ORNL-CSD-9

Eigensystem Computation for Skew-Symmetric Matrices and a Class of Symmetric Matrices

R. C. Ward
L. J. Gray

OAK RIDGE NATIONAL LABORATORY
OPERATED BY UNION CARBIDE CORPORATION FOR THE ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION

BLANK PAGE

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
Price: Printed Copy $4.00, Microfiche $2.25

ORNL/CSD-9
UC-32-Mathematics and Computers

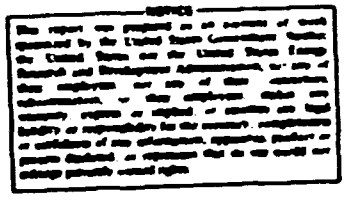
Contract No. W-7405-eng-26

COMPUTER SCIENCES DIVISION

EIGENSYSTEM COMPUTATION FOR SKEW-SYMMETRIC MATRICES AND A CLASS OF SYMMETRIC MATRICES

R. C. Ward and L. J. Gray

MAY 1976



UNION CARBIDE CORPORATION, NUCLEAR DIVISION
operating the
Oak Ridge Gaseous Diffusion Plant . Oak Ridge National Laboratory
Oak Ridge Y-12 Plant . Paducah Gaseous Diffusion Plant
for the
ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION

TABLE OF CONTENTS

Eigensystem Computation for Skew-Symmetric Matrices and a Class of Symmetric Matrices

R. C. Ward
L. J. Gray

ABSTRACT

We present an algorithm for computing the eigenvalues and (optionally) the eigenvectors of either a skew-symmetric matrix or a symmetric tridiagonal matrix with constant diagonal. The algorithm uses only orthogonal similarity transformations and is believed to be the most efficient algorithm available for computing all the eigenvalues or the complete eigensystem.

1. INTRODUCTION

In this paper, we consider the problem of computing the eigenvalues and eigenvectors of a skew-symmetric matrix. As these matrices occur in model calculations in nuclear physics, this question is of practical interest. Paardekooper [4] has presented a Jacobi-like algorithm which, like the Jacobi method, is elegant but somewhat inefficient. The method that we will discuss makes full use of the constant (zero) main diagonal.

Our method, after transforming to a tridiagonal system, is based upon a technique due to Golub [1]. This algorithm is designed for a matrix of dimension 2m x 2m; however, it is easily seen that a skew-symmetric matrix whose dimension is odd has zero as an eigenvalue. We give a method, based upon Givens transformations, for eliminating this eigenvalue, and thus reducing the dimension of the matrix to an even number.

One of the main advantages of this approach is the observation that the eigenvector matrix is half sparse. This results in significant savings over alternative algorithms. We also point out that this procedure can be applied to another class of matrices; namely, symmetric, tridiagonal matrices with a constant main diagonal.

2. DESCRIPTION OF ALGORITHMS

We now describe a three routine package for computing all the eigenvalues and, optionally, all the eigenvectors of a full skew-symmetric matrix or of a tridiagonal symmetric matrix with a constant

diagonal. The three subroutines and their functions are given below:

TRIZD--transforms an arbitrary real skew-symmetric matrix to skew-symmetric tridiagonal form using orthogonal similarity transformations and saving the pertinent information about these transformations.

IMZD--computes the eigenvalues and, optionally, the eigenvectors of a symmetric tridiagonal matrix with zeros on the diagonal or of a skew-symmetric tridiagonal matrix.

TBAKZD--computes the eigenvectors of an arbitrary real skew-symmetric matrix by back transforming the eigenvectors of the corresponding skew-symmetric tridiagonal matrix determined by TRIZD.

Subroutines TRIZD and TBAKZD are straight-forward adaptations of procedures TRED1 and TRBAK1 [3] which accomplish similar functions for arbitrary real symmetric matrices. Thus, we provide a listing of all three routines in the appendix, but we will confine our discussion to the subroutine IMZD.

The algorithm coded as subroutine IMZD has three basic ideas. First, a matrix of size $2m + 1$ is deflated to a $2m \times 2m$ matrix by removal of a zero eigenvalue. Second, the eigenvalue problem for the $2m \times 2m$ matrix is looked at as a singular value problem for an $m \times m$ bidiagonal matrix. Finally, it is recognized that the eigenvector matrix has a simple zero-non-zero structure.

IMZD determines the eigenvalues and eigenvectors of a $n \times n$ tridiagonal matrix $T_n(\pm)$ defined by:

$$T_n(\pm) = \begin{bmatrix} 0 & \alpha_2 & & & & \\ \pm\alpha_2 & 0 & \alpha_3 & & & \\ & \pm\alpha_3 & 0 & \ddots & & \\ & & \ddots & \ddots & \alpha_n & \\ & & & \pm\alpha_n & 0 & \end{bmatrix}.$$

The matrix $T_n(-)$ is skew-symmetric and can be orthogonally transformed into a matrix consisting of the direct sum of 1x1 and 2 x 2 blocks. The 1x1 blocks correspond to zero eigenvalues and the 2 x 2 blocks are skew-symmetric and correspond to a non-zero pair of purely imaginary complex conjugate eigenvalues. The matrix T(+) is symmetric; however, iT(+) is skew-hermitian. Thus, $T_n(+)$ can also be orthogonally transformed into a matrix consisting of the direct sum of 1x1 and 2 x 2 blocks. The 1x1 blocks again correspond to zero eigenvalues and the 2 x 2 blocks are symmetric and correspond to a non-zero pair of real eigenvalues of equal magnitude but opposite sign. The eigenvalues of a symmetric, tridiagonal matrix with a constant main diagonal, i.e., $T_n(+) + \sigma I$, are easily obtained by adding σ to the eigenvalues of $T_n(+)$.

We will assume throughout the remainder of this paper that $a_i \neq 0$ for $2 \leq i \leq n$ since the eigenvalue problem can always be partitioned into subproblems corresponding to matrices with this property. Thus, $T_n(\pm)$ has exactly one zero eigenvalue if n is odd and all non-zero eigenvalues if n is even.

2.1 Zero Eigenvalues

If n is odd, the algorithm begins by transforming $T_n(\pm)$ into the direct sum of a $T_{n-1}(\pm)$ matrix and a 1x1 zero block. To illustrate this finite process, consider the matrix $T_7(\pm)$:

$$\begin{bmatrix} 0 & \alpha_2 & & & & & \\ \pm\alpha_2 & 0 & \alpha_3 & & & & \\ & \pm\alpha_3 & 0 & \alpha_4 & & & \\ & & \pm\alpha_4 & 0 & \alpha_5 & & \\ & & & \pm\alpha_5 & 0 & \alpha_6 & \\ & & & & \pm\alpha_6 & 0 & \alpha_7 \\ & & & & & \pm\alpha_7 & 0 \end{bmatrix}.$$

Applying a Givens orthogonal similarity transformation (plane rotation [5, pp. 47-48]) to the fifth and seventh rows and columns to annihilate α_7 and $\pm\alpha_7$, we obtain the matrix:

$$\begin{bmatrix} 0 & \alpha_2 & & & & & \\ \pm\alpha_2 & 0 & \alpha_3 & & & & \\ & \pm\alpha_3 & 0 & \alpha_4 & & & \\ & & \pm\alpha_4 & 0 & \alpha_5' & & x \\ & & & \pm\alpha_5' & 0 & \alpha_6' & \\ & & & & \pm\alpha_6' & 0 & \\ & & & \pm x & & & 0 \end{bmatrix}.$$

Applying a Givens similarity transformation to the fourth and seventh rows and columns to annihilate x and $\pm x$, we obtain the matrix:

$$\begin{bmatrix} 0 & \alpha_2 & & & & & \\ \pm\alpha_2 & 0 & \alpha_3' & & & & y \\ & \pm x_3' & 0 & \alpha_4' & & & \\ & & \pm\alpha_4' & 0 & \alpha_5' & & \\ & & & \pm\alpha_5' & 0 & \alpha_6' & \\ & & & & \pm\alpha_6' & 0 & \\ & \pm y & & & & & 0 \end{bmatrix}.$$

Finally, a Givens similarity transformation applied to the second and seventh rows and columns annihilates y and $\pm y$ to obtain the desired matrix. It is obvious from the above example that we need not apply the transformations to both the rows and columns since they act on disjoint sets of non-zero elements and they preserve symmetry and skew-symmetry. Therefore, in general, this process requires the computation and application of (n-1)/2 Givens' transformations resulting in 5(n-1)/2 multiplications, (n-1) divisions, (n-1)/2 additions, and (n-1)/2 square roots.

2.2 Eigenvalues of $T_n(\pm)$ for n Even

In the previous section, we showed that the matrix for the eigenvalue problem can always be reduced to an even ordered matrix. This section discusses the reduction by similarity transformations of $T_n(\pm)$ to a matrix $E_n(\pm)$ defined by:

$$E_n(\pm) \equiv \begin{bmatrix} 0 & \lambda_1 & & & & & \\ \pm\lambda_1 & 0 & & & & & \\ & & 0 & \lambda_2 & & & \\ & & \pm\lambda_2 & 0 & & & \\ & & & & \ddots & & \\ & & & & & 0 & \lambda_{n/2} \\ & & & & & \pm\lambda_{n/2} & 0 \end{bmatrix} .$$

The eigenvalues of $T_n(+)$ are then easily recognized as $\pm\lambda_k$ and the eigenvalues of $T_n(-)$ are $\pm i\lambda_k$.

Golub [1] states that the eigenvalues of $T_n(+)$ are $\pm\lambda$, where λ is a singular value of the bidiagonal matrix J defined by:

$$J \equiv \begin{bmatrix} \alpha_2 & \alpha_3 & & & \\ & \alpha_4 & \alpha_5 & & \\ & & \alpha_6 & \ddots & \\ & & & \ddots & \alpha_{n-1} \\ & & & & \alpha_n \end{bmatrix} .$$

The QR-type iteration found in the singular value decomposition algorithm (Golub and Reinsch [2]) can then be used to find the singular values of J and thus, the eigenvalues of $T_n(+)$. Similarly, the eigenvalues of $T_n(-)$ are i times the $\pm$ singular values of J. Therefore, this segment of the algorithm reduces to performing the singular value QR-type iteration on the matrix $T_n(\pm)$. To illustrate one such iteration, consider the matrix $T_6(\pm)$ after the initial Givens orthogonal similarity transformation which enters the shift information has been applied. It has the form:

$$\begin{bmatrix} 0 & \alpha_2' & & x & & \\ \pm\alpha_2' & 0 & \alpha_3' & & & \\ & \pm\alpha_3' & 0 & \alpha_4' & & \\ \pm x & & \pm\alpha_4' & 0 & \alpha_5 & \\ & & & \pm\alpha_5 & 0 & \alpha_6 \\ & & & & \pm\alpha_6 & 0 \end{bmatrix}.$$

Applying another Givens similarity transformation to annihilate x and $\pm x$, we have:

$$\begin{bmatrix} 0 & \alpha_2'' & & & & \\ \pm\alpha_2'' & 0 & \alpha_3'' & & y & \\ & \pm\alpha_3'' & 0 & \alpha_4'' & & \\ & & \pm\alpha_4'' & 0 & \alpha_5' & \\ & \pm y & & \pm\alpha_5' & 0 & \alpha_6 \\ & & & & \pm\alpha_6 & 0 \end{bmatrix}.$$

Applying a third Givens similarity transformation to annihilate y and $\pm y$, we have:

$$\begin{bmatrix} 0 & \alpha_2'' & & & & \\ \pm\alpha_2'' & 0 & \alpha_3''' & & & \\ & \pm\alpha_3''' & 0 & \alpha_4''' & & z \\ & & \pm\alpha_4''' & 0 & \alpha_5'' & \\ & & & \pm\alpha_5'' & 0 & \alpha_6' \\ & & \pm z & & \pm\alpha_6' & 0 \end{bmatrix}.$$

Finally, a Givens similarity transformation applied to the fourth and sixth rows and columns to annihilate z and $\pm z$ completes the iteration. Again, it is obvious that we need not apply the transformations to both the rows and columns. Therefore, one iteration requires the computation and application of n-2 Givens transformations resulting in 5(n-2) multiplications, 2(n-2) divisions, (n-2) multiplications, and (n-2) square roots.

From Golub and Reinsch [2] and Wilkinson [6], we know that the singular value QR-type iteration converges globally and almost always cubically. Thus, the matrix $T_n(\pm)$ converges under this iteration to the desired matrix $E_n(\pm)$ consisting of the direct sum of 2 x 2 blocks.

2.3 Computation of the Eigenvector Matrix

The algorithm has an option for returning to the user the eigenvectors of the matrix $T_n(\pm)$. The three steps for computing these eigenvectors are described in this section. In addition, we briefly

explain the method of computing the eigenvectors of an arbitrary skew-symmetric matrix and why this method is preferred.

The first step in computing the eigenvectors of $T_n(\pm)$ involves determining the correct diagonal matrix used to accumulate the Givens transformations. The first Givens transformation in IMZD is applied to the rows of $T_n(\pm)$, the second to the columns, the third to the rows, etc. Therefore, the matrix elements of interest are α_2, $\pm\alpha_3$, α_4, $\pm\alpha_5$, ..., α_n. Since TRIZD returns the lower subdiagonals of $T_n(-)$, the algorithm requires these elements as input. Thus, the matrix elements of interest are given as input for $T_n(+)$, but they are not given as input for $T_n(-)$. To rectify this problem, we perform a diagonal similarity transformation on $T_n(-)$ where the diagonal elements d_{ii} are given by:

$$d_{ii} = \begin{cases} -1 & \text{if} \quad i = 0 \text{ or } 1 \pmod 4 \\ 1 & \text{if} \quad i = 2 \text{ or } 3 \pmod 4 \end{cases}.$$

Therefore, the matrix elements of interest for both $T_n(+)$ and $T_n(-)$ are the subdiagonal elements. The diagonal matrix used to accumulate the Givens transformations for $T_n(+)$ is the identity matrix and for $T_n(-)$ it is the diagonal matrix defined by d_{ii} above.

The second step in computing the eigenvectors of $T_n(\pm)$ involves accumulating the Givens transformations. By analyzing this particular iteration, one can reduce by a factor of 2 the number of operations which are normally required for accumulating Givens transformations in eigenvalue problems. Since only alternate columns are combined in

the accumulation and since the accumulation matrix was initially diagonal, we observe that the final accumulation matrix is 1/2 sparse and has the form illustrated below on an 6 x 6 matrix.

$$\begin{bmatrix} x & 0 & x & 0 & x & 0 \\ 0 & x & 0 & x & 0 & x \\ x & 0 & x & 0 & x & 0 \\ 0 & x & 0 & x & 0 & x \\ x & 0 & x & 0 & x & 0 \\ 0 & x & 0 & x & 0 & x \end{bmatrix}$$

Thus, the algorithm only has to recompute every other element in each column as it accumulates the transformations. The number of operations per iteration required for this accumulation is $(2n^2-4n)$ multiplications and (n^2-2n) additions.

If the matrix was skew-symmetric, i.e., $T_n(-)$, we easily observe that the real and imaginary parts of the eigenvector corresponding to $i\lambda k$ in $E_n(-)$ are found in the kth and (k+1)th columns, respectively, of the accumulated transformation matrix. Since complex conjugate eigenvalues have complex conjugate eigenvectors, the eigenvector corresponding to $-i\lambda k$ is also easily obtainable from the kth and (k+1)th columns. However, we must perform a third and final step if the eigenvectors of $T_n(+)$ are desired. From the form of $E_n(+)$, we observe that the eigenvector for λ_k is the sum of the kth and (k+1)th columns of the accumulated transformation matrix. The eigenvector for $-\lambda k$ is given by subtracting the (k+1) column from the kth column. Due to the structure

of the accumulation matrix, these eigenvectors can be stored in the kth and (k+1)th column, respectively, by replacement instructions.

To compute the eigenvectors of an arbitrary skew-symmetric matrix, our package transforms the eigenvectors of the tridiagonal skew-symmetric matrix rather than using the accumulated TRIZD transformation matrix in IMZD. This is different from the standard practice as represented by the codes in Wilkinson and Reinsch [7]. If we assume that two iterations are required to determine each eigenvalue, an operation count analysis indicates that the technique of only accumulating the transformations in IMZD and then transforming them in TBAKZD requires a total of $3\ 1/2\ n^3 + O(n^2)$ operations (counting both additions and multiplications). The technique of giving IMZD the accumulated transformation matrix from TRIZD and accumulating the IMZD transformations in this matrix requires $4\ 1/3\ n^3 + O(n^2)$ operations. Thus, substantial savings result even when this very conservative estimate for the number of iterations required to determine each eigenvalue is used.

3. TEST RESULTS

Subroutines TRIZD, IMZD, and TBAKZD have been tested extensively on the IBM 360/91 computer at Oak Ridge National Laboratory using double precision arithmetic (approximately 16.8 decimal digits). In this section, we present the results of some of these test cases.

Test Case A:

$$A = \begin{bmatrix} 0 & 1 & 0 & -5 & 0 & 0 & 0 & 2 \\ -1 & 0 & 0 & 0 & 5 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & -2 & -1 & 5 & 0 \\ 5 & 0 & 0 & 0 & -1 & -2 & 0 & 0 \\ 0 & -5 & 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 0 & 0 & 0 & -5 \\ 0 & 2 & -5 & 0 & 0 & 0 & 0 & 1 \\ -2 & 0 & 0 & 0 & 0 & 5 & -1 & 0 \end{bmatrix}$$

The eigenvalues are $\pm 2i$, $\pm 4i$, $\pm 6i$, and $\pm 8i$. Table 1 presents the eigenvalues computed by IMZD after A was transformed to tridiagonal form by TRIZD. The eigenvectors were also computed in IMZD and transformed to the eigenvectors of A by TBAKZD. These computed eigenvectors, similar to the eigenvalues, were accurate to the sixteenth decimal digit.

Test Case B:

$$B = \begin{bmatrix} 0 & -1 & -2 & 3 & 1 & 4 & -4 & 3 \\ 1 & 0 & -3 & 2 & 6 & -1 & 1 & -2 \\ 2 & 3 & 0 & -5 & -1 & 0 & 4 & 1 \\ -3 & -2 & 5 & 0 & -2 & 1 & 3 & 2 \\ -1 & -6 & 1 & 2 & 0 & 3 & -1 & -2 \\ -4 & 1 & 0 & -1 & -3 & 0 & -2 & 5 \\ 4 & -1 & -4 & -3 & 1 & 2 & 0 & -3 \\ -3 & 2 & -1 & -2 & 2 & -5 & 3 & 0 \end{bmatrix}$$

The eigenvalues of B are {0, 0, ±4i, ±8i, ±12i}. The computed eigenvalues are presented in Table 1. The eigenvectors were computed and again were accurate to the sixteenth decimal digit. This test case was included because the tridiagonal matrix resulting from TRIZD contains a zero subdiagonal. Thus, a partitioning of the problem into two sub-problems occurs in IMZD.

TABLE 1

Eigenvalues of A	Eigenvalues of B
±1.9999 99999 99999 6 i	0.0
±3.9999 99999 99999 6 i	0.0
±5.9999 99999 99999 8 i	±3.9999 99999 99999 6 i
±7.9999 99999 99999 3 i	±8.0000 00000 00000 2 i
	±12.000 00000 00000 0 i

Test Case C:

$$C = \begin{bmatrix} 0 & 2 & -2 & 0 & -4 \\ -2 & 0 & 4 & -5 & 3 \\ 2 & -4 & 0 & 1 & 1 \\ 0 & 5 & -1 & 0 & 2 \\ 1 & -3 & -1 & -2 & 0 \end{bmatrix}$$

The eigenvalues of C are {0, $\pm$4i, $\pm$8i}. The computed eigenvalues are presented in Table 2 with the computed eigenvectors having similar accuracy. This test case was included to test that segment of the algorithm described in Section 2.1 which computes the zero eigenvalue and deflates the matrix when n is odd.

Test Case D:

$$D = \begin{bmatrix} 0 & 1 & & & & & \\ 1 & 0 & 1 & & & & \\ & 1 & 0 & 1 & & & \\ & & 1 & 0 & 1 & & \\ & & & 1 & 0 & 1 & \\ & & & & 1 & 0 & 1 \\ & & & & & 1 & 0 \end{bmatrix}$$

The eigenvalues of D are $2 \cos \frac{k\pi}{7}$ for k = 1, 2, ..., 6. The computed eigenvalues are presented in Table 2 with the first inaccurate digit underlined.

TABLE 2

Eigenvalues of C	Eigenvalues of D
0.0	$\pm$ .44504 18679 12628 7
$\pm$3.9999 99999 99999 7 i	$\pm$1.2469 79603 71746 6
$\pm$7.9999 99999 99999 6 i	$\pm$1.8019 37735 80483 5

REFERENCES

[1] Golub, Gene H., "Least Squares, Singular Values and Matrix Approximations," Aplikace Matematiky 13 (1968), pp. 44-51.

[2] Golub, Gene H. and C. Reinsch, "Singular Value Decomposition and Least Squares Solutions," Numer. Math. 14 (1970), pp. 403-420.

[3] Martin, R. S., C. Reinsch, and J. H. Wilkinson, "Householder's Tridiagonalization of a Symmetric Matrix," Numer. Math. 11 (1968), pp. 181-195.

[4] Paardekooper, M. H. C., "An Eigenvalue Algorithm for Skew-Symmetric Matrices," Numer. Math. 17 (1971), pp. 189-202.

[5] Wilkinson, J. H., The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, 1965.

[6] Wilkinson, J. H., "Global Convergence of Tridiagonal QR Algorithm with Origin Shifts," Lin. Alg. and its Appl. 1 (1968), pp. 409-420.

[7] Wilkinson, J. H. and C. Reinsch, Handbook for Automatic Computation, Vol. II Linear Algebra, Springer Verlag, New York, 1971.

APPENDIX

```
      SUBROUTINE IHZD ( NAX, N, Z, MATZ, SKEW, Z, IERR )                IHZD0001
C                                                                       IHZD0002
C ****                                                                  IHZD0003
C                                                                       IHZD0004
C  FUNCTION  -  COMPUTE THE EIGENVALUES AND OPTIONALLY THE EIGENVECTORS IHZD0005
C                 OF A SYMMETRIC TRIDIAGONAL MATRIX WITH ZERO DIAGONALS IHZD0006
C                 OR A SKEW-SYMMETRIC TRIDIAGONAL MATRIX USING AN       IHZD0007
C                 IMPLICIT QR-TYPE ITERATION                            IHZD0008
C                                                                       IHZD0009
C  PARAMETERS                                                           IHZD0010
C                                                                       IHZD0011
C     NAX       - INPUT INTEGER SPECIFYING THE ROW DIMENSION OF Z AS    IHZD0012
C                   DECLARED IN THE CALLING PROGRAM DIMENSION STATEMENT IHZD0013
C                                                                       IHZD0014
C     N         - INPUT INTEGER SPECIFYING THE ORDER OF THE TRIDIAGONAL IHZD0015
C                   MATRIX                                              IHZD0016
C                                                                       IHZD0017
C     E(N)      - ON INPUT, ARRAY CONTAINING THE LOWER SUBDIAGONAL      IHZD0018
C                   ELEMENTS OF THE TRIDIAGONAL MATRIX IN ITS LAST N-1  IHZD0019
C                   POSITIONS. E(1) IS ARBITRARY.                       IHZD0020
C                 ON OUTPUT, ARRAY CONTAINS THE EIGENVALUES. THE NON-ZEROIHZD0021
C                   EIGENVALUES OCCUR IN PAIRS WITH OPPOSITE SIGNS AND  IHZD0022
C                   ARE FOUND IN ADJACENT LOCATIONS IN E. THE EIGENVALUESIHZD0023
C                   OF SYMMETRIC MATRICES ARE REAL AND THE EIGENVALUES  IHZD0024
C                   OF SKEW-SYMMETRIC MATRICES ARE PURELY IMAGINARY     IHZD0025
C                   COMPLEX NUMBERS. IF AN ERROR EXIT IS MADE, THE      IHZD0026
C                   EIGENVALUES ARE CORRECT FOR INDICES IERR+1,IERR+2...NIHZD0027
C                                                                       IHZD0028
C     MATZ      - INPUT LOGICAL VARIABLE SPECIFYING THE EIGENVECTOR     IHZL0029
C                   OPTION                                              IHZD0030
C                   = .TRUE.   EIGENVECTORS ARE TO BE COMPUTED          IHZD0031
C                   = .FALSE.  EIGENVECTORS ARE NOT TO BE COMPUTED      IHZD0032
C                                                                       IHZD0033
C     SKEW      - INPUT LOGICAL VARIABLE SPECIFYING TYPE OF INPUT MATRIX IHZD0034
C                   = .TRUE.   INPUT TRIDIAGONAL MATRIX IS SKEW-SYMMETRICIHZD0035
C                   = .FALSE.  INPUT TRIDIAGONAL MATRIX IS SYMMETRIC WITHIHZD0036
C                                ZERO DIAGONALS                         IHZD0037
C                   SKEW IS NOT REFERENCED IF MATZ = .FALSE.            IHZD0038
C                                                                       IHZD0039
C     Z(NAX,N)  - OUTPUT ARRAY CONTAINING THE ORTHOGONAL EIGENVECTORS   IHZD0040
C                   OF THE INPUT TRIDIAGONAL MATRIX. EIGENVECTORS CORRE-IHZD0041
C                   SPONDING TO ZERO EIGENVALUES ARE NORMALIZE TO UNIT   IHZD0042
C                   2-NORM (LENGTH) AND THOSE CORRESPONDING TO NON-ZERO  IHZD0043
C                   EIGENVALUES HAVE 2-NORM OF SQUARE ROOT 2. IF THE J-THIHZD0044
C                   EIGENVALUE IS ZERO OR REAL (I.E. E(J)), ITS EIGEN-   IHZD0045
C                   VECTOR IS FOUND IN THE J-TH COLUMN OF Z. IF THE J-TH IHZD0046
C                   EIGENVALUE IS IMAGINARY (I.E. E(J)*I) WITH E(J+1) =  IHZD0047
C                   -E(J), THE REAL PART OF ITS EIGENVECTOR IS FOUND IN  IHZD0048
C                   THE J-TH COLUMN OF Z AND ITS IMAGINARY PART FOUND IN IHZD0049
C                   THE (J+1)-TH COLUMN. IF AN ERROR EXIT IS MADE, Z     IHZD0050
C                   CONTAINS THE EIGENVECTORS ASSOCIATED WITH THE STORED IHZD0051
C                   EIGENVALUES.                                         IHZD0052
C                   Z IS NOT REFERENCED IF MATZ = .FALSE.               IHZD0053
C                                                                       IHZD0054
C     IERR      - OUTPUT ERROR CODE                                     IHZD0055
```

```
C                  = 0   NORMAL RETURN (ALL EIGENVALUES/VECTORS FOUND)   IHZD0056
C                  = J   IF THE J-TH EIGENVALUE HAS NOT BEEN DETERMINED  IHZD0057
C                        AFTER 30 ITERATIONS                             IHZD0058
C                                                                        IHZD0059
C  REQUIRED FUNCTIONS - DABS,DSIGN,DSQRT,MOD                             IHZD0060
C                                                                        IHZD0061
C  MACHINE DEPENDENCE - EPS SPECIFIES THE RELATIVE PRECISION OF FLOATINGIHZD0062
C                       POINT ARITHMETIC. EPS = 16.D0**(-13) FOR LONG  IHZD0063
C                       FORM ARITHMETIC ON IBM S360.                     IHZD0064
C                                                                        IHZD0065
C  REFERENCE  -  L.J. GRAY AND B.C. HARR, EIGENSYSTEM COMPUTATION FOR    IHZD0066
C                SEMI-SYMMETRIC MATRICES AND A CLASS OF SYMMETRIC        IHZD0067
C                MATRICES,  OCCSD-CSD REPORT, 1976.                      IHZD0068
C                                                                        IHZD0069
C ****                                                                   IHZD0070
C                                                                        IHZD0071
      REAL*8 Z(M),Z(MAX,M)                                               IHZD0072
      REAL*8 EPS,P,G,Q,C,S,R,F,TEST                                      IHZD0073
      REAL*8 DABS,DSIGN,DSQRT                                            IHZD0074
      LOGICAL MATZ,SKEW                                                  IHZD0075
C                                                                        IHZD0076
C                                                                        IHZD0077
C *** MACHINE DEPENDENT PARAMETER                                        IHZD0078
C                                                                        IHZD0079
      DATA  EPS  / Z3410000000000000 /                                   IHZD0080
C                                                                        IHZD0081
C                                                                        IHZD0082
      IF (.NOT. MATZ) GO TO 115                                          IHZD0083
C                                                                        IHZD0084
C *** PLACE IDENTITY MATRIX IN Z                                         IHZD0085
C                                                                        IHZD0086
      DO 110 I = 1, N                                                    IHZD0087
         DO 100 J = 1, N                                                 IHZD0088
            Z(I,J) = 0.D0                                                IHZD0089
  100    CONTINUE                                                        IHZD0090
         Z(I,I) = 1.D0                                                   IHZD0091
  110 CONTINUE                                                           IHZD0092
C                                                                        IHZD0093
  115 IERR = 0                                                           IHZD0094
      K = N                                                              IHZD0095
      NM1 = N - 1                                                        IHZD0096
      Z(1) = 0.D0                                                        IHZD0097
      ITS = 0                                                            IHZD0098
C                                                                        IHZD0099
  120 IF (K .LT. 2) GO TO 370                                            IHZD0100
      M0 = K                                                             IHZD0101
C                                                                        IHZD0102
C *** SEARCH FOR NEXT SUBMATRIX TO SOLVE  (MATRIX SPLITTING)             IHZD0103
C                                                                        IHZD0104
      P = 0.D0                                                           IHZD0105
      DO 130 I = 1, NM1                                                  IHZD0106
         J = K - I                                                       IHZD0107
         JP1 = J + 1                                                     IHZD0108
         G = DABS(Z(JP1))                                                IHZD0109
         IF (G .LE. EPS*(DABS(Z(J))+P)) GO TO 140                        IHZD0110
         P = G                                                           IHZD0111
  130 CONTINUE                                                           IHZD0112
      JP1 = 1                                                            IHZD0113
```

```
C                                                                       IHZD0114
  140 L0 = JP1 + 1                                                      IHZD0115
      L0M1 = JP1                                                        IHZD0116
      IF (L0M1 .EQ. N) GO TO 290                                        IHZD0117
      IF (.NOT. MATZ) GO TO 160                                         IHZD0118
      IF (.NOT. SKEW) GO TO 160                                         IHZD0119
C                                                                       IHZD0120
C *** PLACE CORRECT SIGN OF IDENTITY DIAGONALS                          IHZD0121
C                                                                       IHZD0122
      DO 150 I = L0M1, N, 4                                             IHZD0123
         Z(I,I) = -Z(I,I)                                               IHZD0124
         IP3 = I + 3                                                    IHZD0125
         IF (IP3 .GT. N) GO TO 160                                      IHZD0126
         Z(IP3,IP3) = -Z(IP3,IP3)                                       IHZD0127
  150 CONTINUE                                                          IHZD0128
C                                                                       IHZD0129
  160 IF (L0 .EQ. N) GO TO 300                                          IHZD0130
      IBO = N - L0                                                      IHZD0131
      IBO = MOD(IBO,2)                                                  IHZD0132
      L = L0                                                            IHZD0133
      IF (IBO .EQ. 0) GO TO 230                                         IHZD0134
C                                                                       IHZD0135
C *** FIND ZERO EIGENVALUE OF ODD ORDERED SUBMATRICES                   IHZD0136
C                                                                       IHZD0137
      C = 0.D0                                                          IHZD0138
      S = -1.D0                                                         IHZD0139
      DO 190 I = L0, NM1, 2                                             IHZD0140
         K = NM1 + L0 - I                                               IHZD0141
         KP1 = K + 1                                                    IHZD0142
         Q = -S * Z(KP1)                                                IHZD0143
         Z(KP1) = C * Z(KP1)                                            IHZD0144
         IF (DABS(Z(K)) .GT. DABS(Q)) GO TO 170                         IHZD0145
         C = Z(K) / Q                                                   IHZD0146
         R = DSQRT(C*C + 1.D0)                                          IHZD0147
         Z(K) = Q * R                                                   IHZD0148
         S = 1.D0 / R                                                   IHZD0149
         C = C * S                                                      IHZD0150
         GO TO 180                                                      IHZD0151
  170    S = Q / Z(K)                                                   IHZD0152
         R = DSQRT(1.D0 + S*S)                                          IHZD0153
         Z(K) = Z(K) * R                                                IHZD0154
         C = 1.D0 / R                                                   IHZD0155
         S = S * C                                                      IHZD0156
  180    IF (.NOT. MATZ) GO TO 190                                      IHZD0157
C                                                                       IHZD0158
C *** ACCUMULATE TRANSFORMATIONS FOR EIGENVECTORS                       IHZD0159
C                                                                       IHZD0160
         KN1 = K - 1                                                    IHZD0161
         Z(KN1,N) = -S * Z(KN1,KN1)                                     IHZD0162
         Z(KN1,KN1) = C * Z(KN1,KN1)                                    IHZD0163
         DO 185 J = KP1, N, 2                                           IHZD0164
            Z(J,KN1) = S * Z(J,N)                                       IHZD0165
            Z(J,N) = C * Z(J,N)                                         IHZD0166
  185    CONTINUE                                                       IHZD0167
C                                                                       IHZD0168
  190 CONTINUE                                                          IHZD0169
      N = NM1                                                           IHZD0170
      NM1 = N - 1                                                       IHZD0171
```

```
      IF (L0 .EQ. N) GO TO 300                                          IHZD0172
C                                                                       IHZD0173
C *** CHECK FOR CONVERGENCE OR SMALL SUBDIAGONAL ELEMENT                IHZD0174
C                                                                       IHZD0175
  200 DO 210 I = L0, NM1, 2                                             IHZD0176
         K = NM1 + L0 - I                                               IHZD0177
         L = K + 1                                                      IHZD0178
         TEST = EPS * (DABS(E(L)) + DABS(E(K-1)))                       IHZD0179
         IF (DABS(E(K)) .LE. TEST) GO TO 220                            IHZD0180
  210 CONTINUE                                                          IHZD0181
      L = L0                                                            IHZD0182
  220 IF (L .EQ. N) GO TO 300                                           IHZD0183
C                                                                       IHZD0184
C *** FORM SHIFT                                                        IHZD0185
C                                                                       IHZD0186
  230 ITS = ITS + 1                                                     IHZD0187
      IF (ITS .GT. 30) GO TO 360                                        IHZD0188
      P = E(N-3)                                                        IHZD0189
      G = E(N-2)                                                        IHZD0190
      C = E(NM1)                                                        IHZD0191
      S = E(N)                                                          IHZD0192
      P = ((C-P) * (C+P) + (S-G) * (S+G)) / (2.D0 * G * C)              IHZD0193
      R = DSQRT(P*P + 1.D0)                                             IHZD0194
      Q = (G / (P + DSIGN(R,P))) - C                                    IHZD0195
      P = E(L)                                                          IHZD0196
      LM1 = L - 1                                                       IHZD0197
      E(LM1) = ((P-S) * (P+S) + C * Q) / P                              IHZD0198
C                                                                       IHZD0199
C *** PERFORM ONE IMPLICIT QR ITERATION ON CHOLESKY FACTOR              IHZD0200
C                                                                       IHZD0201
      LS = L0M1                                                         IHZD0202
      C = 1.D0                                                          IHZD0203
      S = 1.D0                                                          IHZD0204
      DO 280 I = L, NM1                                                 IHZD0205
         IP1 = I + 1                                                    IHZD0206
         IM1 = I - 1                                                    IHZD0207
         Q = S * E(IP1)                                                 IHZD0208
         E(IP1) = C * E(IP1)                                            IHZD0209
         IF (DABS(E(IM1)) .GT. DABS(Q)) GO TO 240                       IHZD0210
         C = E(IM1) / Q                                                 IHZD0211
         R = DSQRT(C*C + 1.D0)                                          IHZD0212
         E(IM1) = Q * R                                                 IHZD0213
         S = 1.D0 / R                                                   IHZD0214
         C = C * S                                                      IHZD0215
         GO TO 250                                                      IHZD0216
  240    S = Q / E(IM1)                                                 IHZD0217
         R = DSQRT(1.D0 + S*S)                                          IHZD0218
         E(IM1) = E(IM1) * R                                            IHZD0219
         C = 1.D0 / R                                                   IHZD0220
         S = S * C                                                      IHZD0221
  250    P = E(IP1)                                                     IHZD0222
         E(IP1) = -S * E(I) + C * P                                     IHZD0223
         E(I) = C * E(I) + S * P                                        IHZD0224
         IF (.NOT. MATZ) GO TO 280                                      IHZD0225
C                                                                       IHZD0226
C *** ACCUMULATE TRANSFORMATIONS FOR EIGENVECTORS                       IHZD0227
C                                                                       IHZD0228
         DO 260 J = LS, N0, 2                                           IHZD0229
```

```
                P = Z(J,IP1)                                                    IMZD0230
                Z(J,IP1) = -S * Z(J,IN1) + C * P                                IMZD0231
                Z(J,IN1) = C * Z(J,IN1) + S * P                                 IMZD0232
  260        CONTINUE                                                           IMZD0233
             IF (LS .EQ. LOW1) GO TO 270                                        IMZD0234
             LS = LOW1                                                          IMZD0235
             GO TO 280                                                          IMZD0236
  270        LS = L0                                                            IMZD0237
  280     CONTINUE                                                              IMZD0238
          E(LS1) = 0.D0                                                         IMZD0239
          GO TO 200                                                             IMZD0240
C                                                                               IMZD0241
C     *** ITERATION CONVERGED TO ONE ZERO EIGENVALUE                            IMZD0242
C                                                                               IMZD0243
  290     E(M) = 0.D0                                                           IMZD0244
          M = MM1                                                               IMZD0245
          GO TO 310                                                             IMZD0246
C                                                                               IMZD0247
C     *** ITERATION CONVERGED TO EIGENVALUE PAIR                                IMZD0248
C                                                                               IMZD0249
  300     E(MP1) = E(M)                                                         IMZD0250
          E(M) = -E(M)                                                          IMZD0251
          M = M - 2                                                             IMZD0252
C                                                                               IMZD0253
  310     ITS = 0                                                               IMZD0254
          MM1 = M - 1                                                           IMZD0255
          IF (M .GT. L0) GO TO 200                                              IMZD0256
          IF (M .EQ. L0) GO TO 300                                              IMZD0257
          IF (.NOT. MATZ) GO TO 120                                             IMZD0258
          IF (SKEW) GO TO 120                                                   IMZD0259
C                                                                               IMZD0260
C     *** COMPUTE EIGENVECTORS FROM ORTHONORMAL COLUMNS OF Z IF NOT SKEW        IMZD0261
C                                                                               IMZD0262
  320     K = N0                                                                IMZD0263
  330     IF (Z(K) .EQ. 0.D0) GO TO 350                                         IMZD0264
          KM1 = K - 1                                                           IMZD0265
          DO 340 J = LOW1, N0, 2                                                IMZD0266
             Z(J,K) = Z(J,KM1)                                                  IMZD0267
             P = Z(J+1,K)                                                       IMZD0268
             Z(J+1,KM1) = P                                                     IMZD0269
             Z(J+1,K) = -P                                                      IMZD0270
  340     CONTINUE                                                              IMZD0271
          K = KM1                                                               IMZD0272
  350     K = K-1                                                               IMZD0273
          IF (K .GT. LOW1) GO TO 330                                            IMZD0274
          IF (IERR .NE. 0) GO TO 370                                            IMZD0275
          GO TO 120                                                             IMZD0276
C                                                                               IMZD0277
C     *** ERROR EXIT                                                            IMZD0278
C                                                                               IMZD0279
  360     IERR = N                                                              IMZD0280
          IF (.NOT. MATZ) GO TO 370                                             IMZD0281
          IF (.NOT. SKEW) GO TO 320                                             IMZD0282
C                                                                               IMZD0283
  370     RETURN                                                                IMZD0284
          END                                                                   IMZD0285
```

```
      SUBROUTINE TRIZD ( NAX, N, A, E, E2 )                              TRIZ0001
C                                                                        TRIZ0002
C ****                                                                   TRIZ0003
C                                                                        TRIZ0004
C  FUNCTION  -  REDUCES A REAL SKEW-SYMMETRIC MATRIX TO A SKEW-SYMMETRICTRIZ0005
C                 TRIDIAGONAL MATRIX USING ORTHOGONAL SIMILARITY         TRIZ0006
C                 TRANSFORMATIONS                                        TRIZ0007
C                                                                        TRIZ0008
C  PARAMETERS                                                            TRIZ0009
C                                                                        TRIZ0010
C     NAX       - INPUT INTEGER SPECIFYING THE ROW DIMENSION OF A AS     TRIZ0011
C                   DECLARED IN THE CALLING PROGRAM DIMENSION STATEMENT  TRIZ0012
C                                                                        TRIZ0013
C     N         - INPUT INTEGER SPECIFYING THE ORDER OF A                TRIZ0014
C                                                                        TRIZ0015
C     A(NAX,N)  - ON INPUT, A CONTAINS THE REAL SKEW-SYMMETRIC MATRIX.   TRIZ0016
C                   ONLY THE STRICT LOWER TRIANGLE OF THE MATRIX NEED    TRIZ0017
C                   BE SUPPLIED.                                         TRIZ0018
C                 ON OUTPUT, A CONTAINS INFORMATION ABOUT THE ORTHOGONAL TRIZ0019
C                   TRANSFORMATIONS USED IN THE REDUCTION IN ITS FULL    TRIZ0020
C                   LOWER TRIANGLE. THE STRICT UPPER TRIANGLE OF A IS    TRIZ0021
C                   UNALTERED.                                           TRIZ0022
C                                                                        TRIZ0023
C     E(N)      - OUTPUT ARRAY CONTAINING THE LOWER SUBDIAGONAL ELEMENTS TRIZ0024
C                   OF THE TRIDIAGONAL MATRIX IN ITS LAST N-1 POSITIONS. TRIZ0025
C                   E(1) IS SET TO ZERO.                                 TRIZ0026
C                                                                        TRIZ0027
C     E2(N)     - OUTPUT ARRAY CONTAINING THE SQUARES OF THE ELEMENTS    TRIZ0028
C                   OF E.  E2 MAY COINCIDE WITH E IF THE SQUARES ARE NOT TRIZ0029
C                   NEEDED.                                              TRIZ0030
C                                                                        TRIZ0031
C  REQUIRED FUNCTIONS - DABS,DSIGN,DSQRT                                 TRIZ0032
C                                                                        TRIZ0033
C  REFERENCE  -  L.J. GRAY AND R.C. WARD, EIGENSYSTEM COMPUTATION FOR    TRIZ0034
C                   SKEW-SYMMETRIC MATRICES AND A CLASS OF SYMMETRIC     TRIZ0035
C                   MATRICES,  UCCND-CSD REPORT, 1976.                   TRIZ0036
C                                                                        TRIZ0037
C ****                                                                   TRIZ0038
C                                                                        TRIZ0039
      REAL*8 A(NAX,N),E(N),E2(N)                                         TRIZ0040
      REAL*8 F,G,H,SCALE                                                 TRIZ0041
      REAL*8 DABS,DSIGN,DSQRT                                            TRIZ0042
C                                                                        TRIZ0043
      IF (N .EQ. 1) GO TO 230                                            TRIZ0044
C                                                                        TRIZ0045
C *** MAIN DO LOOP  I=N STEP -1 UNTIL 2                                  TRIZ0046
C                                                                        TRIZ0047
      DO 220 II = 2, N                                                   TRIZ0048
         I = N + 2 - II                                                  TRIZ0049
         L = I - 1                                                       TRIZ0050
         H = 0.D0                                                        TRIZ0051
         SCALE = 0.D0                                                    TRIZ0052
C                                                                        TRIZ0053
C *** NORMALIZE ROW                                                      TRIZ0054
C                                                                        TRIZ0055
```

```
      DO 100 K = 1, L                                                    TRIZ0056
         SCALE = SCALE + DABS(A(I,K))                                    TRIZ0057
  100 CONTINUE                                                           TRIZ0058
C                                                                        TRIZ0059
      IF (SCALE .NE. 0.D0) GO TO 120                                     TRIZ0060
  110 E(I) = 0.D0                                                        TRIZ0061
      E2(I) = 0.D0                                                       TRIZ0062
      GO TO 215                                                          TRIZ0063
C                                                                        TRIZ0064
C *** COMPUTE ELEMENTS OF U VECTOR                                       TRIZ0065
C                                                                        TRIZ0066
  120 DO 130 K = 1, L                                                    TRIZ0067
         A(I,K) = A(I,K) / SCALE                                         TRIZ0068
         H = H + A(I,K) * A(I,K)                                         TRIZ0069
  130 CONTINUE                                                           TRIZ0070
C                                                                        TRIZ0071
      E2(I) = SCALE * SCALE * H                                          TRIZ0072
      F = A(I,L)                                                         TRIZ0073
      G = -DSIGN(DSQRT(H),F)                                             TRIZ0074
      E(I) = SCALE * G                                                   TRIZ0075
      H = H - F * G                                                      TRIZ0076
      A(I,L) = F - G                                                     TRIZ0077
      IF (L .EQ. 1) GO TO 200                                            TRIZ0078
C                                                                        TRIZ0079
C *** COMPUTE ELEMENTS OF A*U/H                                          TRIZ0080
C                                                                        TRIZ0081
      DO 180 J = 1, L                                                    TRIZ0082
         G = 0.D0                                                        TRIZ0083
         IF (J .EQ. 1) GO TO 150                                         TRIZ0084
         JM1 = J - 1                                                     TRIZ0085
C                                                                        TRIZ0086
         DO 140 K = 1, JM1                                               TRIZ0087
            G = G + A(J,K) * A(I,K)                                      TRIZ0088
  140    CONTINUE                                                        TRIZ0089
C                                                                        TRIZ0090
  150    IF (J .EQ. L) GO TO 170                                         TRIZ0091
         JP1 = J + 1                                                     TRIZ0092
C                                                                        TRIZ0093
         DO 160 K = JP1, L                                               TRIZ0094
            G = G + A(K,J) * A(I,K)                                      TRIZ0095
  160    CONTINUE                                                        TRIZ0096
C                                                                        TRIZ0097
  170    E(J) = G / H                                                    TRIZ0098
  180 CONTINUE                                                           TRIZ0099
C                                                                        TRIZ0100
C *** COMPUTE REDUCED A                                                  TRIZ0101
C                                                                        TRIZ0102
      DO 190 J = 2, L                                                    TRIZ0103
         F = A(I,J)                                                      TRIZ0104
         G = E(J)                                                        TRIZ0105
         JM1 = J - 1                                                     TRIZ0106
C                                                                        TRIZ0107
         DO 190 K = 1, JM1                                               TRIZ0108
            A(J,K) = A(J,K) + F * E(K) - G * A(I,K)                      TRIZ0109
  190 CONTINUE                                                           TRIZ0110
C                                                                        TRIZ0111
  200 DO 210 K = 1, L                                                    TRIZ0112
         A(I,K) = SCALE * A(I,K)                                         TRIZ0113
```

```
  210     CONTINUE                                                      TRIZ0114
C                                                                       TRIZ0115
  215     A(I,I) = SCALE * DSQRT(H)                                     TRIZ0116
  220 CONTINUE                                                          TRIZ0117
C                                                                       TRIZ0118
  230 E(1) = 0.D0                                                       TRIZ0119
      E2(1) = 0.D0                                                      TRIZ0120
      RETURN                                                            TRIZ0121
      END                                                               TRIZ0122
```

```
      SUBROUTINE TBAK2D ( MAX, N, A, M, Z )                              TBAK0001
C                                                                        TBAK0002
C ****                                                                   TBAK0003
C                                                                        TBAK0004
C  FUNCTION  -  FORMS THE EIGENVECTORS OF A REAL SKEW-SYMMETRIC MATRIX   TBAK0005
C                 BY BACK TRANSFORMING THOSE OF THE CORRESPONDING SKEW-  TBAK0006
C                 SYMMETRIC TRIDIAGONAL MATRIX DETERMINED BY  TRIZD     TBAK0007
C                                                                        TBAK0008
C  PARAMETERS                                                            TBAK0009
C                                                                        TBAK0010
C     MAX       - INPUT INTEGER SPECIFYING THE ROW DIMENSION OF A AND Z  TBAK0011
C                 AS DECLARED IN CALLING PROGRAM DIMENSION STATEMENT     TBAK0012
C                                                                        TBAK0013
C     N         - INPUT INTEGER SPECIFYING THE ORDER OF A                TBAK0014
C                                                                        TBAK0015
C     A(MAX,N)  - INPUT ARRAY CONTAINING INFORMATION ABOUT THE ORTHOGONALTBAK0016
C                 TRANSFORMATIONS USED IN THE REDUCTION BY  TRIZD  IN    TBAK0017
C                 ITS FULL LOWER TRIANGLE                                TBAK0018
C                                                                        TBAK0019
C     M         - INPUT INTEGER SPECIFYING THE NUMBER OF EIGENVECTORS TO TBAK0020
C                 BE BACK TRANSFORMED                                    TBAK0021
C                                                                        TBAK0022
C     Z(MAX,M)  - ON INPUT, Z CONTAINS THE REAL AND IMAGINARY (IF        TBAK0023
C                 COMPLEX) PARTS OF THE EIGENVECTORS TO BE BACK TRANS-   TBAK0024
C                 FORMED IN ITS FIRST M COLUMNS                          TBAK0025
C                ON OUTPUT, Z CONTAINS THE REAL AND IMAGINARY (IF        TBAK0026
C                 COMPLEX) PARTS OF THE TRANSFORMED EIGENVECTORS IN      TBAK0027
C                 ITS FIRST M COLUMNS                                    TBAK0028
C                                                                        TBAK0029
C  REFERENCE  -  L.J. GRAY AND R.C. WARD, EIGENSYSTEM COMPUTATION FOR     TBAK0030
C                 SKEW-SYMMETRIC MATRICES AND A CLASS OF SYMMETRIC       TBAK0031
C                 MATRICES,  UCCND-CSD REPORT, 1976.                     TBAK0032
C                                                                        TBAK0033
C ****                                                                   TBAK0034
C                                                                        TBAK0035
      REAL*8 A(MAX,N),Z(MAX,M)                                           TBAK0036
      REAL*8 H,S                                                         TBAK0037
C                                                                        TBAK0038
      IF (M .EQ. 0) GO TO 140                                            TBAK0039
      IF (N .EQ. 1) GO TO 140                                            TBAK0040
C                                                                        TBAK0041
      DO 130 I = 2, N                                                    TBAK0042
         L = I - 1                                                       TBAK0043
         H = A(I,I)                                                      TBAK0044
         IF (H .EQ. 0.D0) GO TO 130                                      TBAK0045
C                                                                        TBAK0046
         DO 120 J = 1, M                                                 TBAK0047
            S = 0.D0                                                     TBAK0048
C                                                                        TBAK0049
            DO 100 K = 1, L                                              TBAK0050
               S = S + A(I,K) * Z(K,J)                                   TBAK0051
  100       CONTINUE                                                     TBAK0052
C                                                                        TBAK0053
            S = (S / H) / H                                              TBAK0054
C                                                                        TBAK0055
```

```
            DO 110 K = 1, L                                          TBAK0056
               Z(K,J) = Z(K,J) - S * A(I,K)                          TBAK0057
  110       CONTINUE                                                 TBAK0058
C                                                                    TBAK0059
  120    CONTINUE                                                    TBAK0060
C                                                                    TBAK0061
  130 CONTINUE                                                       TBAK0062
C                                                                    TBAK0063
  140 RETURN                                                         TBAK0064
      END                                                            TBAK0065
```