Mathematics Notes

Note 18

March 1970

SEVER: A Routine for Computing Symmetric Matrix Eigenvalues and Eigenvectors with Error Bounds

W. R. Gavin, 9422
Sandia Laboratories, Albuquerque

Abstract

Subroutine SEVER computes a complete eigensystem with error bounds for a real symmetric matrix. Householder's method is used to reduce the matrix to tridiagonal form. The eigenvalues and eigenvectorsof the tridiagonal matrix are computed by the QR algorithm. The computed eigenvalues are subjected to one Rayleigh quotient correction using the computed eigenvectors. Upper bounds on the absolute errors of the computed eigensystem are calculated.

Key word: mathematics

MoN 18

SC-M-70-26

SEVER: A ROUTINE FOR COMPUTING SYMMETRIC MATRIX EIGENVALUES AND EIGENVECTORS WITH ERROR BOUNDS

W. R. Gavin, 9422
Sandia Laboratories, Albuquerque

March 1970

ABSTRACT

Subroutine SEVER computes a complete eigensystem with error bounds for a real symmetric matrix. Householder's method is used to reduce the matrix to tridiagonal form. The eigenvalues and eigenvectors of the tridiagonal matrix are computed by the QR algorithm. The computed eigenvalues are subjected to one Rayleigh quotient correction using the computed eigenvectors. Upper bounds on the absolute errors of the computed eigensystem are calculated.

Key word: mathematics

FOREWORD

The Sandia Laboratories Mathematical Program Library consists of a number of dependable, high-quality, general-purpose, mathematical computing routines. The standards established for the library require that these routines be mathematically sound, effectively implemented, extensively tested, and thoroughly documented. This report documents one such routine.

The library emphasizes the effective coverage of various distinct mathematical areas with a minimum number of routines. Nevertheless, it may contain other routines similar in nature but complementary to the one described here. Additional information on the mathematical program library, a description of the standard format for documenting these routines, and a guide to other routines in the library are contained in SC-M-69-337.

This report is also identified within Sandia Laboratories as Computing Publication ML0005/ALL. This report and its corresponding library routine are expected to be available from COSMIC shortly after publication.

CONTENTS

CONTENTS (Cont)

SEVER: A ROUTINE FOR COMPUTING SYMMETRIC MATRIX EIGENVALUES AND EIGENVECTORS WITH ERROR BOUNDS

1. Introduction

1.1 Background

SEVER was written to provide the Sandia Mathematical Program Library with a quality routine for computing an eigensystem with error bounds of a real symmetric matrix. The methods used in SEVER were chosen and the routine was constructed to avoid some of the more common deficiencies in existing routines for symmetric matrix eigensystem calculation. Among such deficiencies are the lack of precision in the calculation of clustered or multiple eigenvalues, inability to handle eigenvalues which differ only in sign, nonorthogonality of eigenvectors corresponding to eigenvalues of multiplicity greater than one, and lack of any measure of the precision or reliability of the results.

The ALGOL routine of Martin et al[8] was converted to FORTRAN and constitutes that segment of SEVER which reduces the input matrix to tridiagonal form.

The ALGOL routine of Bowdler et al[3] was converted to FORTRAN and is a subroutine called by SEVER to compute the eigenvalues and eigenvectors of the tridiagonal matrix.

1.2 Applicable Programming Languages and Computer Systems

SEVER is written in a subset of Control Data 6600 FORTRAN. Except for two DATA statements containing error messages and two replacement statements involving octal literals, it is believed that the program conforms to American National Standard FORTRAN. However, complete portability of the routine is not implied and, because of machine-dependent

constants (e.g., the smallest positive floating point number which when added to 1.0 yields a floating point sum not 1.0), modified versions of SEVER may be required for machines other than the CDC 6600.

SEVER has executed successfully and has been tested thoroughly under the Control Data 6600 SCOPE operating system with the FUN and RUN compilers.

This routine is maintained for the convenience of the user on a library file on the CDC 6600 and is accessible by means of one control card described in Appendix C.

1.3 Considerations Regarding Use

SEVER computes all of the eigenvalues of a real symmetric matrix and a set of orthonormal eigenvectors. An upper bound on the absolute error of each computed eigenvalue is returned as well as an upper bound on the absolute error of the computed eigenvector. Lengths of residual vectors are also returned. Double precision eigenvalues are computed and returned but these values should be used in conjunction with the eigenvalue absolute error bounds. Both the single and double precision eigenvalues are returned in nonincreasing algebraic order.

SEVER may be used to compute an eigensystem for a complex Hermitian matrix. See Section 7 for details on how this may be done.

The nature of arithmetic and format conversion may cause zero eigenvalues and zero components of eigenvectors to look as if they were nonzero numbers which are very small in absolute value. In these cases the user must determine whether a small eigenvalue or a small eigenvector component is actually zero (see Section 2.7).

2. Usage

2.1 Entry

The calling sequence is:

```
CALL SEVER(A,EV,VEC,ERVAL,DPA,N,NDIM,KER)
```

2.2 Description of Arguments

In the following description an argument which must be preset by the calling program before a call to SEVER is referred to as an input argument. An argument which will return with a value determined in SEVER is referred to as an output argument. SEVER has no arguments which are both input and output. An example of a calling sequence is given in Section 2.4.

A — A real input array which must be dimensioned with exactly NDIM rows and at least NDIM columns. Normally, A should be dimensioned (NDIM, NDIM). The leading N by N subarray of A must contain the real symmetric matrix whose eigensystem is to be computed. A is returned unchanged.

EV — A real output array which, upon return from SEVER, contains the computed eigenvalues in nonincreasing algebraic order. EV must be dimensioned at least N in the calling program, and normally would be dimensioned NDIM.

VEC — The real output array which, upon return from SEVER, contains the computed eigenvectors. The eigenvectors are stored in the columns of VEC corresponding to the eigenvalues in EV. That is, the k-th column of VEC contains a computed eigenvector of length one corresponding to the computed eigenvalue in the k-th element of array EV. VEC must be dimensioned with exactly NDIM rows and at least NDIM columns. Normally, VEC would be dimensioned (NDIM, NDIM).

ERVAL The real output array containing upper bounds on the absolute errors of the computed double precision eigenvalues (see Section 2.7). In case it is not possible to bound the error on some eigenvalue, a -1.0 is returned in the element of ERVAL corresponding to that eigenvalue. ERVAL must be dimensioned at least N and normally would be dimensioned NDIM in the calling program.

DPA A double precision output array which must be dimensioned (NDIM, 3) and declared type double in the calling program. DPA must be supplied by the user in order that certain inner products in the Rayleigh quotient eigenvalue correction and error calculations may be accumulated in double precision. Upon return, the first column of DPA contains the Rayleigh quotient eigenvalues in double precision. The second column contains an upper bound on the absolute error (= relative error) of the computed eigenvector as measured in the two norm. If such a bound cannot be computed, i.e., ERVAL(K) = -1, the length of the k-th residual vector (Section 4) is returned. The third column of DPA contains the length of the residual vector in every case.

N The integer order of the matrix to be resolved. N must not be smaller than 3. (The distinction between N and NDIM facilitates the solution of different size problems in successive calls in the same execution of the calling program.) N is an input argument.

NDIM The integer number of rows and usually the number of columns allocated to A in the calling program. NDIM must not be smaller than N. NDIM is an input argument.

KER An output error indicator. Its value upon return is 1 for normal execution, 2 if the dimensions and size of the matrix are inconsistent, and 3 if more than 30 QR iterations are required for some eigenvalue.

2.3 Restrictions Between Arguments

A and VEC	Treated as square arrays, and each must be dimensioned with exactly NDIM rows and at least N columns. Normally each of these arrays should be dimensioned (NDIM, NDIM).
EV and ERVAL	Arrays, each of which must be dimensioned at least N. Normally each array should be dimensioned NDIM.
A, EV, ERVAL, VEC, and DPA	Must be distinct arrays. If KER is returned as 2, no values are assigned to EV, VEC, ERVAL or DPA. If KER is returned as 3, the values assigned to EV, VEC, ERVAL, and DPA may not be meaningful.
N and NDIM	Must satisfy $3 \leq N \leq NDIM$.

2.4 Principal Uses with Example

The following is a segment of a possible calling program for computing the eigensystem of a 4 x 4 real symmetric matrix A for which up to 25 rows and columns are allocated:

```
      PROGRAM EIGVAL(INPUT,OUTPUT)
      DIMENSION A(25,25),EV(25),ERVAL(25),VEC(25,25)
      DOUBLE PRECISION DPA(25,3)
      DO 1  I = 1,4
    1 READ 101, (A(I,J),J=1,4)
  101 FORMAT(4E10.3)
      CALL SEVER(A,EV,VEC,ERVAL,DPA,4,25,KER)
        .
        .
        .
       Print results of calculation
        .
        .
        .
      END
```

Note that N and NDIM are the only arguments which may appear as constants in the calling statements.

2.5 Library Routines Explicitly Required

SEVER requires the standard routines SQRT, DSQRT, DBLE, IABS, and ABS, and the Sandia Mathematical Program Library routine ERRCHK.[1]

SEVER also references QRVEC which is a subordinate routine contained in the Sandia library. QRVEC is used only by SEVER and is conceptually a part of SEVER.

2.6 User-Supplied Routines Required

SEVER requires no user-supplied routines.

2.7 Cautions and Restrictions

Even though a symmetric matrix is determined by either its upper or lower triangle, the whole matrix must be supplied to the routine. No symmetry check is made. If a nonsymmetric matrix is supplied to the routine, the results will be spurious.

It is extremely likely that zero eigenvalues and zero components of eigenvectors will be returned as very small nonzero numbers. If the absolute value of a computed eigenvalue is smaller than the corresponding value in ERVAL a necessary condition for a zero has been satisfied. The true value of such an eigenvalue may indeed be zero but it is not possible for the routine to make such a determination.

A value of -1.0 for the computed error of an eigenvalue does not necessarily mean that that eigenvalue and its eigenvector are of poor quality. It merely means that it was not possible to compute an upper bound on the absolute error of the computed eigenvalue and computed eigenvector. (See Section 4 for full details on the error calculation.) Furthermore, if some eigenvalue error is returned as -1.0, so must at least one of its adjacent neighbors in the ERVAL array. The error calculation cannot fail for a single eigenvalue.

The computed absolute error bound applies to the computed double precision eigenvalue. The computed single precision eigenvalue may have

a truncation error induced in its last bit. In particular, on the CDC 6600 the single precision eigenvalue may have a truncation-induced error of $2^{\beta-48}$ where the computed eigenvalue is represented by $\alpha 2^{\beta}$ with $1/2 \leq \alpha < 1$.

The second and third columns of the array DPA, which contain vector errors and residual vector lengths, respectively, are not meaningful to full double precision. In fact, only the first significant digit and exponent are usually of any interest.

The routine makes no checks for fault conditions such as underflow or overflow and, if these fault conditions occur, the results may be invalidated.

The routine will not handle a matrix smaller than 3 x 3.

2.8 Error Conditions, Messages, and Codes

The subroutine SEVER contains two error messages. If the input variables N and NDIM do not satisfy $3 \leq N \leq NDIM$, the message

INPUT DIMENSIONS ARE INCONSISTENT

is printed and KER is returned as 2.

If more than thirty QR iterations are required for some eigenvalue, the message

NO CONVERGENCE IN 30 QR ITERATIONS

is printed and KER is returned as 3.

The error messages are processed by the Sandia Mathematical Program Library routine ERRCHK; the occurrence of either error message results in a fatal error unless the user has previously made an appropriate call to ERRSET.[1] To make error conditions nonfatal, ERRSET must be called before SEVER is called. For example, to make the errors

nonfatal and set a maximum of 10 messages to be printed the following call must be made:

CALL ERRSET(10,0).

If ERRSET has been called in order to make the error conditions nonfatal and the input dimensions are inconsistent, then KER is set to 2, a return to the calling program is made, and no computations are done. If ERRSET has been called and more than 30 QR iterations are required for some eigenvalue, then KER is set to 3 and a return is made to the calling program. In this case the results may be spurious.

3. Mathematical Method

3.1 Statement of the Problem

Let A, the machine representation of a real symmetric matrix, have eigenvalues $\{\lambda_i\}$ and orthonormal eigenvectors $\{x_i\}$ so that $Ax_i = \lambda_i x_i$. The problem then is to compute scalars $\{\overline{\lambda}_i\}$, vectors $\{\overline{x}_i\}$, with error bounds $\{\varepsilon_i\}$ and $\{\eta_i\}$ so that $|\lambda_i - \overline{\lambda}_i| \leq \varepsilon_i$ and $\|x_i - \overline{x}_i\| \leq \eta_i$. Hopefully, $|\lambda_i - \overline{\lambda}_i|$ and $\|x_i - \overline{x}_i\|$ are small and "close" to the ε_i and η_i. That is, it is desirable for the error bounds to be small and sharp.

3.2 Methods Used

The n x n matrix A is reduced to tridiagonal form by a sequence of n-2 orthogonal similarity transformations each of which makes one row and one column of the off tridiagonal zero. The eigenvalues and eigenvectors of the tridiagonal matrix are computed by a version of the QR algorithm especially modified for symmetric tridiagonal matrices.[3] The Rayleigh quotient corrected eigenvalue is calculated using the computed eigenvectors. Each error bound is computed as a function of the length of the residual vector and of the separation of the eigenvalue from its nearest neighbor.

3.3 Mathematical Range and Domain

SEVER computes an eigensystem for a real symmetric matrix. The eigenvalue range is the real line, and the eigenvector range is the unit n-sphere.

The precise range and domain of the subroutine would require a very difficult and pedantic description.

3.4 Equations and Discussion

The real symmetric matrix A is reduced to tridiagonal form by a sequence of orthogonal similarity transformations generated by matrices of the form $I-2uu^T$, where I is the identity matrix and u is a column vector of length one. That is, let $A_1 = A$ and $A_{i+1} = P_i^{-1}A_iP_i$ where $P_i = I - 2u_iu_i^T$. A_i is tridiagonal in its first i-1 rows and columns, and A_{n-1} is the desired tridiagonal form. The matrices P_i are called elementary Hermitians and the method of tridiagonalization is called Householder's method (Reference 11, p. 342).

Each similarity transformation can be conveniently carried out as follows:

$$\begin{aligned} A_2 &= (I - 2uu^T)A_1(I - 2uu^T) \\ &= A_1 - 2uu^TA_1 - 2A_1uu^T + 4(u^TA_1u)uu^T \\ &= A_1 - (uq^T + qu^T) \end{aligned} \qquad (1)$$

where $q = p - \alpha u$, $p = 2Au$ and $\alpha = u^Tp$.

Thus no matrix multiplications are required to perform each similarity transformation.

For the first transformation, the vector u is given by $u = \frac{a - c}{\|a - c\|}$ where a is the first column (or row) of A, $c = (a_{11}, \pm s, 0, \ldots, 0)^T$ and s^2 is the sum of the squares of the off diagonal elements of the

first row. Succeeding transformations are carried out by considering a matrix of one smaller order each time.

The QR algorithm for an arbitrary real matrix A consists of forming a sequence $\{A_k\}$ as follows: let $A_1 = A$ and for each k factor $A_k = Q_k R_k$ where Q_k is orthogonal and R_k is upper triangular and define $A_{k+1} = R_k Q_k$. Note that $A_{k+1} = R_k Q_k = Q_k^{-1} A_k Q_k$; thus A_{k+1} is orthogonally similar to A_k and, therefore, to A_1. The factorization $A_k = R_k Q_k$ can always be carried out,[8] and if A_k is nonsingular it is equivalent to a Gram-Schmidt orthogonalization of the linearly independent column of A_k.

The convergence of the QR algorithm for arbitrary matrices is somewhat too detailed for discussion here (see Reference 4, p. 269). For real symmetric matrices, however, convergence of the QR algorithm to a diagonal matrix is guaranteed, and for real symmetric tridiagonal matrices the implementation of this algorithm assumes a rather simple form (see Section 4).

The algorithm may be modified to increase its rate of convergence by redefining R_k at each stage by $R_k = Q_k^T(A_k - \alpha_k I)$ and choose $A_{k+1} = R_k Q_k$. In this case A_{k+1} is not similar to A_1 but is similar to $A_1 - \Sigma\alpha_k I$). The choice of α_k is discussed in Section 4.

Let the eigenvalues and eigenvectors computed by the QR algorithm be $\bar{\lambda}_1, \ldots, \bar{\lambda}_n$ and $\bar{x}_1, \ldots, \bar{x}_n$. Define the Rayleigh quotient $\rho_i = \dfrac{\bar{x}_i^T A \bar{x}_i}{\bar{x}_i^T \bar{x}_i}$ where A is the original matrix. Let ϵ_i be the length of the residual vector, $\epsilon_i = \|A\bar{x}_i - \rho_i \bar{x}_i\|$ and let a_i be the distance from the true eigenvalue λ_i to its nearest neighbor. Then the results which allow calculation of the error bounds of the computed eigensystem are

$$|\lambda_i - \rho_i| \leq \epsilon \ , \tag{2}$$

$$|\lambda_i - \rho_i| \leq \frac{\epsilon^2/a_i}{1 - \epsilon^2/a_i^2} \ , \text{ and} \tag{3}$$

$$\|x_i - \bar{x}_i\|^2 \leq \frac{\epsilon^2}{a_i^2}\left(1 + \frac{\epsilon^2}{a_i^2}\right), \tag{4}$$

where the vectors x_i and $\bar{x}_i$ have length one. For proofs of these results, see Reference 12. Note that these inequalities yield upper bounds on the absolute errors of the computed eigensystem. Furthermore, since $||x_i|| = 1$ the eigenvector inequality bounds both the relative and absolute errors.

3.5 Error Analysis, Bounds, and Estimates

The following three matrix norms are used:

$$||A||_2 = \max \{||Ax|| : ||x|| = 1\} \tag{5}$$

where $||x||$ is the ordinary length of x, or $||x||^2 = x^T x$. Note that, for symmetric matrices, $||A||_2$ is the largest eigenvalue in absolute value of A.

$$||A||_\infty = \max_i \sum_{j=1}^{n} |a_{ij}| \ , \tag{6}$$

the maximum absolute value row sum of A.

$$||A||_E^2 = \sum_{ij} |a_{ij}|^2 = \sum_i |\lambda_i|^2 \ , \tag{7}$$

where the λ_i are the eigenvalues of A.

It is true for all matrices that $||A||_2 \leq ||A||_E$ and if λ is any eigenvalue of A, then $|\lambda| \leq ||A||_t$ for t = 2, ∞, or E. If A is real symmetric then $||A||_2 \leq ||A||_\infty$. While $||A||_E$ and $||A||_\infty$ do not bound the eigenvalues as sharply as $||A||_2$, they have the advantage of ease of computation.

The significant results concerning a posteriori error analysis of real symmetric matrix eigenvalue calculations are as follows: Let A, B, and A+B be real symmetric matrices with eigenvalues $\{\lambda_i\}$, $\{\mu_i\}$, and $\{\nu_i\}$,

respectively, subscripted in nonincreasing order. Then

$$|\lambda_i - \nu_i| \leq \|B\|_2 = \max_i |\mu_i| \leq \|B\|_E \tag{8}$$

(Reference 9, p. 100) and

$$\Sigma(\lambda_i - \nu_i)^2 \leq \|B\|_E^2 = \Sigma|\mu_i|^2 \tag{9}$$

(Reference 11, p. 104). Thus, unlike the matrix inversion problem or the nonsymmetric eigenvalue problem, the eigenvalue problem for real symmetric matrices is not ill conditioned. In fact, it is Lipschitz continuous.

Suppose the real symmetric matrix A, whose elements are represented as decimal fractions, is presented to a digital computer algorithm for eigenvalue calculation. The algorithm computes not the eigenvalues of A but those of a slightly perturbed matrix A+D. The perturbation D arises from inexact machine representation of A and from errors induced in the diagonalization of A.

Equation 8 can be used to bound the errors arising from inexact machine representation of A. Suppose δ is the maximum relative error in the floating point representation of a real number. For most machines δ is bounded above by 1/2 raised to one less than the mantissa length in bits. The matrix A in the machine is A+Δ where the (i,j) element of Δ is δ_{ij} and $|\delta_{ij}| \leq |a_{ij}|\delta$. Thus, using Equation 8 and the fact that $\|\Delta\|_2 \leq \|\Delta\|_\infty$, the result is that $n \max_{i,j} |a_{ij}|$ is the largest possible error in any computed eigenvalue due to imprecise machine representation of the initial data.

While analytical expressions for upper bounds of the error induced in the diagonalization exist, they are cumbersome to compute and are usually quite pessimistic (Reference 11, p. 297). At any rate, these errors are part of the a posteriori errors described in Sections 3.4 and 4.

Unfortunately, the eigenvector problem for real symmetric matrices is not as well conditioned as in the eigenvalue problem. In fact, the

eigenvectors of a real symmetric matrix may not even depend continuously on the matrix elements. However, if an eigenvalue is sufficiently well separated from its nearest neighbor, then the following shows that its corresponding eigenvector calculation is well conditioned: Let A and A+B be real symmetric matrices with eigenvalues $\{\lambda_i\}$ and $\{\nu_i\}$ and corresponding eigenvectors $\{u_i\}$ and $\{v_i\}$ of length one. Let α_k be the distance from λ_k to its nearest neighbor. If $\alpha_k > \|B\|_2$ then

$$\|u_k - v_k\| \leq \gamma(1 + \gamma^2)^{1/2} \tag{10}$$

where $\gamma = \|B\|_2/(\alpha_k - \|B\|_2)$, see Reference 9.

To use this result, suppose B is the perturbation matrix resulting from inexact machine representation of the elements of A, $\|B\|_2 \leq n\delta$ and, if α is sufficiently larger than $\|B\|_2$, then $\|u_k - v_k\|$ is approximately $n\delta/\alpha$. A satisfactory qualitative statement of this result is that if the separation of an eigenvalue is significantly greater than the norm of the perterbation matrix then the corresponding eigenvector problem is well conditioned.

The eigenvector error induced by inexact diagonalization is accounted for in the a posteriori error bound of Equation 4.

Another source of error in the eigensystem calculation is inexact knowledge of the input matrix A. Upper bounds in the perturbations thus caused may be accounted for exactly as those caused by inexact machine representation of the matrices. For example, consider the realistic physical situation in which each element of a 200 x 200 matrix has an absolute error bounded by 5×10^{-3}, i.e., each element is correctly rounded to the nearest hundredth. In this case the norm of the error matrix $\|\Delta\|_\infty$ is bounded by $(200)(5)(10^{-3}) = 1$. Thus, the best possible bound on the absolute error of the computed eigenvalues is one. This rather pessimistic result is not dependent upon the method of eigenvalue calculation but is simply a property of the matrix. However, the absolute error bound can be reduced somewhat in the case of sparse or bandwidth-limited matrices.

4. Programming Method

The program divides naturally into four parts: the tridiagonalization, the calculation of eigenvalues and eigenvectors, the Rayleigh quotient correction, and the calculation of error bounds.

Before tridiagonalization the matrix A is copied into the array VEC, and the tridiagonalization and QR algorithm are carried out entirely within VEC. The tridiagonalization is carried out as described in Section 3.4. If, at the i-th stage, the sum of the squares of the off tridiagonal elements is sufficiently small no transformation is necessary. Indeed if this sum is small but not exactly zero then the corresponding transformation is in danger of not being orthogonal. Such a transformation is skipped if the sum of the squares of the off tridiagonal elements is less than the quotient of the smallest positive floating point number and the smallest positive floating point number which when added to 1.0 yields a sum not 1.0.

The off tridiagonal places of VEC are used to store the tridiagonalizing transformations for later use in calculating eigenvectors. In particular if $P^{-1}AP$ is tridiagonal and x is an eigenvector of $P^{-1}AP$ then Px is an eigenvector of A.

The tridiagonalization is carried out in single precision.

For convenience in programming, the QR algorithm is restated as a QL algorithm (i.e., the triangular factor is lower rather than upper). The algorithm is then stated:

$$Q_k(A_k - \alpha_k I) = L_k \ , \qquad (11)$$

$$A_{k+1} = L_k Q_k^T \ .$$

Note that the Q_k here are the transposes of the Q_k in Section 3.4.

Each Q_k is a product of Jacobi-like rotations P_i:

$$Q_k = P_1 P_2, \ldots, P_{n-1} \tag{12}$$

where each P_i is of the form

$$P_i = \begin{bmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & \ddots & & & & \\ & & & c_i & -s_i & & \\ & & & s_i & c_i & & \\ & & & & & \ddots & \\ & & & & & & 1 \\ & & & & & & & 1 \end{bmatrix} \tag{13}$$

and the c_i and s_i are the cosine and sine of an angle chosen so that

$$P_1 P_2, \ldots, P_{n-1} (A_k - \alpha_k I) = L_k \tag{14}$$

is lower triangular.

The algorithm then becomes:

$$\begin{aligned} &P_1 P_2, \ldots, P_{n-1} (A_k - \alpha_k I) = L_k , \\ &A_{k+1} = L_k P_{n-1}^T P_{n-2}^T, \ldots, P_1^T . \end{aligned} \tag{15}$$

The matrix L_k can have nonzero elements only on the main diagonal and first and second subdiagonals. Thus

$$L_k = \begin{bmatrix} r_1 & & & & \\ q_2 & r_2 & & & \\ t_3 & q_3 & r_3 & & \\ & t_4 & q_4 & r_4 & \\ & & \ddots & \ddots & \ddots \\ & & t_{n-1} & q_{n-1} & r_{n-1} \\ & & & t_n & q_n & r_n \end{bmatrix} . \qquad (16)$$

The product $L_k\ P_{n-1}^T\ P_{n-2}^T, \ldots, P_1^T$ reintroduces symmetry and has only one superdiagonal of nonzero elements. Hence, the second subdiagonal of the product is zero and there is no need to determine the t_i.

The lower triangular factor L_k and the (k+1)st iterate A_{k+1} can be determined simultaneously as follows (although a sea of algebra is necessary to verify it): Let $d_i^{(k)}$ $i = 1, \ldots, n$ be the diagonal elements of A_k and $e_i^{(k)}$ $i = 1, \ldots, n-1$ be the subdiagonal elements. Then proceeding from n to 1 we have:

$$\begin{aligned} p_n &= d_n^{(k)} - \alpha_k,\ c_n = 1,\ s_n = 0 \\ r_{i+1} &= (p_{i+1}^2 + (e_i^{(k)})^2)^{1/2} \\ g_{i+1} &= c_{i+1}\ e_i^{(k)} \\ h_{i+1} &= c_{i+1}\ p_{i+1} \\ e_{i+1}^{(k+1)} &= s_{i+1}\ r_{i+1} \\ c_i &= p_{i+1}/r_{i+1} \\ s_i &= e_i^{(k)}/r_{i+1} \\ p_i &= c_i\ (d_i^{(k)} - \alpha_k) - s_i g_{i+1} \\ d_{i+1}^{(k+1)} &= h_{i+1} + s_i[c_i g_{i+1} + s_i\ (d_i^{(k)} - \alpha_k)] \end{aligned} \qquad (17)$$

for i = n-1, n-2, ..., 1 and

$$e_1^{(k+1)} = s_1 p_1 \qquad d_1^{(k+1)} = c_1 p_1 \tag{18}$$

Suppose the (k-1)st eigenvalue has been found and is stored in d_{k-1}. At this stage the diagonal of the iterated matrix has eigenvalues in the first k-1 places and zeros in the first k-1 subdiagonal elements $e_1, \ldots e_{k-1}$. To find the k-th eigenvalue the QL iterations are continued until e_k is negligible or until 30 iterations are carried out. Furthermore, suppose some e_j, j = k+1, ..., n is negligible, then the QL iterations are carried out only on the submatrix with diagonal d_k, ..., d_j. That is, the matrix is considered to be the direct sum of two matrices of smaller order.

The criterion for negligibility is as follows: Just before iterating for the k-th eigenvalue set $h_k = \varepsilon(|d_k| + |e_k|)$ then any off diagonal element e_i for which $|e_i| \leq \max h_j$, j = 1, ..., k is regarded as negligible, where ε is the smallest positive floating point number which when added to 1.0 yields a sum not 1.0. On the CDC 6600 ε is taken to be 2^{-47}.

At the start of the iteration to find the k-th eigenvalue the shift α_k is taken to be the eigenvalue of the matrix

$$\begin{vmatrix} d_k & e_k \\ e_k & d_{k+1} \end{vmatrix}$$

which is nearer to d_k. This eigenvalue can be expressed as

$$d_k - \frac{e_k}{p + \text{sign}(p)\sqrt{p^2 + 1}},$$

where $p = \frac{d_{k+1} - d_k}{2 e_k}$.

The QR algorithm for symmetric tridiagonal matrices is known to converge (Reference 11, p. 548), and in testing it converged very rapidly. The routine allows up to thirty iterations per eigenvalue but this number is quite high. In testing on the CDC 6600 no more than seven were ever required.

The QR algorithm is carried out in single precision. For a quite detailed discussion of the computer implementation of this algorithm, see Reference 3.

The Rayleigh quotient ρ_i is computed using the relationship

$$\rho_i = \bar{x}_i^T A \bar{x}_i = \bar{x}_i^T (\bar{\lambda}_i \bar{x}_i + \eta_i) = \bar{\lambda}_i + \bar{x}_i^T \eta_i \qquad (19)$$

where $\bar{\lambda}_i$ and $\bar{x}_i$ are the computed eigenvalue and eigenvector respectively, A is the original matrix, and η_i is the residual vector $A\bar{x}_i - \bar{\lambda}_i \bar{x}_i$. The calculation of ρ_i is carried out in double precision. The $\bar{\lambda}_i$ is then changed from its QR computed eigenvalue to ρ_i which is then converted to single precision. Thus, upon return, EV contains the single precision versions of the members of the first column of DPA. Equation 2 of Section 3.4 shows that there is a true eigenvalue λ in the interval $[\rho_i - \epsilon_i, \rho_i + \epsilon_i]$. Thus a safe lower bound a_i on the distance between λ_i and its nearest neighbor may be computed:

$$a_1 = \rho_1 - \epsilon_1 - \rho_2 - \epsilon_2 , \qquad (20)$$

$$a_i = \min[\rho_i - \epsilon_i - \rho_{i+1} - \epsilon_{i+1}, \rho_{i-1} - \epsilon_{i-1} - \rho_i - \epsilon_i] \qquad (21)$$

for $i = 2, \ldots, n-1$, and

$$a_n = \rho_{n-1} - \epsilon_{n-1} - \rho_n - \epsilon_n . \qquad (22)$$

The expressions for the eigenvalue and eigenvector errors, Equations 3 and 4, are useful only if $\epsilon << a$. Accordingly, if $10 \cdot \epsilon \geq a$ then no error calculation is attempted and the corresponding value of ERVAL is returned as -1.0. If $10\epsilon < a$ then the eigenvalue and eigenvector errors are computed as ϵ^2/a and ϵ/a respectively and the factors $(1 - \epsilon^2/a^2)^{-1}$ and $(1 + \epsilon^2/a^2)$ of Equations 3 and 4 are ignored.

5. Space, Time, and Accuracy Considerations

The size of SEVER is about 1200_{10} words on the CDC 6600. In addition, at least $2n^2 + 8n$ words for the matrix, eigenvalues, eigenvectors, and error bounds are required and are supplied by the user's calling program. On the CDC 6600 it is possible to resolve a 200 x 200 matrix.

For n larger than about 15, the time required for a complete eigensystem calculation is very nearly proportional to n^3. For n smaller than 15, time required for subroutine linkage and initialization dominate and no simple relationship between time and n seems to exist. See Appendix B for machine-dependent details.

Accuracy is discussed in Section 3.5 and is not subject to user control. For machine-dependent details see Appendix B.

6. Testing Methods

6.1 General Discussion

There are many ways to test a routine which computes the eigensystem of a real symmetric matrix. A list of possible tests includes:

(1) $\det (A - \lambda I) = 0$

(2) $\text{trace } (A) = \Sigma\lambda_i$

(3) $Ax = \lambda x$ or $\|Ax - \lambda x\| = 0$

(4) Sum of idempotents, i.e., $A = \Sigma\lambda_i x_i x_i^T$, where $\|x\| = 1$

(5) Computed solution compared to known exact solution

(6) Orthogonality of computed eigenvectors

(7) The eigenvalues of A + I are 1 plus those of A, and the eigenvectors the same for each. Similarly the eigenvalues of αA are α times those of A.

(8) Extensive running of the routine for "shakedown" purposes and timing determinations

(9) Plausible results.

Items (1) and (4) are considered inconclusive and unnecessarily difficult to compute. Item (2) offers no additional information or assurance that is not given by other tests used to authenticate SEVER.

From a logical point of view, successful testing, no matter how extensive, is a necessary but not a sufficient condition for a routine to be valid.

6.2 Kinds of Tests Used

Item (3) is an internal check in SEVER. The value of $\|AX_i - \overline{\lambda}_i \overline{x}_i\|$ is returned in DPA(I,3) for the I-th eigenvalue-eigenvector pair. Of course one must test for a small value, since a zero will occur rarely. This test is quite inconclusive.

Item (5) is probably the most important test. Assurance of authenticity, a measure of precision of the routine, and a check on the sharpness of the error bounds are obtained by comparing true and computed solutions to nontrivial problems whose solutions are known.

Item (6) is checked by computing the Gram matrix of the matrix whose columns are the computed eigenvectors. The (i,j) element of the Gram matrix may be thought of as the cosine of the angle between eigenvectors i and j. This test measures the orthogonality of the eigenvectors but not their quality.

Item (7) is useful for checking the scale independence of the routine and offers a further check on the error bounds.

Items (8) and (9) are, of course, present in all tests and may be examined further by resolution of matrices whose elements are numbers chosen from a sequence of random numbers.

6.3 Normal Cases Tested

It is easy to construct small matrices with known eigensystems using Test (4) of Section 6.1. Such matrices usually present no difficulty and do not constitute a significant test. It is relatively easy to construct matrices whose elements are pseudo-random numbers and this was done to test in the sense of Items (8) and (9). It is extremely unlikely that a matrix of random numbers would exhibit clustered or multiple eigenvalues or any other difficulties.

6.4 Difficult Cases Tested

Possible difficulties arise with closely spaced eigenvalues and eigenvalues of multiplicity greater than one. In the latter case an eigenvalue admits linearly independent but nonorthogonal eigenvectors. For example, the matrix

$$\begin{bmatrix} 7 & 2 & 0 \\ 2 & 4 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

has eigenvalues 8, 3, and 3. The vectors $(-1,2,-1)^T$ and $(-2,4,3)^T$ are a pair of linearly independent nonorthogonal eigenvectors corresponding to eigenvalue 3. The eigenvalue 3 does, however, admit a pair of orthogonal eigenvectors. In fact, infinitely many such pairs exist.

A difficulty, in some methods, occurs when the largest and smallest eigenvalues are equal in absolute value. This situation is no problem to the methods used in SEVER and is not considered a difficult case for this routine.

The example of Rosser et al[10] is about as pathological as an eight by eight with small integer elements can get:

$$R_8 = \begin{bmatrix} 611 & 196 & -192 & 407 & -8 & -52 & -49 & 29 \\ 196 & 899 & 113 & -192 & -71 & -43 & -8 & -44 \\ -192 & 113 & 899 & 196 & 61 & 49 & 8 & 52 \\ 407 & -192 & 196 & 611 & 8 & 44 & 59 & -23 \\ -8 & -71 & 61 & 8 & 411 & -599 & 208 & 208 \\ -52 & -43 & 49 & 44 & -599 & 411 & 208 & 208 \\ -49 & -8 & 8 & 59 & 208 & 208 & 99 & -911 \\ 29 & -44 & 52 & -23 & 208 & 208 & -911 & 99 \end{bmatrix}$$

R_8 has eigenvalues:

$$\begin{array}{r} 10\sqrt{10405} = 1020.0490184299968238 \\ 1020 \\ 510+100\sqrt{26} = 1019.9019513592784830 \\ 1000 \\ 1000 \\ 510-100\sqrt{26} = 0.098048640721516997 1776 \\ \hline 0 \\ -10\sqrt{10405} = -1020.0490184299968238 \end{array}$$

See Appendix B for machine-dependent results of SEVER with R_8.

A large class of useful test matrices may be constructed using the Kronecker product $A \otimes B$. Let A be m by m and B be n by n. Then $A \otimes B$ is the nm by nm matrix:

$$A \otimes B = \begin{bmatrix} a_{11}\ B & a_{12}\ B & \cdots\cdots & a_{1m}\ B \\ a_{21}\ B & a_{22}\ B & \cdots\cdots & a_{2m}\ B \\ \vdots & & & \\ a_{m1}\ B & a_{m2}\ B & \cdots\cdots & a_{mm}\ B \end{bmatrix} . \qquad (23)$$

The eigenvalues of $A \otimes B$ are the nm products of an eigenvalue of A times one of B. Let x and y be eigenvectors of A and B with eigenvalues α and β respectively, then $x \otimes y$ is an eigenvector of $A \otimes B$ corresponding to $\alpha\beta$ where $x \otimes y$ is the mn component vector:

$$(x_1y_1, x_1y_2, \ldots, x_1y_n, X_2y_1, \ldots, x_2y_n, \ldots, x_my_1, \ldots, x_my_n)^T .$$

For details and proofs see Reference 6 or Reference 2.

A 32 x 32 test matrix A ($A = B \otimes R_8$) was constructed where

$$B = \begin{bmatrix} 15/16 & 5/8 & 5/16 & -1/8 \\ 5/8 & 15/16 & 1/8 & 5/16 \\ 5/16 & 1/8 & 15/16 & 5/8 \\ -1/8 & 5/16 & 5/8 & 15/16 \end{bmatrix} \qquad (24)$$

The eigenvalues of B are 2, 1.125, 0.5, and 0.125 exactly.

The matrix $A = B \otimes R_8$ is large enough to be nontrivial, is singular (in fact, rank (A) = 28), has multiple and clustered eigenvalues, and its nonzero eigenvalues span five orders of magnitude.

It is thought that matrices A, A + I, αA, and $\alpha A + I$ where $\alpha = 2^{-11}$ form a fairly significant test collection. Furthermore, each of these matrices admits an exact machine representation so that errors in the computed eigensystem are totally due to errors in the diagonalization. See Appendix B for machine-dependent results of SEVER with these matrices.

A nontrivial test matrix with predetermined eigenvalues $d_1, \ldots, d_n$ can be constructed as the product U^TDU where U is a nontrivial orthogonal matrix and $D = \text{diag}(d_1, \ldots, d_n)$. An extremely convenient choice for U is $1/\sqrt{n}\ H_n$ where H_n is an n x n symmetric Hadamard matrix of plus ones and minus ones and n is a power of two. A Hadamard matrix is one in which any two rows (or columns) agree in exactly half their places and disagree in half.[5] H_n for $n = 2^k$ can be constructed as the k-fold Kronecker product of the matrix

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

with itself.[7]

The product $P = 1/n\ H_n D H_n$, called a Hadamard product, has eigenvalues $d_1, \ldots, d_n$ and the columns (or rows) of $1/\sqrt{n}\ H_n$ form an orthogonal set of eigenvectors. Furthermore, if n is a power of two then each element of p is exactly representable in a binary computer for a large class of D.

It is thought the Hadamard products $1/8\ H_8 D_8 H_8$ and $1/16\ H_{16} D_{16} H_{16}$ form a significant pair of tests where D_8 has diagonal elements 1.0 and $1.0 + 2^{-k}$ for k = 38 to 44 and D_{16} has diagonal elements 1.0 and $1.0 + 2^{-k}$ for k = 24 to 38.

An example which further illustrates the difficulty in distinguishing between multiple and clustered eigenvalues is W_{21}^+, see Reference 11, p. 309. W_{21}^+ is a tridiagonal matrix whose main diagonal is 10, 9, 8, ..., 1, 0, 1, ..., 8, 9, 10, and whose sub and super diagonals are all 1's. W_{21}^+ has pairs of extremely clustered eigenvalues. In fact the pair of largest eigenvalues agrees to 15 decimal places. See Appendix B for machine-dependent results of calculations with this matrix.

An example which illustrates the sharpness of the error bounds is W_{21}^-, again Reference 11, p. 309. W_{21}^- is identical with W_{21}^+ except the elements on the lower half of the main diagonal of W_{21}^- are the negatives of those on the lower half of the main diagonal of W_{21}^+. W_{21}^- is singular and its nonzero eigenvalues occur in pairs differing only in sign.

6.5 Range, Error, and Fault Checks Tested

Both of the diagnostics in SEVER were checked. The diagnostics for the consistency of the input dimensions were checked by calling the routine with faulty parameters. The author was able to find no matrix with an eigenvalue which required more than thirty QR iterations. However, the viability of the related error check was tested by arbitrary lowering the limit on QR iterations to 5 in order to simulate an occurence of the error condition.

7. Remarks

Suppose C = A + iB is an n x n complex Hermitian matrix. Equating real and imaginary parts of

$$(A + iB)\ (x + iy) = \lambda(x + iy)$$

yields the 2n x 2n symmetric problem DZ = λZ where, in blocks,

$$D = \begin{pmatrix} A & -B \\ B & A \end{pmatrix} \qquad \text{and} \qquad Z = \begin{pmatrix} x \\ y \end{pmatrix} .$$

Each eigenvalue of C occurs as an eigenvalue of D of multiplicity two. The first n components of an eigenvector of D from the real part of an eigenvector of C and the remaining n components form the imaginary part. That is, let x be a 2n component eigenvector of D corresponding to eigenvalue λ and let the first n components of x be an n vector x_1 and the remaining n components be an n vector x_2. Then $x_1 + ix_2$ is an eigenvector of C corresponding to eigenvalue λ. Furthermore, the eigenvectors of C computed in this way are of length one and those corresponding to distinct eigenvalues are orthogonal.

There will be two real orthogonal eigenvectors of D corresponding to each one of C. However, when these two real 2n component vectors are each converted to n component complex eigenvectors of C it is easily seen that one is a scalar multiple of the other and that the scalar is a complex number of absolute value one.

In case D has some eigenvalue of multiplicity 2k, k > 1, care must be used in choosing eigenvectors of C in order to guarantee orthogonality.

8. Certification

This routine was subjected to a wide variety of tests. The performance of the routine throughout the tests was checked carefully.

The nature of the tests, the reliability of the routine, the error analyses conducted, and the observed variation in accuracy are reported in this document. While it is believed that the facts recorded and the judgments expressed regarding accuracy and reliability are strong indications of the general quality and validity of the routine, the tests should not be considered to be exhaustive. The use of this routine outside of the stated range of application or in violation of stated restrictions may produce unspecified results. The statements made in this document are intended to apply only to those versions of the indicated routine which are released by the Sandia Laboratories Mathematical Program Library project.

The author wrote SEVER and converted to FORTRAN the ALGOL routine in Reference 8 for tridiagonalization; converted the ALGOL routine in Reference 3 for eigenvalues and eigenvectors; added the Rayleigh quotient correction and error bound calculation; and performed the tests described above.

REFERENCES

1. Bailey, C. B., A Guide to the Sandia Mathematical Program Library, SC-M-69-337, Sandia Laboratories, Albuquerque, New Mexico, January 1970.

2. Bourbaki, N., Algèbre Multilinéaire, Hermann, Paris, p. 16 (1948).

3. Bowdler, H., Martin, R. S., Rensch, C. R., and Wilkinson, J. H., "The QR and QL Algorithms for Symmetric Matrices," Numerische Mathematik 11, 181-195 (1968).

4. Francis, J. G. F., "The QR Transformation, Parts I and II," Computer Journal 4, 265-271, 332-345 (1961, 1962).

5. Golomb, S. W., and Baumert, L. D., "The Search for Hadamard Matrices," The American Mathematical Monthly, 70, 12-17 (1963).

6. Halmos, P. R., Finite Dimensional Vector Spaces, Princeton, N.J. p. 96 (1958).

7. Hanson, R. J., "Certifying Linear Equation Solvers," SIGNUM Newsletter, 4, 21-29 (1969).

8. Martin, R. S., Reinsch, C., and Wilkinson, J. H., "Householder's Tridiagonalization of a Symmetric Matrix," Numerische Mathematik 11, 181-195 (1968).

9. Ortega, J., "The Givens-Householder Method for Symmetric Matrices," Mathematical Methods for Digital Computers, Vol. 2, edited by A. Ralston and H. Wilf, New York (1967).

10. Rosser, J. B., Lanczos, C., Hestenes, M. R., and Karush, W., "Separation of Close Eigenvalues of a Real Symmetric Matrix," Journal of Research of the National Bureau of Standards, 47, No. 4, 291-297 (1951).

11. Wilkinson, J. H., The Algebraic Eigenvalue Problem, Oxford (1965).

12. Wilkinson, J. H., "Rigorous Error Bounds for Computed Eigensystems," Computer Journal 4, 230-241 (1961).

APPENDIX A

The SEVER Listing

The changes which may have to be made for use of SEVER on machines other than the CDC 6600 are the DATA statements in lines SEV 480 and SEV 490 and the octal constants in lines SEV 640 and SEV 2370. ERRCHK is called in SEV 530 and SEV 1580.

```
      SUBROUTINE SEVER(A,EV,VEC,ERVAL,DPA,N,NDIM,KER)                    SEV   10
C                                                                        SEV   20
C     SANDIA MATHEMATICAL PROGRAM LIBRARY                                SEV   30
C     MATHEMATICAL COMPUTING SERVICES DIVISION  9422                     SEV   40
C     SANDIA LABORATORIES                                                SEV   50
C     P. O. BOX 5800                                                     SEV   60
C     ALBUQUERQUE, NEW MEXICO  87115                                     SEV   70
C                                                                        SEV   80
C     WRITTEN BY WILLIAM R. GAVIN                                        SEV   90
C                                                                        SEV  100
C     CONTROL DATA 6600 VERSION                                          SEV  110
C                                                                        SEV  120
C     ABSTRACT                                                           SEV  130
C                                                                        SEV  140
C        THIS ROUTINE COMPUTES ALL THE EIGENVALUES OF A REAL SYMMETRIC   SEV  150
C        MATRIX AND AN EIGENVECTOR OF LENGTH ONE CORRESPONDING TO EACH   SEV  160
C        EIGENVALUE.  THE COMPUTED EIGENVALUES ARE SUBJECTED TO A        SEV  170
C        RAYLEIGH QUOTIENT CORRECTION.  UPPER BOUNDS ON THE ABSOLUTE     SEV  180
C        ERRORS OF THE COMPUTED EIGENSYSTEM ARE CALCULATED.              SEV  190
C                                                                        SEV  200
C     DESCRIPTION OF PARAMETERS                                          SEV  210
C                                                                        SEV  220
C       A        THE SYMMETRIC MATRIX WHOSE EIGENSYSTEM IS TO BE COMPUTEDSEV  230
C       EV       CONTAINS THE EIGENVALUES OF A IN NON-INCREASING        SEV  240
C                ALGEBRAIC ORDER                                        SEV  250
C       VEC      CONTAINS THE EIGENVECTORS. COLUMN J OF VEC CONTAINS THE SEV  260
C                EIGENVECTOR CORRESPONDING TO THE EIGENVALUE IN EV(J).   SEV  270
C                THE EIGENVECTOR IS OF LENGTH UNITY.                    SEV  280
C       ERVAL    CONTAINS UPPER BOUNDS ON THE ABSOLUTE ERRORS OF THE    SEV  290
C                COMPUTED EIGENVALUES                                   SEV  300
C       DPA      IS AN NDIM BY 3 DOUBLE PRECISION ARRAY                 SEV  310
C                  COL. 1  CONTAINS RAYLEIGH CORRECTED EIGENVALUES.     SEV  320
C                  COL. 2  CONTAINS THE LENGTHS OF THE RESIDUAL VECTORS SEV  330
C                  COL. 3  CONTAINS UPPER BOUNDS ON THE ABSOLUTE ERRORS SEV  340
C                          OF THE COMPUTED EIGENVECTORS AS MEASURED IN   SEV  350
C                          THE TWO NORM                                  SEV  360
C       N        ORDER OF THE MATRIX WHOSE EIGENSYSTEM IS COMPUTED      SEV  370
C       NDIM     DIMENSION OF ARRAYS PASSED INTO SUBROUTINE             SEV  380
C       KER      OUTPUT ERROR CODE                                      SEV  390
C                1 NORMAL  (NO ERROR)                                   SEV  400
C                2 N LESS THAN 3 OR GREATER THAN NDIM                   SEV  410
C                3 MORE THAN 30 QR ITERATIONS REQUIRED                  SEV  420
C                                                                       SEV  430
C------                                                                 SEV  440
      DIMENSION A(NDIM,ND..),EV(NDIM),VEC(NDIM,NDIM),ERVAL(NDIM)        SEV  450
      DIMENSION DPA(NDIM,3),MESG(7),MESG2(7)                            SEV  460
      DOUBLE PRECISION DPA,DS,DP,DTV                                    SEV  470
      DATA(MESG= 2  ,43H INPUT DIMENSIONS ARE INCONSISTANT          )   SEV  480
      DATA(MESG2= 3 ,43H NO CONVERGENCE IN 30 QR ITERATIONS         )   SEV  490
      KER=1                                                             SEV  500
      IF(   N.LE.2.OR.  N.GT.NDIM) KER=2                                SEV  510
      IF(KER.EQ.1) GOTO 1                                               SEV  520
      CALL ERRCHK(43,MESG)                                              SEV  530
      RETURN                                                            SEV  540
```

```
C                                                                       SEV  550
C          COPY A INTO VECTOR ARRAY                                     SEV  560
C                                                                       SEV  570
    1 DO 2 I=1,N                                                        SEV  580
      DO 2 J=1,N                                                        SEV  590
    2 VEC(I,J)=A(I,J)                                                   SEV  600
      NM1=N-1                                                           SEV  610
      NM2=N-2                                                           SFV  620
C******                                                                 SEV  630
      EPS=0006140000000000000                                           SEV  640
C          EPS IS THE QUOTIENT OF THE SMALLEST POSITIVE FLOATING POINT   SEV  650
C          NUMBER AND THE SMALLEST POSITIVE FLOATING POINT NUMBER WHICH  SEV  660
C          WHEN ADDED TO 1.0 YIELDS A SUM NOT 1.0 .  THIS IS A MACHINE   SEV  670
C          DEPENDENT CONSTANT AND MUST BE CHANGED FOR USE OF THIS        SEV  680
C          ROUTINE ON COMPUTERS OTHER THAN THE CDC 6600.                 SEV  690
C******                                                                 SEV  700
      NP2=N+2                                                           SEV  710
C                                                                       SEV  720
C          BFGIN TRIDIAGONALIZATION                                     SEV  730
C                                                                       SEV  740
C------                                                                 SEV  750
C          THE TRIDIAGONALIZATION IS A FORTRAN VERSION OF THE ALGOL     SEV  760
C          ROUTINE OF MARTIN, REINSCH, AND WILKINSON. NUMER MATH 11,    SEV  770
C          181-195(1968).                                               SEV  780
C------                                                                 SEV  790
      DO 15 IB=2,N                                                      SEV  800
      I=NP2-IB                                                          SEV  810
      L=I-2                                                             SEV  820
      FACT=VEC(I,I-1)                                                   SEV  830
      GSTO=0.0                                                          SEV  840
      IF(L.EQ.0) GOTO 4                                                 SEV  850
      DO 3 K=1,L                                                        SEV  860
    3 GSTO=GSTO+VEC(I,K)**2                                             SEV  870
    4 HOLD=GSTO+FACT*FACT                                               SEV  880
C------                                                                 SEV  890
C          OMIT THIS TRANSFORMATION IF GSTO IS TOO SMALL TO MAKE        SEV  900
C          IT ORTHOGONAL                                                SEV  910
C------                                                                 SEV  920
      IF(GSTO.GT.EPS) GOTO 5                                            SEV  930
      ERVAL(I)=FACT                                                     SEV  940
      HOLD=0.0                                                          SEV  950
      GOTO 14                                                           SEV  960
    5 L=L+1                                                             SEV  970
      TEMP=SQRT(HOLD)                                                   SEV  980
      IF(FACT.GT.0.0) TEMP=-TEMP                                        SEV  990
      ERVAL(I)=TEMP                                                     SEV 1000
      GSTO=TEMP                                                         SEV 1010
      HOLD=HOLD-FACT*GSTO                                               SEV 1020
      VEC(I,I-1)=FACT-GSTO                                              SEV 1030
      FACT=0.0                                                          SEV 1040
      DO 9 J=1,L                                                        SEV 1050
      VEC(J,I)=VEC(I,J)/HOLD                                            SEV 1060
      GSTO=0.0                                                          SEV 1070
      DO 6 K=1,J                                                        SEV 1080
```

```
    6 GSTO=GSTO+VEC(J,K)*VEC(I,K)                                          SEV 1090
      JP1=J+1                                                              SEV 1100
      IF(JP1.GT.L)      GOTO 8                                             SEV 1110
      DO 7 K=JP1,L                                                         SEV 1120
    7 GSTO=GSTO+VEC(K,J)*VEC(I,K)                                          SEV 1130
    8 ERVAL(J)=GSTO/HOLD                                                   SEV 1140
      FACT=FACT+GSTO*VEC(J,I)                                              SEV 1150
    9 CONTINUE                                                             SEV 1160
   10 HH=FACT/(HOLD+HOLD)                                                  SEV 1170
      DO 13 J=1,L                                                          SEV 1180
      FACT=VEC(I,J)                                                        SEV 1190
      ERVAL(J)=ERVAL(J)-HH*FACT                                            SEV 1200
      GSTO=ERVAL(J)                                                        SEV 1210
      DO 12 K=1,J                                                          SEV 1220
   12 VEC(J,K)=VEC(J,K)-FACT*ERVAL(K)-GSTO*VEC(I,K)                        SEV 1230
   13 CONTINUE                                                             SEV 1240
   14 EV(I)=HOLD                                                           SEV 1250
   15 CONTINUE                                                             SEV 1260
       EV (1)=0.0                                                         SEV 1270
      ERVAL(1)=0.0                                                         SEV 1280
C                                                                          SEV 1290
C          ACCUMULATE TRANSFORMATION MATRIX                                SEV 1300
C                                                                          SEV 1310
      EV(1)=VEC(1,1)                                                       SEV 1320
      VEC(1,1)=1.0                                                         SEV 1330
      DO 25 I=2,N                                                          SEV 1340
      L=I-1                                                                SEV 1350
      IF(EV(I).EQ.0.0)  GOTO 23                                            SEV 1360
      DO 22 J=1,L                                                          SEV 1370
      GSTO=0.0                                                             SEV 1380
      DO 19 K=1,L                                                          SEV 1390
   19 GSTO=GSTO+VEC(I,K)*VEC(K,J)                                          SEV 1400
      DO 21 K=1,L                                                          SEV 1410
   21 VEC(K,J)=VEC(K,J)-GSTO*VEC(K,I)                                      SEV 1420
   22 CONTINUE                                                             SEV 1430
   23 EV(I)=VEC(I,I)                                                       SEV 1440
      VEC(I,I)=1.0                                                         SEV 1450
      DO 24 J=1,L                                                          SEV 1460
      VEC(I,J)=0.0                                                         SEV 1470
   24 VEC(J,I)=0.0                                                         SEV 1480
   25 CONTINUE                                                             SEV 1490
C------                                                                    SEV 1500
C          THE TRIDIAGONALIZATION IS NOW COMPLETE                          SEV 1510
C          DIAGONAL ELEMENTS ARE IN EV                                     SEV 1520
C          OFF DIAGONAL ELEMENTS ARE IN ERVAL                              SEV 1530
C          COMPUTE THE EIGENVALUES AND EIGENVECTORS OF THE TRIDIAGONAL     SEV 1540
C          MATRIX                                                          SEV 1550
C------                                                                    SEV 1560
      CALL QRVEC(EV,ERVAL,VEC,N,NDIM,KER)                                  SEV 1570
      IF(KER.NE.1)  CALL ERRCHK(43,MESG2)                                  SEV 1580
C------                                                                    SEV 1590
C          COMPUTE RAYLEIGH CORRECTED EIGENVALUES                          SEV 1600
C------                                                                    SEV 1610
      DO 49  IV=1,N                                                        SEV 1620
```

```
      DO 45    J=1,N                                                    SEV 1630
      DS=0.0D+00                                                        SEV 1640
      DO 43   K=1,N                                                     SEV 1650
      DP=DBLE(A(J,K))*DBLE(VEC(K,IV))                                   SEV 1660
   43 DS=DS+DP                                                          SEV 1670
      DS=DS-DBLE(EV(IV))*DBLE(VEC(J,IV))                                SEV 1680
   45 DPA(J,3)=DS                                                       SEV 1690
      DS=0.0D+00                                                        SEV 1700
      DP=0.0D+00                                                        SEV 1710
      DO 47 K=1,N                                                       SEV 1720
      DP=DP+DPA(K,3)**2                                                 SEV 1730
   47 DS=DS+DBLE(VEC(K,IV))*DPA(K,3)                                    SEV 1740
      DPA(IV,2)=DSQRT(DP)                                               SEV 1750
      DPA(IV,1)=DBLE(EV(IV))+DS                                         SEV 1760
      EV(IV)=DPA(IV,1)                                                  SEV 1770
   49 CONTINUE                                                          SEV 1780
C------                                                                 SEV 1790
C           A TRUE EIGENVALUE IS IN THE INTERVAL                        SEV 1800
C         * DPA(IV,1)-DPA(IV,2) , DPA(IV,1)+DPA(IV,2) *                 SEV 1810
C------                                                                 SEV 1820
C         COMPUTE UPPER BOUNDS ON THE ABSOLUTE ERRORS IN THE            SEV 1830
C         COMPUTED EIGENSYSTEM                                          SEV 1840
C------                                                                 SEV 1850
      DS = DPA(1,1)-DPA(1,2)-DPA(2,1)-DPA(2,2)                          SEV 1860
      IF(DPA(1,2).LT.1.0D+01*DS) GOTO 51                                SEV 1870
      ERVAL(1)=-1.0                                                     SEV 1880
      DPA(1,3)=DPA(1,2)                                                 SEV 1890
      ERVAL(2)=-1.0                                                     SEV 1900
      DPA(2,3)=DPA(2,2)                                                 SEV 1910
      NUP=3                                                             SEV 1920
      GOTO 52                                                           SEV 1930
   51 DPA(1,3)=DPA(1,2)/DS                                              SEV 1940
      ERVAL(1)=DPA(1,3)*DPA(1,2)                                        SEV 1950
      NUP=2                                                             SEV 1960
   52 IF(NUP.GT.NM1) GOTO 57                                            SEV 1970
      DO 54 K=NUP,NM1                                                   SEV 1980
      DP=DPA(K-1,1)-DPA(K,1)-DPA(K-1,2)-DPA(K,2)                        SEV 1990
      DS=DPA(  K,1)-DPA(K,2)-DPA(K+1,1)-DPA(K+1,2)                      SEV 2000
      IF(DS.LT.DP) DP=DS                                                SEV 2010
      IF(DPA(K,2).LT.1.0D+01*DP)  GOTO 53                               SEV 2020
      DPA(K,3)=DPA(K,2)                                                 SEV 2030
      ERVAL(K)=-1.0                                                     SEV 2040
      GOTO 54                                                           SEV 2050
   53 DPA(K,3)=DPA(K,2)/DP                                              SEV 2060
      ERVAL(K)=DPA(K,3)*DPA(K,2)                                        SEV 2070
   54 CONTINUE                                                          SEV 2080
   57 CONTINUE                                                          SEV 2090
      DS = DPA(NM1,1)-DPA(NM1,2)-DPA(N,1)-DPA(N,2)                      SEV 2100
      IF(DPA(N,2).LT.1.0D+01*DS)GOTO 56                                 SEV 2110
      ERVAL(N) = -1.0                                                   SEV 2120
      DPA(N,3)=DPA(N,2)                                                 SEV 2130
      RETURN                                                            SEV 2140
   56 DPA(N,3)=DPA(N,2)/DS                                              SEV 2150
      ERVAL(N)=DPA(N,2)*DPA(N,3)                                        SEV 2160
      RETURN                                                            SEV 2170
       END                                                              SEV 2180
```

```
      SUBROUTINE QRVEC(DG,EG,VEC,N,NDIM,KER)                            SEV 2190
C------                                                                 SEV 2200
C     QRVEC     COMPUTES EIGENVALUES AND EIGENVECTORS OF A SYMMETRIC    SEV 2210
C               TRIDIAGONAL MATRIX                                      SEV 2220
C                                                                       SEV 2230
C        DG     ARRAY OF DIAGONAL ELEMENTS                              SEV 2240
C                                                                       SEV 2250
C        EC     ARRAY OF SUB(SUPER) DIAGONAL ELEMENTS                   SEV 2260
C                                                                       SEV 2270
C        VEC    VECTOR ARRAY.  THE VECTORS ARE TRANSFORMED INTO VECTORS SEV 2280
C               OF THE ORIGINAL MATRIX AS THEY ARE COMPUTED.            SEV 2290
C------                                                                 SEV 2300
C------                                                                 SEV 2310
C         THIS SUBROUTINE IS A FORTRAN VERSION OF THE ALGOL ROUTINE     SEV 2320
C         OF BOWDLER, MARTIN, REINSCH, AND WILKINSON.  NUMER. MATH. 11, SEV 2330
C         293-306(1968)                                                 SEV 2340
C------                                                                 SEV 2350
      DIMENSION DG(NDIM),EG(NDIM),VEC(NDIM,NDIM)                        SEV 2360
      EPS=0164140000000000002000                                        SEV 2370
C******                                                                 SEV 2380
C         EPS IS THE SMALLEST POSITIVE FLOATING POINT NUMBER WHICH WHEN SEV 2390
C         ADDED TO 1.0 YIELDS A FLOATING POINT SUM NOT 1.0 . THIS IS A  SEV 2400
C         MACHINE DEPENDENT CONSTANT AND MUST BE CHANGED FOR USE OF     SEV 2410
C         THIS PROGRAM ON COMPUTERS OTHER THAN THE CDC 6600.            SEV 2420
C******                                                                 SEV 2430
      NM1=N-1                                                           SEV 2440
      DO 1 I=2,N                                                        SEV 2450
    1 EG(I-1)=EG(I)                                                     SEV 2460
      EG(N)=0.0                                                         SEV 2470
      B=0.0                                                             SEV 2480
      F=0.0                                                             SEV 2490
C                                                                       SEV 2500
C         DIAGONALIZATION LOOP                                          SEV 2510
C                                                                       SEV 2520
      DO 14 L=1,N                                                       SEV 2530
      J=0                                                               SEV 2540
      H=EPS*(ABS(DG(L))+ABS(EG(L)))                                     SEV 2550
      IF(B.LT.H) B=H                                                    SEV 2560
C                                                                       SEV 2570
C         LOOK FOR SMALL SUB-DIAGONAL ELEMENT                           SEV 2580
C                                                                       SEV 2590
      DO  2 M=L,N                                                       SEV 2600
      KM=M                                                              SEV 2610
      IF(ABS(EG(M)).LT.B)  GOTO 3                                       SEV 2620
    2 CONTINUE                                                          SEV 2630
    3 IF(KM.EQ.L)  GOTO 13                                              SEV 2640
C                                                                       SEV 2650
C         CHECK ITERATION COUNT                                         SEV 2660
C                                                                       SEV 2670
    4 IF(J.GT.30)  GOTO 18                                              SEV 2680
      J=J+1                                                             SEV 2690
C                                                                       SEV 2700
C         FORM THE SHIFT FOR THIS ITERATION                             SEV 2710
C                                                                       SEV 2720
```

```
      P=(DG(L+1)-DG(L))/(EG(L)+EG(L))                                   SEV 2730
      R=SQRT(P*P+1.0)                                                   SEV 2740
      IF(P) 5,6,6                                                       SEV 2750
    5 DENOM=P-R                                                         SEV 2760
      GOTO 7                                                            SEV 2770
    6 DENOM=P+R                                                         SEV 2780
    7 H=DG(L)-EG(L)/DENOM                                               SEV 2790
      DO 8 I=L,N                                                        SEV 2800
    8 DG(I)=DG(I)-H                                                     SEV 2810
      F=F+H                                                             SEV 2820
C                                                                       SEV 2830
C         ONE QL TRANSFORMATION                                         SEV 2840
C                                                                       SEV 2850
      P=DG(KM)                                                          SEV 2860
      C=1.0                                                             SEV 2870
      S=0.0                                                             SEV 2880
      KM1=KM-1                                                          SEV 2890
      KL=KM1+L                                                          SEV 2900
      DO 12 IX=L,KM1                                                    SEV 2910
      I=KL-IX                                                           SEV 2920
      G=C*EG(I)                                                         SEV 2930
      H=C*P                                                             SEV 2940
      IF(ABS(P).LT.ABS(EG(I)))GOTO 9                                    SEV 2950
      C=EG(I)/P                                                         SEV 2960
      R=SQRT(C*C+1.0)                                                   SEV 2970
      EG(I+1)=S*P*R                                                     SEV 2980
      S=C/R                                                             SEV 2990
      C=1.0/R                                                           SEV 3000
      GOTO 10                                                           SEV 3010
    9 C=P/EG(I)                                                         SEV 3020
      R=SQRT(C*C+1.0)                                                   SEV 3030
      EG(I+1)=S*EG(I)*R                                                 SEV 3040
      S=1.0/R                                                           SEV 3050
      C=C*S                                                             SEV 3060
   10 P=C*DG(I)-S*G                                                     SEV 3070
      DG(I+1)=H+S*(C*G+S*D  G(I))                                       SEV 3080
C                                                                       SEV 3090
C         END OF QL STEP   FORM THE VECTOR                              SEV 3100
C                                                                       SEV 3110
      DO 11 K=1,N                                                       SEV 3120
      H=VEC(K,I+1)                                                      SEV 3130
      VEC(K,I+1)=S*VEC(K,I)+C*H                                         SEV 3140
      VEC(K,I)=C*VEC(K,I)-S*H                                           SEV 3150
   11 CONTINUE                                                          SEV 3160
   12 CONTINUE                                                          SEV 3170
      EG(L)=S*P                                                         SEV 3180
      DG(L)=C*P                                                         SEV 3190
      IF(ABS(EG(L)).GT.B)  GOTO 4                                       SEV 3200
C                                                                       SEV 3210
C         AN EIGENVALUE HAS BEEN FOUND   UNDO THE SHIFT                 SEV 3220
C                                                                       SEV 3230
   13 DG(L)=DG(L)+F                                                     SEV 3240
   14 CONTINUE                                                          SEV 3250
C                                                                       SEV 3260
```

```
C          END OF DIAGONALIZATION LOOP                                    SEV 3270
C                                                                         SEV 3280
C         SORT EIGENVALUES IN DESCENDING ORDER AND PLACE VECTORS          SEV 3290
C          ACCORDINGLY                                                    SEV 3300
C                                                                         SEV 3310
      DO 17 J=1,NM1                                                       SEV 3320
      IX=J                                                                SEV 3330
      JP1=J+1                                                             SEV 3340
      TP=DG(J)                                                            SEV 3350
      DO 15 K=JP1,N                                                       SEV 3360
      IF(DG(K).LE.TP)  GOTO 15                                            SEV 3370
      TP=DG(K)                                                            SEV 3380
      IX=K                                                                SEV 3390
   15 CONTINUE                                                            SEV 3400
      IF(IX.EQ.J)  GOTO 17                                                SEV 3410
      TP=DG(IX)                                                           SEV 3420
      DG(IX)=DG(J)                                                        SEV 3430
      DG(J)=TP                                                            SEV 3440
C                                                                         SEV 3450
C         INTERCHANGE COLUMNS OF VECTOR MATRIX                            SEV 3460
C                                                                         SEV 3470
      DO 16 L=1,N                                                         SEV 3480
      TP=VEC(L,J)                                                         SEV 3490
      VEC(L,J)=VEC(L,IX)                                                  SEV 3500
   16 VEC(L,IX)=TP                                                        SEV 3510
   17 CONTINUE                                                            SEV 3520
      RETURN                                                              SEV 3530
C                                                                         SEV 3540
C          ERROR RETURN (MORE THAN 30 ITERATIONS)                         SEV 3550
C                                                                         SEV 3560
   18 KER=3                                                               SEV 3570
      RETURN                                                              SEV 3580
      END                                                                 SEV 3590
```

APPENDIX B

Test Results of SEVER on the CDC 6600

APPENDIX B

Test Results of SEVER on the CDC 6600

The table of execution times for SEVER was constructed using matrices whose elements were taken from a sequence of random numbers uniformly distributed between -1 and +1. The sequence was computed using the CDC 6600 library function RANF. Two separate executions of each order from 10 through 100 in increments of 5 were done and the table reflects the average of these two times. One 150 x 150 randomly constructed matrix was resolved. In no case were any of the eigenvalues of the randomly constructed matrices clustered or multiple. Thus it was possible for SEVER to compute an error bound for each eigenvalue of each matrix.

Table of Execution Times in Seconds

n	Time	n	Time	n	Time
10	0.110	45	7.014	80	37.242
15	0.315	50	9.459	85	44.460
20	0.709	55	12.463	90	52.662
25	1.298	60	16.173	95	61.651
30	2.191	65	20.338	100	72.216
35	3.402	70	25.235	150	235.104
40	4.966	75	30.825		

For the larger values of n, the time in seconds is nearly kn^3 where k is approximately 7×10^{-5}.

One 200 x 200 matrix was loaded to verify that SEVER was indeed capable of handling such a matrix on a CDC 6600 with 3×2^{15} available core locations. In this case the required arrays and subroutine SEVER as compiled by the FUN compiler required approximately 86,000 locations. Subroutine SEVER requires about 1200 decimal locations.

The examples which follow, with the exception of the last one, are as described in Section 6.4. In all examples except the four 32 x 32 Kronecker products and the 16 x 16 Hadamard product, the upper triangle of the test matrix is displayed. In all cases the single precision eigenvalues, eigenvalue errors, and residual lengths are displayed followed by the double precision eigenvalues with 28 significant figures printed out. The eigenvectors and Gram matrices are displayed where space allows. Only the first eight components of the eighth, sixteenth, twenty-fourth, and thirty-second eigenvectors of the Kronecker products are displayed. These components constitute all of the distinct elements of the eigenvectors of these matrices. Only the first eight components of the first eight eigenvectors of the 16 x 16 Hadamard product are displayed.

The Hadamard products H_{16} and H_8 illustrate the inevitable deterioration in the quality of the computed eigenvectors in the case of extremely clustered eigenvalues. The computed eigensystems of these matrices also point out the inconclusiveness of small residual vectors as measures of the quality of computed eigenvectors. The computed eigenvalues of both matrices are correct to at least the first 48 bits of their floating point representations.

In cases where the exact eigenvalues and eigenvector components are known the decimal position of the first significant digit of the actual absolute error is underlined. The reader can compare this error with the computed error bound.

The last example is Rosser's matrix, R8, with the last two bits of each element randomly perturbed. This example is meant to show that the error induced by inexact knowledge of the input matrix can be greater than the error induced by inexact diagonalization.

THE FACTOR B , FIRST FACTOR OF KRONECKER PRODUCT SEC 6.4
EIGENVALUES 2.0, 1.125, 0.5, 0.125 EXACTLY

I	J	A(I,J)	I	J	A(I,J)	I	J	A(I,J)	I	J	A(I,J)	I	J	A(I,J)	I	J	A(I,J)	I	J	A(I,J)
1	1	9.375E-01	1	2	6.250E-01	1	3	3.125E-01	1	4	-1.250E-01	2	2	9.375E-01	2	3	1.250E-01	2	4	-3.125E-01
3	3	9.375E-01	3	4	-6.250E-01	4	4	9.375E-01												

	EIGENVALUE	VALUE ERROR	VECTOR ERROR	RESIDUAL LENGTH		EIGENVALUE	VALUE ERROR	VECTOR ERROR	RESIDUAL LENGTH
1	2.000000000000E+00	8.63E-27	9.93D-14	8.69D-14	2	1.125000000000E+00	9.89E-27	1.26D-13	7.86D-14
3	5.000000000000E-01	1.19E-27	5.64D-14	2.11D-14	4	1.250000000000E-01	4.76E-29	1.13D-14	4.22D-15

1 1.99999999999999999999999994D+00 2 1.12499999999999999999999992D+00 3 4.99999999999999999999999985D-01

4 1.25000000000000000000000003D-01

EIGENVECTOR NO. 1
-.50000000000 -.50000000000 -.50000000000 .50000000000

EIGENVECTOR NO. 2
.50000000000 .50000000000 -.50000000000 .50000000000

EIGENVECTOR NO. 3
.50000000000 -.50000000000 .50000000000 .50000000000

EIGENVECTOR NO. 4
.50000000000 -.50000000000 -.50000000000 -.50000000000

GRAM MATRIX

1	1.0E+00	-8.0E-15	3.6E-15	-8.0E-15
2	1.0E+00	3.6E-15	4.4E-15	
3	1.0E+00	3.6E-15		
4	1.0E+00			

EIGEN SYSTEM CALCULATION TOOK .012 SECONDS

ROSSER,S MATRIX R8 SEC 6.4 SECOND FACTOR OF KRONECKER PRODUCT

I	J	A(I,J)	I	J	A(I,J)	I	J	A(I,J)	I	J	A(I,J)	I	J	A(I,J)	I	J	A(I,J)	I	J	A(I,J)
1	1	6.110E+02	1	2	1.960E+02	1	3	-1.920E+02	1	4	4.070E+02	1	5	-8.000E+00	1	6	-5.200E+01	1	7	-4.900E+01
1	8	2.900E+01	2	2	8.990E+02	2	3	1.130E+02	2	4	-1.920E+02	2	5	-7.100E+01	2	6	-4.300E+01	2	7	-8.000E+00
2	8	-4.400E+01	3	3	8.990E+02	3	4	1.960E+02	3	5	6.100E+01	3	6	4.900E+01	3	7	8.000E+00	3	8	5.200E+01
4	4	6.110E+02	4	5	8.000E+00	4	6	4.400E+01	4	7	5.900E+01	4	8	-2.300E+01	5	5	4.110E+02	5	6	-5.990E+02
5	7	2.080E+02	5	8	2.080E+02	6	6	4.110E+02	6	7	2.080E+02	6	8	2.080E+02	7	7	9.900E+01	7	8	-9.110E+02
8	8	9.900E+01																		

	EIGENVALUE	VALUE ERROR	VECTOR ERROR	RESIDUAL LENGTH		EIGENVALUE	VALUE ERROR	VECTOR ERROR	RESIDUAL LENGTH
1	1.020049018430E+03	9.48E-20	1.39D-09	6.82D-11	2	1.020000000000E+03	1.17E-19	1.55D-09	7.58D-11
3	1.019901951359E+03	2.52E-20	5.07D-10	4.97D-11	4	1.000000000000E+03	-1.00E+00	4.22D-11	4.22D-11
5	1.000000000000E+03	-1.00E+00	5.31D-11	5.31D-11	6	9.804864072152E-02	2.89E-20	5.43D-10	5.32D-11
7	1.460509139013E-23	2.99E-20	5.52D-10	5.41D-11	8	-1.020049018430E+03	5.59E-24	7.40D-14	7.55D-11

1 1.020049018429996823846313134D+03 2 1.019999999999999999999937170+03 3 1.019901951359278483002829260D+03

4 1.000000000000000000000000018D+03 5 1.000000000000000000000000025D+03 6 9.804864072151699717759827549D-02

7 1.460509139012753551394369222D-23 8 -1.020049018429996823846313784D+03

EIGENVECTOR NO. 1
.63244793386 .31622396687 .31622396687 .63244793385 -.00155008018 -.00155008029 .00310016049 .00310016046

EIGENVECTOR NO. 2
.22360679766 -.44721359553 -.44721359554 .22360679770 .44721359542 -.44721359542 .22360679792 -.22360679792

EIGENVECTOR NO. 3
-.06232066164 .03116033092 -.03116033067 .06232066155 -.31468878890 .31468878890 .62937757750 -.62937757750

EIGENVECTOR NO. 4
-.09408298680 .64469804973 .18816597360 -.32234902487 .38382257766 -.44904144568 .19191128883 -.22452072284

EIGENVECTOR NO. 5
.37302599319 .41995764628 -.74605198638 -.20997882314 -.24633357238 .07976076771 -.12316678619 .03988038386

EIGENVECTOR NO. 6
.62937757756 -.31468878878_ .31468878878_ -.62937757756 -.03116033080 .03116033080 .06232066160_ -.06232066160_

EIGENVECTOR NO. 7
-.04472135955_ -.08944271910_ .08944271910_ .04472135955_ -.62609903370_ -.62609903370_ -.31304951685_ -.31304951685_

EIGENVECTOR NO. 8
.00310016048_ .00155008024_ .00155008024_ .00310016048_ .31622396691_ .31622396691_ -.63244793383_ -.63244793383_

GRAM MATRIX

1	1.0E+00	1.5E-15	-3.7E-16	4.7E-15	1.7E-14	8.2E-15	-9.9E-16	-2.8E-16
2	1.0E+00	-4.4E-15	-6.9E-15	-1.4E-14	2.9E-15	-1.2E-14	5.3E-15	
3	1.0E+00	8.0E-15	4.2E-15	2.2E-16	2.0E-14	0.		
4	1.0E+00	7.9E-15	1.4E-15	-6.2E-15	7.1E-15			
5	1.0E+00	1.0E-15	-4.5E-15	-1.2E-15				
6	1.0E+00	-5.3E-15	1.8E-15					
7	1.0E+00	2.1E-14						
8	1.0E+00							

EIGEN SYSTEM CALCULATION TOOK .054 SECONDS

THE KRONECKER PRODUCT B X R8 SEC 6.4.

	EIGENVALUE	VALUE ERROR	VECTOR ERROR	RESIDUAL LENGTH		EIGENVALUE	VALUE ERROR	VECTOR ERROR	RESIDUAL LENGTH
1	2.040098036860E+03	1.05E-17	1.04D-08	1.02D-09	2	2.040000000000E+03	8.01E-18	9.04D-09	8.86D-10
3	2.039803902719E+03	2.84E-18	3.81D-09	7.46D-10	4	2.000000000000E+03	-1.00E+00	8.70D-10	8.70D-10
5	2.000000000000E+03	-1.00E+00	1.03D-09	1.03D-09	6	1.147555145734E+03	5.52E-18	1.00D-08	5.51D-10
7	1.147500000000E+03	5.30E-18	9.80D-09	5.41D-10	8	1.147389695279E+03	2.35E-18	4.62D-09	5.09D-10
9	1.125000000000E+03	-1.00E+00	3.95D-10	3.95D-10	10	1.125000000000E+03	-1.00E+00	5.12D-10	5.12D-10
11	5.100245092150E+02	1.68E-18	8.29D-09	2.03D-10	12	5.100000000000E+02	1.67E-18	8.24D-09	2.02D-10
13	5.099509756796E+02	7.29E-19	3.86D-09	1.89D-10	14	5.000000000000E+02	-1.00E+00	1.71D-10	1.71D-10
15	5.000000000000E+02	-1.00E+00	1.82D-10	1.82D-10	16	1.275061273037E+02	4.67E-19	8.73D-09	5.35D-11
17	1.275000000000E+02	5.81E-19	9.74D-09	5.97D-11	18	1.274877439199E+02	3.92E-19	5.65D-09	6.93D-11
19	1.250000000000E+02	-1.00E+00	4.32D-11	4.32D-11	20	1.250000000000E+02	-1.00E+00	4.62D-11	4.62D-11
21	1.960972814430E-01	1.46E-20	4.12D-10	3.54D-11	22	1.103047208117E-01	1.46E-20	4.89D-10	3.00D-11
23	4.902432036076E-02	2.59E-20	8.39D-10	3.09D-11	24	1.225608009019E-02	4.96E-20	2.01D-09	2.47D-11
25	2.510526491680E-21	-1.00E+00	2.90D-11	2.90D-11	26	6.799816822252E-22	-1.00E+00	2.33D-11	2.33D-11
27	1.194522438737E-21	-1.00E+00	1.77D-11	1.77D-11	28	1.047267910160E-21	-1.00E+00	2.89D-11	2.89D-11
29	-1.275061273037E+02	1.45E-22	1.07D-12	1.36D-10	30	-5.100245092150E+02	1.08E-21	1.68D-12	6.43D-10
31	-1.147555145734E+03	1.81E-21	1.69D-12	1.08D-09	32	-2.040098036860E+03	5.22E-22	7.65D-13	6.83D-10

1 2.040098036859993647692626298D+03 2 2.039999999999999999999982189D+03 3 2.039803902718556966005661683D+03

4 1.959999999999999999999999381D+03 5 1.999999999999999999999999166D+03 6 1.147555145733746426827102536D+03

7 1.147499999999999999999997747D+03 8 1.147389695279188293378176831D+03 9 1.124999999999999999999999776D+03

10 1.124999999999999999999999691D+03 11 5.100245092149984119231553011D+02 12 5.100000000000000000000002562D+02

13 5.099509756796392415014121895D+02 14 4.999999999999999999999999133D+02 15 4.999999999999999999999999163D+02

16 1.275061273037496029807891935D+02 17 1.274999999999999999999979151D+02 18 1.274877439199098103753547777D+02

19 1.250000000000000000000000359D+02 20 1.249999999999999999999999973D+02 21 1.960972814430339943546908091D-01

22 1.103047208117066218243810830-01 23 4.902432036075849858550193550D-02 24 1.225608009018962464590062071D-02

25 2.510526491680026233399059313D-21 26 6.799816822252175679543366877D-22 27 1.194522438737278142218712682D-21

28 1.047267910160431992753048862D-21 29 -1.275061273037496029807891341D+02 30 -5.100245092149984119231562715D+02

31 -1.147555145733746426827102084D+03 32 -2.040098036859993647692627267D+03

EIGENVECTOR NO. 8

-.03116033080 .01558016543 -.01558016537 .03116033079 -.15734439442 .15734439442 .31468878877 -.31468878877

EIGENVECTOR NO. 16

.31622396692 .15811198345 .15811198346 .31622396692 -.00077504011 -.00077504012 .00155008024 .00155008024

EIGENVECTOR NO. 24

-.31468878869 .15734439434 -.15734439434 .31468878869 .01558016535 -.01558016544 -.03116033081 .03116033077

EIGENVECTOR NO. 32

.00155008024 .00077504012 .00077504012 .00155008024 .15811198346 .15811198346 -.31622396691 -.31622396691

EIGEN SYSTEM CALCULATION TOOK 2.456 SECONDS

PREVIOUS KRONECKER PRODUCT + IDENTITY

	EIGENVALUE	VALUE ERROR	VECTOR ERROR	RESIDUAL LENGTH		EIGENVALUE	VALUE ERROR	VECTOR ERROR	RESIDUAL LENGTH
1	2.041098036860E+03	8.22E-18	9.16D-09	8.98D-10	2	2.041000000000E+03	8.21E-18	9.15D-09	8.97D-10
3	2.040803902719E+03	2.27E-18	3.40D-09	6.67D-10	4	2.001000000000E+03	-1.00E+00	8.29D-10	8.29D-10
5	2.001000000000E+03	-1.00E+00	8.82D-10	8.82D-10	6	1.148555145734E+03	4.02E-18	8.54D-09	4.71D-10
7	1.148500000000E+03	4.10E-18	8.63D-09	4.76D-10	8	1.148389695279E+03	2.23E-18	4.50D-09	4.96D-10
9	1.126000000000E+03	-1.00E+00	4.19D-10	4.19D-10	10	1.126000000000E+03	-1.00E+00	4.60D-10	4.60D-10
11	5.110245092150E+02	1.12E-18	6.77D-09	1.66D-10	12	5.110000000000E+02	1.17E-18	6.91D-09	1.69D-10
13	5.109509756796E+02	6.01E-19	3.50D-09	1.72D-10	14	5.010000000000E+02	-1.00E+00	1.51D-10	1.51D-10
15	5.010000000000E+02	-1.00E+00	1.63D-10	1.63D-10	16	1.285061273037E+02	5.09E-19	9.12D-09	5.59D-11
17	1.285000000000E+02	6.74E-19	1.05D-08	6.43D-11	18	1.284877439199E+02	3.28E-19	5.17D-09	6.34D-11
19	1.260000000000E+02	-1.00E+00	3.59D-11	3.59D-11	20	1.260000000000E+02	-1.00E+00	4.18D-11	4.18D-11
21	1.196097281443E+00	6.22E-21	2.69D-10	2.31D-11	22	1.110304720812E+00	1.12E-20	4.27D-10	2.62D-11
23	1.049024320361E+00	1.37E-20	6.10D-10	2.24D-11	24	1.012256080090E+00	4.33E-20	1.88D-09	2.30D-11
25	1.000000000000E+00	-1.00E+00	4.53D-11	4.53D-11	26	1.000000000000E+00	-1.00E+00	2.24D-11	2.24D-11
27	1.000000000000E+00	-1.00E+00	1.72D-11	1.72D-11	28	1.000000000000E+00	-1.00E+00	2.21D-11	2.21D-11
29	-1.265061273037E+02	2.10E-22	1.28D-12	1.63D-10	30	-5.090245092150E+02	7.23E-22	1.37D-12	5.26D-10
31	-1.146555145734E+03	1.32E-21	1.44D-12	9.16D-10	32	-2.039098036860E+03	3.40E-22	6.17D-13	5.51D-10

1 2.041098036859993647692626891D+03

2 2.040999999999999999999947538D+03

3 2.040803902718556966005695905D+03

4 2.000999999999999999999995506D+03

5 2.000999999999999999999999447D+03

6 1.148555145733746426827102341D+03

7 1.148499999999999999999998883D+03

8 1.148389695279188293378175032D+03

9 1.125999999999999999999997480D+03

10 1.125999999999999999999999746D+03

11 5.110245092149984119231562012D+02

12 5.110000000000000000000004565D+02

13 5.109509756796392415014111781D+02

14 5.009999999999999999999999446D+02

15 5.009999999999999999999999311D+02

16 1.285061273037496029807890797D+02

17 1.284999999999999999999997196D+02

18 1.284877439199098103753531121D+02

19 1.259599999999999999999999950D+02

20 1.260000000000000000000000327D+02

21 1.196097281443033994354956657D+00

22 1.110304720811706621824686257D+00

23 1.049024320360758498584185272D+00

24 1.012256080090189624648237608D+00

25 1.000000000000000000000061240D+00

26 1.000000000000000000002193051D+00

27 1.000000000000000000000183512D+00

28 1.000000000000000000001439113D+00

29 -1.265061273037496029807891067D+02

30 -5.090245092149984119231564482D+02

31 -1.146555145733746426827102249D+03

32 -2.039098036859993647692627360D+03

EIGENVECTOR NO. 8

-.03116033079_ .01558016540_ -.01558016540_ .03116033081_ -.15734439439_ .15734439439_ .31468878878_ -.31468878878_

EIGENVECTOR NO. 16

-.31622396690 -.15811198349 -.15811198349 -.31622396690 .00077504015 .00077504009 -.00155008022 -.00155008026

EIGENVECTOR NO. 24

-.31468878869 .15734439432 -.15734439432 .31468878869 .01558016525 -.01558016554 -.03116033086 .03116033071

EIGENVECTOR NO. 32

-.00155008024_ -.00077504012_ -.00077504012_ -.00155008024_ -.15811198346_ -.15811198346_ .31622396691_ .31622396691_

EIGEN SYSTEM CALCULATION TOOK 2.434 SECONDS

THE KRONECKER PRODUCT B X R8 TIMES 2**-11

	EIGENVALUE	VALUE ERROR	VECTOR ERROR	RESIDUAL LENGTH		EIGENVALUE	VALUE ERROR	VECTOR ERROR	RESIDUAL LENGTH
1	9.961416195605E-01	5.14E-21	1.04D-08	4.96D-13	2	9.960937500000E-01	3.91E-21	9.04D-09	4.33D-13
3	9.959979993743E-01	1.39E-21	3.81D-09	3.64D-13	4	9.765625000000E-01	-1.00E+00	4.25D-13	4.25D-13
5	9.765625000000E-01	-1.00E+00	5.02D-13	5.02D-13	6	5.603296610028E-01	2.69E-21	1.00D-08	2.69D-13
7	5.603027343750E-01	2.59E-21	9.80D-09	2.64D-13	8	5.602488746480E-01	1.15E-21	4.62D-09	2.49D-13
9	5.493164062500E-01	-1.00E+00	1.93D-13	1.93D-13	10	5.493164062500E-01	-1.00E+00	2.50D-13	2.50D-13
11	2.490354048901E-01	8.23E-22	8.29D-09	9.92D-14	12	2.490234375000E-01	8.13E-22	8.24D-09	9.87D-14
13	2.489994998436E-01	3.56E-22	3.86D-09	9.23D-14	14	2.441406250000E-01	-1.00E+00	8.33D-14	8.33D-14
15	2.441406250000E-01	-1.00E+00	8.89D-14	8.89D-14	16	6.225885122253E-02	2.28E-22	8.73D-09	2.61D-14
17	6.225585937500E-02	2.84E-22	9.74D-09	2.91D-14	18	6.224987496089E-02	1.91E-22	5.65D-09	3.38D-14
19	6.103515625000E-02	-1.00E+00	2.11D-14	2.11D-14	20	6.103515625000E-02	-1.00E+00	2.26D-14	2.26D-14
21	9.575062570461E-05	7.12E-24	4.12D-10	1.73D-14	22	5.385972695884E-05	7.15E-24	4.89D-10	1.46D-14
23	2.393765642615E-05	1.26E-23	8.39D-10	1.51D-14	24	5.984414106538E-06	2.42E-23	2.01D-09	1.20D-14
25	1.225843013516E-24	-1.00E+00	1.41D-14	1.41D-14	26	3.320223057740E-25	-1.00E+00	1.14D-14	1.14D-14
27	5.832629095397E-25	-1.00E+00	8.63D-15	8.63D-15	28	5.113612842580E-25	-1.00E+00	1.41D-14	1.41D-14
29	-6.225885122253E-02	7.08E-26	1.07D-12	6.64D-14	30	-2.490354048901E-01	5.27E-25	1.68D-12	3.14D-13
31	-5.603296610028E-01	8.85E-25	1.69D-12	5.25D-13	32	-9.961416195605E-01	2.55E-25	7.65D-13	3.33D-13

1 9.96141619560543773287415184 8D-01

2 9.96093749999999999999991303 1D-01

3 9.95997999374295393557451993 9D-01

4 9.76562499999999999999999697 6D-01

5 9.76562499999999999999999593 0D-01

6 5.60329661002805872474171160 4D-01

7 5.60302734374999999999998899 8D-01

8 5.60248874648041158876062905 9D-01

9 5.49316406249999999999999890 6D-01

10 5.49316406249999999999999848 9D-01

11 2.49035404890135943321853174 4D-01

12 2.49023437500000000000000125 1D-01

13 2.48999499843573848389361420 7D-01

14 2.44140624999999999999999577 0D-01

15 2.44140624999999999999999591 0D-01

16 6.22588512225333858304634734 0D-02

17 6.22558593749999999999898199 0D-02

18 6.22498749608934620973412000 7D-02

19 6.10351562500000000000000175 4D-02

20 6.10351562499999999999999869 0D-02

21 9.57506257046064425560013716 4D-05

22 5.38597269588411239377139200 5D-05

23 2.39376564261516106374521169 4D-05

24 5.98441410653790265913116245 7D-06

25 1.22584301351563780927688443 0D-24

26 3.32022305774032015602703460 8D-25

27 5.83262909539686592880230801 6D-25

28 5.11361284258023433961449639 6D-25

29 -6.22588512225339858304634443 7D-02

30 -2.49035404890135943321853648 2D-01

31 -5.60329661002805872474170339 4D-01

32 -9.96141619560543773287415657 6D-01

EIGENVECTOR NO. 8

-.03116033080 .01558016543 -.01558016537 .03116033079 -.15734439442 .15734439442 .31468878877 -.31468878877

EIGENVECTOR NO. 16

.31622396692 .15811198345 .15811198346 .31622396692 -.00077504011 -.00077504012 .00155008024 .00155008024

EIGENVECTOR NO. 24

-.31468678869 .15734439434 -.15734439434 .31468878869 .01558016535 -.01558016544 -.03116033081 .03116033077

EIGENVECTOR NO. 32

.00155008024 .00077504012 .00077504012 .00155008024 .15811198346 .15811198346 -.31622396691 -.31622396691

EIGEN SYSTEM CALCULATION TOOK 2.462 SECONDS

PREVIOUS KRONECKER PRODUCT + IDENTITY

	EIGENVALUE	VALUE ERROR	VECTOR ERROR	RESIDUAL LENGTH		EIGENVALUE	VALUE ERROR	VECTOR ERROR	RESIDUAL LENGTH
1	1.9961416195 61E+00	9.91E-21	1.44D-08	6.89D-13	2	1.9960937500 00E+00	6.31E-21	1.15D-08	5.49D-13
3	1.9959979993 74E+00	2.91E-21	5.51D-09	5.26D-13	4	1.9765625000 00E+00	-1.00E+00	6.00D-13	6.00D-13
5	1.9765625000 00E+00	-1.00E+00	6.27D-13	6.27D-13	6	1.5603296610 03E+00	7.82E-21	1.70D-08	4.59D-13
7	1.5603027343 75E+00	5.34E-21	1.41D-08	3.79D-13	8	1.5602488746 48E+00	2.26E-21	6.48D-09	3.49D-13
9	1.5493164062 50E+00	-1.00E+00	3.38D-13	3.38D-13	10	1.5493164062 50E+00	-1.00E+00	4.42D-13	4.42D-13
11	1.2490354048 90E+00	7.51E-21	2.50D-08	3.00D-13	12	1.2490234375 00E+00	5.69E-21	2.18D-08	2.61D-13
13	1.2489994998 44E+00	2.30E-21	9.80D-09	2.35D-13	14	1.2441406250 00E+00	-1.00E+00	2.15D-13	2.15D-13
15	1.2441406250 00E+00	-1.00E+00	3.01D-13	3.01D-13	16	1.0622588512 23E+00	1.64E-20	7.84D-08	2.35D-13
17	1.0622558593 75E+00	1.02E-20	5.84D-08	1.75D-13	18	1.0622498749 61E+00	4.24E-21	2.66D-08	1.59D-13
19	1.0610351562 50E+00	-1.00E+00	1.19D-13	1.19D-13	20	1.0610351562 50E+00	-1.00E+00	2.01D-13	2.01D-13
21	1.0000957506 26E+00	4.36E-22	3.23D-09	1.35D-13	22	1.0000538597 27E+00	8.34E-22	5.28D-09	1.58D-13
23	1.0000239376 56E+00	1.37E-21	8.75D-09	1.57D-13	24	1.0000059844 14E+00	4.94E-21	2.87D-08	1.72D-13
25	1.0000000000 00E+00	-1.00E+00	1.01D-13	1.01D-13	26	1.0000000000 00E+00	-1.00E+00	1.74D-13	1.74D-13
27	1.0000000000 00E+00	-1.00E+00	2.02D-13	2.02D-13	28	1.0000000000 00E+00	-1.00E+00	2.14D-13	2.14D-13
29	9.3774114877 75E-01	1.15E-25	1.36D-12	8.45D-14	30	7.5096459510 99E-01	1.06E-25	7.55D-13	1.41D-13
31	4.3967033899 72E-01	2.64E-25	9.55D-13	2.97D-13	32	3.8583804394 56E-03	8.37E-26	4.38D-13	1.91D-13

1 1.9961416195605437732874132560D+00

2 1.9960937499999999999999995710D+00

3 1.9959979993742953935574453760D+00

4 1.9765624999999999999999960800D+00

5 1.9765624999999999999999955900D+00

6 1.5603296610028058724741698970D+00

7 1.5603027343749999999998934600D+00

8 1.5602488746480411588760734460D+00

9 1.5493164062499999999999984000D+00

10 1.5493164062499999999999975700D+00

11 1.2490354048901359433218510650D+00

12 1.2490234374999999999999901580D+00

13 1.2489994998435738483893731830D+00

14 1.2441406249999999999999992900D+00

15 1.2441406249999999999999986300D+00

16 1.0622588512225339858304632230D+00

17 1.0622558593749999999999594300D+00

18 1.0622498749608934620973805320D+00

19 1.0610351562500000000000001060D+00

20 1.0610351562500000000000001070D+00

21 1.0000957506257046064425429820D+00

22 1.0000538597269588411239423880D+00

23 1.0000239376564261516106244440D+00

24 1.0000059844141065379026127900D+00

25 1.0000000000000000000000214010D+00

26 1.0000000000000000000000076740D+00

27 1.0000000000000000000000030680D+00

28 1.0000000000000000000001059400D+00

29 9.3774114877746601416953647700D-01

30 7.5096459510986405667814617530D-01

31 4.3967033899719412752582884350D-01

32 3.8583804394562267125842685530D-03

EIGENVECTOR NO. 8

.03116033081 -.01558016552 .01558016528 -.03116033079 .15734439449 -.15734439449 -.31468878873 .31468878873

EIGENVECTOR NO. 16

.31622396690 .15811198347 .15811198347 .31622396691 -.00077504014 -.00077504010 .00155008025 .00155008023

EIGENVECTOR NO. 24

.31468878889 -.15734439472 .15734439472 -.31468878889 -.01558016693 .01558016388 .03116033006 -.03116033158

EIGENVECTOR NO. 32

-.00155008024 -.00077504012 -.00077504012 -.00155008024 -.15811198346 -.15811198346 .31622396691 .31622396691

EIGEN SYSTEM CALCULATION TOOK 2.424 SECONDS

8 BY 8 HADAMARD PRODUCT WITH EIGENVALUES 1.0, 1.0+2.**(-K) , K=38 TO 44

I	J	A(I,J)	I	J	A(I,J)	I	J	A(I,J)	I	J	A(I,J)	I	J	A(I,J)	I	J	A(I,J)	I	J	A(I,J)
1	1	1.000E+00	1	2	3.055E-13	1	3	5.471E-13	1	4	1.776E-13	1	5	8.029E-13	1	6	2.629E-13	1	7	4.761E-13
1	8	1.634E-13	2	2	1.000E+00	2	3	1.776E-13	2	4	5.471E-13	2	5	2.629E-13	2	6	8.029E-13	2	7	1.634E-13
2	8	4.761E-13	3	3	1.000E+00	3	4	3.055E-13	3	5	4.761E-13	3	6	1.634E-13	3	7	8.029E-13	3	8	2.629E-13
4	4	1.000E+00	4	5	1.634E-13	4	6	4.761E-13	4	7	2.629E-13	4	8	8.029E-13	5	5	1.000E+00	5	6	3.055E-13
5	7	5.471E-13	5	8	1.776E-13	6	6	1.000E+00	6	7	1.776E-13	6	8	5.471E-13	7	7	1.000E+00	7	8	3.055E-13
8	8	1.000E+00																		

	EIGENVALUE	VALUE ERROR	VECTOR ERROR	RESIDUAL LENGTH		EIGENVALUE	VALUE ERROR	VECTOR ERROR	RESIDUAL LENGTH
1	1.000000000004E+00	2.62E-15	3.96D-02	6.62D-14	2	1.000000000002E+00	8.95E-15	1.08D-01	8.25D-14
3	1.000000000001E+00	1.51E-14	2.27D-01	6.54D-14	4	1.000000000000E+00	1.75E-13	1.81D+00	9.67D-14
5	1.000000000000E+00	-1.00E+00	8.67D-14	8.67D-14	6	1.000000000000E+00	-1.00E+00	7.59D-14	7.59D-14
7	1.000000000000E+00	-1.00E+00	1.13D-13	1.13D-13	8	1.000000000000E+00	-1.00E+00	7.22D-14	7.22D-14

1 1.00000000000036378655718443890+00 2 1.00000000000018186406410513710+00 3 1.00000000000090915988331009 2D+00

4 1.00000000000045381698381193 0D+00 5 1.00000000000021703616395393 4D+00 6 1.00000000000011245121373088 1D+00

7 1.00000000000006205743204792 5D+00 8 1.00000000000000510630557202 6D+00

EIGENVECTOR NO. 1
-.35187470152 -.35348363843 -.35156967021 -.35559229298 -.35124933787 -.35676462594 -.35035161593 -.35825327496

EIGENVECTOR NO. 2
.35476267545 -.34463231480 .35588036826 -.35821163833 .35261386114 -.34027085947 .36064810770 -.36085696182

EIGENVECTOR NO. 3
.34334931916 .36285830964 -.33896973303 -.34896813823 .34556956410 .36346253367 -.36124357821 -.36306387119

EIGENVECTOR NO. 4
.37647510702 -.37164756723 -.33777308215 .30509934017 .35116118347 -.33312187382 -.37043559577 .37611358253

EIGENVECTOR NO. 5
.27354733003 .23701606394 .38387002433 .47750286817 -.26461202441 -.26353078902 -.38828835941 -.45099247078

EIGENVECTOR NO. 6
.34641879028 -.19990765410 .40220348720 -.43191148369 -.39191306059 .23816548205 -.35931811724 .39024046873

EIGENVECTOR NO. 7
-.27517281591 -.55498759644 .23196389938 .26651432411 .26410038114 .55009570342 -.22490901635 .26171204355

EIGENVECTOR NO. 8
.46980159455 -.28694755779 -.39727283350 .21179602047 -.46496288597 .29168988866 .38601270594 -.20972605153

GRAM MATRIX

1	1.0E+00	2.0E-14	0.	0.	8.9E-16	-3.6E-15	4.4E-16	1.3E-15
2	1.0E+00	5.3E-15	-8.9E-16	5.3E-15	4.4E-15	3.1E-15	2.2E-15	
3	1.0E+00	8.9E-16	4.4E-15	-2.7E-15	2.7E-15	-4.4E-16		
4	1.0E+00	3.6E-15	1.1E-14	-1.3E-15	0.			
5	1.0E+00	-3.6E-15	2.2E-14	8.9E-16				
6	1.0E+00	7.1E-15	-4.0E-15					
7	1.0E+00	-8.2E-15						
8	1.0E+00							

EIGEN SYSTEM CALCULATION TOOK .056 SECONDS

16 BY 16 HADAMARD PRODUCT WITH EIGENVALUES 1.0, 1.0 + 2.**(-K),K =24 TO 38

	EIGENVALUE	VALUE ERROR	VECTOR ERROR	RESIDUAL LENGTH		EIGENVALUE	VALUE ERROR	VECTOR ERROR	RESIDUAL LENGTH
1	1.0000000596046E+00	5.22E-19	4.18D-06	1.25D-13	2	1.0000000298023E+00	5.94E-19	6.83D-06	1.02D-13
3	1.0000000149011E+00	1.80E-18	1.55D-05	1.16D-13	4	1.0000000074506E+00	3.61E-18	3.11D-05	1.16D-13
5	1.0000000037253E+00	7.18E-18	6.21D-05	1.16D-13	6	1.0000000018626E+00	1.45E-17	1.25D-04	1.16D-13
7	1.0000000009313E+00	3.34E-17	2.68D-04	1.25D-13	8	1.0000000004657E+00	6.66E-17	5.35D-04	1.24D-13
9	1.0000000002328E+00	1.19E-16	1.01D-03	1.17D-13	10	1.0000000001164E+00	3.81E-16	2.56D-03	1.48D-13
11	1.0000000000582E+00	6.20E-16	4.64D-03	1.34D-13	12	1.0000000000291E+00	2.13E-15	1.22D-02	1.74D-13
13	1.0000000000015E+00	3.57E-15	2.26D-02	1.58D-13	14	1.0000000000007E+00	6.15E-15	4.28D-02	1.44D-13
15	1.0000000000004E+00	5.43E-15	4.02D-02	1.35D-13	16	1.0000000000000E+00	3.75E-15	3.33D-02	1.13D-13

1 1.0000000596046447753732922550+00 2 1.0000000298023223876802903310+00 3 1.0000000149011611938213245260+00

4 1.0000000074505805968223064180+00 5 1.0000000037252902983557861740+00 6 1.0000000018626451490316764200+00

7 1.0000000009313225732109236290+00 8 1.0000000004656612865205161870+00 9 1.0000000002328306400518974070+00

10 1.0000000001164153124578152310+00 11 1.0000000000582076439285893810+00 12 1.0000000000291037104056447180+00

13 1.0000000000145518194883253710+00 14 1.0000000000072760179054836000+00 15 1.0000000000036378452087666570+00

16 1.0000000000000003210715192840+00

EIGENVECTOR NO. 1
-.24999994027 -.25000008320 -.24999998902 -.24999990545 -.24999999445 -.24999997286 -.24999996414 -.25000005315

EIGENVECTOR NO. 2
-.24999994011 .24999956374 -.24999999602 .24999996798 -.25000008757 .24999967387 -.25000003997 .25000013798

EIGENVECTOR NO. 3
.24999953042 .25000011258 -.24999979238 -.24999948328 .24999949967 .25000021392 -.25000032565 -.25000005527

EIGENVECTOR NO. 4
.25000003985 -.24999952463 -.24999987525 .24999861373 .25000075423 -.25000004890 -.25000032012 .24999985048

EIGENVECTOR NO. 5
.24999895945 .25000065059 .24999818884 .25000143246 -.24999955142 -.24999981964 -.25000007171 -.24999745204

EIGENVECTOR NO. 6
-.25000133029 .24999782513 -.24999903281 .25000119234 .25000364038 -.24999519868 .25000229876 -.24999928243

EIGENVECTOR NO. 7
.24998855396 .25001091363 -.24999231262 -.25001543002 -.24998956859 -.25001461863 .25000326876 .24999036375

EIGENVECTOR NO. 8
.25001490960 -.25000542746 -.24999339273 .24998120107 -.24999467854 .24999314766 .25001949545 -.24996820661

EIGEN SYSTEM CALCULATION TOOK .346 SECONDS

W21+ SEC 6.4

I	J	A(I,J)	I	J	A(I,J)	I	J	A(I,J)	I	J	A(I,J)	I	J	A(I,J)	I	J	A(I,J)	I	J	A(I,J)
1	1	1.000E+01	1	2	1.000E+00	2	2	9.000E+00	2	3	1.000E+00	3	3	8.000E+00	3	4	1.000E+00	4	4	7.000E+00
4	5	1.000E+00	5	5	6.000E+00	5	6	1.000E+00	6	6	5.000E+00	6	7	1.000E+00	7	7	4.000E+00	7	8	1.000E+00
8	8	3.000E+00	8	9	1.000E+00	9	9	2.000E+00	9	10	1.000E+00	10	10	1.000E+00	10	11	1.000E+00	11	12	1.000E+00
12	12	1.000E+00	12	13	1.000E+00	13	13	2.000E+00	13	14	1.000E+00	14	14	3.000E+00	14	15	1.000E+00	15	15	4.000E+00
15	16	1.000E+00	16	16	5.000E+00	16	17	1.000E+00	17	17	6.000E+00	17	18	1.000E+00	18	18	7.000E+00	18	19	1.000E+00
19	19	8.000E+00	19	20	1.000E+00	20	20	9.000E+00	20	21	1.000E+00	21	21	1.000E+01						

	EIGENVALUE	VALUE ERROR	VECTOR ERROR	RESIDUAL LENGTH		EIGENVALUE	VALUE ERROR	VECTOR ERROR	RESIDUAL LENGTH
1	1.074619418290E+01	-1.00E+00	1.11D-13	1.11D-13	2	1.074619418290E+01	-1.00E+00	1.28D-13	1.28D-13
3	9.210678647361E+00	1.23E-16	1.48D-03	8.31D-14	4	9.210678647305E+00	1.71E-16	1.74D-03	9.81D-14
5	8.038941122829E+00	2.78E-18	1.99D-05	1.40D-13	6	8.038941115814E+00	1.87E-18	1.63D-05	1.14D-13
7	7.003952209529E+00	8.44E-20	4.53D-07	1.86D-13	8	7.003951798616E+00	9.14E-20	4.72D-07	1.94D-13
9	6.000234031584E+00	5.38E-21	1.80D-08	2.98D-13	10	6.000217522257E+00	5.95E-21	1.90D-08	3.13D-13
11	5.000244425002E+00	4.25E-22	9.59D-10	4.43D-13	12	4.999782477743E+00	3.51E-22	8.72D-10	4.03D-13
13	4.004354023441E+00	6.56E-23	8.90D-11	7.39D-13	14	3.996048201384E+00	6.30E-23	8.71D-11	7.23D-13
15	3.043099292579E+00	1.37E-23	1.29D-11	1.06D-12	16	2.961058884186E+00	1.39E-23	1.30D-11	1.07D-12
17	2.130209219362E+00	4.65E-24	3.69D-12	1.26D-12	18	1.789321352695E+00	5.70E-24	4.09D-12	1.39D-12
19	9.475343675293E-01	5.21E-24	2.74D-12	1.90D-12	20	2.538058170967E-01	6.70E-24	3.11D-12	2.16D-12
21	-1.125441522120E+00	3.39E-24	1.57D-12	2.16D-12					

1 1.0746194182903357513429992190+01 2 1.0746194182903357750717378270+01 3 9.2106786473613320536760919970+00

4 9.2106786473049186481968798470+00 5 8.0389411228290232359051641020+00 6 8.0389411158142733087434060580+00

7 7.0039522095286756738128988650+00 8 7.0039517986163749692715866630+00 9 6.0002340315841670166171874030+00

10 6.0002175222570581400137905160+00 11 5.0002444250019130080653914520+00 12 4.9997824777429018599865223710+00

13 4.0043540234408567350974690880+00 14 3.9960482013836250307295016860+00 15 3.0430992925788237393316432530+00

16 2.9610588841857266916133742310+00 17 2.1302092193625059944850954420+00 18 1.7893213526950814060446742850+00

19 9.4753436752929327885064110970-01 20 2.5380581709667816771009218500-01 21 -1.1254415221199842222987733470+00

EIGEN SYSTEM CALCULATION TOOK .714 SECONDS

I	J	A(I,J)	I	J	A(I,J)	I	J	A(I,J)	I	J	A(I,J)	I	J	A(I,J)	I	J	A(I,J)	I	J	A(I,J)
1	1	1.000E+01	1	2	1.000E+00	2	2	9.000E+00	2	3	1.000E+00	3	3	8.000E+00	3	4	1.000E+00	4	4	7.000E+00
4	5	1.000E+00	5	5	6.000E+00	5	6	1.000E+00	6	6	5.000E+00	6	7	1.000E+00	7	7	4.000E+00	7	8	1.000E+00
8	8	3.000E+00	8	9	1.000E+00	9	9	2.000E+00	9	10	1.000E+00	10	10	1.000E+00	10	11	1.000E+00	11	12	1.000E+00
12	12	-1.000E+00	12	13	1.000E+00	13	13	-2.000E+00	13	14	1.000E+00	14	14	-3.000E+00	14	15	1.000E+00	15	15	-4.000E+00
15	16	1.000E+00	16	16	-5.000E+00	16	17	1.000E+00	17	17	-6.000E+00	17	18	1.000E+00	18	18	-7.000E+00	18	19	1.000E+00
19	19	-8.000E+00	19	20	1.000E+00	20	20	-9.000E+00	20	21	1.000E+00	21	21	-1.000E+01						

	EIGENVALUE	VALUE ERROR	VECTOR ERROR	RESIDUAL LENGTH		EIGENVALUE	VALUE ERROR	VECTOR ERROR	RESIDUAL LENGTH
1	1.074619418290E+01	7.23E-27	6.86D-14	1.05D-13	2	9.210678647333E+00	2.14E-27	4.27D-14	5.00D-14
3	8.038941119306E+00	2.27E-27	4.68D-14	4.85D-14	4	7.003952002665E+00	2.10E-26	1.45D-13	1.45D-13
5	6.000225680185E+00	5.22E-26	2.26D-13	2.26D-13	6	5.000008158673E+00	9.06E-26	3.01D-13	3.01D-13
7	4.000000205070E+00	1.51E-25	3.88D-13	3.88D-13	8	3.000000003808E+00	4.05E-25	6.30D-13	6.30D-13
9	2.000000000054E+00	8.70E-25	9.33D-13	9.33D-13	10	1.000000000001E+00	1.50E-24	1.22D-12	1.22D-12
11	6.890201209496E-25	2.53E-24	1.59D-12	1.59D-12	12	-1.000000000001E+00	2.64E-24	1.62D-12	1.62D-12
13	-2.000000000054E+00	3.21E-24	1.79D-12	1.79D-12	14	-3.000000003806E+00	4.57E-24	2.14D-12	2.14D-12
15	-4.000000205070E+00	7.71E-24	2.78D-12	2.78D-12	16	-5.000008158673E+00	1.07E-23	3.26D-12	3.26D-12
17	-6.000225680185E+00	1.29E-23	3.59D-12	3.59D-12	18	-7.003952002665E+00	1.78E-23	4.21D-12	4.22D-12
19	-8.038941119306E+00	2.10E-23	4.51D-12	4.67D-12	20	-9.210678647333E+00	1.86E-23	3.99D-12	4.67D-12
21	-1.074619418290E+01	5.46E-24	1.89D-12	2.89D-12					

1 1.0746194182903357570586883960+01 2 9.2106786473330464883276996300+00 3 8.0389411193064408897674058980+00

4 7.0039520026653613283661145940+00 5 6.0002256801851703441252727840+00 6 5.0000081586729450104640503870+00

7 4.0000002050704378003186423100+00 8 3.0000000038081268835353456430+00 9 2.0000000000544881077033794980+00

10 1.0000000000006197453800191220+00 11 6.8902012094958866202998391360-25 12 -1.0000000000006197453800177750+00

13 -2.0000000000544881077033782680+00 14 -3.0000000038081268835353443600+00 15 -4.0000002050704378003186405580+00

16 -5.0000081586729450104640478510+00 17 -6.0002256801851703441252699980+00 18 -7.0039520026653613283661111710+00

19 -8.0389411193064408897674023750+00 20 -9.2106786473330464883276965310+00 21 -1.0746194182903357570586882440+01

EIGEN SYSTEM CALCULATION TOOK .726 SECONDS

THE MATRIX R8 WITH RANDOM PURTERBATIONS ON THE LAST TWO BITS OF EACH ELEMENT

I	J	A(I,J)	I	J	A(I,J)	I	J	A(I,J)	I	J	A(I,J)	I	J	A(I,J)	I	J	A(I,J)	I	J	A(I,J)
1	1	6.110E+02	1	2	1.960E+02	1	3	-1.920E+02	1	4	4.070E+02	1	5	-8.000E+00	1	6	-5.200E+01	1	7	-4.900E+01
1	8	2.900E+01	2	2	8.990E+02	2	3	1.130E+02	2	4	-1.920E+02	2	5	-7.100E+01	2	6	-4.300E+01	2	7	-8.000E+00
2	8	-4.400E+01	3	3	8.990E+02	3	4	1.960E+02	3	5	6.100E+01	3	6	4.900E+01	3	7	8.000E+00	3	8	5.200E+01
4	4	6.110E+02	4	5	8.000E+00	4	6	4.400E+01	4	7	5.900E+01	4	8	-2.300E+01	5	5	4.110E+02	5	6	-5.990E+02
5	7	2.080E+02	5	8	2.080E+02	6	6	4.110E+02	6	7	2.080E+02	6	8	2.080E+02	7	7	9.900E+01	7	8	-9.110E+02
8	8	9.900E+01																		

	EIGENVALUE	VALUE ERROR	VECTOR ERROR	RESIDUAL LENGTH		EIGENVALUE	VALUE ERROR	VECTOR ERROR	RESIDUAL LENGTH
1	1.020049018430E+03	6.02E-20	1.11D-09	5.43D-11	2	1.020000000000E+03	3.63E-20	8.61D-10	4.22D-11
3	1.019901951359E+03	1.28E-20	3.62D-10	3.55D-11	4	1.000000000000E+03	-1.00E+00	3.33D-11	3.33D-11
5	1.000000000000E+03	-1.00E+00	5.22D-11	5.22D-11	6	9.804864072211E-02	1.49E-20	3.90D-10	3.82D-11
7	-7.339622242113E-13	3.29E-20	5.79D-10	5.68D-11	8	-1.020049018430E+03	6.66E-24	8.21D-14	6.38D-11

1 1.020049018430002007959563330D+03 2 1.020000000000004865796653261D+03 3 1.019901951359282897094886675D+03

4 1.000000000000004666978094841D+03 5 1.000000000000004479810120908D+03 6 9.804864072210512597422682421D-02

7 -7.339622242113050923997203486D-13 8 -1.020049018430001189414331787D+03

EIGENVECTOR NO. 1
.63244793382 .31622396693 .31622396692 .63244793382 -.00155008025 -.00155008023 .00310016048 .00310016048

EIGENVECTOR NO. 2
.22360679776 -.44721359549 -.44721359550 .22360679777 .44721359547 -.44721359547 .22360679782 -.22360679782

EIGENVECTOR NO. 3
-.06232066162 .03116033085 -.03116033075 .06232066157 -.31468878883 .31468878883 .62937757754 -.62937757754

EIGENVECTOR NO. 4
-.13883586915 .58872126327 .27767173831 -.29436063163 .41097867757 -.45541432400 .20548933879 -.22770716200

EIGENVECTOR NO. 5
.35878210858 .49538598504 -.71756421716 -.24769299252 -.19772841621 .02444981590 -.09886420811 .01222490795

EIGENVECTOR NO. 6
.62937757756 -.31468878878 .31468878878 -.62937757756 -.03116033082 .03116033078 .06232066159 -.06232066161

EIGENVECTOR NO. 7
-.04472135957 -.08944271909 .08944271909 .04472135957 -.62609903370 -.62609903370 -.31304951685 -.31304951685

EIGENVECTOR NO. 8
.00310016048 .00155008024 .00155008024 .00310016048 .31622396691 .31622396691 -.63244793383 -.63244793383

GRAM MATRIX

1	1.0E+00	1.7E-15	-1.4E-15	4.0E-16	8.8E-15	1.8E-14	-1.5E-15	5.6E-17
2	1.0E+00	-1.8E-14	1.8E-15	-2.2E-15	4.0E-15	0.	8.9E-15	
3	1.0E+00	-5.3E-15	1.5E-15	-6.7E-16	1.6E-14	2.3E-14		
4	1.0E+00	1.4E-16	5.3E-15	1.8E-15	8.9E-15			
5	1.0E+00	-1.6E-15	-3.3E-15	-2.8E-15				
6	1.0E+00	-4.1E-15	2.0E-15					
7	1.0E+00	8.0E-15						
8	1.0E+00							

EIGEN SYSTEM CALCULATION TOOK .056 SECONDS

APPENDIX C

Control Cards for Using SEVER on the CDC 6600

APPENDIX C

Control Cards for Using SEVER on the CDC 6600

SEVER is maintained in a library file for the convenience of the Control Data 6600 users at Sandia Laboratories, Albuquerque, New Mexico. The name of the file is MATHLIB. Questions concerning the availability of SEVER on the Control Data 6600 at Sandia Laboratories, Livermore, California, should be directed to the Numerical Applications Division 8321.

One control card, COLLECT, is required for using the mathematical library file. The COLLECT processor operates on one relocatable binary file and from one to six library files. The library files are searched for routines which contain entry points matching external references in the relocatable binary file. Such routines are added to the relocatable binary file.

A complete typical example follows:

```
JOB CARD
ACCOUNT CARD
FUN,S.
COLLECT,LGO,MATHLIB.
REDUCE.
LGO.
7/8/9 punch in column 1
Program
7/8/9 punch in column 1
Data
6/7/8/9 punch in column 1
```

In the above example, external references in LGO are satisfied, if possible, by selectively adding routines to LGO from MATHLIB. Additional information on the COLLECT processor with examples is contained in UR0004/6600.

DISTRIBUTION:
D. R. Cotter, 100
W. C. Myre, 1210
O. M. Stuetzer, 1220
D. M. Olson, 1510
V. E. Blake, Jr., 1520
T. B. Lane, 1540
R. T. Othmer, 1541
R. L. Peurifoy, Jr., 1550
I. M. Moore, 1610
B. E. Arthur, Jr., 1640
R. R. Prairie, 1643
W. L. Stevens, 1650
C. Winter, 1710
D. R. Morrison, 1720
H. H. Wicke, 1721
R. J. Thompson, 1722
R. G. Clem, 1730
H. D. Sivinski, 1740
A. A. Lieber, 1910
C. B. McCampbell, 2310
K. L. Gillespie, 2320
J. P. Shoup, 2330
M. T. Abegg, 2340
W. C. Kraft, 2440
T. S. Church, 2450
G. H. Roth, 2490
C. M. Tapp, 2610
G. W. Rodgers, 2620
E. G. Franzak, 2630
J. A. Hood, 2650
G. W. Gobeli, 5110
O. E. Jones, 5130
J. E. Schirber, 5150
W. Herrmann, 5160
L. D. Bertholf, 5162
C. H. Karnes, 5165
A. W. Snyder, 5220
C. R. Mehl, 5230
J. H. Renken, 5231
E. H. Beckner, 5240
H. M. Stoller, 5310
J. E. McDonald, Actg., 5320
R. L. Schwoebel, 5330
R. G. Kepler, 5510
W. M. O'Neill, 5520
M. J. Davis, 5530
R. L. Brin, 7250
T. L. Pace, 7260
J. E. Stiegler, 7280
W. T. Moffat, 7290
E. K. Montoya, 7292
M. McWhirter, 7320
M. Cowan, 7340
F. W. Neilson, 7360
S. A. Moore, 7370
J. W. Jones, 7510
W. E. Caldes, 7520
R. J. Dill, 7530
J. W. Pearce, 8120
J. D. Benton, 8121
D. E. Gregson, 8130
L. E. Davies, 8150
R. A. Baroody, 8160
G. E. Brandvold, 8170
C. S. Selvage, 8180
R. H. Meinken, 8310
J. W. Weihe, 8320
J. N. Rogers, 8321
A. G. Schuknecht, 8322
G. W. Anderson, Jr., 8330
J. L. Wirth, 8340
A. N. Blackwell, 8350
C. D. Broyles, 9110
G. E. Hansche, 9120
H. E. Viney, 9130
J. R. Banister, 9150
M. L. Kramm, 9210
J. C. Eckhart, 9220
H. H. Patterson, 9230
R. E. Hepplewhite, 9240
R. C. Maydew, 9320
W. H. Curry, 9322
E. C. Rightley, 9323
H. R. Vaughn, 9325
D. F. McVey, 9328
K. J. Touryan, 9340
W. C. Scrivner, 9400
L. E. Mahuron, 9410
J. E. Wesbrook, 9411
C. E. Katzenberger, 9412
G. D. Horne, Jr., 9413
E. C. Domme, 9414
J. L. Tischhauser, 9420
A. R. Iacoletti, 9421
D. A. Young, 9422
W. R. Gavin, 9422 (50)
D. K. Robbins, 9424
B. T. Fox, 9427
P. Van Delinder, 9427
For: COSMIC (3)
A. J. Arenholz, 9428
A. J. Clark, Jr., 9510
G. J. Hildebrandt, 9520
L. C. Baldwin, 3412
G. C. McDonald, 3416
Attn: M. S. Goldstein
B. R. Allen, 3422
C. H. Sproul, 3428-1 (65)
B. F. Hefley, 8232 (5)