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

$$AX_q = X_q A_q, \quad A_q \equiv X_q^T A X_q.$$  \hspace{1cm} (8.1.1)

Typically, $A_q = \Lambda_q$, a diagonal matrix whose nonzero entries are the eigenvalues $\lambda_i$. 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

\[
Q_{j+1}B_{j+1} = AQ_j - Q_jA_j - Q_{j-1}B_j^T \equiv P_j
\]  

(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,

\[
A_j \equiv Q_j^T(AQ_j - Q_{j-1}B_j^T)
\]  

(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 LANCII to generate a sequence of \( Q \)-blocks for that iteration. Subroutine LANCZS then calls the subroutine DIAGOM to diagonalize the
8.1. **INTRODUCTION**

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 $-Az$ rather that $Az$. 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
C and modifications of them or portions of them are NOT to be
C incorporated into any commercial codes or used for any other
C commercial purposes such as consulting for other companies,
C without legal agreements with the authors of these Codes.
C BLE00100
C If these Codes or portions of them are used in other scientific or
C engineering research works the names of the authors of these codes
C and appropriate references to their written work to be
C incorporated in the derivative works.
C BLE00160
C BLE00170
C This header is not to be removed from these codes.
C BLE00180
C BLE00190
C BLE00200
C DOCUMENTATION BLOCK Lanczos EIGENVALUE/EIGENVECTOR PROGRAMS
C (1) REAL SYMMETRIC MATRICES
C BLE00210
C (2) FACTORED INVERSES OF REAL SYMMETRIC MATRICES
C BLE00220
C BLE00230
C BLE00240
C---------------------------------------------------------------BLE00250
C REFERENCE: Cullum and Willoughby, Chapter 7,
C Lanczos Algorithms for Large Symmetric Eigenvalue Computations
C VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in
C Applied Mathematics, 2002. SIAM Publications,
C Philadelphia, PA. USA
C BLE00260
C BLE00270
C BLE00280
C BLE00290
C BLE00290
C BLE00300
C BLE00310
C BLE00320
C---------------------------------------------------------------BLE00330
C REAL SYMMETRIC MATRICES:
C BLE00340
C BLE00350
C BLE00360
C GIVEN A REAL SYMMETRIC MATRIX A THE FILES BLEVAL, BLSUB AND
C BLMULT CAN BE USED TO COMPUTE A FEW EXTREME EIGENVALUES
C OF A, THAT IS THE ALGEBRARICALLY-LARGEST OR THE ALGEBRARICALLY-
C SMALLEST EIGENVALUES, AND A BASIS FOR THE CORRESPONDING
C EIGENSPACE.
C BLE00370
C BLE00380
C BLE00390
C BLE00400
C BLE00410
C BLE00420
C FACTORED INVERSES OF REAL SYMMETRIC MATRICES:
C BLE00430
C BLE00440
C GIVEN A REAL SYMMETRIC MATRIX A, THE BLOCK PROCEDURE
C CAN BE APPLIED TO AN ASSOCIATED B-MATRIX WHICH IS A
C SCALED, SHIFTED AND PERMUTED VERSION OF A. THAT IS,
C B = S0*P*A*P' + SHIFT*I WHERE THE SCALE SO AND THE SHIFT
C ARE CHOSEN BY THE USER TO PLACE THE DESIRED EIGENVALUES
C AT THE EXTREME OF THE SPECTRUM OF B- INVERSE, AND THE
C BLE00450
C BLE00460
C BLE00470
C BLE00480
C BLE00490
C BLE00500
8.2. DOCUMENTATION FOR THE CODES IN CHAPTERS 8 AND 9


ALGORITHM:


THE IDEAS USED IN THESE PROGRAMS ARE DISCUSSED IN THE FOLLOWING REFERENCES.


C----PORTABILITY-----------------------------------------------BLE01200
C EXCEPT FOR THE FOLLOWING CONSTRUCTIONS WHICH CAN BE EASILY MODIFIED BY THE USER TO MATCH THE PARTICULAR COMPUTER BEING USED, THE PROGRAM STATEMENTS ARE PORTABLE.
C NONPORTABLE STATEMENTS.
C IN BLEVAL, BLEVAL (MAIN PROGRAMS)
C 1. DATA/MACHEP STATEMENT
C 2. ALL READ(5,*) STATEMENTS (FREE FORMAT)
C 3. FORMAT(2044) USED FOR THE EXPLANATORY HEADER ARRAY, EXPLANBLE01370
C 4. FORMAT(4220) WHICH CAN BE USED TO WRITE LARGE VECTOR FILES
C 5. THE COMMON BLOCK: LOOPS.
C IN BLMULT, BLIMULT
C 1. IN BMATV, BLSOLV, AND USPEC, THE ENTRIES WHICH PASS THE STORAGE LOCATIONS OF THE ARRAYS DEFINING THE USER-SPECIFIED MATRIX OR FACTORIZATION.
C IN BLSUB
C 1. ALL STATEMENTS ARE PORTABLE EXCEPT THE ENTRY TO SUBROUTINE IPERM WHICH PASSES THE PERMUTATION USED TO OBTAIN THE B-MATRIX FROM SUBROUTINE USPEC. SUBROUTINE IPERM IS USED ONLY IN CASE (2).
C C----MATRIX SPECIFICATION-------------------------------------BLE01520
C SUBROUTINE USPEC IS USED TO SPECIFY THE MATRIX WHICH THE BLOCK LANCZOS PROCEDURE WILL USE. IN CASE (1) THIS IS THE USER-SPECIFIED A-MATRIX. IN CASE (2) THE FACTORIZATION OF THE ASSOCIATED B-MATRIX IS SPECIFIED. SUBROUTINE USPEC HAS THE CALLING SEQUENCE CALL USPEC(N,MATNO,NNZ,AVER)
8.2. DOCUMENTATION FOR THE CODES IN CHAPTERS 8 AND 9

C WHERE N IS THE ORDER OF THE USER-SUPPLIED MATRIX A, 
C MATNO IS AN <= 8 DIGIT INTEGER USED AS A MATRIX AND 
C TEST IDENTIFICATION NUMBER, NNZ IS THE AVERAGE NUMBER 
C OF NONZERO ENTRIES IN EACH COLUMN, AND AVER IS THE 
C AVERAGE SIZE OF THE NONZERO ENTRIES IN THE MATRIX USED 
C BY LANCZS. NOTE THAT NNZ AND AVER ARE DEFINED AS DOUBLE 
C PRECISION SCALARS. THE MAIN PROGRAMS ASSUME THAT THEY 
C ARE COMPUTED IN USPEC. THE USPEC SUBROUTINE 
C DEFINES AND DIMENSIONS THE ARRAYS REQUIRED TO 
C SPECIFY THE MATRIX THAT WILL BE USED BY THE LANCZS 
C SUBROUTINE AND Initializes THESE ARRAYS. THE STORAGE 
C LOCATIONS OF THESE ARRAYS ARE THEN PASSED TO THE 
C SUBROUTINE BMATV IN CASE (1) AND TO THE SUBROUTINE BSOLV 
C IN CASE (2). SAMPLE SUBROUTINES ARE INCLUDED FOR EACH 
C CASE. CASE (1) ASSUMES THAT THE A-MATRIX IS STORED ON 
C FILE 8. CASE (2) ASSUMES THAT THE FACTORIZATION OF THE 
C B-MATRIX IS STORED ON FILE 7. 
C IN CASE (1): 
C BMATV IS THE SUBROUTINE USED BY THE LANCZS SUBROUTINE 
C THAT GENERATES THE LANCZOS T-MATRICES. SUBROUTINE 
C BMATV HAS THE CALLING SEQUENCE 
C CALL BMATV(W,U) 
C WHERE U AND W ARE DOUBLE PRECISION VECTORS. FOR A GIVEN 
C W, BMATV CALCULATES U = A*W FOR THE USER-SPECIFIED MATRIX A. 
C A SAMPLE BMATV IS INCLUDED FOR AN ARBITRARY SPARSE, 
C SYMMETRIC A-MATRIX STORED IN THE SPARSE FORMAT SPECIFIED 
C IN THE CORRESPONDING SAMPLE USPEC SUBROUTINE. 
C IN CASE (2): 
C THE LANCZOS T-MATRICES ARE GENERATED USING SPARSE MATRIX 
C INVERSION, USING THE SUBROUTINE BSOLV. THE CALLING 
C SEQUENCE OF BSOLV IS 
C CALL BSOLV(U,V) 
C WHERE U AND V ARE DOUBLE PRECISION VECTORS. FOR A GIVEN V, 
C BSOLV COMPUTES U = (B-INVERSE)*V USING A SPARSE 
C FACTORIZATION OF THE B-MATRIX ASSOCIATED WITH THE USER-
C SPECIFIED A-MATRIX. 
C THE FOLLOWING SPARSE MATRIX FORMAT IS USED TO STORE THE 
C MATRICES IN THE SAMPLE PROGRAMS: 
C ICOL(K), K = 1,NZL, NUMBER OF SUBDIAGONAL NONZEROS IN COLUMN K. 
C IROW(K), K = 1,NZS, ROW INDEX OF ASD(K). 
C AD(K), K=1,N, CONTAINS THE DIAGONAL ELEMENTS OF THE A-MATRIX. 
C ASD(K), K=1,NZS CONTAINS THE SUBDIAGONAL ELEMENTS OF A BY COLUMN. 
C NZS = NUMBER OF NONZERO ELEMENTS BELOW THE DIAGONAL OF A 
C NZL = INDEX OF LAST COLUMN WITH NONZERO SUBDIAGONAL ENTRIES 
C N = ORDER OF THE A-MATRIX. 
C IN CASE (1) THE A-MATRIX IS STORED IN THIS FORMAT ON FILE 8. 

BLE01610
BLE01620
BLE01630
BLE01640
BLE01650
BLE01660
BLE01670
BLE01680
BLE01690
BLE01700
BLE01710
BLE01720
BLE01730
BLE01740
BLE01750
BLE01760
BLE01770
BLE01780
BLE01790
BLE01800
BLE01810
BLE01820
BLE01830
BLE01840
BLE01850
BLE01860
BLE01870
BLE01880
BLE01890
BLE01900
BLE01910
BLE01920
BLE01930
BLE01940
BLE01950
BLE01960
BLE01970
BLE01980
BLE01990
BLE02000
BLE02010
BLE02020
BLE02030
BLE02040
BLE02050
BLE02060
BLE02070
BLE02080
BLE02090
BLE02100
BLE02110
BLE02120
BLE02130
BLE02140
BLE02150
C IN CASE (2), IN THE SAMPLE USPEC PROVIDED WHICH IS ONLY
C FOR POSITIVE DEFINITE B-MATRICES, THE SPARSE CHOLESKY FACTOR
C OF B, L, IS STORED ON FILE 7 IN THE ABOVE SPARSE FORMAT
C USING ARRAYS BD AND BSD. IN CASE (2) THE OPTIONAL AUXILIARY
C PROGRAMS PERMUT AND LORDER ALSO REQUIRE THE A-MATRIX;
C HOWEVER, THE BLOCK LANCZOS PROCEDURE ONLY USES THE
C FACTORIZATION OF THE B-MATRIX.
C
C-----MACHEP--------------------------------------------------------------BLE02250
C
C MACHEP IS A MACHINE DEPENDENT PARAMETER SPECIFYING THE RELATIVE
C PRECISION OF THE FLOATING POINT ARITHMETIC USED.
C MACHEP = 2.2 * 10**-16 FOR DOUBLE PRECISION ARITHMETIC ON
C IBM 370-3081.
C
C THE USER WILL HAVE TO RESET THIS PARAMETER TO
C THE CORRESPONDING VALUE FOR THE MACHINE BEING USED. NOTE THAT
C IF A MACHINE WITH A MACHINE EPSILON THAT IS MUCH LARGER THAN THE
C VALUE GIVEN HERE IS BEING USED, THEN THERE COULD BE
C PROBLEMS WITH THE TOLERANCES.
C
C-----SUBROUTINES AND FUNCTIONS USER MUST SUPPLY----------------------------BLE02400
C
C GENRAN, FINPRO, MASK, USPEC, AND
C CASE (1) BMATV: CASE (2) BLSOLV :
C
C GENRAN = COMPUTES K PSEUDO-RANDOM NUMBERS AND STORES THEM IN
C THE REAL ARRAY, G. THIS SUBROUTINE IS USED TO
C GENERATE STARTING VECTORS FOR THE BLOCK LANCZOS
C PROCEDURE. CALLED FROM LANCZS SUBROUTINE.
C USER CAN SUPPLY STARTING VECTORS FOR THE BLOCK
C PROCEDURES. ANY ADDITIONAL VECTORS REQUIRED ARE
C GENERATED RANDOMLY BY GENRAN. VECTORS SUPPLIED MUST
C BE STORED ON FILE 10. THE NUMBER OF SUCH VECTORS TO
C BE READ IN IS SPECIFIED BY THE PARAMETER KSET. THE
C EXISTING CALLING SEQUENCE IS
C
C CALL GENRAN(IIX,G,K).
C
C WHERE IIX = INTEGER SEED, G = REAL ARRAY WHOSE DIMENSION
C MUST BE >= K. K PSEUDO-RANDOM NUMBERS ARE GENERATED
C AND PLACED IN G.
C
C FINPRO = DOUBLE PRECISION FUNCTION WHICH COMPUTES THE INNER
C PRODUCT OF 2 DOUBLE PRECISION VECTORS OF DIMENSION N.
C EXISTING CALLING SEQUENCE IS
C
C CALL FINPRO(N,V,J,W,K).
C
C COMPUTES THE INNER PRODUCT OF DIMENSION N OF THE VECTORS
C V AND W. SUCCESSIVE COMPONENTS OF V AND OF W ARE STORED BLE02700
AT LOCATIONS THAT ARE, RESPECTIVELY, J AND K UNITS APART.

MASK = MASKS OVERFLOW AND UNDERFLOW. OPTIONAL.
USER MUST SUPPLY OR COMMENT OUT CALL.

USPEC = DIMENSIONS AND Initializes Arrays Needed to Specify
MATRIX used by LANCZS Subroutine. See Matrix
SPECIFICATION Section.

BMATV = CASE (1) ONLY: Computes Matrix-Vector Multiply For
User-Supplied A-MATRIX. See Matrix Specification Section.

BLSOLV = CASE (2) ONLY: For Given Vector V, Computes U Such
B*U = V, Given the Sparse Factorization of the B-MATRIX.

-----PARAMETER CONTROLS-------------------------------

PARAMETER CONTROLS ARE INTRODUCED TO Control Various
Aspects of These Programs.

THE Flag EFLAG Specifies the Number of Computational Phases.

EFLAG = (0,1) MEANS

(0) Program Terminates after Completing Phase 1
Computations.

(1) Program Completes Both Phase 1 and Phase 2 of
The Computations.

THE Flag OFLAG Controls the Orthogonality Checks Between the
Jth Q-Block Generated and That Vector in the 1st Q-Block That
Is Generating Descendants. For Safety, Oflag Should be 1.

OFLAG = (0,1) MEANS

(0) No Orthogonality Checks are Made on Phase
1 Portion of the Computations. Orthogonality
Checks are Always Made on Phase 2 Portion.

(1) Program Checks Orthogonality of Generated
Q-Blocks W.R.T. That Vector in the 1st Q-Block
That is Generating Descendants in Both Phase
1 and Phase 2 of the Computations.

THE Flag IWRITE Determines the Amount of Output to File 6
During the Computations

IWRITE = (0,1) MEANS

(0) Abbreviated Output to File 6.

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

C FOR BOTH CASES:
C N, MATNO = INTEGERS. SIZE OF USER-SPECIFIED MATRIX AND MATRIX
C IDENTIFICATION NUMBER OF 8 OR FEWER DIGITS.
C
C MDIMQ, MDIMTM = INTEGERS. USER-SPECIFIED DIMENSIONS OF THE
C Q-ARRAY AND OF THE TM-ARRAY. MDIMQ >= N*KMAX
C AND MDIMTM >= MXBLK**2.
C
C MAXIT,MAXIT2 = INTEGERS. MAXIMUM NUMBER OF CALLS TO BMATV
C (CASE(1)) OR TO BLSOLV (CASE (2)) ALLOWED
C RESPECTIVELY, IN PHASE 1 AND IN PHASE 2.
C
C RELTOL = DOUBLE PRECISION SCALAR. RELATIVE TOLERANCE USED
C TO COMPUTE CONVERGENCE CRITERION FOR PHASE 2 OF
C THE BLOCK PROCEDURE.
C
C SEED = INTEGER. SEED FOR RANDOM NUMBER GENERATOR.
C USED IN GENERATION OF STARTING VECTORS FOR
C THE BLOCK PROCEDURES.
C
C KMAX = INTEGER. MXBLK = (KMAX - 1) IS MAXIMUM ALLOWED SIZE
C FOR THE SMALL LANZUOS T-MATRICES.
C
C KM = INTEGER. NUMBER OF EIGENVALUES AND EIGENVECTORS
C TO BE COMPUTED.
C
C KACT = INTEGER. INITIAL NUMBER OF VECTORS IN THE 1ST Q-BLOCK.
C IF THERE IS ANY POSSIBILITY THAT THE KM-TH DESIRED
C EIGENVALUE IS MULTIPLE, AND THE USER NEEDS TO KNOW
C THIS, THEN THE USER SHOULD SET KACT > KM. OTHERWISE,
C THIS PROGRAM WILL NOT BE ABLE TO DETERMINE THAT THAT
C EIGENVALUE IS MULTIPLE UNLESS THE (KM-1)-TH AND KM-TH
C HAPPEN TO BE MULTIPLE. IF IN FACT, THE KM-TH
C EIGENVALUE IS MULTIPLE AND THE USER NEEDS A BASIS FOR
C THE CORRESPONDING EIGENSPACE, THEN THE PROCEDURE SHOULD
C BE RERUN WITH THE EXISTING EIGENVECTORS APPROXIMATIONS
C AS STARTING VECTORS AND A LARGER KACT TO GUARANTEE THAT
C A COMPLETE BASIS FOR THAT EIGENSPACE HAS BEEN OBTAINED.
C
C KSET = INTEGER. NUMBER OF STARTING VECTORS SUPPLIED BY THE
C USER. THESE VECTORS SHOULD BE ON FILE 10.
C
C NSTAG = INTEGER. NUMBER OF THE ITERATION BEYOND WHICH THE
CHAPTER 8. REAL SYMMETRIC MATRICES, BLOCK LANCZOS CODE

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
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.
C
C NNZ = DOUBLE PRECISION SCALAR. AVERAGE NUMBER OF NONZERO BLE04450
C ENTRIES PER ROW IN THE MATRIX USED IN THE LANCZOS BLE04460
C PROCEDURE.
C
C AVER = DOUBLE PRECISION SCALAR. AVERAGE SIZE OF THE NONZERO BLE04500
C ENTRIES IN THE MATRIX USED IN THE LANCZOS PROCEDURE.
C
C CASE (2) ONLY:
C
C SO, SHIFT = DOUBLE PRECISION SCALARS. MATRIX USED BY LANCZS BLE04550
C SUBROUTINE IS B = SO*P*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.
C
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.
C
C------CONVERGENCE TEST-----------------------------------------------BLE04640
C
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) .IE. 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.
C
C------ARRAYS REQUIRED------------------------------------------------BLE04770
C
C Q(J) = DOUBLE PRECISION 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.
C
C E(J) = DOUBLE PRECISION ARRAY. ITS DIMENSION MUST BE AT BLE04900
LEAST MXBLK = KMAX - 1. ON EACH ITERATION CONTAINS
THE COMPUTED EIGENVALUES OF THE LANCZOS T-MATRIX.
BLE04910
BLE04920
BLE04930

TM(J) = DOUBLE PRECISION ARRAY. ITS DIMENSION MUST BE AT
LEAST MXBLK**2 WHERE MXBLK = KMAX - 1. CONTAINS
THE LANCZOS T-MATRIX GENERATED ON EACH ITERATION
AND THEN THE COMPUTED EIGENVECTORS OF THIS MATRIX.
BLE04940
BLE04950
BLE04960

EISPACK SUBROUTINES ARE USED FOR THE SMALL
EIGENELEMENT COMPUTATIONS. EISPACK SUBROUTINE
BLE04970
BLE04980

TRED2 IS USED TO REDUCE THE GIVEN T-MATRIX TO
TRIANGULAR FORM. THE EIGENVALUE PROBLEM FOR THE
TRIANGULAR MATRIX IS THEN SOLVED USING THE EISPACK
BLE04990
BLE05000
BLE05010
BLE05020
BLE05030
BLE05040

EXPLAN(J) = REAL ARRAY. ITS DIMENSION IS 20. THIS ARRAY IS
USED TO ALLOW EXPLANATORY COMMENTS IN THE INPUT FILES.
BLE05050
BLE05060
BLE05070

G(J) = REAL ARRAY. ITS DIMENSION MUST BE >= N. IT IS USED
FOR HOLDING THE PSEUDO-RANDOM NUMBERS USED TO GENERATE
ANY STARTING VECTORS NOT SUPPLIED BY THE USER.
BLE05080
BLE05090
BLE05100
BLE05110

RESIDL(J), = DOUBLE PRECISION ARRAYS. DIMENSION >= MAXIMUM
RESIDK(J), NUMBER OF ITERATIONS ALLOWED. MAXIMUM IS
CURRENTLY SET TO 100. USED TO MONITOR THE
RATE OF CONVERGENCE.
BLE05120
BLE05130
BLE05140
BLE05150
BLE05160

TD(J), TOD(J), = DOUBLE PRECISION ARRAYS. DIMENSION >= MXBLK.
SM(J) WORK SPACES.
BLE05170
BLE05180
BLE05190

DESC(J), XLFT(J), = INTEGER ARRAYS. DIMENSION >= MXBLK.
LEFT(J) WORK SPACES.
BLE05200
BLE05210
BLE05220

DIR(2,J) = 2-DIMENSIONAL INTEGER ARRAY. COLUMN DIMENSION >=
MXBLK, ROW DIMENSION 2. KEEPS TRACK OF NUMBER
OF VECTORS IN EACH QBLOCK.
BLE05230
BLE05240
BLE05250
BLE05260

CASE (2) ONLY:
BLE05270
BLE05280

IPR(J), IPT(J) = INTEGER ARRAYS. EACH OF DIMENSION AT LEAST N.
USED TO STORE THE REORDERING (IF ANY) OF
THE GIVEN MATRIX.
BLE05290
BLE05300
BLE05310
BLE05320

OTHER ARRAYS
BLE05330
BLE05340

THE USER IN THE SUBROUTINE USPEC MUST SPECIFY WHATEVER ARRAYS
ARE REQUIRED TO DEFINE THE MATRIX BEING USED BY LANCZS.
BLE05350
BLE05360
BLE05370
BLE05380

SUBROUTINES INCLUDED--------------------------------------------BLE05390
BLE05400
BLE05410

LANCZS = CONTAINS MAJOR LOOP FOR BLOCK LANCZOS PROCEDURES.
CALLS MAIN PROGRAM, CALLS SUBROUTINE LANC1
TO GENERATE WITHIN A GIVEN ITERATION THE Q- BLOCKS
AND CORRESPONDING LANCZOS T-MATRICES. THEN CALLS
BLE05420
BLE05430
BLE05440
BLE05450
CHAPTER 8. REAL SYMMETRIC MATRICES, BLOCK LANCZOS CODE

C SUBROUTINE DIAGS TO COMPUTE THE EIGENELEMENTS
C OF THE LANCZOS T-MATRIX AND TO MAP THE RELEVANT
C T-EIGENVECTORS INTO RITZ VECTORS FOR THE A-MATRIX.
C
LANCI1 = ON EACH ITERATION OF BLOCK LANCZOS COMPUTES
C Q-SUBBLOCKS.
C
DIAGM = CALLS EISPACK SUBROUTINES TO COMPUTE THE
C EIGENELEMENTS OF THE SMALL LANCZOS T-MATRICES
C GENERATED ON EACH ITERATION OF BLOCK LANCZOS.
C COMPUTES CORRESPONDING RITZ VECTORS FOR A-MATRIX.
C MONITORS CONVERGENCE OF BLOCK LANCZOS PROCEDURE.
C
START = GENERATES ANY REQUIRED STARTING VECTORS FOR 1ST
C Q-BLOCK FOR FIRST ITERATION OF BLOCK LANCZOS.
C
ORTHG = GIVEN A SET OF Q-VECTORS, Q(J), J = MA,MB,
C ORTHOGONALIZES THESE VECTORS W.R.T. THE Q-VECTORS
Q(J), J = 1,MA-1.
C
LPERM = (USED IN CASE (2) ONLY) GIVEN A MATRIX B AND A
C PERMUTATION P DEFINED IN THE VECTORS IPR AND IPT,
C AND A VECTOR X COMPUTE EITHER (P-TRANSPOSE)*X OR PX.
C
CASE (2) ONLY:
C FOR OPTIONAL PRELIMINARY PROCESSING:
C PERMUT (STAND-ALONE PROGRAM):
C USES THE NONZERO STRUCTURE OF A GIVEN MATRIX A.
C CAN BE USED TO OBTAIN A REORDERING OF A THAT WILL PRESERVE
C THE SPARSENESS OF A UNDER FACTORIZATION. PERMUT CALLS
C CALLS THE SPARSPAK PACKAGE, (A. GEORGE, J. LIU, E. NG,
C U. WATERLOO). SEE THE PERMUT FORTRAN CODE FOR DETAILS.
C LORDER (STAND-ALONE PROGRAM):
C GIVEN A MATRIX C IN SPARSE FORMAT AND A PERMUTATION P,
C COMPUTES THE REORDERED MATRIX B = P*C*P' AND WRITES IT
C TO FILE 9 IN SPARSE FORMAT. SEE THE LORDER FORTRAN CODE
C FOR DETAILS.
C LFACI (STAND-ALONE PROGRAM):
C GIVEN A POSITIVE DEFINITE MATRIX D IN SPARSE FORMAT,
C COMPUTES THE SPARSE CHOLESKY FACTOR L OF B AND WRITES IT
C TO FILE 7 IN SPARSE FORMAT. THUS, B = L*L'.
C SEE THE LFAC FORTRAN CODE FOR DETAILS.
C LTEST (STAND-ALONE MAIN PROGRAM):
C (USER MUST PROVIDE 3 SUBROUTINES)
C GIVEN THE FACTORIZATION OF A SPARSE MATRIX B, COMPUTES
C THE SOLUTION OF THE EQUATION B*U = B*V1 FOR A KNOWN BUT
C RANDOMLY-GENERATED VECTOR V1, SOLVING WITH AND WITHOUT ITERATIVE
C REFINEMENT TO OBTAIN A ROUGH CHECK ON THE NUMERICAL CONDITION
C OF THE B-MATRIX. THIS PROGRAM USES 3 USER-SUPPLIED SUBROUTINES
C CMATV, CMATS AND BLSOLV. SEE THE LTEST FORTRAN CODE FOR DETAILS.
C
C
C-----OTHER PROGRAMS PROVIDED------------------------------------------BLE06020
C
C LECOMPAC = TRANSLATES A REAL SYMMETRIC MATRIX PROVIDED
C IN THE FORMAT I, J, A(I,J) INTO THE SPARSE
C MATRIX FORMAT USED IN THE SAMPLE SUBROUTINES
C PROVIDED. IT ASSUMES THAT THE MATRIX
C ENTRIES ARE GIVEN EITHER COLUMN BY COLUMN OR
C ROW BY ROW. THE DATA SET CREATED IS WRITTEN TO
C FILE 8.
C
C
C
C-----------------------------------------------BLE06140
8.3 BLEVAL: Main Program, Eigenvalue and Eigenvector Computation

C-----BLEVAL (FEW EXTREME EIGENVALUES AND EIGENVECTORS)----------BLE00010
C (REAL SYMMETRIC MATRICES) BLE00020
C
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.
C BLE00170
C BLE00180
C This header is not to be removed from these codes.
C BLE00190
C BLE00200
C REFERENCE: Cullum and Willoughby, Chapter 7, BLE00201
C Lanczos Algorithms for Large Symmetric Eigenvalue Computations BLE00202
C VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in BLE00203
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 TRIDIAGONALIZATION BLE00240
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.
C 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.
C BLE00400
C BLE00410
C PFORT VERIFIER IDENTIFIED THE FOLLOWING NONPORTABLE CONSTRUCTIONS:BLE00420
C 1. DATA MACHEP DEFINITION
C BLE00430
C 2. FORMAT (20A4) USED FOR READING EXPLANATORY COMMENTS. BLEE0440
C 3. FREE FORMAT (5,*) , USED FOR PARAMETER INPUT FROM FILE 5. BLEE0450
C 4. COMMON/LOOPS/ AS CONSTRUCTED IS NOT PORTABLE BLEE0460
C
C----------------------------------------------------------------------
DOUBLE PRECISION Q(44000),E(50),TM(2500),TOD(50),TD(50),EPSM,NNZ BLEE0480
DOUBLE PRECISION SM(100),ERRMAX,SPREC,MACHEP,AYER,RELTUL,ERRMAN BLEE0480
DOUBLE PRECISION EVAL, RESIDL(100), RESIDK(100), RESID, FRACT BLEE0480
REAL EXPLAN(20),G(2000) BLEE0510
INTEGER DIR(2,100),DESC(100),LEFT(100),XLFT(100) BLEE0530
INTEGER SEED,OFLAG,EFLAG BLEE0540
COMMON/LOOPS/MAXIT,ITER BLEE0550
COMMON/RANDOM/SEED BLEE0550
COMMON/FLAGS/EFLAG,OFLAG BLEE0570
DOUBLE PRECISION DABS, DFLOAT BLEE0580
C----------------------------------------------------------------------
EXTERNAL BMATV BLEE0600
DATA MACHEP/Z3410000000000000/ BLEE0610
C----------------------------------------------------------------------
C ARRAYS MUST DIMENSIONED AS follows:
C
C 1. Q: \( KMAX*N \) BLEE0640
C 2. G: \( N \) BLEE0660
C 3. E: \( MXLK \) BLEE0680
C 4. TM: \( MXLK**2 \) BLEE0690
C 5. TOD, TD, SM, DESC, LEFT, XLFT: \( MXLK \) BLEE0700
C 6. DIR: ROW DIMENSION = 2; COLUMN DIMENSION \( MXLK \) BLEE0710
C 7. RESIDL, RESIDK: \( MAXIMUM NUMBER OF ITERATIONS ALLOWED. \) BLEE0720
C     PROGRAM CURRENTLY TERMINATES IF MORE THAN 100 ITERATIONS
C     ARE REQUESTED. USED TO MONITOR CONVERGENCE.
C 8. EXPLAN: DIMENSION = 20.
C
C----------------------------------------------------------------------
C OUTPUT HEADER
WRITE(6,10)
10 FORMAT(/' BLOCK LANCZOS PROCEDURE FOR REAL SYMMETRIC MATRICES'
[blank line]
1 /' 2ND AND SUCCEEDING BLOCKS CONTAIN ONLY ONE VECTOR'/')
C
C SET PROGRAM PARAMETERS
EPSM = 2.0D*MACHEP BLEE0830
SPREC = 1.0D-5 BLEE0840
MMPM = -1000 BLEE0860
C
C READ USER-SPECIFIED PARAMETERS FROM INPUT FILE 5 (FREE FORMAT)
C
C SELECT THE AMOUNT OF INTERMEDIATE OUTPUT DESIRED (IWRITE =0,1).
C IWRITE = 1 INCREASES THE AMOUNT OF INTERMEDIATE OUTPUT WRITTEN
C TO FILE 6 ON EACH ITERATION OF THE BLOCK LANCZOS PROCEDURE.
C
C READ(5,20) EXPLAN
C
C----------------------------------------------------------------------

20 FORMAT(20A4)
READ(5,*) IWRITE BLEE0940
C
C READ ORDER (N) OF MATRIX AND MATRIX IDENTIFICATION NUMBER (MATNO)
C
C----------------------------------------------------------------------

READ(5,20) EXPLAN BLEE0970
READ(5,20) EXPLAN BLEE0980
READ(5,*) N,MATNO

C

C READ USER-SPECIFIED DIMENSIONS OF Q-ARRAY (MDIMQ) AND OF THE
C TM-ARRAY (MDIMTM). READ MAXIMUM NUMBER (MAXIT) OF MATRIX-VECTOR
C MULTIPLIES ALLOWED IN PHASE 1.
C READ(5,20) EXPLAN
C READ(5,*) MDIMQ, MDIMTM, MAXIT

C

C READ FLAGS: EFLAG = (0,1). EFLAG = 0, MEANS PROGRAM STOPS
C AFTER COMPLETING PHASE 1 PORTION OF BLOCK LANZOS PROCEDURE.
C EFLAG = 1, MEANS PROGRAM Completes BOTH PHASES BEFORE
C TERMINATING.
C
C OFLAG = (0,1). OFLAG = 0, MEANS THAT IN PHASE 1 PORTION
C OF THE COMPUTATION, THE PROGRAM DOES NO ORTHOGONALITY CHECKS
C ON THE Q-BLOCKS GENERATED. OFLAG = 1 MEANS THAT IN THE
C PHASE 1 PORTION AND IN THE PHASE 2 PORTIONS OF THE COMPUTATIONS
C THE PROGRAM CHECKS THE ORTHOGONALITY OF THE Q-BLOCKS GENERATED
C W.R.T. THAT VECTOR IN THE FIRST BLOCK THAT IS GENERATING
C DESCENDANTS. NOTE THAT IN PHASE 2, THE PROGRAM ALWAYS MAKES
C THIS CHECK OF ORTHOGONALITY REGARDLESS OF THE VALUE OF OFLAG.
C FOR SAFETY, OFLAG SHOULD ALWAYS BE SET TO 1, ALTHOUGH IN MANY
C PROBLEMS THIS IS NOT NECESSARY.
C READ(5,20) EXPLAN
C READ(5,*) EFLAG,OFLAG

C

C READ SEED USED BY SUBROUTINE GENRAN TO OBTAIN THOSE STARTING
C VECTORS WHICH ARE GENERATED RANDOMLY.
C READ(5,20) EXPLAN
C READ(5,*) SEED

C

C SPECIFY MAXIMUM T-SIZE ALLOWED (KMAX-1); INITIAL SIZE OF
C STARTING BLOCK (KACT); NUMBER OF STARTING VECTORS SUPPLIED (KSET)
C SEE BLOCK LANZOS HEADER FOR COMMENTS ON THE SIZE OF KACT.
C READ(5,20) EXPLAN
C READ(5,*) KMAX,KACT,KSET

C

C SPECIFY NUMBER OF EXTREME EIGENVALUES AND EIGENVECTORS TO BE
C COMPUTED (KM). USER CAN SPECIFY THAT THE ALGEBRAICALLY-
C SMALLEST EIGENVALUES ARE BEING COMPUTED BY SETTING KM < 0.
C PROGRAM THEN ASSUMES THAT THE MATRIX-VECTOR MULTIPLY
C SUBROUTINE WHICH THE USER HAS PROVIDED IS COMPUTING -A*X
C INSTEAD OF A*X AND INTERNALLY IT COMPUTES THE |KM|
C ALGEBRAICALLY-LARGEST EIGENVALUES OF -A.
C READ(5,20) EXPLAN
C READ(5,*) KM
C IF(KM.EQ.0) GO TO 490
C KML = IABS(KM)

C

C STAGNATION OF CONVERGENCE OF THE KM-TH EIGENVALUE WILL BE
C TESTED AFTER NSTAG ITERATIONS. CONVERGENCE WILL BE SAID TO
C HAVE STAGNATED IF THE RATIO OF THE SQUARE OF THE CURRENT KM-TH
C RESIDUAL TO THE SQUARE OF THE CORRESPONDING RESIDUAL OBTAINED
C IN 10 ITERATIONS EARLIER IS GREATER THAN FRAC. NSTAG SHOULD BE
C >= 25. IN THE TESTS FRAC WAS SET TO .01.
C READ(5,20) EXPLAN
8.3. BLEVAL: MAIN PROGRAM, EIGENVALUE AND EIGENVECTOR COMPUTATIONS

READ(5,*) NSTAG, FRACT

C READ IN THE RELATIVE TOLERANCE (RELTOL) USED TO DETERMINE A
C CONVERGENCE CRITERION FOR PHASE 2, AND THE MAXIMUM NUMBER (MAXIT2)
C OF MATRIX-VECTOR MULTIPLIES ALLOWED IN PHASE 2.
C READ(5,20) EXPLAN
IF(EFLAG.EQ.1) READ(5,*) RELTOL, MAXIT2

C CONSISTENCY CHECKS
C PROCEDURE REQUIRES ENOUGH ROOM IN Q-ARRAY FOR AT LEAST 2
C BLOCKS OF SIZE KACT PLUS A WORKING VECTOR OF LENGTH N.
Mxbklk = Kmax - 1
Mxbklk2 = Mxbklk*KACT
IF(MDIM.LT.Mxbklk2) GO TO 470
Kmax = N*KMAX
IF(MDIM.LT.KMAX) GO TO 510
IF(KxtN.KACT) GO TO 370
IF(Mxbklk.GT.N) GO TO 390
IF(2*KACT.GT.Mxbklk) GO TO 450

C DEFINE AND INITIALIZE THE ARRAYS FOR THE USER-SPECIFIED
C A-MATRIX AND PASS THE STORAGE LOCATIONS OF THESE ARRAYS AND
C OF ANY OTHER PARAMETERS NEEDED TO DEFINE THE MATRIX TO THE
C MATRIX-VECTOR MULTIPLY SUBROUTINE BMATV.
C CALL USPEC(N, MATNO, NNZ, AVER)

C MASK OVERFLOW AND UNDERFLOW
C CALL MASK

C ARE THERE STARTING VECTORS TO READ IN FROM FILE 10 (KSET.NE.0) ?
IF(KSET.EQ.0) GO TO 70

C READ(10,30) NOLD,KACT
30 FORMAT(16,I4)
IF(NOILD.NE.0.OR.KSET.GT.KACT) GO TO 410
DO 50 J=1,KSET
READ(10,20) EXPLAN
READ(10,40) EVAL, RESID
50 FORMAT(E20.12,E13.4)
READ(10,20) EXPLAN
LINT= (J-1)*N + 1
LFIN = J*N
50 READ(10,60) (Q(JL), JL = LINT,LFIN)
60 FORMAT(4E20.12)

70 CONTINUE

C WRITE TO A SUMMARY OF THE PARAMETERS FOR THIS RUN TO FILE 6
C MXBLK = KMAX - 1
WRITE(6,80) N, NNZ, AVER, MATNO
CHAPTER 8. REAL SYMMETRIC MATRICES, BLOCK LANCZOS CODE

80 FORMAT(/'6X,'ORDER OF MATRIX ',5X,'AVERAGE NONZEROS PER ROW'/) BLE2090
115,E26.4/4X,'AVERAGE SIZE OF NONZERO ENTRIES',5X,'MATRIX ID'/ BLE2010
1E25.4,I21/) BLE2110
C WRITE(6,90) MDIMQ, MDIMTM BLE2120
90 FORMAT(/'18X,'USER-SPECIFIED'/2X,'MAX. DIMENSION Q-ARRAY',4X,'MAX.
1DIMENSION TM-ARRAY'/I16,I26/) BLE2140
C WRITE(6,100) OFLAG, EFLAG BLE2160
100 FORMAT(/'4X,'0FLAG',4X,'EFLAG'/18,I9/) BLE2170
C IF(EFLAG.EQ.1) WRITE(6,110) MAXIT,RELTOL,MAXIT2 BLE2180
110 FORMAT(/'4X,' MAXIT ',8X,' RELTOL ',6X,' MAXIT2 '/I10,E20.6,I12/) BLE2210
IF(EFLAG.EQ.0) WRITE(6,120) MAXIT BLE2220
120 FORMAT(/'4X,' MAXIT '/I10/) BLE2230
C WRITE(6,130) SEED BLE2240
130 FORMAT(/'4X,' SEED FOR RANDOM NUMBER GENERATOR'/I24/) BLE2250
C IF(KM.GT.0) WRITE(6,140) KML BLE2270
140 FORMAT(/'4X,' COMPUTE THE',3,' ALGEBRAICALLY-LARGEST EIGENVALUES AND BLE2290
1CORRESPONDING VECTORS'/) BLE2300
150 FORMAT(/'4X,' COMPUTE THE',3,' ALGEBRAICALLY-SMALLEST EIGENVALUES AND BLE2320
1CORRESPONDING VECTORS'/) PROGRAM ASSUMES THAT USER IS PROVIDING -BLE2330
1A*X INSTEAD OF A**X'/ AND COMPUTES THE ALGEBRAICALLY-LARGEST EIGENBLE2340
1VALUES OF - A ,'/ HOWEVER ON EXIT, FILE 15 CONTAINS THE ALGEBRAIBLE2350
1LY-SMALLEST EIGENVALUES OF'/ THE ORIGINAL A-MATRIX AND CORRESPONDABLE2360
1ING EIGENVECTORS./) BLE2370
IF(KM.LT.0) KM = - KM BLE2380
C BLE2390
C COMPUTE PHASE 1 CONVERGENCE TOLERANCE BLE2400
IF(AVER.GE.1.) BLE2410
1ERRMAX = 2.DO*DFLOAT(N+1000)*NNZ*AVER*MACHEP BLE2420
IF(AVER.LT.1.) BLE2430
1ERRMAX = 2.DO*DFLOAT(N+1000)*NNZ*AVER**2*MACHEP BLE2440
C BLE2450
WRITE(6,160) KACT, MBLK, KSET BLE2460
160 FORMAT(/'4X,' ON INITIAL ITERATIONS, THE FIRST BLOCK CONTAINS ',3,' BLE2470
1ECTORS'/) HOWEVER THE SIZE OF THE FIRST BLOCK MAY CHANGE AS THE ITBLE2480
1ERATIONS PROCEED'/ THE MAXIMUM SIZE T-MATRIX THAT CAN BE GENERATEBLE2490
1D IS ',4,' THE USER SUPPLIED ',3,' STARTING VECTORS'/) BLE2500
C BLE2510
WRITE(6,170) BLE2520
170 FORMAT(/'4X,' ITERATIVE PROCEDURE'/) PROCEDURE MONITORES THE SIZES OF TBLE2530
1HE NORM(GRADIENTS)**2 ON EACH '/ ITERATION. CONVERGENCE IS SAID BLE2540
1TO HAVE OCCURRED WHEN ALL / RELEVANT (NORMS)**2 ARE LESS THAN ERRBLE2550
1MAX',1,E10.3/ TYPICALLY, PHASE 1 ERRMAX YIELDS SOMETHING LESS THAN BLE2560
1' SINGLE PRECISION ACCURACY. PHASE 2 REFINES THE VECTORS OBTAINEDBLE2570
1'/ ON PHASE 1, ACCORDING TO THE ACCURACY SPECIFIED BY THE USER/') BLE2580
C BLE2590
WRITE(6,180) ERRMAX BLE2600
180 FORMAT(/'4X,' PHASE 1 CONVERGENCE CRITERION, ERRMAX ',E22.3/) BLE2610
C BLE2620
C--------------------------------------------------------------- BLE2630
C PASS STORAGE LOCATIONS OF VARIOUS ARRAYS TO LANCZS AND LANC1
C SUBROUTINES
C
CALL LANZP(DIR, DESC, SM, TM, TD, G, XLFT, LEFT, SPREC)
CALL LANC1P(DIR, DESC, SM, XLFT, LEFT)
C
C---------------------------------------------- BLE02640
C
C ENTER PHASE 1 OF BLOCK LANCZOS PROCEDURE. BLOCK PROCEDURE
C HAS 2 POSSIBLE PHASES. USER SPECIFIES PHASE 1 ONLY OR PHASE 1
C AND PHASE 2 BY SETTING EFLAG = 0 OR 1, RESPECTIVELY. PHASE 1
C COMPUTES VECTORS THAT MAY BE SOMEWHAT LESS ACCURATE THAN SINGLE
C PRECISION. PHASE 2 TAKES THE VECTORS OBTAINED IN PHASE 1
C AND ATTEMPTS TO REFINE THEM. THE USER SPECIFIES THE DEGREE
C OF REFINEMENT DESIRED BY SETTING THE VALUES OF RELTUL AND MAXIT2.
C BOTH PHASES SHOULD BE USED.
C
IPHASE = 1
NITER = 0
190 ITER = 0
RESIDL(1) = FRAC
RESIDL(2) = NSTAG

C---------------------------------------------- BLE02660
C
C CALL INITIATES THE BLOCK LANCZOS PROCEDURE.
C ON RETURN EIGENVALUE APPROXIMATIONS ARE IN E(I), I=1,KACT
C IN ALGEBRAICALLY DECREASING ORDER. EIGENVECTOR APPROXIMATIONS
C ARE IN FIRST N*KACT LOCATIONS IN THE Q-ARRAY.
C
CALL LANCZS(BMATV, KML, KSET, KACT, MXBLK, N, Q, E, RESIDL, RESIDK, ERRMAX, BLE02920
1 IPHASE, NITER, IWRITE)
C
C---------------------------------------------- BLE02940
C
IF(IPHASE.EQ.MPMIN) WRITE(15,200) N,KACT
200 FORMAT(2I10,' PHASE 2 TERMINATED '/' PROGRAM INDICATES ACCURACY ...
1ECIFIED BY USER IS NOT ACHIEVABLE'/')

C
ITERA = IABS(ITER)

IF(IWRITE.NE.MPMIN.AND.ITER.GT.0) WRITE(6,210) IPHASE, ITERA
210 FORMAT(3X,'PHASE COMPLETED',5X,'NUMBER MATRIX-VECTOR MULTIPLIES
1USED'/I10,I30)

C
IF(IWRITE.EQ.MPMIN.OR.ITER.LT.0) WRITE(6,220) IPHASE, ITERA
220 FORMAT(3X,'PHASE TERMINATED',5X,'NUMBER MATRIX-VECTOR MULTIPLIES
1USED'/I10,I30)

C
IF(ITER.GT.0.AND.IWRITE.NE.MPMIN) GO TO 250

C
IF(ITER.LT.0) WRITE(6,230)
230 FORMAT(3X,' SMALL EIGENVALUE SUBROUTINE DEFAULTED'/ 'BLOCK LANCZOS BLE03130
1 PROCEDURE STOPS AFTER SAVING CURRENT EIGENVECTOR APPROXIMATIONS'/BLE03140
1/)

C
WRITE(15,240)
WRITE(6,240)
240 FORMAT(//′ BLOCK LANCZOS PROCEDURE TERMINATES WITHOUT CONVERGENCE BLE03190
1′′ USER SHOULD EXAMINE OUTPUT TO DETERMINE REASONS FOR TERMINATION BLE03200
1′′′) BLE03210
C BLE03220
C WRITE EIGENVALUE AND EIGENVECTOR APPROXIMATIONS CONTAINED IN
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(16,14,112,′ PHASE 1, ORDER A-MATRIX, SIZE OF Q(1), SEED′) BLE03270
IF(IPHASE.EQ.2) WRITE(15,270) N,KACT,SEED BLE03280
270 FORMAT(16,14,112,′ 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(J) 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 MULTIPLICATES 
11TILIES ′,16′ EXCEEDED MAXIMUM NUMBER ′,16′ ALLOWED′/) BLE03520
C BLE03530
IF(ITER.LT.0) WRITE(15,330) BLE03540
330 FORMAT(′/′ USER BEWARE. EIGENVALUE COMPUTATIONS DEFAULTED BECAUSE BLE03550
1SE′′′ EISPACK SUBROUTINE DEFAULTED. EIGENVALUE AND EIGENVECTOR BLE03560
1 APPROXIMATIONS′′′ ABOVE WERE THOSE AVAILABLE AT THE TIME OF DEFAULT BLE03570
1AULT′′′ SOMETHING IS SERIOUSLY WRONG.′′′/) BLE03580
C BLE03590
C CHECK FOR TERMINATION AFTER PHASE 1 BLE03600
C ITER < 0 MEANS EISPACK SUBROUTINE DEFAULTED BLE03610
C I PHASE = MPPMINS MEANS THAT PHASE 2 TERMINATED DUE TO ORTHOGONALITY BLE03620
C IW RITE = MPPMINS MEANS THAT CONVERGENCE APPEARS TO HAVE STAGNATED BLE03630
C ITER > MAXIT MEANS MAXIMUM NUMBER OF MATRIX-VECTOR MULTIPLICATES BLE03640
C ALLOWED BY USER WAS EXCEEDED BLE03650
IF(ITER.LT.0.OR.ITER.GT.MAXIT) GO TO 530 BLE03660
IF(IPHASE.EQ.MPPMIN.OR.IWRITE.EQ.MPPMIN) GO TO 530 BLE03670
IF(EFLAG.NE.1.OR.IPHASE.EQ.2) GO TO 530 BLE03680
C BLE03690
C ENTER 2ND PHASE OF COMPUTATION TO ATTEMPT TO OBTAIN MORE BLE03700
C ACCURATE EIGENVECTOR APPROXIMATIONS. BLE03710
C USER CONTROLS THE SIZE OF THE ERROR TOLERANCE BY SPECIFYING BLE03720
C BLE03730
8.3. BLEVAL: MAIN PROGRAM, EIGENVALUE AND EIGENVECTOR COMPUTATIONS

C THE PARAMETER RELTOL.
C
IPHASE = 2
MAXIT = MAXIT2
KSET = KACT
C
ERROR TOLERANCE USES THE CONVERGED EIGENVALUE LARGEST IN
MAGNITUDE.
TD(1) = DABS(E(1))
IF(KML.EQ.1) GO TO 350
DO 340 J = 2,KML
340 IF(DABS(E(J)).GT.TD(1)) TD(1) = DABS(E(J))
350 TD(1) = DMAX1(TD(1),1.D0)
ERRMAN = RELTOL**2 * TD(1)**2
IF(ERRMAN.GE.ERRMAX) GO TO 430
ERRMAX = ERRMAN
C
WRITE(6,360) ERRMAX, MAXIT2
360 FORMAT(// ENTER PHASE 2 OF COMPUTATION'/ CONVERGENCE CRITERION //BLE03920
1S REDUCED TO ',E13.4/' NO MORE THAN ',I5,' MATRIX MULTIPLIED glare
1S WILL BE ALLOWED.'/ PROGRAM WILL TERMINATE IF BLOCK ORTHOGONALITY BLE03940
1 PROBLEMS MATERIALIZE'/)
C
GO TO 190
C
INCONSISTENCIES IN THE DATA
C
WRITE(6,380) KM,KACT
380 FORMAT(// PROGRAM TERMINATES BECAUSE THE NUMBER OF EIGENVALUES BLE04020
1REQUESTED, KM =' ,I3,' IS LARGER THAN THE SIZE OF THE FIRST Q BLOC BLE04030
1K, KACT =' ,I3,' SPECIFIED'/ USER MUST RESET KM OR KACT'/)
GO TO 530
C
WRITE(6,400) KMAX,N
400 FORMAT(// PROGRAM TERMINATES BECAUSE KMAX = ',I5,' IS TOO LARGE FOB BLE04080
1R THE SIZE, N = ',I5,' OF THE GIVEN MATRIX'/ USER MUST DECREASE BLE04090
1THE SIZE OF KMAX'/)
GO TO 530
C
WRITE(6,420) NOLD,N,KACT,KSET
420 FORMAT(// PROGRAM TERMINATES BECAUSE FAULT OCCURRED IN READING IN BLE04140
1THE EIGENVECTOR APPROXIMATIONS'/ EITHER THE SIZE MATRIX SPECIFIED BLE04150
10N THE EIGENVECTOR FILE',I6,' DID NOT MATCH THE SIZE SPECIFIED 'BLE04160
1,15,' IN THE PROGRAM OR THE NUMBER'/ OF VECTORS IN FILE 10 = 'BLE04170
1,14,' IS LESS THAN THE NUMBER ',I13,' USER SAID WERE THERE'/)
GO TO 530
C
WRITE(6,440) ERRMAN, ERRMAX
440 FORMAT(// COMPUTED PHASE 2 CONVERGENCE CRITERION ',E13.4/' IS LARG BLE04220
1GER THAN PHASE 1 CRITERION ',E13.4/' SO PROGRAM TERMINATES'/)
GO TO 530
C
WRITE(6,460) KACT,MXBLK
460 FORMAT(// PROGRAM TERMINATES BECAUSE THERE IS NOT ENOUGH ROOM TO BLE04270
1GENERATE 2 BLOCKS', ' BECAUSE KACT = ',I3,' AND MXBLK = ',I4/)
GO TO 530
C
C
470 WRITE(6,480) MDIMTM, MXBLK
480 FORMAT(/' PROGRAM TERMINATES BECAUSE THE DIMENSION ',I6,', OF THE TABLE04330
1-M ARRAY/'' IS TOO SMALL FOR THE LARGEST T-MATRIX ALLOWED ',I4) BLE04340
GO TO 530
C
BLE04350
C
490 WRITE(6,500)
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/
GO TO 530
BLE04400
C
BLE04410
C
510 WRITE(6,520) MDIMQ, KMAX,N
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**
C **IBM Research, T.J. Watson Research Center
C **Yorktown Heights, N.Y. 10598
C * Los Alamos National Laboratory
C * Los Alamos, New Mexico 87544
C E-mail: cullumj@lanl.gov
C
C These codes are copyrighted by the authors. These codes
C and modifications of them or portions of them are NOT to be
C incorporated into any commercial codes or used for any other
C commercial purposes such as consulting for other companies,
C without legal agreements with the authors of these Codes.
C If these Codes or portions of them are used in other scientific or
C engineering research works the names of the authors of these codes
C and appropriate references to their written work are to be
C incorporated in the derivative works.
C
C This header is not to be removed from these codes.
C
C REFERENCE: Cullum and Willoughby, Chapter 7,
C Lanczos Algorithms for Large Symmetric Eigenvalue Computations
C VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in
C Applied Mathematics, 2002. SIAM Publications,
C Philadelphia, PA. USA
C
C CONTAINS SAMPLE USPEC AND BMATV SUBROUTINES FOR USE WITH
C THE BLOCK LANCZOS PROCEDURE FOR REAL SYMMETRIC MATRICES.
C PROGRAMS ARE USED WITH BLEVAL AND BLSUB FILES.
C
C NONPORTABLE CONSTRUCTIONS:
C 1. THE ENTRY MECHANISM USED TO PASS THE STORAGE
C LOCATIONS OF THE USER-SPECIFIED MATRIX FROM THE
C SUBROUTINE USPEC TO THE MATRIX-VECTOR SUBROUTINE
C BMATV.
C
C 2. IN THE SAMPLE USPEC AND BMATV SUBROUTINES FOR DIAGONAL
C TEST MATRICES: FREE FORMAT (8,* ) AND THE FORMAT (20A4).
C
C-----USPEC (GENERAL SYMMETRIC SPARSE MATRICES)--------------------------BLM00330
C
C SUBROUTINE USPEC(N,MATNO,NNZ,AYER)
C SUBROUTINE GUSPEC(N,MATNO,NNZ,AYER)
C
C DOUBLE PRECISION ASD(10000),AD(5010),AYER,NNZ
C INTEGER IRW(10000),ICUL(5010)
C
C USPEC DIMENSIONS AND Initializes THE ARRAYS NEEDED TO DEFINE
C THE USER-SPECIFIED MATRIX AND THEN PASSES THE STORAGE LOCATIONS
C OF THESE ARRAYS TO THE MULTIPLY SUBROUTINE BMATV.

541
CHAPTER 8. REAL SYMMETRIC MATRICES, BLOCK LANCZOS CODE

C MATRIX IS STORED IN FOLLOWING SPARSE MATRIX FORMAT:
C N = ORDER OF A-MATRIX,
C NZS = NUMBER OF NONZERO SUBDIAGONAL ENTRIES,
C NZL = INDEX OF LAST COLUMN CONTAINING NONZERO SUBDIAGONAL ENTRIES,
C ICOL(J), J=1,NZL IS THE NUMBER OF NONZERO SUBDIAGONAL ELEMENTS
C IN COLUMN J.
C IROW(K), K = 1,NZS IS THE CORRESPONDING ROW INDEX FOR ASD(K).
C AD(I), I=1,N CONTAINS DIAGONAL ENTRIES (INCLUDING ANY 0
C DIAGONAL ENTRIES).
C ASD(K), K=1,NZS CONTAINS NONZERO SUBDIAGONAL ENTRIES, BY COLUMN
C FOR J > NZL THERE ARE NO NONZERO SUBDIAGONAL ELEMENTS IN COLUMN J.
C ICOL(J) = 0 IS ALLOWED
C----------------------------------------
C ARRAYS THAT DEFINE THE MATRIX ARE READ IN FROM FILE 8
C
C READ(8,10) NZS,NOLD,NZL,MATOLD
C 10 FORMAT(I10,216,I8)
C
C WRITE(6,20) NZS,NOLD,NZL,MATOLD
C 20 FORMAT(I10,216,I8,' = NZS,NOLD,NZL,MATOLD/) C
C
C TEST OF PARAMETER CORRECTNESS
C ITEMP = (NOLD-N)**2 + (MATNO-MATOLD)**2 C
C IF(ITEMP.EQ.0) GO TO 40 C
C
C WRITE(6,30) NOLD,N,MATOLD,MATNO
C 30 FORMAT('/' PROGRAM TERMINATES BECAUSE EITHER THE SIZE ',I4,' OF THE
C 1 MATRIX '/ READ FROM FILE 8 DIFFERS FROM THE SIZE ',I4,' SPECIFIED,
C 1 BY '/ THE USER OR THE MATNO ',I8,' READ IN DIFFERS FROM THE MATNO
C 1 '/ I8,' SPECIFIED BY THE USER'/)
C GO TO 100 C
C
C 40 CONTINUE C
C
C NUMBER OF NONZERO SUBDIAGONAL ENTRIES IN EACH COLUMN IS READ
C THEN THE CORRESPONDING ROW INDEX FOR EACH SUCH ENTRY IS READ
C READ(8,50) (ICOL(K), K=1,NZL)
C READ(8,50) (IROW(K), K=1,NZS)
C 50 FORMAT(13I6) C
C
C DIAGONAL IS READ FIRST, THEN NONZERO BELOW DIAGONAL ENTRIES
C READ(8,60) (AD(K), K=1,N) C
C READ(8,60) (ASD(K), K=1,NZS)
C 60 FORMAT(4E19.10) C
C
C COMPUTE NNZ, THE AVERAGE NUMBER OF NONZEROS PER COLUMN, AND
C AVER, THE AVERAGE SIZE OF NONZERO ENTRIES.
C ITCOL = 0
C AVER = 0.DO
C DO 70 K = 1,N
C IF(DABS(AD(K)).EQ.0.DO) GO TO 70
C ITCOL = ITCOL + 1
C 70 CONTINUE
C
8.4. BLMULT: SAMPLE MATRIX-VECTOR MULTIPLY SUBROUTINES

\begin{verbatim}
AVER = AVER + DABS(AD(K))
70 CONTINUE
   NTCOL = ITCOL
   DO 80 K = 1,N
50 ITCOL = ITCOL + 2*ICOL(K)
   NNZ = DFLOAT(ITCOL)/DFLOAT(N)
   DO 90 K = 1,NZS
90 AVER = AVER + DABS(ASD(K))
   AVER = AVER/DFLOAT(NZS + NTCOL)
C---------------------------------------------------------------BLM000
C   PASS STORAGE LOCATIONS OF ARRAYS THAT DEFINE THE MATRIX TO
C   THE MATRIX-VECTOR MULTIPLY SUBROUTINE BMATV
C---------------------------------------------------------------BLM011
C CALL BMATVE(ASD,AD,ICOL,IROW,N,NZL)
C---------------------------------------------------------------BLM012
C RETURN
100 STOP
C-------------END OF USPEC--------------------------------------BLM013
END
C-------------MATRIX-VECTOR MULTIPLY FOR REAL SPARSE SYMMETRIC MATRICES-------------BLM014
C---------------------------------------------------------------BLM020
C SUBROUTINE BMATV(W,U)
C SUBROUTINE GBMATV(W,U)
C---------------------------------------------------------------BLM021
DOUBLE PRECISION  U(1),W(1),ASD(1),AD(1)
INTEGER  IROW(1),ICOL(1)
COMMON/LOOPS/MAXIT,ITER
C---------------------------------------------------------------BLM022
C SPARSE MATRIX-VECTOR MULTIPLY FOR LANCZS  U = A*W
C SEE USPEC SUBROUTINE FOR DESCRIPTION OF THE ARRAYS THAT DEFINE
C THE A-MATRIX
C---------------------------------------------------------------BLM023
C GO TO 3
C---------------------------------------------------------------BLM024
C STORAGE LOCATIONS OF ARRAYS ARE PASSED TO BMATV FROM USPEC
C---------------------------------------------------------------BLM025
C ENTRY BMATVE(ASD,AD,ICOL,IROW,N,NZL)
C---------------------------------------------------------------BLM026
C GO TO 4
C---------------------------------------------------------------BLM027
3 CONTINUE
C INCREMENT THE A*W COUNTER
   ITER = ITER + 1
C COMPUTE THE DIAGONAL TERMS
   DO 10 I = 1,N
10 U(I) = AD(I)*W(I)
C---------------------------------------------------------------BLM028
C COMPUTE BY COLUMN
\end{verbatim}
CHAPTER 8. REAL SYMMETRIC MATRICES, BLOCK LANCZOS CODE

LLAST = 0
DO 30 J = 1,NZL
C
IF (ICOL(J).EQ.0) GO TO 30
LLAST = LLAST + 1
C
DO 20 L = LFIRST,LLAST
I = IROW(L)
C
U(I) = U(I) + ASD(L)*W(J)
U(J) = U(J) + ASD(L)*W(I)
C
20 CONTINUE
C
30 CONTINUE
C
4 RETURN
C-----END OF BMATV-----------------------------------------------BLM01730
END
C
C-----MATRIX-VECTOR MULTIPLY FOR DIAGONAL TEST MATRICES------------BLM01760
C
BMATV COMPUTES U = (DIAGONAL MATRIX) * W
C
SUBROUTINE BMATV(W,U)
C
SUBROUTINE DBMATV(W,U)
C
C-----DOUBLE PRECISION W(1),U(1),D(1)-----------------------------BLM01820
COMMON/LOOPS/MAXIT,ITER
C
GO TO 3
C-------END OF DIAGONAL TEST MATRIX MULTIPLY------------------------BLM02030
END
C
C-------START OF USPEC FOR DIAGONAL TEST MATRIX---------------------BLM02060
C
SUBROUTINE USPEC(N,MATNO,NNZ,AVER)
C
SUBROUTINE DUSPEC(N,MATNO,NNZ,AVER)
C
8.4. BLMULT: SAMPLE MATRIX-VECTOR MULTIPLY SUBROUTINES

               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
      NUNIF = NOLD - NUNIF                                     BLM02200
      WRITE(6,20) NOLD,SPACE,NUNIF,D(1),SHIFT                BLM02210
20 FORMAT(/' DIAGONAL TEST MATRIX, SIZE = ',I4/,' MOST ENTR
    ES ARE IRREGULARLY SPACED. FIRST ENTRY IS ',E10.3/,' SHIF
    T = ',E10.3/)                                             BLM02220
               BLM02230
C
IF(N.EQ.NOLD) GO TO 100                                       BLM02240
C
COMPUTE THE UNIFORM PORTION OF THE SPECTRUM                   BLM02250
      DO 30 J=2,NUNIF                                           BLM02260
      30 D(J) = D(1) - DFLOAT(J-1)*SPACE                      BLM02270
      NUNIF1=NUNIF + 1                                         BLM02280
      READ(8,10) EXPLAN                                         BLM02290
      DO 40 J=NUNIF1,N                                         BLM02300
      40 READ(8,*) D(J)                                        BLM02310
      IF(SHIFT.EQ.0.) GO TO 60                                  BLM02320
      DO 50 J=1,N                                              BLM02330
      50 D(J) = D(J) + SHIFT                                    BLM02340
C
PRINT OUT THE EIGENVALUES OF INTEREST                         BLM02350
      WRITE(6,70) (D(I), I=1,10)                              BLM02360
      NB = NUNIF - 2                                           BLM02370
      WRITE(6,80) (D(I), I = NB,N)                             BLM02380
60 FORMAT(/' BLOCK LANCZOS TEST, 1ST 10 ENTRIES OF DIAGONAL TEST
                           MAT')                         BLM02390
50 FORMAT(/' MIDDLE UNIFORM PORTION OF MATRIX IS NOT PRINTED OUT'/
               BLM02400
1' END OF UNIFORM PLUS NONUNIFORM SECTION = '/(3E25.16))      BLM02410
               BLM02420
C
DIAGONAL GENERATION COMPLETE                                   BLM02430
C
COMPUTE NNZ AND AVER                                          BLM02440
      NNZ = 1.DO                                               BLM02450
      AVER = 0.DO                                              BLM02460
      DO 90 K = 1,N                                           BLM02470
      90 AVER = AVER + DABS(D(K))                             BLM02480
      AVER = AVER/DFLOAT(N)                                    BLM02490
C
CALL ENTRY TO MATRIX-VECTOR MULTIPLY SUBROUTINE TO PASS        BLM02500
C
STORAGE LOCATION OF D-ARRAY AND ORDER OF A-MATRIX.             BLM02510
C
CALL MYDIAE(D,N)                                             BLM02520
C---------------------------------------------------------------BLM02530
RETURN
100 WRITE(6,110) NOLD,N                                       BLM02540
C
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 are not to be incorporated into any commercial codes or used for any other purposes without legal agreements with the authors of these Codes. BLS00120
C If these Codes or portions of them are used in other scientific or engineering research the names of the authors of these codes are to be referenced in the written work. BLS00140
C This header is not to be removed from these codes. BLS00180
C BLS00190
C BLS00196
C BLS00200
C PFORT VERIFIER IDENTIFIED THE FOLLOWING NONPORTABLE
C CONSTRUCTIONS:
C 1. ENTRY MECHANISMS USED TO PASS THE STORAGE LOCATIONS OF SEVERAL ARRAYS FROM THE MAIN PROGRAM TO THE SUBROUTINES LANCZS AND LANC11.
C 2. COMMON BLOCK: LOOPS: USED IN LANCZS AND LANC11.
C SUBROUTINES: LANCZS, LANC11, ORTHOG, START, AND DIAGON ARE USED WITH THE BLOCK LANCZS PROGRAMS BLEVAL AND BLIEVAL. LPERM IS USED WITH BLIEVAL.
C BLS00260
C BLS00270
C BLS00280
C BLS00290
C BLS00310
C BLS00320
C------LANCZS FOR BLOCK LANCZOS PROCEDURE-------------------------------BLS00330
C ON EACH ITERATION CALLS LANC11 SUBROUTINE TO GENERATE THE Q-SUBBLOCKS AND THEN CALLS DIAGON SUBROUTINE TO DIAGNOLIZE THE SMALL SYMMETRIC MATRIX WHICH IS THE PROJECTION OF THE MATRIX BEING USED BY LANCZS ONTO THE SUBSPACE SPANNED BY THESE Q-BLOCKS.
C
C SUBROUTINE LANCZS (MATVEC, KML, KSET, KACT, MXBLK, N, Q, E, RESIDL, RESIDK, ERRMAX, I PHASE, NITER, IWRITE)
C---------------------------------------------------------------BLS00440
DOUBLE PRECISION E(1),Q(1),ERRMAX,SPREC,RESN,FRACT,RKM,SUM BLS00450
DOUBLE PRECISION TM(1),SM(1),TD(*),TN(*),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
ENTRY RECEIVES STORAGE LOCATIONS OF SEVERAL OF THE ARRAYS BLS00560
USED BY THE LANCZS SUBROUTINE. THIS ALLOWS USER TO SPECIFY BLS00570
THE DIMENSIONS OF THESE ARRAYS IN THE MAIN PROGRAM. BLS00580
ENTRY LANZP(DIR,DESC,SM,TM,TD,T,G,XLFT,LEFT,SPREC) BLS00590
GO TO 4 BLS00610
C---------------------------------------------------------------BLS00620
C 3 CONTINUE BLS00630
C 3 BLS00640
KM = KML BLS00650
MMT = MXXBLK*MXXBLK BLS00660
MPMIN = -1000 BLS00670
IKACT = KACT + 10 BLS00680
FRACT = RESIDL(1) BLS00690
NSTAG = RESIDL(2) BLS00700
IOPTH0 = 0 BLS00710
C CONSTRUCT STARTING VECTORS BLS00720
IF(KSET.EQ.0) GO TO 10 BLS00730
C---------------------------------------------------------------BLS00740
CALL ORTHOG(1,KSET,N,Q) BLS00750
10 CALL START (KSET+1,KACT,N,Q,G,ERRMAX) BLS00760
C---------------------------------------------------------------BLS00770
20 CONTINUE BLS00780
C INITIALIZE THE LANCZS T-MATRIX. BLS00790
DO 30 J=1,MMT BLS00800
30 TM(J)=0.DO BLS00810
C C C
C INITIALIZE THE Q-BLOCK DIRECTORY BLS00820
DIR(1,1)=1 BLS00830
DIR(2,1)=KACT BLS00840
C C
C ORTHOGONALIZE THE STARTING VECTORS BLS00850
IF(NITER.EQ.0) GO TO 40 BLS00860
C---------------------------------------------------------------BLS00870
CALL ORTHOG(1,KACT,N,Q) BLS00880
C---------------------------------------------------------------BLS00890
40 CONTINUE BLS00900
C C
C GENERATE THE QSUBBLOCS USED ON ITERATION NITER AND STORE IN BLS00910
THE Q-ARRAY BLS00920
C C
C
DO 90 I=1,MXBLK

C

CALL LANCI1(MATVEC,MXBLK,NITER,I,N,Q,KA,T,KML,ERRMAX,RESN,RKM, C
IND,KA,CTR,N,WRITE)

C

HAS CONVERGENCE OCCURRED?

II = I+1

IF (I.EQ.1.AND.DIR(1,I).EQ.DIR(2,II)) GO TO 140

C

WAS THERE ROOM FOR ANOTHER Q-BLOCK?

IF (DIR(2,II).NE.DIR(1,II)) GO TO 100

C

IF OFLAG = 1 OR IPHASE = 2, CHECK THE ORTHOGONALITY OF

THE Q-SUBBLOCKS GENERATED WITH RESPECT TO THAT VECTOR

IN THE 1ST Q-BLOCK WHICH IS GENERATING DESCENDANTS.

IN PHASE 2 LOSSES IN ORTHOGONALITY ARE USED TO

DETERMINE WHEN THE LIMITS ON THE ACHIEVABLE ACCURACY HAVE

BEEN REACHED.

IF(OFLAG.EQ.0.AND.IPHASE.EQ.1) GO TO 90

C

L1=DIR(1,II)

LL1 = (L1-1)*N + 1

IND1 = (IND-1)*N + 1

SUM = FINPRO(N,Q(IND1),1,Q(LL1),1)

IF(DABS(SUM).GT.SPREC) GO TO 80

C

IF(IWRITE.EQ.1) WRITE(6,50) IND,L1,SUM,I

50 FORMAT('/' INNER PRODUCT OF VECTORS ',I3,' AND ',I3,' = ',E13.3/
1' THIS VIOLATES ORTHOGONALITY TEST. TERMINATE BLOCK GENERATION'
1' WITH ',I3,'TH BLOCK '/)

C

ORTHOGONALITY TEST VIOLATED, TERMINATE BLOCK GENERATION

FOR THIS ITERATION. IN PHASE 2 KEEP TRACK OF NUMBER OF

SUCH VIOLATIONS THAT LIMIT THE NUMBER OF BLOCKS TO < 10.

TERMINATE AFTER 3 SUCH VIOLATIONS IN PHASE 2.

IF(IPHASE.NE.1.AND.I.LT.IKACT) IORTH0 = IORTH0 + 1

IF(IORTH0.LT.3.AND.II.NE.2) GO TO 70

WRITE(6,60)

60 FORMAT('/' THE ORTHOGONALITY TEST HAS FAILED THREE TIMES'/
1' TERMINATE THE BLOCK PROCEDURE'/)

IPHASE = -1000

C

WRITE THE CURRENT EIGENVECTOR/EIGENVALUE

APPROXIMATIONS TO FILE 15

GO TO 160

C

TERM INATE THE Q-BLOCK GENERATION ON THIS ITERATION

70 DIR(2,II)=DIR(2,I)

GO TO 100

C
80 CONTINUE
C
C END OF ORTHOGONALITY TESTS
C
90 CONTINUE
C
C END OF RECURSIVE Q-BLOCK GENERATION
C
100 CONTINUE
MM = DIR(2,II)
IF(IWRITE.EQ.1) WRITE (6,110) MM,I
110 FORMAT(’ T-MATRIX IS OF ORDER ’,I3, ’ NUMBER OF BLOCKS = ’,I3)
C
C*****************************************************************************
C
C DIAGONALIZE THE PROJECTION MATRIX TM. ON RETURN THE
C UPDATED APPROXIMATIONS TO THE DESIRED EIGENVECTORS ARE IN THE
C FIRST KACT COLUMNS OF THE Q-ARRAY.
C
C UPDATED EIGENVALUE APPROXIMATIONS ARE IN E.
C
TD(1) = RKM
TD(2) = FRAC
IERR = NSTAG
C
CALL DIAGOM(MXBLK,MM,TM,KACT,N,Q,E,RESIDL,RESIDK,
1 RESN,IND,KACTN,MM,TD,TUD,NITER,IERR,IWRITE)
C*****************************************************************************
C
C INCREMENT COUNTER FOR NUMBER OF BLOCK LANCZOS ITERATIONS
NITER = NITER + 1
C
IWRITE = MPMIN MEANS BLOCK LANCZOS PROCEDURE TERMINATED ABNORMALLY
IF(IWRITE.EQ.MPMIN) GO TO 160
C
IERR .NE. 0 MEANS EISPACK SUBROUTINE DEFAULTED
IF(IERR.EQ.0) GO TO 130
WRITE(6,120)
120 FORMAT(’ EISPACK SIGNALS TROUBLE IN SMALL IMTQL2 EIGENVALUE SUBRUTINE’)
1UTINE,’/’ SO BLOCK LANCZOS PROGRAM TERMINATES’/)
ITER = -ITER
C
RETURN
C
130 IF (ITER.GE.MAXIT) GO TO 160
C
C UPDATED APPROXIMATIONS WERE OBTAINED WITHOUT EXCEEDING
C MAXIMUM NUMBER OF MATRIX-VECTOR MULTIPLIES SET BY THE USER.
C
GO TO 20
C
140 WRITE(6,150)
150 FORMAT(’ BLOCK LANCZOS PROCEDURE CONVERGED’)
C
C BLOCK LANCZOS PROCEDURE HAS CONVERGED.
C ATTEMPT TO IMPROVE THE APPROXIMATE EIGENVECTORS BY DIAGONALIZING
C THE SMALL PROJECTION MATRIX OBTAINED BY USING ONLY THE
C FIRST BLOCK IN Q-ARRAY.
C
160 KACT2 = KACT*MXBBLK
DO 170 KK = 1,KACT2
170 TM(KK) = 0.D0
CALL ORTHOG(1,KACT,N,Q)
KKO = 1-N
KACTP1 = (KACT)*N + 1
JJ0 = -MXBBLK-1
DO 190 K=1,KACT
JJ0 = JJ0 + MXBBLK + 1
KKO = KKO + N
CALL MATVEC(Q(KKO),Q(KACTP1))
LL0 = (K-2)*N + 1
JJ = JJ0
DO 180 L=K,KACT
LL0 = LL0 + N
JJ=J+1
TM(JJ) = FINPRO(N,Q(LL0),1,Q(KACTP1),1)
180 CONTINUE
C
190 CONTINUE
C
USE EISPACK SUBROUTINE TRED2 TO TRIDIAGONALIZE T-MATRIX
TM = (1ST Q-BLOCK)-TRANSPOSE*A*(1ST Q-BLOCK).
ON RETURN DIAGONAL ELEMENTS COMPUTED ARE IN TD, OFF-DIAGONAL
ELEMENTS ARE IN T0D, TRANSFORMATIONS USED ARE IN TM.
THEN USE EISPACK SUBROUTINE IMTQL2 TO DIAGONALIZE THE T-MATRIX.
ON RETURN. EIGENVALUES ARE IN TD IN ASCENDING ORDER.
CORRESPONDING EIGENVECTORS ARE IN TM.
CALL TRED2(MXBBLK,KACT,TM,TD,T0D,TM)
CALL IMTQL2(MXBBLK,KACT,TD,T0D,TM,IERR)
IF(IERR.EQ.0) GO TO 200
WRITE(6,120)
ITER = -ITER
RETURN
C
COMPUTE SUCCESSIVELY THE JTH-COMPONENTS OF THE RITZ VECTORS.
REORDER THE EIGENVALUES (AND EIGENVECTORS) SO THAT THEY
ARE IN ALGEBRAICALLY DECREASING ORDER.
200 DO 220 J=1,N
JJO = - MXBBLK
JLO = -N + J
DO 210 K=1,KACT
T0D(K)=0.D0
CHAPTER 8. REAL SYMMETRIC MATRICES, BLOCK LANCZOS CODE

\[ \text{JJ0} = \text{JJ0} + \text{MXBLK} \]
\[ \text{JJ} = \text{JJ0} \]
\[ \text{JL} = \text{JL0} \]
\[ \text{DO } 210 \text{ L}=1,\text{KACT} \]
\[ \text{JJ} = \text{JJ} + 1 \]
\[ \text{JL} = \text{JL} + N \]
\[ 210 \text{TOD}(K) = \text{TOD}(K) + \text{TM(JJ)} \cdot q(JL) \]
\[ \text{JK} = \text{JL0} \]
\[ \text{DO } 220 \text{ K}=1,\text{KACT} \]
\[ \text{JK} = \text{JK} + N \]
\[ \text{KACTK} = \text{KACT} - K + 1 \]
\[ q(JK) = \text{TOD}(\text{KACTK}) \]
\[ 220 \text{ CONTINUE} \]
\[ \text{DO } 230 \text{ K}=1,\text{KACT} \]
\[ \text{KACTK} = \text{KACT} - K + 1 \]
\[ 230 \text{ E(K)} = \text{TOD}(\text{KACTK}) \]
\[ C \]
\[ \text{C HAS CONVERGENCE OCCURRED?} \]
\[ \text{IF(1.EQ.1.AND.DIR(2,I).EQ.DIR(2,I+1)) GO TO 250} \]
\[ C \]
\[ \text{C CONVERGENCE HAS NOT OCCURRED, PROCEDURE TERMINATED FOR SOME} \]
\[ \text{C OTHER REASON} \]
\[ \text{WRITE(6,240)} \]
\[ 240 \text{ FORMAT}//' BLOCK LANCZOS PROCEDURE TERMINATES WITHOUT CONVERGENCE'.BlS02870 \]
\[ 1/' AFTER WRITING THE CURRENT EIGENVALUE AND EIGENVECTOR APPROXIMAT.BLS02880 \]
\[ 1IONS'/' TO FILE 15'/) \]
\[ C \]
\[ \text{RETURN} \]
\[ C \]
\[ 250 \text{ IF(IPHASE.EQ.1) WRITE(6,260) (E(K), K=1,KACT) \]
\[ \text{IF(IPHASE.EQ.2) WRITE(6,270) (E(K), K=1,KACT) \]
\[ 260 \text{ FORMAT}// AT END OF PHASE 1, COMPUTED EIGENVALUES ='/4E20.12) BLS02960 \]
\[ 270 \text{ FORMAT}// AT END OF PHASE 2, COMPUTED EIGENVALUES ='/4E20.12) BLS02960 \]
\[ C \]
\[ C \]
\[ C \]
\[ C \]
\[ \text{C----END OF LANCZS-----------------------------------------------BLS02990} \]
\[ 4 \text{ RETURN} \]
\[ \text{END} \]
\[ C \]
\[ \text{C----START OF LANCI1---------------------------------------------BLS03030} \]
\[ \text{C GENERATES THE Q-SUBBLOCKS ON EACH ITERATION OF THE BLOCK LANCZOS} \]
\[ \text{C PROCEDURE.} \]
\[ \text{C} \]
\[ \text{SUBROUTINE LANCI1(MATVEC,MXBLK,NITER,I,N,Q,KACT,KML,ERRMAX,} \]
\[ \text{1RESN,RXM,IND,KACTN,IWRITE) \]
\[ \text{C} \]
\[ \text{C-----------------------------------------------BLS03100} \]
\[ \text{DOUBLE PRECISION Q(1),TM(1),S,SM(1),T,ERRMAX,SUM,RESN,RXM} \]
\[ \text{INTEGER DIR(2,*) DESC(1),LEFT(1),XLFT(*) \]
\[ \text{DOUBLE PRECISION FINPRO, DSQRT \]
\[ \text{EXTERNAL MATVEC} \]
\[ \text{C-----------------------------------------------BLS03150} \]
\[ 0 \text{ TO 3} \]
\[ \text{C-----------------------------------------------BLS03170} \]
\[ \text{C ALLOWS PASSAGE OF LOCATIONS OF SOME OF THE ARRAYS USED BY LANCI1} \]
\[ BLS03180 \]
C SO THAT THESE ARRAYS CAN BE DIMENSIONED IN THE MAIN PROGRAM
ENTRY LANCP1(DIR,DESC,TM,SM,XLFT,LEFT)
GO TO 4
C-----------------------------------------------
3 CONTINUE
C SIZE OF FIRST BLOCK CAN CHANGE.
IF(I.EQ.1) KACTN = KACT
C XLFT(I+2) IS CUMULATIVE TOTAL OF VECTORS IN 1ST QBLOCK NOT
C GENERATING DESCENDANTS.
IF(I.GT.1) GO TO 10
XLFT(1) = 0
XLFT(2) = 0
10 XLFT(I+2) = XLFT(I+1)
C INITIALIZE THE DIRECTORY FOR NEXT QBLOCK Q(I+1)
I2=DIR(2,1)
I1=DIR(1,1)
DIR(1,I+1)=I2+1
DIR(2,I+1)=I2
C IS THERE ROOM FOR ANOTHER QBLOCK?
MS = I2-I1+1
IF (MS+I2.LE.MXBLK) GO TO 70
C NOT ENOUGH ROOM TO GENERATE ANOTHER BLOCK
C COMPLETE THE TM-MATRIX. NOTE THAT THE TM-MATRIX IS
C DIMENSIONED AS (MXBLK,1) AND THE EISPACK SUBROUTINES
C REQUIRE THE LOWER TRIANGULAR PART OF THIS MATRIX.
I3=I2+1
JI30 = (I3-1)*N
JI31 = JI30 + 1
JK1 = (I1-2)*N + 1
DO 60 K=I1,12
JK1 = JK1 + N
C-----------------------------------------------
CALL MATVEC(Q(JK1),Q(JI31))
C-----------------------------------------------
C COMPUTE LAST DIAGONAL BLOCK IN TM-MATRIX FOR THIS ITERATION
JL1 = (K-2)*N + 1
KK = (K-1)*MXBLK + K - 1
20 DO 30 L=K,I2
KK = KK + 1
JL1 = JL1 + N
C-----------------------------------------------
TM(KK) = FINPRO(N,Q(JL1),1,Q(JI31),1)
C-----------------------------------------------
30 CONTINUE
C COMPUTE ASSOCIATED CORRECTION TERMS IN TM-MATRIX.
IF(XLFT(I),EQ,0) GO TO 50
LUP = XLFT(I)
DO 40 JJ = 1,LUP
L= LEFT(JJ)
JL1 = (L-1)*N + 1
C---------------------------------------------------------------
SUM = FINPRO(Q,JI1),1,Q(JL1),1)
---------------------------------------------------------------
C
KK = (L-1)*MXBLK + K
TM(KK) = SUM + TM(KK)
40 CONTINUE
C
50 CONTINUE
C
60 CONTINUE
C RETURN
C
C ON EVERY BLOCK PASS THROUGH HERE TO GENERATE THE ITH-BLOCK
C DIAGONAL ENTRY A(I) OF THE TM-MATRIX, EXCEPT THE LAST DIAGONAL
C BLOCK WHICH IS GENERATED ABOVE
C
70 CONTINUE
C COMPUTE (A-MATRIX)*(ITH-Q-BLOCK)
KA=I2
DO 80 K=I1,I2
KA=KA+1
JKA1 = (KA-1)*N + 1
JK1 = (K-1)*N + 1
C---------------------------------------------------------------
CALL MATVEC(Q(JK1),Q(JKA1))
---------------------------------------------------------------
C
DESC(K)=KA
80 DESC(KA)=K
C
C COMPUTE (A-MATRIX)*(ITH-Q-BLOCK) - ((I-1)TH-Q-BLOCK)*B(I)-TRANS
C WHERE B(I) DENOTES THE ITH SUBDIAGONAL BLOCK
C
IF(I.EQ.1) GO TO 110
J1 = DIR(1,1-1)
J2 = DIR(2,1-1)
DO 100 K=I1,I2
KD=DESC(K)
JKDO = (KD-1)*N
KK = (J1-2)*MXBLK + K
DO 90 L=J1,J2
JL = (L-1)*N
KK = KK + MXBLK
S=TM(KK)
JKD = JKDO
DO 90 J=1,N
JKD = JKD + 1
JL = JL + 1
90 CONTINUE
C
90 Q(JKD) = Q(JKD) - S*Q(JL)
100 CONTINUE
   LINT = (KD-1)*N + 1
   LFIN = K*N
C
C   COMPUTE A(I)
C
110 DO 130 K=I1,I2
   KKMX = (K-1)*MXBLK
   KD=DESC(K)
   JKD1 = (KD-1)*N + 1
   JL1 = (K-2)*N + 1
   DO 120 L=K,I2
   JL = JL1 + N
   KK = KKMX + L
   IF(L.LT.K) KK=(L-1)*MXBLK + K
   S=TM(KK)
   JKD = JKD1
   DO 140 J=1,N
   JL = JL1 + 1
   JKD = JKD + 1
140 Q(JKD) = Q(JKD) - S*Q(JL)
C
C   REORTHOGONALIZE THE BLOCK P(I) WITH RESPECT TO ALL VECTORS
C   IN THE 1ST QBLOCK THAT ARE NOT CURRENTLY GENERATING ANY
C   DESCENDANTS.  NOTE THAT 2ND Q-BLOCK IS REORTHOGONALIZED
C   ELSEWHERE.
C   IF(XLFT(I).EQ.0) GO TO 170
   LUP = XLFT(I)
   DO 160 JJ = 1,LUP
   L= LEFT(JJ)
   JL0 = (L-1)*N
   LLMX = (L-1)*MXBLK
   JL1 = JL0 + 1
   JKD1 = JKD0 + 1
C---------------------------------------------------------------------
   SUM = FINPRO(N,Q(JL1),1,Q(JKD1),1)
C---------------------------------------------------------------------
   JKD = JKD0
   JL = JL0
CHAPTER 8. REAL SYMMETRIC MATRICES, BLOCK LANCZOS CODE

DO 150 J=1,N
J KD = J KD + 1
JL = J L + 1
150 Q(J KD) = Q(J KD) - SUM* Q(JL)
K K = L L M X + K
T M(K K) = SUM + T M(K K)
C
160 CONTINUE
170 CONTINUE
C
C GENERATE B(I+1)
C
K 1 = DESC(I 1)
K 2 = DESC(I 2)
IFLAG=0
C
C COMPUTE NORMS
C
180 CONTINUE
J K 1 = (K 1 - 2)*N + 1
D O 190 K = K 1 , K 2
J K 1 = J K 1 + N
C----------------------------------------------BL05070
S M(K) = FINPRO(N, Q(J K 1), 1, Q(J K 1), 1)
C----------------------------------------------BL05090
190 CONTINUE
C
IF(I.EQ.1.AND.K1.EQ.I2+1) WRITE(6,200) NITER,
1 (K, S M(K), K =K 1,K 2)
200 FORMAT(// ON ITERATION', I4,' NORM(GRADIENTS)**2 OF 1ST BLOCK = 'BL05140
1/5(I4,E12.3))
C
C TEST FOR CONVERGENCE OF BLOCK LANCZOS
C
C IF(I.GT.1.OR.K1.GT.I2+1) GO TO 250
C
C TEST THE FIRST KM OF THE EIGENVALUES FOR CONVERGENCE
K 2 L = K 1 + K M L - 1
R K M = S M(K 2 L)
D O 210 K = K 1,K 2 L
IF(S M(K).GT.ERRMAX ) GO TO 220
210 CONTINUE
G 0 TO 430
C
C CAN WE REDUCE K ACT? IF A SMALL RESIDUAL (GRADIENT) IS IDENTIFIED,BL05290
C SIZE OF 1ST BLOCK MAY BE REDUCED.
C
220 IF(K M L.EQ.K ACT) GO TO 250
D O 230 K = K 2 L,K 2
IF(S M(K).GT.ERRMAX) GO TO 230
K S A V = K
K ACT N = K S A V - K ACT
G 0 TO 240
C
230 CONTINUE
DO 320 J=1,N
  JK = JK + 1
  JK1 = JK1 + 1
  T=Q(JK1)
  Q(JK1)=Q(JK)
  320 Q(JK)=T
  MA=DESC(K)
  MB=DESC(K1)
  DESC(K1)=MA
  DESC(K)=MB
  DESC(MA)=K1
  DESC(MB)=K
  330 CONTINUE
C
  DIR(2, I+1)=K1
C
  IFLAG=1
C
  K1=K1+1
  IF(I.EQ.1) GO TO 340
  IF (K1.LE.K2) GO TO 180
  C RETURN TO LANCZS
C
  RETURN
C
  IMPLICIT VECTOR DEFLATION
C
  340 CONTINUE
  J= XLFT(I+2)
  IF(K1.GT.K2) GO TO 360
  DO 350 L= K1,K2
     J = J+1
  350 LEFT(J) = DESC(L)
  360 XLFT(I+2) = J
  C
  FORCE REORTHOGONALIZATION OF 2ND AND 3RD QBLOCKS W.R.T. THOSE
  C VECTORS IN 1ST QBLOCK THAT ARE NOT GENERATING DESCIANDANTS
C
  ON THIS ITERATION.
  IF(I.GT.1) GO TO 370
  XLFT(1) = XLFT(3)
  XLFT(2) = XLFT(3)
  370 IJJ = I + 2
  IJJJ= XLFT(IJJ)
  C
  IF(IJJJ.EQ.0) GO TO 390
  IF(IWRITE.EQ.1) WRITE(6,380) (LEFT(IJ),IJ= 1, IJJJ)
  380 FORMAT(VECTORS NOT GENERATING DESCENDANTS ARE '/(10I6))
C
  390 IF(I.EQ.1.AND.KML.GT.1) GO TO 400
C
  RETURN
C
  REORTHOGONALIZE 2ND QBLOCK W.R.T VECTORS IN 1ST BLOCK NOT
  C GENERATING DESCENDANTS
  400 IF(XLFT(I).EQ.0) RETURN
**** SUB: OTHER SUBROUTINES USED BY THE CODES IN CHAPTERS 8 AND 9 ****

```
LUP = XLFT(I)                          BLS06490
KD = DIR(2,I+1)                        BLS06500
JKDO = (KD-1)*N                        BLS06510
DO 420 JJ = 1,LUP                      BLS06520
L = LEFT(JJ)                           BLS06530
JLO = (L-1)*N                          BLS06540
JL1 = JLO + 1                          BLS06550
JKD1 = JKDO + 1                        BLS06560
C----------------------------------------BLS06570
SUM = FINPRO(N,Q(JKD1),1,Q(JL1),1)     BLS06580
C----------------------------------------BLS06590

JL = JLO
JKD = JKDO
DO 410 J=1,N
JL = JL + 1
JKD = JKD + 1
410 Q(JKD) = Q(JKD) - SUM *Q(JL)
420 CONTINUE
C
RETURN
C
EXIT IF CONVERGENCE OF DESIRED EIGENVECTORS IS CONFIRMED.
C
430 CONTINUE
DO 440 L=K1,K2
M=DESC(L)
440 DESC(M)=0
DIR(2,2)=DIR(2,1)
C
WRITE(6,450) ERRMAX
450 FORMAT(/' CONVERGENCE OBSERVED, ALL RESIDUALS**2 .LT. ERRMAX = ', I3)
        ERRMAX = 1 E20.12)
C
RETURN
C-----END OF LANCI---------------------------------------------BLS06840
END
C
C-----ORTHOG----------------------------------------------------BLS06870
C ORTHOGONALIZE COLUMNS M = MA,MB OF Q-ARRAY W.R.T COLUMNS M = 1,MB BLS06880
C
SUBROUTINE ORTHOG(MA,MB,N,Q)
C
DOUBLE PRECISION Q(1), S
DOUBLE PRECISION FINPRO, DSQRT
C---------------------------------------------BLS06920
C MAIN LOOP
DO 50 M = MA,MB
MMO = (M-1)*N
LLO = -N
DO 40 L = 1,M
LLO = LLO + N
LL = LLO + 1
MM = MMO + 1
```


C--------------------------------------------------------------BLS07040
  S = FINPRO(N,Q(LL),1,Q(MM),1)
C--------------------------------------------------------------BLS07060
C
  IF (M.EQ.L) GO TO 20
C
  MM = MMO
  LL = LLO
  DO 10 I=1,N
  LL = LL + 1
  MM = MM + 1
  10 Q(MM) = Q(MM) - S*Q(LL)
  GO TO 40
C
20 S = DSQRT(S)
  MM = MMO
  DO 30 I=1,N
  MM = MM + 1
  30 Q(MM) = Q(MM)/S
C
40 CONTINUE
50 CONTINUE
C
RETURN
C--------------------------------------------------------------BLS07280
END
C--------------------------------------------------------------BLS07310
C  GENERATES PSEUDO-RANDOM STARTING VECTORS.
C
C  SUBROUTINE START(KA,KB,N,Q,G,ERRMAX)
C
C--------------------------------------------------------------BLS07360
DOUBLE PRECISION  Q(1), ERRMAX, S
REAL  G(1)
COMMON/RANDOM/IIX
DOUBLE PRECISION FINPRO, DSQRT
C--------------------------------------------------------------BLS07410
IF(KA.GT.KB) RETURN
C
IIL = IIX
DO 110 K = KA,KB
  KKO = (K-1)*N
C
CALL GENRAN(IIL,G,N)
C--------------------------------------------------------------BLS07500
C
  KK = KKO
  DO 10 I = 1,N
  KK = KK + 1
  10 Q(KK) = G(I)
  LLO = -N
20 DO 70 L=1,K
  LLO = LLO + N
BLS07580
8.5. BLSUB: OTHER SUBROUTINES USED BY THE CODES IN CHAPTERS 8 AND 9  561

LL = LL0 + 1
KK = KK0 + 1
S = FINPRO(N,Q(LL),1,Q(KK),1)

C----------------------------- BLS07610
C
IF (K.EQ.L) GO TO 40
C
LL = LL0
KK = KK0
d0 30 I=1,N
LL = LL + 1
KK = KK + 1
30 Q(KK) = Q(KK) - S*Q(LL)
GO TO 70
C
40 S = DSQRT(S)
IF(S.LE.ERRMAX) GO TO 80
KK = KK0
d0 50 I=1,N
KK = KK + 1
50 Q(KK) = Q(KK)/S
C
WRITE(6,60) K
60 FORMAT(I6,' TH STARTING VECTOR IS GENERATED RANDOMLY')
C
70 CONTINUE
GO TO 110
C
80 CALL GENRAN(IIX,G,N)
C----------------------------- BLS07900
C
WRITE(6,90) K
90 FORMAT(/I6,' TH RANDOM VECTOR REJECTED, GENERATE ANOTHER/')
C
KK = KK0
d0 100 I = 1,N
KK = KK + 1
100 Q(KK) = G(I)
GO TO 20
C
110 CONTINUE
RETURN
C-------END OF START----------------------------------- BLS08030
END
C
C-------START OF DIAGOM--------------------------------- BLS08060
C
DIAGOM CALLS THE EISPACK SUBROUTINES TRED2 AND IMTQL2 TO
C DIAGONALIZE THE SMALL SYMMETRIC MATRICES GENERATED AT EACH
C ITERATION OF BLOCK Lanczos.
C
SUBROUTINE DIAGOM(MXBLK,MM,TM,KACT,N,Q,E,RESID,RESK,RESN,IND,
1 KACTN,KM,TD,TD2,NITER,IER,WRITE)
C
C------------------- BLS08100
C
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.1.0.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.
INDEXP = IND
ICOUNT = 0
KACT = KACT + 1
KWANT = KACT
C CHECK THAT THERE IS ENOUGH ROOM TO ENLARGE THE 1ST QBLOCK
IF(2*KACT.GT.MXBLK) GO TO 230
GO TO 80
C 60 RESID(ICOUNT) = RESN
INDEXP = IND
GO TO 80
C 70 ICOUNT = 1
RESID(1) = RESN
INDEXP = IND
C USE EISPACK SUBROUTINES TO DIAGONALIZE THE SMALL TM-MATRIX.
C 80 CALL TRED2(MXBLK,MM,TD,TUD,TM)
CALL IMTQL2(MXBLK,MM,TD,TUD,TM,IERR)
C-----------------------------------------------------------------------
IF(IERR.EQ.0) GO TO 90
RETURN
C SELECT RELEVANT EIGENVALUES AND EIGENVECTORS OF THE T-MATRIX.
C 90 CONTINUE
C IMTQL2 RETURNS EIGENVALUES (AND CORRESPONDING EIGENVECTORS) IN
C ALGEBRAICALLY-ASCENDING ORDER. REARRANGE TO DESCENDING ORDER.
C
DO 100 L=1,MM
MML = MM-L+1
100 E(L) = TD(MML)
C 110 WRITE(6,120) KACT, (E(J), J=1,KACT)
120 FORMAT(’COMPUTED’,I4,’ALGEBRAICALLY-LARGEST EIGENVALUES’/(4E20.1))
C COMPUTE ESTIMATE MAXIMUM EIGENVALUE AND OF SPREAD
C IF(NITER.GT.1) GO TO 140
EMAX = DMAX1(DABS(E(1)),DABS(E(MM)))
SPREAD = DABS(E(1) - E(MM))
EGAP = SPREAD/DFLOAT(N)
C 130 FORMAT(/4X,’ESTIMATED NORM OF MATRIX’,4X,’ESTIMATED SPREAD’,6X,’EBL&iosc9150
1READ*(SIZE)’(1)/E28.4,E20.4,E24.3)
C 140 CONTINUE
C COMPUTE RITZ VECTORS
DO 180 I=1,N
DO 150 KK=1,KWANT
TUD(KK)=0.DO
K = MM - KK + 1
C
CHAPTER 8. REAL SYMMETRIC MATRICES, BLOCK LANCZOS CODE

IL = -N + I
DO 150 L = 1,MM
IL = IL + N
150 TOD(KK) = TOD(KK) + TM(L,K)*Q(IL)
IKK = -N + I
160 DO 170 KK=1,KACT
IKK = IKK + N
170 Q(IKK)=TOD(KK)
180 CONTINUE
C BLS09260
C ON FILE 13 SAVE ANY EXTRA VECTORS NO LONGER NEEDED IN 1ST Q-BLOCK BLS09340
IF(KWANT.EQ.KACT) GO TO 290
K1 = KACT + 1
K2 = KWANT
DUMMY = 100.
DO 190 K = K1,K2
LINT = (K-1)*N + 1
LFIN = K*N
WRITE(13,210) E(K),DUMMY,K
WRITE(13,220) (Q(L), L=LINT,LFIN)
190 CONTINUE
KDELTA = KWANT - KACT
WRITE(13,200) KDELTA
200 FORMAT(/' ABOVE ARE ',I3,' VECTORS STRIPPED FROM A 1ST Q-BLOCK'/) BLS09470
1' DURING A BLOCK LANCZOS RUN WHICH COULD BE USED AS STARTING VECTORS BLS09480
1RS'/' IN A LATER RUN IF THE USER DECIDES THAT THESE EIGENVALUES SHOULD BLS09490
1GULD'/' BE COMPUTED AFTER ALL. FORMAT USED IN THE SAME AS WAS USED BLS09500
1D'//' IN THE CORRESPONDING BLSTART FILE'/') BLS09510
210 FORMAT(/E20.12,E13.4,I6,=' EVAL,DUMMY,EVAL NUMBER,EVEC='/) BLS09520
220 FORMAT(4E20.12) BLS09530
GO TO 290
C BLS09540
C DEFAULT, SIZE OF 1ST Q-BLOCK TOO LARGE FOR MXBLK BLS09560
230 IWRITE = -1000
WRITE(6,240) KACT,MXBLK
WRITE(15,240) KACT,MXBLK
240 FORMAT(/' BLOCK LANCZOS PROCEDURE TRIED TO INCREASE THE SIZE OF 1BL BLS09600
1ST QBLOCK'/' TO ',I3,' BUT THIS IS NOT FEASIBLE BECAUSE TWICE THIS BLS09610
1 SIZE'/' IS G.T. MXBLK WHICH EQUALS ',I4,' USER CAN RESTART PROGRAM BLS09620
1WITH LARGER MXBLK'/') BLS09630
GO TO 290
C BLS09640
C DEFAULT, CONVERGENCE RATE IS TOO SLOW BLS09660
250 IWRITE = -1000
WRITE(6,260) NITER,RATIO,FRACT
WRITE(15,260) NITER,RATIO,FRACT
260 FORMAT(/' ON ITERATION ',I3,' CONVERGENCE APPEARS TO BE STAGNATED BLS09700
1'//' RATIO OF SQUARE OF CURRENT KM-TH RESIDUAL TO CORRESPONDING SQU BLS09710
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, USE BLS09740
1R'/' CAN EITHER INCREASE KACT OR KMAX OR RESET THE STAGNATION PARA BLS09750
1METERS'//' NSTAG AND FRACT, AND RESTART THE BLOCK PROCEDURE USING TBLS BLS09760
1HE'//' CURRENT EIGENVECTOR APPROXIMATIONS AS STARTING VECTORS'/') BLS09770
GO TO 290
C BLS09780
8.5. BLSUB: OTHER SUBROUTINES USED BY THE CODES IN CHAPTERS 8 AND 9

C
270 WRITE = -1000
WRITE(6,280)
WRITE(15,280)

280 FORMAT(//' SOMETHING IS SERIOUSLY WRONG. NUMBER OF ITERATIONS IS
1EXCESSIVE!'// PROGRAM TERMINATES FOR USER TO DECIDE WHAT TO DO'/)
1ALTERNATIVES INCLUDE INCREASING KACT OR KMAX OR BOTH, AND RESTARTING
1ING'// USING THE CURRENT APPROXIMATIONS AS STARTING VECTORS'/)

C
290 CONTINUE
RETURN
C-----END OF DIAGOM-----------------------------------------------
END
C-----LPERM PERMUTES VECTORS--------------------------------------
C
SUBROUTINE LPERM(W,U,IPERM)

DOUBLE PRECISION U(1),W(1)
INTEGER IPR(1),IPT(1)

C SUBROUTINE HAS 2 BRANCHES: IPERM = 1, CALCULATES
C U = P**W WHERE P IS THE PERMUTATION REPRESENTED BY IPR
C LET J = IPR(K) THEN U(K) = W(J), K = 1,N. WE SET W(K)=U(K), K=1,N
C IPERM = 2, USING THE PERMUTATION IPT (P-TRANSPOSE) U = P'*W, W=U
C LET J = IPT(K) THEN U(K) = W(J), K=1,N. WE SET W(K) = U(K), K=1,N
C
GO TO 3
C
ENTRY LPERME(IPR,IPT,N)
GO TO 4
C
3 CONTINUE
IF(IPERM.EQ.2) GO TO 10
C
IPERM = 1
DO 20 K = 1,N
J = IPR(K)
20 W(K) = W(J)
DO 30 K = 1,N
30 W(K) = U(K)
GO TO 60
C
IPERM = 2
10 DO 40 K = 1,N
J = IPT(K)
40 W(K) = W(J)
DO 50 K = 1,N
50 W(K) = U(K)
60 CONTINUE
C
GO TO 10

C-----END OF LPERM-----------------------------------------------
4 RETURN
END
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 $n \times n$ real symmetric matrix $A$.

Sample Specifications of Input/Output Files for BLEVAL

---------------------------------------------
BLEVAL EXEC
FI O6 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 (DIM. 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 KM MAX 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 PRACT (NO. ITNS BEFORE TEST CONVERGENCE, TEST FRACTION
  25 .01
LINE 9 RELTOL MAXIT2 (PHASE 2, CONVERGE. TOL. , Max Ax-mults
  .000000001 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

$$B \equiv P C P^T, \quad C \equiv (SCALE) * A + (SHIFT) * I,$$

(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, $\lambda_i$, $1 \leq i \leq q$, of $B^{-1}$ and corresponding orthonormal real vectors $X_q \equiv (x_1, \ldots, x_q)$ such that

$$B^{-1}X_q = X_q A_q, \quad A_q \equiv X^T_q A X_q.$$  

(9.1.2)

Typically, $A_q = A_p$, a diagonal matrix whose nonzero entries are the eigenvalues $\lambda_i$. 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 LANCZI 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 $B$, 

$$B = LL^T$$

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, $B = PCP^T$. 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. 10596   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 are NOT to be
C incorporated into any commercial codes or used for any other
C commercial purposes such as consulting for other companies,
C without legal agreements with the authors of these Codes.
C If these Codes or portions of them are used in other scientific or
C engineering research works the names of the authors of these codes
C and appropriate references to their written work are to be
C incorporated in the derivative works.
C This header is not to be removed from these codes.
C BLI00090
C BLI00100
C REFERENCE: Cullum and Willoughby, Chapter 7,
C Lanczos Algorithms for Large Symmetric Eigenvalue Computations
C VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in
C Applied Mathematics, 2002. SIAM Publications,
C Philadelphia, PA. USA
C BLI00210
C BLI00220
C BLI00225
C CONTAINS MAIN PROGRAM FOR COMPUTING A FEW EIGENVALUES
C AND CORRESPONDING EIGENVECTORS OF A REAL SYMMETRIC MATRIX
C BY COMPUTING A FEW OF THE ALGEBRAICALLY-LARGEST OR
C ALGEBRAICALLY-SMALLEST EIGENVALUES OF THE INVERSE OF A SCALED,
C SHIFTED, AND PERMUTED VERSION B OF THE ORIGINAL A-MATRIX
C USING A BLOCK FORM OF LANCZOS TRIDIAGONALIZATION WITH LIMITED
C REORTHOGONALIZATION. THIS BLOCK PROCEDURE IS ITERATIVE AND
C REQUIRES A SUBROUTINE BLSOLV THAT FOR ANY GIVEN VECTOR W
C Computes U SUCH THAT B*U = W. THE SAMPLE BLSOLV SUBROUTINES
C PROVIDED FOR SPARSE MATRICES ARE ONLY FOR THE CASE THAT B IS
C POSITIVE DEFINITE AND USE THE CHOLESKY FACTORS OF B. HOWEVER,
C THE USER COULD REPLACE THESE BY A SUBROUTINE WHICH COMPUTES
C FOR AN INDEFINITE MATRIX THE FACTORIZATION L*D*(L-TRANSPOSE).
C BLI00300
C BLI00310
C BLI00320
C BLI00330
C BLI00340
C BLI00350
C THIS BLOCK PROCEDURE Computes THE ALGEBRAICALLY-LARGEST
C EIGENVALUES OF THE INVERSE OF THE B-MATRIX, UNLESS THE USER
C SUPPLIES -(B-INV)X RATHER THAN (B-INV)*X, IN WHICH
C CASE IT COMPUTES THE CORRESPONDING ALGEBRAICALLY-SMALLEST
C EIGENVALUES OF (B-INV) BY COMPUTING THE ALGEBRAICALLY-
C LARGEST EIGENVALUES OF -(B-INV). IN THIS CASE THE SIGNS
C OF THE COMPUTED EIGENVALUES ARE CHANGED PRIOR TO WRITING TO
C FILE 15 SO THAT ON EXIT, FILE 15 CONTAINS THE ALGEBRAICALLY-
C SMALLEST EIGENVALUES OF B-INVERSE ALONG WITH THE CORRESPONDING
C BLI00360
C BLI00370
C BLI00380
C BLI00390
C BLI00400
C BLI00410
C BLI00420
C BLI00430
C BLI00440
EIGENVALUES OF THE ORIGINAL A-MATRIX AND CORRESPONDING EIGENVECTORS. THE MATRIX \( B = A^*P*A + \text{SHIFT}*I \) WHERE THE SCALE SO AND SHIFT READ IN THIS PROGRAM, AND THE PERMUTATION \( P \) IS DEFINED IN THE CORRESPONDING USPEC SUBROUTINE. THE PROGRAM ASSUMES THAT THE FACTORIZATION READ IN USPEC CORRESPONDS TO THE SO, SHIFT AND PERMUTATION READ IN. THE SO AND SHIFT ARE CHOSEN SO THAT THE DESIRED EIGENVALUES ARE AT THE EXTREME OF THE SPECTRUM OF \( B^{-1} \).

THIS IS AN ITERATIVE 'BLOCK' LANCZOS PROCEDURE FOR WHICH ON EVERY ITERATION, THE 2ND AND SUCCEEDING BLOCKS CONTAIN ONLY ONE VECTOR WHICH IS SELECTED ON THE BASIS OF ITS EXPECTED INFLUENCE ON THE CONVERGENCE. Q-BLOCKS GENERATED ON A GIVEN ITERATION ARE REORTHOGONALIZED ONLY W.R.T. THOSE VECTORS IN THE FIRST Q-BLOCK WHICH ARE NOT ALLOWED TO GENERATE DESCENDANTS ON THAT ITERATION.

PFORT VERIFIER IDENTIFIED THE FOLLOWING NONPORTABLE CONSTRUCTIONS:

1. DATA MACHEP DEFINITION
2. FORMAT (20A4) USED FOR READING EXPLANATORY COMMENTS.
3. FREE FORMAT (5,*), USED FOR PARAMETER INPUT FROM FILE 5.
4. COMMON/LOOPS/ AS CONSTRUCTED IS NOT PORTABLE

DOUBLE PRECISION Q(44000),E(50),TM(2500),TD(50),TD(50),EPSM,NNZ
DOUBLE PRECISION SM(100),ERRMAX,SPREC,MACHEP,AVER,RELTL,ERRMAN
DOUBLE PRECISION EVAL, RESIDL(100), RESIDK(100), RESID, FRACT
DOUBLE PRECISION SO,SHIFT
REAL EXPLAN(20),G(2000)
INTEGER DIR(2,100),DESC(100),LEFT(100),XLFT(100)
INTEGER SEED,OFLAG,EFLAG
COMMON/LOOPS/MAXIT,ITER
COMMON /RANDOM/SEED
COMMON/FLAGS/EFLAG,OFLAG
DOUBLE PRECISION DABS, DFLOAT

EXTERNAL BL SOLV
DATA MACHEP/Z3410000000000000/
9.2. **BLENVAL: MAIN PROGRAM, EIGENVALUE AND EIGENVECTOR COMPUTATIONS** 573

```c
C OUTPUT HEADER
WRITE(6,10) BLIO1000
WRITE(6,10) BLIO1010
10 FORMAT(/' BLOCK LANCZOS PROCEDURE, USES FACTORED INVERSE OF A USER'BLIO1020
1-SPECIFIED MATRIX'/ 2ND AND SUCCEEDING BLOCKS GENERATED ON EACH BLIO1030
BLOCK ITERATION '/' CONTAIN ONLY ONE VECTOR'/) BLIO1040
C BLIO1050
C SET PROGRAM PARAMETERS BLIO1060
EPSM = 2.DO*MACHEP BLIO1070
SPREC = 1.D-5 BLIO1080
MPMIN = -1000 BLIO1090
C BLIO1100
C READ USER-SPECIFIED PARAMETERS FROM INPUT FILE 5 (FREE FORMAT) BLIO1110
C BLIO1120
C SELECT THE AMOUNT OF INTERMEDIATE OUTPUT DESIRED (IWRITE =0,1). BLIO1130
C IWRITE = 1 INCREASES THE AMOUNT OF INTERMEDIATE OUTPUT WRITTEN BLIO1140
C TO FILE 6 ON EACH ITERATION OF THE BLOCK LANCZOS PROCEDURE. BLIO1150
READ(5,20) EXPLAIN BLIO1160
20 FORMAT(20A4) BLIO1170
READ(5,* ) IWRITE BLIO1180
C BLIO1190
C READ ORDER (N) OF MATRIX AND MATRIX IDENTIFICATION NUMBER (MATNO) BLIO1200
C READ SCALE (SO) AND SHIFT (SHIFT) APPLIED TO MATRIX AND BLIO1210
C FLAG JPERM. JPERM = (0,1): JPERM = 1 MEANS MATRIX HAS BEEN BLIO1220
C PERMUTED
READ(5,20) EXPLAIN BLIO1230
READ(5,* ) N,MATNO,SO,SHIFT,JPERM BLIO1240
C BLIO1250
C READ USER-SPECIFIED DIMENSIONS OF Q-ARRAY (MDIMQ) AND OF THE BLIO1260
C TM-ARRAY (MDIMTM). READ MAXIMUM NUMBER (MAXIT) OF CALLS TO THE BLIO1270
C BLSOLV SUBROUTINE ALLOWED IN PHASE 1. BLIO1280
READ(5,20) EXPLAIN BLIO1290
READ(5,* ) MDIMQ, MDIMTM, MAXIT BLIO1300
C BLIO1310
C READ FLAGS: EFLAG = (0,1). EFLAG = 0, MEANS PROGRAM STOPS BLIO1320
C AFTER COMPLETING PHASE 1 PORTION OF BLOCK LANCZOS PROCEDURE. BLIO1330
C EFLAG = 1, MEANS PROGRAM COMPLETES BOTH PHASES BEFORE BLIO1340
C TERMINATING.
C OFLAG = (0,1). OFLAG = 0, MEANS THAT IN PHASE 1 PORTION BLIO1350
C OF THE COMPUTATION, THE PROGRAM DOES NO ORTHOGONALITY CHECKS BLIO1360
C ON THE Q-BLOCKS GENERATED. OFLAG = 1 MEANS THAT IN THE BLIO1370
C PHASE 1 PORTION AND IN THE PHASE 2 PORTIONS OF THE COMPUTATIONS BLIO1380
C THE PROGRAM CHECKS THE ORTHOGONALITY OF THE Q-BLOCKS GENERATED BLIO1390
C W.R.T. THAT VECTOR IN THE FIRST BLOCK THAT IS GENERATING BLIO1400
C DESCENDANTS. NOTE THAT IN PHASE 2, THE PROGRAM ALWAYS MAKES BLIO1410
C THIS CHECK OF ORTHOGONALITY REGARDLESS OF THE VALUE OF OFLAG. BLIO1420
C FOR SAFETY, OFLAG SHOULD ALWAYS BE SET TO 1, ALTHOUGH FOR MANY BLIO1430
C PROBLEMS THIS IS NOT NECESSARY.
READ(5,20) EXPLAIN BLIO1440
READ(5,* ) EFLAG,OFLAG BLIO1450
C BLIO1460
C READ SEED USED BY SUBROUTINE GERNAN TO OBTAIN THOSE STARTING BLIO1470
C VECTORS WHICH ARE GENERATED RANDOMLY.
READ(5,20) EXPLAIN BLIO1480
READ(5,* ) SEED BLIO1490
C BLIO1500
C C BLIO1510
C READ(5,* ) SEED BLIO1520
READ(5,* ) SEED BLIO1530
C BLIO1540
```
C SPECIFY MAXIMUM T-SIZE ALLOWED (KMAX-1); INITIAL SIZE OF  
C STARTING BLOCK (KACT); NUMBER OF STARTING VECTORS SUPPLIED (KSET)BLI01560  
C SEE BLOCK LANZOS HEADER FOR COMMENTS REGARDING THE SIZE OF KACT. BLI01570  
READ(5,20) EXPLAIN  
READ(5,* ) KMAX,KACT,KSET  
C BLI01590  
C SPECIFY NUMBER (KM) OF EXTREME EIGENVALUES AND EIGENVECTORS  
C OF B-INVERSE TO BE COMPUTED. THE BLOCK PROCEDURE WORKS WITH THE  
C INVERSE OF THE MATRIX B = SO*P*A*P’ + SHIFT*I, USING A  
C FACTORIZATION OF B. TO INDICATE THAT THE ALGEBRAICALLY-  
C SMALLEST EIGENVALUES OF B-INVERSE ARE BEING COMPUTED SET KM < 0. BLI01650  
C IF KM < 0, THE PROGRAM ASSUMES THAT BLSOLV SUBROUTINE WHICH  
C THE USER HAS PROVIDED IS COMPUTING -(B-INVERSE)*X  
C INSTEAD OF (B-INVERSE)*X AND INTERNALLY IT COMPUTES THE |KM|  
C ALGEBRAICALLY-LARGEST EIGENVALUES OF -(B-INVERSE).  
READ(5,20) EXPLAIN  
READ(5,* ) KM  
IF(KM.EQ.0) GO TO 540  
KML = IABS(KM)  
C BLI01670  
C STAGNATION OF CONVERGENCE OF THE KM-TH EIGENVALUE WILL BE  
C TESTED AFTER NSTAG ITERATIONS. CONVERGENCE WILL BE SAID TO  
C HAVE STAGNATED IF THE RATIO OF THE SQUARE OF THE CURRENT KM-TH  
C RESIDUAL TO THE SQUARE OF THE CORRESPONDING RESIDUAL OBTAINED  
C 10 ITERATIONS EARLIER IS GREATER THAN FRAC. NSTAG SHOULD BE  
C >= 25. FRACT WAS SET EQUAL TO .01 IN THE TESTS.  
READ(5,20) EXPLAIN  
READ(5,* ) NSTAG, FRACT  
C BLI01680  
C READ IN THE RELATIVE TOLERANCE (RELTOL) USED TO DETERMINE A  
C CONVERGENCE CRITERION FOR PHASE 2, AND THE MAXIMUM NUMBER (MAXIT2)BLI01850  
C OF CALLS TO SUBROUTINE BLSOLV ALLOWED IN PHASE 2.  
READ(5,20) EXPLAIN  
IF(FFLAG.EQ.1) READ(5,* ) RELTOL, MAXIT2  
C BLI01880  
C CONSISTENCY CHECKS  
C PROCEDURE REQUIRES ENOUGH ROOM IN THE Q-ARRAY FOR AT LEAST 2  
C BLOCKS OF SIZE KACT PLUS A WORKING VECTOR OF LENGTH N.  
C BLI01910  
C MUBLK = KMAX -1  
MUBLK2 = MUBLK*MUBLK  
IF(MDIMT.LT.MUBLK2) GO TO 520  
C BLI01950  
NMAX = N*KMAX  
IF(MDIMQ.LT.NMAX) GO TO 560  
IF(KML.GT.KACT) GO TO 420  
IF(MUBLK.GT.N) GO TO 440  
IF(2*KACT.GT.MUBLK) GO TO 500  
C BLI02000  
C-------------------------------------BLI02020  
C DEFINE AND INITIALIZE THE ARRAYS NEEDED TO DEFINE THE  
C FACTORIZATION OF THE B-MATRIX. PASS THE STORAGE LOCATIONS  
C OF THESE ARRAYS TO THE SUBROUTINE BLSOLV.  
C CALL USPEC(N,MATNO,NNZ,AYER)  
C BLI02070  
C-------------------------------------BLI02090
9.2. **BZEVAL: MAIN PROGRAM, EIGENVALUE AND EIGENVECTOR COMPUTATIONS** 575

C    MASK OVERFLOW AND UNDERFLOW
CALL MASK
C
C    ARE THERE STARTING VECTORS TO READ IN FROM FILE 10 (KSET.NE.0) ?
IF(KSET.EQ.0) GO TO 70
C
READ(10,30) NOLD,KACT
30 FORMAT(I6,I4)
     IF(NOLD.NE.N.OR.KSET.GT.KACT) GO TO 460
     DO 50 J=1,KSET
     READ(10,20) EXPLAN
     READ(10,40) EVAL,RESID
50 FORMAT(E20.12,E13.4)
     READ(10,20) EXPLAN
     LINT= (J-1)*N + 1
     LFIN = J*N
     50 READ(10,60) (Q(JL), JL = LINT,LFIN)
60 FORMAT(4E20.12)
C
70 CONTINUE
C
C    WRITE TO A SUMMARY OF THE PARAMETERS FOR THIS RUN TO FILE 6
C
MXBLK = KMAX - 1
WRITE(6,80) N, NNZ, AVER, MATNO
80 FORMAT(6X,'ORDER OF B-MATRIX ',5X,'AVERAGE NUMBER NONZEROES PER?
10W IN FACTOR'/
     1115,E47.4/3X,'CRUDE ESTIMATE OF SIZE NONZERO ENTRIES',5X,'MATRIX
     1D'/E31.4,I21/)
C
WRITE(6,90) SO, SHIFT
90 FORMAT(6X,'SCALE USED ON A-MATRIX',5X,'SHIFT USED ON A-MATRIX'/
1E26.4,E27.4/)
C
WRITE(6,100) MDIMQ, MDIMTM
100 FORMAT(6X,'USER-SPECIFIED'/2X,'MAX. DIMENSION Q-ARRAY',4X,'MAX.
1DIMENSION TM-ARRAY'/I16,I26/)
C
WRITE(6,110) OFLAG, EFLAG
110 FORMAT(6X,'OFLAG',4X,'EFLAG'/I8,I9/)
C
IF(OFLAG.EQ.1) WRITE(6,120) SPREC
120 FORMAT(6X,'ORTHOGONALITY TEST TOLERANCE'/E25.2)
C
IF(EFLAG.EQ.1) WRITE(6,130) MAXIT,RELTOL,MATIT2
130 FORMAT(6X,'MAXIT ',8X,'RELTOL ',6X,'MAXIT2 '/I10,E20.6,I12/)
C
IF(EFLAG.EQ.0) WRITE(6,140) MAXIT
140 FORMAT(6X,'MAXIT '/I10/)
C
WRITE(6,150) SEED
150 FORMAT(6X,' SEED FOR RANDOM NUMBER GENERATOR'/I24/)
C
IF(KM.GT.0) WRITE(6,160) KML
160 FORMAT(6X,' COMPUTE THE',I3,' ALGEBRAICALLY-LARGEST EIGENVALUES AND

CORRESPONDING VECTORS'/' OF THE INVERSE OF B = (SO*P*A*P-TRANS + BLI02650 1SHIFT*I') /
1SHIFT = BLI02650 IF(KM,LT.0) WRITE(6,170) KML
170 FORMAT('/' COMPUTE THE',I3,' ALGEBRAICALLY-SMALLEST EIGENVALUES AND BLI02680 1 CORRESPONDING VECTORS'/ OF THE INVERSE OF THE MATRIX B = (SO*P*BLI02690 1*P-TRANS + SHIFT*I')./' PROGRAM ASSUMES THAT USER IS PROVIDING -(BLI02700 1-INVERSE)*X INSTEAD OF (B-INVERSE)*X'/' AND COMPUTES THE ALGEBRAIC
1LALY-LARGEST EIGENVALUES OF -(B-INVERSE).'/.' HOWEVER ON EXIT, FILEBLI02720 1 CONTAINS THE ALGEBRAICALLY-SMALLEST EIGENVALUES'/' OF B-INVERSEBLI02730 1E, THE CORRESPONDING EIGENVALUES OF THE ORIGINAL A-MATRIX'/' AND TBLI02740 1E, CORRESPONDING EIGENVECTORS OF A.'/')
1C
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.DO*DFLOAT(N+1000) *NNZ*AVER*MACHEP BLI02800 IF(AVER.LT.1.) BLI02810
1ERRMAX = 2.DO*DFLOAT(N+1000) *NNZ*AVER**2*MACHEP BLI02820
C
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'/')
C
WRITE(6,190) BLI02890
190 FORMAT('/' ITERATIVE PROCEDURE'/.' PROCEDURE MONITORS THE SIZES OF TBLI02910 1HE NORM(GRADEENTS)**2 ON EACH'/.' ITERATION. CONVERGENCE IS SAID BLI02920 1TO HAVE OCCURRED WHEN ALL'/.' RELEVANT (NORMS)**2 ARE LESS THAN ERBLI02930 1MAX',E10.3'/.' PHASE 1 ERRMAX MAY YIELD SOMETHING LESS THAN SINGLE PRBLI02940 1ECISION ACCURACY'/.' PHASE 2 REFINES THE VECTORS OBTAINED ON PHASEBLI02950 1E 1, ACCORDING TO'/.' THE ACCURACY SPECIFIED BY THE USER'/')
C
WRITE(6,200) ERRMAX BLI02970
200 FORMAT('/' PHASE 1 CONVERGENCE CRITERION, ERRMAX '/E22.3') BLI02990
C
PASS STORAGE LOCATIONS OF VARIOUS ARRAYS TO LANCZS AND LANC11 BLI03000
C
SUBROUTINES BLI03020
CALL LANZP(DIR,DESC,SM,TM,TOD,TD,G,XLF,T,LEFT,SPREC)
CALL LANCP1(DIR,DESC,SM,XLF,LEFT)
C
-------------------------------BLI03010
C
-------------------------------BLI03020
C
ENTER PHASE 1 OF BLOCK LANCZOS PROCEDURE. BLOCK PROCEDURE BLI03040
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
C
IPHASE = 1 BLI03170
NITER = 0 BLI03180
210 ITER = 0 BLI03190
RESIDL(1) = FRAC{T}
RESIDL(2) = NSTAT
C
CALL INITIATES THE BLOCK LANCZOS PROCEDURE.
C ON RETURN EIGENVALUE APPROXIMATIONS ARE IN E(I),
C I = 1,KACT, IN ALGEBRAICALLY DECREASING ORDER.  CORRESPONDING
C EIGENVECTOR APPROXIMATIONS ARE IN FIRST N*KACT LOCATIONS IN
C THE Q-ARRAY.

CALL LANCZOS(BLSOLV,KML,KSET,KACT,MXBLK,N,Q,E,RESIDL,RESIDK,ERRMAX,BL0I3300
1 IPHASE,NITER,IWRITE)

C
IF(IPHASE.EQ.MPMIN) WRITE(15,220) N,KACT
220 FORMAT(2I10,' PHASE 2 TERMINATED '/' PROGRAM INDICATES ACCURACY SPBL0I3360
1ECIFIED BY USER IS NOT ACHIEVABLE/')

ITERA = IABS(ITER)
IF(IWRITE.NE.MPMIN.AND.ITER.GT.0) WRITE(6,230) IPHASE,ITERA
230 FORMAT(/1X,'PHASE COMPLETED',5X,'NUMBER CALLS TO BLSOLV SUBROUTINE
1E USED'/I10,I32)

C
IF(IWRITE.EQ.MPMIN.OR.ITER.LT.0) WRITE(6,240) IPHASE,ITERA
240 FORMAT(/1X,'PHASE TERMINATED',5X,'NUMBER CALLS TO BLSOLV SUBROUTINE
1E USED'/I10,I32)

C
IF(ITER.GT.0.AND.IWRITE.NE.MPMIN) GO TO 270

C
IF(ITER.LT.0) WRITE(6,250)
250 FORMAT('// SMALL EIGENVALUE SUBROUTINE DEFAULTED//' BLOCK LANCZOS BL0I3510
1 PROCEDURE STOPS AFTER SAVING CURRENT EIGENVECTOR APPROXIMATIONS/BL0I3520
1/)

C
WRITE(15,260)
WRITE(6,260)
260 FORMAT('// BLOCK LANCZOS PROCEDURE TERMINATES WITHOUT CONVERGENCE BL0I3570
1//' USER SHOULD EXAMINE OUTPUT TO DETERMINE REASONS FOR TERMINATION/BL0I3580
1//')

C
WRITE EIGENVALUE AND EIGENVECTOR APPROXIMATIONS CONTAINED IN
C THE FIRST Q-BLOCK TO FILE 15

270 IF(IPHASE.EQ.1) WRITE(15,280) N,KACT,SEED
280 FORMAT(16,I4,I12,' PHASE 1, ORDER A-MATRIX, SIZE OF Q(1), SEED')
IF(IPHASE.EQ.2) WRITE(15,290) N,KACT,SEED
290 FORMAT(16,I4,I12,' PHASE 2, ORDER A-MATRIX, SIZE OF Q(1), SEED')

C
C PERMUTE THE EIGENVECTORS IF NECESSARY
IF(JPERM.EQ.0) GO TO 310
LINT = -N + 1
KACT1 = KACT*N + 1
DO 300 J = 1,KACT
LINT = LINT + N
300 C

C
C
C--------------------------------------------------------------------------BLI03750
IPERM = 2 BLI03760
CALL LPERM(Q(LINT),Q(KACT),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.DO/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 WRITE(15,330) E(J), SM(JJ),TD(J) BLI04010
340 WRITE(15,360) (Q(L), L=LINT,LFIN) BLI04030
340 WRITE(15,360) (Q(L), L=LINT,LFIN) BLI04040
350 FORMAT(4E20.12) BLI04050
360 FORMAT(''/ ABOVE ARE COMPUTED APPROXIMATE EIGENVECTORS'/) BLI04060
C BLI04070
C IF(ITER.GT.MAXIT) WRITE(15,370) ITER,MAXIT BLI04080
370 FORMAT(''/ 1 SUBROUTINE',I6//' EXCEEDED MAXIMUM NUMBER ',I6,' ALLOWED'/) BLI04090
C BLI04100
C IF(ITER.LT.0) WRITE(15,380) BLI04110
380 FORMAT(''/ USER BEWARE. EIGENELEMENT COMPUTATIONS DEFAULTED BECAU
380 FORMAT(''/ USER BEWARE. EIGENELEMENT COMPUTATIONS DEFAULTED BECAU
1SE//' EISPACK SUBROUTINE DEFAULTED. EIGENVALUE AND EIGENVECTOR BLI04120
1 APPROXIMATIONS'/ ABOVE WERE THOSE AVAILABLE AT THE TIME OF DEFBLI04130
1AULT'/ SOMETHING IS SERIOUSLY WRONG.'/) BLI04140
C BLI04150
C CHECK FOR TERMINATION AFTER PHASE 1 BLI04160
C IT<0 MEANS EISPACK SUBROUTINE DEFAULTED BLI04170
C IPHASE = MPMIN MEANS THAT PHASE 2 TERMINATED DUE TO ORTHOGONALITY BLI04180
C IWRI = MPMIN MEANS THAT CONVERGENCE APPEARS TO HAVE STAGNATED BLI04190
C ITER > MAXIT MEANS MAXIMUM NUMBER OF CALLS TO BLSOLV BLI04200
C ALLOWED BY USER WAS EXCEEDED BLI04210
IF(ITER.LT.0.OR.ITER.GT.MAXIT) GO TO 580 BLI04220
IF(IPHASE.EQ.MPMIN.OR.IWRITE.EQ.MPMIN) GO TO 580 BLI04230
IF(EFLAG.NE.1.OR.IPHASE.EQ.2) GO TO 580 BLI04240
C BLI04250
C ENTER 2ND PHASE OF COMPUTATION TO ATTEMPT TO OBTAIN MORE BLI04260
C

9.2. BLIEVAL: MAIN PROGRAM, EIGENVALUE AND EIGENVECTOR COMPUTATIONS

C ACCURATE EIGENVECTOR APPROXIMATIONS.
C USER CONTROLS THE SIZE OF THE ERROR TOLERANCE BY SPECIFYING
C THE PARAMETER RELTOL.
C
IPHASE = 2
MAXIT = MAXIT2
KSET = KACT
C
C ERROR TOLERANCE USES THE CONVERGED EIGENVALUE LARGEST IN
C MAGNITUDE.
C
TD(1) = DABS(E(1))
IF(KML.EQ.1) GO TO 400
DO 390 J = 2,KML
390 IF(DABS(E(J)).GT.TD(1)) TD(1) = DABS(E(J))
C
400 TD(1) = DMAX1(TD(1),1.DO)
ERRMAN = RELTOL**2 * TD(1)**2
IF(ERRMAN.GE.ERRMAX) GO TO 480
ERRMAX = ERRMAN
C
WRITE(6,410) ERRMAX, MAXIT2

410 FORMAT(//' ENTER PHASE 2 OF COMPUTATION'//' CONVERGENCE CRITERION IBL104500
1IS REDUCED TO ',E13.4/' NO MORE THAN ',I5,' CALLS TO SUBROUTINE BLSBL104510
10LW WILL BE ALLOWED.'//' PROGRAM WILL TERMINATE IF BLOCK ORTHOGONALIBL104520
1ITY PROBLEMS MATERIALIZE')/'

C
GO TO 210
C
C INCONSISTENCIES IN THE DATA
C
420 WRITE(6,430) KM,KACT
430 FORMAT(//' PROGRAM TERMINATES BECAUSE THE NUMBER OF EIGENVALUES
1REQUESTED, KM =',I3,' IS LARGER THAN THE SIZE OF THE FIRST Q BLOCKIBL104610
1K, KACT =',I3,' SPECIFIED'//' USER MUST RESET KM OR KACT')/'
GO TO 580
C
C
440 WRITE(6,450) KMAX,N
450 FORMAT(//' PROGRAM TERMINATES BECAUSE KMAX = ',I5,' IS TOO LARGE FOBIBL104660
1R THE SIZE, N = ',I5,' OF THE GIVEN MATRIX'//' USER MUST DECREASEIBL104670
1THE SIZE OF KMAX.')/
GO TO 580
C
C
460 WRITE(6,470) NOLD,N,KACT,KSET
470 FORMAT(//' PROGRAM TERMINATES BECAUSE FAULT OCCURRED IN READING IN BL104720
1THE EIGENVECTOR APPROXIMATIONS'//' EITHER THE SIZE MATRIX SPECIFIEDIBL104730
1ON THE EIGENVECTOR FILE',,I6,' DID NOT MATCH THE SIZE SPECIFIED 'BL104740
1,15,' IN THE PROGRAM OR THE NUMBER'//' OF VECTORS IN FILE 10 = ',BL104750
1,14,' IS LESS THAN THE NUMBER ',I3,' USER SAID WERE THERE')/'
GO TO 580
C
C
480 WRITE(6,490) ERRMAN, ERRMAX
490 FORMAT(//' COMPUTED PHASE 2 CONVERGENCE CRITERION ',E13.4/' IS LARGERIBL104800
1GER THAN PHASE 1 CRITERION ',E13.4/' SO PROGRAM TERMINATES')/
GO TO 580
C
C
500 WRITE(6,510) KACT,MXBLK
510 FORMAT(/' PROGRAM TERMINATES BECAUSE THERE IS NOT ENOUGH ROOM TO' / BLI04850
 1 GENERATE 2 BLOCKS', ' BECAUSE KACT = ',I3,' AND MXBLK = ', I4/) / BLI04860
  GO TO 580 / BLI04870
  C / BLI04880
  C / BLI04890
  520 WRITE(6,530) MDIMT, 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**
C **IBM Research, T.J. Watson Research Center
C **Yorktown Heights, N.Y. 10598
C * Los Alamos National Laboratory
C * Los Alamos, New Mexico 87544
C E-mail: cullum@lanl.gov
C
C These codes are copyrighted by the authors. These codes
C and modifications of them or portions of them are NOT to be
C incorporated into any commercial codes or used for any other
C commercial purposes such as consulting for other companies,
C without legal agreements with the authors of these Codes.
C If these Codes or portions of them are used in other scientific or
C engineering research works the names of the authors of these codes
C and appropriate references to their written work are to be
C incorporated in the derivative works.
C
C This header is not to be removed from these codes.
C
C REFERENCE: Cullum and Willoughby, Chapter 7,
C Lanczos Algorithms for Large Symmetric Eigenvalue Computations
C VOL. 1 Theory. Republished as Volume 41 in SIAM CLASSICS in
C Applied Mathematics, 2002. SIAM Publications,
C Philadelphia, PA. USA
C
C CONTAINS SUBROUTINES LANCZS AND SAMPLE USPEC AND BLSOLV
C USED BY THE VERSION OF THE BLOCK LANCZS ALGORITHMS FOR
C FACTORED INVERSES OF REAL SYMMETRIC MATRICES, BLIVAL.
C NOTE THAT SAMPLE BLSOLV FOR SPARSE MATRICES ASSUMES THAT
C B-MATRIX IS POSITIVE DEFINITE AND USES CHOLESKY FACTORS.
C HOWEVER, THE USER CAN DIRECTLY REPLACE THAT SUBROUTINE BY
C A SUBROUTINE FOR INDEFINITE MATRICES THAT COMPUTES THE
C GENERALIZED FACTORIZATION L*D*(L-TRANSPOSE).
C
C NONPORTABLE CONSTRUCTIONS:
C 1. THE ENTRY MECHANISM USED TO PASS THE STORAGE LOCATIONS
C OF THE FACTORIZATION OF THE MATRIX THAT WILL BE USED
C BY THE LANCZS SUBROUTINE TO THE SUBROUTINE BLSOLV.
C 2. IN THE SAMPLE USPEC AND BLSOLV SUBROUTINES PROVIDED:
C THE FREE FORMAT (7,*), THE FORMAT (20A4) USED FOR
C READING EXPLANATORY COMMENTS IN THE MATRIX SPECIFICATION
C FILES, AND THE HEX FORMAT (4220) USED IN THE USPECS.
C 3. THE common BLOCK: LOOPS
C
C------USPEC FOR FACTORED INVERSES OF REAL SYMMETRIC MATRICES----------------------BLI00400
C
C SUBROUTINE USPEC(N,MATNO,NNZ,AVER)
C SUBROUTINE USPEC(N,MATNO,NNZ,AVER)
C
C------------------------------------------------------------------------------------------BLI00450
DOUBLE PRECISION  BD(2200),BSD(10000),NNZ,AVER  
INTEGER  KCOL(2200),KROW(10000),IPR(2200),IPT(2200)  
blas0460
blas0470
blas0480
blas0490
blas0500
blas0510
blas0520
blas0530
blas0540
blas0550
blas0560
blas0570
blas0580
blas0590
blas0600
blas0610
blas0620
blas0630
blas0640
blas0650
blas0660
blas0670
blas0680
blas0690
blas0700
blas0710
blas0720
blas0730
blas0740
blas0750
blas0760
blas0770
blas0780
blas0790
blas0800
blas0810
blas0820
blas0830
blas0840
blas0850
blas0860
blas0870
blas0880
blas0890
blas0900
blas0910
blas0920
blas0930
blas0940
blas0950
blas0960
blas0970
blas0980
blas0990
blas1000
C------------------------------------------------------------------------BLI01010
60 CONTINUE
  
C------------------------------------------------------------------------BLI01020
C COMPUTE NNZ, THE AVERAGE NUMBER OF NONZEROS PER COLUMN, AND
C AVER, THE AVERAGE SIZE OF NONZERO ENTRIES IN THE FACTORS
C OF THE B-MATRIX. FROM THIS, ESTIMATE (TOO CRUDELY) THE
C AVERAGE FOR B-INVERSE AS AVER = 1/AVER.
C------------------------------------------------------------------------BLI01030
  ITCOL = 0
AVER = 0.DO
DO 70 K = 1,N
IF(ABS(BD(K)).EQ.0.DO) GO TO 70
ITCOL = ITCOL + 1
AVER = AVER + ABS(BD(K))
70 CONTINUE
NTCOL = ITCOL
DO 80 K = 1,N
ITCOL = ITCOL + 2*KCOL(K)
NNZ = DREAL(ITCOL)/DREAL(N)
DO 90 K = 1,NZS
90 AVER = AVER + ABS(BS(1,K))
AVER = AVER/DREAL(NZS + NTCOL)
AVER = 1.DO/AVER
C------------------------------------------------------------------------BLI01040
C PASS STORAGE LOCATIONS OF FACTORS TO INVERSION SUBROUTINE BLNIN
CALL BSOLVE(BS,BD,KCOL,KROW,N,NZT,NZL)
C------------------------------------------------------------------------BLI01050
C------------------------------------------------------------------------BLI01060
GO TO 120
C------------------------------------------------------------------------BLI01070
C DEFAULT EXIT
WRITE(6,110)
110 FORMAT(/' TERMINATE. PARAMETERS IN CHOLESKY FACTOR FILE?'/
1'DO NOT AGREE WITH THOSE SPECIFIED BY THE USER?'/)
STOP
C------------------------------------------------------------------------BLI01080
C------------------------------------------------------------------------BLI01090
C END OF USPEC
RETURN
END
C------------------------------------------------------------------------BLI01100
C----------------------------------------------------------------------BLI01110
C SUBROUTINE BLNIN(V,U)
SUBROUTINE CBSOLVE(V,U)
C----------------------------------------------------------------------BLI01120
C----------------------------------------------------------------------BLI01130
C----------------------------------------------------------------------BLI01140
C----------------------------------------------------------------------BLI01150
C----------------------------------------------------------------------BLI01160
C----------------------------------------------------------------------BLI01170
C----------------------------------------------------------------------BLI01180
C----------------------------------------------------------------------BLI01190
C----------------------------------------------------------------------BLI01200
C----------------------------------------------------------------------BLI01210
C----------------------------------------------------------------------BLI01220
C----------------------------------------------------------------------BLI01230
C----------------------------------------------------------------------BLI01240
C----------------------------------------------------------------------BLI01250
C----------------------------------------------------------------------BLI01260
C----------------------------------------------------------------------BLI01270
C----------------------------------------------------------------------BLI01280
C----------------------------------------------------------------------BLI01290
C----------------------------------------------------------------------BLI01300
C----------------------------------------------------------------------BLI01310
C----------------------------------------------------------------------BLI01320
C----------------------------------------------------------------------BLI01330
C----------------------------------------------------------------------BLI01340
C----------------------------------------------------------------------BLI01350
C----------------------------------------------------------------------BLI01360
C----------------------------------------------------------------------BLI01370
C----------------------------------------------------------------------BLI01380
C----------------------------------------------------------------------BLI01390
C----------------------------------------------------------------------BLI01400
C----------------------------------------------------------------------BLI01410
C----------------------------------------------------------------------BLI01420
C----------------------------------------------------------------------BLI01430
C----------------------------------------------------------------------BLI01440
C----------------------------------------------------------------------BLI01450
C----------------------------------------------------------------------BLI01460
C----------------------------------------------------------------------BLI01470
C----------------------------------------------------------------------BLI01480
C----------------------------------------------------------------------BLI01490
C----------------------------------------------------------------------BLI01500
C----------------------------------------------------------------------BLI01510
C----------------------------------------------------------------------BLI01520
C----------------------------------------------------------------------BLI01530
C----------------------------------------------------------------------BLI01540
ENTRY BSOLVE(BS,BD,KCOL,KROW,N,NZT,NZL)
GO TO 4

C---------------------------------------------------------------BLI01560
3 CONTINUE

ITER = ITER + 1
ZERO = 0.0D0
ONE = 1.0D0
C SOLVE B*U = V FOR U WHERE B = L*L'
C SET U = V. FIRST SOLVE L*U = U FOR U, THEN SOLVE L'*U = U FOR U
KL = 0
DO 10 K = 1,N
10 U(K) = V(K)
DO 30 K = 1,N
TEMP = U(K)/BD(K)
U(K) = TEMP
IF (KCOL(K).EQ.0.OR.K.EQ.N) GO TO 30
KF = KL + 1
KL = KL + KCOL(K)
DO 20 KK = KF,KL
20 KR = KROW(KK)
20 U(KR) = U(KR) - TEMP*BSD(KK)
30 CONTINUE

NP1 = N+1
KF = NZT + 1
DO 50 K = 1,N
L = NP1 - K
TEMP = U(L)
IF (KCOL(L).EQ.0.OR.L.EQ.N) GO TO 50
KL = KF - 1
KF = KF - KCOL(L)
DO 40 LL = KF,KL
40 LR = KROW(LL)
40 TEMP = TEMP - BSD(LL)*U(LL)
50 U(L) = TEMP/BD(L)
60 CONTINUE

C
4 RETURN
C---------------------------------------------------------------BLI01930
C--END OF BLSOLV-----------------------------------------------BLI01940
C---------------------------------------------------------------BLI01960
C--SUBROUTINES FOR DIAGONAL TEST MATRICES----------------------BLI01970
C BLSOLV AND USPEC SUBROUTINES FOR DIAGONAL TEST MATRICES
C---------------------------------------------------------------BLI01980
C---------------------------------------------------------------BLI01990
C---------------------------------------------------------------BLI02000
C SUBROUTINE DBSOLV(Y,U)
SUBROUTINE BLSOLV(Y,U)
C---------------------------------------------------------------BLI02040
DOUBLE PRECISION V(1),U(1),D(1)
COMMON/LOOPS/Maxit,ITER
C---------------------------------------------------------------BLI02070
GO TO 3
C---------------------------------------------------------------BLI02080
C BELOW ENTRY IS FOR A DIAGONAL TEST MATRIX

BLI02100
9.3. **BLIMULT: SAMPLE MATRIX-VECTOR MULTPLY SUBROUTINES**

```plaintext
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
  30 CONTINUE                  BLI02190
  4 RETURN                    BLI02200
C----END OF 'DIAGONAL' TEST MATRIX BLISOLVE----------------------------------BLI02220
   END                      BLI02230
C
C----START OF USPEC FOR DIAGONAL TEST MATRIX--------------------------BLI02250
C
   SUBROUTINE USPEC(N,MATNO,NNZ,AVER)     BLI02260
   SUBROUTINE DUSPEC(N,MATNO,NNZ,AVER)     BLI02270
   C
C---------------------------------------------------------------BLI02290
   DOUBLE PRECISION D(1000),DI(1000),SHIFT,SPACE,NNZ,AVER       BLI02310
   DOUBLE PRECISION DABS, DFLOAT                     BLI02320
   REAL EXPLAN(20)                                      BLI02330
C---------------------------------------------------------------BLI02340
   C
   READ(7,10) EXPLAN                  BLI02350
   10 FORMAT(20A4)                        BLI02360
   READ(7,*), NOLD, NUNIF, SPACE, D(1), SHIFT        BLI02370
   NUNIF = NOLD - NUNIF                BLI02380
   WRITE(6,20) NOLD, SPACE, NUNIF, D(1), SHIFT     BLI02390
   20 FORMAT(//' DIAGONAL TEST MATRIX, SIZE = ',I4,' IS THE INVERSE OF MBLI02410
      MATRIX WITH MOST ENTRIES',E10.3,' UNITS APART AND WITH ',I3,' ENTRIEBLI02420
      IS IRREGULARLY SPACED',' FIRST ENTRY WAS ',E13.4,' SHIFT = ',E10.3 BLI02430
      1/)                                BLI02440
   C
   IF(N.NE.NOLD) GO TO 120               BLI02450
   C
   COMPUTE THE UNIFORM PORTION OF THE SPECTRUM                BLI02460
   DO 30 J=NOLD, NUNIF                                       BLI02470
   30 D(J) = D(1) - DFLOAT(J-1)*SPACE                       BLI02480
      NUNIF1=NUNIF + 1                                     BLI02490
   REA(D,10) EXPLAN                                      BLI02500
   DO 40 J=NUNIF1,N                                      BLI02510
   40 READ(7,*), D(J)                                     BLI02520
       NB = NUNIF - 2                                     BLI02530
   C
   IF(SHIFT.EQ.0.) GO TO 60                                BLI02540
   DO 50 J=1,N                                          BLI02550
   50 D(J) = D(J) + SHIFT                                  BLI02560
   C
   COMPUTE EIGENVALUES OF INVERSE FOR PRINTOUT ONLY        BLI02570
   60 DO 70 J = 1,N                                        BLI02580
   70 DI(J) = 1.0/D(J)                                    BLI02590
   WRITE(6,80) (J,DI(J), J=1,N )                          BLI02600
   80 FORMAT(//' INVERSE BLOCK LANCZOS TEST, LANCZS USES INVERSE OF GIVENBLI02640
      MATRIX',' ENTRIES OF INVERSE OF DIAGONAL TEST MATRIX = '/(I4,E20.1BLI02650
   )
```
12,(I4,E20.12,I4,E20.12))
C
C  DIAGONAL GENERATION COMPLETE
C
C  COMPUTE NNZ AND AVER
        NNZ = 1.DO
        AVER = 0.DO
        DO 90 K = 1,N
         90 AVER = AVER + DABS(DI(K))
        AVER = AVER/DFLOAT(N)
        AVER = 1.DO/AVER
C
C  COMPUTE THE GAPS
        N1 = N-1
        DO 100 K = 1,N1
         100 DI(K) = DI(K+1) - DI(K)
        WRITE(6,110) (K,DI(K), K=1,N1)
        110 FORMAT(/' GAPS BETWEEN EIGENVALUES/(I4,E13.4,I4,E13.4,I4,E13.4,I4)'/1,E13.4))
C
C---------------------------------------------------------------------------------------
C  PASS STORAGE LOCATIONS OF D AND N TO DSOLV SUBROUTINE
        CALL DSOLVE(D,N)
C---------------------------------------------------------------------------------------
C  RETURN
C
C---------------------------------------------------------------------------------------
C  WRITE(6,130) NOLD,N
        130 FORMAT(/' PROGRAM TERMINATES BECAUSE NOLD = ',I5,' DOES NOT EQUAL N'/1,'I5)
C---------------------------------------------------------------------------------------
C----END OF USPEC SUBROUTINE FOR 'DIAGONAL' TEST MATRICES-------------------------------
C
STOP
END
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 IWRIE (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 FRACQ (NO. ITNS BEFORE TEST CONVERGENCE, TEST FRACTION
25 .05
LINE 9 RELTOL MAXIT2 (PHASE 2,CONVERGE.TOL., Max Ax-Mults
.00000001 1000
```

---