5.2 Eigenvalues and Eigenvectors of a Hermitian Complex Matrix

A. Purpose

Compute the N eigenvalues and right eigenvectors of an N × N complex Hermitian matrix A. A complex matrix is Hermitian if its diagonal elements are real and its off-diagonal pairs $a_{i,j}$ and $a_{j,i}$ are complex conjugates of each other. Such a matrix will have real eigenvalues. The eigenvectors will in general be complex but can be chosen so that the matrix V of N eigenvectors is unitary, *i.e.*, the conjugate transpose of V is the inverse of V.

B. Usage

B.1 Program Prototype

REAL AR(LDA,≥N), **AI**(LDA,≥N) [LDA≥N] , **EVAL**(≥N)

REAL VR(LDA,≥N), **VI**(LDA,≥N), **WORK**(≥3N)

INTEGER LDA, N, IERR

Assign values to AR(,), AI(,), LDA, and N.

CALL SHERQL(AR, AI, LDA, N, EVAL, VR, VI, WORK, IERR)

Results are returned in EVAL(), VR(,), VI(,), and IERR.

B.2 Argument Definitions

AR(,), AI(,) [inout] On entry the locations on and below the diagonal of these arrays must contain the lower-triangular elements of the N × N complex Hermitian matrix A with the real part in AR(,) and the imaginary part in AI(,). On return AR(,) and AI(,) contain information about the unitary transformations used in the reduction of A in their lower triangle. The strict upper triangles of AR(,) and AI(,), and the diagonal of AR(,) are unaltered.

LDA [in] Dimension of the first subscript of the arrays AR(,), AI(,), VR(,), and VI(,). Require LDA ≥ N.

N [in] Order of the complex Hermitian matrix A. N ≥ 1.

EVAL() [out] Array in which the N real eigenvalues of A will be stored by the subroutine. The eigenvalues will be sorted with the algebraically smallest eigenvalues first.

VR(,), VI(,) [out] On return contains the real and imaginary parts of the eigenvectors with column k corresponding to the eigenvalue in EVAL(k). These N eigenvectors will be mutually orthogonal and will have a unit unitary norm.

WORK() [scratch] An array of at least 3N locations used as temporary space.

IERR [out] On exit this is set to 0 if the QL algorithm converges, otherwise see Section E.

B.3 Modifications for Double Precision

Change SHERQL to DHERQL, and the REAL type statement to DOUBLE PRECISION.

C. Examples and Remarks

Consider the following complex Hermitian matrix:

$$A = \begin{bmatrix} 25 & -3-4i & -8+6i & 0 \\ -3+4i & 25 & 0 & -8-6i \\ -8-6i & 0 & 25 & 3-4i \\ 0 & -8+6i & 3+4i & 25 \end{bmatrix}$$

unit unitary norm associated with these eigenvalues are column vectors with the following quadruples of elements: $(0.5i,\ 0.4+0.3i,\ -0.3+0.4i,\ 0.5)$, $(-0.5i,\ 0.4+0.3i,\ 0.3-0.4i,\ 0.5)$, $(-0.5i,\ -0.4-0.3i,\ -0.3+0.4i,\ 0.5)$, and $(0.5i,\ -0.4-0.3i,\ 0.3-0.4i,\ 0.5)$, respectively.

The code in DRSHERQL, given below, computes the eigenvalues and eigenvectors of this matrix. Output from this program is given in the file ODSHERQL.

Before the call to SHERQL, the matrix is saved in order to compute the relative residual matrix D defined as

$$C = (AW - W\Lambda)/\gamma$$

where 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.

Recall that if $\mathbf{v}$ is an eigenvector, then so is $\alpha\mathbf{v}$ for any nonzero complex scalar α. More generally, if an eigenvalue, λ, of a complex Hermitian matrix occurs with multiplicity k, there will be an associated k-dimensional complex subspace in which every vector is an eigenvector for λ. This subroutine will return eigenvectors constituting an orthogonal basis for such an eigenspace.

D. Functional Description

Householder complex unitary similarity transformations are used to transform the matrix A to a Hermitian tridiagonal matrix. Additional unitary similarity transformations are used to transform the matrix to a real tridiagonal matrix. From this point, this subroutine uses the same method as the subroutine SSYMQR of Chapter 5.1. As was the case there, all routines are minor modifications of EISPACK routines, [1].

References

1. 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.

E. Error Procedures and Restrictions

If the QL algorithm fails to converge in 30 iterations on the J^{th} eigenvalue the subroutine sets IERR = J. In this case $J-1$ eigenvalues are computed correctly but the eigenvalues are not ordered, and the eigenvectors are not computed. If N ≤ 0 on entry, IERR is set to -1. In either case an error message is printed using IERM1 of Chapter 19.2 with an error level of 0, before the return.

F. Supporting Information

The source language is ANSI Fortran 77.

Entry	Required Files
DHERQL	DHERQL, DIMQL, ERFIN, ERMSG, IERM1, IERV1
SHERQL	ERFIN, ERMSG, IERM1, IERV1, SHERQL, SIMQL

Converted by: F. T. Krogh, JPL, October 1991.

DRSHERQL

```
c      program DRSHERQL
c>> 1996-05-28 DRSHERQL  Krogh Added external statement.
c>> 1994-10-19 DRSHERQL  Krogh  Changes to use M77CON
c>> 1994-09-23 DRSHERQL  CLL
c>> 1992-04-23 CLL
c>> 1992-03-04 DRSHERQL  Krogh Initial version.
c      Demonstrate Hermitian eigenvalue/eigenvector subroutine SHERQL.
c     ------------------------------------------------------------------
c--S replaces "?": DR?HERQL, ?HERQL, ?VECP, ?MATP, ?DOT
c     ------------------------------------------------------------------
      integer I, IERR, J, LDA, LDA3, N
      parameter (LDA = 4)
      parameter (LDA3 = 3*LDA)
      real              AR(LDA, LDA), AI(LDA, LDA)
      real              ARSAV(LDA, LDA), AISAV(LDA, LDA), ANORM
      external SDOT
      real              SDOT, DR(LDA,LDA), DI(LDA,LDA), EVAL(LDA)
      real              VR(LDA, LDA), VI(LDA, LDA), WORK(LDA3)
      data AR(1,1)           /  25.0e0 /
      data (AR(2,I), I=1,2) /  -3.0e0,  25.0e0 /
      data (AR(3,I), I=1,3) /  -08.0e0,  0.0e0,  25.0e0 /
      data (AR(4,I), I=1,4) /  0.0e0,  -08.0e0,  3.0e0, 25.0e0 /
      data AI(1,1)           /  0.0e0 /
      data (AI(2,I), I=1,2) /  4.0e0,  0.0e0 /
      data (AI(3,I), I=1,3) /  -06.0e0,  0.0e0, 0.0e0 /
      data (AI(4,I), I=1,4) / 0.0e0,  06.0e0, 4.0e0, 0.0e0 /
      data ANORM / 46.0e0 /
      data N /LDA/
c     ------------------------------------------------------------------
      print*,'DRSHERQL..  Demo driver for SHERQL.'
c
c     First copy AR() and AI() to ARSAV() and AISAV() for later
c     residual check.
c
      do 20 I = 1,N
         do 10 J = 1,I
            ARSAV(I,J) = AR(I,J)
            ARSAV(J,I) = ARSAV(I,J)
            AISAV(I,J) = AI(I,J)
            AISAV(J,I) = -AISAV(I,J)
   10    continue
   20 continue

      call SHERQL(AR, AI, LDA, N, EVAL, VR, VI, WORK, IERR)
      if (IERR .eq. 0) then
         call SVECP(EVAL, N, '0 Eigenvalues')
         call SMATP(VR,LDA,N,N,
     *     '0 Real parts of eigenvectors as column vectors')
         call SMATP(VI,LDA,N,N,
     *     '0 Imaginary parts of eigenvectors as column vectors')
c
c        As a check compute D = (ASAV*EVEC - EVEC*EVAL) / ANORM.
c        Expect D to be close to the machine precision.
c
         do 40 J = 1, LDA
            do 30 I = 1, LDA
```

```
            DR(I, J) = (SDOT(LDA,ARSAV(I,1),LDA,VR(1,J),1) -
     *                  SDOT(LDA,AISAV(I,1),LDA,VI(1,J),1) -
     *                  VR(I,J)*EVAL(J))/ANORM
            DI(I, J) = (SDOT(LDA,ARSAV(I,1),LDA,VI(1,J),1) +
     *                  SDOT(LDA,AISAV(I,1),LDA,VR(1,J),1) -
     *                  VI(I,J)*EVAL(J))/ANORM
  30        continue
  40      continue
        call SMATP(DR, LDA, N, N,
     * '0Real part of residual matrix D = (A*EVEC - EVEC*EVAL) / ANORM')
        call SMATP(DI, LDA, N, N,
     * '0Imag part of residual matrix D = (A*EVEC - EVEC*EVAL) / ANORM')
      else
         print '(/a,i5)',' Convergence failure in SHERQL, IERR =',IERR
      end if
      stop
      end
```

ODSHERQL

```
DRSHERQL..  Demo driver for SHERQL.

 Eigenvalues
  1 TO   4     10.00000        20.00001        30.00000        40.00000

 Real parts of eigenvectors as column vectors

                COL   1         COL   2         COL   3         COL   4
ROW   1     -2.9802322E-08  -2.9802322E-08  -4.4703484E-08  -4.4703484E-08
ROW   2      0.4000000       -0.3999998       0.3999999       -0.4000001
ROW   3     -0.3000000       -0.3000006       0.3000004        0.3000001
ROW   4      0.4999999       -0.4999999      -0.5000001        0.4999999

 Imaginary parts of eigenvectors as column vectors

                COL   1         COL   2         COL   3         COL   4
ROW   1      0.5000000        0.5000000       0.5000001        0.5000000
ROW   2      0.3000001       -0.3000001       0.3000002       -0.3000000
ROW   3      0.3999999        0.4000002      -0.3999999       -0.3999998
ROW   4     -0.000000         0.000000        0.000000        -0.000000

Real part of residual matrix D = (A*EVEC - EVEC*EVAL) / ANORM

                COL   1         COL   2         COL   3         COL   4
ROW   1      1.2957556E-09   4.9238626E-08  -5.3773757E-08  -2.5915062E-08
ROW   2     -1.0366025E-08   6.2196150E-08   4.1464101E-08   4.1464101E-08
ROW   3      4.1464101E-08  -2.0732051E-08   0.000000       -2.0732051E-08
ROW   4     -8.2928203E-08   0.000000       -2.0732051E-08   0.000000

Imag part of residual matrix D = (A*EVEC - EVEC*EVAL) / ANORM

                COL   1         COL   2         COL   3         COL   4
ROW   1     -5.1830124E-08  -1.8658845E-07   2.0732051E-08   0.000000
ROW   2      1.0366025E-08   0.000000        0.000000        0.000000
ROW   3     -4.1464101E-08  -4.1464101E-08  -2.0732051E-08  -4.1464101E-08
ROW   4     -1.2957531E-08   1.0366025E-08   0.000000        1.2957531E-08
```