5.4 Eigenvalues and Eigenvectors of an Unsymmetric Matrix

A. Purpose

Compute all eigenvalues and right eigenvectors of a real N × N unsymmetric matrix A. Some or all of the eigenvalues and eigenvectors may be complex.

B. Usage

B.1 Program Prototype, Single Precision

REAL **A**(LDA,≥N) [LDA≥N], **VR**(≥N), **VI**(≥ N), **VEC**(LDA,≥N), WORK(≥N)

INTEGER LDA, N, IFLAG(≥N)

Assign values to A(,), LDA, and N.

CALL SEVVUN(A, LDA, N, VR, VI, VEC, IFLAG, WORK)

Results are returned in VR(), VI(), VEC(,), and IFLAG(1). The contents of A(,), IFLAG(), and WORK() will be modified.

B.2 Argument Definitions

A(,), LDA, N A(,) is [inout], LDA and N are [in]. On entry A(,) must contain the N × N matrix A whose eigenvalues and eigenvectors are to be computed. The integer LDA is the dimension of the first subscript of the arrays A(,) and VEC. Require LDA ≥ N. On return the contents of A(,) will be modified.

VR(), VI() [out] The subroutine will store the J^{th} eigenvalue in VR(J) and VI(J), J = 1, ..., N. The real part is stored in VR(J), and the imaginary part in VI(J). If the J^{th} eigenvalue is real VI(J) will be zero. The eigenvalues will be sorted so that VR(1) ≤ VR(2) ≤ ... ≤ VR(N), and if VR(J) = VR(J+1) for some J then |VI(J)| ≤ |VI($J+1$)|.

Complex eigenvalues will occur in conjugate pairs. Such pairs will be stored in adjacent locations with the eigenvalue having positive imaginary part preceding its conjugate partner.

VEC(,) [out] The eigenvectors will be stored in this array. If the J^{th} eigenvalue is real then the J^{th} eigenvector will be real and will be stored in column J of VEC(,). It will be normalized to have unit Euclidean length.

If the J^{th} and $(J+1)^{st}$ eigenvalues are a complex conjugate pair, then the J^{th} eigenvector will be complex, say $\mathbf{u} + i\mathbf{v}$, and the $(J+1)^{st}$ eigenvector will be its complex conjugate vector, $\mathbf{u} - i\mathbf{v}$. The subroutine will store $\mathbf{u}$ in column J of VEC(,) and will store $\mathbf{v}$ in column $J+1$ of VEC(,). The eigenvector $\mathbf{u} + i\mathbf{v}$ will be normalized to have unit unitary norm and real first component, *i.e.* the first component of $\mathbf{v}$ will be zero.

IFLAG() [out, scratch] The N-array IFLAG() will be used as INTEGER working space. In addition, the first location, IFLAG(1), will be used to pass information back to the user as follows:

= 1 If successful and all eigenvalues are real.

= 2 If successful and some eigenvalues are complex.

See Section E for use of IFLAG(1) in error conditions.

WORK() [scratch] Working space.

B.3 Modifications for Double Precision

Change SEVVUN to DEVVUN, and the REAL type statement to DOUBLE PRECISION.

C. Examples and Remarks

The following unsymmetric matrix A is given on page 84 of [1].

$$A = \begin{bmatrix} 8 & -1 & -5 \\ -4 & 4 & -2 \\ 18 & -5 & -7 \end{bmatrix}.$$

The eigenvalues are 1, $2+4i$, and $2-4i$. The (unnormalized) right eigenvectors are column vectors with the following triples of elements: (1, 2, 1), (1, $1+i$, $1-i$), and (1, $1-i$, $1+i$). This example illustrates the way SEVVUN returns complex eigenvalues and eigenvectors in storage.

The demonstration program DRSEVVUN below applies SEVVUN to compute eigenvalues and eigenvectors for the above matrices. The results are in the file ODSEVVUN. Before the call to SEVVUN, the matrix is saved in order to compute the relative residual matrix D defined as

$$D = (AW - W\Lambda)/\gamma,$$

where A denotes the current test matrix, W is the matrix whose columns are the computed eigenvectors of A, Λ is the diagonal matrix of eigenvalues, and γ is the maximum-row-sum norm of A. The (possibly complex) matrix D is packed into the array D(,) and printed.

D. Functional Description.

Given an N × N real unsymmetric matrix A there exists an N × N nonsingular matrix C such that the matrix

$$U = C^{-1}AC$$

is N × N upper triangular. The matrices C and U may be complex. The diagonal elements of U are called the eigenvalues of A. This set of N numbers is uniquely determined by A although C and U are not unique. Note that λ is an eigenvalue of A if and only if $A - \lambda I$ is singular.

A nonzero vector $\mathbf{w}$ is a right eigenvector of A associated with an eigenvalue λ if

$$A\mathbf{w} = \mathbf{w}\lambda$$

If A has N distinct eigenvalues then it will also have N linearly independent eigenvectors. If the eigenvalues of A are not all distinct then an eigenvalue λ of multiplicity μ may have any number of linearly independent eigenvectors from 1 to μ. If some multiple eigenvalue of A has fewer linearly independent eigenvectors than its multiplicity the matrix is called defective.

If a set of computed eigenvalues returned by SEVVUN are equal or nearly equal, it is not uncommon for the associated computed eigenvectors to be linearly dependent, or nearly so. This subroutine cannot be used to distinguish between defective and nondefective matrices.

The subroutine SEVVUN was developed using the subroutines BALANC, ELMHES, ELTRAN, HQR2, and BALBAK from the EISPACK package of eigenvalue-eigenvector subroutines, [2]. The Fortran subroutines in EISPACK are based directly on the earlier set of Algol procedures described in [3].

Subroutine SEVVUN first calls SEVBH which consists of the two EISPACK subroutines BALANC and ELMHES.

BALANC applies similarity permutations to isolate eigenvalues available by inspection, if any. Then, applies diagonal similarity scaling to balance the size of the matrix elements

$$B = D^{-1}P^{T}APD.$$

ELMHES reduces B to upper Hessenberg form using stabilized elementary transformations

$$H = G^{-1}BG.$$

The remainder of SEVVUN consists of a minor modification of three EISPACK subroutines: ELTRN, HQR2, and BALBAK.

ELTRN computes explicitly the matrix G that was stored in factored form by SEVBH.

HQR2 applies the QR algorithm to H. This is an iterative process which reduces H to a real nearly-upper-triangular matrix R

$$R = Q^{T}HQ.$$

The transformations applied to H are also applied to G forming

$$K = GQ.$$

The matrix R has a mixture of single elements and 2×2 blocks on its diagonal and is otherwise upper triangular.

The eigenvalues of A are the single diagonal elements of R along with the eigenvalues of the 2×2 blocks on the diagonal of R. These latter eigenvalues are computed by direct formulas.

The eigenvectors of R, say $\mathbf{z}_1$, ..., $\mathbf{z}_N$ are each computed by a single back substitution process without any iteration. These eigenvectors are transformed to eigenvectors of B, say $\mathbf{s}_1$, ..., $\mathbf{s}_N$ by computing

$$\mathbf{s}_j = K\mathbf{z}_j, \quad j = 1, ..., N.$$

BALBAK transforms the vectors s_j to eigenvectors of A, say w_j, $j = 1$, ..., N by computing

$$\mathbf{w}_j = PD\mathbf{s}_j, \quad j = 1, ..., N.$$

SEVVUN normalizes each eigenvector $\mathbf{w}_j$ to have unit unitary norm and a real first component, and then reorders the eigenvalues along with their associated eigenvectors, to achieve the ordering described in Section B.

References

1. R. T. Gregory and D. L. Karney, **A Collection of Matrices for Testing Computational Algorithms**, J. Wiley and Sons, New York (1969) 153 pages.
2. B. T. Smith, J. M. Boyle, B. S. Garbow, Y. Ikebe, V. C. Klema, and C. B. Moler, **Matrix Eigensystem Routines — EISPACK Guide**, *Lecture Notes in Computer Science 6*, Springer Verlag, Berlin (1974) 387 pages.
3. J. H. Wilkinson and C. Reinsch, **Handbook for Automatic Computation, Vol. II. Linear Algebra**, Springer Verlag, Berlin (1971) 439 pages.

E. Error Procedures and Restrictions

If N ≤ 0 or if there is convergence failure in the QR algorithm the error processing subroutine ERMSG of Chapter 19.2 will be called with an error level of 0 to print an error message. Upon return, IFLAG(1) = 3 or 4 to indicate N ≤ 0 or convergence failure, respectively. In these

error conditions all computed eigenvalues and eigenvectors should be regarded as invalid.

If a set of computed eigenvalues are equal or nearly equal, the set of associated computed eigenvectors will frequently not have as large a numerical rank as would be possible for the given matrix.

F. Supporting Information

The source language is ANSI Fortran 77.

The EISPACK package of Fortran subroutines was acquired at JPL from Argonne National Laboratories where it was developed with financial support from the AEC and the NSF. The subroutine SEVVUN was written by F. T. Krogh, JPL, October 1991.

Entry	Required Files
DEVVUN	AMACH, DEVBH, DEVVUN, DNRM2, DSCAL, ERFIN, ERMSG
SEVVUN	AMACH, ERFIN, ERMSG, SEVBH, SEVVUN, SNRM2, SSCAL

DRSEVVUN

```
c      program DRSEVVUN
c>> 1996-05-28 DRSEVVUN  Krogh Added external statement.
c>> 1994-10-19 DRSEVVUN  Krogh  Changes to use M77CON
c>> 1994-09-22 DRSEVVUN  CLL
c>> 1992-04-23 CLL
c>> 1992-03-04 DRSEVVUN  Krogh Initial version.
c      Demonstrate unsymmetric eigenvalue/eigenvector subroutine SEVVUN.
c     ----------------------------------------------------------------------
c--S replaces "?": DR?EVVUN, ?EVVUN, ?VECP, ?MATP, ?DOT
c     ----------------------------------------------------------------------
      integer I, J, LDA, N
      parameter (LDA = 3)
      integer IFLAG(LDA)
      real              A(LDA,LDA), VR(LDA), VI(LDA), VEC(LDA, LDA)
      real              WORK(LDA), ASAV(LDA, LDA), D(LDA, LDA), ANORM
      external SDOT
      real              SDOT
      data (A(1,I), I=1,3) /  8.0e0, -1.0e0, -5.0e0 /
      data (A(2,I), I=1,3) / -4.0e0,  4.0e0, -2.0e0 /
      data (A(3,I), I=1,3) / 18.0e0, -5.0e0, -7.0e0 /
      data ANORM / 30.0e0 /
      data N / LDA /
c     ----------------------------------------------------------------------
      print*,'DRSEVVUN..  Demo driver for SEVVUN.'
c
c     First copy A() to ASAV() for later residual check.
c
         do 20 J = 1, N
            do 10 I = 1, N
               ASAV(I, J) = A(I, J)
   10       continue
   20    continue

         call SEVVUN(A(1, 1), LDA, N, VR, VI, VEC, IFLAG, WORK)

         print '(a, I2)',' IFLAG(1) =',IFLAG(1)
         if (IFLAG(1) .le. 2) then
            call SVECP(VR, N, ' Real part of the eigenvalues')
            call SVECP(VI, N, ' Imaginary part of the eigenvalues')
            call SMATP(VEC, LDA, N, N,
     *          '0 Eigenvectors as columns or pairs of columns')
c
c           As a check compute D = (ASAV*VEC - VEC*EVAL) / ANORM.
```

```
c            Expect D to be close to the machine precision.
c
             do 50 J = 1, N
                if (VI(J) .eq. 0.0e0) then
c                  Compute residual for a real eigenvalue and eigenvector
                   do 30 I = 1, N
                      D(I, J) = (SDOT(N, ASAV(I,1), LDA, VEC(1,J), 1) -
     *                    VEC(I,J) * VR(J)) / ANORM
   30              continue
                else if (VI(J) .gt. 0.0e0) then
                   do 40 I = 1, N
                      D(I, J) = (SDOT(N,ASAV(I,1),LDA,VEC(1,J),1) -
     *                    VEC(I,J)*VR(J)+VEC(I,J+1)*VI(J))/ANORM
                      D(I, J+1) = (SDOT(N,ASAV(I,1),LDA,VEC(1,J+1),1) -
     *                    VEC(I,J)*VI(J)-VEC(I,J+1)*VR(J))/ANORM
   40              continue
                end if
   50        continue
             call SMATP(D, LDA, N, N,
     *       '0 Packed residual matrix D = (A*EVEC - EVEC*EVAL) / ANORM')
          else
             print'(/a)', ' Failure in SEVVUN. '
          end if
       stop
       end
```

ODSEVVUN

```
DRSEVVUN..  Demo driver for SEVVUN.
IFLAG(1) = 2
Real part of the eigenvalues
  1 TO   3   0.9999997       2.000000       2.000000
Imaginary part of the eigenvalues
  1 TO   3    0.000000       4.000000      -4.000000

 Eigenvectors as columns or pairs of columns

                COL   1        COL   2        COL   3
ROW   1       0.4082483      0.4472136       0.000000
ROW   2       0.8164966      0.4472136      0.4472136
ROW   3       0.4082483      0.4472136     -0.4472135

 Packed residual matrix D = (A*EVEC - EVEC*EVAL) / ANORM

                COL   1        COL   2        COL   3
ROW   1       0.000000     9.9341078E-09 -1.1920929E-08
ROW   2     1.9868216E-09  7.9472864E-09  1.9868216E-09
ROW   3     2.3841858E-08  2.7815501E-08 -3.1789146E-08
```