     SUBROUTINE INVERM DOES INVERSE ITERATION, I.E. SOLVES    CSL05550
C     (T(1,KMAXU) - EVAL)*U = RHS  FOR EACH EIGENVALUE TO OBTAIN THE  CSL05560
C     DESIRED T-EIGENVECTOR.                                   CSL05570
C                                                              CSL05580
      IF(IWRITE.EQ.1)  WRITE(6,570) J                          CSL05590
  570 FORMAT(/I6,'TH EIGENVALUE')                              CSL05600
C                                                              CSL05610
      CALL INVERM(ALPHA,BETA,V1,TVEC(KINT),EVAL,ERROR,TERROR,EPSM,  CSL05620
     1     GR,GC,INTERC,KMAXU,IT,IWRITE)                       CSL05630
C                                                              CSL05640
C--------------------------------------------------------------CSL05650
C                                                              CSL05660
      TERR(J) = TERROR                                         CSL05670
      TLAST(J) = ERROR                                         CSL05680
      KMAXU1 = KMAXU + 1                                       CSL05690
      TBETA(J) = CDABS(BETA(KMAXU1))*ERROR                     CSL05700
C                                                              CSL05710
C     AFTER COMPUTING EACH OF THE T-EIGENVECTORS,              CSL05720
C     CHECK THE SIZE OF THE ERROR ESTIMATE, ERROR.             CSL05730
C     IF THIS ESTIMATE IS NOT AS SMALL AS DESIRED AND          CSL05740
C     |MA(J)| < ML(J), ATTEMPT TO INCREASE THE SIZE OF |MA(J)| CSL05750
C     AND REPEAT THE T-EIGENVECTOR COMPUTATIONS.               CSL05760
C                                                              CSL05770
      IF(ERROR.LT.ERTOL.OR.TFLAG.EQ.1)  GO TO 680              CSL05780
C                                                              CSL05790
      IF(ERROR.GE.ERRMIN) GO TO 580                            CSL05800
C     LAST COMPONENT IS LESS THAN MINIMAL TO DATE              CSL05810
      ERRMIN = ERROR                                           CSL05820
      MABEST = MA(J)                                           CSL05830
  580 CONTINUE                                                 CSL05840
C                                                              CSL05850
      IF(MA(J).GT.0)  ITEST = MA(J) + IDELTA(J)                CSL05860
      IF(MA(J).LT.0)  ITEST = -(IABS(MA(J)) + IDELTA(J))       CSL05870
      IF(IABS(ITEST).LE.ML(J).AND.ICOUNT.LE.10) GO TO 600      CSL05880
C     NEW MA(J) IS GREATER THAN MAXIMUM ALLOWED.               CSL05890
      IF(ERCONT.EQ.0.OR.MABEST.EQ.MPMIN) GO TO 620             CSL05900
      TFLAG = 1                                                CSL05910
      MA(J) = MABEST                                           CSL05920
      MTOL = MTOL - KMAXU                                      CSL05930
      WRITE(6,590) MA(J)                                       CSL05940
```

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

```
C     THE LOOP ON T-EIGENVECTOR COMPUTATIONS IS COMPLETE.            CSL06500
C     WRITE OUT THE SIZE T-MATRICES THAT WILL BE USED FOR            CSL06510
C     THE RITZ VECTOR COMPUTATIONS.                                  CSL06520
C                                                                    CSL06530
      WRITE(10,730)                                                  CSL06540
  730 FORMAT(/' SIZES OF T-MATRICES THAT WILL BE USED IN THE RITZ COMPUTCSL06550
     1ATIONS'/5X,'J',13X,'REAL(GOODEV)',13X,'IMAG(GOODEV)',1X,'MA(J)') CSL06560
C                                                                    CSL06570
      WRITE(10,740)   (J,GOODEV(J),MA(J), J=1,NGOOD)                 CSL06580
  740 FORMAT(I6,2E25.14,I6)                                          CSL06590
      WRITE(10,490)                                                  CSL06600
C                                                                    CSL06610
      WRITE(6,750) MTOL                                              CSL06620
  750 FORMAT(/' THE CUMULATIVE LENGTH OF THE T-EIGENVECTORS IS',I18) CSL06630
C                                                                    CSL06640
      WRITE(6,760) NTVEC,NGOOD                                       CSL06650
  760 FORMAT(/I6,' T-EIGENVECTORS OUT OF',I6,' REQUESTED WERE COMPUTED')CSL06660
C                                                                    CSL06670
C     SAVE THE T-EIGENVECTORS ON FILE 11?                            CSL06680
      IF(TVSTOP.NE.1.AND.SVTVEC.EQ.0) GO TO 820                      CSL06690
C                                                                    CSL06700
      WRITE(11,770) NTVEC,MTOL,MATNO,SVSEED                          CSL06710
  770 FORMAT(I6,3I12,' = NTVEC,MTOL,MATNO,SVSEED')                   CSL06720
C                                                                    CSL06730
      DO 800 J=1,NGOODC                                              CSL06740
C     IF MP(J) = MPMIN THEN NO SUITABLE T-EIGENVECTOR IS AVAILABLE   CSL06750
C     FOR THAT EIGENVALUE.                                           CSL06760
      IF(MP(J).EQ.MPMIN) WRITE(11,780) J,MA(J),GOODEV(J),MP(J)       CSL06770
  780 FORMAT(2I6,2E20.12,I6/' TH EIGVAL,T-SIZE,EVALUE,FLAG,NO EIGVEC') CSL06780
      IF(MP(J).NE.MPMIN) WRITE(11,790) J,MA(J),GOODEV(J),MP(J)       CSL06790
  790 FORMAT(I6,I6,2E20.12,I6/' T-EIGVEC,SIZE T,EVALUE OF A,MP(J)')   CSL06800
      IF(MP(J).EQ.MPMIN) GO TO 800                                   CSL06810
      KI = MINT(J)                                                   CSL06820
      KF = MFIN(J)                                                   CSL06830
C                                                                    CSL06840
      WRITE(11,260) (TVEC(K), K=KI,KF)                               CSL06850
C                                                                    CSL06860
  800 CONTINUE                                                       CSL06870
C                                                                    CSL06880
      IF(TVSTOP.NE.1) GO TO 820                                      CSL06890
C                                                                    CSL06900
      WRITE(6,810) TVSTOP, NTVEC,NGOOD                               CSL06910
  810 FORMAT(/' USER SET TVSTOP = ',I1/                              CSL06920
     1' THEREFORE PROGRAM TERMINATES AFTER T-EIGENVECTOR COMPUTATIONS'/ CSL06930
     1' T-EIGENVECTORS THAT WERE COMPUTED ARE SAVED ON FILE 11'/     CSL06940
     1I8,' T-EIGENVECTORS WERE COMPUTED OUT OF',I7,' REQUESTED'/)    CSL06950
C                                                                    CSL06960
      GO TO 1550                                                     CSL06970
C                                                                    CSL06980
  820 CONTINUE                                                       CSL06990
C     IF NOT ABLE TO COMPUTE ALL THE REQUESTED T-EIGENVECTORS        CSL07000
C     CONTINUE WITH THE LANCZOS VECTOR COMPUTATIONS ANYWAY?          CSL07010
C                                                                    CSL07020
      IF(NTVEC.NE.NGOOD.AND.LVCONT.EQ.0) GO TO 1450                  CSL07030
C                                                                    CSL07040
```

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

```
C                                                                     CSL07600
C---------------------------------------------------------------------CSL07610
C                                                                     CSL07620
      CALL GENRAN(IIL,G,N)                                            CSL07630
C                                                                     CSL07640
C---------------------------------------------------------------------CSL07650
C                                                                     CSL07660
      DO 900 I = 1,N                                                  CSL07670
  900 GC(I) = G(I)                                                    CSL07680
C                                                                     CSL07690
      DO 910 I = 1,N                                                  CSL07700
  910 V2(I) = DCMPLX(GR(I),GC(I))                                     CSL07710
C                                                                     CSL07720
C---------------------------------------------------------------------CSL07730
      CALL INPRDC(V2,V2,SUMC,N)                                       CSL07740
C---------------------------------------------------------------------CSL07750
C                                                                     CSL07760
      SUMC = ONE/CDSQRT(SUMC)                                         CSL07770
      DO 920 I = 1,N                                                  CSL07780
      V1(I) = ZEROC                                                   CSL07790
  920 V2(I) = V2(I)*SUMC                                              CSL07800
C                                                                     CSL07810
C     LOOP FOR GENERATING REQUIRED RITZ VECTORS (IVEC = 1,KMAXU)      CSL07820
C                                                                     CSL07830
      IVEC = 1                                                        CSL07840
      BATA = ZEROC                                                    CSL07850
C                                                                     CSL07860
      GO TO 980                                                       CSL07870
C                                                                     CSL07880
  930 CONTINUE                                                        CSL07890
C                                                                     CSL07900
C---------------------------------------------------------------------CSL07910
C                                                                     CSL07920
C   CMATV(V2,V1,BATA) CALCULATES  V1 = A*V2 - BATA*V1                 CSL07930
      CALL CMATV(V2,V1,BATA)                                          CSL07940
      CALL INPRDC(V2,V1,ALFA,N)                                       CSL07950
C                                                                     CSL07960
C---------------------------------------------------------------------CSL07970
C                                                                     CSL07980
      DO 940 J=1,N                                                    CSL07990
  940 V1(J) = V1(J)-ALFA*V2(J)                                        CSL08000
C                                                                     CSL08010
C---------------------------------------------------------------------CSL08020
      CALL INPRDC(V1,V1,BATA,N)                                       CSL08030
C---------------------------------------------------------------------CSL08040
C                                                                     CSL08050
      BATA = CDSQRT(BATA)                                             CSL08060
      SUMC = ONE/BATA                                                 CSL08070
C                                                                     CSL08080
      TEMPC = BETA(IVEC)                                              CSL08090
      TEMP = CDABS(BATA - TEMPC)/CDABS(TEMPC)                         CSL08100
      IF (TEMP.LT.1.0D-10)GO TO 960                                   CSL08110
C                                                                     CSL08120
C     IF THE BETA BEING REGENERATED DO NOT MATCH THE HISTORY FILE     CSL08130
C     THEN SOMETHING IS WRONG IN THE LANCZOS VECTOR GENERATION        CSL08140
```

```
C      AND PROGRAM TERMINATES FOR USER TO CORRECT THE PROBLEM        CSL08150
C      WHICH MUST BE IN THE STARTING VECTOR GENERATION OR IN         CSL08160
C      THE MATRIX-VECTOR MULTIPLY SUBROUTINE CMATV SUPPLIED.         CSL08170
C      THIS PART OF THE COMPUTATIONS MUST BE IDENTICAL TO THE        CSL08180
C      CORRESPONDING PART IN THE EIGENVALUE COMPUTATIONS.           CSL08190
C                                                                   CSL08200
       WRITE(6,950) IVEC,BATA,BETA(IVEC),TEMP                       CSL08210
  950 FORMAT(/2X,'IVEC',16X,'BATA',10X,'BETA(IVEC)',14X,'RELDIF'/I6, CSL08220
     13E20.12/' IN LANCZOS VECTOR REGENERATION THE ENTRIES OF THE TRIDIACSL08230
     1GONAL MATRICES BEING'/' GENERATED ARE NOT THE SAME AS THOSE IN THECSL08240
     1 MATRIX SUPPLIED ON FILE 2.'/' THEREFORE SOMETHING IS BEING INITIACSL08250
     1LIZED OR COMPUTED DIFFERENTLY FROM THE WAY'/' IT WAS COMPUTED IN TCSL08260
     1HE EIGENVALUE COMPUTATIONS'/' THE PROGRAM TERMINATES FOR THE USER CSL08270
     1TO DETERMINE WHAT THE PROBLEM IS'/)                          CSL08280
       GO TO 1550                                                   CSL08290
C                                                                   CSL08300
C                                                                   CSL08310
  960 CONTINUE                                                      CSL08320
       DO 970 J = 1,N                                               CSL08330
       TEMPC = SUMC*V1(J)                                           CSL08340
       V1(J) = V2(J)                                                CSL08350
  970 V2(J) = TEMPC                                                 CSL08360
C                                                                   CSL08370
  980 CONTINUE                                                      CSL08380
C                                                                   CSL08390
       LFIN = 0                                                     CSL08400
       DO 1000 J = 1,NGOODC                                         CSL08410
       LL = LFIN                                                    CSL08420
       LFIN = LFIN + N                                              CSL08430
C                                                                   CSL08440
       IF(IABS(MA(J)).LT.IVEC.OR.MP(J).EQ.MPMIN) GO TO 1000         CSL08450
       II = IVEC + MINT(J) - 1                                      CSL08460
       TEMPC = TVEC(II)                                             CSL08470
C      II IS THE (IVEC)TH COMPONENT OF THE T-EIGENVECTOR CONTAINED  CSL08480
C      IN TVEC(MINT(J)).                                            CSL08490
C                                                                   CSL08500
       DO 990 K = 1,N                                               CSL08510
       LL = LL + 1                                                  CSL08520
  990 RITVEC(LL) = TEMPC*V2(K) + RITVEC(LL)                         CSL08530
C                                                                   CSL08540
 1000 CONTINUE                                                      CSL08550
C                                                                   CSL08560
       IVEC = IVEC + 1                                              CSL08570
       IF (IVEC.LE.KMAXU) GO TO 930                                 CSL08580
C                                                                   CSL08590
C                                                                   CSL08600
C      RITZVECTOR GENERATION IS COMPLETE. NORMALIZE EACH RITZVECTOR. CSL08610
C      NOTE THAT IF CERTAIN RITZ VECTORS WERE NOT COMPUTED THEN THE  CSL08620
C      CORRESPONDING PORTION OF THE RITVEC ARRAY WAS NOT UTILIZED.   CSL08630
C                                                                   CSL08640
       LFIN = 0                                                     CSL08650
       DO 1050 J = 1,NGOODC                                         CSL08660
C                                                                   CSL08670
       KK = LFIN                                                    CSL08680
       LFIN = LFIN + N                                              CSL08690
```

```
      IF(MP(J).EQ.MPMIN) GO TO 1050                            CSL08700
C                                                              CSL08710
      DO 1010 K = 1,N                                          CSL08720
      KK = KK + 1                                              CSL08730
 1010 V2(K) = RITVEC(KK)                                       CSL08740
C                                                              CSL08750
C--------------------------------------------------------------CSL08760
      CALL INPRDC(V2,V2,SUMC,N)                                CSL08770
C--------------------------------------------------------------CSL08780
C                                                              CSL08790
      SUMC = CDSQRT(SUMC)                                      CSL08800
      RNORM(J) = CDABS(SUMC)                                   CSL08810
      TEMP = DABS(ONE-RNORM(J))                                CSL08820
      SUMC = DCMPLX(ONE,ZERO)/SUMC                             CSL08830
C                                                              CSL08840
      KK = LFIN - N                                            CSL08850
      DO 1020 K = 1,N                                          CSL08860
      KK = KK + 1                                              CSL08870
      V2(K) = SUMC*V2(K)                                       CSL08880
 1020 RITVEC(KK) = V2(K)                                       CSL08890
C                                                              CSL08900
C     COMPUTE THE 'REAL' NORM                                  CSL08910
C                                                              CSL08920
C--------------------------------------------------------------CSL08930
      CALL CINPRD(V2,V2,SUM,N)                                 CSL08940
C--------------------------------------------------------------CSL08950
C                                                              CSL08960
      IF (IWRITE.NE.0) WRITE(6,1030) J,GOODEV(J)               CSL08970
 1030 FORMAT(/I5,' TH EIGENVALUE CONSIDERED = ',2E20.12/)      CSL08980
C                                                              CSL08990
      IF (IWRITE.NE.0) WRITE(6,1040) TERR(J),TBETA(J),RNORM(J),SUM CSL09000
 1040 FORMAT(' NORM OF ERROR IN T-EIGENVECTOR = ',E14.3/       CSL09010
     1 ' CDABS(BETA(MA(J)+1)*U(MA(J))) ',E14.3/                CSL09020
     1 ' CDABS(EUCLIDEAN-NORM(RITVEC))  = ',E14.3/             CSL09030
     1 ' HERMITIAN-NORM(RITVEC)**2  = ',E14.3/)                CSL09040
C                                                              CSL09050
      LINT = LFIN - N + 1                                      CSL09060
      EVAL = GOODEV(J)                                         CSL09070
C                                                              CSL09080
C--------------------------------------------------------------CSL09090
C                                                              CSL09100
      CALL CMATV(RITVEC(LINT),V2,EVAL)                         CSL09110
C                                                              CSL09120
C--------------------------------------------------------------CSL09130
C                                                              CSL09140
C     COMPUTE ERROR IN RITZ VECTOR CONSIDERED AS A EIGENVECTOR OF A. CSL09150
C     V2 = A*RITVEC - EVAL*RITVEC                              CSL09160
C                                                              CSL09170
C--------------------------------------------------------------CSL09180
      CALL CINPRD(V2,V2,SUM,N)                                 CSL09190
C--------------------------------------------------------------CSL09200
C                                                              CSL09210
      SUM = DSQRT(SUM)                                         CSL09220
      ERR(J) = SUM                                             CSL09230
      GAP = ABS(AMINGP(J))                                     CSL09240
```

```
      ERRDGP(J) = SUM/GAP                                           CSL09250
C                                                                   CSL09260
 1050 CONTINUE                                                      CSL09270
C                                                                   CSL09280
C                                                                   CSL09290
C     RITZVECTORS ARE NORMALIZED AND ERROR ESTIMATES ARE IN ERR ARRAY  CSL09300
C     AND IN ERRDGP ARRAY. STORE EVERYTHING                        CSL09310
C                                                                   CSL09320
C                                                                   CSL09330
      WRITE(9,1060)                                                 CSL09340
 1060 FORMAT(3X,'REAL(GOODEV)',3X,'IMAG(GOODEV)',1X,'MA(J)',7X,'AMINGAP'CSL09350
     1 ,4X,'AERROR',2X,'AERR/GAP',4X,'TERROR')                     CSL09360
C                                                                   CSL09370
      WRITE(13,1070)                                                CSL09380
 1070 FORMAT(8X,'REAL(GOODEV)',8X,'IMAG(GOODEV)',2X,'RITZNORM',3X,'AMINGCSL09390
     1AP',2X,'TBETA(J)',2X,'TLAST(J)')                             CSL09400
C                                                                   CSL09410
      DO 1100 J=1,NGOODC                                            CSL09420
C                                                                   CSL09430
      IF(MP(J).EQ.MPMIN) GO TO 1100                                CSL09440
C                                                                   CSL09450
      WRITE(9,1080)GOODEV(J),MA(J),AMINGP(J),ERR(J),ERRDGP(J),TERR(J)  CSL09460
 1080 FORMAT(2E15.8,I6,E14.6,3E10.3)                               CSL09470
C                                                                   CSL09480
      WRITE(13,1090) GOODEV(J),RNORM(J),AMINGP(J),TBETA(J),TLAST(J)  CSL09490
 1090 FORMAT(2E20.12,4E10.3)                                       CSL09500
C                                                                   CSL09510
 1100 CONTINUE                                                      CSL09520
C                                                                   CSL09530
      IF(MREJEC.EQ.0) GO TO 1180                                   CSL09540
      WRITE(9,1110)                                                 CSL09550
 1110 FORMAT(/' RITZ VECTORS WERE NOT COMPUTED FOR THE FOLLOWING EIGENVACSL09560
     1LUES'/' EITHER BECAUSE THEY HAD NOT CONVERGED OR BECAUSE THE ERRORCSL09570
     1 ESTIMATE'/'  WAS NOT AS SMALL AS DESIRED'/)                 CSL09580
C                                                                   CSL09590
      WRITE(13,1120)                                                CSL09600
 1120 FORMAT(/' RITZ VECTORS WERE NOT COMPUTED FOR THE FOLLOWING EIGENVACSL09610
     1LUES'/' EITHER BECAUSE THEY HAD NOT CONVERGED OR BECAUSE'/' THE ERCSL09620
     1ROR ESTIMATE WAS NOT AS SMALL AS DESIRED'/)                  CSL09630
C                                                                   CSL09640
      DO 1170 J = 1,NGOODC                                         CSL09650
      IF(MP(J).NE.MPMIN) GO TO 1170                                CSL09660
C     WRITE OUT MESSAGE FOR EACH EIGENVALUE FOR WHICH NO EIGENVECTOR  CSL09670
C     WAS COMPUTED.                                                 CSL09680
C                                                                   CSL09690
      WRITE(9,1130)                                                 CSL09700
 1130 FORMAT(6X,'GOODEV(J)',3X,'MA(J)',5X,'AMINGP(J)',6X,'TLAST(J)',3X, CSL09710
     1'MP(J)')                                                     CSL09720
      WRITE(9,1140) GOODEV(J),MA(J),AMINGP(J),TBETA(J),MP(J)       CSL09730
 1140 FORMAT(2E15.8,I8,2E14.4,I8)                                  CSL09740
C                                                                   CSL09750
      WRITE(13,1150)                                                CSL09760
 1150 FORMAT(6X,'REAL(GOODEV(J))',6X,'IMAG(GOODEV(J))',4X,'MA(J)',3X, CSL09770
     1'MP(J)')                                                     CSL09780
      WRITE(13,1160) GOODEV(J),MA(J),MP(J)                         CSL09790
```

```
 1160 FORMAT(2E15.8,2I8)                                          CSL09800
C                                                                 CSL09810
 1170 CONTINUE                                                    CSL09820
 1180 CONTINUE                                                    CSL09830
C                                                                 CSL09840
      WRITE(9,1190)                                               CSL09850
 1190 FORMAT(/' ABOVE ARE ERROR ESTIMATES FOR THE A AND T EIGENVECTORS'/CSL09860
     1 ' ASSOCIATED WITH THE GOODEV LISTED IN COLUMN 1'/          CSL09870
     1 ' AERROR = NORM(A*X - EV*X)   TERROR = NORM(T*Y - EV*Y) '/ CSL09880
     1 ' WHERE T = T(1,MA(J))    X = RITZ VECTOR = V*Y  V = SUCCESSIVE'/CSL09890
     1 ' LANCZOS VECTORS. A MINGAP = GAP TO NEAREST A-EIGENVALUE'//) CSL09900
C                                                                 CSL09910
      WRITE(13,1200)                                              CSL09920
 1200 FORMAT(/' ABOVE ARE ERROR ESTIMATES FOR THE A AND T EIGENVECTORS'/CSL09930
     1 ' ASSOCIATED WITH THE GOODEV LISTED IN COLUMN 1'/          CSL09940
     1 ' AERROR = NORM(A*X-EV*X)   TERROR = NORM(T*Y-EV*Y) WHERE' CSL09950
     1 /' T = T(1,MA(J))     X = RITZ VECTOR = V*Y  V = SUCCESSIVE '/ CSL09960
     1 ' LANCZOS VECTORS. A MINGAP = GAP TO NEAREST A-EIGENVALUE'/ CSL09970
     1 ' AERROR AND TERROR ARE GIVEN IN FILE 9. RNORM = NORM(X)'/ CSL09980
     1 ' BETA(M+1)*ABS(Y(M)) IS AN ESTIMATOR OF NORM(A*X-EV*X)'//) CSL09990
C                                                                 CSL10000
C     NUMBER OF RITZ VECTORS COMPUTED                             CSL10010
      NCOMPU = NGOODC - MREJEC                                    CSL10020
      WRITE(12,1210) N,NCOMPU,NGOODC,MATNO                        CSL10030
 1210 FORMAT(3I6,I12,' SIZE A, NO.RITZVECS, NO.EVALUES,MATNO')    CSL10040
C                                                                 CSL10050
      LFIN = 0                                                    CSL10060
      DO 1270 J = 1,NGOODC                                        CSL10070
      LINT = LFIN + 1                                             CSL10080
      LFIN = LFIN + N                                             CSL10090
C                                                                 CSL10100
      IF(MP(J).EQ.MPMIN) GO TO 1250                               CSL10110
C     RITZ VECTOR WAS COMPUTED                                    CSL10120
      WRITE(12,1220) J, GOODEV(J), MP(J)                          CSL10130
 1220 FORMAT(I6,4X,2E20.12,I6,' J, EIGENVAL, MP(J)')              CSL10140
C                                                                 CSL10150
      WRITE(12,1230) ERR(J),ERRDGP(J)                            CSL10160
 1230 FORMAT(2E15.5,' = NORM(A*Z-EVAL*Z) AND  NORM(A*Z-EVAL*Z)/MINGAP') CSL10170
C                                                                 CSL10180
      WRITE(12,1240) (RITVEC(LL), LL=LINT,LFIN)                   CSL10190
C1240 FORMAT(4Z20)                                                CSL10200
 1240 FORMAT(2(2E20.12))                                          CSL10210
      GO TO 1270                                                  CSL10220
C     NO RITZ VECTOR WAS COMPUTED FOR THIS EIGENVALUE            CSL10230
 1250 WRITE(12,1260) J,GOODEV(J),MP(J)                            CSL10240
 1260 FORMAT(I6,4X,E20.12,I6,' J,EIGVALUE,NO RITZ VECTOR COMPUTED') CSL10250
C                                                                 CSL10260
 1270 CONTINUE                                                    CSL10270
C                                                                 CSL10280
C     DID ANY T-MATRICES INCLUDE OFF-DIAGONAL ENTRIES SMALLER THAN CSL10290
C     DESIRED, AS SPECIFIED BY BTOL?                              CSL10300
C                                                                 CSL10310
      IF(IB.GT.0) GO TO 1300                                      CSL10320
      WRITE(6,1280) KMAXU                                         CSL10330
 1280 FORMAT(/' FOR LARGEST T-MATRIX CONSIDERED',I7,' CHECK THE SIZE OF CSL10340
```

```
     1BETAS')                                                  CSL10350
C                                                              CSL10360
C-------------------------------------------------------------CSL10370
C                                                              CSL10380
      CALL TNORM(ALPHA,BETA,BKMIN,TEMP,KMAXU,IBMT)             CSL10390
C                                                              CSL10400
C-------------------------------------------------------------CSL10410
C                                                              CSL10420
      IF(IBMT.LT.0) WRITE (6,1290)                             CSL10430
 1290 FORMAT(/' WARNING THE T-MATRICES FOR ONE OR MORE OF THE EIGENVALUECSL10440
     1S CONSIDERED'/' HAD AN OFF-DIAGONAL ENTRY THAT WAS SMALLER THAN THCSL10450
     1E BETA TOLERANCE THAT WAS SPECIFIED'/)                   CSL10460
 1300 CONTINUE                                                 CSL10470
C                                                              CSL10480
      GO TO 1550                                               CSL10490
C                                                              CSL10500
 1310 WRITE(6,1320) NGOOD,NMAX,MDIMRV                          CSL10510
 1320 FORMAT(/I4,' RITZ VECTORS WERE REQUESTED BUT THE REQUIRED DIMENSIOCSL10520
     1N',I6/' IS LARGER THAN THE USER-SPECIFIED DIMENSION OF RITVEC',I6 CSL10530
     1/' THEREFORE, THE EIGENVECTOR PROCEDURE TERMINATES FOR THE USER TOCSL10540
     1 INTERVENE')                                             CSL10550
C                                                              CSL10560
      GO TO 1550                                               CSL10570
C                                                              CSL10580
 1330 WRITE(6,1340) NOLD,N,MATOLD,MATNO                        CSL10590
 1340 FORMAT(/' PARAMETERS READ FROM FILE 3 DO NOT AGREE WITH USER-SPECICSL10600
     1FIED'/' PARAMETERS, NOLD,N,MATOLD,MATNO = '/2I6,2I12/    CSL10610
     1' THEREFORE PROGRAM TERMINATES FOR USER TO RESOLVE DIFFERENCES'/) CSL10620
C                                                              CSL10630
      GO TO 1550                                               CSL10640
C                                                              CSL10650
 1350 WRITE(6,1360)                                            CSL10660
 1360 FORMAT(/' PARAMETERS IN ALPHA,BETA FILE READ IN DO NOT AGREE WITH CSL10670
     1 THOSE'/' SPECIFIED BY THE USER.  THEREFORE, THE PROGRAM TERMINATECSL10680
     1S FOR'/' THE USER TO RESOLVE THE DIFFERENCES'/)          CSL10690
C                                                              CSL10700
      GO TO 1550                                               CSL10710
C                                                              CSL10720
 1370 WRITE(6,1380) KMAX,MEV                                   CSL10730
 1380 FORMAT(/' IN ALPHA, BETA HISTORY HEADER KMAX =',I6/      CSL10740
     1' BUT EIGENVALUES WERE COMPUTED AT MEV = ',I6,' PROGRAM STOPS'/)  CSL10750
C                                                              CSL10760
      GO TO 1550                                               CSL10770
C                                                              CSL10780
 1390 WRITE(6,1400)                                            CSL10790
 1400 FORMAT(/' PROGRAM COMPUTED 1ST GUESSES AT T-MATRIX SIZES'/' READ TCSL10800
     1HEM TO FILE 10, THEN TERMINATED AS REQUESTED.')          CSL10810
      GO TO 1550                                               CSL10820
C                                                              CSL10830
 1410 WRITE(6,1420) MTOL, MDIMTV                               CSL10840
 1420 FORMAT(/' PROGRAM TERMINATES BECAUSE THE MINIMAL TVEC DIMENSION ANCSL10850
     1TICIPATED',I7/'  IS LARGER THAN THE TVEC DIMENSION',I7,' SPECIFIEDCSL10860
     1 BY THE USER.'/'  USER MAY RESET THE TVEC DIMENSION AND RESTART THCSL10870
     1E PROGRAM')                                              CSL10880
      GO TO 1550                                               CSL10890
```

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

7.5 CSLEMULT: LANCZS and Sample Matrix-Vector Multiply Subroutines

```
C-----CSLEMULT--(COMPLEX SYMMETRIC MATRICES)--------------------------CSL00010
C  Authors:  Jane Cullum and Ralph A. Willoughby (Deceased)          CSL00020
C            Los Alamos National Laboratory                          CSL00030
C            Los Alamos, New Mexico 87544                            CSL00040
C                                                                    CSL00050
C            E-mail:  cullumj@lanl.gov                               CSL00060
C                                                                    CSL00070
C  These codes are copyrighted by the authors.  These codes         CSL00080
C  and modifications of them or portions of them are NOT to be       CSL00090
C  incorporated into any commercial codes or used for any other      CSL00100
C  commercial purposes such as consulting for other companies,       CSL00110
C  without legal agreements with the authors of these Codes.         CSL00120
C  If these Codes or portions of them                                CSL00130
C  are used in other scientific or engineering research works        CSL00140
C  the names of the authors of these codes and appropriate           CSL00150
C  references to their written work are to be incorporated in the     CSL00160
C  derivative works.                                                 CSL00170
C                                                                    CSL00180
C  This header is not to be removed from these codes.                CSL00190
C                                                                    CSL00191
C          REFERENCE:  Cullum and Willoughby, Chapter 6,            CSL00192
C          Lanczos Algorithms for Large Symmetric Eigenvalue ComputationsCSL00193
C          VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in CSL00194
C          Applied Mathematics, 2002. SIAM Publications,             CSL00195
C          Philadelphia, PA. USA                                     CSL00196
C                                                                    CSL00197
C                                                                    CSL00198
C                                                                    CSL00200
C     CONTAINS SUBROUTINE LANCZS USED IN THE COMPLEX SYMMETRIC       CSL00210
C     VERSION OF THE LANCZOS PROCEDURES PLUS SAMPLE USPEC AND        CSL00220
C     CMATV SUBROUTINES.                                             CSL00230
C                                                                    CSL00240
C     PORTABILITY:                                                   CSL00250
C     THESE PROGRAMS ARE NOT PORTABLE DUE TO THE USE OF COMPLEX*16   CSL00260
C     VARIABLES AND CORRESPONDING FUNCTIONS.  MOREOVER, THE PFORT    CSL00270
C     VERIFIER IDENTIFIED THE FOLLOWING ADDITIONAL NONPORTABLE       CSL00280
C     CONSTRUCTIONS:                                                 CSL00290
C                                                                    CSL00300
C     1.  ENTRIES USED TO PASS THE STORAGE LOCATIONS OF THE          CSL00310
C         ARRAYS AND PARAMETERS NEEDED TO SPECIFY THE GIVEN MATRIX   CSL00320
C         FROM THE USPEC SUBROUTINE TO THE MATRIX-VECTOR MULTIPLY    CSL00330
C         SUBROUTINE CMATV.                                          CSL00340
C     2.  IN THE SAMPLE USPEC SUBROUTINES PROVIDED: THE FREE FORMAT  CSL00350
C         READ(8,*) AND THE FORMAT (20A4).  IN THE SAMPLE CMATV:     CSL00360
C         THE COMPUTATION OF INDICES:  IN THE AUXILIARY SUBROUTINE   CSL00370
C         USED FOR COMPUTING THE KNOWN EIGENVALUES OF TEST CLASS 2   CSL00380
C         MATRICES, THE DATA/MACHEP DEFINITION.                      CSL00390
C                                                                    CSL00400
C-----LANCZS-COMPUTE THE LANCZOS TRIDIAGONAL MATRICES----------------CSL00410
C                                                                    CSL00420
```

```
      SUBROUTINE LANCZS(MATVEC,V1,V2,ALPHA,BETA,                CSL00430
     1GR,GC,G,KMAX,MOLD1,N,IIX)                                 CSL00440
C                                                               CSL00450
C-------------------------------------------------------------CSL00460
      COMPLEX*16 V1(1), V2(1), BATA, ZEROC, TEMP, SUMC         CSL00470
      COMPLEX*16  ALPHA(1), BETA(1)                            CSL00480
      DOUBLE PRECISION SUM, ONE, ZERO, GR(1), GC(1)            CSL00490
      REAL  G(1)                                               CSL00500
      EXTERNAL MATVEC                                          CSL00510
C     COMPLEX*16  CDSQRT, DCMPLX                               CSL00520
C-------------------------------------------------------------CSL00530
C                                                               CSL00540
      ZERO = 0.D0                                              CSL00550
      ONE = 1.D0                                               CSL00560
      ZEROC = DCMPLX(ZERO,ZERO)                                CSL00570
C                                                               CSL00580
      IF(MOLD1.GT.1)GO TO 50                                   CSL00590
C                                                               CSL00600
C   ALPHA/BETA GENERATION STARTS AT I = 1                      CSL00610
C   MOLD1 = 1 SET V1 = 0. AND V2 = RANDOM UNIT VECTOR          CSL00620
      IIL=IIX                                                  CSL00630
C                                                               CSL00640
C-------------------------------------------------------------CSL00650
      CALL GENRAN(IIL,G,N)                                     CSL00660
C-------------------------------------------------------------CSL00670
C                                                               CSL00680
      DO 10 I = 1,N                                            CSL00690
   10 GR(I) = G(I)                                             CSL00700
C                                                               CSL00710
C-------------------------------------------------------------CSL00720
      CALL GENRAN(IIL,G,N)                                     CSL00730
C-------------------------------------------------------------CSL00740
C                                                               CSL00750
      DO 20 I = 1,N                                            CSL00760
   20 GC(I) = G(I)                                             CSL00770
C                                                               CSL00780
      DO 30 I = 1,N                                            CSL00790
   30 V2(I) = DCMPLX(GR(I),GC(I))                              CSL00800
C                                                               CSL00810
C-------------------------------------------------------------CSL00820
      CALL INPRDC(V2,V2,SUMC,N)                                CSL00830
C-------------------------------------------------------------CSL00840
C                                                               CSL00850
      SUMC = ONE/CDSQRT(SUMC)                                  CSL00860
      DO 40 I = 1,N                                            CSL00870
      V1(I) = ZEROC                                            CSL00880
   40 V2(I) = V2(I)*SUMC                                       CSL00890
      BETA(1) = ZEROC                                          CSL00900
C                                                               CSL00910
C   ALPHA BETA GENERATION LOOP                                 CSL00920
   50 CONTINUE                                                 CSL00930
C                                                               CSL00940
      DO 80 I=MOLD1,KMAX                                       CSL00950
      SUMC = BETA(I)                                           CSL00960
C                                                               CSL00970
```

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

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

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

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

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

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

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

```
C     MINGAP CALCULATION FOR DISTINCT A-EIGENVALUES                 CSL04830
      NM1 = NEXACT - 1                                              CSL04840
      G(NEXACT) = U(NM1)-U(NEXACT)                                  CSL04850
      G(1) = U(2)-U(1)                                              CSL04860
C                                                                   CSL04870
      DO 110 J = 2,NM1                                              CSL04880
      T0 = U(J)-U(J-1)                                              CSL04890
      T1 = U(J+1)-U(J)                                              CSL04900
      G(J) = T1                                                     CSL04910
      IF (T0.LT.T1) G(J) = -T0                                      CSL04920
  110 CONTINUE                                                      CSL04930
C                                                                   CSL04940
C     NEXACT DISTINCT A-EIGENVALUES ARE IN U IN ASCENDING ORDER     CSL04950
C     MP = MULTIPLICITIES OF THE DISTINCT EIGENVALUES OF A          CSL04960
C     G = TRUE MINIMUM GAP IN A FOR EACH OF THESE EIGENVALUES       CSL04970
C     G < 0 INDICATES THE LEFT-HAND GAP WAS MINIMAL.                CSL04980
C     OUTPUT MULTIPLICITIES, DISTINCT EVS, AND MINGAPS TO FILE 9    CSL04990
C                                                                   CSL05000
      WRITE(9,120)                                                  CSL05010
  120 FORMAT(5X,'I',1X,'AMULT',5X,'TRUE A-EIGENVALUE(I)',           CSL05020
     1 3X,'A-MINGAP(I)')                                            CSL05030
C                                                                   CSL05040
      WRITE(9,130)(J,MP(J),U(J),G(J), J=1,NEXACT)                   CSL05050
  130 FORMAT(2I6,E25.16,E14.3)                                      CSL05060
C                                                                   CSL05070
      WRITE(9,140)                                                  CSL05080
  140 FORMAT(' NEXACT DISTINCT A-EIGENVALUES ARE IN ASCENDING ORDER'/ CSL05090
     1 ' AMULT = MULTIPLICITIES OF THE DISTINCT EIGENVALUES OF A.'/ CSL05100
     2 ' A-MINGAP(I) = TRUE MINIMUM GAP IN A FOR EACH EIGENVALUE.'/ CSL05110
     3 ' A-MINGAP(I).LT.0 INDICATES THE LEFT-HAND GAP WAS MINIMAL.'//) CSL05120
C                                                                   CSL05130
C     WE ORDER U VECTOR BY INCREASING SIZE OF THE GAPS              CSL05140
C                                                                   CSL05150
      DO 150 K = 1,N                                                CSL05160
  150 MP(K) = K                                                     CSL05170
C                                                                   CSL05180
      DO 170 K = 2,N                                                CSL05190
      KM1 = K-1                                                     CSL05200
C                                                                   CSL05210
      DO 160 L = 1,KM1                                              CSL05220
      JJ = K - L                                                    CSL05230
      IF (ABS(G(JJ+1)).GE.ABS(G(JJ))) GO TO 170                     CSL05240
      EE = U(JJ)                                                    CSL05250
      U(JJ) = U(JJ+1)                                               CSL05260
      U(JJ+1) = EE                                                  CSL05270
      GG = G(JJ)                                                    CSL05280
      G(JJ) = G(JJ+1)                                               CSL05290
      G(JJ+1) = GG                                                  CSL05300
      IEE = MP(JJ)                                                  CSL05310
      MP(JJ) = MP(JJ+1)                                             CSL05320
  160 MP(JJ+1) = IEE                                                CSL05330
C                                                                   CSL05340
  170 CONTINUE                                                      CSL05350
C                                                                   CSL05360
      WRITE(9,180)                                                  CSL05370
```

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

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

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

```
C                                                               CSL00440
C-------------------------------------------------------------CSL00450
      COMPLEX*16  ALPHA(1),BETA(1),V1(1),V2(1),VS(1)           CSL00460
      COMPLEX*16  U,Z,X1,RATIO,BETAM,TEMP,ZEROC                CSL00470
      DOUBLE PRECISION EST,ESTR,ESTC,SUM,XU,NORM,TSUM,GSUM     CSL00480
      DOUBLE PRECISION  EPS,EPS3,EPS4,ZERO,ONE,GR(1),GC(1),GAP CSL00490
      REAL G(1),GG(1)                                          CSL00500
      INTEGER  MP(1), INTERC(1)                                CSL00510
      REAL  ABS                                                CSL00520
      DOUBLE PRECISION  DABS, DMIN1, DSQRT, DFLOAT, CDABS, DIMAG, DREAL CSL00530
C     COMPLEX*16 DCMPLX                                        CSL00540
C-------------------------------------------------------------CSL00550
C                                                               CSL00560
C     COMPUTES ERROR ESTIMATES FOR COMPUTED ISOLATED GOOD T-EIGENVALUES CSL00570
C     IN VS AND WRITES THESE EIGENVALUES AND ESTIMATES TO FILE 4. CSL00580
C     BY DEFINITION A GOOD T-EIGENVALUE IS ISOLATED IF ITS CLOSEST CSL00590
C     NEIGHBOR IS ALSO GOOD, OR IF ITS CLOSEST NEIGHBOR IS    CSL00600
C     SPURIOUS BUT THAT NEIGHBOR IS FAR ENOUGH AWAY.  SO      CSL00610
C     IN PARTICULAR, WE WILL COMPUTE ESTIMATES FOR ANY GOOD   CSL00620
C     T-EIGENVALUE THAT IS IN A CLUSTER OF GOOD T-EIGENVALUES. CSL00630
C                                                               CSL00640
C     USES INVERSE ITERATION ON T(1,MEV) SOLVING THE EQUATION CSL00650
C     (T - X1*I)V2 = RIGHT-HAND SIDE (RANDOMLY-GENERATED)     CSL00660
C     FOR EACH SUCH GOOD T-EIGENVALUE X1.                     CSL00670
C                                                               CSL00680
C     PROGRAM REFACTORS T-X1*I ON EACH ITERATION OF INVERSE ITERATION. CSL00690
C     TYPICALLY ONLY ONE ITERATION IS NEEDED PER T-EIGENVALUE X1. CSL00700
C                                                               CSL00710
C     ON ENTRY AND EXIT                                       CSL00720
C     MEV = ORDER OF T :  N = ORDER OF ORIGINAL MATRIX A      CSL00730
C     ALPHA, BETA CONTAIN THE NONZERO ENTRIES OF THE T-MATRIX CSL00740
C     VS = COMPUTED DISTINCT EIGENVALUES OF T(1,MEV)          CSL00750
C     MP = T-MULTIPLICITY OF EACH T-EIGENVALUE IN VS. MP(I) = -1 MEANS CSL00760
C          VS(I) IS A GOOD T-EIGENVALUE BUT THAT IT IS SITTING CLOSE TO CSL00770
C          A SPURIOUS T-EIGENVALUE.  MP(I) = 0 MEANS VS(I) IS SPURIOUS. CSL00780
C          ESTIMATES ARE COMPUTED ONLY FOR THOSE T-EIGENVALUES CSL00790
C          WITH MP(I) = 1. FLAGGING WAS DONE IN SUBROUTINE ISOEV CSL00800
C          PRIOR TO ENTERING INVERR.                          CSL00810
C     NISO = NUMBER OF ISOLATED GOOD T-EIGENVALUES CONTAINED IN VS CSL00820
C     NDIS =  NUMBER OF DISTINCT T-EIGENVALUES IN VS          CSL00830
C     IKL = SEED FOR RANDOM NUMBER GENERATOR                  CSL00840
C     EPS = 2. * MACHINE EPSILON                              CSL00850
C                                                               CSL00860
C     IN PROGRAM:                                             CSL00870
C     ITER = MAXIMUM NUMBER OF INVERSE ITERATION STEPS ALLOWED FOR EACH CSL00880
C          X1.   ITER = IT ON ENTRY.                          CSL00890
C     GR,GC = ARRAYS OF DIMENSION AT LEAST MEV + NISO.  USED TO STORE CSL00900
C        RANDOMLY-GENERATED RIGHT-HAND SIDE.  THIS IS NOT     CSL00910
C        REGENERATED FOR EACH X1. G IS ALSO USED TO STORE ERROR CSL00920
C        ESTIMATES AS THEY ARE COMPUTED FOR LATER PRINTOUT.   CSL00930
C     V1,V2 = WORK SPACES USED IN THE FACTORIZATION OF T(1,MEV). CSL00940
C     AT THE END OF THE INVERSE ITERATION COMPUTATION FOR X1, V2 CSL00950
C     CONTAINS THE UNIT EIGENVECTOR OF T(1,MEV) CORRESPONDING TO X1. CSL00960
C     V1 AND V2 MUST BE OF DIMENSION AT LEAST MEV.            CSL00970
C                                                               CSL00980
```

```
C     ON EXIT                                                   CSL00990
C     GG(J) = MINIMUM GAP IN T(1,MEV) FOR EACH VS(J), J=1,NDIS  CSL01000
C     G(I) = |BETAM|*|V2(MEV)| = ERROR ESTIMATE FOR ISOLATED GOOD  CSL01010
C             T-EIGENVALUES, WHERE I = 1,NISO  AND  BETAM = BETA(MEV+1)CSL01020
C             T(1,MEV) CORRESPONDING TO ITH ISOLATED GOOD T-EIGENVALUE.CSL01030
C                                                               CSL01040
C     IF FOR SOME X1 IT.GT.ITER THEN THE ERROR ESTIMATE IN G IS MARKED  CSL01050
C     WITH A - SIGN.                                            CSL01060
C                                                               CSL01070
C     V2 = ISOLATED GOOD T-EIGENVALUES                          CSL01080
C     V1 = MINIMAL T-GAPS FOR THE T-EIGENVALUES IN V2.          CSL01090
C     THESE ARE CONSTRUCTED FOR WRITE-OUT PURPOSES ONLY AND NOT CSL01100
C     NEEDED ELSEWHERE IN THE PROGRAM.                          CSL01110
C-----------------------------------------------------------------CSL01120
C                                                               CSL01130
C     LABEL OUTPUT FILE  4                                      CSL01140
      IF (MMB.EQ.1) WRITE(4,10)                                 CSL01150
   10 FORMAT(' INVERSE ITERATION ERROR ESTIMATES'/)             CSL01160
C                                                               CSL01170
C     FILE 6 (TERMINAL) OUTPUT OF ERROR ESTIMATES               CSL01180
      IF (IWRITE.NE.0.AND.NISO.NE.0) WRITE(6,20)                CSL01190
   20 FORMAT(/' INVERSE ITERATION ERROR ESTIMATES'/'  JISO',' JDIST',8X CSL01200
     1,'GOOD T-EIGENVALUE',4X,'BETAM*UM',5X,'TMINGAP')          CSL01210
C                                                               CSL01220
C     INITIALIZATION AND PARAMETER SPECIFICATION                CSL01230
      ZERO = 0.0D0                                              CSL01240
      ONE = 1.0D0                                               CSL01250
      ZEROC = DCMPLX(ZERO,ZERO)                                 CSL01260
      NG = 0                                                    CSL01270
      NISO = 0                                                  CSL01280
      ITER = IT                                                 CSL01290
      MP1 = MEV+1                                               CSL01300
      MM1 = MEV-1                                               CSL01310
      BETAM = BETA(MP1)                                         CSL01320
      BETA(MP1) = ZEROC                                         CSL01330
C                                                               CSL01340
C     CALCULATE SCALE AND TOLERANCES                            CSL01350
      TSUM = CDABS(ALPHA(1))                                    CSL01360
      DO 30 I = 2,MEV                                           CSL01370
   30 TSUM = TSUM + CDABS(ALPHA(I)) + CDABS(BETA(I))            CSL01380
C                                                               CSL01390
      EPS3 = EPS*TSUM                                           CSL01400
      EPS4 = DFLOAT(MEV)*EPS3                                   CSL01410
C                                                               CSL01420
C     GENERATE SCALED RANDOM RIGHT-HAND SIDE                    CSL01430
      ILL = IKL                                                 CSL01440
C                                                               CSL01450
C-----------------------------------------------------------------CSL01460
      CALL GENRAN(ILL,G,MEV)                                    CSL01470
C-----------------------------------------------------------------CSL01480
C                                                               CSL01490
      DO 40 I = 1,MEV                                           CSL01500
   40 GR(I) = G(I)                                              CSL01510
C                                                               CSL01520
C-----------------------------------------------------------------CSL01530
```

```
      CALL GENRAN(ILL,G,MEV)                                   CSL01540
C-------------------------------------------------------------CSL01550
C                                                             CSL01560
      DO 50 I = 1,MEV                                          CSL01570
   50 GC(I) = G(I)                                             CSL01580
C                                                             CSL01590
      GSUM = ZERO                                              CSL01600
      DO 60 I = 1,MEV                                          CSL01610
   60 GSUM = GSUM + DABS(GR(I)) + DABS(GC(I))                  CSL01620
      GSUM = EPS4/GSUM                                         CSL01630
C                                                             CSL01640
      DO 70 I = 1,MEV                                          CSL01650
      GR(I) = GSUM*GR(I)                                       CSL01660
   70 GC(I) = GSUM*GC(I)                                       CSL01670
C                                                             CSL01680
C     LOOP ON ISOLATED GOOD T-EIGENVALUES IN VS (MP(I) = 1) TO CSL01690
C     CALCULATE CORRESPONDING UNIT EIGENVECTOR OF T(1,MEV)    CSL01700
C                                                             CSL01710
      DO 200 JEV = 1,NDIS                                      CSL01720
      IF (MP(JEV).EQ.0) GO TO 200                              CSL01730
      NG = NG + 1                                              CSL01740
      IF (MP(JEV).NE.1) GO TO 200                              CSL01750
      IT = 1                                                   CSL01760
      NISO = NISO + 1                                          CSL01770
      X1 = VS(JEV)                                             CSL01780
C                                                             CSL01790
C     INITIALIZE RIGHT HAND SIDE FOR INVERSE ITERATION        CSL01800
C     AND THE FLAG ON WHICH ROWS ARE INTERCHANGED             CSL01810
      DO 80 I = 1,MEV                                          CSL01820
      INTERC(I) = 0                                            CSL01830
   80 V2(I) = DCMPLX(GR(I),GC(I))                              CSL01840
C                                                             CSL01850
C     TRIANGULAR FACTORIZATION WITH NEAREST NEIGHBOR PIVOT     CSL01860
C     STRATEGY. INTERCHANGES ARE LABELLED BY SETTING INTERC = 1. CSL01870
C                                                             CSL01880
   90 CONTINUE                                                 CSL01890
      U = ALPHA(1)-X1                                          CSL01900
      Z = BETA(2)                                              CSL01910
C                                                             CSL01920
      DO 110 I = 2,MEV                                         CSL01930
      IF (CDABS(BETA(I)).GT.CDABS(U)) GO TO 100                CSL01940
C     NO INTERCHANGE                                           CSL01950
      V1(I-1) = Z/U                                            CSL01960
      V2(I-1) = V2(I-1)/U                                      CSL01970
      V2(I) = V2(I)-BETA(I)*V2(I-1)                            CSL01980
      RATIO = BETA(I)/U                                        CSL01990
      U = ALPHA(I)-X1-Z*RATIO                                  CSL02000
      Z = BETA(I+1)                                            CSL02010
      GO TO 110                                                CSL02020
  100 CONTINUE                                                 CSL02030
C     INTERCHANGE CASE                                         CSL02040
      RATIO = U/BETA(I)                                        CSL02050
      INTERC(I) = 1                                            CSL02060
      V1(I-1) = ALPHA(I)-X1                                    CSL02070
      U = Z-RATIO*V1(I-1)                                      CSL02080
```

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

```
  170 V2(II) = SUM*V2(II)                                         CSL02640
C                                                                 CSL02650
C     SAVE ERROR ESTIMATE FOR LATER OUTPUT                        CSL02660
      EST = CDABS(BETAM)*CDABS(V2(MEV))                           CSL02670
      ESTR = DABS(DREAL(V2(MEV)))                                 CSL02680
      ESTC = DABS(DIMAG(V2(MEV)))                                 CSL02690
      GSUM = CDABS(BETAM)                                         CSL02700
      IF (IT.GT.ITER) EST = -EST                                  CSL02710
      G(NISO) = EST                                               CSL02720
      IF (IWRITE.EQ.0) GO TO 200                                  CSL02730
C                                                                 CSL02740
C     FILE 6 (TERMINAL) OUTPUT OF ERROR ESTIMATES.                CSL02750
      GAP = GG(JEV)                                               CSL02760
      WRITE(6,180) NISO,JEV,X1,EST,GAP                            CSL02770
  180 FORMAT(2I6,2E20.12,2E12.3)                                  CSL02780
      WRITE(6,190) JEV, X1, EST,ESTR,ESTC                         CSL02790
  190 FORMAT(I6,2E20.12,3E11.3)                                   CSL02800
C                                                                 CSL02810
  200 CONTINUE                                                    CSL02820
C                                                                 CSL02830
C     END ERROR ESTIMATE LOOP ON ISOLATED GOOD T-EIGENVALUES.     CSL02840
C     GENERATE DISTINCT MINGAPS FOR T(1,MEV).  THIS IS USEFUL AS AN CSL02850
C     INDICATOR OF THE GOODNESS OF THE INVERSE ITERATION ESTIMATES. CSL02860
C     TRANSFER ISOLATED GOOD T-EIGENVALUES AND CORRESPONDING TMINGAPS CSL02870
C     TO V2 AND V1 FOR OUTPUT PURPOSES ONLY.                      CSL02880
C                                                                 CSL02890
      ISO = 0                                                     CSL02900
      DO 210 J = 1,NDIS                                           CSL02910
      IF (MP(J).NE.1) GO TO 210                                   CSL02920
      ISO = ISO+1                                                 CSL02930
      GR(ISO) = GG(J)                                             CSL02940
      V2(ISO) = VS(J)                                             CSL02950
  210 CONTINUE                                                    CSL02960
      IF(NISO.EQ.0) GO TO 270                                     CSL02970
C                                                                 CSL02980
C     ERROR ESTIMATES ARE WRITTEN TO FILE 4                       CSL02990
      WRITE(4,220)MEV,NDIS,NG,NISO,N,IKL,ITER,GSUM                CSL03000
  220 FORMAT(1X,'TSIZE',2X,'NDIS',1X,'NGOOD',2X,'NISO',1X,'ASIZE'/5I6/ CSL03010
     1 4X,'RHSEED',2X,'MXINIT',5X,'BETAM'/I10,I8,E10.3)           CSL03020
C                                                                 CSL03030
      WRITE(4,230)                                                CSL03040
  230 FORMAT(2X,'GOODEVNO',11X,'R(GOODEV)',11X,'I(GOODEV)',       CSL03050
     1 6X,'BETAM*UM',7X,'TMINGAP')                                CSL03060
C                                                                 CSL03070
      ISPUR = 0                                                   CSL03080
      I = 0                                                       CSL03090
      DO 260 J = 1,NDIS                                           CSL03100
      IF(MP(J).NE.0) GO TO 240                                    CSL03110
      ISPUR = ISPUR + 1                                           CSL03120
      GO TO 260                                                   CSL03130
  240 IF(MP(J).NE.1) GO TO 260                                    CSL03140
      I = I + 1                                                   CSL03150
      IGOOD = J - ISPUR                                           CSL03160
      WRITE(4,250) IGOOD,V2(I),G(I),GR(I)                         CSL03170
  250 FORMAT(I10,2E20.12,2E14.3)                                  CSL03180
```

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

```
      IF (BSIZE.GE.BTOL) GO TO 40                               CSL03740
C                                                               CSL03750
C     DEFAULT.  BSIZE IS SMALLER THAN TOLERANCE BTOL SPECIFIED IN MAIN  CSL03760
C     PROGRAM.  PROGRAM TERMINATES FOR USER TO DECIDE WHAT TO DO  CSL03770
C     BECAUSE LOCAL ORTHOGONALITY OF THE LANCZOS VECTORS COULD BE  CSL03780
C     LOST.                                                     CSL03790
C                                                               CSL03800
      IB = -IB                                                  CSL03810
      WRITE(6,30) MEV                                           CSL03820
   30 FORMAT(/' BETA TEST INDICATES POSSIBLE LOSS OF LOCAL ORTHOGONALITYCSL03830
     1 OVER 1ST',I6,' LANCZOS VECTORS'/)                        CSL03840
C                                                               CSL03850
   40 CONTINUE                                                  CSL03860
C                                                               CSL03870
      WRITE(6,50) IB                                            CSL03880
   50 FORMAT(/' MINIMUM BETA RATIO OCCURS AT',I6,' TH BETA'/)   CSL03890
C                                                               CSL03900
      WRITE(6,60) MEV,BMIN,TMAX,BSIZE                           CSL03910
   60 FORMAT(/1X,'TSIZE',6X,'MIN BETA',5X,'TKMAX',6X,'MIN RATIO'/  CSL03920
     1 I6,E14.3,E10.3,E15.3/)                                   CSL03930
C                                                               CSL03940
C-----END OF TNORM---------------------------------------------------CSL03950
      RETURN                                                    CSL03960
      END                                                       CSL03970
C                                                               CSL03980
C-----START OF LUMP-------------------------------------------------CSL03990
C                                                               CSL04000
      SUBROUTINE LUMP(VC,V1,VA,RELTOL,SPUTOL,SCALE2,LINDEX,TFLAG,LOOP)  CSL04010
C                                                               CSL04020
C------------------------------------------------------------------CSL04030
      COMPLEX*16 VC(1),V1(1),ZEROC,SUMC                         CSL04040
      DOUBLE PRECISION  VA(1),RELTOL,SPUTOL,SCALE2              CSL04050
      DOUBLE PRECISION  THOLD,TH1,TH2,DGAP,ZERO,ONE             CSL04060
      INTEGER  LINDEX(1),TFLAG(1)                               CSL04070
      DOUBLE PRECISION  DFLOAT, DMAX1, CDABS                    CSL04080
C     COMPLEX*16 DCMPLX                                         CSL04090
C------------------------------------------------------------------CSL04100
C     VC(J) = JTH DISTINCT T-EIGENVALUE,  VA(J) = |VC(J)|, IN ORDER  CSL04110
C     OF INCREASING MAGNITUDE.                                  CSL04120
C     LINDEX(J) = T-MULTIPLICITY OF JTH DISTINCT T-EIGENVALUE   CSL04130
C     LOOP = NUMBER OF DISTINCT T-EIGENVALUES                   CSL04140
C     LUMP 'COMBINES' COMPUTED 'GOOD' T-EIGENVALUES THAT ARE 'TOO CLOSE'CSL04150
C     VALUE OF RELTOL IS 1.D-8.                                 CSL04160
C                                                               CSL04170
C     IF IN A SET OF T-EIGENVALUES TO BE COMBINED THERE IS AN EIGENVALUECSL04180
C     WITH LINDEX=1, THEN THE VALUE OF THE COMBINED T-EIGENVALUES IS SETCSL04190
C     EQUAL TO THE VALUE OF THAT EIGENVALUE.  NOTE THAT IF A SPURIOUS  CSL04200
C     T-EIGENVALUE IS TO BE 'COMBINED' WITH A GOOD EIGENVALUE, THEN THISCSL04210
C     IS DONE ONLY BY INCREASING THE INDEX, LINDEX, FOR THAT EIGENVALUE  CSL04220
C     NUMERICAL VALUES OF SPURIOUS T-EIGENVALUES ARE NEVER COMBINED WITHCSL04230
C     THOSE OF GOOD T-EIGENVALUES.                              CSL04240
C------------------------------------------------------------------CSL04250
      ZERO = 0.0D0                                              CSL04260
      ONE = 1.D0                                                CSL04270
      ZEROC = DCMPLX(ZERO,ZERO)                                 CSL04280
```

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

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

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

```
C                                                                      CSL05940
      MEV1 = MEV - 1                                                   CSL05950
C                                                                      CSL05960
      IF (SAVTEV.GE.0) GO TO 40                                        CSL05970
C                                                                      CSL05980
      READ(10,10) MEV                                                  CSL05990
   10 FORMAT(I6)                                                       CSL06000
   20 FORMAT(20A4)                                                     CSL06010
      MEV1 = MEV - 1                                                   CSL06020
      READ(10,30) (VS(K), K = 1,MEV)                                   CSL06030
   30 FORMAT(4Z20)                                                     CSL06040
      READ(10,20) EXPLAN                                               CSL06050
      READ(10,20) EXPLAN                                               CSL06060
      READ(10,30) (V2(K), K = 1,MEV1)                                  CSL06070
      GO TO 90                                                         CSL06080
C                                                                      CSL06090
   40 CONTINUE                                                         CSL06100
C                                                                      CSL06110
      DO 50 J = 1,MEV                                                  CSL06120
      VS(J) = ALPHA(J)                                                 CSL06130
   50 V1(J) = BETA(J)                                                  CSL06140
C                                                                      CSL06150
      WRITE(6,60) MEV                                                  CSL06160
   60 FORMAT(/' COMPUTE EIGENVALUES OF T(1,',I4,') USING CMTQL1'/)     CSL06170
C                                                                      CSL06180
C----------------------------------------------------------------------CSL06190
      CALL CMTQL1(MEV,VS,V1,IERR)                                      CSL06200
C----------------------------------------------------------------------CSL06210
C                                                                      CSL06220
C     WRITE(6,70) IERR                                                 CSL06230
   70 FORMAT(' T-EIGENVALUES VIA CMTQL1'/' IERR = ',I6/)               CSL06240
C                                                                      CSL06250
      IF (IERR.EQ.0) GO TO 90                                          CSL06260
C                                                                      CSL06270
      WRITE(6,80)                                                      CSL06280
   80 FORMAT(' ON RETURN FROM CMTQL1 ERROR FLAG WAS NOT ZERO'/)        CSL06290
      GO TO 410                                                        CSL06300
C                                                                      CSL06310
   90 CONTINUE                                                         CSL06320
C                                                                      CSL06330
C     T-EIGENVALUES ARE IN VS IN INCREASING ORDER OF MAGNITUDE         CSL06340
      DO 100 J = 1,MEV                                                 CSL06350
  100 EVMAG(J) = CDABS(VS(J))                                          CSL06360
C                                                                      CSL06370
C     THE MAGNITUDES OF THE T-EIGENVALUES ARE IN EVMAG, IN ORDER OF    CSL06380
C     INCREASING MAGNITUDE                                             CSL06390
C     WRITE(13,105) (EVMAG(J), J = 1,MEV)                              CSL06400
C 105 FORMAT(' MAGNITUDES OF T-EIGENVALUES'/(4E20.12))                 CSL06410
C                                                                      CSL06420
      IF(SAVTEV.NE.1) GO TO 130                                        CSL06430
      WRITE(10,110) MEV                                                CSL06440
  110 FORMAT(I6,' = ORDER OF T-MATRIX, T-EIGVALS =')                   CSL06450
      WRITE(10,120) (VS(J), J = 1,MEV)                                 CSL06460
C 120 FORMAT(4Z20)                                                     CSL06470
  120 FORMAT(4E20.12)                                                  CSL06480
```

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

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

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

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

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

```
   90 CONTINUE                                             CSL09240
C     END OF I LOOP                                        CSL09250
C                                                          CSL09260
C     UPDATE PARAMETERS FOR I = L CASE                     CSL09270
      D(L) = D(L) - P                                      CSL09280
      E(L) = G                                             CSL09290
      E(M) = CZERO                                         CSL09300
      GO TO 20                                             CSL09310
C                                                          CSL09320
C     ORDER EIGENVALUES   P = D(L)                         CSL09330
  100 IF (L.EQ.1) GO TO 120                                CSL09340
      DO 110 II = 2,L                                      CSL09350
      I = L+2-II                                           CSL09360
      IF (CDABS(P).GE.CDABS(D(I-1))) GO TO 130             CSL09370
      D(I) = D(I-1)                                        CSL09380
  110 CONTINUE                                             CSL09390
C                                                          CSL09400
  120 I = 1                                                CSL09410
C                                                          CSL09420
  130 D(I) = P                                             CSL09430
C                                                          CSL09440
  140 CONTINUE                                             CSL09450
      GO TO 160                                            CSL09460
C                                                          CSL09470
  150 IERR = L                                             CSL09480
C-----END OF CMTQL1-------------------------------------------CSL09490
  160 RETURN                                               CSL09500
      END                                                  CSL09510
C                                                          CSL09520
C-----COMGAP-------------------------------------------------CSL09530
C                                                          CSL09540
      SUBROUTINE COMGAP(VC,VA,GG,MP,IND,M,IGAP,ITAG)       CSL09550
C                                                          CSL09560
C-----------------------------------------------------------CSL09570
      COMPLEX*16 VC(1),Z                                   CSL09580
      DOUBLE PRECISION  VA(1),T0,T1,TU,TK                  CSL09590
      REAL  GG(1),GTEMP                                    CSL09600
      INTEGER  MP(1),IND(1)                                CSL09610
      REAL ABS                                             CSL09620
      DOUBLE PRECISION CDABS                               CSL09630
C-----------------------------------------------------------CSL09640
C     IF IGAP = 0 WE DO NOT ORDER EIGENVALUES BY INCREASING GAP SIZE   CSL09650
C     AND WE DO NOT WRITE GAP OUTPUT TO FILE 12            CSL09660
C                                                          CSL09670
C     VA(K) = |VC(K)|  VA(K) <= VA(K+1)                    CSL09680
C     GG(K) = MIN |VC(K)-VC(J)|  J .NE. K.                 CSL09690
C-----------------------------------------------------------CSL09700
      TU = VA(M) + VA(M)                                   CSL09710
      K = 0                                                CSL09720
   10 K = K+1                                              CSL09730
      IF (K.GT.M) GO TO 60                                 CSL09740
      INDEX = 0                                            CSL09750
      T1 = TU                                              CSL09760
      TK = VA(K)                                           CSL09770
      Z = VC(K)                                            CSL09780
```

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

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

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

```
      RATIO = BETA(I)/U                                         CSL11440
      U = ALPHA(I)-X1-Z*RATIO                                   CSL11450
      Z = BETA(I+1)                                             CSL11460
      GO TO 60                                                  CSL11470
C     PIVOT INTERCHANGE                                         CSL11480
   50 CONTINUE                                                  CSL11490
      RATIO = U/BETA(I)                                         CSL11500
      INTERC(I) = 1                                             CSL11510
      V1(I-1) = ALPHA(I)-X1                                     CSL11520
      U = Z-RATIO*V1(I-1)                                       CSL11530
      Z = -RATIO*BETA(I+1)                                      CSL11540
      TEMP = V2(I-1)                                            CSL11550
      V2(I-1) = V2(I)                                           CSL11560
      V2(I) = TEMP-RATIO*V2(I)                                  CSL11570
   60 CONTINUE                                                  CSL11580
C                                                               CSL11590
      IF (CDABS(U).EQ.ZERO) U= DCMPLX(EPS3,EPS3)                CSL11600
C                                                               CSL11610
C     SMALLNESS TEST AND DEFAULT VALUE FOR LAST COMPONENT       CSL11620
C     PIVOT(I-1) = |BETA(I)| FOR INTERCHANGE CASE               CSL11630
C     (I-1,I+1) ELEMENT IN RIGHT FACTOR = BETA(I+1)             CSL11640
C     END OF FACTORIZATION AND FORWARD SUBSTITUTION             CSL11650
C                                                               CSL11660
C     BACK SUBSTITUTION                                         CSL11670
      V2(MEV) = V2(MEV)/U                                       CSL11680
      DO 80 II = 1,MM1                                          CSL11690
      I = MEV-II                                                CSL11700
      IF (INTERC(I+1).EQ.1) GO TO 70                            CSL11710
C     NO PIVOT INTERCHANGE                                      CSL11720
      V2(I) = V2(I)-V1(I)*V2(I+1)                               CSL11730
      GO TO 80                                                  CSL11740
C     PIVOT INTERCHANGE                                         CSL11750
   70 V2(I) = (V2(I)-V1(I)*V2(I+1)-BETA(I+2)*V2(I+2))/BETA(I+1) CSL11760
   80 CONTINUE                                                  CSL11770
C                                                               CSL11780
C                                                               CSL11790
C     TESTS FOR CONVERGENCE OF INVERSE ITERATION                CSL11800
C     IF SUM |V2| COMPS. LE. 1 AND IT. LE. ITER DO ANOTHER INVIT STEP  CSL11810
C                                                               CSL11820
      NORM = CDABS(V2(MEV))                                     CSL11830
      DO 90 II = 1,MM1                                          CSL11840
      I = MEV-II                                                CSL11850
   90 NORM = NORM+CDABS(V2(I))                                  CSL11860
C                                                               CSL11870
C     IS DESIRED GROWTH IN VECTOR ACHIEVED ?                    CSL11880
C     IF NOT, DO ANOTHER INVERSE ITERATION STEP UNLESS NUMBER ALLOWED ISCSL11890
C     EXCEEDED.                                                 CSL11900
      IF (NORM.GE.ONE) GO TO 110                                CSL11910
C                                                               CSL11920
      IT=IT+1                                                   CSL11930
      IF (IT.GT.ITER) GO TO 110                                 CSL11940
C                                                               CSL11950
      XU = EPS4/NORM                                            CSL11960
      DO 100 I=1,MEV                                            CSL11970
      INTERC(I) = 0                                             CSL11980
```

```
  100 V2(I) = V2(I)*XU                                             CSL11990
C                                                                  CSL12000
      GO TO 40                                                     CSL12010
C                                                                  CSL12020
C     NORMALIZE COMPUTED T-EIGENVECTOR : V2 = V2/||V2||            CSL12030
C                                                                  CSL12040
  110 CONTINUE                                                     CSL12050
C                                                                  CSL12060
C-----------------------------------------------------------------CSL12070
      CALL CINPRD(V2,V2,SUM,MEV)                                   CSL12080
C-----------------------------------------------------------------CSL12090
C                                                                  CSL12100
      SUM = ONE/DSQRT(SUM)                                         CSL12110
      DO 120 II = 1,MEV                                            CSL12120
  120 V2(II) = SUM*V2(II)                                          CSL12130
C                                                                  CSL12140
C     SAVE ERROR ESTIMATE FOR LATER OUTPUT                         CSL12150
      ERROR = CDABS(V2(MEV))                                       CSL12160
C                                                                  CSL12170
C     GENERATE ERRORV = ||T*V2 - X1*V2||.                         CSL12180
C     LOOP IS BOTTOM UP BECAUSE LAST COMPONENTS MAY BE VERY SMALL  CSL12190
      V1(MEV) = ALPHA(MEV)*V2(MEV)+BETA(MEV)*V2(MEV-1)-X1*V2(MEV)  CSL12200
      DO 130 J = 2,MM1                                             CSL12210
      JM = MP1 - J                                                 CSL12220
      V1(JM) = ALPHA(JM)*V2(JM) + BETA(JM)*V2(JM-1) + BETA(JM+1)*V2(JM+1CSL12230
     1) - X1*V2(JM)                                                CSL12240
  130 CONTINUE                                                     CSL12250
C                                                                  CSL12260
      V1(1) = ALPHA(1)*V2(1) + BETA(2)*V2(2) - X1*V2(1)            CSL12270
C                                                                  CSL12280
C-----------------------------------------------------------------CSL12290
      CALL CINPRD(V1,V1,ERRORV,MEV)                                CSL12300
C-----------------------------------------------------------------CSL12310
C                                                                  CSL12320
      ERRORV = DSQRT(ERRORV)                                       CSL12330
      IF (IT.GT.ITER) ERRORV = -ERRORV                            CSL12340
      IF (IWRITE.EQ.0) GO TO 150                                   CSL12350
C                                                                  CSL12360
C     FILE 6 (TERMINAL) OUTPUT OF ERROR ESTIMATES.                CSL12370
      WRITE(6,140) MEV,X1,ERROR,ERRORV                             CSL12380
  140 FORMAT(1X,'TSIZE',10X,'RE(GOODEV)',10X,'IM(GOODEV)',11X,'U(M)', CSL12390
     1 9X,'TERROR'/I6,2E20.12,2E15.5)                              CSL12400
C                                                                  CSL12410
C     RESTORE BETA(MEV+1) = BETAM                                  CSL12420
  150 CONTINUE                                                     CSL12430
      BETA(MP1) = BETAM                                            CSL12440
C-----END OF INVERM-----------------------------------------------CSL12450
      RETURN                                                       CSL12460
      END                                                          CSL12470
C                                                                  CSL12480
C-----START OF INNER PRODUCT ROUTINE------------------------------CSL12490
C                                                                  CSL12500
C     COMPUTES EUCLIDEAN INNER PRODUCT OF 2 COMPLEX VECTORS        CSL12510
C     SUMC = (V2-TRANSPOSE)*V1                                     CSL12520
C                                                                  CSL12530
```

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

7.7   CSLEVAL: CSLEVEC: File Definitions, Sample Input Files

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

CSLEVAL computes eigenvalues of diagonalizable complex symmetric matrices.

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




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

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

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

```
Sample Specifications of the Input/Output Files for CSLEVEC
-------------------------------------------------------------------
  CSLEVEC EXEC LANCZOS EIGENVECTOR PROGRAM COMPLEX SYMMETRIC CASE
FI 06 TERM
FILEDEF  2 DISK &1       HISTORY   A (RECFM F LRECL 80 BLOCK 80
FILEDEF  3 DISK &1       GOODEV    A (RECFM F LRECL 80 BLOCK 80
FILEDEF  4 DISK &1       ERRINV    A (RECFM F LRECL 80 BLOCK 80
FILEDEF  5 DISK CSLEVEC  INPUT     A (RECFM F LRECL 80 BLOCK 80
FILEDEF  8 DISK &1       INPUT     A (RECFM F LRECL 80 BLOCK 80
FILEDEF  9 DISK &1       ERREST    A (RECFM F LRECL 80 BLOCK 80
FILEDEF 10 DISK &1       BOUNDS    A (RECFM F LRECL 80 BLOCK 80
FILEDEF 11 DISK &1       TEIGVECS  A (RECFM F LRECL 80 BLOCK 80
FILEDEF 12 DISK &1       RITZVECS  A (RECFM F LRECL 80 BLOCK 80
FILEDEF 13 DISK &1       PAIGE     A (RECFM F LRECL 80 BLOCK 80
LOAD  CSLEVEC  CSLESUB  CSLEMULT
-------------------------------------------------------------------
```

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

Chapter 8

Real Symmetric Matrices, Block Lanczos Code

8.1 Introduction

The FORTRAN codes in this chapter address the question of using an iterative 'block' Lanczos procedure to compute a 'few' extreme eigenvalues and a basis for the corresponding invariant subspace of a given real symmetric matrix A. An eigenvalue is extreme if it is one of the algebraically-smallest or the algebraically-largest eigenvalues.

For a given real symmetric matrix A, these codes compute the q algebraically-largest eigenvalues, $\lambda_i, 1 \leq i \leq q$, of A and corresponding orthonormal real vectors $X_q \equiv (x_1, \ldots, x_q)$ such that

$$AX_q = X_q A_q, \quad A_q \equiv X_q^T A X_q. \tag{8.1.1}$$

Typically, $A_q = \Lambda_q$, a diagonal matrix whose nonzero entries are the eigenvalues λ_i . The number q is small and specified by the user.

Real symmetric matrices are discussed in detail in Stewart [24]. See Section 2.1 for a brief summary of the properties of real symmetric matrices which we use. The Lanczos procedure included in this chapter is not a true block Lanczos procedure. It is a hybrid Lanczos algorithm which combines ideas from the iterative block Lanczos procedures such as the one in Cullum and Donath [4, 3] and from the single-vector Lanczos procedure given in Chapter 2.

Several differences between the single-vector Lanczos codes in Chapters 2 through Chapter 7 and the iterative 'block' Lanczos codes should be stated explicitly. The single-vector Lanczos codes do not have the capability of directly computing the A-multiplicities of the computed eigenvalues. The 'block' procedures however, will determine the true A-multiplicity of a given computed eigenvalue and compute a complete invariant subspace for such an eigenvalue, as long as the number of Lanczos vectors in the first block is large enough. In order to determine A-multiplicities the single-vector codes have to do additional computation. In some cases these multiplicities and a basis for the required eigenspace can be determined without too much additional computation. This is true for example, whenever the desired eigenvalues replicate readily during the single-vector Lanczos computations.

The single-vector Lanczos procedures in Chapters 2 through Chapter 7 function in two stages. First the eigenvalues of the matrix being considered are computed, and then a separate program is used to compute

the corresponding desired eigenvectors. The iterative 'block' Lanczos codes obtain approximations to the eigenvalues and to the eigenvectors simultaneously. Both types of codes are restartable from pre-existing computations. However, restarting has a different meaning for the two different types of codes. In the single-vector codes, restarting means computing a larger Lanczos T-matrix, starting from a pre-existing smaller one. The eigenvalue and eigenvector computations are then repeated on the larger T-matrix. In the iterative block procedures, restarting means using the current approximations to the eigenvectors (or more correctly to a basis for the desired eigenspace), to initiate another iteration of the 'block' Lanczos procedure.

The single-vector Lanczos procedures in Chapters 2 through 7 are iterative only in the sense that one may consider several Lanczos T-matrices of different sizes before achieving the desired convergence. However, the 'block' procedure presented here is genuinely iterative. On each iteration a block version of the Lanczos recursion is used to generate a sequence of blocks of Lanczos vectors, simultaneously generating a 'small' real symmetric Lanczos T-matrix. The eigenvalues and eigenvectors of this small Lanczos matrix are computed and mapped into approximating eigenvectors for the given matrix using the Lanczos vectors. These approximate eigenvectors then become the starting block of Lanczos vectors for the next iteration of the block Lanczos procedure. This 'block' procedure is described in detail in Section 7.5 of Chapter 7 in Volume 1.

As we said earlier, the 'block' procedure included here is a hybrid of the single-vector and of the basic iterative block Lanczos procedures. This procedure is based upon a modification of the following basic block version of the Lanczos recursion

$$Q_{j+1}B_{j+1} = AQ_j - Q_jA_j - Q_{j-1}B_j^T \equiv P_j \qquad (8.1.2)$$

for $j = 1, 2, \ldots, s$ where the coefficient matrices A_j and B_{j+1} are block analogs of the scalar coefficients in the single vector Lanczos recursion. In the standard block procedure,

$$A_j \equiv Q_j^T(AQ_j - Q_{j-1}B_j^T) \qquad (8.1.3)$$

and each B_{j+1} is obtained by the Gram-Schmidt orthogonalization of the columns of P_j and $s \ll n$, the order of the given A-matrix. Our single-vector Lanczos procedures do not use any reorthogonalization at any point in the computations. However, in our block procedures we require near-orthogonality of the Q-blocks. This orthogonality is maintained by incorporating reorthogonalization of the blocks generated within a given iteration, with respect to certain vectors in the first Lanczos block.

The sequence of 'blocks' generated on each iteration of this hybrid procedure has the property that the first Q-block, Q_1, contains at least as many vectors as the user is trying to compute. However, the second and succeeding blocks contain exactly one vector. The corresponding Lanczos T-matrices are not block tridiagonal. Each has a border of blocks occupying the first q rows and columns and is tridiagonal below these rows and columns.

The convergence of these procedures is monitored by the subroutine DIAGOM. Convergence requires reasonable gaps between the eigenvalues requested and the eigenvalues not being approximated by the block procedure. Typically, it is the ratio of these gaps to the spread, and the distribution of the A-eigenvalues over the A-spread which controls the rate of convergence. In particular, an iterative block Lanczos procedure may have difficulty with a matrix with evenly-distributed eigenvalues. Heuristics are incorporated which allow the number of vectors used in the first Lanczos block to vary. If the convergence stagnates the procedure will terminate to allow the user to intervene and reset the program parameters if desired.

BLEVAL, the main 'block' program for these real symmetric eigenelement computations, calls the subroutine LANCZS which on each iteration then calls the subroutine LANCI1 to generate a sequence of Q-blocks for that iteration. Subroutine LANCZS then calls the subroutine DIAGOM to diagonalize the

Lanczos T-matrix generated on that iteration and to compute the updated approximations to the desired eigenspace. Convergence is checked and if it has not occurred, another iteration of the block Lanczos procedure is carried out.

In this 'block' procedure there is no identification or 'spurious' test for the eigenvalues of the Lanczos T-matrix. Since near-orthogonality of the Lanczos blocks is maintained, the q algebraically-largest eigenvalues of the T-matrices are approximations to the q algebraically-largest eigenvalues of the A-matrix being used in the recursions. This statement however, is not true for the other eigenvalues of these T-matrices because the orthogonality maintained is only with respect to the eigenspace which goes with the first q eigenvalues. The accuracy of the computed eigenvalues and eigenvectors is estimated on each iteration as part of the process of computing the second block of Lanczos vectors.

All computations are in double precision real arithmetic. The user must supply a subroutine USPEC which defines and initializes the A-matrix and a subroutine BMATV which computes Ax for any specified vector x. The small T-matrix eigenelement computations use two subroutines from the EISPACK Library [23, 8], TRED2 and IMTQL2. If the q algebraically-smallest eigenvalues are required, then the user must supply the programs with a subroutine which computes $-Ax$ rather that Ax. The user should refer to Chapter 7 in Volume 1 for more details on iterative block Lanczos procedures.

8.2   Documentation for the Codes in Chapters 8 and 9

```
C-----BLEVALHD-------------------------------------------------BLE00010
C  Authors:  Jane Cullum* and Bill Donath**                    BLE00020
C         **IBM Research, T.J. Watson Research Center           BLE00030
C         **Yorktown Heights, N.Y. 10598                        BLE00040
C         * Los Alamos National Laboratory                      BLE00050
C         * Los Alamos, New Mexico 87544                         BLE00060
C            E-mail:  cullumj@lanl.gov                           BLE00065
C                                                                BLE00070
C  These codes are copyrighted by the authors.  These codes     BLE00080
C  and modifications of them or portions of them are NOT to be   BLE00090
C  incorporated into any commercial codes or used for any other  BLE00100
C  commercial purposes such as consulting for other companies,   BLE00110
C  without legal agreements with the authors of these Codes.     BLE00120
C  If these Codes or portions of them are used in other scientific or BLE00130
C  engineering research works the names of the authors of these codes BLE00140
C  and appropriate references to their written work are to be    BLE00150
C  incorporated in the derivative works.                         BLE00160
C                                                                BLE00170
C  This header is not to be removed from these codes.            BLE00180
C                                                                BLE00190
C                                                                BLE00200
C     DOCUMENTATION BLOCK LANCZOS EIGENVALUE/EIGENVECTOR PROGRAMS BLE00210
C     (1) REAL SYMMETRIC MATRICES                                BLE00220
C     (2) FACTORED INVERSES OF REAL SYMMETRIC MATRICES           BLE00230
C                                                                BLE00240
C-------------------------------------------------------------BLE00250
C         REFERENCE: Cullum and Willoughby, Chapter 7,          BLE00260
C         Lanczos Algorithms for Large Symmetric Eigenvalue ComputationsBLE00270
C         VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in  BLE00280
C         Applied Mathematics, 2002. SIAM Publications,          BLE00290
C         Philadelphia, PA. USA                                  BLE00290
C                                                                BLE00300
C                                                                BLE00310
C                                                                BLE00320
C-------------------------------------------------------------BLE00330
C                                                                BLE00340
C     REAL SYMMETRIC MATRICES:                                   BLE00350
C                                                                BLE00360
C     GIVEN A REAL SYMMETRIC MATRIX A THE FILES BLEVAL, BLSUB AND BLE00370
C     BLMULT CAN BE USED TO COMPUTE A FEW EXTREME EIGENVALUES    BLE00380
C     OF A, THAT IS THE ALGEBRAICALLY-LARGEST OR THE ALGEBRAICALLY- BLE00390
C     SMALLEST EIGENVALUES, AND A BASIS FOR THE CORRESPONDING    BLE00400
C     EIGENSPACE.                                                BLE00410
C                                                                BLE00420
C     FACTORED INVERSES OF REAL SYMMETRIC MATRICES:              BLE00430
C                                                                BLE00440
C     GIVEN A REAL SYMMETRIC MATRIX A, THE BLOCK PROCEDURE       BLE00450
C     CAN BE APPLIED TO AN ASSOCIATED B-MATRIX WHICH IS A        BLE00460
C     SCALED, SHIFTED AND PERMUTED VERSION OF A.  THAT IS,       BLE00470
C     B = S0*P*A*P' + SHIFT*I WHERE THE SCALE S0 AND THE SHIFT   BLE00480
C     ARE CHOSEN BY THE USER TO PLACE THE DESIRED EIGENVALUES    BLE00490
C     AT THE EXTREME OF THE SPECTRUM OF B-INVERSE, AND THE       BLE00500
```

```
C      PERMUTATION P IS CHOSEN SO THAT THE SPARSITY OF THE A-MATRIX        BLE00510
C      IS PRESERVED IN THE SPARSITY OF THE FACTORIZATION OF B.            BLE00520
C      THE INVERSE BLOCK PROCEDURE REQUIRES A SUBROUTINE BLSOLV           BLE00530
C      THAT FOR A GIVEN VECTOR U, COMPUTES THE VECTOR V SUCH THAT         BLE00540
C      B*V = U, USING THE FACTORIZATION OF B.  THE SAMPLE BLSOLV          BLE00550
C      SUBROUTINE PROVIDED ASSUMES THAT THE B-MATRIX IS POSITIVE          BLE00560
C      DEFINITE AND THAT THE CHOLESKY FACTORS OF B ARE SUPPLIED           BLE00570
C      ON FILE 7.  HOWEVER, THE USER MAY REPLACE THIS SUBROUTINE          BLE00580
C      BY ONE THAT COMPUTES A MORE GENERAL FACTORIZATION                  BLE00590
C      L*D*(L-TRANSPOSE) FOR AN INDEFINITE SYMMETRIC MATRIX.              BLE00600
C      THE BLOCK PROCEDURE USED IN THIS FASHION USES THE FILES            BLE00610
C      BLIEVAL, BLIMULT AND BLSUB.                                        BLE00620
C                                                                         BLE00630
C                                                                         BLE00640
C      ALGORITHM:                                                         BLE00650
C      THESE PROGRAMS USE A BLOCK FORM OF LANCZOS TRIDIAGONALIZATION      BLE00660
C      WITH REORTHOGONALIZATION ONLY WITH RESPECT TO VECTORS             BLE00670
C      IN THE 1ST Q-BLOCK.  THE PROCEDURES ARE ITERATIVE, GENERATING      BLE00680
C      ON EACH ITERATION A SMALL SYMMETRIC LANCZOS MATRIX, T.            BLE00690
C      THE EIGENVALUES AND EIGENVECTORS OF THE SMALL MATRIX ARE           BLE00700
C      COMPUTED USING SUBROUTINES FROM THE EISPACK LIBRARY.              BLE00710
C      THE RELEVANT SUBSET OF THE T-EIGENVECTORS IS THEN MAPPED           BLE00720
C      INTO THE LARGE N-SPACE CORRESPONDING TO THE MATRIX BEING           BLE00730
C      USED BY THE LANCZS SUBROUTINE, CONVERGENCE IS CHECKED,             BLE00740
C      AND IF CONVERGENCE OF THE DESIRED EIGENVALUES AND                  BLE00750
C      EIGENVECTORS HAS NOT YET OCCURRED, THEN THE CURRENT                BLE00760
C      APPROXIMATIONS TO THE DESIRED EIGENSPACE ARE USED AS               BLE00770
C      STARTING VECTORS FOR THE NEXT ITERATION OF BLOCK LANCZOS.          BLE00780
C                                                                         BLE00790
C      USERS SHOULD NOTE THAT TYPICALLY IN THE BLOCK LANCZOS              BLE00800
C      PROCEDURES, IT IS THE RATIO OF THE GAPS TO THE SPREAD THAT         BLE00810
C      CONTROLS THE CONVERGENCE ALONG WITH HOW THE EIGENVALUES            BLE00820
C      ARE DISTRIBUTED OVER THAT SPREAD.   THE BIGGER THE GAPS            BLE00830
C      BETWEEN THE ONES BEING COMPUTED AND THE CLOSEST ONES NOT           BLE00840
C      BEING COMPUTED AND THE WEAKER THE SPREAD, THE FASTER THE           BLE00850
C      CONVERGENCE WILL BE.  WITHOUT DECENT GAPS THIS PROCEDURE           BLE00860
C      WILL NOT CONVERGE.  THE PROGRAMS CONTAIN CHECKS ON                 BLE00870
C      THE ACTUAL RATE OF CONVERGENCE WHICH WILL CAUSE THE                BLE00880
C      PROCEDURE TO TERMINATE IF CONVERGENCE IS NOT OCCURRING             BLE00890
C      SUFFICIENTLY RAPIDLY.  THE USER MAY THEN CHANGE EITHER OR          BLE00900
C      BOTH THE MAXIMUM SIZE T-MATRIX ALLOWED AND THE NUMBER              BLE00910
C      OF VECTORS IN THE FIRST Q-BLOCK AND RERUN THE PROCEDURE            BLE00920
C      WITH THE CURRENT APPROXIMATION TO THE DESIRED EIGENSPACE           BLE00930
C      AS THE STARTING BLOCK OF VECTORS.                                  BLE00940
C                                                                         BLE00950
C                                                                         BLE00960
C      THE IDEAS USED IN THESE PROGRAMS ARE DISCUSSED IN THE FOLLOWING    BLE00970
C      REFERENCES.                                                        BLE00980
C                                                                         BLE00990
C      1.  JANE CULLUM AND RALPH A. WILLOUGHBY, LANCZOS ALGORITHMS        BLE01000
C          FOR LARGE SYMMETRIC MATRICES,  PROGRESS IN                     BLE01010
C          SCIENTIFIC COMPUTING, EDITORS, G. GOLUB, H.O. KREISS,          BLE01020
C          S. ARBARBANEL, AND R. GLOWINSKI,  BIRKHAUSER BOSTON INC.,      BLE01030
C          CAMBRIDGE, MASSACHUSETTS, 1984.                                BLE01040
C                                                                         BLE01050
```

```
C     2.  JANE CULLUM AND W.E. DONATH, A BLOCK LANCZOS ALGORITHM      BLE01060
C         FOR COMPUTING THE Q ALGEBRAICALLY-LARGEST EIGENVALUES AND   BLE01070
C         A CORRESPONDING EIGENSPACE OF LARGE, SPARSE REAL SYMMETRIC  BLE01080
C         MATRICES, PROCEEDINGS OF THE 1974 IEEE CONFERENCE ON        BLE01090
C         DECISION AND CONTROL, PHOENIX, ARIZONA, PP.505-509, NOVEMBER BLE01100
C         1974.                                                       BLE01110
C                                                                     BLE01120
C     3.  JANE CULLUM,  AN ACCELERATED 'BLOCK' LANCZOS ALGORITHM      BLE01130
C         FOR A FEW EXTREME EIGENVALUES OF A LARGE, SPARSE REAL       BLE01140
C         SYMMETRIC MATRIX.  IBM REPORT 1983.  PRESENTED AT THE       BLE01150
C         SPARSE MATRIX CONFERENCE, FAIRFIELD GLADE, TENNESSEE,       BLE01160
C         OCTOBER 1982.                                               BLE01170
C                                                                     BLE01180
C                                                                     BLE01190
C-----PORTABILITY-----------------------------------------------------BLE01200
C                                                                     BLE01210
C     PROGRAMS WERE TESTED FOR PORTABILITY USING THE PFORT VERIFIER.  BLE01220
C     FOR DETAILS OF THE VERIFIER SEE FOR EXAMPLE, B. G. RYDER AND    BLE01230
C     A. D. HALL, "THE PFORT VERIFIER", COMPUTING SCIENCE TECHNICAL   BLE01240
C     REPORT 12, BELL LABORATORIES, MURRAY HILL, NEW JERSEY 07974,    BLE01250
C     (REVISED), JANUARY 1981.                                        BLE01260
C                                                                     BLE01270
C     EXCEPT FOR THE FOLLOWING CONSTRUCTIONS WHICH CAN BE EASILY      BLE01280
C     MODIFIED BY THE USER TO MATCH THE PARTICULAR COMPUTER BEING     BLE01290
C     USED, THE PROGRAM STATEMENTS ARE PORTABLE.                      BLE01300
C                                                                     BLE01310
C     NONPORTABLE STATEMENTS.                                         BLE01320
C                                                                     BLE01330
C     IN BLEVAL, BLIEVAL (MAIN PROGRAMS)                              BLE01340
C         1.  DATA/MACHEP STATEMENT                                   BLE01350
C         2.  ALL READ(5,*) STATEMENTS (FREE FORMAT)                  BLE01360
C         3.  FORMAT(20A4) USED FOR THE EXPLANATORY HEADER ARRAY, EXPLANBLE01370
C         4.  FORMAT(4Z20) WHICH CAN BE USED TO WRITE LARGE VECTOR    BLE01380
C             FILES                                                   BLE01390
C         5.  THE COMMON BLOCK: LOOPS.                                BLE01400
C     IN BLMULT, BLIMULT                                              BLE01410
C         1.  IN BMATV, BLSOLV, AND USPEC, THE ENTRIES WHICH          BLE01420
C             PASS THE STORAGE LOCATIONS OF THE ARRAYS DEFINING       BLE01430
C             THE USER-SPECIFIED MATRIX OR FACTORIZATION.            BLE01440
C     IN BLSUB                                                        BLE01450
C         1.  ALL STATEMENTS ARE PORTABLE EXCEPT THE ENTRY TO         BLE01460
C             SUBROUTINE LPERM WHICH PASSES THE PERMUTATION USED      BLE01470
C             TO OBTAIN THE B-MATRIX FROM SUBROUTINE USPEC.           BLE01480
C             SUBROUTINE LPERM IS USED ONLY IN CASE (2).              BLE01490
C                                                                     BLE01500
C                                                                     BLE01510
C-----MATRIX SPECIFICATION--------------------------------------------BLE01520
C                                                                     BLE01530
C     SUBROUTINE USPEC IS USED TO SPECIFY THE MATRIX WHICH THE BLOCK  BLE01540
C     LANCZOS PROCEDURE WILL USE.  IN CASE (1) THIS IS THE USER-      BLE01550
C     SPECIFIED A-MATRIX.   IN CASE (2) THE FACTORIZATION OF THE      BLE01560
C     ASSOCIATED B-MATRIX IS SPECIFIED.  SUBROUTINE USPEC HAS THE     BLE01570
C     CALLING SEQUENCE                                                BLE01580
C                                                                     BLE01590
C         CALL USPEC(N,MATNO,NNZ,AVER)                                BLE01600
```

```
C                                                                      BLE01610
C      WHERE N IS THE ORDER OF THE USER-SUPPLIED MATRIX A,             BLE01620
C      MATNO IS AN <= 8 DIGIT INTEGER USED AS A MATRIX AND             BLE01630
C      TEST IDENTIFICATION NUMBER, NNZ IS THE AVERAGE NUMBER           BLE01640
C      OF NONZERO ENTRIES IN EACH COLUMN, AND AVER IS THE              BLE01650
C      AVERAGE SIZE OF THE NONZERO ENTRIES IN THE MATRIX USED          BLE01660
C      BY LANCZS.  NOTE THAT NNZ AND AVER ARE DEFINED AS DOUBLE        BLE01670
C      PRECISION SCALARS.  THE MAIN PROGRAMS ASSUME THAT THEY          BLE01680
C      ARE COMPUTED IN USPEC.  THE USPEC SUBROUTINE                    BLE01690
C      DEFINES AND DIMENSIONS THE ARRAYS REQUIRED TO                   BLE01700
C      SPECIFY THE MATRIX THAT WILL BE USED BY THE LANCZS              BLE01710
C      SUBROUTINE AND INITIALIZES THESE ARRAYS.  THE STORAGE           BLE01720
C      LOCATIONS OF THESE ARRAYS ARE THEN PASSED TO THE                BLE01730
C      SUBROUTINE BMATV IN CASE (1) AND TO THE SUBROUTINE BSOLV        BLE01740
C      IN CASE (2).  SAMPLE SUBROUTINES ARE INCLUDED FOR EACH          BLE01750
C      CASE.  CASE (1) ASSUMES THAT THE A-MATRIX IS STORED ON          BLE01760
C      FILE 8.  CASE (2) ASSUMES THAT THE FACTORIZATION OF THE         BLE01770
C      B-MATRIX IS STORED ON FILE 7.                                   BLE01780
C                                                                      BLE01790
C      IN CASE (1) :                                                   BLE01800
C      BMATV IS THE SUBROUTINE USED BY THE LANCZS SUBROUTINE           BLE01810
C      THAT GENERATES THE LANCZOS T-MATRICES.  SUBROUTINE              BLE01820
C      BMATV HAS THE CALLING SEQUENCE                                  BLE01830
C                                                                      BLE01840
C           CALL BMATV(W,U)                                            BLE01850
C                                                                      BLE01860
C      WHERE U AND W ARE DOUBLE PRECISION VECTORS.  FOR A GIVEN        BLE01870
C      W, BMATV CALCULATES U = A*W  FOR THE USER-SPECIFIED MATRIX A.   BLE01880
C      A SAMPLE BMATV IS INCLUDED FOR AN ARBITRARY SPARSE,             BLE01890
C      SYMMETRIC A-MATRIX STORED IN THE SPARSE FORMAT SPECIFIED        BLE01900
C      IN THE CORRESPONDING SAMPLE USPEC SUBROUTINE.                   BLE01910
C                                                                      BLE01920
C      IN CASE (2):                                                    BLE01930
C      THE LANCZOS T-MATRICES ARE GENERATED USING SPARSE MATRIX        BLE01940
C      INVERSION, USING THE SUBROUTINE BLSOLV.  THE CALLING            BLE01950
C      SEQUENCE OF BLSOLV IS                                           BLE01960
C                                                                      BLE01970
C              CALL BLSOLV(U,V)                                        BLE01980
C                                                                      BLE01990
C      WHERE U AND V ARE DOUBLE PRECISION VECTORS.  FOR A GIVEN V,     BLE02000
C      BLSOLV COMPUTES U = (B-INVERSE)*V USING A SPARSE                BLE02010
C      FACTORIZATION OF THE B-MATRIX ASSOCIATED WITH THE USER-         BLE02020
C      SPECIFIED A-MATRIX.                                             BLE02030
C                                                                      BLE02040
C      THE FOLLOWING SPARSE MATRIX FORMAT IS USED TO STORE THE         BLE02050
C      MATRICES IN THE SAMPLE PROGRAMS:                                BLE02060
C      ICOL(K), K = 1,NZL, NUMBER OF SUBDIAGONAL NONZEROS IN COLUMN K. BLE02070
C      IROW(K), K = 1,NZS, ROW INDEX OF ASD(K).                        BLE02080
C      AD(K), K=1,N, CONTAINS THE DIAGONAL ELEMENTS OF THE A-MATRIX.   BLE02090
C      ASD(K), K=1,NZS  CONTAINS THE SUBDIAGONAL ELEMENTS OF A BY COLUMN.BLE02100
C      NZS = NUMBER OF NONZERO ELEMENTS BELOW THE DIAGONAL OF A        BLE02110
C      NZL = INDEX OF LAST COLUMN WITH NONZERO SUBDIAGONAL ENTRIES     BLE02120
C      N = ORDER OF THE A-MATRIX.                                      BLE02130
C                                                                      BLE02140
C      IN CASE (1) THE A-MATRIX IS STORED IN THIS FORMAT ON FILE 8.    BLE02150
```

```
C      IN CASE (2), IN THE SAMPLE USPEC PROVIDED WHICH IS ONLY        BLE02160
C      FOR POSITIVE DEFINITE B-MATRICES, THE SPARSE CHOLESKY FACTOR   BLE02170
C      OF B, L, IS STORED ON FILE 7 IN THE ABOVE SPARSE FORMAT        BLE02180
C      USING ARRAYS BD AND BSD.  IN CASE (2) THE OPTIONAL AUXILIARY   BLE02190
C      PROGRAMS PERMUT AND LORDER ALSO REQUIRE THE A-MATRIX;          BLE02200
C      HOWEVER, THE BLOCK LANCZOS PROCEDURE ONLY USES THE             BLE02210
C      FACTORIZATION OF THE B-MATRIX.                                 BLE02220
C                                                                     BLE02230
C                                                                     BLE02240
C-----MACHEP-------------------------------------------------------BLE02250
C                                                                     BLE02260
C                                                                     BLE02270
C      MACHEP IS A MACHINE DEPENDENT PARAMETER SPECIFYING THE RELATIVE BLE02280
C      PRECISION OF THE FLOATING POINT ARITHMETIC USED.              BLE02290
C      MACHEP = 2.2 * 10**-16 FOR DOUBLE PRECISION ARITHMETIC ON     BLE02300
C      IBM 370-3081.                                                  BLE02310
C                                                                     BLE02320
C      THE USER WILL HAVE TO RESET THIS PARAMETER TO                 BLE02330
C      THE CORRESPONDING VALUE FOR THE MACHINE BEING USED.  NOTE THAT BLE02340
C      IF A MACHINE WITH A MACHINE EPSILON THAT IS MUCH LARGER THAN THE BLE02350
C      VALUE GIVEN HERE IS BEING USED, THEN THERE COULD BE           BLE02360
C      PROBLEMS WITH THE TOLERANCES.                                 BLE02370
C                                                                     BLE02380
C                                                                     BLE02390
C-----SUBROUTINES AND FUNCTIONS USER MUST SUPPLY----------------------BLE02400
C                                                                     BLE02410
C                                                                     BLE02420
C      GENRAN,  FINPRO,  MASK,  USPEC,  AND                          BLE02430
C      CASE (1)  BMATV: CASE (2)  BLSOLV :                           BLE02440
C                                                                     BLE02450
C      GENRAN = COMPUTES K PSEUDO-RANDOM NUMBERS AND STORES THEM IN   BLE02460
C               THE REAL ARRAY, G.  THIS SUBROUTINE IS USED TO       BLE02470
C               GENERATE STARTING VECTORS FOR THE BLOCK LANCZOS      BLE02480
C               PROCEDURE.  CALLED FROM LANCZS SUBROUTINE.           BLE02490
C               USER CAN SUPPLY STARTING VECTORS FOR THE BLOCK       BLE02500
C               PROCEDURES.  ANY ADDITIONAL VECTORS REQUIRED ARE     BLE02510
C               GENERATED RANDOMLY BY GENRAN.  VECTORS SUPPLIED MUST  BLE02520
C               BE STORED ON FILE 10.  THE NUMBER OF SUCH VECTORS TO  BLE02530
C               BE READ IN IS SPECIFIED BY THE PARAMETER KSET.  THE  BLE02540
C               EXISTING CALLING SEQUENCE IS                         BLE02550
C                                                                     BLE02560
C                     CALL GENRAN(IIX,G,K).                          BLE02570
C                                                                     BLE02580
C               WHERE IIX =INTEGER SEED, G = REAL ARRAY WHOSE DIMENSION BLE02590
C               MUST BE >= K.  K PSEUDO-RANDOM NUMBERS ARE GENERATED  BLE02600
C               AND PLACED IN G.                                     BLE02610
C                                                                     BLE02620
C      FINPRO = DOUBLE PRECISION FUNCTION WHICH COMPUTES THE INNER    BLE02630
C               PRODUCT OF 2 DOUBLE PRECISION VECTORS OF DIMENSION N. BLE02640
C               EXISTING CALLING SEQUENCE IS                         BLE02650
C                                                                     BLE02660
C                     CALL FINPRO(N,V,J,W,K).                        BLE02670
C                                                                     BLE02680
C               COMPUTES THE INNER PRODUCT OF DIMENSION N OF THE VECTORS BLE02690
C               V AND W.  SUCCESSIVE COMPONENTS OF V AND OF W ARE STORED BLE02700
```

```
C            AT LOCATIONS THAT ARE ,RESPECTIVELY, J AND K UNITS APART.BLE02710
C                                                                    BLE02720
C     MASK = MASKS OVERFLOW AND UNDERFLOW.  OPTIONAL.                 BLE02730
C            USER MUST SUPPLY OR COMMENT OUT CALL.                    BLE02740
C                                                                    BLE02750
C   USPEC = DIMENSIONS AND INITIALIZES ARRAYS NEEDED TO SPECIFY       BLE02760
C            MATRIX USED BY LANCZS SUBROUTINE.  SEE MATRIX            BLE02770
C            SPECIFICATION SECTION.                                   BLE02780
C                                                                    BLE02790
C   BMATV = CASE (1) ONLY: COMPUTES MATRIX-VECTOR MULTIPLY FOR        BLE02800
C            USER-SUPPLIED A-MATRIX. SEE MATRIX SPECIFICATION SECTION.BLE02810
C                                                                    BLE02820
C   BLSOLV = CASE (2) ONLY:  FOR GIVEN VECTOR V, COMPUTES U SUCH      BLE02830
C            B*U = V, GIVEN THE SPARSE FACTORIZTION OF THE B-MATRIX.  BLE02840
C                                                                    BLE02850
C                                                                    BLE02860
C-----PARAMETER CONTROLS------------------------------------------------BLE02870
C                                                                    BLE02880
C                                                                    BLE02890
C     PARAMETER CONTROLS ARE INTRODUCED TO CONTROL VARIOUS           BLE02900
C     ASPECTS OF THESE PROGRAMS.                                     BLE02910
C                                                                    BLE02920
C     THE FLAG EFLAG SPECIFIES THE NUMBER OF COMPUTATIONAL PHASES.    BLE02930
C                                                                    BLE02940
C     EFLAG  = (0,1)  MEANS                                          BLE02950
C                                                                    BLE02960
C                (0) PROGRAM TERMINATES AFTER COMPLETING PHASE 1      BLE02970
C                    COMPUTATIONS.                                    BLE02980
C                                                                    BLE02990
C                (1) PROGRAM COMPLETES BOTH PHASE 1 AND PHASE 2 OF    BLE03000
C                    THE COMPUTATIONS.                                BLE03010
C                                                                    BLE03020
C     THE FLAG OFLAG CONTROLS THE ORTHOGONALITY CHECKS BETWEEN THE    BLE03030
C     JTH Q-BLOCK GENERATED AND THAT VECTOR IN THE 1ST Q-BLOCK THAT   BLE03040
C     IS GENERATING DESCENDANTS.  FOR SAFETY, OFLAG SHOULD BE 1.      BLE03050
C                                                                    BLE03060
C     OFLAG = (0,1)  MEANS                                           BLE03070
C                                                                    BLE03080
C                (0) NO ORTHOGONALITY CHECKS ARE MADE ON PHASE        BLE03090
C                    1 PORTION OF THE COMPUTATIONS.  ORTHOGONALITY    BLE03100
C                    CHECKS ARE ALWAYS MADE ON PHASE 2 PORTION.       BLE03110
C                                                                    BLE03120
C                (1) PROGRAM CHECKS ORTHOGONALITY OF GENERATED        BLE03130
C                    Q-BLOCKS W.R.T. THAT VECTOR IN THE 1ST Q-BLOCK   BLE03140
C                    THAT IS GENERATING DESCENDANTS IN BOTH PHASE     BLE03150
C                    1 AND PHASE 2 OF THE COMPUTATIONS.               BLE03160
C                                                                    BLE03170
C     THE FLAG IWRITE DETERMINES THE AMOUNT OF OUTPUT TO FILE 6       BLE03180
C     DURING THE COMPUTATIONS                                        BLE03190
C                                                                    BLE03200
C     IWRITE = (0,1)  MEANS                                          BLE03210
C                                                                    BLE03220
C                (0) ABBREVIATED OUTPUT TO FILE 6.                    BLE03230
C                                                                    BLE03240
C                (1) ADDITIONAL COMMENTARY ON THE COMPUTATIONS IS     BLE03250
```

```
C                    PRINTED TO FILE 6.                             BLE03260
C                                                                   BLE03270
C     THE PROGRAM ALWAYS WRITES A LIST OF THE COMPUTED EIGENVALUES  BLE03280
C     AND THE BASIS FOR THE CORRESPONDING EIGENSPACE TO FILE 15,    BLE03290
C     ALONG WITH ESTIMATES OF THE ERRORS IN THESE COMPUTED VALUES.  BLE03300
C                                                                   BLE03310
C-----INPUT/OUTPUT FILES--------------------------------------------BLE03320
C                                                                   BLE03330
C     ANY INPUT DATA OTHER THAN THE A-MATRIX, THE FACTORIZATION     BLE03340
C     OF THE B-MATRIX OR USER-SPECIFIED STARTING VECTORS SHOULD     BLE03350
C     BE STORED ON FILE 5.  SEE SAMPLE INPUT/OUTPUT FROM TYPICAL RUN. BLE03360
C     THE READ STATEMENTS IN THE GIVEN FORTRAN PROGRAM ASSUME THAT  BLE03370
C     THE DATA STORED ON FILE 5 IS IN FREE FORMAT.  USER SHOULD NOTE BLE03380
C     THAT 'FREE FORMAT' IS NOT CLASSIFIED AS PORTABLE BY PFORT SO THAT BLE03390
C     THE USER MAY HAVE TO MODIFY THE READ STATEMENTS FROM FILE 5 TO BLE03400
C     CONFORM TO WHAT IS PERMISSIBLE ON THE COMPUTER BEING USED.    BLE03410
C                                                                   BLE03420
C     FILE 6 WAS USED AS THE INTERACTIVE TERMINAL OUTPUT FILE.      BLE03430
C     THIS FILE PROVIDES A RUNNING ACCOUNT OF THE PROGRESS OF THE   BLE03440
C     COMPUTATIONS.  THE AMOUNT OF INFORMATION PRINTED OUT IS       BLE03450
C     CONTROLLED BY THE PARAMETER IWRITE.                           BLE03460
C                                                                   BLE03470
C DESCRIPTION OF OTHER I/O FILES                                    BLE03480
C                                                                   BLE03490
C FILE (K)     CONTAINS:                                            BLE03500
C                                                                   BLE03510
C     (7)      INPUT FILE:                                          BLE03520
C              USED IN CASE (2).  CONTAINS THE FACTORIZATION        BLE03530
C              OF THE B-MATRIX.                                     BLE03540
C                                                                   BLE03550
C     (8)      INPUT FILE:                                          BLE03560
C              USED IN CASE (1).  CONTAINS THE ARRAYS REQUIRED      BLE03570
C              TO SPECIFY THE A-MATRIX.                             BLE03580
C                                                                   BLE03590
C     (10)     INPUT FILE:                                          BLE03600
C              CONTAINS USER-SUPPLIED STARTING VECTORS, IF ANY.     BLE03610
C              TYPICALLY, THESE WOULD BE 1 OR MORE EIGENVECTOR      BLE03620
C              APPROXIMATIONS OBTAINED DURING AN EARLIER RUN.       BLE03630
C                                                                   BLE03640
C     (13)     OUTPUT FILE:                                         BLE03650
C              CONTAINS EXTRA EIGENVECTOR APPROXIMATIONS THAT       BLE03660
C              WOULD OTHERWISE BE LOST UPON ANY REDUCTION IN THE    BLE03670
C              SIZE OF THE 1ST Q-BLOCK.  IF AT ANY STAGE IN THE     BLE03680
C              BLOCK PROCEDURE, THE SIZE OF THE 1ST Q-BLOCK IS      BLE03690
C              REDUCED FROM KACT TO KACTN, THE Q-VECTORS FROM       BLE03700
C              K = KACTN+1,KACT ARE WRITTEN TO FILE 13 FOR POSSIBLE BLE03710
C              USE AS STARTING VECTORS IN A LATER RUN OF THE        BLE03720
C              BLOCK LANCZOS PROCEDURE.                             BLE03730
C                                                                   BLE03740
C     (15)     OUTPUT FILE:                                         BLE03750
C              CONTAINS COMPUTED EIGENVALUES AND CORRESPONDING      BLE03760
C              COMPUTED EIGENSPACE AVAILABLE AT THE TIME OF         BLE03770
C              TERMINATION OF THE BLOCK LANCZOS PROCEDURE.          BLE03780
C                                                                   BLE03790
C-----PARAMETERS SET BY THE BLOCK PROGRAMS--------------------------BLE03800
```

```
C                                                                  BLE03810
C                                                                  BLE03820
C    SPREC = TOLERANCE USED IN CHECKING ORTHOGONALITY BETWEEN      BLE03830
C            COMPUTED Q-BLOCKS AND THAT VECTOR IN THE FIRST        BLE03840
C            Q-BLOCK THAT IS GENERATING DESCENDANTS.  SEE COMMENTS BLE03850
C            ON OFLAG.                                             BLE03860
C                                                                  BLE03870
C-----USER-SPECIFIED PARAMETERS ---------------------------------------BLE03880
C                                                                  BLE03890
C                                                                  BLE03900
C    FOR BOTH CASES:                                               BLE03910
C                                                                  BLE03920
C    N, MATNO = INTEGERS. SIZE OF USER-SPECIFIED MATRIX AND MATRIX BLE03930
C               IDENTIFICATION NUMBER OF 8 OR FEWER DIGITS.        BLE03940
C                                                                  BLE03950
C    MDIMQ, MDIMTM = INTEGERS.  USER-SPECIFIED DIMENSIONS OF THE   BLE03960
C                    Q-ARRAY AND OF THE TM-ARRAY.  MDIMQ >= N*KMAX BLE03970
C                    AND MDIMTM >= MXBLK**2.                       BLE03980
C                                                                  BLE03990
C    MAXIT,MAXIT2  = INTEGERS.  MAXIMUM NUMBER OF CALLS TO BMATV    BLE04000
C                    (CASE(1)) OR TO BLSOLV (CASE (2)) ALLOWED     BLE04010
C                    RESPECTIVELY, IN PHASE 1 AND IN PHASE 2.      BLE04020
C                                                                  BLE04030
C    RELTOL = DOUBLE PRECISION SCALAR.  RELATIVE TOLERANCE USED    BLE04040
C             TO COMPUTE CONVERGENCE CRITERION FOR PHASE 2 OF      BLE04050
C             THE BLOCK PROCEDURE.                                 BLE04060
C                                                                  BLE04070
C    SEED   = INTEGER.  SEED FOR RANDOM NUMBER GENERATOR.          BLE04080
C             USED IN GENERATION OF STARTING VECTORS FOR           BLE04090
C             THE BLOCK PROCEDURES.                                BLE04100
C                                                                  BLE04110
C    KMAX   = INTEGER.  MXBLK = (KMAX - 1) IS MAXIMUM ALLOWED SIZE BLE04120
C             FOR THE SMALL LANCZOS T-MATRICES.                    BLE04130
C                                                                  BLE04140
C    KM     = INTEGER.  NUMBER OF EIGENVALUES AND EIGENVECTORS     BLE04150
C             TO BE COMPUTED.                                      BLE04160
C                                                                  BLE04170
C    KACT   = INTEGER.  INITIAL NUMBER OF VECTORS IN THE 1ST Q-BLOCK. BLE04180
C             IF THERE IS ANY POSSIBILITY THAT THE KM-TH DESIRED   BLE04190
C             EIGENVALUE IS MULTIPLE, AND THE USER NEEDS TO KNOW   BLE04200
C             THIS, THEN THE USER SHOULD SET KACT > KM.  OTHERWISE, BLE04210
C             THIS PROGRAM WILL NOT BE ABLE TO DETERMINE THAT THAT BLE04220
C             EIGENVALUE IS MULTIPLE UNLESS THE (KM-1)-TH AND KM-TH BLE04230
C             HAPPEN TO BE MULTIPLE.  IF IN FACT, THE KM-TH        BLE04240
C             EIGENVALUE IS MULTIPLE AND THE USER NEEDS A BASIS FOR BLE04250
C             THE CORRESPONDING EIGENSPACE, THEN THE PROCEDURE SHOULD BLE04260
C             BE RERUN WITH THE EXISTING EIGENVECTORS APPROXIMATIONS BLE04270
C             AS STARTING VECTORS AND A LARGER KACT TO GUARANTEE THAT BLE04280
C             A COMPLETE BASIS FOR THAT EIGENSPACE HAS BEEN OBTAINED. BLE04290
C                                                                  BLE04300
C    KSET =  INTEGER.  NUMBER OF STARTING VECTORS SUPPLIED BY THE  BLE04310
C            THE USER.  THESE VECTORS SHOULD BE ON FILE 10.        BLE04320
C                                                                  BLE04330
C                                                                  BLE04340
C    NSTAG = INTEGER.  NUMBER OF THE ITERATION BEYOND WHICH THE    BLE04350
```

```
C               CHANGE IN THE KM-TH RESIDUAL OVER THE PAST 10 ITERATIONS   BLE04360
C               IS MONITORED AND USED AS A MEASURE OF THE RATE OF          BLE04370
C               CONVERGENCE OF THE BLOCK PROCEDURE.                        BLE04380
C                                                                         BLE04390
C     FRACT = DOUBLE PRECISION SCALAR.  EXPECTED OR HOPED FOR             BLE04400
C               FRACTIONAL CHANGE IN THE KM-TH RESIDUAL OVER THE PAST     BLE04410
C               BLOCK LANCZOS ITERATIONS USED TO TEST FOR STAGNATION      BLE04420
C               OF CONVERGENCE.                                           BLE04430
C                                                                         BLE04440
C     NNZ   = DOUBLE PRECISION SCALAR.  AVERAGE NUMBER OF NONZERO         BLE04450
C               ENTRIES PER ROW IN THE MATRIX USED IN THE LANCZOS         BLE04460
C               PROCEDURE.                                                BLE04470
C                                                                         BLE04480
C                                                                         BLE04490
C     AVER  = DOUBLE PRECISION SCALAR.  AVERAGE SIZE OF THE NONZERO       BLE04500
C               ENTRIES IN THE MATRIX USED IN THE LANCZOS PROCEDURE.      BLE04510
C                                                                         BLE04520
C     CASE (2) ONLY:                                                      BLE04530
C                                                                         BLE04540
C     SO, SHIFT = DOUBLE PRECISION SCALARS.  MATRIX USED BY LANCZS        BLE04550
C                 SUBROUTINE IS B = SO*P*A*P' + SHIFT*I  WHERE P          BLE04560
C                 DENOTES A PERMUTATION MATRIX SELECTED TO PRESERVE       BLE04570
C                 THE SPARSITY OF A IN THE FACTORIZATION OF B.            BLE04580
C                 SO AND SHIFT ARE CHOSEN BY THE USER SO THAT THE         BLE04590
C                 DESIRED EIGENVALUES BECOME THE EXTREME EIGENVALUES      BLE04600
C                 OF B-INVERSE.                                           BLE04610
C                                                                         BLE04620
C                                                                         BLE04630
C-----CONVERGENCE TEST-----------------------------------------------BLE04640
C                                                                         BLE04650
C                                                                         BLE04660
C     THE CONVERGENCE TEST INCORPORATED IN THIS PROGRAM IS               BLE04670
C     BASED UPON THE FOLLOWING FACT:  GIVEN A REAL SYMMETRIC             BLE04680
C     MATRIX A, A VECTOR X OF NORM 1, AND A SCALAR EVAL                  BLE04690
C     THEN THERE EXISTS AN EIGENVALUE AEVAL OF A SUCH THAT              BLE04700
C     DABS(AEVAL - EVAL) .LE. NORM(A*X - EVAL*X).  WITHIN                BLE04710
C     EACH ITERATION OF THE BLOCK LANCZOS PROCESS THESE TYPES           BLE04720
C     OF NORMS ARE COMPUTED IN THE PROCESS OF COMPUTING THE             BLE04730
C     2ND Q-BLOCK.                                                      BLE04740
C                                                                         BLE04750
C                                                                         BLE04760
C-----ARRAYS REQUIRED------------------------------------------------BLE04770
C                                                                         BLE04780
C                                                                         BLE04790
C     Q(J)    = DOUBLE PRECISON ARRAY.  ITS DIMENSION MUST BE AT         BLE04800
C               LEAST AS LARGE AS KMAX*N, WHERE N IS THE ORDER OF        BLE04810
C               THE GIVEN MATRIX, AND MXBLK = KMAX - 1 IS THE            BLE04820
C               MAXIMUM SIZE T-MATRIX ALLOWED ON ANY GIVEN              BLE04830
C               ITERATION.  THE COLUMNS OF Q HOLD THE LANCZOS           BLE04840
C               VECTORS GENERATED ON EACH ITERATION OF BLOCK            BLE04850
C               LANCZOS PLUS THERE MUST BE AN ADDITIONAL COLUMN         BLE04860
C               AVAILABLE FOR WORK SPACE.  THE FIRST KACT COLUMNS       BLE04870
C               OF Q CONTAIN THE CURRENT APPROXIMATING EIGENSPACE.      BLE04880
C                                                                         BLE04890
C     E(J)    = DOUBLE PRECISION ARRAY.  ITS DIMENSION MUST BE AT        BLE04900
```

```
C                    LEAST MXBLK = KMAX - 1.  ON EACH ITERATION CONTAINS   BLE04910
C                    THE COMPUTED EIGENVALUES OF THE LANCZOS T-MATRIX.     BLE04920
C                                                                          BLE04930
C      TM(J)   = DOUBLE PRECISION ARRAY.  ITS DIMENSION MUST BE AT         BLE04940
C                    LEAST MXBLK**2 WHERE MXBLK = KMAX - 1.  CONTAINS      BLE04950
C                    THE LANCZOS T-MATRIX GENERATED ON EACH ITERATION      BLE04960
C                    AND THEN THE COMPUTED EIGENVECTORS OF THIS MATRIX.    BLE04970
C                    EISPACK SUBROUTINES ARE USED FOR THE SMALL           BLE04980
C                    EIGENELEMENT COMPUTATIONS.  EISPACK SUBROUTINE        BLE04990
C                    TRED2 IS USED TO REDUCE THE GIVEN T-MATRIX TO         BLE05000
C                    TRIDIAGONAL FORM.  THE EIGENELEMENT PROBLEM FOR THE   BLE05010
C                    TRIDIAGONAL MATRIX IS THEN SOLVED USING THE EISPACK   BLE05020
C                    SUBROUTINE IMTQL2.                                    BLE05030
C                                                                          BLE05040
C      EXPLAN(J) = REAL ARRAY.  ITS DIMENSION IS 20.  THIS ARRAY IS        BLE05050
C                  USED TO ALLOW EXPLANATORY COMMENTS IN THE INPUT FILES.  BLE05060
C                                                                          BLE05070
C      G(J) = REAL ARRAY.  ITS DIMENSION MUST BE >= N.  IT IS USED         BLE05080
C             FOR HOLDING THE PSEUDO-RANDOM NUMBERS USED TO GENERATE       BLE05090
C             ANY STARTING VECTORS NOT SUPPLIED BY THE USER.               BLE05100
C                                                                          BLE05110
C      RESIDL(J),  = DOUBLE PRECISION ARRAYS.  DIMENSION >= MAXIMUM        BLE05120
C      RESIDK(J),    NUMBER OF ITERATIONS ALLOWED.  MAXIMUM IS             BLE05130
C                    CURRENTLY SET TO 100. USED TO MONITOR THE             BLE05140
C                    RATE OF CONVERGENCE.                                  BLE05150
C                                                                          BLE05160
C      TD(J), TOD(J), = DOUBLE PRECISION ARRAYS.  DIMENSION >= MXBLK.      BLE05170
C         SM(J)         WORK SPACES.                                       BLE05180
C                                                                          BLE05190
C      DESC(J), XLFT(J), = INTEGER ARRAYS.  DIMENSION >= MXBLK.            BLE05200
C         LEFT(J)          WORK SPACES.                                    BLE05210
C                                                                          BLE05220
C      DIR(2,J)  = 2-DIMENSIONAL INTEGER ARRAY.  COLUMN DIMENSION >=       BLE05230
C                  MXBLK, ROW DIMENSION 2.  KEEPS TRACK OF NUMBER          BLE05240
C                  OF VECTORS IN EACH QBLOCK.                              BLE05250
C                                                                          BLE05260
C      CASE (2) ONLY:                                                      BLE05270
C                                                                          BLE05280
C      IPR(J), IPT(J) = INTEGER ARRAYS.  EACH OF DIMENSION AT LEAST N.     BLE05290
C                       USED TO STORE THE REORDERING (IF ANY) OF          BLE05300
C                       THE GIVEN MATRIX.                                  BLE05310
C                                                                          BLE05320
C      OTHER ARRAYS                                                        BLE05330
C                                                                          BLE05340
C      THE USER IN THE SUBROUTINE USPEC MUST SPECIFY WHATEVER ARRAYS       BLE05350
C      ARE REQUIRED TO DEFINE THE MATRIX BEING USED BY LANCZS.            BLE05360
C                                                                          BLE05370
C                                                                          BLE05380
C-----SUBROUTINES INCLUDED-----------------------------------------------BLE05390
C                                                                          BLE05400
C                                                                          BLE05410
C      LANCZS = CONTAINS MAJOR LOOP FOR BLOCK LANCZOS PROCEDURES.          BLE05420
C               CALLED FROM MAIN PROGRAM, CALLS SUBROUTINE LANCI1          BLE05430
C               TO GENERATE WITHIN A GIVEN ITERATION THE Q-BLOCKS          BLE05440
C               AND CORRESPONDING LANCZOS T-MATRICES.  THEN CALLS          BLE05450
```

```
C              SUBROUTINE DIAGOM TO COMPUTE THE EIGENELEMENTS        BLE05460
C              OF THE LANCZOS T-MATRIX AND TO MAP THE RELEVANT       BLE05470
C              T-EIGENVECTORS INTO RITZ VECTORS FOR THE A-MATRIX.    BLE05480
C                                                                   BLE05490
C     LANCI1 =  ON EACH ITERATION OF BLOCK LANCZOS COMPUTES          BLE05500
C                Q-SUBBLOCKS.                                        BLE05510
C                                                                   BLE05520
C     DIAGOM = CALLS EISPACK SUBROUTINES TO COMPUTE THE              BLE05530
C              EIGENELEMENTS OF THE SMALL LANCZOS T-MATRICES         BLE05540
C              GENERATED ON EACH ITERATION OF BLOCK LANCZOS.         BLE05550
C              COMPUTES CORRESPONDING RITZ VECTORS FOR A-MATRIX.     BLE05560
C              MONITORS CONVERGENCE OF BLOCK LANCZOS PROCEDURE.      BLE05570
C                                                                   BLE05580
C     START  = GENERATES ANY REQUIRED STARTING VECTORS FOR 1ST       BLE05590
C              Q-BLOCK FOR FIRST ITERATION OF BLOCK LANCZOS.         BLE05600
C                                                                   BLE05610
C     ORTHOG = GIVEN A SET OF Q-VECTORS, Q(J), J = MA,MB,            BLE05620
C              ORTHOGONALIZES THESE VECTORS W.R.T. THE Q-VECTORS     BLE05630
C              Q(J), J = 1,MA-1.                                     BLE05640
C                                                                   BLE05650
C     LPERM  = (USED IN CASE (2) ONLY) GIVEN A MATRIX B AND A        BLE05660
C              PERMUTATION P DEFINED IN THE VECTORS IPR AND IPT,     BLE05670
C              AND A VECTOR X COMPUTE EITHER (P-TRANSPOSE)*X OR PX.   BLE05680
C                                                                   BLE05690
C     CASE (2) ONLY:                                                BLE05700
C     FOR OPTIONAL PRELIMINARY PROCESSING:                          BLE05710
C                                                                   BLE05720
C     PERMUT  (STAND-ALONE PROGRAM):                                BLE05730
C     USES THE NONZERO STRUCTURE OF A GIVEN MATRIX A.               BLE05740
C     CAN BE USED TO OBTAIN A REORDERING OF A THAT WILL PRESERVE    BLE05750
C     THE SPARSENESS OF A UNDER FACTORIZATION.  PERMUT CALLS        BLE05760
C     CALLS THE SPARSPAK PACKAGE, (A. GEORGE, J. LIU, E. NG,        BLE05770
C     U. WATERLOO).  SEE THE PERMUT FORTRAN CODE FOR DETAILS.       BLE05780
C                                                                   BLE05790
C     LORDER (STAND-ALONE PROGRAM) :                                BLE05800
C     GIVEN A MATRIX C IN SPARSE FORMAT AND A PERMUTATION P,        BLE05810
C     COMPUTES THE REORDERED MATRIX B = P*C*P' AND WRITES IT        BLE05820
C     TO FILE 9 IN SPARSE FORMAT.  SEE THE LORDER FORTRAN CODE      BLE05830
C     FOR DETAILS.                                                  BLE05840
C                                                                   BLE05850
C     LFACT (STAND-ALONE PROGRAM) :                                 BLE05860
C     GIVEN A POSITIVE DEFINITE MATRIX B IN SPARSE FORMAT,          BLE05870
C     COMPUTES THE SPARSE CHOLESKY FACTOR L OF B AND WRITES IT      BLE05880
C     TO FILE 7 IN SPARSE FORMAT.  THUS, B = L*L'.                  BLE05890
C     SEE THE LFACT FORTRAN CODE FOR DETAILS.                       BLE05900
C                                                                   BLE05910
C     LTEST (STAND-ALONE MAIN PROGRAM ) :                           BLE05920
C          (USER MUST PROVIDE 3 SUBROUTINES)                        BLE05930
C     GIVEN THE FACTORIZATION OF A SPARSE MATRIX B, COMPUTES        BLE05940
C     THE SOLUTION OF THE EQUATION B*U = B*V1 FOR A KNOWN BUT       BLE05950
C     RANDOMLY-GENERATED VECTOR V1, SOLVING WITH AND WITHOUT ITERATIVE BLE05960
C     REFINEMENT TO OBTAIN A ROUGH CHECK ON THE NUMERICAL CONDITION BLE05970
C     OF THE B-MATRIX.  THIS PROGRAM USES 3 USER-SUPPLIED SUBROUTINES BLE05980
C     CMATV, CMATS AND BLSOLV.  SEE THE LTEST FORTRAN CODE FOR DETAILS. BLE05990
C                                                                   BLE06000
```

```
C                                                                       BLE06010
C-----OTHER PROGRAMS PROVIDED----------------------------------------BLE06020
C                                                                       BLE06030
C                                                                       BLE06040
C     LECOMPAC  =  TRANSLATES A REAL SYMMETRIC MATRIX PROVIDED          BLE06050
C                  IN THE FORMAT  I, J, A(I,J) INTO THE SPARSE          BLE06060
C                  MATRIX FORMAT USED IN THE SAMPLE SUBROUTINES         BLE06070
C                  PROVIDED. IT ASSUMES THAT THE MATRIX                 BLE06080
C                  ENTRIES ARE GIVEN EITHER COLUMN BY COLUMN OR         BLE06090
C                  ROW BY ROW.  THE DATA SET CREATED IS WRITTEN TO      BLE06100
C                  FILE 8.                                              BLE06110
C                                                                       BLE06120
C                                                                       BLE06130
C------------------------------------------------------------------BLE06140
```

8.3   BLEVAL: Main Program, Eigenvalue and Eigenvector Computations

```
C-----BLEVAL (FEW EXTREME EIGENVALUES AND EIGENVECTORS)-----------------BLE00010
C                     (REAL SYMMETRIC MATRICES)                    BLE00020
C  Authors:  Jane Cullum* and Bill Donath**                        BLE00030
C          **IBM Research, T.J. Watson Research Center             BLE00040
C          **Yorktown Heights, N.Y. 10598                          BLE00050
C          * Los Alamos National Laboratory                        BLE00060
C          * Los Alamos, New Mexico 87544                          BLE00065
C             E-mail:  cullumj@lanl.gov                            BLE00070
C                                                                  BLE00080
C  These codes are copyrighted by the authors.  These codes        BLE00090
C  and modifications of them or portions of them are NOT to be      BLE00100
C  incorporated into any commercial codes or used for any other     BLE00110
C  commercial purposes such as consulting for other companies,      BLE00120
C  without legal agreements with the authors of these Codes.        BLE00130
C  If these Codes or portions of them are used in other scientific or  BLE00140
C  engineering research works the names of the authors of these codes  BLE00150
C  and appropriate references to their written work are to be        BLE00160
C  incorporated in the derivative works.                           BLE00170
C                                                                  BLE00180
C  This header is not to be removed from these codes.              BLE00190
C                                                                  BLE00200
C          REFERENCE: Cullum and Willoughby, Chapter 7,           BLE00201
C          Lanczos Algorithms for Large Symmetric Eigenvalue ComputationsBLE00202
C          VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in  BLE00203
C          Applied Mathematics, 2002. SIAM Publications,          BLE00204
C          Philadelphia, PA. USA                                  BLE00205
C                                                                  BLE00206
C                                                                  BLE00210
C     CONTAINS MAIN PROGRAM FOR COMPUTING A FEW OF THE ALGEBRAICALLY-  BLE00220
C     LARGEST EIGENVALUES AND CORRESPONDING EIGENVECTORS OF A REAL    BLE00230
C     SYMMETRIC MATRIX, USING A BLOCK FORM OF LANCZOS TRIDIAGONALIZATIONBLE00240
C     WITH LIMITED REORTHOGONALIZATION.  PROCEDURE IS ITERATIVE.      BLE00250
C     PROCEDURE CAN BE USED TO COMPUTE THE ALGEBRAICALLY-SMALLEST     BLE00260
C     EIGENVALUES BY THE USER SUPPLYING -A*X RATHER THAN A*X, IN      BLE00270
C     WHICH CASE IT COMPUTES THE CORRESPONDING ALGEBRAICALLY-LARGEST  BLE00280
C     EIGENVALUES OF -A.  IN THIS CASE THE SIGNS OF THE COMPUTED      BLE00290
C     EIGENVALUES ARE CHANGED PRIOR TO WRITING TO FILE 15 SO THAT     BLE00300
C     ON EXIT, FILE 15 CONTAINS THE ALGEBRAICALLY-SMALLEST EIGENVALUES BLE00310
C     OF A ALONG WITH THE CORRESPONDING EIGENVECTORS.               BLE00320
C                                                                  BLE00330
C     ITERATIVE 'BLOCK' LANCZOS PROCEDURE FOR WHICH ON EVERY        BLE00340
C     ITERATION, THE 2ND AND SUCCEEDING BLOCKS CONTAIN ONLY ONE     BLE00350
C     VECTOR WHICH IS SELECTED ON THE BASIS OF ITS EXPECTED INFLUENCE BLE00360
C     ON THE CONVERGENCE.   Q-BLOCKS GENERATED ON A GIVEN ITERATION  BLE00370
C     ARE REORTHOGONALIZED ONLY W.R.T. THOSE VECTORS IN THE FIRST    BLE00380
C     Q-BLOCK THAT ARE NOT GENERATING DESCENDANTS ON THAT           BLE00390
C     ITERATION.                                                   BLE00400
C                                                                  BLE00410
C     PFORT VERIFIER IDENTIFIED THE FOLLOWING NONPORTABLE CONSTRUCTIONS:BLE00420
C     1.  DATA MACHEP DEFINITION                                   BLE00430
```

```
C     2.  FORMAT (20A4) USED FOR READING EXPLANATORY COMMENTS.      BLE00440
C     3.  FREE FORMAT (5,*), USED FOR PARAMETER INPUT FROM FILE 5.   BLE00450
C     4.  COMMON/LOOPS/ AS CONSTRUCTED IS NOT PORTABLE              BLE00460
C                                                                    BLE00470
C--------------------------------------------------------------------BLE00480
      DOUBLE PRECISION Q(44000),E(50),TM(2500),TOD(50),TD(50),EPSM,NNZ  BLE00490
      DOUBLE PRECISION SM(100),ERRMAX,SPREC,MACHEP,AVER,RELTOL,ERRMAN   BLE00500
      DOUBLE PRECISION  EVAL, RESIDL(100), RESIDK(100), RESID, FRACT    BLE00510
      REAL EXPLAN(20),G(2000)                                           BLE00520
      INTEGER   DIR(2,100),DESC(100),LEFT(100),XLFT(100)               BLE00530
      INTEGER SEED,OFLAG,EFLAG                                          BLE00540
      COMMON/LOOPS/MAXIT,ITER                                           BLE00550
      COMMON /RANDOM/SEED                                               BLE00560
      COMMON/FLAGS/EFLAG,OFLAG                                          BLE00570
      DOUBLE PRECISION  DABS, DFLOAT                                    BLE00580
C--------------------------------------------------------------------BLE00590
      EXTERNAL BMATV                                                    BLE00600
      DATA MACHEP/Z3410000000000000/                                   BLE00610
C--------------------------------------------------------------------BLE00620
C                                                                    BLE00630
C     ARRAYS MUST DIMENSIONED AS FOLLOWS:                            BLE00640
C                                                                    BLE00650
C     1.  Q:   >=  KMAX*N                                            BLE00660
C     2.  G:   >= N                                                  BLE00670
C     3.  E:   >=  MXBLK                                             BLE00680
C     4.  TM:   >=  MXBLK**2                                         BLE00690
C     5.  TOD, TD, SM, DESC, LEFT, XLFT:   >= MXBLK                  BLE00700
C     6.  DIR: ROW DIMENSION = 2;   COLUMN DIMENSION >= MXBLK        BLE00710
C     7.  RESIDL, RESIDK:  >= MAXIMUM NUMBER OF ITERATIONS ALLOWED.  BLE00720
C         PROGRAM CURRENTLY TERMINATES IF MORE THAN 100 ITERATIONS   BLE00730
C         ARE REQUESTED.  USED TO MONITOR CONVERGENCE.               BLE00740
C     8.  EXPLAN: DIMENSION = 20.                                    BLE00750
C                                                                    BLE00760
C--------------------------------------------------------------------BLE00770
C     OUTPUT HEADER                                                  BLE00780
      WRITE(6,10)                                                      BLE00790
   10 FORMAT(/' BLOCK LANCZOS PROCEDURE FOR REAL SYMMETRIC MATRICES'  BLE00800
     1 /' 2ND AND SUCCEEDING BLOCKS CONTAIN ONLY ONE VECTOR'//)       BLE00810
C                                                                    BLE00820
C     SET PROGRAM PARAMETERS                                         BLE00830
      EPSM = 2.D0*MACHEP                                               BLE00840
      SPREC = 1.D-5                                                    BLE00850
      MPMIN = -1000                                                    BLE00860
C                                                                    BLE00870
C     READ USER-SPECIFIED PARAMETERS FROM INPUT FILE 5 (FREE FORMAT)  BLE00880
C                                                                    BLE00890
C     SELECT THE AMOUNT OF INTERMEDIATE OUTPUT DESIRED (IWRITE =0,1).  BLE00900
C     IWRITE = 1 INCREASES THE AMOUNT OF INTERMEDIATE OUTPUT WRITTEN   BLE00910
C     TO FILE 6 ON EACH ITERATION OF THE BLOCK LANCZOS PROCEDURE.     BLE00920
      READ(5,20) EXPLAN                                                BLE00930
   20 FORMAT(20A4)                                                     BLE00940
      READ(5,*)  IWRITE                                                BLE00950
C                                                                    BLE00960
C     READ ORDER (N) OF MATRIX AND MATRIX IDENTIFICATION NUMBER (MATNO) BLE00970
      READ(5,20) EXPLAN                                                BLE00980
```

```
      READ(5,*) N,MATNO                                        BLE00990
C                                                              BLE01000
C     READ USER-SPECIFIED DIMENSIONS OF Q-ARRAY (MDIMQ) AND OF THE  BLE01010
C     TM-ARRAY (MDIMTM).  READ MAXIMUM NUMBER (MAXIT) OF MATRIX-VECTOR  BLE01020
C     MULTIPLIES ALLOWED IN PHASE 1.                           BLE01030
      READ(5,20) EXPLAN                                        BLE01040
      READ(5,*) MDIMQ, MDIMTM, MAXIT                           BLE01050
C                                                              BLE01060
C     READ FLAGS:   EFLAG = (0,1).  EFLAG = 0, MEANS PROGRAM STOPS  BLE01070
C     AFTER COMPLETING PHASE 1 PORTION OF BLOCK LANCZOS PROCEDURE.  BLE01080
C     EFLAG = 1, MEANS PROGRAM COMPLETES BOTH PHASES BEFORE     BLE01090
C     TERMINATING.                                             BLE01100
C     OFLAG = (0,1).  OFLAG = 0, MEANS THAT IN PHASE 1 PORTION  BLE01110
C     OF THE COMPUTATION, THE PROGRAM DOES NO ORTHOGONALITY CHECKS  BLE01120
C     ON THE Q-BLOCKS GENERATED.  OFLAG = 1 MEANS THAT IN THE   BLE01130
C     PHASE 1 PORTION AND IN THE PHASE 2 PORTIONS OF THE COMPUTATIONS  BLE01140
C     THE PROGRAM CHECKS THE ORTHOGONALITY OF THE Q-BLOCKS GENERATED  BLE01150
C     W.R.T. THAT VECTOR IN THE FIRST BLOCK THAT IS GENERATING  BLE01160
C     DESCENDANTS.  NOTE THAT IN PHASE 2, THE PROGRAM ALWAYS MAKES  BLE01170
C     THIS CHECK OF ORTHOGONALITY REGARDLESS OF THE VALUE OF OFLAG.  BLE01180
C     FOR SAFETY, OFLAG SHOULD ALWAYS BE SET TO 1, ALTHOUGH IN MANY  BLE01190
C     PROBLEMS THIS IS NOT NECESSARY.                          BLE01200
      READ(5,20) EXPLAN                                        BLE01210
      READ(5,*) EFLAG,OFLAG                                    BLE01220
C                                                              BLE01230
C     READ SEED USED BY SUBROUTINE GENRAN TO OBTAIN THOSE STARTING  BLE01240
C     VECTORS WHICH ARE GENERATED RANDOMLY.                    BLE01250
      READ(5,20) EXPLAN                                        BLE01260
      READ(5,*) SEED                                           BLE01270
C                                                              BLE01280
C     SPECIFY MAXIMUM T-SIZE ALLOWED (KMAX-1); INITIAL SIZE OF  BLE01290
C     STARTING BLOCK (KACT);  NUMBER OF STARTING VECTORS SUPPLIED (KSET)BLE01300
C     SEE BLOCK LANCZOS HEADER FOR COMMENTS ON THE SIZE OF KACT.  BLE01310
      READ(5,20) EXPLAN                                        BLE01320
      READ(5,*) KMAX,KACT,KSET                                 BLE01330
C                                                              BLE01340
C     SPECIFY NUMBER OF EXTREME EIGENVALUES AND EIGENVECTORS TO BE  BLE01350
C     COMPUTED (KM).  USER CAN SPECIFY THAT THE ALGEBRAICALLY-  BLE01360
C     SMALLEST EIGENVALUES ARE BEING COMPUTED BY SETTING KM < 0.  BLE01370
C     PROGRAM THEN ASSUMES THAT THE MATRIX-VECTOR MULTIPLY      BLE01380
C     SUBROUTINE WHICH THE USER HAS PROVIDED IS COMPUTING -A*X  BLE01390
C     INSTEAD OF A*X AND INTERNALLY IT COMPUTES THE |KM|        BLE01400
C     ALGEBRAICALLY-LARGEST EIGENVALUES OF -A.                 BLE01410
      READ(5,20) EXPLAN                                        BLE01420
      READ(5,*) KM                                             BLE01430
      IF(KM.EQ.0) GO TO 490                                    BLE01440
      KML = IABS(KM)                                           BLE01450
C                                                              BLE01460
C     STAGNATION OF CONVERGENCE OF THE KM-TH EIGENVALUE WILL BE  BLE01470
C     TESTED AFTER NSTAG ITERATIONS.  CONVERGENCE WILL BE SAID TO  BLE01480
C     HAVE STAGNATED IF THE RATIO OF THE SQUARE OF THE CURRENT KM-TH  BLE01490
C     RESIDUAL TO THE SQUARE OF THE CORRESPONDING RESIDUAL OBTAINED  BLE01500
C     10 ITERATIONS EARLIER IS GREATER THAN FRACT.  NSTAG SHOULD BE  BLE01510
C     >= 25.  IN THE TESTS FRACT WAS SET TO .01.               BLE01520
      READ(5,20) EXPLAN                                        BLE01530
```

```
      READ(5,*) NSTAG, FRACT                                      BLE01540
C                                                                 BLE01550
C     READ IN THE RELATIVE TOLERANCE (RELTOL) USED TO DETERMINE A BLE01560
C     CONVERGENCE CRITERION FOR PHASE 2, AND THE MAXIMUM NUMBER (MAXIT2)BLE01570
C     OF MATRIX-VECTOR MULTIPLIES ALLOWED IN PHASE 2.             BLE01580
      READ(5,20) EXPLAN                                           BLE01590
      IF(EFLAG.EQ.1)  READ(5,*)  RELTOL, MAXIT2                   BLE01600
C                                                                 BLE01610
C     CONSISTENCY CHECKS                                          BLE01620
C     PROCEDURE REQUIRES ENOUGH ROOM IN Q-ARRAY FOR AT LEAST 2    BLE01630
C     BLOCKS OF SIZE KACT PLUS A WORKING VECTOR OF LENGTH N.      BLE01640
      MXBLK = KMAX -1                                             BLE01650
      MXBLK2 = MXBLK*MXBLK                                        BLE01660
      IF(MDIMTM.LT.MXBLK2) GO TO 470                              BLE01670
      NKMAX = N*KMAX                                              BLE01680
      IF(MDIMQ.LT.NKMAX)   GO TO 510                              BLE01690
      IF(KML.GT.KACT) GO TO 370                                   BLE01700
      IF(MXBLK.GT.N)  GO TO 390                                   BLE01710
      IF(2*KACT.GT.MXBLK)  GO TO 450                              BLE01720
C                                                                 BLE01730
C-----------------------------------------------------------------BLE01740
C     DEFINE AND INITIALIZE THE ARRAYS FOR THE USER-SPECIFIED     BLE01750
C     A-MATRIX AND PASS THE STORAGE LOCATIONS OF THESE ARRAYS AND BLE01760
C     OF ANY OTHER PARAMTERS NEEDED TO DEFINE THE MATRIX TO THE   BLE01770
C     MATRIX-VECTOR MULTIPLY SUBROUTINE BMATV.                    BLE01780
C                                                                 BLE01790
      CALL USPEC(N,MATNO,NNZ,AVER)                                BLE01800
C                                                                 BLE01810
C-----------------------------------------------------------------BLE01820
C     MASK OVERFLOW AND UNDERFLOW                                 BLE01830
      CALL MASK                                                   BLE01840
C                                                                 BLE01850
C-----------------------------------------------------------------BLE01860
C     ARE THERE STARTING VECTORS TO READ IN FROM FILE 10 (KSET.NE.0) ?  BLE01870
      IF(KSET.EQ.0) GO TO 70                                      BLE01880
C                                                                 BLE01890
      READ(10,30) NOLD,KACT                                       BLE01900
   30 FORMAT(I6,I4)                                               BLE01910
      IF(NOLD.NE.N.OR.KSET.GT.KACT) GO TO 410                     BLE01920
      DO 50 J=1,KSET                                              BLE01930
      READ(10,20) EXPLAN                                          BLE01940
      READ(10,40) EVAL,RESID                                      BLE01950
   40 FORMAT(E20.12,E13.4)                                        BLE01960
      READ(10,20) EXPLAN                                          BLE01970
      LINT= (J-1)*N + 1                                           BLE01980
      LFIN = J*N                                                  BLE01990
   50 READ(10,60) (Q(JL), JL = LINT,LFIN)                         BLE02000
   60 FORMAT(4E20.12)                                             BLE02010
C                                                                 BLE02020
   70 CONTINUE                                                    BLE02030
C                                                                 BLE02040
C     WRITE TO A SUMMARY OF THE PARAMETERS FOR THIS RUN TO FILE 6 BLE02050
C                                                                 BLE02060
      MXBLK = KMAX - 1                                            BLE02070
      WRITE(6,80)  N, NNZ, AVER, MATNO                            BLE02080
```

```
   80 FORMAT(/6X,'ORDER OF MATRIX ',5X,'AVERAGE NONZEROES PER ROW'/      BLE02090
     1I15,E26.4/4X,'AVERAGE SIZE OF NONZERO ENTRIES',5X,'MATRIX ID'/      BLE02100
     1E25.4,I21/)                                                        BLE02110
C                                                                        BLE02120
      WRITE(6,90) MDIMQ, MDIMTM                                          BLE02130
   90 FORMAT(/18X,'USER-SPECIFIED'/2X,'MAX. DIMENSION Q-ARRAY',4X,'MAX.  BLE02140
     1DIMENSION TM-ARRAY'/I16,I26/)                                      BLE02150
C                                                                        BLE02160
      WRITE(6,100)  OFLAG, EFLAG                                         BLE02170
  100 FORMAT(/4X,'OFLAG',4X,'EFLAG'/I8,I9/)                              BLE02180
C                                                                        BLE02190
      IF(EFLAG.EQ.1)  WRITE(6,110) MAXIT,RELTOL,MAXIT2                   BLE02200
  110 FORMAT(/4X,' MAXIT ',8X,' RELTOL  ',6X,' MAXIT2 '/I10,E20.6,I12/)  BLE02210
      IF(EFLAG.EQ.0)  WRITE(6,120) MAXIT                                 BLE02220
  120 FORMAT(/4X,' MAXIT '/I10/)                                         BLE02230
C                                                                        BLE02240
      WRITE(6,130)  SEED                                                 BLE02250
  130 FORMAT(/' SEED FOR RANDOM NUMBER GENERATOR'/I24/)                  BLE02260
C                                                                        BLE02270
      IF(KM.GT.0)  WRITE(6,140) KML                                      BLE02280
  140 FORMAT(/' COMPUTE THE',I3,' ALGEBRAICALLY-LARGEST EIGENVALUES AND  BLE02290
     1CORRESPONDING VECTORS'/)                                           BLE02300
      IF(KM.LT.0)  WRITE(6,150) KML                                      BLE02310
  150 FORMAT(/' COMPUTE THE',I3,' ALGEBRAICALLY-SMALLEST EIGENVALUES AND BLE02320
     1 CORRESPONDING VECTORS'/' PROGRAM ASSUMES THAT USER IS PROVIDING -BLE02330
     1A*X'/' INSTEAD OF A*X'/' AND COMPUTES THE ALGEBRAICALLY-LARGEST EIGENBLE02340
     1VALUES OF -A.'/' HOWEVER ON EXIT, FILE 15 CONTAINS THE ALGEBRAICALBLE02350
     1LY-SMALLEST EIGENVALUES OF'/' THE ORIGINAL A-MATRIX AND CORRESPONDBLE02360
     1ING EIGENVECTORS.'/)                                               BLE02370
      IF(KM.LT.0) KM = - KM                                              BLE02380
C                                                                        BLE02390
C     COMPUTE PHASE 1 CONVERGENCE TOLERANCE                              BLE02400
      IF(AVER.GE.1.)                                                     BLE02410
     1ERRMAX = 2.D0*DFLOAT(N+1000)*NNZ*AVER*MACHEP                       BLE02420
      IF(AVER.LT.1.)                                                     BLE02430
     1ERRMAX = 2.D0*DFLOAT(N+1000)*NNZ*AVER**2*MACHEP                    BLE02440
C                                                                        BLE02450
      WRITE(6,160) KACT,MXBLK,KSET                                       BLE02460
  160 FORMAT(/' ON INITIAL ITERATIONS, THE FIRST BLOCK CONTAINS ',I3,' VBLE02470
     1ECTORS'/' HOWEVER THE SIZE OF THE FIRST BLOCK MAY CHANGE AS THE ITBLE02480
     1ERATIONS PROCEED'/' THE MAXIMUM SIZE T-MATRIX THAT CAN BE GENERATEBLE02490
     1D IS ',I4/' THE USER SUPPLIED ',I3,' STARTING VECTORS'/)           BLE02500
C                                                                        BLE02510
      WRITE(6,170)                                                       BLE02520
  170 FORMAT(/' ITERATIVE PROCEDURE'/' PROCEDURE MONITORS THE SIZES OF TBLE02530
     1HE NORM(GRADIENTS)**2 ON EACH'/' ITERATION.  CONVERGENCE IS SAID   BLE02540
     1TO HAVE OCCURRED WHEN ALL'/' RELEVANT (NORMS)**2 ARE LESS THAN ERRBLE02550
     1MAX',E10.3/' TYPICALLY, PHASE 1 ERRMAX YIELDS SOMEWHAT LESS THAN'/BLE02560
     1' SINGLE PRECISION ACCURACY.  PHASE 2 REFINES THE VECTORS OBTAINEDBLE02570
     1'/' ON PHASE 1, ACCORDING TO THE ACCURACY SPECIFIED BY THE USER'/)BLE02580
C                                                                        BLE02590
      WRITE(6,180) ERRMAX                                                BLE02600
  180 FORMAT(//' PHASE 1 CONVERGENCE CRITERION, ERRMAX '/E22.3/)         BLE02610
C                                                                        BLE02620
C------------------------------------------------------------------------BLE02630
```

```
C      PASS STORAGE LOCATIONS OF VARIOUS ARRAYS TO LANCZS AND LANCI1   BLE02640
C      SUBROUTINES                                                     BLE02650
C                                                                      BLE02660
       CALL LANZP(DIR,DESC,SM,TM,TOD,TD,G,XLFT,LEFT,SPREC)             BLE02670
       CALL LANCP1(DIR,DESC,TM,SM,XLFT,LEFT)                           BLE02680
C                                                                      BLE02690
C------------------------------------------------------------------BLE02700
C                                                                      BLE02710
C      ENTER PHASE 1 OF BLOCK LANCZOS PROCEDURE.  BLOCK PROCEDURE      BLE02720
C      HAS 2 POSSIBLE PHASES.  USER SPECIFIES PHASE 1 ONLY OR PHASE 1  BLE02730
C      AND PHASE 2 BY SETTING EFLAG = 0 OR 1, RESPECTIVELY. PHASE 1    BLE02740
C      COMPUTES VECTORS THAT MAY BE SOMEWHAT LESS ACCURATE THAN SINGLE BLE02750
C      PRECISION.  PHASE 2 TAKES THE VECTORS OBTAINED IN PHASE 1       BLE02760
C      AND ATTEMPTS TO REFINE THEM.  THE USER SPECIFIES THE DEGREE     BLE02770
C      OF REFINEMENT DESIRED BY SETTING THE VALUES OF RELTOL AND MAXIT2. BLE02780
C      BOTH PHASES SHOULD BE USED.                                     BLE02790
       IPHASE = 1                                                      BLE02800
       NITER = 0                                                       BLE02810
  190 ITER = 0                                                         BLE02820
       RESIDL(1) = FRACT                                               BLE02830
       RESIDL(2) = NSTAG                                               BLE02840
C                                                                      BLE02850
C------------------------------------------------------------------BLE02860
C      CALL INITIATES THE BLOCK LANCZOS PROCEDURE.                     BLE02870
C      ON RETURN EIGENVALUE APPROXIMATIONS ARE IN E(I), I=1,KACT       BLE02880
C      IN ALGEBRAICALLY DECREASING ORDER.   EIGENVECTOR APPROXIMATIONS BLE02890
C      ARE IN FIRST N*KACT LOCATIONS IN THE Q-ARRAY.                   BLE02900
C                                                                      BLE02910
       CALL LANCZS(BMATV,KML,KSET,KACT,MXBLK,N,Q,E,RESIDL,RESIDK,ERRMAX, BLE02920
     1 IPHASE,NITER,IWRITE)                                            BLE02930
C                                                                      BLE02940
C------------------------------------------------------------------BLE02950
C                                                                      BLE02960
       IF(IPHASE.EQ.MPMIN) WRITE(15,200) N,KACT                        BLE02970
  200 FORMAT(2I10,' PHASE 2 TERMINATED '/' PROGRAM INDICATES ACCURACY SPBLE02980
     1ECIFIED BY USER IS NOT ACHIEVABLE'/)                             BLE02990
C                                                                      BLE03000
       ITERA = IABS(ITER)                                              BLE03010
       IF(IWRITE.NE.MPMIN.AND.ITER.GT.0)  WRITE(6,210) IPHASE,ITERA    BLE03020
  210 FORMAT(/1X,'PHASE COMPLETED',5X,' NUMBER MATRIX-VECTOR MULTIPLIES BLE03030
     1USED'/I10,I30)                                                   BLE03040
C                                                                      BLE03050
       IF(IWRITE.EQ.MPMIN.OR.ITER.LT.0)  WRITE(6,220) IPHASE,ITERA     BLE03060
  220 FORMAT(/1X,'PHASE TERMINATED',5X,' NUMBER MATRIX-VECTOR MULTIPLIESBLE03070
     1 USED'/I10,I30)                                                  BLE03080
C                                                                      BLE03090
       IF(ITER.GT.0.AND.IWRITE.NE.MPMIN)  GO TO 250                    BLE03100
C                                                                      BLE03110
       IF(ITER.LT.0)  WRITE(6,230)                                     BLE03120
  230 FORMAT(//' SMALL EIGENVALUE SUBROUTINE DEFAULTED'/' BLOCK LANCZOS BLE03130
     1 PROCEDURE STOPS AFTER SAVING CURRENT EIGENVECTOR APPROXIMATIONS'/BLE03140
     1/)                                                               BLE03150
C                                                                      BLE03160
       WRITE(15,240)                                                   BLE03170
       WRITE(6,240)                                                    BLE03180
```

```
  240 FORMAT(//' BLOCK LANCZOS PROCEDURE TERMINATES WITHOUT CONVERGENCE BLE03190
     1'/' USER SHOULD EXAMINE OUTPUT TO DETERMINE REASONS FOR TERMINATIOBLE03200
     1N'//)                                                             BLE03210
C                                                                       BLE03220
C        WRITE EIGENVALUE AND EIGENVECTOR APPROXIMATIONS CONTAINED IN    BLE03230
C        THE FIRST Q-BLOCK TO FILE 15                                    BLE03240
C                                                                       BLE03250
  250 IF(IPHASE.EQ.1)  WRITE(15,260) N,KACT,SEED                         BLE03260
  260 FORMAT(I6,I4,I12,' PHASE 1, ORDER A-MATRIX, SIZE OF Q(1), SEED')   BLE03270
      IF(IPHASE.EQ.2)  WRITE(15,270) N,KACT,SEED                         BLE03280
  270 FORMAT(I6,I4,I12,' PHASE 2, ORDER A-MATRIX, SIZE OF Q(1), SEED')   BLE03290
C                                                                       BLE03300
      JJ=KACT                                                           BLE03310
      LINT = -N+1                                                       BLE03320
      LFIN = 0                                                          BLE03330
      DO 290 J=1,KACT                                                   BLE03340
      LINT = LINT + N                                                   BLE03350
      LFIN = LFIN + N                                                   BLE03360
      JJ=JJ+1                                                           BLE03370
C                                                                       BLE03380
C        NOTE THAT RESIDUAL PRINTED OUT CORRESPONDS TO VALUE OBTAINED    BLE03390
C        PRIOR TO FINAL PROJECTION Q(1)-TRANSPOSE*AQ(1) DONE BEFORE      BLE03400
C        TERMINATION                                                    BLE03410
C                                                                       BLE03420
      IF(KM.LT.0)   E(J) = -E(J)                                        BLE03430
      WRITE(15,280) E(J), SM(JJ)                                        BLE03440
  280 FORMAT(/E20.12,E13.4,'= EIGENVALUE, NORM(ERROR)**2,EIGENVECTOR='/)BLE03450
  290 WRITE(15,300) (Q(L), L=LINT,LFIN)                                  BLE03460
      WRITE(15,310)                                                     BLE03470
  300 FORMAT(4E20.12)                                                    BLE03480
  310 FORMAT(/' ABOVE ARE COMPUTED APPROXIMATE EIGENVECTORS'/)           BLE03490
C                                                                       BLE03500
      IF(ITER.GT.MAXIT) WRITE(15,320) ITER,MAXIT                        BLE03510
  320 FORMAT(//' PROCEDURE TERMINATED BECAUSE NUMBER OF MATRIX-VECTOR MUBLE03520
     1LTIPLIES ',I6/' EXCEEDED MAXIMUM NUMBER ',I6,' ALLOWED'//)        BLE03530
C                                                                       BLE03540
      IF(ITER.LT.0) WRITE(15,330)                                       BLE03550
  330 FORMAT(//' USER BEWARE.  EIGENELEMENT COMPUTATIONS DEFAULTED BECAUBLE03560
     1SE'/'  EISPACK SUBROUTINE DEFAULTED.  EIGENVALUE AND EIGENVECTORBLE03570
     1 APPROXIMATIONS'/'  ABOVE WERE THOSE AVAILABLE AT THE TIME OF DEFBLE03580
     1AULT'/' SOMETHING IS SERIOUSLY WRONG.'//)                         BLE03590
C                                                                       BLE03600
C        CHECK FOR TERMINATION AFTER PHASE 1                            BLE03610
C        ITER < 0 MEANS EISPACK SUBROUTINE DEFAULTED                    BLE03620
C        IPHASE = MPMIN MEANS THAT PHASE 2 TERMINATED DUE TO ORTHOGONALITY BLE03630
C        IWRITE = MPMIN MEANS THAT CONVERGENCE APPEARS TO HAVE STAGNATED  BLE03640
C        ITER > MAXIT MEANS MAXIMUM NUMBER OF MATRIX-VECTOR MULTIPLIES   BLE03650
C             ALLOWED BY USER WAS EXCEEDED                              BLE03660
      IF(ITER.LT.0.OR.ITER.GT.MAXIT) GO TO 530                          BLE03670
      IF(IPHASE.EQ.MPMIN.OR.IWRITE.EQ.MPMIN) GO TO 530                  BLE03680
      IF(EFLAG.NE.1.OR.IPHASE.EQ.2) GO TO 530                           BLE03690
C                                                                       BLE03700
C        ENTER 2ND PHASE OF COMPUTATION TO ATTEMPT TO OBTAIN MORE        BLE03710
C        ACCURATE EIGENVECTOR APPROXIMATIONS.                           BLE03720
C        USER CONTROLS THE SIZE OF THE ERROR TOLERANCE BY SPECIFYING     BLE03730
```

```
C     THE PARAMETER RELTOL.                                     BLE03740
C                                                               BLE03750
      IPHASE = 2                                                BLE03760
      MAXIT = MAXIT2                                            BLE03770
      KSET = KACT                                               BLE03780
C                                                               BLE03790
C     ERROR TOLERANCE USES THE CONVERGED EIGENVALUE LARGEST IN  BLE03800
C     MAGNITUDE.                                                BLE03810
      TD(1) = DABS(E(1))                                        BLE03820
      IF(KML.EQ.1) GO TO 350                                    BLE03830
      DO 340 J = 2,KML                                          BLE03840
  340 IF(DABS(E(J)).GT.TD(1))    TD(1) = DABS(E(J))             BLE03850
  350 TD(1) = DMAX1(TD(1),1.D0)                                 BLE03860
      ERRMAN = RELTOL**2 * TD(1)**2                             BLE03870
      IF(ERRMAN.GE.ERRMAX) GO TO 430                            BLE03880
      ERRMAX = ERRMAN                                           BLE03890
C                                                               BLE03900
      WRITE(6,360) ERRMAX, MAXIT2                               BLE03910
  360 FORMAT(//' ENTER PHASE 2 OF COMPUTATION'/' CONVERGENCE CRITERION IBLE03920
     1S REDUCED TO ',E13.4/' NO MORE THAN ',I5,' MATRIX VECTOR MULTIPLIEBLE03930
     1S WILL BE ALLOWED.'/' PROGRAM WILL TERMINATE IF BLOCK ORTHGONALITYBLE03940
     1 PROBLEMS MATERIALIZE'/)                                  BLE03950
C                                                               BLE03960
      GO TO 190                                                 BLE03970
C                                                               BLE03980
C     INCONSISTENCIES IN THE DATA                               BLE03990
C                                                               BLE04000
  370 WRITE(6,380) KM,KACT                                      BLE04010
  380 FORMAT(/' PROGRAM TERMINATES BECAUSE THE NUMBER OF EIGENELEMENTS  BLE04020
     1REQUESTED, KM =',I3/'  IS LARGER THAN THE SIZE OF THE FIRST Q BLOCBLE04030
     1K, KACT =',I3,' SPECIFIED'/'  USER MUST RESET KM OR KACT'/)       BLE04040
       GO TO 530                                                BLE04050
C                                                               BLE04060
  390 WRITE(6,400) KMAX,N                                       BLE04070
  400 FORMAT(/' PROGRAM TERMINATES BECAUSE KMAX = ',I5,' IS TOO LARGE FOBLE04080
     1R THE SIZE, N = ',I5,', OF THE GIVEN MATRIX'/'  USER MUST DECREASEBLE04090
     1THE SIZE OF KMAX.'/)                                      BLE04100
       GO TO 530                                                BLE04110
C                                                               BLE04120
  410 WRITE(6,420) NOLD,N,KACT,KSET                             BLE04130
  420 FORMAT(/' PROGRAM TERMINATES BECAUSE FAULT OCCURRED IN READING IN BLE04140
     1THE EIGENVECTOR APPROXIMATIONS'/' EITHER THE SIZE MATRIX SPECIFIEDBLE04150
     1ON THE EIGENVECTOR FILE' ,I6/' DID NOT MATCH THE SIZE SPECIFIED 'BLE04160
     1,I5,' IN THE PROGRAM OR THE NUMBER'/'   OF VECTORS IN FILE 10 = 'BLE04170
     1,I4,' IS LESS THAN THE NUMBER ',I3/'  USER SAID WERE THERE'/)     BLE04180
       GO TO 530                                                BLE04190
C                                                               BLE04200
  430 WRITE(6,440)  ERRMAN, ERRMAX                              BLE04210
  440 FORMAT(/' COMPUTED PHASE 2 CONVERGENCE CRITERION ',E13.4/'  IS LARBLE04220
     1GER THAN PHASE 1 CRITERION ',E13.4/'  SO PROGRAM TERMINATES'/)    BLE04230
       GO TO 530                                                BLE04240
C                                                               BLE04250
  450 WRITE(6,460)  KACT,MXBLK                                  BLE04260
  460 FORMAT(/' PROGRAM TERMINATES BECAUSE THERE IS NOT ENOUGH ROOM TO  BLE04270
     1GENERATE 2 BLOCKS',' BECAUSE KACT = ',I3,' AND MXBLK = ', I4/)    BLE04280
```

```
      GO TO 530                                               BLE04290
C                                                             BLE04300
C                                                             BLE04310
  470 WRITE(6,480)  MDIMTM, MXBLK                             BLE04320
  480 FORMAT(/' PROGRAM TERMINATES BECAUSE THE DIMENSION ',I6,' OF THE TBLE04330
     1M ARRAY'/'   IS TOO SMALL FOR THE LARGEST T-MATRIX ALLOWED ',I4)  BLE04340
      GO TO 530                                               BLE04350
C                                                             BLE04360
  490 WRITE(6,500)                                            BLE04370
  500 FORMAT(/' USER SPECIFIED NUMBER OF EIGENVALUES OF INTEREST AS 0'/'BLE04380
     1 PROGRAM TERMINATES FOR USER TO RESET KM TO DESIRED NONZERO VALUE'BLE04390
     1/)                                                      BLE04400
      GO TO 530                                               BLE04410
C                                                             BLE04420
  510 WRITE(6,520)  MDIMQ, KMAX,N                             BLE04430
  520 FORMAT(/' PROGRAM TERMINATES BECAUSE THE DIMENSION ',I6,' OF THE QBLE04440
     1-ARRAY'/'   IS TOO SMALL TO HOLD ',I5, ' VECTORS OF LENGTH ',I4)  BLE04450
      GO TO 530                                               BLE04460
C                                                             BLE04470
  530 CONTINUE                                                BLE04480
C                                                             BLE04490
      STOP                                                    BLE04500
C-----END OF MAIN PROGRAM FOR BLOCK LANCZOS PROCEDURE------------------BLE04510
      END                                                     BLE04520
```

8.4 BLMULT: Sample Matrix-Vector Multiply Subroutines

```
C-----BLMULT-------------------------------------------------------BLM00010
C  Authors:  Jane Cullum* and Bill Donath**                         BLM00020
C            **IBM Research, T.J. Watson Research Center             BLM00030
C            **Yorktown Heights, N.Y. 10598                          BLM00040
C             * Los Alamos National Laboratory                       BLM00045
C             * Los Alamos, New Mexico 87544                         BLM00050
C               E-mail:  cullumj@lanl.gov                            BLM00060
C                                                                    BLM00070
C  These codes are copyrighted by the authors.  These codes          BLM00080
C  and modifications of them or portions of them are NOT to be       BLM00090
C  incorporated into any commercial codes or used for any other      BLM00100
C  commercial purposes such as consulting for other companies,       BLM00110
C  without legal agreements with the authors of these Codes.         BLM00120
C  If these Codes or portions of them are used in other scientific or BLM00130
C  engineering research works the names of the authors of these codes BLM00140
C  and appropriate references to their written work are to be         BLM00150
C  incorporated in the derivative works.                             BLM00160
C                                                                    BLM00170
C  This header is not to be removed from these codes.                BLM00180
C                                                                    BLM00190
C        REFERENCE: Cullum and Willoughby, Chapter 7,                BLM00191
C        Lanczos Algorithms for Large Symmetric Eigenvalue ComputationsBLM00192
C        VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in BLM00193
C        Applied Mathematics, 2002. SIAM Publications,               BLM00194
C        Philadelphia, PA. USA                                       BLM00195
C                                                                    BLM00196
C                                                                    BLM00200
C     CONTAINS SAMPLE USPEC AND BMATV SUBROUTINES FOR USE WITH       BLM00210
C     THE BLOCK LANCZOS PROCEDURE FOR REAL SYMMETRIC MATRICES.       BLM00220
C     PROGRAMS ARE USED WITH BLEVAL AND BLSUB FILES.                 BLM00230
C                                                                    BLM00240
C     NONPORTABLE CONSTRUCTIONS:                                     BLM00250
C     1.  THE ENTRY MECHANISM USED TO PASS THE STORAGE              BLM00260
C         LOCATIONS OF THE USER-SPECIFIED MATRIX FROM THE            BLM00270
C         SUBROUTINE USPEC TO THE MATRIX-VECTOR SUBROUTINE           BLM00280
C         BMATV.                                                     BLM00290
C     2.  IN THE SAMPLE USPEC AND BMATV SUBROUTINES FOR DIAGONAL     BLM00300
C         TEST MATRICES: FREE FORMAT (8,*) AND THE FORMAT (20A4).    BLM00310
C                                                                    BLM00320
C-----USPEC (GENERAL SYMMETRIC SPARSE MATRICES)---------------------BLM00330
C                                                                    BLM00340
C     SUBROUTINE USPEC(N,MATNO,NNZ,AVER)                             BLM00350
      SUBROUTINE GUSPEC(N,MATNO,NNZ,AVER)                            BLM00360
C                                                                    BLM00370
C-----------------------------------------------------------------BLM00380
      DOUBLE PRECISION  ASD(10000),AD(5010),AVER,NNZ                 BLM00390
      INTEGER  IROW(10000),ICOL(5010)                               BLM00400
C-----------------------------------------------------------------BLM00410
C    USPEC DIMENSIONS AND INITIALIZES THE ARRAYS NEEDED TO DEFINE    BLM00420
C    THE USER-SPECIFIED MATRIX AND THEN PASSES THE STORAGE LOCATIONS BLM00430
C    OF THESE ARRAYS TO THE MULTIPLY SUBROUTINE BMATV.               BLM00440
```

```
C                                                                   BLM00450
C    MATRIX IS STORED IN FOLLOWING SPARSE MATRIX FORMAT:            BLM00460
C    N = ORDER OF A-MATRIX,                                         BLM00470
C    NZS = NUMBER OF NONZERO SUBDIAGONAL ENTRIES,                   BLM00480
C    NZL = INDEX OF LAST COLUMN CONTAINING NONZERO SUBDIAGONAL ENTRIES, BLM00490
C    ICOL(J), J=1,NZL IS THE NUMBER OF NONZERO SUBDIAGONAL ELEMENTS BLM00500
C            IN COLUMN J.                                           BLM00510
C    IROW(K), K = 1,NZS IS THE CORRESPONDING ROW INDEX FOR ASD(K).  BLM00520
C    AD(I), I=1,N CONTAINS DIAGONAL ENTRIES (INCLUDING ANY 0        BLM00530
C          DIAGONAL ENTRIES).                                       BLM00540
C    ASD(K), K=1,NZS CONTAINS NONZERO SUBDIAGONAL ENTRIES, BY COLUMN BLM00550
C    FOR J > NZL THERE ARE NO NONZERO SUBDIAGONAL ELEMENTS IN COLUMN J. BLM00560
C    ICOL(J) = 0 IS ALLOWED                                         BLM00570
C                                                                   BLM00580
C-------------------------------------------------------------------BLM00590
C    ARRAYS THAT DEFINE THE MATRIX ARE READ IN FROM FILE 8          BLM00600
C                                                                   BLM00610
      READ(8,10) NZS,NOLD,NZL,MATOLD                                BLM00620
   10 FORMAT(I10,2I6,I8)                                            BLM00630
C                                                                   BLM00640
      WRITE(6,20) NZS,NOLD,NZL,MATOLD                               BLM00650
   20 FORMAT(I10,2I6,I8,' = NZS,NOLD,NZL,MATOLD'/)                  BLM00660
C                                                                   BLM00670
C    TEST OF PARAMETER CORRECTNESS                                  BLM00680
      ITEMP = (NOLD-N)**2 + (MATNO-MATOLD)**2                       BLM00690
C                                                                   BLM00700
      IF(ITEMP.EQ.0) GO TO 40                                       BLM00710
C                                                                   BLM00720
      WRITE(6,30) NOLD,N,MATOLD,MATNO                               BLM00730
   30 FORMAT(/' PROGRAM TERMINATES BECAUSE EITHER THE SIZE ',I4,' OF THEBLM00740
     1 MATRIX'/' READ FROM FILE 8 DIFFERS FROM THE SIZE ',I4,' SPECIFIEDBLM00750
     1 BY'/' THE USER OR THE MATNO ',I8,' READ IN DIFFERS FROM THE MATNOBLM00760
     1 '/ I8,' SPECIFIED BY THE USER'/)                            BLM00770
      GO TO 100                                                     BLM00780
C                                                                   BLM00790
   40 CONTINUE                                                      BLM00800
C                                                                   BLM00810
C    NUMBER OF NONZERO SUBDIAGONAL ENTRIES IN EACH COLUMN IS READ   BLM00820
C    THEN THE CORRESPONDING ROW INDEX FOR EACH SUCH ENTRY IS READ   BLM00830
      READ(8,50) (ICOL(K), K=1,NZL)                                 BLM00840
      READ(8,50) (IROW(K), K=1,NZS)                                 BLM00850
   50 FORMAT(13I6)                                                  BLM00860
C                                                                   BLM00870
C    DIAGONAL IS READ FIRST, THEN NONZERO BELOW DIAGONAL ENTRIES    BLM00880
      READ(8,60) (AD(K), K=1,N)                                     BLM00890
      READ(8,60) (ASD(K), K=1,NZS)                                  BLM00900
   60 FORMAT(4E19.10)                                               BLM00910
C                                                                   BLM00920
C    COMPUTE NNZ, THE AVERAGE NUMBER OF NONZEROS PER COLUMN, AND    BLM00930
C    AVER, THE AVERAGE SIZE OF NONZERO ENTRIES.                     BLM00940
      ITCOL = 0                                                     BLM00950
      AVER = 0.D0                                                   BLM00960
      DO 70 K = 1,N                                                 BLM00970
      IF(DABS(AD(K)).EQ.0.D0)  GO TO 70                             BLM00980
      ITCOL = ITCOL + 1                                             BLM00990
```

```
      AVER = AVER + DABS(AD(K))                                  BLM01000
   70 CONTINUE                                                   BLM01010
      NTCOL = ITCOL                                             BLM01020
      DO 80 K = 1,N                                             BLM01030
   80 ITCOL = ITCOL + 2*ICOL(K)                                 BLM01040
      NNZ = DFLOAT(ITCOL)/DFLOAT(N)                             BLM01050
      DO 90 K = 1,NZS                                           BLM01060
   90 AVER = AVER + DABS(ASD(K))                                BLM01070
      AVER = AVER/DFLOAT(NZS + NTCOL)                           BLM01080
C                                                               BLM01090
C---------------------------------------------------------------BLM01100
C     PASS STORAGE LOCATIONS OF ARRAYS THAT DEFINE THE MATRIX TO BLM01110
C     THE MATRIX-VECTOR MULTIPLY SUBROUTINE BMATV                BLM01120
C                                                               BLM01130
      CALL BMATVE(ASD,AD,ICOL,IROW,N,NZL)                       BLM01140
C---------------------------------------------------------------BLM01150
C                                                               BLM01160
      RETURN                                                    BLM01170
  100 STOP                                                      BLM01180
C-----END OF USPEC----------------------------------------------BLM01190
      END                                                       BLM01200
C                                                               BLM01210
C-----MATRIX-VECTOR MULTIPLY FOR REAL SPARSE SYMMETRIC MATRICES--------BLM01220
C                                                               BLM01230
C     SUBROUTINE BMATV(W,U)                                     BLM01240
      SUBROUTINE GBMATV(W,U)                                    BLM01250
C                                                               BLM01260
C---------------------------------------------------------------BLM01270
      DOUBLE PRECISION   U(1),W(1),ASD(1),AD(1)                 BLM01280
      INTEGER  IROW(1),ICOL(1)                                  BLM01290
      COMMON/LOOPS/MAXIT,ITER                                   BLM01300
C---------------------------------------------------------------BLM01310
C   SPARSE MATRIX-VECTOR MULTIPLY FOR LANCZS  U = A*W            BLM01320
C   SEE USPEC SUBROUTINE FOR DESCRIPTION OF THE ARRAYS THAT DEFINE BLM01330
C   THE A-MATRIX                                                BLM01340
C---------------------------------------------------------------BLM01350
C                                                               BLM01360
      GO TO 3                                                   BLM01370
C                                                               BLM01380
C---------------------------------------------------------------BLM01390
C   STORAGE LOCATIONS OF ARRAYS ARE PASSED TO BMATV FROM USPEC  BLM01400
C                                                               BLM01410
      ENTRY BMATVE(ASD,AD,ICOL,IROW,N,NZL)                      BLM01420
C---------------------------------------------------------------BLM01430
C                                                               BLM01440
      GO TO 4                                                   BLM01450
C                                                               BLM01460
    3 CONTINUE                                                  BLM01470
C     INCREMENT THE A*W COUNTER                                 BLM01480
      ITER = ITER + 1                                           BLM01490
C     COMPUTE THE DIAGONAL TERMS                                BLM01500
      DO 10 I = 1,N                                             BLM01510
   10 U(I) = AD(I)*W(I)                                         BLM01520
C                                                               BLM01530
C     COMPUTE BY COLUMN                                         BLM01540
```

```
        LLAST = 0                                                BLM01550
        DO 30 J = 1,NZL                                          BLM01560
C                                                                BLM01570
        IF (ICOL(J).EQ.0) GO TO 30                               BLM01580
        LFIRST = LLAST + 1                                       BLM01590
        LLAST = LLAST + ICOL(J)                                  BLM01600
C                                                                BLM01610
        DO 20 L = LFIRST,LLAST                                   BLM01620
        I = IROW(L)                                              BLM01630
C                                                                BLM01640
        U(I) = U(I) + ASD(L)*W(J)                                BLM01650
        U(J) = U(J) + ASD(L)*W(I)                                BLM01660
C                                                                BLM01670
   20 CONTINUE                                                   BLM01680
C                                                                BLM01690
   30 CONTINUE                                                   BLM01700
C                                                                BLM01710
    4 RETURN                                                     BLM01720
C-----END OF BMATV-------------------------------------------------BLM01730
        END                                                      BLM01740
C                                                                BLM01750
C-----MATRIX-VECTOR MULTIPLY FOR DIAGONAL TEST MATRICES-----------BLM01760
C     BMATV COMPUTES  U = (DIAGONAL MATRIX) * W                  BLM01770
C                                                                BLM01780
        SUBROUTINE BMATV(W,U)                                    BLM01790
C       SUBROUTINE DBMATV(W,U)                                   BLM01800
C                                                                BLM01810
C----------------------------------------------------------------BLM01820
        DOUBLE PRECISION  W(1),U(1),D(1)                         BLM01830
        COMMON/LOOPS/MAXIT,ITER                                  BLM01840
C----------------------------------------------------------------BLM01850
        GO TO 3                                                  BLM01860
C----------------------------------------------------------------BLM01870
C   STORAGE LOCATIONS OF ARRAYS ARE PASSED TO BMATV FROM USPEC   BLM01880
        ENTRY MVDIAE(D,N)                                        BLM01890
C----------------------------------------------------------------BLM01900
        GO TO 4                                                  BLM01910
C                                                                BLM01920
    3 CONTINUE                                                   BLM01930
C     INCREMENT THE LOOP COUNTER                                 BLM01940
        ITER = ITER + 1                                          BLM01950
C                                                                BLM01960
        DO 10 I=1,N                                              BLM01970
   10 U(I)= D(I)*W(I)                                            BLM01980
C  10 U(I)= -D(I)*W(I)                                           BLM01990
C                                                                BLM02000
    4 RETURN                                                     BLM02010
C                                                                BLM02020
C-----END OF DIAGONAL TEST MATRIX  MULTIPLY----------------------BLM02030
        END                                                      BLM02040
C                                                                BLM02050
C-----START OF USPEC FOR DIAGONAL TEST MATRIX--------------------BLM02060
C                                                                BLM02070
        SUBROUTINE USPEC(N,MATNO,NNZ,AVER)                       BLM02080
C       SUBROUTINE DUSPEC(N,MATNO,NNZ,AVER)                      BLM02090
```

```
C                                                                  BLM02100
C----------------------------------------------------------------BLM02110
      DOUBLE PRECISION  D(1000),SPACE,SHIFT,AVER,NNZ              BLM02120
      DOUBLE PRECISION  DABS, DFLOAT                              BLM02130
      REAL  EXPLAN(20)                                           BLM02140
C----------------------------------------------------------------BLM02150
C                                                                  BLM02160
      READ(8,10) EXPLAN                                          BLM02170
   10 FORMAT(20A4)                                               BLM02180
      READ(8,*) NOLD,NUNIF,SPACE,D(1),SHIFT                      BLM02190
      NNUNIF = NOLD - NUNIF                                      BLM02200
      WRITE(6,20) NOLD,SPACE,NNUNIF,D(1),SHIFT                   BLM02210
   20 FORMAT(/' DIAGONAL TEST MATRIX, SIZE = ',I4/' MOST ENTRIES ARE ', BLM02220
     1E10.3,' UNITS APART.',I3,' ENTRIES'/' ARE IRREGULARLY SPACED. FIRSBLM02230
     1T ENTRY IS ',E10.3,' SHIFT = ',E10.3/)                    BLM02240
C                                                                  BLM02250
      IF(N.NE.NOLD) GO TO 100                                    BLM02260
C     COMPUTE THE UNIFORM PORTION OF THE SPECTRUM                BLM02270
      DO 30 J=2,NUNIF                                            BLM02280
   30 D(J) = D(1) - DFLOAT(J-1)*SPACE                            BLM02290
      NUNIF1=NUNIF + 1                                           BLM02300
      READ(8,10)  EXPLAN                                         BLM02310
      DO 40 J=NUNIF1,N                                           BLM02320
   40 READ(8,*) D(J)                                             BLM02330
C                                                                  BLM02340
      IF(SHIFT.EQ.0.) GO TO 60                                   BLM02350
      DO 50 J=1,N                                                BLM02360
   50 D(J) = D(J) + SHIFT                                        BLM02370
C                                                                  BLM02380
C     PRINT OUT THE EIGENVALUES OF INTEREST                      BLM02390
   60 WRITE(6,70) (D(I), I=1,10 )                                BLM02400
      NB = NUNIF - 2                                             BLM02410
      WRITE(6,80) (D(I), I = NB,N)                               BLM02420
   70 FORMAT(/' BLOCK LANCZOS TEST, 1ST 10 ENTRIES OF DIAGONAL TEST MATRBLM02430
     1IX = '/(3E22.14))                                          BLM02440
   80 FORMAT(/' MIDDLE UNIFORM PORTION OF MATRIX IS NOT PRINTED OUT'/  BLM02450
     1' END OF UNIFORM PLUS NONUNIFORM SECTION = '/(3E25.16))    BLM02460
C                                                                  BLM02470
C     DIAGONAL GENERATION COMPLETE                               BLM02480
C     COMPUTE NNZ AND AVER                                       BLM02490
      NNZ = 1.D0                                                 BLM02500
      AVER = 0.D0                                                BLM02510
      DO 90 K = 1,N                                              BLM02520
   90 AVER = AVER + DABS(D(K))                                   BLM02530
      AVER = AVER/DFLOAT(N)                                      BLM02540
C                                                                  BLM02550
C----------------------------------------------------------------BLM02560
C   CALL ENTRY TO MATRIX-VECTOR MULTIPLY SUBROUTINE  TO PASS     BLM02570
C   STORAGE LOCATION OF D-ARRAY AND ORDER OF A-MATRIX.          BLM02580
C                                                                  BLM02590
      CALL MVDIAE(D,N)                                           BLM02600
C----------------------------------------------------------------BLM02610
C                                                                  BLM02620
      RETURN                                                     BLM02630
  100 WRITE(6,110) NOLD,N                                        BLM02640
```

```
  110 FORMAT(' PROGRAM TERMINATES BECAUSE NOLD = ',I5,'DOES NOT EQUAL N BLM02650
     1 =',I5)                                                          BLM02660
C-----END OF USPEC SUBROUTINE FOR 'DIAGONAL' TEST MATRICES-------------BLM02670
      STOP                                                             BLM02680
      END                                                             BLM02690
```

8.5   BLSUB: Other Subroutines used by the Codes in Chapters 8 and 9

```
C-----BLSUB----------------------------------------------------------BLS00010
C Authors:  Jane Cullum* and Bill Donath**                           BLS00020
C           **IBM Research, T.J. Watson Research Center              BLS00030
C           **Yorktown Heights, N.Y. 10598                           BLS00040
C           * Los Alamos National Laboratory                         BLS00050
C           * Los Alamos, New Mexico 87544                           BLS00060
C              E-mail:  cullumj@lanl.gov                              BLS00065
C                                                                     BLS00070
C These codes are copyrighted by the authors.  These codes           BLS00080
C and modifications of them or portions of them are NOT to be         BLS00090
C incorporated into any commercial codes or used for any other        BLS00100
C commercial purposes such as consulting for other companies,         BLS00110
C without legal agreements with the authors of these Codes.           BLS00120
C If these Codes or portions of them are used in other scientific or  BLS00130
C engineering research works the names of the authors of these codes  BLS00140
C and appropriate references to their written work are to be          BLS00150
C incorporated in the derivative works.                               BLS00160
C                                                                     BLS00170
C This header is not to be removed from these codes.                  BLS00180
C                                                                     BLS00190
C        REFERENCE: Cullum and Willoughby, Chapter 7,                 BLS00191
C        Lanczos Algorithms for Large Symmetric Eigenvalue ComputationsBLS00192
C        VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in  BLS00193
C        Applied Mathematics, 2002. SIAM Publications,                BLS00194
C        Philadelphia, PA. USA                                        BLS00195
C                                                                     BLS00196
C                                                                     BLS00200
C    PFORT VERIFIER IDENTIFIED THE FOLLOWING NONPORTABLE              BLS00210
C    CONSTRUCTIONS:                                                   BLS00220
C    1. ENTRY MECHANISMS USED TO PASS THE STORAGE LOCATIONS OF        BLS00230
C       SEVERAL ARRAYS FROM THE MAIN PROGRAM TO THE SUBROUTINES       BLS00240
C       LANCZS AND LANCI1.                                            BLS00250
C    2. COMMON BLOCK: LOOPS: USED IN LANCZS AND LANCI1.               BLS00260
C                                                                     BLS00270
C    SUBROUTINES:    LANCZS, LANCI1, ORTHOG, START, AND DIAGOM        BLS00280
C                    ARE USED WITH THE BLOCK LANCZOS PROGRAMS         BLS00290
C                    BLEVAL AND BLIEVAL.  LPERM IS USED WITH BLIEVAL. BLS00300
C                                                                     BLS00310
C                                                                     BLS00320
C-----LANCZS FOR BLOCK LANCZOS PROCEDURE-----------------------------BLS00330
C                                                                     BLS00340
C    ON EACH ITERATION CALLS LANCI1 SUBROUTINE TO GENERATE            BLS00350
C    THE Q-SUBBLOCKS AND THEN CALLS DIAGOM SUBROUTINE TO              BLS00360
C    DIAGONALIZE THE SMALL SYMMETRIC MATRIX WHICH IS THE PROJECTION   BLS00370
C    OF THE MATRIX BEING USED BY LANCZS ONTO THE SUBSPACE SPANNED     BLS00380
C    BY THESE Q-BLOCKS.                                               BLS00390
C                                                                     BLS00400
C        SUBROUTINE LANCZS(MATVEC,KML,KSET,KACT,MXBLK,N,Q,E,RESIDL,   BLS00410
C       1 RESIDK,ERRMAX,IPHASE,NITER,IWRITE)                          BLS00420
C                                                                     BLS00430
```

```
C-------------------------------------------------------------------BLS00440
      DOUBLE PRECISION  E(1),Q(1),ERRMAX,SPREC,RESN,FRACT,RKM,SUM     BLS00450
      DOUBLE PRECISION  TM(1),SM(1),TD(*),TOD(1),RESIDL(*),RESIDK(1)  BLS00460
      REAL  G(1)                                                      BLS00470
      INTEGER  EFLAG,OFLAG,DIR(2,*),DESC(1),LEFT(1),XLFT(*)           BLS00480
      DOUBLE PRECISION FINPRO                                         BLS00490
      COMMON /LOOPS/MAXIT,ITER                                        BLS00500
      COMMON/FLAGS/EFLAG,OFLAG                                        BLS00510
      EXTERNAL MATVEC                                                 BLS00520
C-------------------------------------------------------------------BLS00530
      GO TO 3                                                         BLS00540
C-------------------------------------------------------------------BLS00550
C     ENTRY RECEIVES STORAGE LOCATIONS OF SEVERAL OF THE ARRAYS       BLS00560
C     USED BY THE LANCZS SUBROUTINE.  THIS ALLOWS USER TO SPECIFY     BLS00570
C     THE DIMENSIONS OF THESE ARRAYS IN THE MAIN PROGRAM.             BLS00580
C                                                                     BLS00590
      ENTRY LANZP(DIR,DESC,SM,TM,TOD,TD,G,XLFT,LEFT,SPREC)            BLS00600
      GO TO 4                                                         BLS00610
C-------------------------------------------------------------------BLS00620
C                                                                     BLS00630
    3 CONTINUE                                                        BLS00640
C                                                                     BLS00650
      KM = KML                                                        BLS00660
      MMT = MXBLK*MXBLK                                               BLS00670
      MPMIN = -1000                                                   BLS00680
      IKACT = KACT + 10                                               BLS00690
      FRACT = RESIDL(1)                                               BLS00700
      NSTAG = RESIDL(2)                                               BLS00710
      IORTHO = 0                                                      BLS00720
C                                                                     BLS00730
C     CONSTRUCT STARTING VECTORS                                      BLS00740
      IF(KSET.EQ.0) GO TO 10                                          BLS00750
C-------------------------------------------------------------------BLS00760
      CALL ORTHOG(1,KSET,N,Q)                                         BLS00770
   10 CALL START (KSET+1,KACT,N,Q,G,ERRMAX)                           BLS00780
C-------------------------------------------------------------------BLS00790
   20 CONTINUE                                                        BLS00800
C     INITIALIZE THE LANCZOS T-MATRIX.                                BLS00810
      DO 30 J=1,MMT                                                   BLS00820
   30 TM(J)=0.D0                                                      BLS00830
C                                                                     BLS00840
C     INITIALIZE THE Q-BLOCK DIRECTORY                                BLS00850
      DIR(1,1)=1                                                      BLS00860
      DIR(2,1)=KACT                                                   BLS00870
C                                                                     BLS00880
C     ORTHOGONALIZE THE STARTING VECTORS                             BLS00890
      IF(NITER.EQ.0) GO TO 40                                         BLS00900
C-------------------------------------------------------------------BLS00910
      CALL ORTHOG(1,KACT,N,Q)                                         BLS00920
C-------------------------------------------------------------------BLS00930
   40 CONTINUE                                                        BLS00940
C                                                                     BLS00950
C     GENERATE THE QSUBBLOCKS USED ON ITERATION NITER AND STORE IN    BLS00960
C     THE Q-ARRAY                                                     BLS00970
C                                                                     BLS00980
```

```
      DO 90 I=1,MXBLK                                             BLS00990
C                                                                 BLS01000
C---------------------------------------------------------------BLS01010
      CALL LANCI1(MATVEC,MXBLK,NITER,I,N,Q,KACT,KML,ERRMAX,RESN,RKM,   BLS01020
     1 IND,KACTN,IWRITE)                                          BLS01030
C---------------------------------------------------------------BLS01040
C                                                                 BLS01050
C     HAS CONVERGENCE OCCURRED?                                  BLS01060
      II = I+1                                                    BLS01070
      IF (I.EQ.1.AND.DIR(2,I).EQ.DIR(2,II)) GO TO 140            BLS01080
C                                                                 BLS01090
C     WAS THERE ROOM FOR ANOTHER Q-BLOCK?                        BLS01100
      IF (DIR(2,II).LT.DIR(1,II))  GO TO 100                     BLS01110
C                                                                 BLS01120
C     IF OFLAG = 1 OR IPHASE = 2, CHECK THE ORTHOGONALITY OF     BLS01130
C     THE Q-SUBBLOCKS GENERATED WITH RESPECT TO THAT VECTOR      BLS01140
C     IN THE 1ST Q-BLOCK WHICH IS GENERATING DESCENDANTS.        BLS01150
C     IN PHASE 2 LOSSES IN ORTHOGONALITY ARE USED TO            BLS01160
C     DETERMINE WHEN THE LIMITS ON THE ACHIEVABLE ACCURACY HAVE  BLS01170
C     BEEN REACHED.                                              BLS01180
C                                                                 BLS01190
      IF(OFLAG.EQ.0.AND.IPHASE.EQ.1) GO TO 90                    BLS01200
C                                                                 BLS01210
      L1=DIR(1,II)                                                BLS01220
      LL1 = (L1-1)*N + 1                                         BLS01230
      IND1 = (IND-1)*N + 1                                       BLS01240
C---------------------------------------------------------------BLS01250
      SUM = FINPRO(N,Q(IND1),1,Q(LL1),1)                         BLS01260
C---------------------------------------------------------------BLS01270
C                                                                 BLS01280
      IF(DABS(SUM).LT.SPREC)GO  TO 80                            BLS01290
C                                                                 BLS01300
      IF(IWRITE.EQ.1) WRITE(6,50)  IND,L1,SUM,I                  BLS01310
   50 FORMAT(/' INNER PRODUCT OF VECTORS ',I3,' AND ',I3,' = ',E13.3/  BLS01320
     1' THIS VIOLATES ORTHOGONALITY TEST.  TERMINATE BLOCK GENERATION'  BLS01330
     1/' WITH ',I3,'TH BLOCK ')                                  BLS01340
C                                                                 BLS01350
C     ORTHOGONALITY TEST VIOLATED, TERMINATE BLOCK GENERATION    BLS01360
C     FOR THIS ITERATION.  IN PHASE 2 KEEP TRACK OF NUMBER OF    BLS01370
C     SUCH VIOLATIONS THAT LIMIT THE NUMBER OF BLOCKS TO < 10.   BLS01380
C     TERMINATE AFTER 3 SUCH VIOLATIONS IN PHASE 2.              BLS01390
      IF(IPHASE.NE.1.AND.I.LT.IKACT)  IORTHO = IORTHO + 1        BLS01400
      IF(IORTHO.LT.3.AND.II.NE.2)  GO TO 70                      BLS01410
      WRITE(6,60)                                                 BLS01420
   60 FORMAT(/' THE ORTHOGONALITY TEST HAS FAILED THREE TIMES'/  BLS01430
     1' TERMINATE THE BLOCK PROCEDURE'/)                         BLS01440
      IPHASE = -1000                                             BLS01450
C     BEFORE TERMINATING WRITE THE CURRENT EIGENVECTOR/EIGENVALUE  BLS01460
C     APPROXIMATIONS TO FILE 15                                  BLS01470
      GO TO 160                                                  BLS01480
C                                                                 BLS01490
C     TERMINATE THE Q-BLOCK GENERATION ON THIS ITERATION         BLS01500
   70 DIR(2,II)=DIR(2,I)                                         BLS01510
      GO TO 100                                                  BLS01520
C                                                                 BLS01530
```

```
    80 CONTINUE                                                       BLS01540
C                                                                     BLS01550
C     END OF ORTHOGONALITY TESTS                                      BLS01560
C                                                                     BLS01570
    90 CONTINUE                                                       BLS01580
C                                                                     BLS01590
C     END OF RECURSIVE Q-BLOCK GENERATION                             BLS01600
C                                                                     BLS01610
   100 CONTINUE                                                       BLS01620
       MM = DIR(2,II)                                                 BLS01630
       IF(IWRITE.EQ.1)  WRITE (6,110)  MM,I                          BLS01640
   110 FORMAT(' T-MATRIX IS OF ORDER ',I3, ' NUMBER OF BLOCKS = ',I3) BLS01650
C                                                                     BLS01660
C---------------------------------------------------------------------BLS01670
C     DIAGONALIZE THE PROJECTION MATRIX TM.  ON RETURN THE            BLS01680
C     UPDATED APPROXIMATIONS TO THE DESIRED EIGENVECTORS ARE IN THE   BLS01690
C     FIRST KACT COLUMNS OF THE Q-ARRAY.                              BLS01700
C     UPDATED EIGENVALUE APPROXIMATIONS ARE IN E.                     BLS01710
       TD(1) = RKM                                                    BLS01720
       TD(2) = FRACT                                                  BLS01730
       IERR = NSTAG                                                   BLS01740
C                                                                     BLS01750
       CALL DIAGOM(MXBLK,MM,TM,KACT,N,Q,E,RESIDL,RESIDK,              BLS01760
      1 RESN,IND,KACTN,KM,TD,TOD,NITER,IERR,IWRITE)                   BLS01770
C---------------------------------------------------------------------BLS01780
C                                                                     BLS01790
C     INCREMENT COUNTER FOR NUMBER OF BLOCK LANCZOS ITERATIONS        BLS01800
       NITER = NITER + 1                                              BLS01810
C     IWRITE = MPMIN MEANS BLOCK LANCZOS PROCEDURE TERMINATED ABNORMALLYBLS01820
       IF(IWRITE.EQ.MPMIN)  GO TO 160                                 BLS01830
C     IERR .NE. 0 MEANS EISPACK SUBROUTINE DEFAULTED                  BLS01840
       IF(IERR.EQ.0) GO TO 130                                        BLS01850
       WRITE(6,120)                                                   BLS01860
   120 FORMAT(/' EISPACK SIGNALS TROUBLE IN SMALL IMTQL2 EIGENVALUE SUBROBLS01870
      1UTINE,'/' SO BLOCK LANCZOS PROGRAM TERMINATES'/)               BLS01880
       ITER = -ITER                                                   BLS01890
C                                                                     BLS01900
       RETURN                                                         BLS01910
C                                                                     BLS01920
   130 IF (ITER.GE.MAXIT) GO TO 160                                   BLS01930
C                                                                     BLS01940
C     UPDATED APPROXIMATIONS WERE OBTAINED WITHOUT EXCEEDING          BLS01950
C     MAXIMUM NUMBER OF MATRIX-VECTOR MULTIPLIES SET BY THE USER.     BLS01960
C     CONTINUE BLOCK LANCZOS LOOP ITERATIONS                          BLS01970
C                                                                     BLS01980
       GO TO 20                                                       BLS01990
C                                                                     BLS02000
   140 WRITE(6,150)                                                   BLS02010
   150 FORMAT(//' BLOCK LANCZOS PROCEDURE CONVERGED'//)               BLS02020
C                                                                     BLS02030
C     BLOCK LANCZOS PROCEDURE HAS CONVERGED.                          BLS02040
C     ATTEMPT TO IMPROVE THE APPROXIMATE EIGENVECTORS BY DIAGONALIZING BLS02050
C     THE SMALL PROJECTION MATRIX OBTAINED BY USING ONLY THE          BLS02060
C     FIRST BLOCK IN Q-ARRAY.                                         BLS02070
C                                                                     BLS02080
```

```
    160 KACT2 = KACT*MXBLK                                          BLS02090
        DO 170 KK = 1,KACT2                                         BLS02100
    170 TM(KK) = 0.D0                                               BLS02110
C------------------------------------------------------------------BLS02120
        CALL ORTHOG(1,KACT,N,Q)                                    BLS02130
C------------------------------------------------------------------BLS02140
        KK0 = 1-N                                                  BLS02150
        KACTP1 = (KACT)*N + 1                                      BLS02160
        JJ0 = -MXBLK-1                                             BLS02170
        DO 190 K=1,KACT                                            BLS02180
        JJ0 = JJ0 + MXBLK + 1                                      BLS02190
        KK0 = KK0 + N                                              BLS02200
C------------------------------------------------------------------BLS02210
        CALL MATVEC(Q(KK0),Q(KACTP1))                             BLS02220
C------------------------------------------------------------------BLS02230
        LL0 = (K-2)*N + 1                                         BLS02240
        JJ = JJ0                                                  BLS02250
        DO 180 L=K,KACT                                           BLS02260
        LL0 = LL0 + N                                             BLS02270
        JJ=JJ+1                                                   BLS02280
C------------------------------------------------------------------BLS02290
        TM(JJ) = FINPRO(N,Q(LL0),1,Q(KACTP1),1)                   BLS02300
C------------------------------------------------------------------BLS02310
    180 CONTINUE                                                   BLS02320
C                                                                  BLS02330
    190 CONTINUE                                                   BLS02340
C                                                                  BLS02350
C------------------------------------------------------------------BLS02360
C     USE EISPACK SUBROUTINE TRED2 TO TRIDIAGONALIZE TM-MATRIX    BLS02370
C     TM = (1ST Q-BLOCK)-TRANSPOSE*A*(1ST Q-BLOCK).               BLS02380
C     ON RETURN DIAGONAL ELEMENTS COMPUTED ARE IN TD, OFF-DIAGONAL BLS02390
C     ELEMENTS ARE IN TOD, TRANSFORMATIONS USED ARE IN TM.        BLS02400
C     THEN USE EISPACK SUBROUTINE IMTQL2 TO DIAGONALIZE THE T-MATRIX. BLS02410
C     ON RETURN.  EIGENVALUES ARE IN TD IN ASCENDING ORDER.       BLS02420
C     CORRESPONDING EIGENVECTORS ARE IN TM.                       BLS02430
C                                                                  BLS02440
        CALL TRED2(MXBLK,KACT,TM,TD,TOD,TM)                       BLS02450
        CALL IMTQL2(MXBLK,KACT,TD,TOD,TM,IERR)                    BLS02460
C------------------------------------------------------------------BLS02470
C                                                                  BLS02480
        IF(IERR.EQ.0)  GO TO 200                                  BLS02490
        WRITE(6,120)                                              BLS02500
        ITER = -ITER                                              BLS02510
C                                                                  BLS02520
        RETURN                                                    BLS02530
C                                                                  BLS02540
C     COMPUTE SUCCESSIVELY THE JTH-COMPONENTS OF THE RITZ VECTORS. BLS02550
C     REORDER THE EIGENVALUES (AND EIGENVECTORS) SO THAT THEY     BLS02560
C     ARE IN ALGEBRAICALLY DECREASING ORDER.                     BLS02570
C                                                                  BLS02580
    200 DO 220 J=1,N                                              BLS02590
        JJ0 = - MXBLK                                             BLS02600
        JL0 = -N + J                                              BLS02610
        DO 210 K=1,KACT                                           BLS02620
        TOD(K)=0.D0                                               BLS02630
```

```
          JJ0 = JJ0 + MXBLK                                        BLS02640
          JJ= JJ0                                                  BLS02650
          JL = JL0                                                 BLS02660
          DO 210 L=1,KACT                                          BLS02670
          JJ=JJ+1                                                  BLS02680
          JL = JL + N                                              BLS02690
      210 TOD(K)=TOD(K)+TM(JJ)*Q(JL)                               BLS02700
          JK = JL0                                                 BLS02710
          DO 220 K=1,KACT                                          BLS02720
          JK = JK + N                                              BLS02730
          KACTK = KACT - K + 1                                     BLS02740
          Q(JK)=TOD(KACTK)                                         BLS02750
      220 CONTINUE                                                 BLS02760
          DO 230 K=1,KACT                                          BLS02770
          KACTK = KACT - K + 1                                     BLS02780
      230 E(K)=TD(KACTK)                                           BLS02790
C                                                                  BLS02800
C     HAS CONVERGENCE OCCURRED?                                    BLS02810
          IF(I.EQ.1.AND.DIR(2,I).EQ.DIR(2,I+1))  GO TO 250         BLS02820
C                                                                  BLS02830
C     CONVERGENCE HAS NOT OCCURRED, PROCEDURE TERMINATED FOR SOME  BLS02840
C     OTHER REASON                                                 BLS02850
          WRITE(6,240)                                             BLS02860
      240 FORMAT(//' BLOCK LANCZOS PROCEDURE TERMINATES WITHOUT CONVERGENCE'BLS02870
         1/' AFTER WRITING THE CURRENT EIGENVALUE AND EIGENVECTOR APPROXIMATBLS02880
         1IONS'/' TO FILE 15'/)                                    BLS02890
C                                                                  BLS02900
          RETURN                                                   BLS02910
C                                                                  BLS02920
      250 IF(IPHASE.EQ.1) WRITE(6,260) (E(K), K=1,KACT)            BLS02930
          IF(IPHASE.EQ.2) WRITE(6,270) (E(K), K=1,KACT)            BLS02940
      260 FORMAT(/' AT END OF PHASE 1, COMPUTED EIGENVALUES ='/(4E20.12))  BLS02950
      270 FORMAT(/' AT END OF PHASE 2, COMPUTED EIGENVALUES ='/(4E20.12))  BLS02960
C                                                                  BLS02970
C                                                                  BLS02980
C-----END OF LANCZS-------------------------------------------------------BLS02990
        4 RETURN                                                   BLS03000
          END                                                      BLS03010
C                                                                  BLS03020
C-----START OF LANCI1-----------------------------------------------------BLS03030
C     GENERATES THE Q-SUBBLOCKS ON EACH ITERATION OF THE BLOCK LANCZOS  BLS03040
C     PROCEDURE.                                                   BLS03050
C                                                                  BLS03060
          SUBROUTINE LANCI1(MATVEC,MXBLK,NITER,I,N,Q,KACT,KML,ERRMAX,  BLS03070
         1RESN,RKM,IND,KACTN,IWRITE)                               BLS03080
C                                                                  BLS03090
C------------------------------------------------------------------------BLS03100
          DOUBLE PRECISION  Q(1),TM(1),S,SM(1),T,ERRMAX,SUM,RESN,RKM  BLS03110
          INTEGER  DIR(2,*),DESC(1),LEFT(1),XLFT(*)                BLS03120
          DOUBLE PRECISION FINPRO, DSQRT                           BLS03130
          EXTERNAL MATVEC                                          BLS03140
C------------------------------------------------------------------------BLS03150
          GO TO 3                                                  BLS03160
C------------------------------------------------------------------------BLS03170
C     ALLOWS PASSAGE OF LOCATIONS OF SOME OF THE ARRAYS USED BY LANCI1  BLS03180
```

```
C      SO THAT THESE ARRAYS CAN BE DIMENSIONED IN THE MAIN PROGRAM    BLS03190
C                                                                     BLS03200
       ENTRY LANCP1(DIR,DESC,TM,SM,XLFT,LEFT)                         BLS03210
       GO TO 4                                                        BLS03220
C----------------------------------------------------------------BLS03230
     3 CONTINUE                                                       BLS03240
C                                                                     BLS03250
C      SIZE OF FIRST BLOCK CAN CHANGE.                                BLS03260
       IF(I.EQ.1) KACTN = KACT                                        BLS03270
C                                                                     BLS03280
C      XLFT(I+2) IS CUMULATIVE TOTAL OF VECTORS IN 1ST QBLOCK NOT     BLS03290
C      GENERATING DESCENDANTS.                                        BLS03300
C                                                                     BLS03310
       IF(I.GT.1) GO TO 10                                            BLS03320
       XLFT(1) = 0                                                    BLS03330
       XLFT(2) = 0                                                    BLS03340
    10 XLFT(I+2) = XLFT(I+1)                                          BLS03350
C                                                                     BLS03360
C      INITIALIZE THE DIRECTORY FOR NEXT QBLOCK Q(I+1)                BLS03370
C                                                                     BLS03380
       I2=DIR(2,I)                                                    BLS03390
       I1=DIR(1,I)                                                    BLS03400
       DIR(1,I+1)=I2+1                                                BLS03410
       DIR(2,I+1)=I2                                                  BLS03420
C                                                                     BLS03430
C      IS THERE ROOM FOR ANOTHER QBLOCK?                              BLS03440
C                                                                     BLS03450
       MS = I2-I1+1                                                   BLS03460
       IF (MS+I2.LE.MXBLK) GO TO 70                                   BLS03470
C                                                                     BLS03480
C      NOT ENOUGH ROOM TO GENERATE ANOTHER BLOCK                      BLS03490
C      COMPLETE THE TM-MATRIX. NOTE THAT THE TM-MATRIX IS             BLS03500
C      DIMENSIONED AS (MXBLK,1) AND THE EISPACK SUBROUTINES           BLS03510
C      REQUIRE THE LOWER TRIANGULAR PART OF THIS MATRIX.              BLS03520
C                                                                     BLS03530
       I3=I2+1                                                        BLS03540
       JI30 = (I3-1)*N                                                BLS03550
       JI31 = JI30 + 1                                                BLS03560
       JK1 = (I1-2)*N + 1                                             BLS03570
       DO 60 K=I1,I2                                                  BLS03580
       JK1 = JK1 + N                                                  BLS03590
C----------------------------------------------------------------BLS03600
       CALL MATVEC(Q(JK1),Q(JI31))                                   BLS03610
C----------------------------------------------------------------BLS03620
C      COMPUTE LAST DIAGONAL BLOCK IN TM-MATRIX FOR THIS ITERATION    BLS03630
C                                                                     BLS03640
       JL1 = (K-2)*N + 1                                              BLS03650
       KK = (K-1)*MXBLK +  K - 1                                      BLS03660
    20 DO 30 L=K,I2                                                   BLS03670
       KK = KK + 1                                                    BLS03680
       JL1 = JL1 + N                                                  BLS03690
C----------------------------------------------------------------BLS03700
       TM(KK) = FINPRO(N,Q(JL1),1,Q(JI31),1)                         BLS03710
C----------------------------------------------------------------BLS03720
    30 CONTINUE                                                       BLS03730
```

```
C                                                                          BLS03740
C       COMPUTE ASSOCIATED CORRECTION TERMS IN TM-MATRIX.                  BLS03750
        IF(XLFT(I).EQ.0) GO TO 50                                          BLS03760
        LUP = XLFT(I)                                                      BLS03770
        DO 40 JJ = 1,LUP                                                   BLS03780
        L= LEFT(JJ)                                                        BLS03790
        JL1 = (L-1)*N + 1                                                  BLS03800
C------------------------------------------------------------------------BLS03810
        SUM = FINPRO(N,Q(JI31),1,Q(JL1),1)                                BLS03820
C------------------------------------------------------------------------BLS03830
        KK = (L-1)*MXBLK + K                                               BLS03840
        TM(KK) = SUM + TM(KK)                                              BLS03850
     40 CONTINUE                                                           BLS03860
C                                                                          BLS03870
     50 CONTINUE                                                           BLS03880
C                                                                          BLS03890
     60 CONTINUE                                                           BLS03900
C                                                                          BLS03910
        RETURN                                                             BLS03920
C                                                                          BLS03930
C       ON EVERY BLOCK PASS THROUGH HERE TO GENERATE THE ITH-BLOCK         BLS03940
C       DIAGONAL ENTRY A(I) OF THE TM-MATRIX, EXCEPT THE LAST DIAGONAL     BLS03950
C       BLOCK WHICH IS GENERATED ABOVE                                     BLS03960
C                                                                          BLS03970
     70 CONTINUE                                                           BLS03980
C       COMPUTE (A-MATRIX)*(ITH-Q-BLOCK)                                   BLS03990
        KA=I2                                                              BLS04000
        DO 80 K=I1,I2                                                      BLS04010
        KA=KA+1                                                            BLS04020
        JKA1 = (KA-1)*N + 1                                                BLS04030
        JK1 = (K-1)*N + 1                                                  BLS04040
C------------------------------------------------------------------------BLS04050
        CALL MATVEC(Q(JK1),Q(JKA1))                                       BLS04060
C------------------------------------------------------------------------BLS04070
        DESC(K)=KA                                                         BLS04080
     80 DESC(KA)=K                                                         BLS04090
C                                                                          BLS04100
C       COMPUTE (A-MATRIX)*(ITH-q-BLOCK) - ((I-1)TH-Q-BLOCK)*B(I)-TRANS    BLS04110
C       WHERE B(I) DENOTES THE ITH SUBDIAGONAL BLOCK                       BLS04120
C                                                                          BLS04130
        IF(I.EQ.1) GO TO 110                                              BLS04140
        J1 = DIR(1,I-1)                                                   BLS04150
        J2 = DIR(2,I-1)                                                   BLS04160
        DO 100 K=I1,I2                                                    BLS04170
        KD=DESC(K)                                                        BLS04180
        JKD0 = (KD-1)*N                                                   BLS04190
        KK = (J1-2)*MXBLK + K                                             BLS04200
        DO 90 L=J1,J2                                                     BLS04210
        JL = (L-1)*N                                                      BLS04220
        KK = KK + MXBLK                                                   BLS04230
        S=TM(KK)                                                          BLS04240
        JKD = JKD0                                                        BLS04250
        DO 90 J=1,N                                                       BLS04260
        JKD = JKD + 1                                                     BLS04270
        JL = JL + 1                                                       BLS04280
```

```
   90 Q(JKD) = Q(JKD) - S*Q(JL)                                    BLS04290
  100 CONTINUE                                                     BLS04300
      LINT = (KD-1)*N + 1                                          BLS04310
      LFIN = KD*N                                                  BLS04320
C                                                                  BLS04330
C     COMPUTE A(I)                                                 BLS04340
C                                                                  BLS04350
  110 DO 130 K=I1,I2                                               BLS04360
      KKMX = (K-1)*MXBLK                                           BLS04370
      KD=DESC(K)                                                   BLS04380
      JKD1 = (KD-1)*N+ 1                                           BLS04390
      JL1 = (K-2)*N + 1                                            BLS04400
      DO 120 L=K,I2                                                BLS04410
      JL1 = JL1 + N                                                BLS04420
      KK = KKMX + L                                                BLS04430
C-----------------------------------------------------------------BLS04440
      TM(KK) = FINPRO(N,Q(JL1),1,Q(JKD1),1)                        BLS04450
C-----------------------------------------------------------------BLS04460
  120 CONTINUE                                                     BLS04470
  130 CONTINUE                                                     BLS04480
C                                                                  BLS04490
C     COMPUTE P(I) = P(I) - (ITH-Q-BLOCK)*A(I)                     BLS04500
C                                                                  BLS04510
      DO 170 K=I1,I2                                               BLS04520
      KKMX = (K-1)*MXBLK                                           BLS04530
      KD=DESC(K)                                                   BLS04540
      JKD0 = (KD-1)*N                                              BLS04550
      JL = (I1-1)*N                                                BLS04560
      DO 140 L=I1,I2                                               BLS04570
      KK = KKMX + L                                                BLS04580
      IF(L.LT.K) KK=(L-1)*MXBLK + K                                BLS04590
      S=TM(KK)                                                     BLS04600
      JKD = JKD0                                                   BLS04610
      DO 140 J=1,N                                                 BLS04620
      JL = JL + 1                                                  BLS04630
      JKD = JKD + 1                                                BLS04640
  140 Q(JKD) = Q(JKD) - S*Q(JL)                                    BLS04650
C                                                                  BLS04660
C     REORTHOGONALIZE THE BLOCK P(I) WITH RESPECT TO ALL VECTORS   BLS04670
C     IN THE 1ST QBLOCK THAT ARE NOT CURRENTLY GENERATING ANY      BLS04680
C     DESCENDANTS.  NOTE THAT 2ND Q-BLOCK IS REORTHOGONALIZED      BLS04690
C     ELSEWHERE.                                                   BLS04700
      IF(XLFT(I).EQ.0) GO TO 170                                   BLS04710
      LUP = XLFT(I)                                                BLS04720
      DO 160 JJ = 1,LUP                                            BLS04730
      L= LEFT(JJ)                                                  BLS04740
      JL0 = (L-1)*N                                                BLS04750
      LLMX = (L-1)*MXBLK                                           BLS04760
      JL1 = JL0 + 1                                                BLS04770
      JKD1 = JKD0 + 1                                              BLS04780
C-----------------------------------------------------------------BLS04790
      SUM = FINPRO(N,Q(JL1),1,Q(JKD1),1)                           BLS04800
C-----------------------------------------------------------------BLS04810
      JKD = JKD0                                                   BLS04820
      JL = JL0                                                     BLS04830
```

```
      DO 150 J=1,N                                              BLS04840
      JKD = JKD + 1                                             BLS04850
      JL = JL + 1                                               BLS04860
  150 Q(JKD) = Q(JKD) - SUM* Q(JL)                              BLS04870
      KK = LLMX + K                                             BLS04880
      TM(KK) = SUM + TM(KK)                                     BLS04890
C                                                               BLS04900
  160 CONTINUE                                                  BLS04910
  170 CONTINUE                                                  BLS04920
C                                                               BLS04930
C                                                               BLS04940
C   GENERATE B(I+1)                                             BLS04950
C                                                               BLS04960
      K1=DESC(I1)                                               BLS04970
      K2=DESC(I2)                                               BLS04980
      IFLAG=0                                                   BLS04990
C                                                               BLS05000
C     COMPUTE NORMS                                             BLS05010
C                                                               BLS05020
  180 CONTINUE                                                  BLS05030
      JK1 = (K1-2)*N + 1                                        BLS05040
      DO 190 K=K1,K2                                            BLS05050
      JK1 = JK1 + N                                             BLS05060
C-------------------------------------------------------------BLS05070
      SM(K) = FINPRO(N,Q(JK1),1,Q(JK1),1)                       BLS05080
C-------------------------------------------------------------BLS05090
  190 CONTINUE                                                  BLS05100
C                                                               BLS05110
        IF(I.EQ.1.AND.K1.EQ.I2+1)  WRITE(6,200) NITER,          BLS05120
     1 (K,SM(K), K =K1,K2)                                      BLS05130
  200 FORMAT(//' ON ITERATION', I4,' NORM(GRADIENTS)**2 OF 1ST BLOCK = 'BLS05140
     1/5(I4,E12.3))                                             BLS05150
C                                                               BLS05160
C     TEST FOR CONVERGENCE OF BLOCK LANCZOS                     BLS05170
C                                                               BLS05180
      IF(I.GT.1.OR.K1.GT.I2+1)  GO TO 250                       BLS05190
C                                                               BLS05200
C     TEST THE FIRST KM OF THE EIGENVALUES FOR CONVERGENCE      BLS05210
      K2L = K1 + KML - 1                                        BLS05220
      RKM = SM(K2L)                                             BLS05230
      DO 210 K = K1,K2L                                         BLS05240
      IF(SM(K).GT.ERRMAX )  GO TO 220                           BLS05250
  210 CONTINUE                                                  BLS05260
      GO TO 430                                                 BLS05270
C                                                               BLS05280
C     CAN WE REDUCE KACT?  IF A SMALL RESIDUAL (GRADIENT) IS IDENTIFIED,BLS05290
C     SIZE OF 1ST BLOCK MAY BE REDUCED.                         BLS05300
  220 IF(KML.EQ.KACT) GO TO 250                                 BLS05310
      DO 230 K = K2L,K2                                         BLS05320
      IF(SM(K).GT.ERRMAX) GO TO 230                             BLS05330
      KSAV = K                                                  BLS05340
      KACTN = KSAV - KACT                                       BLS05350
      GO TO 240                                                 BLS05360
C                                                               BLS05370
  230 CONTINUE                                                  BLS05380
```

```
      GO TO 250                                         BLS05390
C                                                       BLS05400
  240 K2 = KSAV                                         BLS05410
C                                                       BLS05420
C     GENERATE THE TRANSPOSE OF B(I)                    BLS05430
C                                                       BLS05440
  250 CONTINUE                                          BLS05450
C                                                       BLS05460
C     DETERMINE THE MAXIMAL NORM                        BLS05470
      K=K1                                              BLS05480
      S=SM(K)                                           BLS05490
      DO 260 L=K1,K2                                    BLS05500
      IF (SM(L).LT.S) GOTO 260                          BLS05510
      K=L                                               BLS05520
      S=SM(L)                                           BLS05530
  260 CONTINUE                                          BLS05540
C     FOR 2ND QBLOCK, SAVE INDEX AND SIZE OF MAXIMAL NORM BLS05550
      IF(I.GT.1) GO TO 270                              BLS05560
      IND = K - KACT                                    BLS05570
      RESN = SM(K)                                      BLS05580
C                                                       BLS05590
  270 IF(S.LE.ERRMAX)GO TO 340                          BLS05600
C                                                       BLS05610
      IF(IFLAG.EQ.1) GO TO 340                          BLS05620
C                                                       BLS05630
      S=DSQRT(S)                                        BLS05640
      JK0 = (K-1)*N                                     BLS05650
      JK = JK0                                          BLS05660
      DO 280 J=1,N                                      BLS05670
      JK = JK + 1                                       BLS05680
  280 Q(JK)=Q(JK)/S                                     BLS05690
      JL0 = (K1-2)*N                                    BLS05700
      DO 310 L=K1,K2                                    BLS05710
      JL0 = JL0 + N                                     BLS05720
      LL=(DESC(L) - 1)*MXBLK + K1                       BLS05730
      IF (L.NE.K) GOTO 290                              BLS05740
      TM(LL)=S                                          BLS05750
      GO TO 310                                         BLS05760
  290 JK1 = JK0 + 1                                     BLS05770
      JL1 = JL0 + 1                                     BLS05780
C-----------------------------------------------------------BLS05790
      T = FINPRO(N,Q(JK1),1,Q(JL1),1)                   BLS05800
C-----------------------------------------------------------BLS05810
      TM(LL)=T                                          BLS05820
      JK = JK0                                          BLS05830
      JL = JL0                                          BLS05840
      DO 300 J=1,N                                      BLS05850
      JK = JK + 1                                       BLS05860
      JL = JL + 1                                       BLS05870
  300 Q(JL) = Q(JL) - T*Q(JK)                           BLS05880
  310 CONTINUE                                          BLS05890
      IF (K.EQ.K1) GOTO 330                             BLS05900
C                                                       BLS05910
      JK1 = (K1-1)*N                                    BLS05920
      JK = JK0                                          BLS05930
```

```
          DO 320 J=1,N                                          BLS05940
          JK = JK + 1                                           BLS05950
          JK1 = JK1 + 1                                         BLS05960
          T=Q(JK1)                                              BLS05970
          Q(JK1)=Q(JK)                                          BLS05980
      320 Q(JK)=T                                               BLS05990
          MA=DESC(K)                                            BLS06000
          MB=DESC(K1)                                           BLS06010
          DESC(K1)=MA                                           BLS06020
          DESC(K)=MB                                            BLS06030
          DESC(MA)=K1                                           BLS06040
          DESC(MB)=K                                            BLS06050
      330 CONTINUE                                              BLS06060
C                                                               BLS06070
          DIR(2,I+1)=K1                                         BLS06080
C                                                               BLS06090
          IFLAG=1                                               BLS06100
C                                                               BLS06110
          K1=K1+1                                               BLS06120
          IF(I.EQ.1) GO TO 340                                  BLS06130
          IF (K1.LE.K2) GO TO 180                               BLS06140
C         RETURN TO LANCZS                                      BLS06150
C                                                               BLS06160
          RETURN                                                BLS06170
C                                                               BLS06180
C    IMPLICIT VECTOR DEFLATION                                  BLS06190
C                                                               BLS06200
      340 CONTINUE                                              BLS06210
          J= XLFT(I+2)                                          BLS06220
          IF(K1.GT.K2) GO TO 360                                BLS06230
          DO 350 L= K1,K2                                       BLS06240
          J = J+1                                               BLS06250
      350 LEFT(J) = DESC(L)                                     BLS06260
      360 XLFT(I+2) = J                                         BLS06270
C                                                               BLS06280
C         FORCE REORTHGONALIZATION OF 2ND AND 3RD QBLOCKS W.R.T. THOSE  BLS06290
C         VECTORS IN 1ST QBLOCK THAT ARE NOT GENERATING DESCENDANTS     BLS06300
C         ON THIS ITERATION.                                   BLS06310
          IF(I.GT.1)  GO TO 370                                 BLS06320
          XLFT(1) = XLFT(3)                                     BLS06330
          XLFT(2) = XLFT(3)                                     BLS06340
      370 IJJ = I + 2                                           BLS06350
          IJJJ= XLFT(IJJ)                                       BLS06360
C                                                               BLS06370
          IF(IJJJ.EQ.0) GO TO 390                               BLS06380
          IF(IWRITE.EQ.1)  WRITE(6,380) (LEFT(IJ),IJ= 1,IJJJ)   BLS06390
      380 FORMAT(' VECTORS NOT GENERATING DESCENDANTS ARE '/(10I6))  BLS06400
C                                                               BLS06410
      390 IF(I.EQ.1.AND.KML.GT.1) GO TO 400                     BLS06420
C                                                               BLS06430
          RETURN                                                BLS06440
C                                                               BLS06450
C         REORTHOGONALIZE 2ND QBLOCK W.R.T VECTORS IN 1ST BLOCK NOT     BLS06460
C         GENERATING DESCENDANTS                               BLS06470
      400 IF(XLFT(I).EQ.0) RETURN                               BLS06480
```

```
      LUP = XLFT(I)                                             BLS06490
      KD = DIR(2,I+1)                                           BLS06500
      JKD0 = (KD-1)*N                                           BLS06510
      DO 420 JJ = 1,LUP                                         BLS06520
      L = LEFT(JJ)                                              BLS06530
      JL0 = (L-1)*N                                             BLS06540
      JL1 = JL0 + 1                                             BLS06550
      JKD1 = JKD0 + 1                                           BLS06560
C---------------------------------------------------------------BLS06570
      SUM = FINPRO(N,Q(JKD1),1,Q(JL1),1)                        BLS06580
C---------------------------------------------------------------BLS06590
      JL = JL0                                                  BLS06600
      JKD = JKD0                                                BLS06610
      DO 410 J=1,N                                              BLS06620
      JL = JL + 1                                               BLS06630
      JKD = JKD + 1                                             BLS06640
  410 Q(JKD) = Q(JKD) - SUM *Q(JL)                              BLS06650
  420 CONTINUE                                                  BLS06660
C                                                               BLS06670
      RETURN                                                    BLS06680
C                                                               BLS06690
C     EXIT IF CONVERGENCE OF DESIRED EIGENVECTORS IS CONFIRMED. BLS06700
C                                                               BLS06710
  430 CONTINUE                                                  BLS06720
      DO 440 L=K1,K2                                            BLS06730
      M=DESC(L)                                                 BLS06740
  440 DESC(M)=0                                                 BLS06750
      DIR(2,2)=DIR(2,1)                                         BLS06760
C                                                               BLS06770
      WRITE(6,450) ERRMAX                                       BLS06780
  450 FORMAT(/' CONVERGENCE OBSERVED, ALL RESIDUALS**2 .LT. ERRMAX = ', BLS06790
     1 E20.12)                                                  BLS06800
C                                                               BLS06810
C                                                               BLS06820
    4 RETURN                                                    BLS06830
C-----END OF LANCI1---------------------------------------------BLS06840
      END                                                       BLS06850
C                                                               BLS06860
C-----ORTHOG----------------------------------------------------BLS06870
C     ORTHOGONALIZE COLUMNS M = MA,MB OF Q-ARRAY W.R.T COLUMNS M = 1,MB BLS06880
C                                                               BLS06890
      SUBROUTINE ORTHOG(MA,MB,N,Q)                              BLS06900
C                                                               BLS06910
C---------------------------------------------------------------BLS06920
      DOUBLE PRECISION  Q(1), S                                 BLS06930
      DOUBLE PRECISION FINPRO, DSQRT                            BLS06940
C---------------------------------------------------------------BLS06950
C     MAIN LOOP                                                 BLS06960
      DO 50 M = MA,MB                                           BLS06970
      MM0 = (M-1)*N                                             BLS06980
      LL0 = -N                                                  BLS06990
      DO 40 L = 1,M                                             BLS07000
      LL0 = LL0 + N                                             BLS07010
      LL = LL0 + 1                                              BLS07020
      MM = MM0 + 1                                              BLS07030
```

```
C-----------------------------------------------------------------------BLS07040
      S = FINPRO(N,Q(LL),1,Q(MM),1)                             BLS07050
C-----------------------------------------------------------------------BLS07060
C                                                               BLS07070
      IF (M.EQ.L) GO TO 20                                      BLS07080
C                                                               BLS07090
      MM = MM0                                                  BLS07100
      LL = LL0                                                  BLS07110
      DO 10 I=1,N                                               BLS07120
      LL = LL + 1                                               BLS07130
      MM = MM + 1                                               BLS07140
   10 Q(MM) = Q(MM) - S*Q(LL)                                   BLS07150
      GO TO 40                                                  BLS07160
C                                                               BLS07170
   20 S = DSQRT(S)                                              BLS07180
      MM = MM0                                                  BLS07190
      DO 30 I=1,N                                               BLS07200
      MM = MM + 1                                               BLS07210
   30 Q(MM) = Q(MM)/S                                           BLS07220
C                                                               BLS07230
   40 CONTINUE                                                  BLS07240
   50 CONTINUE                                                  BLS07250
C                                                               BLS07260
      RETURN                                                    BLS07270
C-----END OF ORTHOG-----------------------------------------------------BLS07280
      END                                                       BLS07290
C                                                               BLS07300
C-----START-------------------------------------------------------------BLS07310
C     GENERATES PSEUDO-RANDOM STARTING VECTORS.                 BLS07320
C                                                               BLS07330
      SUBROUTINE START(KA,KB,N,Q,G,ERRMAX)                      BLS07340
C                                                               BLS07350
C-----------------------------------------------------------------------BLS07360
      DOUBLE PRECISION  Q(1), ERRMAX, S                         BLS07370
      REAL  G(1)                                                BLS07380
      COMMON/RANDOM/IIX                                         BLS07390
      DOUBLE PRECISION FINPRO, DSQRT                            BLS07400
C-----------------------------------------------------------------------BLS07410
      IF(KA.GT.KB) RETURN                                       BLS07420
C                                                               BLS07430
      IIL = IIX                                                 BLS07440
      DO 110 K = KA,KB                                          BLS07450
      KK0 = (K-1)*N                                             BLS07460
C                                                               BLS07470
C-----------------------------------------------------------------------BLS07480
      CALL GENRAN(IIL,G,N)                                      BLS07490
C-----------------------------------------------------------------------BLS07500
C                                                               BLS07510
      KK = KK0                                                  BLS07520
      DO 10 I = 1,N                                             BLS07530
      KK = KK + 1                                               BLS07540
   10 Q(KK) = G(I)                                              BLS07550
      LL0 = -N                                                  BLS07560
   20 DO 70 L=1,K                                               BLS07570
      LL0 = LL0 + N                                             BLS07580
```

```
      LL = LL0 + 1                                          BLS07590
      KK = KK0 + 1                                          BLS07600
C-----------------------------------------------------------BLS07610
      S = FINPRO(N,Q(LL),1,Q(KK),1)                         BLS07620
C-----------------------------------------------------------BLS07630
C                                                           BLS07640
      IF (K.EQ.L) GO TO 40                                  BLS07650
C                                                           BLS07660
      LL = LL0                                              BLS07670
      KK = KK0                                              BLS07680
      DO 30 I=1,N                                           BLS07690
      LL = LL + 1                                           BLS07700
      KK = KK + 1                                           BLS07710
   30 Q(KK) = Q(KK) - S*Q(LL)                               BLS07720
      GO TO 70                                              BLS07730
C                                                           BLS07740
   40 S = DSQRT(S)                                          BLS07750
      IF(S.LE.ERRMAX) GO TO 80                              BLS07760
      KK = KK0                                              BLS07770
      DO 50 I=1,N                                           BLS07780
      KK = KK + 1                                           BLS07790
   50 Q(KK) = Q(KK)/S                                       BLS07800
C                                                           BLS07810
      WRITE(6,60) K                                         BLS07820
   60 FORMAT(I6,' TH STARTING VECTOR IS GENERATED RANDOMLY')BLS07830
C                                                           BLS07840
   70 CONTINUE                                              BLS07850
      GO TO 110                                             BLS07860
C                                                           BLS07870
C-----------------------------------------------------------BLS07880
   80 CALL GENRAN(IIX,G,N)                                  BLS07890
C-----------------------------------------------------------BLS07900
C                                                           BLS07910
      WRITE(6,90) K                                         BLS07920
   90 FORMAT(/I6,' TH RANDOM VECTOR REJECTED, GENERATE ANOTHER'/)BLS07930
C                                                           BLS07940
      KK = KK0                                              BLS07950
      DO 100 I = 1,N                                        BLS07960
      KK = KK + 1                                           BLS07970
  100 Q(KK) = G(I)                                          BLS07980
      GO TO 20                                              BLS07990
C                                                           BLS08000
  110 CONTINUE                                              BLS08010
      RETURN                                                BLS08020
C-----END OF START------------------------------------------BLS08030
      END                                                   BLS08040
C                                                           BLS08050
C-----START OF DIAGOM---------------------------------------BLS08060
C     DIAGOM CALLS THE EISPACK SUBROUTINES TRED2 AND IMTQL2 TO  BLS08070
C     DIAGONALIZE THE SMALL SYMMETRIC MATRICES GENERATED AT EACH BLS08080
C     ITERATION OF BLOCK LANCZOS.                           BLS08090
C                                                           BLS08100
      SUBROUTINE DIAGOM(MXBLK,MM,TM,KACT,N,Q,E,RESID,RESK,RESN,IND, BLS08110
     1 KACTN,KM,TD,TOD,NITER,IERR,IWRITE)                   BLS08120
C                                                           BLS08130
```

```
C-------------------------------------------------------------------BLS08140
      DOUBLE PRECISION   TM(MXBLK,1),Q(1),E(1),TD(*),TOD(1),RESID(1)   BLS08150
      DOUBLE PRECISION   RESK(1),RESN,RATIO,FRACT,RKM,EMAX,SPREAD,EGAP  BLS08160
      DOUBLE PRECISION   DABS,DFLOAT,DMAX1                              BLS08170
C-------------------------------------------------------------------BLS08180
      IF(NITER.GE.100) GO TO 270                                       BLS08190
      RKM = TD(1)                                                      BLS08200
      FRACT = TD(2)                                                    BLS08210
      NSTAG = IERR                                                     BLS08220
      KWANT = KACT                                                     BLS08230
C                                                                      BLS08240
C     STORE KM-TH RESIDUALS**2 FOR CHECK ON STAGNATION OF CONVERGENCE  BLS08250
      NITER1 = NITER + 1                                               BLS08260
      RESK(NITER1) = RKM                                               BLS08270
      IF(NITER.LE.NSTAG) GO TO 10                                      BLS08280
C     TEST FOR STAGNATION                                              BLS08290
      NITERM = NITER - 10                                              BLS08300
      RATIO = RKM / RESK(NITERM)                                       BLS08310
      IF(RATIO.GT.FRACT) GO TO 250                                     BLS08320
C                                                                      BLS08330
   10 CONTINUE                                                         BLS08340
C                                                                      BLS08350
C     TEST GAPS TO DETERMINE IF SIZE OF 1ST Q-BLOCK CAN BE REDUCED     BLS08360
      IF(NITER.EQ.0) GO TO 40                                          BLS08370
      IF(KM.EQ.KACT.OR.NITER.LT.10) GO TO 30                           BLS08380
      KACT1 = KACT - 1                                                 BLS08390
      DO 20 K = KM,KACT1                                               BLS08400
      RATIO =  DABS(E(K+1) - E(K))                                     BLS08410
      IF(RATIO.LT.25*EGAP) GO TO 20                                    BLS08420
      KACT = K                                                         BLS08430
      GO TO 40                                                         BLS08440
   20 CONTINUE                                                         BLS08450
C                                                                      BLS08460
C     IF KACT.NE.KACTN, THEN SUBROUTINE LANCI1 IDENTIFIED A VERY       BLS08470
C     SMALL RESIDUAL FOR SOME E(J), J>= KM.                            BLS08480
   30 IF(KACT.EQ.KACTN)  GO TO 50                                      BLS08490
      RATIO = DABS(E(KACTN+1) - E(KACTN))                              BLS08500
      IF(RATIO.LE.EGAP)  GO TO 50                                      BLS08510
      KACT = KACTN                                                     BLS08520
   40 ICOUNT = 1                                                       BLS08530
      INDEXP = IND                                                     BLS08540
      RESID(1) = RESN                                                  BLS08550
      GO TO 80                                                         BLS08560
C                                                                      BLS08570
   50 CONTINUE                                                         BLS08580
      IF(IND.NE.INDEXP) GO TO 70                                       BLS08590
C     INDEX OF VECTOR OF MAXIMUM NORM IS SAME AS ON PREVIOUS ITERATION BLS08600
      ICOUNT = ICOUNT + 1                                              BLS08610
      IF(ICOUNT.LE.5) GO TO 60                                         BLS08620
      ITEST = ICOUNT - 4                                               BLS08630
      RATIO = RESID(ITEST)/RESN                                        BLS08640
      IF(DABS(RATIO).GT.10.D0) GO TO 60                                BLS08650
C                                                                      BLS08660
C     CONVERGENCE STAGNATED, ADD NEXT RITZ VECTOR IN THE CHAIN         BLS08670
C     TO THE 1ST Q-BLOCK AND RESET THE FLAGS THAT KEEP TRACK OF        BLS08680
```

```
C      CONVERGENCE.                                              BLS08690
       INDEXP = IND                                             BLS08700
       ICOUNT = 0                                               BLS08710
       KACT = KACT + 1                                          BLS08720
       KWANT = KACT                                             BLS08730
C      CHECK THAT THERE IS ENOUGH ROOM TO ENLARGE THE 1ST QBLOCK BLS08740
       IF(2*KACT.GT.MXBLK) GO TO 230                            BLS08750
       GO TO 80                                                 BLS08760
C                                                               BLS08770
   60 RESID(ICOUNT) = RESN                                      BLS08780
       INDEXP = IND                                             BLS08790
       GO TO 80                                                 BLS08800
C                                                               BLS08810
   70 ICOUNT = 1                                                BLS08820
       RESID(1) = RESN                                          BLS08830
       INDEXP = IND                                             BLS08840
C                                                               BLS08850
C------------------------------------------------------------------BLS08860
C      USE EISPACK SUBROUTINES TO DIAGONALIZE THE SMALL TM-MATRIX. BLS08870
C                                                               BLS08880
   80 CALL TRED2(MXBLK,MM,TM,TD,TOD,TM)                         BLS08890
       CALL IMTQL2(MXBLK,MM,TD,TOD,TM,IERR)                     BLS08900
C------------------------------------------------------------------BLS08910
       IF(IERR.EQ.0) GO TO 90                                   BLS08920
       RETURN                                                   BLS08930
C                                                               BLS08940
C   SELECT RELEVANT EIGENVALUES AND EIGENVECTORS OF THE T-MATRIX. BLS08950
   90 CONTINUE                                                  BLS08960
C                                                               BLS08970
C   IMTQL2 RETURNS EIGENVALUES (AND CORRESPONDING EIGENVECTORS) IN BLS08980
C   ALGEBRAICALLY-ASCENDING ORDER.  REARRANGE TO DESCENDING ORDER. BLS08990
C                                                               BLS09000
       DO 100 L=1,MM                                            BLS09010
       MML = MM-L+1                                             BLS09020
  100 E(L) = TD(MML)                                            BLS09030
C                                                               BLS09040
  110 WRITE(6,120) KACT, (E(J), J=1,KACT)                       BLS09050
  120 FORMAT(' COMPUTED',I4,' ALGEBRAICALLY-LARGEST EIGENVALUES'/(4E20.1BLS09060
   12))                                                         BLS09070
C                                                               BLS09080
C      COMPUTE ESTIMATE MAXIMUM EIGENVALUE AND OF SPREAD        BLS09090
       IF(NITER.GT.1) GO TO 140                                 BLS09100
       EMAX = DMAX1(DABS(E(1)),DABS(E(MM)))                     BLS09110
       SPREAD = DABS(E(1) - E(MM))                              BLS09120
       EGAP = SPREAD/DFLOAT(N)                                  BLS09130
       IF(NITER.EQ.1)  WRITE(6,130) EMAX,SPREAD,EGAP           BLS09140
  130 FORMAT(/4X,'ESTIMATED NORM OF MATRIX',4X,'ESTIMATED SPREAD',6X,'SPBLS09150
   1READ*(SIZE)*(-1)'/E28.4,E20.4,E24.3)                       BLS09160
  140 CONTINUE                                                  BLS09170
C                                                               BLS09180
C      COMPUTE RITZ VECTORS                                     BLS09190
       DO 180 I=1,N                                             BLS09200
       DO 150 KK=1,KWANT                                        BLS09210
       TOD(KK)=0.D0                                             BLS09220
       K = MM - KK + 1                                          BLS09230
```

```
          IL = - N + I                                            BLS09240
          DO 150 L = 1,MM                                         BLS09250
          IL = IL + N                                             BLS09260
  150 TOD(KK) = TOD(KK) + TM(L,K)*Q(IL)                           BLS09270
          IKK = -N + I                                            BLS09280
  160 DO 170 KK=1,KACT                                            BLS09290
          IKK = IKK + N                                           BLS09300
  170 Q(IKK)=TOD(KK)                                              BLS09310
  180 CONTINUE                                                    BLS09320
C                                                                 BLS09330
C     ON FILE 13 SAVE ANY EXTRA VECTORS NO LONGER NEEDED IN 1ST Q-BLOCK BLS09340
          IF(KWANT.EQ.KACT) GO TO 290                             BLS09350
          K1 = KACT + 1                                           BLS09360
          K2 = KWANT                                              BLS09370
          DUMMY = 100.                                            BLS09380
          DO 190 K = K1,K2                                        BLS09390
          LINT = (K-1)*N + 1                                      BLS09400
          LFIN = K*N                                              BLS09410
          WRITE(13,210) E(K),DUMMY,K                              BLS09420
          WRITE(13,220) (Q(L), L=LINT,LFIN)                       BLS09430
  190 CONTINUE                                                    BLS09440
          KDELTA = KWANT - KACT                                   BLS09450
          WRITE(13,200) KDELTA                                    BLS09460
  200 FORMAT(/' ABOVE ARE ',I3,' VECTORS STRIPPED FROM A 1ST Q-BLOCK'/  BLS09470
     1' DURING A BLOCK LANZCOS RUN WHICH COULD BE USED AS STARTING VECTOBLS09480
     1RS'/' IN A LATER RUN IF THE USER DECIDES THAT THESE EIGENVALUES SHBLS09490
     10ULD'/' BE COMPUTED AFTER ALL.  FORMAT USED IN THE SAME AS WAS USEBLS09500
     1D'/' IN THE CORRESPONDING BLSTARTV FILE'/)                 BLS09510
  210 FORMAT(/E20.12,E13.4,I6,' = EVAL,DUMMY,EVAL NUMBER,EVEC=')   BLS09520
  220 FORMAT(4E20.12)                                            BLS09530
          GO TO 290                                              BLS09540
C                                                                 BLS09550
C     DEFAULT, SIZE OF 1ST Q-BLOCK TOO LARGE FOR MXBLK            BLS09560
  230 IWRITE = -1000                                             BLS09570
          WRITE(6,240) KACT,MXBLK                                 BLS09580
          WRITE(15,240) KACT,MXBLK                                BLS09590
  240 FORMAT(//' BLOCK LANCZOS PROCEDURE TRIED TO INCREASE THE SIZE OF 1BLS09600
     1ST QBLOCK'/' TO ',I3,' BUT THIS IS NOT FEASIBLE BECAUSE TWICE THISBLS09610
     1 SIZE'/' IS G.T. MXBLK WHICH EQUALS ',I4/' USER CAN RERUN PROGRAM BLS09620
     1WITH LARGER MXBLK'/)                                       BLS09630
          GO TO 290                                              BLS09640
C                                                                 BLS09650
C     DEFAULT, CONVERGENCE RATE IS TOO SLOW                       BLS09660
  250 IWRITE = -1000                                             BLS09670
          WRITE(6,260) NITER,RATIO,FRACT                          BLS09680
          WRITE(15,260) NITER,RATIO,FRACT                         BLS09690
  260 FORMAT(//' ON ITERATION ',I3,' CONVERGENCE APPEARS TO BE STAGNATEDBLS09700
     1'/' RATIO OF SQUARE OF CURRENT KM-TH RESIDUAL TO CORRESPONDING SQUBLS09710
     1ARE'/' 10 ITERATIONS EARLIER IS ',E10.3,' COMPARED TO '/    BLS09720
     1' USER-SPECIFIED RATIO ',E10.3,'. THEREFORE, PROGRAM TERMINATES'/'BLS09730
     1 USER SHOULD LOOK AT THE OUTPUT. IF CONVERGENCE HAS STAGNATED, USEBLS09740
     1R'/' CAN EITHER INCREASE KACT OR KMAX OR RESET THE STAGNATION PARABLS09750
     1METERS'/' NSTAG AND FRACT, AND RESTART THE BLOCK PROCEDURE USING TBLS09760
     1HE'/' CURRENT EIGENVECTOR APPROXIMATIONS AS STARTING VECTORS'/)   BLS09770
          GO TO 290                                              BLS09780
```

```
C                                                             BLS09790
  270 IWRITE = -1000                                          BLS09800
      WRITE(6,280)                                            BLS09810
      WRITE(15,280)                                           BLS09820
  280 FORMAT(//' SOMETHING IS SERIOUSLY WRONG.  NUMBER OF ITERATIONS IS BLS09830
     1EXCESSIVE'/' PROGRAM TERMINATES FOR USER TO DECIDE WHAT TO DO'/'  BLS09840
     1ALTERNATIVES INCLUDE INCREASING KACT OR KMAX OR BOTH, AND RESTARTIBLS09850
     1NG'/' USING THE CURRENT APPROXIMATIONS AS STARTING VECTORS'//)     BLS09860
C                                                             BLS09870
  290 CONTINUE                                                BLS09880
      RETURN                                                  BLS09890
C-----END OF DIAGOM-------------------------------------------------BLS09900
      END                                                     BLS09910
C-----LPERM PERMUTES VECTORS----------------------------------   BLS09920
C                                                             BLS09930
      SUBROUTINE LPERM(W,U,IPERM)                             BLS09940
C                                                             BLS09950
C---------------------------------------------------------------BLS09960
      DOUBLE PRECISION  U(1),W(1)                             BLS09970
      INTEGER  IPR(1),IPT(1)                                  BLS09980
C---------------------------------------------------------------BLS09990
C     SUBROUTINE HAS 2 BRANCHES:  IPERM = 1,  CALCULATES      BLS10000
C     U = P*W  WHERE  P IS THE PERMUTATION REPRESENTED BY IPR  BLS10010
C     LET J = IPR(K) THEN U(K) = W(J), K = 1,N. WE SET W(K)=U(K), K=1,N BLS10020
C     IPERM = 2, USING THE PERMUTATION IPT (P-TRANSPOSE) U = P'*W, W=U  BLS10030
C     LET J = IPT(K) THEN U(K) = W(J), K=1,N. WE SET W(K) = U(K), K=1,N BLS10040
C---------------------------------------------------------------BLS10050
C                                                             BLS10060
      GO TO 3                                                 BLS10070
C---------------------------------------------------------------BLS10080
      ENTRY LPERME(IPR,IPT,N)                                 BLS10090
      GO TO 4                                                 BLS10100
C---------------------------------------------------------------BLS10110
C                                                             BLS10120
    3 CONTINUE                                                BLS10130
      IF(IPERM.EQ.2) GO TO 10                                 BLS10140
C     IPERM = 1                                               BLS10150
      DO 20 K = 1,N                                           BLS10160
      J = IPR(K)                                              BLS10170
   20 U(K) = W(J)                                             BLS10180
      DO 30 K = 1,N                                           BLS10190
   30 W(K) = U(K)                                             BLS10200
      GO TO 60                                                BLS10210
C     IPERM = 2                                               BLS10220
   10 DO 40  K = 1,N                                          BLS10230
      J = IPT(K)                                              BLS10240
   40 U(K) = W(J)                                             BLS10250
      DO 50 K = 1,N                                           BLS10260
   50 W(K) = U(K)                                             BLS10270
   60 CONTINUE                                                BLS10280
C                                                             BLS10290
C                                                             BLS10300
C-----END OF LPERM------------------------------------------   BLS10310
    4 RETURN                                                  BLS10320
      END                                                     BLS10330
```

8.6 BLEVAL: File Definitions, Sample Input File

Below is a listing of the input/output files which are accessed by the real symmetric block Lanczos eigenvalue/eigenvector program, BLEVAL. BLEVAL computes a few extreme eigenvalues and corresponding eigenvectors of a real symmetric matrix A. Also below is a sample of the input file which BLEVAL requires on file 5. The parameters in this file are supplied in free format. File 8 contains data for the nxn real symmetric matrix A.

```
Sample Specifications of Input/Output Files for BLEVAL
------------------------------------------------------------------
 BLEVAL EXEC
FI 06 TERM
FILEDEF  5 DISK BLEVAL    INPUT     A (RECFM F LRECL 80 BLOCK 80
FILEDEF  8 DISK &1        INPUT     A (RECFM F LRECL 80 BLOCK 80
FILEDEF 10 DISK &1        BLSTARTV  A (RECFM F LRECL 80 BLOCK 80
FILEDEF 13 DISK &1        BLEXTRAV  A (RECFM F LRECL 80 BLOCK 80
FILEDEF 15 DISK &1        BLEIGVEC  A (RECFM F LRECL 80 BLOCK 80
*IMTQL2 AND TRED2 ARE 2 EISPACK LIBRARY SUBROUTINES
LOAD  BLEVAL  BLSUB  BLMULT  IMTQL2  TRED2
------------------------------------------------------------------
```

```
Sample Input File for BLEVAL
-----------------------------------------------------------------------
 LINE 1 IWRITE (SPECIFY MESSAGE LEVEL TO FILE 6: 1 MEANS DETAILED
          1
 LINE 2    N      MATNO  (SIZE OF A-MATRIX, MATRIX IDENT. NUMBER
        528       528
 LINE 3 MDIMQ        MDIMTM       MAXIT (DIMS. Q, TM, MAX Ax-mults
       40000         2500         1000
 LINE 4 EFLAG  OFLAG (EFLAG=(0,1) 1=2PHASES. OFLAG: 1=ORTHOG CHECK
          1       1
 LINE 5 SEED      (STARTING VECTOR SEED, RANDOM NUMBER GENERATOR
      3482736
 LINE 6 KMAX  KACT  KSET (MAX T SIZE +1,SIZE 1ST BLOCK,VECS SUPPLIED
        21     4      0
 LINE 7 KM    (NUMB. EVS FOR ALG-LARGEST, -(NUMB. EVS) FOR ALG-SMALLEST
         4
 LINE 8 NSTAG  FRACT (NO. ITNS BEFORE TEST CONVERGENCE, TEST FRACTION
         25    .01
 LINE 9 RELTOL    MAXIT2 (PHASE 2, CONVERGE. TOL. , Max Ax-mults
    .00000001     1000
-----------------------------------------------------------------------
```

Chapter 9

Factored Inverses, Real Symmetric Block Lanczos Code

9.1 Introduction

The FORTRAN codes in this chapter address the question of using an iterative block Lanczos procedure to compute a 'few' eigenvalues and a basis for the corresponding eigenspace of a real symmetric matrix A by computing a few extreme eigenvalues and a corresponding basis for the inverse of a real symmetric matrix B obtained from A by scaling, shifting and permuting A. For a given real symmetric matrix A, the codes consider the inverse of a matrix B where

$$B \equiv PCP^T, \quad C \equiv (SCALE) * A + (SHIFT) * I, \tag{9.1.1}$$

$SCALE$ and $SHIFT$ are specified by the user, and the permutation matrix P is chosen so that for a sparse matrix A (or C), the resulting factorization of the associated B matrix is also sparse. An eigenvalue is 'extreme' if it is one of the algebraically-smallest or the algebraically-largest eigenvalues in the eigenvalue spectrum.

Specifically, for a given real symmetric matrix A and associated B-matrix as defined in Eqn(9.1.1), the codes in this chapter compute the q algebraically-largest eigenvalues, λ_i , $1 \leq i \leq q$, of B^{-1} and corresponding orthonormal real vectors $X_q \equiv (x_1, \ldots, x_q)$ such that

$$B^{-1}X_q = X_qA_q, \quad A_q \equiv X_q^T A X_q. \tag{9.1.2}$$

Typically, $A_q = \Lambda_q$, a diagonal matrix whose nonzero entries are the eigenvalues λ_i . The number q is small and specified by the user.

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

The Lanczos code contained in this chapter is a simple modification of the hybrid 'block' Lanczos procedure given in Chapter 8 to handle the factored inverse of the B-matrix given in Eqn(9.1.1). Therefore please see Chapter 8, Section 8.1, for comments about this procedure and for comments regarding the differences between iterative block Lanczos procedures and single-vector Lanczos procedures.

BLIEVAL is the main 'block' program for the factored inverse version of the 'block' Lanczos codes in

Chapter 8. BLIEVAL uses the same subroutines as the real symmetric codes in Chapter 8, with the exception of the user-supplied subroutines. The user must supply a subroutine USPEC which defines and initializes the matrix which is to be used by the LANCZS and LANCI1 subroutines. In the factored inverse case, USPEC specifies the factorization of the particular B-matrix being used. These Lanczos programs do not require the A-matrix. However, the user must supply the scalars $SCALE$ and $SHIFT$, and the permutation P (if any). The user must also supply a subroutine BLSOLV which solves the system of equations $Bu = x$ for any given vector x.

The sample USPEC and BLSOLV subroutines provided assume that the B-matrix being used is positive definite and that the Cholesky factors of B,

$$B = LL^T \tag{9.1.3}$$

where L is a lower triangular matrix, are used for the matrix-vector multiply, $B^{-1}x$, for any given vector x. However, the user may replace these subroutines by subroutines which define and use a more general factorization. These Lanczos codes only require that the BLSOLV subroutine solves the system $Bu = x$, rapidly and accurately.

All computations are in double precision real arithmetic. On each iteration, the accuracy of the computed eigenvectors is checked in the process of computing the second block of Lanczos vectors on that iteration. Note that the eigenvectors of B^{-1} are simple permutations of the eigenvectors of A. These permutations are undone prior to the termination of the block procedure. The corresponding eigenvalues of A are obtained from those of B^{-1} by a simple scalar transformation which is included in the codes. The eigenelement computations for the small Lanczos matrices use two subroutines from the EISPACK Library [23, 8], TRED2 and IMTQL2.

Several optional preprocessing programs are provided, PERMUT, LORDER, LFACT, and LTEST. Listings for these programs are given in Chapter 4. PERMUT calls the SPARSPAK Library [23, 8] to attempt to identify a reordering or permutation P of a given matrix A for which sparseness is preserved under factorization of the permuted matrix. LORDER takes a given matrix C and permutation P and computes the sparse matrix format for the permuted matrix, $B \equiv PCP^T$. LFACT computes the Cholesky factors of a given positive definite matrix. LTEST performs a very crude check on the numerical condition of the matrix supplied to it, by solving a system of equations with and without iterative refinement, LINPACK [7].

The usefulness of this code for computing a few interior eigenvalues of a given real symmetric matrix is dubious. For such an application one would have to select a shift $SHIFT$ that places the desired eigenvalues of the A-matrix on the extreme of the spectrum of the associated matrix B^{-1} and is chosen so that the B-matrix is well-conditioned numerically. This is not a trivial task. The user should refer to Chapter 7 of Volume 1 of this book for more details on iterative block Lanczos procedures.

9.2 BLIEVAL: Main Program, Eigenvalue and Eigenvector Computations

```
C-----BLIEVAL (FEW EXTREME EIGENVALUES AND EIGENVECTORS)----------------BLI00010
C             (USING FACTORED INVERSE OF A REAL SYMMETRIC MATRIX)       BLI00020
C  Authors:  Jane Cullum* and Bill Donath**                            BLI00025
C          **IBM Research, T.J. Watson Research Center                  BLI00030
C          **Yorktown Heights, N.Y. 10598                              BLI00040
C          * Los Alamos National Laboratory                            BLI00050
C          * Los Alamos, New Mexico 87544                              BLI00060
C             E-mail:  cullumj@lanl.gov                                BLI00070
C                                                                       BLI00080
C  These codes are copyrighted by the authors.  These codes            BLI00090
C  and modifications of them or portions of them are NOT to be          BLI00100
C  incorporated into any commercial codes or used for any other        BLI00110
C  commercial purposes such as consulting for other companies,          BLI00120
C  without legal agreements with the authors of these Codes.           BLI00130
C  If these Codes or portions of them are used in other scientific or   BLI00140
C  engineering research works the names of the authors of these codes   BLI00150
C  and appropriate references to their written work are to be          BLI00160
C  incorporated in the derivative works.                               BLI00170
C                                                                       BLI00180
C  This header is not to be removed from these codes.                  BLI00190
C                                                                       BLI00195
C         REFERENCE: Cullum and Willoughby, Chapter 7,                 BLI00200
C         Lanczos Algorithms for Large Symmetric Eigenvalue ComputationsBLI00205
C         VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in  BLI00210
C         Applied Mathematics, 2002. SIAM Publications,                BLI00215
C         Philadelphia, PA. USA                                        BLI00220
C                                                                       BLI00225
C     CONTAINS MAIN PROGRAM FOR COMPUTING A FEW EIGENVALUES            BLI00230
C     AND CORRESPONDING EIGENVECTORS OF A REAL SYMMETRIC MATRIX        BLI00235
C     BY COMPUTING A FEW OF THE ALGEBRAICALLY-LARGEST OR              BLI00240
C     ALGEBRAICALLY-SMALLEST EIGENVALUES OF THE INVERSE OF A SCALED,   BLI00250
C     SHIFTED, AND PERMUTED VERSION B OF THE ORIGINAL A-MATRIX         BLI00260
C     USING A BLOCK FORM OF LANCZOS TRIDIAGONALIZATION WITH LIMITED    BLI00270
C     REORTHOGONALIZATION.  THIS BLOCK PROCEDURE IS ITERATIVE AND      BLI00280
C     REQUIRES A SUBROUTINE BLSOLV THAT FOR ANY GIVEN VECTOR W         BLI00290
C     COMPUTES U SUCH THAT B*U = W.   THE SAMPLE BLSOLV SUBROUTINES    BLI00300
C     PROVIDED FOR SPARSE MATRICES ARE ONLY FOR THE CASE THAT B IS     BLI00310
C     POSITIVE DEFINITE AND USE THE CHOLESKY FACTORS OF B.  HOWEVER,   BLI00320
C     THE USER COULD REPLACE THESE BY A SUBROUTINE WHICH COMPUTES      BLI00330
C     FOR AN INDEFINITE MATRIX THE FACTORIZATION L*D*(L-TRANSPOSE).    BLI00340
C                                                                       BLI00350
C     THIS BLOCK PROCEDURE COMPUTES THE ALGEBRAICALLY-LARGEST          BLI00360
C     EIGENVALUES OF THE INVERSE OF THE B-MATRIX, UNLESS THE USER      BLI00370
C     SUPPLIES -(B-INVERSE)*X RATHER THAN (B-INVERSE)*X, IN WHICH      BLI00380
C     CASE IT COMPUTES THE CORRESPONDING ALGEBRAICALLY-SMALLEST        BLI00390
C     EIGENVALUES OF (B-INVERSE) BY COMPUTING THE ALGEBRAICALLY-       BLI00400
C     LARGEST EIGENVALUES OF -(B-INVERSE).  IN THIS CASE THE SIGNS     BLI00410
C     OF THE COMPUTED EIGENVALUES ARE CHANGED PRIOR TO WRITING TO      BLI00420
C     FILE 15 SO THAT ON EXIT, FILE 15 CONTAINS THE ALGEBRAICALLY-     BLI00430
C     SMALLEST EIGENVALUES OF B-INVERSE ALONG WITH THE CORRESPONDING   BLI00440
```

```
C       EIGENVALUES OF THE ORIGINAL A-MATRIX AND CORRESPONDING          BLI00450
C       EIGENVECTORS.  THE MATRIX B = S0*P*A*P' + SHIFT*I WHERE THE      BLI00460
C       SCALE S0 AND SHIFT ARE READ IN THIS PROGRAM, AND THE            BLI00470
C       PERMUTATION P IS DEFINED IN THE CORRESPONDING USPEC SUBROUTINE.  BLI00480
C       THE PROGRAM ASSUMES THAT THE FACTORIZATION READ IN USPEC        BLI00490
C       CORRESPONDS TO THE S0, SHIFT AND PERMUTATION READ IN.  THE S0    BLI00500
C       AND SHIFT ARE CHOSEN SO THAT THE DESIRED EIGENVALUES ARE AT      BLI00510
C       THE EXTREME OF THE SPECTRUM OF B-INVERSE.                        BLI00520
C                                                                        BLI00530
C       THIS IS AN ITERATIVE 'BLOCK' LANCZOS PROCEDURE FOR WHICH ON      BLI00540
C       EVERY ITERATION, THE 2ND AND SUCCEEDING BLOCKS CONTAIN ONLY ONE  BLI00550
C       VECTOR WHICH IS SELECTED ON THE BASIS OF ITS EXPECTED INFLUENCE  BLI00560
C       ON THE CONVERGENCE.   Q-BLOCKS GENERATED ON A GIVEN ITERATION    BLI00570
C       ARE REORTHOGONALIZED ONLY W.R.T. THOSE VECTORS IN THE FIRST      BLI00580
C       Q-BLOCK WHICH ARE NOT ALLOWED TO GENERATE DESCENDANTS ON         BLI00590
C       THAT ITERATION.                                                  BLI00600
C                                                                        BLI00610
C       PFORT VERIFIER IDENTIFIED THE FOLLOWING NONPORTABLE CONSTRUCTIONS:BLI00620
C       1.  DATA MACHEP DEFINITION                                       BLI00630
C       2.  FORMAT (20A4) USED FOR READING EXPLANATORY COMMENTS.         BLI00640
C       3.  FREE FORMAT (5,*), USED FOR PARAMETER INPUT FROM FILE 5.     BLI00650
C       4.  COMMON/LOOPS/ AS CONSTRUCTED IS NOT PORTABLE                 BLI00660
C                                                                        BLI00670
C-----------------------------------------------------------------------BLI00680
        DOUBLE PRECISION Q(44000),E(50),TM(2500),TOD(50),TD(50),EPSM,NNZ BLI00690
        DOUBLE PRECISION SM(100),ERRMAX,SPREC,MACHEP,AVER,RELTOL,ERRMAN  BLI00700
        DOUBLE PRECISION  EVAL, RESIDL(100), RESIDK(100), RESID, FRACT   BLI00710
        DOUBLE PRECISION S0,SHIFT                                        BLI00720
        REAL EXPLAN(20),G(2000)                                          BLI00730
        INTEGER   DIR(2,100),DESC(100),LEFT(100),XLFT(100)              BLI00740
        INTEGER SEED,OFLAG,EFLAG                                         BLI00750
        COMMON/LOOPS/MAXIT,ITER                                          BLI00760
        COMMON /RANDOM/SEED                                              BLI00770
        COMMON/FLAGS/EFLAG,OFLAG                                         BLI00780
        DOUBLE PRECISION  DABS, DFLOAT                                   BLI00790
C-----------------------------------------------------------------------BLI00800
        EXTERNAL BLSOLV                                                  BLI00810
        DATA MACHEP/Z3410000000000000/                                  BLI00820
C-----------------------------------------------------------------------BLI00830
C                                                                        BLI00840
C       ARRAYS MUST DIMENSIONED AS FOLLOWS:                              BLI00850
C                                                                        BLI00860
C       1.  Q:   >=  KMAX*N                                              BLI00870
C       2.  G:   >= N                                                    BLI00880
C       3.  E:   >=  MXBLK                                               BLI00890
C       4.  TM:   >=  MXBLK**2                                           BLI00900
C       5.  TOD, TD, SM, DESC, LEFT, XLFT:   >= MXBLK                    BLI00910
C       6.  DIR:  ROW DIMENSION = 2;   COLUMN DIMENSION >= MXBLK         BLI00920
C       7.  RESIDL, RESIDK:  >= MAXIMUM NUMBER OF ITERATIONS ALLOWED.    BLI00930
C           PROGRAM CURRENTLY TERMINATES IF MORE THAN 100 ITERATIONS     BLI00940
C           ARE REQUESTED.  USED TO MONITOR CONVERGENCE. SEE SUBROUTINE  BLI00950
C           DIAGOM.                                                      BLI00960
C       8.  EXPLAN: DIMENSION = 20.                                      BLI00970
C                                                                        BLI00980
C-----------------------------------------------------------------------BLI00990
```

```
C      OUTPUT HEADER                                                 BLI01000
       WRITE(6,10)                                                    BLI01010
    10 FORMAT(/' BLOCK LANCZOS PROCEDURE, USES FACTORED INVERSE OF A USERBLI01020
      1-SPECIFIED MATRIX'/' 2ND AND SUCCEEDING BLOCKS GENERATED ON EACH BBLI01030
      1LOCK ITERATION '/' CONTAIN ONLY ONE VECTOR'//)                 BLI01040
C                                                                     BLI01050
C      SET PROGRAM PARAMETERS                                         BLI01060
       EPSM = 2.D0*MACHEP                                             BLI01070
       SPREC = 1.D-5                                                  BLI01080
       MPMIN = -1000                                                  BLI01090
C                                                                     BLI01100
C      READ USER-SPECIFIED PARAMETERS FROM INPUT FILE 5 (FREE FORMAT) BLI01110
C                                                                     BLI01120
C      SELECT THE AMOUNT OF INTERMEDIATE OUTPUT DESIRED (IWRITE =0,1). BLI01130
C      IWRITE = 1 INCREASES THE AMOUNT OF INTERMEDIATE OUTPUT WRITTEN BLI01140
C      TO FILE 6 ON EACH ITERATION OF THE BLOCK LANCZOS PROCEDURE.    BLI01150
       READ(5,20) EXPLAN                                              BLI01160
    20 FORMAT(20A4)                                                   BLI01170
       READ(5,*)  IWRITE                                              BLI01180
C                                                                     BLI01190
C      READ ORDER (N) OF MATRIX AND MATRIX IDENTIFICATION NUMBER (MATNO) BLI01200
C      READ SCALE (S0) AND SHIFT (SHIFT) APPLIED TO MATRIX AND        BLI01210
C      FLAG JPERM.  JPERM = (0,1):  JPERM = 1 MEANS MATRIX HAS BEEN   BLI01220
C      PERMUTED                                                       BLI01230
       READ(5,20) EXPLAN                                              BLI01240
       READ(5,*) N,MATNO,S0,SHIFT,JPERM                               BLI01250
C                                                                     BLI01260
C      READ USER-SPECIFIED DIMENSIONS OF Q-ARRAY (MDIMQ) AND OF THE   BLI01270
C      TM-ARRAY (MDIMTM).  READ MAXIMUM NUMBER (MAXIT) OF CALLS TO THE BLI01280
C      BLSOLV SUBROUTINE ALLOWED IN PHASE 1.                          BLI01290
       READ(5,20) EXPLAN                                              BLI01300
       READ(5,*) MDIMQ, MDIMTM, MAXIT                                 BLI01310
C                                                                     BLI01320
C      READ FLAGS:   EFLAG = (0,1).  EFLAG = 0, MEANS PROGRAM STOPS   BLI01330
C      AFTER COMPLETING PHASE 1 PORTION OF BLOCK LANCZOS PROCEDURE.   BLI01340
C      EFLAG = 1, MEANS PROGRAM COMPLETES BOTH PHASES BEFORE          BLI01350
C      TERMINATING.                                                   BLI01360
C      OFLAG = (0,1).  OFLAG = 0, MEANS THAT IN PHASE 1 PORTION       BLI01370
C      OF THE COMPUTATION, THE PROGRAM DOES NO ORTHOGONALITY CHECKS   BLI01380
C      ON THE Q-BLOCKS GENERATED.  OFLAG = 1 MEANS THAT IN THE        BLI01390
C      PHASE 1 PORTION AND IN THE PHASE 2 PORTIONS OF THE COMPUTATIONS BLI01400
C      THE PROGRAM CHECKS THE ORTHOGONALITY OF THE Q-BLOCKS GENERATED BLI01410
C      W.R.T. THAT VECTOR IN THE FIRST BLOCK THAT IS GENERATING       BLI01420
C      DESCENDANTS.  NOTE THAT IN PHASE 2, THE PROGRAM ALWAYS MAKES   BLI01430
C      THIS CHECK OF ORTHGONALITY REGARDLESS OF THE VALUE OF OFLAG.   BLI01440
C      FOR SAFETY, OFLAG SHOULD ALWAYS BE SET TO 1, ALTHOUGH FOR MANY BLI01450
C      PROBLEMS THIS IS NOT NECESSARY.                                BLI01460
       READ(5,20) EXPLAN                                              BLI01470
       READ(5,*) EFLAG,OFLAG                                          BLI01480
C                                                                     BLI01490
C      READ SEED USED BY SUBROUTINE GENRAN TO OBTAIN THOSE STARTING   BLI01500
C      VECTORS WHICH ARE GENERATED RANDOMLY.                          BLI01510
       READ(5,20) EXPLAN                                              BLI01520
       READ(5,*) SEED                                                 BLI01530
C                                                                     BLI01540
```

```
C       SPECIFY MAXIMUM T-SIZE ALLOWED (KMAX-1); INITIAL SIZE OF         BLI01550
C       STARTING BLOCK (KACT);  NUMBER OF STARTING VECTORS SUPPLIED (KSET)BLI01560
C       SEE BLOCK LANCZOS HEADER FOR COMMENTS REGARDING THE SIZE OF KACT. BLI01570
        READ(5,20) EXPLAN                                                BLI01580
        READ(5,*) KMAX,KACT,KSET                                         BLI01590
C                                                                        BLI01600
C       SPECIFY NUMBER (KM) OF EXTREME EIGENVALUES AND EIGENVECTORS      BLI01610
C       OF B-INVERSE TO BE COMPUTED.  THE BLOCK PROCEDURE WORKS WITH THE BLI01620
C       INVERSE OF THE MATRIX B = S0*P*A*P' + SHIFT*I, USING A           BLI01630
C       FACTORIZATION OF B.  TO INDICATE THAT THE ALGEBRAICALLY-         BLI01640
C       SMALLEST EIGENVALUES OF B-INVERSE ARE BEING COMPUTED SET KM < 0. BLI01650
C       IF KM < 0, THE PROGRAM ASSUMES THAT BLSOLV SUBROUTINE WHICH      BLI01660
C       THE USER HAS PROVIDED IS COMPUTING -(B-INVERSE)*X               BLI01670
C       INSTEAD OF (B-INVERSE)*X AND INTERNALLY IT COMPUTES THE |KM|     BLI01680
C       ALGEBRAICALLY-LARGEST EIGENVALUES OF -(B-INVERSE).              BLI01690
        READ(5,20) EXPLAN                                                BLI01700
        READ(5,*) KM                                                     BLI01710
        IF(KM.EQ.0) GO TO 540                                            BLI01720
        KML = IABS(KM)                                                   BLI01730
C                                                                        BLI01740
C       STAGNATION OF CONVERGENCE OF THE KM-TH EIGENVALUE WILL BE        BLI01750
C       TESTED AFTER NSTAG ITERATIONS.  CONVERGENCE WILL BE SAID TO      BLI01760
C       HAVE STAGNATED IF THE RATIO OF THE SQUARE OF THE CURRENT KM-TH   BLI01770
C       RESIDUAL TO THE SQUARE OF THE CORRESPONDING RESIDUAL OBTAINED    BLI01780
C       10 ITERATIONS EARLIER IS GREATER THAN FRACT.  NSTAG SHOULD BE    BLI01790
C       >= 25.  FRACT WAS SET EQUAL TO .01 IN THE TESTS.                 BLI01800
        READ(5,20) EXPLAN                                                BLI01810
        READ(5,*) NSTAG, FRACT                                           BLI01820
C                                                                        BLI01830
C       READ IN THE RELATIVE TOLERANCE (RELTOL) USED TO DETERMINE A      BLI01840
C       CONVERGENCE CRITERION FOR PHASE 2, AND THE MAXIMUM NUMBER (MAXIT2)BLI01850
C       OF CALLS TO SUBROUTINE BLSOLV ALLOWED IN PHASE 2.               BLI01860
        READ(5,20) EXPLAN                                                BLI01870
        IF(EFLAG.EQ.1)  READ(5,*)  RELTOL, MAXIT2                        BLI01880
C                                                                        BLI01890
C       CONSISTENCY CHECKS                                              BLI01900
C       PROCEDURE REQUIRES ENOUGH ROOM IN THE Q-ARRAY FOR AT LEAST 2     BLI01910
C       BLOCKS OF SIZE KACT PLUS A WORKING VECTOR OF LENGTH N.           BLI01920
        MXBLK = KMAX -1                                                  BLI01930
        MXBLK2 = MXBLK*MXBLK                                             BLI01940
        IF(MDIMTM.LT.MXBLK2) GO TO 520                                   BLI01950
        NKMAX = N*KMAX                                                   BLI01960
        IF(MDIMQ.LT.NKMAX)   GO TO 560                                   BLI01970
        IF(KML.GT.KACT) GO TO 420                                        BLI01980
        IF(MXBLK.GT.N)  GO TO 440                                        BLI01990
        IF(2*KACT.GT.MXBLK)  GO TO 500                                   BLI02000
C                                                                        BLI02010
C------------------------------------------------------------------------BLI02020
C       DEFINE AND INITIALIZE THE ARRAYS NEEDED TO DEFINE THE           BLI02030
C       FACTORIZATION OF THE B-MATRIX.  PASS THE STORAGE LOCATIONS       BLI02040
C       OF THESE ARRAYS TO THE SUBROUTINE BLSOLV.                       BLI02050
C                                                                        BLI02060
        CALL USPEC(N,MATNO,NNZ,AVER)                                     BLI02070
C                                                                        BLI02080
C------------------------------------------------------------------------BLI02090
```

```
C     MASK OVERFLOW AND UNDERFLOW                                BLI02100
      CALL MASK                                                  BLI02110
C                                                                BLI02120
C------------------------------------------------------------------BLI02130
C     ARE THERE STARTING VECTORS TO READ IN FROM FILE 10 (KSET.NE.0) ? BLI02140
      IF(KSET.EQ.0) GO TO 70                                     BLI02150
C                                                                BLI02160
      READ(10,30) NOLD,KACT                                      BLI02170
   30 FORMAT(I6,I4)                                              BLI02180
      IF(NOLD.NE.N.OR.KSET.GT.KACT) GO TO 460                    BLI02190
      DO 50 J=1,KSET                                             BLI02200
      READ(10,20) EXPLAN                                         BLI02210
      READ(10,40) EVAL,RESID                                     BLI02220
   40 FORMAT(E20.12,E13.4)                                       BLI02230
      READ(10,20) EXPLAN                                         BLI02240
      LINT= (J-1)*N + 1                                          BLI02250
      LFIN = J*N                                                 BLI02260
   50 READ(10,60) (Q(JL), JL = LINT,LFIN)                        BLI02270
   60 FORMAT(4E20.12)                                            BLI02280
C                                                                BLI02290
   70 CONTINUE                                                   BLI02300
C                                                                BLI02310
C     WRITE TO A SUMMARY OF THE PARAMETERS FOR THIS RUN TO FILE 6 BLI02320
C                                                                BLI02330
      MXBLK = KMAX - 1                                           BLI02340
      WRITE(6,80)  N, NNZ, AVER, MATNO                           BLI02350
   80 FORMAT(/4X,'ORDER OF B-MATRIX ',5X,'AVERAGE NUMBER NONZEROES PER RBLI02360
     10W IN FACTOR'/                                             BLI02370
     1I15,E47.4/3X,'CRUDE ESTIMATE OF SIZE NONZERO ENTRIES',5X,'MATRIX IBLI02380
     1D'/E31.4,I21/)                                             BLI02390
C                                                                BLI02400
      WRITE(6,90)  S0, SHIFT                                     BLI02410
   90 FORMAT(/4X,'SCALE USED ON A-MATRIX',5X,'SHIFT USED ON A-MATRIX'/  BLI02420
     1E26.4,E27.4/)                                              BLI02430
C                                                                BLI02440
      WRITE(6,100) MDIMQ, MDIMTM                                 BLI02450
  100 FORMAT(/18X,'USER-SPECIFIED'/2X,'MAX. DIMENSION Q-ARRAY',4X,'MAX. BLI02460
     1DIMENSION TM-ARRAY'/I16,I26/)                              BLI02470
C                                                                BLI02480
      WRITE(6,110)  OFLAG, EFLAG                                 BLI02490
  110 FORMAT(/4X,'OFLAG',4X,'EFLAG'/I8,I9/)                      BLI02500
C                                                                BLI02510
      IF(OFLAG.EQ.1)  WRITE(6,120)  SPREC                        BLI02520
  120 FORMAT(/4X,'ORTHOGONALITY TEST TOLERANCE'/E25.2)           BLI02530
C                                                                BLI02540
      IF(EFLAG.EQ.1)  WRITE(6,130) MAXIT,RELTOL,MAXIT2           BLI02550
  130 FORMAT(/4X,' MAXIT ',8X,'  RELTOL  ',6X,' MAXIT2 '/I10,E20.6,I12/)BLI02560
      IF(EFLAG.EQ.0)  WRITE(6,140) MAXIT                         BLI02570
  140 FORMAT(/4X,' MAXIT '/I10/)                                 BLI02580
C                                                                BLI02590
      WRITE(6,150)  SEED                                         BLI02600
  150 FORMAT(/' SEED FOR RANDOM NUMBER GENERATOR'/I24/)          BLI02610
C                                                                BLI02620
      IF(KM.GT.0)  WRITE(6,160) KML                              BLI02630
  160 FORMAT(/' COMPUTE THE',I3,' ALGEBRAICALLY-LARGEST EIGENVALUES AND BLI02640
```

```
      1CORRESPONDING VECTORS'/' OF THE INVERSE OF B =  (SO*P*A*P-TRANS + BLI02650
      1HIFT*I)'/)                                                   BLI02660
       IF(KM.LT.0)  WRITE(6,170) KML                               BLI02670
  170 FORMAT(/' COMPUTE THE',I3,' ALGEBRAICALLY-SMALLEST EIGENVALUES ANDBLI02680
      1 CORRESPONDING VECTORS'/' OF THE INVERSE OF THE MATRIX B = (SO*P*ABLI02690
      1*P-TRANS + SHIFT*I).'/' PROGRAM ASSUMES THAT USER IS PROVIDING -(BBLI02700
      1-INVERSE)*X INSTEAD OF (B-INVERSE)*X'/' AND COMPUTES THE ALGEBRAICBLI02710
      1ALLY-LARGEST EIGENVALUES OF -(B-INVERSE).'/' HOWEVER ON EXIT, FILEBLI02720
      1 15 CONTAINS THE ALGEBRAICALLY-SMALLEST EIGENVALUES'/' OF B-INVERSBLI02730
      1E, THE CORRESPONDING EIGENVALUES OF THE ORIGINAL A-MATRIX'/' AND TBLI02740
      1HE CORRESPONDING EIGENVECTORS OF A.'/)                       BLI02750
C                                                                   BLI02760
C     NOTE THAT THE ESTIMATE FOR AVER IN THE INVERSE CASE IS VERY CRUDE BLI02770
C     COMPUTE PHASE 1 CONVERGENCE TOLERANCE                         BLI02780
       IF(AVER.GE.1.)                                               BLI02790
      1ERRMAX = 2.D0*DFLOAT(N+1000)*NNZ*AVER*MACHEP                 BLI02800
       IF(AVER.LT.1.)                                               BLI02810
      1ERRMAX = 2.D0*DFLOAT(N+1000)*NNZ*AVER**2*MACHEP             BLI02820
C                                                                   BLI02830
      WRITE(6,180) KACT,MXBLK,KSET                                  BLI02840
  180 FORMAT(/' ON INITIAL ITERATIONS, THE FIRST BLOCK CONTAINS ',I3,' VBLI02850
      1ECTORS'/' HOWEVER THE SIZE OF THE FIRST BLOCK MAY CHANGE AS THE ITBLI02860
      1ERATIONS PROCEED'/' THE MAXIMUM SIZE T-MATRIX THAT CAN BE GENERATEBLI02870
      1D IS ',I4/' THE USER SUPPLIED ',I3,' STARTING VECTORS'/)     BLI02880
C                                                                   BLI02890
      WRITE(6,190)                                                  BLI02900
  190 FORMAT(/' ITERATIVE PROCEDURE'/' PROCEDURE MONITORS THE SIZES OF TBLI02910
      1HE NORM(GRADIENTS)**2 ON EACH'/' ITERATION.  CONVERGENCE IS SAID  BLI02920
      1TO HAVE OCCURRED WHEN ALL'/' RELEVANT (NORMS)**2 ARE LESS THAN ERRBLI02930
      1MAX',E10.3/' PHASE 1 ERRMAX MAY YIELD SOMEWHAT LESS THAN SINGLE PRBLI02940
      1ECISION ACCURACY.'/'  PHASE 2 REFINES THE VECTORS OBTAINED ON PHASBLI02950
      1E 1, ACCORDING TO'/' THE ACCURACY SPECIFIED BY THE USER'/)    BLI02960
C                                                                   BLI02970
      WRITE(6,200) ERRMAX                                           BLI02980
  200 FORMAT(//' PHASE 1 CONVERGENCE CRITERION, ERRMAX '/E22.3/)    BLI02990
C                                                                   BLI03000
C-------------------------------------------------------------------BLI03010
C     PASS STORAGE LOCATIONS OF VARIOUS ARRAYS TO LANCZS AND LANCI1  BLI03020
C     SUBROUTINES                                                    BLI03030
C                                                                   BLI03040
      CALL LANZP(DIR,DESC,SM,TM,TOD,TD,G,XLFT,LEFT,SPREC)           BLI03050
      CALL LANCP1(DIR,DESC,TM,SM,XLFT,LEFT)                         BLI03060
C                                                                   BLI03070
C-------------------------------------------------------------------BLI03080
C                                                                   BLI03090
C     ENTER PHASE 1 OF BLOCK LANCZOS PROCEDURE.  BLOCK PROCEDURE     BLI03100
C     HAS 2 POSSIBLE PHASES.  USER SPECIFIES PHASE 1 ONLY OR PHASE 1 BLI03110
C     AND PHASE 2 BY SETTING EFLAG = 0 OR 1, RESPECTIVELY. PHASE 1   BLI03120
C     COMPUTES VECTORS THAT ARE USUALLY ACCURATE TO SINGLE PRECISION. BLI03130
C     PHASE 2 TAKES THE VECTORS OBTAINED IN PHASE 1 AND REFINES THEM. BLI03140
C     THE USER SPECIFIES THE DEGREE OF REFINEMENT DESIRED BY SELECTING BLI03150
C     THE VALUE OF RELTOL AND MAXIT2.  BOTH PHASES SHOULD BE USED.   BLI03160
       IPHASE = 1                                                   BLI03170
       NITER = 0                                                    BLI03180
  210 ITER = 0                                                      BLI03190
```

```
      RESIDL(1) = FRACT                                          BLI03200
      RESIDL(2) = NSTAG                                          BLI03210
C                                                                BLI03220
C----------------------------------------------------------------BLI03230
C     CALL INITIATES THE BLOCK LANCZOS PROCEDURE.               BLI03240
C     ON RETURN EIGENVALUE APPROXIMATIONS ARE IN E(I),          BLI03250
C     I = 1,KACT, IN ALGEBRAICALLY DECREASING ORDER.   CORRESPONDING  BLI03260
C     EIGENVECTOR APPROXIMATIONS ARE IN FIRST N*KACT LOCATIONS IN  BLI03270
C     THE Q-ARRAY.                                              BLI03280
C                                                                BLI03290
      CALL LANCZS(BLSOLV,KML,KSET,KACT,MXBLK,N,Q,E,RESIDL,RESIDK,ERRMAX,BLI03300
     1 IPHASE,NITER,IWRITE)                                      BLI03310
C                                                                BLI03320
C----------------------------------------------------------------BLI03330
C                                                                BLI03340
      IF(IPHASE.EQ.MPMIN) WRITE(15,220) N,KACT                   BLI03350
  220 FORMAT(2I10,' PHASE 2 TERMINATED '/' PROGRAM INDICATES ACCURACY SPBLI03360
     1ECIFIED BY USER IS NOT ACHIEVABLE'/)                       BLI03370
C                                                                BLI03380
      ITERA = IABS(ITER)                                         BLI03390
      IF(IWRITE.NE.MPMIN.AND.ITER.GT.0)  WRITE(6,230) IPHASE,ITERA  BLI03400
  230 FORMAT(/1X,'PHASE COMPLETED',5X,' NUMBER CALLS TO BLSOLV SUBROUTINBLI03410
     1E USED'/I10,I32)                                           BLI03420
C                                                                BLI03430
      IF(IWRITE.EQ.MPMIN.OR.ITER.LT.0)  WRITE(6,240) IPHASE,ITERA  BLI03440
  240 FORMAT(/1X,'PHASE TERMINATED',5X,' NUMBER CALLS TO BLSOLV SUBROUTIBLI03450
     1NE USED'/I10,I32)                                          BLI03460
C                                                                BLI03470
      IF(ITER.GT.0.AND.IWRITE.NE.MPMIN)  GO TO 270               BLI03480
C                                                                BLI03490
      IF(ITER.LT.0)  WRITE(6,250)                                BLI03500
  250 FORMAT(//' SMALL EIGENVALUE SUBROUTINE DEFAULTED'/' BLOCK LANCZOS BLI03510
     1 PROCEDURE STOPS AFTER SAVING CURRENT EIGENVECTOR APPROXIMATIONS'/BLI03520
     1/)                                                         BLI03530
C                                                                BLI03540
      WRITE(15,260)                                              BLI03550
      WRITE(6,260)                                               BLI03560
  260 FORMAT(//' BLOCK LANCZOS PROCEDURE TERMINATES WITHOUT CONVERGENCE BLI03570
     1'/' USER SHOULD EXAMINE OUTPUT TO DETERMINE REASONS FOR TERMINATIOBLI03580
     1N'//)                                                      BLI03590
C                                                                BLI03600
C     WRITE EIGENVALUE AND EIGENVECTOR APPROXIMATIONS CONTAINED IN  BLI03610
C     THE FIRST Q-BLOCK TO FILE 15                              BLI03620
C                                                                BLI03630
  270 IF(IPHASE.EQ.1)  WRITE(15,280) N,KACT,SEED                 BLI03640
  280 FORMAT(I6,I4,I12,' PHASE 1, ORDER A-MATRIX, SIZE OF Q(1), SEED')  BLI03650
      IF(IPHASE.EQ.2)  WRITE(15,290) N,KACT,SEED                 BLI03660
  290 FORMAT(I6,I4,I12,' PHASE 2, ORDER A-MATRIX, SIZE OF Q(1), SEED')  BLI03670
C                                                                BLI03680
C     PERMUTE THE EIGENVECTORS IF NECESSARY                     BLI03690
      IF(JPERM.EQ.0) GO TO 310                                   BLI03700
      LINT = -N + 1                                              BLI03710
      KACT1 = KACT*N + 1                                         BLI03720
      DO 300 J = 1,KACT                                          BLI03730
      LINT = LINT + N                                            BLI03740
```

```
C-----------------------------------------------------------------------BLI03750
      IPERM = 2                                               BLI03760
      CALL LPERM(Q(LINT),Q(KACT1),IPERM)                      BLI03770
C-----------------------------------------------------------------------BLI03780
  300 CONTINUE                                                 BLI03790
C                                                              BLI03800
C     COMPUTE THE EIGENVALUES OF THE A-MATRIX                  BLI03810
  310 DO 320 J = 1,KACT                                        BLI03820
      IF(KM.LT.0)   E(J) = -E(J)                               BLI03830
      TD(J) = 1.D0/E(J)                                        BLI03840
  320 TD(J) = (TD(J) - SHIFT)/S0                               BLI03850
C                                                              BLI03860
C     NOTE THAT RESIDUAL PRINTED OUT CORRESPONDS TO VALUE OBTAINED BLI03870
C     PRIOR TO FINAL PROJECTION Q(1)-TRANSPOSE*AQ(1) DONE BEFORE  BLI03880
C     TERMINATION                                              BLI03890
      JJ=KACT                                                  BLI03900
      LINT = -N + 1                                            BLI03910
      LFIN = 0                                                 BLI03920
      DO 340 J=1,KACT                                          BLI03930
      LINT = LINT + N                                          BLI03940
      LFIN = LFIN + N                                          BLI03950
      JJ=JJ+1                                                  BLI03960
C                                                              BLI03970
C     NOTE THAT RESIDUAL PRINTED OUT CORRESPONDS TO VALUE OBTAINED BLI03980
C     PRIOR TO FINAL PROJECTION Q(1)-TRANSPOSE*(B-INVERS)*Q(1) DONE BLI03990
C     BEFORE TERMINATION                                       BLI04000
C                                                              BLI04010
      WRITE(15,330) E(J), SM(JJ),TD(J)                         BLI04020
  330 FORMAT(/E20.12,E13.4,E20.12,'BI-EVAL,ER**2,A-EVAL,A-EVEC'/) BLI04030
  340 WRITE(15,350) (Q(L), L=LINT,LFIN)                        BLI04040
      WRITE(15,360)                                            BLI04050
  350 FORMAT(4E20.12)                                          BLI04060
  360 FORMAT(/' ABOVE ARE COMPUTED APPROXIMATE EIGENVECTORS'/) BLI04070
C                                                              BLI04080
      IF(ITER.GT.MAXIT) WRITE(15,370) ITER,MAXIT              BLI04090
  370 FORMAT(//' PROCEDURE TERMINATED BECAUSE NUMBER OF CALLS TO BLSOLV BLI04100
     1 SUBROUTINE',I6/' EXCEEDED MAXIMUM NUMBER ',I6,' ALLOWED'//) BLI04110
C                                                              BLI04120
      IF(ITER.LT.0) WRITE(15,380)                             BLI04130
  380 FORMAT(//' USER BEWARE.  EIGENELEMENT COMPUTATIONS DEFAULTED BECAUBLI04140
     1SE'/'  EISPACK SUBROUTINE DEFAULTED.  EIGENVALUE AND EIGENVECTORBLI04150
     1 APPROXIMATIONS'/'  ABOVE WERE THOSE AVAILABLE AT THE TIME OF DEFBLI04160
     1AULT'/' SOMETHING IS SERIOUSLY WRONG.'//)               BLI04170
C                                                              BLI04180
C     CHECK FOR TERMINATION AFTER PHASE 1                      BLI04190
C     ITER < 0 MEANS EISPACK SUBROUTINE DEFAULTED              BLI04200
C     IPHASE = MPMIN MEANS THAT PHASE 2 TERMINATED DUE TO ORTHOGONALITY BLI04210
C     IWRITE = MPMIN MEANS THAT CONVERGENCE APPEARS TO HAVE STAGNATED BLI04220
C     ITER > MAXIT MEANS MAXIMUM NUMBER OF CALLS TO BLSOLV     BLI04230
C            ALLOWED BY USER WAS EXCEEDED                      BLI04240
      IF(ITER.LT.0.OR.ITER.GT.MAXIT) GO TO 580                BLI04250
      IF(IPHASE.EQ.MPMIN.OR.IWRITE.EQ.MPMIN) GO TO 580        BLI04260
      IF(EFLAG.NE.1.OR.IPHASE.EQ.2) GO TO 580                 BLI04270
C                                                              BLI04280
C     ENTER 2ND PHASE OF COMPUTATION TO ATTEMPT TO OBTAIN MORE BLI04290
```

```
C     ACCURATE EIGENVECTOR APPROXIMATIONS.                     BLI04300
C     USER CONTROLS THE SIZE OF THE ERROR TOLERANCE BY SPECIFYING   BLI04310
C     THE PARAMETER RELTOL.                                    BLI04320
C                                                              BLI04330
      IPHASE = 2                                               BLI04340
      MAXIT = MAXIT2                                           BLI04350
      KSET = KACT                                              BLI04360
C                                                              BLI04370
C     ERROR TOLERANCE USES THE CONVERGED EIGENVALUE LARGEST IN     BLI04380
C     MAGNITUDE.                                               BLI04390
      TD(1) = DABS(E(1))                                       BLI04400
      IF(KML.EQ.1) GO TO 400                                   BLI04410
      DO 390 J = 2,KML                                         BLI04420
  390 IF(DABS(E(J)).GT.TD(1))    TD(1) = DABS(E(J))            BLI04430
  400 TD(1) = DMAX1(TD(1),1.D0)                                BLI04440
      ERRMAN = RELTOL**2 * TD(1)**2                            BLI04450
      IF(ERRMAN.GE.ERRMAX) GO TO 480                           BLI04460
      ERRMAX = ERRMAN                                          BLI04470
C                                                              BLI04480
      WRITE(6,410) ERRMAX, MAXIT2                              BLI04490
  410 FORMAT(//' ENTER PHASE 2 OF COMPUTATION'/' CONVERGENCE CRITERION IBLI04500
     1S REDUCED TO ',E13.4/' NO MORE THAN ',I5,' CALLS TO SUBROUTINE BLSBLI04510
     10LV WILL BE ALLOWED.'/' PROGRAM WILL TERMINATE IF BLOCK ORTHGONALIBLI04520
     1TY PROBLEMS MATERIALIZE'/)                               BLI04530
C                                                              BLI04540
      GO TO 210                                                BLI04550
C                                                              BLI04560
C     INCONSISTENCIES IN THE DATA                              BLI04570
C                                                              BLI04580
  420 WRITE(6,430) KM,KACT                                     BLI04590
  430 FORMAT(/' PROGRAM TERMINATES BECAUSE THE NUMBER OF EIGENELEMENTS  BLI04600
     1REQUESTED, KM =',I3/'  IS LARGER THAN THE SIZE OF THE FIRST Q BLOCBLI04610
     1K, KACT =',I3,' SPECIFIED'/'  USER MUST RESET KM OR KACT'/)   BLI04620
      GO TO 580                                                BLI04630
C                                                              BLI04640
  440 WRITE(6,450) KMAX,N                                      BLI04650
  450 FORMAT(/' PROGRAM TERMINATES BECAUSE KMAX = ',I5,' IS TOO LARGE FOBLI04660
     1R THE SIZE, N = ',I5,', OF THE GIVEN MATRIX'/'  USER MUST DECREASEBLI04670
     1THE SIZE OF KMAX.'/)                                     BLI04680
      GO TO 580                                                BLI04690
C                                                              BLI04700
  460 WRITE(6,470) NOLD,N,KACT,KSET                            BLI04710
  470 FORMAT(/' PROGRAM TERMINATES BECAUSE FAULT OCCURRED IN READING IN BLI04720
     1THE EIGENVECTOR APPROXIMATIONS'/' EITHER THE SIZE MATRIX SPECIFIEDBLI04730
     10N THE EIGENVECTOR FILE' ,I6/'  DID NOT MATCH THE SIZE SPECIFIED 'BLI04740
     1,I5,' IN THE PROGRAM OR THE NUMBER'/'   OF VECTORS IN FILE 10 = 'BLI04750
     1,I4,' IS LESS THAN THE NUMBER ',I3/'   USER SAID WERE THERE'/)   BLI04760
      GO TO 580                                                BLI04770
C                                                              BLI04780
  480 WRITE(6,490)  ERRMAN, ERRMAX                             BLI04790
  490 FORMAT(/' COMPUTED PHASE 2 CONVERGENCE CRITERION ',E13.4/'  IS LARBLI04800
     1GER THAN PHASE 1 CRITERION ',E13.4/'  SO PROGRAM TERMINATES'/)   BLI04810
      GO TO 580                                                BLI04820
C                                                              BLI04830
  500 WRITE(6,510)  KACT,MXBLK                                 BLI04840
```

```
  510 FORMAT(/' PROGRAM TERMINATES BECAUSE THERE IS NOT ENOUGH ROOM TO  BLI04850
     1GENERATE 2 BLOCKS',' 	 BECAUSE KACT = ',I3,' AND MXBLK = ', I4/)   BLI04860
        GO TO 580                                                        BLI04870
C                                                                        BLI04880
C                                                                        BLI04890
  520 WRITE(6,530)  MDIMTM, MXBLK                                        BLI04900
  530 FORMAT(/' PROGRAM TERMINATES BECAUSE THE DIMENSION ',I6,' OF THE TBLI04910
     1M ARRAY'//   IS TOO SMALL FOR THE LARGEST T-MATRIX ALLOWED ',I4)   BLI04920
        GO TO 580                                                        BLI04930
C                                                                        BLI04940
  540 WRITE(6,550)                                                       BLI04950
  550 FORMAT(/' USER SPECIFIED NUMBER OF EIGENVALUES OF INTEREST AS 0'//BLI04960
     1 PROGRAM TERMINATES FOR USER TO RESET KM TO DESIRED NONZERO VALUE'BLI04970
     1/)                                                                 BLI04980
        GO TO 580                                                        BLI04990
C                                                                        BLI05000
  560 WRITE(6,570)  MDIMQ, KMAX,N                                        BLI05010
  570 FORMAT(/' PROGRAM TERMINATES BECAUSE THE DIMENSION ',I6,' OF THE QBLI05020
     1-ARRAY'//   IS TOO SMALL TO HOLD ',I5, ' VECTORS OF LENGTH ',I4)   BLI05030
        GO TO 580                                                        BLI05040
C                                                                        BLI05050
  580 CONTINUE                                                           BLI05060
C                                                                        BLI05070
      STOP                                                               BLI05080
C-----END OF MAIN PROGRAM FOR INVERSE BLOCK LANCZOS PROCEDURE-----------BLI05090
      END                                                                BLI05100
```

9.3 BLIMULT: Sample Matrix-Vector Multiply Subroutines

```
C---BLIMULT-(INVERSES OF REAL SYMMETRIC MATRICES)----------------------BLI00010
C  Authors:  Jane Cullum* and Bill Donath**                           BLE00020
C            **IBM Research, T.J. Watson Research Center              BLE00030
C            **Yorktown Heights, N.Y. 10598                           BLE00040
C            * Los Alamos National Laboratory                         BLE00050
C            * Los Alamos, New Mexico 87544                           BLE00060
C              E-mail:  cullumj@lanl.gov                              BLE00065
C                                                                     BLI00070
C  These codes are copyrighted by the authors.  These codes          BLI00080
C  and modifications of them or portions of them are NOT to be        BLI00090
C  incorporated into any commercial codes or used for any other       BLI00100
C  commercial purposes such as consulting for other companies,        BLI00110
C  without legal agreements with the authors of these Codes.          BLI00120
C  If these Codes or portions of them are used in other scientific or BLI00130
C  engineering research works the names of the authors of these codes BLI00140
C  and appropriate references to their written work are to be         BLI00150
C  incorporated in the derivative works.                              BLI00160
C                                                                     BLI00170
C  This header is not to be removed from these codes.                 BLI00180
C                                                                     BLI00190
C        REFERENCE: Cullum and Willoughby, Chapter 7,                 BLI00191
C        Lanczos Algorithms for Large Symmetric Eigenvalue ComputationsBLI00192
C        VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in  BLI00193
C        Applied Mathematics, 2002. SIAM Publications,                BLI00194
C        Philadelphia, PA. USA                                        BLI00195
C                                                                     BLI00200
C     CONTAINS SUBROUTINES  LANCZS AND SAMPLE USPEC AND BLSOLV        BLI00210
C     USED BY THE VERSION OF THE BLOCK LANCZOS ALGORITHMS FOR         BLI00220
C     FACTORED INVERSES OF REAL SYMMETRIC MATRICES, BLIVAL.           BLI00230
C     NOTE THAT SAMPLE BLSOLV FOR SPARSE MATRICES ASSUMES THAT        BLI00240
C     B-MATRIX IS POSITIVE DEFINITE AND USES CHOLESKY FACTORS.        BLI00250
C     HOWEVER, THE USER CAN DIRECTLY REPLACE THAT SUBROUTINE BY       BLI00260
C     A SUBROUTINE FOR INDEFINITE MATRICES THAT COMPUTES THE          BLI00270
C     GENERALIZED FACTORIZATION L*D*(L-TRANSPOSE).                    BLI00280
C                                                                     BLI00290
C     NONPORTABLE CONSTRUCTIONS:                                      BLI00300
C     1.  THE ENTRY MECHANISM USED TO PASS THE STORAGE LOCATIONS      BLI00310
C         OF THE FACTORIZATION OF THE MATRIX THAT WILL BE USED        BLI00320
C         BY THE LANCZS SUBROUTINE TO THE SUBROUTINE BLSOLV.          BLI00330
C     2.  IN THE SAMPLE USPEC AND BLSOLV SUBROUTINES PROVIDED:        BLI00340
C         THE FREE FORMAT (7,*), THE FORMAT (20A4) USED FOR           BLI00350
C         READING EXPLANATORY COMMENTS IN THE MATRIX SPECIFICATION    BLI00360
C         FILES, AND THE HEX FORMAT (4Z20) USED IN THE USPECS.        BLI00370
C     3.  THE COMMON BLOCK: LOOPS                                     BLI00380
C                                                                     BLI00390
C-----USPEC FOR FACTORED INVERSES OF REAL SYMMETRIC MATRICES------------BLI00400
C                                                                     BLI00410
      SUBROUTINE CUSPEC(N,MATNO,NNZ,AVER)                             BLI00420
C     SUBROUTINE USPEC(N,MATNO,NNZ,AVER)                              BLI00430
C                                                                     BLI00440
C----------------------------------------------------------------------BLI00450
```

```
      DOUBLE PRECISION  BD(2200),BSD(10000),NNZ,AVER             BLI00460
      INTEGER  KCOL(2200),KROW(10000),IPR(2200),IPT(2200)        BLI00470
C----------------------------------------------------------------BLI00480
C     THIS SAMPLE SUBROUTINE ASSUMES THAT B IS POSITIVE DEFINITE  BLI00490
C     USER COULD REPLACE BY SIMILAR SUBROUTINE FOR GENERAL FACTORIZATIONBLI00500
C     DIMENSIONS ARRAYS NEEDED TO DEFINE CHOLESKY FACTOR OF B-MATRIX,  BLI00510
C     READS CHOLESKY FACTOR FROM FILE 7, AND THEN PASSES STORAGE  BLI00520
C     LOCATIONS OF THESE ARRAYS TO THE B-MATRIX SOLVE SUBROUTINE BLSOLV.BLI00530
C                                                                 BLI00540
C     HERE WE HAVE B = P*C*P' = L*L'  WHERE  C = S0*A + SHIFT*I.  BLI00550
C     P IS A PERMUTATION MATRIX DEFINED BY THE VECTOR MAPS IPR AND IPT. BLI00560
C     THE ITH ROW OF B CORRESPONDS TO THE JTH ROW OF C (A) WHERE  BLI00570
C     J = IPR(I) AND I = IPT(J).                                 BLI00580
C                                                                 BLI00590
C     THE B-CHOLESKY FACTOR IS STORED IN THE FOLLOWING SPARSE FORMAT:  BLI00600
C     N = ORDER OF THE B-MATRIX.                                 BLI00610
C     NZT = NUMBER OF NONZERO SUBDIAGONAL ENTRIES IN THE CHOLESKY  BLI00620
C           FACTOR, L.                                           BLI00630
C     KCOL(J), J=1,N IS THE NUMBER OF NONZERO SUBDIAGONAL ELEMENTS IN  BLI00640
C             COLUMN J OF L.                                     BLI00650
C     KROW(K), K=1,NZT IS THE ROW INDEX FOR CORRESPONDING ENTRY BSD(K). BLI00660
C     BD(J), J = 1,N CONTAINS THE DIAGONAL ENTRIES OF L.         BLI00670
C     BSD(K), K =1,NZT CONTAINS THE NONZERO SUBDIAGONAL ENTRIES OF L  BLI00680
C     JPERM = (0,1):  1 MEANS CHOLEKSY FACTOR CORRESPONDS TO     BLI00690
C                     PERMUTED C.  0 MEANS NO PERMUTATION WAS USED.  BLI00700
C----------------------------------------------------------------BLI00710
C     READ CHOLESKY FACTOR FROM FILE 7.  MUST BE STORED          BLI00720
C     IN SPARSE MATRIX FORMAT.                                   BLI00730
      READ(7,10) NZT,NOLD,NZL,MATOLD,JPERM                       BLI00740
   10 FORMAT(I10,2I6,I8,I6)                                      BLI00750
C                                                                 BLI00760
      WRITE(6,20) NZT,NZL,N,NOLD,MATOLD,JPERM                    BLI00770
   20 FORMAT(' HEADER, CHOLESKY FACTOR FILE'/                    BLI00780
     1 3X,'NZT',3X,'NZL',5X,'N',2X,'NOLD',2X,'MATOLD',1X,'JPERM'/  BLI00790
     1 4I6,I8,I6/)                                               BLI00800
C                                                                 BLI00810
      IF (N.NE.NOLD.OR.MATNO.NE.MATOLD) GO TO 100                BLI00820
C                                                                 BLI00830
      READ(7,30) (KCOL(K), K = 1,NZL)                            BLI00840
      READ(7,30) (KROW(K), K = 1,NZT)                            BLI00850
   30 FORMAT(13I6)                                               BLI00860
      READ(7,40) (BD(K), K = 1,N)                                BLI00870
      READ(7,40) (BSD(K), K = 1,NZT)                             BLI00880
   40 FORMAT(4Z20)                                               BLI00890
C  20 FORMAT(3E25.16)                                            BLI00900
C                                                                 BLI00910
C     DOES CHOLESKY FACTOR CORRESPOND TO PERMUTED B?             BLI00920
      IF(JPERM.EQ.0)  GO TO 60                                   BLI00930
      READ(7,30) (IPR(K), K = 1,N)                               BLI00940
C                                                                 BLI00950
      DO 50 K = 1,N                                              BLI00960
      J = IPR(K)                                                 BLI00970
   50 IPT(J) = K                                                 BLI00980
C----------------------------------------------------------------BLI00990
      CALL LPERME(IPR,IPT,N)                                     BLI01000
```

```
C--------------------------------------------------------------------BLI01010
   60 CONTINUE                                                        BLI01020
C                                                                     BLI01030
C     COMPUTE NNZ, THE AVERAGE NUMBER OF NONZEROS PER COLUMN, AND     BLI01040
C     AVER, THE AVERAGE SIZE OF NONZERO ENTRIES IN THE FACTORS        BLI01050
C     OF THE B-MATRIX.  FROM THIS, ESTIMATE (TOO CRUDELY) THE         BLI01060
C     AVERAGE FOR B-INVERSE AS AVER = 1/AVER.                         BLI01070
      ITCOL = 0                                                       BLI01080
      AVER = 0.D0                                                     BLI01090
      DO 70 K = 1,N                                                   BLI01100
      IF(DABS(BD(K)).EQ.0.D0)  GO TO 70                               BLI01110
      ITCOL = ITCOL + 1                                               BLI01120
      AVER = AVER + DABS(BD(K))                                       BLI01130
   70 CONTINUE                                                        BLI01140
      NTCOL = ITCOL                                                   BLI01150
      DO 80 K = 1,N                                                   BLI01160
   80 ITCOL = ITCOL + 2*KCOL(K)                                       BLI01170
      NNZ = DFLOAT(ITCOL)/DFLOAT(N)                                   BLI01180
      DO 90 K = 1,NZS                                                 BLI01190
   90 AVER = AVER + DABS(BSD(K))                                      BLI01200
      AVER = AVER/DFLOAT(NZS + NTCOL)                                 BLI01210
      AVER = 1.D0/AVER                                                BLI01220
C                                                                     BLI01230
C--------------------------------------------------------------------BLI01240
C     PASS STORAGE LOCATIONS OF FACTORS TO INVERSION SUBROUTINE BLSOLV BLI01250
      CALL BSOLVE(BSD,BD,KCOL,KROW,N,NZT,NZL)                         BLI01260
C--------------------------------------------------------------------BLI01270
C                                                                     BLI01280
      GO TO 120                                                       BLI01290
C                                                                     BLI01300
  100 CONTINUE                                                        BLI01310
C     DEFAULT EXIT                                                    BLI01320
      WRITE(6,110)                                                    BLI01330
  110 FORMAT(/' TERMINATE.  PARAMETERS IN CHOLESKY FACTOR FILE'/      BLI01340
     1' DO NOT AGREE WITH THOSE SPECIFIED BY THE USER'/)              BLI01350
      STOP                                                            BLI01360
C                                                                     BLI01370
  120 CONTINUE                                                        BLI01380
C-----END OF USPEC---------------------------------------------------BLI01390
      RETURN                                                          BLI01400
      END                                                             BLI01410
C                                                                     BLI01420
C-----BLSOLV-(FACTORED INVERSES OF REAL SYMMETRIC MATRICES)----------BLI01430
C                                                                     BLI01440
C     SUBROUTINE BLSOLV(V,U)                                          BLI01450
      SUBROUTINE CBSOLV(V,U)                                          BLI01460
C                                                                     BLI01470
C--------------------------------------------------------------------BLI01480
      DOUBLE PRECISION  BD(1),BSD(1),U(1),V(1),TEMP,ZERO,ONE          BLI01490
      INTEGER  KCOL(1),KROW(1)                                        BLI01500
      COMMON/LOOPS/MAXIT,ITER                                         BLI01510
C--------------------------------------------------------------------BLI01520
      GO TO 3                                                         BLI01530
C--------------------------------------------------------------------BLI01540
      ENTRY BSOLVE(BSD,BD,KCOL,KROW,N,NZT,NZL)                        BLI01550
```

```
      GO TO 4                                              BLI01560
C----------------------------------------------------------BLI01570
    3 CONTINUE                                             BLI01580
      ITER = ITER + 1                                      BLI01590
      ZERO = 0.0D0                                         BLI01600
      ONE  = 1.0D0                                         BLI01610
C     SOLVE B*U = V FOR U WHERE  B = L*L'                  BLI01620
C     SET U = V. FIRST SOLVE L*U = U FOR U, THEN SOLVE L'*U = U FOR U   BLI01630
      KL = 0                                               BLI01640
      DO 10 K = 1,N                                        BLI01650
   10 U(K) = V(K)                                          BLI01660
      DO 30 K = 1,N                                        BLI01670
      TEMP = U(K)/BD(K)                                    BLI01680
      U(K) = TEMP                                          BLI01690
      IF (KCOL(K).EQ.0.OR.K.EQ.N) GO TO 30                 BLI01700
      KF = KL + 1                                          BLI01710
      KL = KL + KCOL(K)                                    BLI01720
      DO 20 KK = KF,KL                                     BLI01730
      KR = KROW(KK)                                        BLI01740
   20 U(KR) = U(KR) - TEMP*BSD(KK)                         BLI01750
   30 CONTINUE                                             BLI01760
      NP1 = N+1                                            BLI01770
      KF = NZT + 1                                         BLI01780
      DO 50 K = 1,N                                        BLI01790
      L = NP1 - K                                          BLI01800
      TEMP = U(L)                                          BLI01810
      IF (KCOL(L).EQ.0.OR.L.EQ.N) GO TO 50                 BLI01820
      KL = KF - 1                                          BLI01830
      KF = KF - KCOL(L)                                    BLI01840
      DO 40 LL = KF,KL                                     BLI01850
      LR = KROW(LL)                                        BLI01860
   40 TEMP = TEMP - BSD(LL)*U(LR)                          BLI01870
   50 U(L) = TEMP/BD(L)                                    BLI01880
   60 CONTINUE                                             BLI01890
C                                                          BLI01900
    4 RETURN                                               BLI01910
C                                                          BLI01920
C-----END OF BLSOLV-----------------------------------------BLI01930
      END                                                  BLI01940
C                                                          BLI01950
C-----SUBROUTINES FOR DIAGONAL TEST MATRICES---------------BLI01960
C     BLSOLV AND USPEC SUBROUTINES FOR DIAGONAL TEST MATRICES   BLI01970
C                                                          BLI01980
C-------BLSOLV DIAGONAL TEST MATRIX------------------------BLI01990
C                                                          BLI02000
C     SUBROUTINE DBSOLV(V,U)                               BLI02010
      SUBROUTINE BLSOLV(V,U)                               BLI02020
C                                                          BLI02030
C----------------------------------------------------------BLI02040
      DOUBLE PRECISION  V(1),U(1),D(1)                     BLI02050
      COMMON/LOOPS/MAXIT,ITER                              BLI02060
C----------------------------------------------------------BLI02070
      GO TO 3                                              BLI02080
C----------------------------------------------------------BLI02090
C     BELOW ENTRY IS FOR A DIAGONAL TEST MATRIX            BLI02100
```

```
      ENTRY DSOLVE(D,N)                                          BLI02110
      GO TO 4                                                    BLI02120
C------------------------------------------------------------------BLI02130
    3 CONTINUE                                                   BLI02140
      ITER = ITER + 1                                            BLI02150
   10 DO 20 I=1,N                                                BLI02160
   20 U(I)= V(I)/D(I)                                            BLI02170
C  20 U(I)= -V(I)/D(I)                                           BLI02180
C                                                                BLI02190
   30 CONTINUE                                                   BLI02200
    4 RETURN                                                     BLI02210
C-----END OF 'DIAGONAL' TEST MATRIX  BLSOLV--------------------------BLI02220
      END                                                        BLI02230
C                                                                BLI02240
C-----START OF USPEC FOR DIAGONAL TEST MATRIX-----------------------BLI02250
C                                                                BLI02260
      SUBROUTINE USPEC(N,MATNO,NNZ,AVER)                         BLI02270
C     SUBROUTINE DUSPEC(N,MATNO,NNZ,AVER)                        BLI02280
C                                                                BLI02290
C------------------------------------------------------------------BLI02300
      DOUBLE PRECISION  D(1000),DI(1000),SHIFT,SPACE,NNZ,AVER    BLI02310
      DOUBLE PRECISION  DABS, DFLOAT                             BLI02320
      REAL  EXPLAN(20)                                           BLI02330
C------------------------------------------------------------------BLI02340
C                                                                BLI02350
      READ(7,10) EXPLAN                                          BLI02360
   10 FORMAT(20A4)                                               BLI02370
      READ(7,*) NOLD,NUNIF,SPACE,D(1),SHIFT                      BLI02380
      NNUNIF = NOLD - NUNIF                                      BLI02390
      WRITE(6,20) NOLD,SPACE,NNUNIF,D(1),SHIFT                   BLI02400
   20 FORMAT(/' DIAGONAL TEST MATRIX, SIZE = ',I4/' IS THE INVERSE OF MABLI02410
     1TRIX WITH MOST ENTRIES',E10.3/' UNITS APART AND WITH ',I3,' ENTRIEBLI02420
     1S IRREGULARLY SPACED'/' FIRST ENTRY WAS ',E13.4,' SHIFT = ',E10.3 BLI02430
     1/)                                                         BLI02440
C                                                                BLI02450
      IF(N.NE.NOLD) GO TO 120                                    BLI02460
C     COMPUTE THE UNIFORM PORTION OF THE SPECTRUM                BLI02470
      DO 30 J=2,NUNIF                                            BLI02480
   30 D(J) = D(1) - DFLOAT(J-1)*SPACE                            BLI02490
      NUNIF1=NUNIF + 1                                           BLI02500
      READ(7,10)  EXPLAN                                         BLI02510
      DO 40 J=NUNIF1,N                                           BLI02520
   40 READ(7,*) D(J)                                             BLI02530
      NB = NUNIF - 2                                             BLI02540
C                                                                BLI02550
      IF(SHIFT.EQ.0.) GO TO 60                                   BLI02560
      DO 50 J=1,N                                                BLI02570
   50 D(J) = D(J) + SHIFT                                        BLI02580
C                                                                BLI02590
C     COMPUTE EIGENVALUES OF INVERSE FOR PRINTOUT ONLY           BLI02600
   60 DO 70 J = 1,N                                              BLI02610
   70 DI(J) = 1.D0/D(J)                                          BLI02620
      WRITE(6,80) (J,DI(J), J=1,N )                              BLI02630
   80 FORMAT(/' INVERSE BLOCK LANCZOS TEST, LANCZS USES INVERSE OF GIVENBLI02640
     1MATRIX'/' ENTRIES OF INVERSE OF DIAGONAL TEST MATRIX = '/(I4,E20.1BLI02650
```

```
     12,I4,E20.12,I4,E20.12))                                      BLI02660
C                                                                  BLI02670
C     DIAGONAL GENERATION COMPLETE                                 BLI02680
C                                                                  BLI02690
C     COMPUTE NNZ AND AVER                                         BLI02700
      NNZ = 1.D0                                                   BLI02710
      AVER = 0.D0                                                  BLI02720
      DO 90 K = 1,N                                                BLI02730
   90 AVER = AVER + DABS(DI(K))                                    BLI02740
      AVER = AVER/DFLOAT(N)                                        BLI02750
      AVER = 1.D0/AVER                                             BLI02760
C                                                                  BLI02770
C     COMPUTE THE GAPS                                             BLI02780
      N1 = N-1                                                     BLI02790
      DO 100 K = 1,N1                                              BLI02800
  100 DI(K) = DI(K+1) - DI(K)                                      BLI02810
      WRITE(6,110) (K,DI(K), K=1,N1)                               BLI02820
  110 FORMAT(/' GAPS BETWEEN EIGENVALUES'/(I4,E13.4,I4,E13.4,I4,E13.4,I4BLI02830
     1,E13.4))                                                    BLI02840
C                                                                  BLI02850
C-----------------------------------------------------------------BLI02860
C     PASS STORAGE LOCATIONS OF D AND N TO DSOLV SUBROUTINE        BLI02870
      CALL DSOLVE(D,N)                                             BLI02880
C-----------------------------------------------------------------BLI02890
C                                                                  BLI02900
      RETURN                                                       BLI02910
  120 WRITE(6,130) NOLD,N                                          BLI02920
  130 FORMAT(' PROGRAM TERMINATES BECAUSE NOLD = ',I5,'DOES NOT EQUAL N BLI02930
     1 =',I5)                                                      BLI02940
C-----END OF USPEC SUBROUTINE FOR 'DIAGONAL' TEST MATRICES-------------BLI02950
      STOP                                                         BLI02960
      END                                                          BLI02970
```

9.4 BLIEVAL: File Definitions, Sample Input File

Below is a listing of the input/output files which are accessed by the real symmetric block Lanczos eigenvalue/eigenvector program, BLIEVAL. This program calculates a few eigenvalues and corresponding eigenvectors of a real symmetric matrix A by computing a few extreme eigenvalues and corresponding eigenvectors of the inverse of a real symmetric matrix B obtained from A by scaling, shifting and permuting A.

Also below is a sample of an input file which BLIEVAL requires on file 5. The parameters in this file are supplied in free format. File 8 contains data for the nxn real symmetric matrix A.

```
Sample Definitions of Input/Output Files for BLIEVAL
-----------------------------------------------------------------
 BLIEVAL EXEC
FI 06 TERM
FILEDEF  5 DISK BLIEVAL    INPUT     A (RECFM F LRECL 80 BLOCK 80
FILEDEF  8 DISK &1         INPUT     A (RECFM F LRECL 80 BLOCK 80
FILEDEF 10 DISK &1         BLSTARTV  A (RECFM F LRECL 80 BLOCK 80
FILEDEF 13 DISK &1         BLEXTRAV  A (RECFM F LRECL 80 BLOCK 80
FILEDEF 15 DISK &1         BLEIGVEC  A (RECFM F LRECL 80 BLOCK 80
*IMTQL2 AND TRED2 ARE 2 EISPACK LIBRARY SUBROUTINES
LOAD  BLIEVAL  BLSUB  BLIMULT  IMTQL2  TRED2
-----------------------------------------------------------------
```

```
Sample Input File for BLIEVAL
---------------------------------------------------------------------------
 LINE 1 IWRITE (SPECIFY MESSAGE LEVEL TO FILE 6: 1 MEANS DETAILED
             1
 LINE 2 N    MATNO   SO    SHIFT    JPERM (SIZE,ID,SCALE,SHIFT,PERM?
        1250   1250   1.      0.       0
 LINE 3 MDIMQ          MDIMTM        MAXIT (DIMS. Q, TM, MAX Ax-Mults
        40000          2500          1000
 LINE 4 EFLAG  OFLAG ( EFLAG=(0,1) 1=2PHASES. OFLAG: 1=ORTHOG CHECK
            1       1
 LINE 5 SEED      (STARTING VECTOR SEED, RANDOM NUMBER GENERATOR
       3482736
 LINE 6 KMAX  KACT  KSET (MAX T SIZE +1,SIZE 1ST BLOCK,VECTORS SUPPLIED
          31     3     0
 LINE 7 KM (NUMB. EVS FOR ALG-LARGEST, -(NUMB. EVS) FOR ALG-SMALLEST
          3
 LINE 8 NSTAG  FRACT (NO. ITNS BEFORE TEST CONVERGENCE, TEST FRACTION
           25    .05
 LINE 9 RELTOL         MAXIT2 (PHASE 2,CONVERGE.TOL., Max Ax-Mults
     .00000001          1000
---------------------------------------------------------------------------
```

Chapter 10

Errata: Volume I: Theory

1. Chapter 4: Page 162: Section: Real Symmetric Generalized Eigenvalue Problems

    (a) Line 10: $Ax = \lambda x$ should be $Ax = \lambda Bx$

    (b) In Eqn(4.9.11),
    $$|\beta_{i+1}| = \|L^{-1}(Av_i - \alpha_i v_i - \beta_i v_{i-1})\|$$
    should be
    $$|\beta_{i+1}| = \|L^{-1}(Av_i - \alpha_i Bv_i - \beta_i Bv_{i-1})\|$$

REFERENCES

[1] Bjorck,Å. (1967). *Solving linear least square problems by Gram-Schmidt orthogonalization.* BIT, **7**, 1 – 21.

[2] Bunch J. R. and Kaufman, L. (1977). *Some stable methods for calculating inertia and solving symmetric linear systems.* Math. Comp., **31**, 163 – 179.

[3] Cullum, J., and Donath, W. E. (1974). *A block generalization of the symmetric s-step Lanczos algorithm.* IBM T. J. Watson Research Center, Research Report RC 4845, Yorktown Heights, New York, USA, 10598. (May 1974).

[4] Cullum, J., and Donath, W. E. (1974). *A block Lanczos algorithms for computing the q algebraically largest eigenvalues and a corresponding eigenspace for large, sparse symmetric matrices.* In Proceedings 1974 IEEE Conference on Decision and Control, 505–509.

[5] Cullum, Jane K, and Willoughby, Ralph A. (1985, 2002). *Lanczos Algorithms for Large Symmetric Eigenvalue Computations, Vol. I, Theory.* Birkhäuser, Basel 1985. (Republished as Volume **41**, SIAM Classics in Applied Mathematics, 2002.)

[6] Cullum, J. (1978). *The simultaneous computation of a few of the algebraically largest and smallest eigenvalues of a large, symmetric, sparse matrix.* BIT, **18**, 265-275.

[7] Dongarra, J. J., Bunch, J. R., Moler, C. B., and Stewart, G. W.(1979) *LINPACK User's Guide.* SIAM, Philadelphia, PA, USA.

[8] Garbow, B. S., Boyle, J. M., Dongarra, J. J., and Moler, C. B. (1977) *Matrix Eigensystem Routines - EISPACK Guide Extension.* Lecture Notes in Computer Science **51**, Second Edition, Springer, New York.

[9] George, A., Liu, J. W. H., and Ng, E.(1979). *SPARSPAK User Guide.* CS Dept. Tech. Report, U. Waterloo. Waterloo, Canada.

[10] George, A., and Liu, J. W. H. (1981). *Computer Solution of Large Sparse Positive Definite Systems.* Prentice-Hall, Englewood Cliffs, NJ.

[11] Golub, G. H., and Kahan, W. (1965). *Calculating the singular values and pseudoinverse of a matrix.* SIAM J. Numer. Anal. **2**, 205-224.

[12] Golub, G. H., and Underwood, R. (1977). *The block Lanczos method for computing eigenvalues.* In Mathematical Software III. Rice, J. R. (Ed.), Academic Press, New York, 361-377.

[13] Golub, G. H., Luk, F. T., and Overton, M. L. (1981). *A block Lanczos method for computing the singular values and corresponding singular vectors of a matrix.* Trans. Math. Software, **7**, 149-169.

[14] Lanczos, C. (1950). *An iterative method for the solution of the eigenvalue problem of linear differential and integral operators.*, J. Res. Nat. Bur. Standards, Sect. B, **45**, 255-282.

[15] Lanczos, C. (1961). *Linear Differential Operators.* Van Nostrand, New York.

[16] Moro, G., and Freed, J. H. (1981). *Calculation of ESR spectra and related Fokker-Planck forms by the use of the Lanczos algorithm.* J. Chem. Phys. **74**, 3757-3773.

[17] Paige, C. C. (1971). *The computation of eigenvalues and eigenvectors of very large sparse matrices.* Ph.D. Thesis, U. London.

[18] Paige, C. C. (1972). *Computational variants of the Lanczos method for the eigenproblem.* J.Inst. Math. Appl. **10**, 373-381.

[19] Paige, C. C. (1976). *Error analysis of the Lanczos algorithms for tridiagonalizing a symmetric matrix.* J. Inst. Math. Appl. **18**, 341-349.

[20] Paige, C. C. (1980). *Accuracy and effectiveness of the Lanczos algorithm for the symmetric eigenproblem.* Linear Algebra Appl. **34**, 235-258.

[21] Ryder, B. G. (1974). *The PFORT Verifier.* Software-Practice and Experience, **4**, 359-377.

[22] Ryder, B. G., and Hall, A. D.(1981). *PFORT Verifier.* Bell Laboratory Computer Science Technical Report 12. Bell Laboratory, Murray Hill, New Jersey, USA

[23] Smith, B. T., Boyle, J. M., Garbow, B. S., Ikebe, Y., Klema, V. C., and Moler, C. B. (1976), *Matrix Eigensystem Routines - EISPACK Guide.* Lecture Notes in Computer Science, **6**, Second Edition, Springer, New York.

[24] Stewart, G. W. (1973). *Introduction to Matrix Computations.* Academic Press, New York.

[25] Wilkinson, J. H. (1965). *The Algebraic Eigenvalue Problem.* Oxford University Press, New York.

Related Work by the Authors

[1] Cullum, J. and Zhang, T. (2002). *Two-Sided Arnoldi and Nonsymmetric Lanczos Algorithms.* SIAM J. Mat. Anal. Appl., **24**, 303 – 319.

[2] Cullum, J. and Ruehli, A. (2001), *Pseudospectra analysis, nonlinear eigenvalue problems, and studying systems with time delays,* BIT, **41**, 265–281

[3] Cullum, J.K. and Ruehli. A.E. (2000), *Method for analyzing the stability and passivity of system models,* US Patent, 6058258 (May 5, 2000).

[4] Cullum, Jane (1996). *Arnoldi versus nonsymmetric Lanczos algorithms for solving matrix eigenvalue problems,* BIT **36**, 470–493.

[5] Cullum, Jane and Willoughby, Ralph A. (1996). *A QL procedure for computing the eigenvalues of complex symmetric tridiagonal matrices,* SIAM J. Matrix Anal. Appl., **17**, 83–109.

[6] Cullum, Jane and Willoughby, Ralph (1991). *Computing eigenvalues of large matrices, some Lanczos algorithms and a shift and invert strategy.* Advances in Numerical Partial Differential Equations and Optimization, Eds., S. Gomez, J.P. Hennart and R.A. Tapia, SIAM Publications Philadelphia, PA, 198–246.

[7] Cullum, J., Kerner, W. and Willoughby, R.A. (1989). *A generalized nonsymmetric Lanczos procedure.,* Computer Physics Communications, North-Holland, **53**, 19–48.

[8] Cullum, Jane and Willoughby, Ralph A., (1986). Eds., *Large Scale Eigenvalue Problems,* Mathematics Studies, **127**, North-Holland.

[9] Cullum, Jane and Willoughby, Ralph A. (1986). *A practical procedure for computing eigenvalues of large sparse nonsymmetric matrices.* In Large Scale Eigenvalue Problems, Mathematics Studies, eds., J. Cullum and R.A. Willoughby, **127**, North-Holland, 193–240.

[10] Cullum, Jane and Willoughby, Ralph A. (1984). *A Lanczos algorithm for the modal analysis of very large, nondefective, nonsymmetric matrices,* in Proceedings of the IEEE 23rd Conference on Decision and Control, Las Vegas, Nevada, Dec. 12–14, 1984, IEEE Press, 1758–1761.

[11] Chow, J.H., Cullum, Jane, and Willoughby, Ralph A. (1983). *A sparsity-based technique for identifying slow-coherent areas in large power systems,* IEEE Transactions on Power Apparatus and Systems, **103**, 463–473.

[12] Cullum, J., Willoughby, R. A., and Lake, M. (1983). *A Lanczos algorithm for computing singular values and vectors of large matrices.* SIAM J. Sci. Statist. Comput. **4**, 197–215.

[13] Cullum, J., and Willoughby, R. A. (1981). *Computing eigenvalues of very large symmetric matrices-an implementation of a Lanczos algorithm with no reorthogonalization.* J. Comput. Phys. **44**, 329–358.

[14] Cullum, Jane and Willoughby, Ralph A. (1980). *The Lanczos phenomenon: An interpretation based upon conjugate gradient optimization,* Linear Algebra and Its Applications, **29**, 63–90.

[15] Cullum, Jane and Willoughby, R. A. (1980). *Computing eigenvectors (and eigenvalues) of large, symmetric matrices using Lanczos tridiagonalization.* Lecture Notes in Mathematics No. **773**, Numerical Analysis, Proceedings Dundee, Scotland, 1979, Ed. G. A. Watson, Springer-Verlag, 46–63.

[16] Cullum, Jane (1979). *Using Lanczos tridiagonalization to compute eigenvalues and eigenvectors of large symmetric matrices,* Conjugate Gradient Methods and Similar Techniques, ed. I. S. Duff, AERE Report R-9636, Computer Science Division, A.E.R.E. Harwell, Oxfordshire, England, December 1979, 72–90.

[17] Cullum, Jane and Willoughby, R. A. (1979). *Fast modal analysis of large sparse but unstructured symmetric matrices..* Proceedings of 17th IEEE Conference on Decision and Control, San Diego, Calif., January 1979, IEEE Press, 45–53.

[18] Cullum, J., and Willoughby, R. A. (1979). *Lanczos and the computation in specified intervals of the spectrum of large sparse real symmetric matrices.* in Proceedings of Symposium on Sparse Matrix Theory, Knoxville, Tenn., Nov. 2–3, 1978, Sparse Matrix Proceedings 1978, Eds., I. S. Duff and G. W. Stewart, SIAM, Philadelphia, 220-255.

[19] Cullum, J., Donath, W. E., and Wolfe, P. (1975). *The minimization of certain nondifferentiable sums of eigenvalues of symmetric matrices.* In Nondifferentiable Optimization, Mathematical Programming Study **3**, North-Holland, 35-55,