NAG Library Routine Document

F08YKF (DTGEVC)

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

F08YKF (DTGEVC) computes some or all of the right and/or left generalized eigenvectors of a pair of real matrices (A, B) which are in generalized real Schur form.

2 Specification

```
SUBROUTINE F08YKF (SIDE, HOWMNY, SELECT, N, A, LDA, B, LDB, VL, LDVL,      &
                   VR, LDVR, MM, M, WORK, INFO)

INTEGER             N, LDA, LDB, LDVL, LDVR, MM, M, INFO
REAL (KIND=nag_wp) A(LDA,*), B(LDB,*), VL(LDVL,*), VR(LDVR,*),             &
                   WORK(6*N)
LOGICAL             SELECT(*)
CHARACTER(1)        SIDE, HOWMNY
```

The routine may be called by its LAPACK name ***dtgevc***.

3 Description

F08YKF (DTGEVC) computes some or all of the right and/or left generalized eigenvectors of the matrix pair (A, B) which is assumed to be in generalized upper Schur form. If the matrix pair (A, B) is not in the generalized upper Schur form, then F08XEF (DHGEQZ) should be called before invoking F08YKF (DTGEVC).

The right generalized eigenvector x and the left generalized eigenvector y of (A, B) corresponding to a generalized eigenvalue λ are defined by

$$(A - \lambda B)x = 0$$

and

$$y^{\mathrm{H}}(A - \lambda B) = 0.$$

If a generalized eigenvalue is determined as $0/0$, which is due to zero diagonal elements at the same locations in both A and B, a unit vector is returned as the corresponding eigenvector.

Note that the generalized eigenvalues are computed using F08XEF (DHGEQZ) but F08YKF (DTGEVC) does not explicitly require the generalized eigenvalues to compute eigenvectors. The ordering of the eigenvectors is based on the ordering of the eigenvalues as computed by F08YKF (DTGEVC).

If all eigenvectors are requested, the routine may either return the matrices X and/or Y of right or left eigenvectors of (A, B), or the products ZX and/or QY, where Z and Q are two matrices supplied by you. Usually, Q and Z are chosen as the orthogonal matrices returned by F08XEF (DHGEQZ). Equivalently, Q and Z are the left and right Schur vectors of the matrix pair supplied to F08XEF (DHGEQZ). In that case, QY and ZX are the left and right generalized eigenvectors, respectively, of the matrix pair supplied to F08XEF (DHGEQZ).

A must be block upper triangular; with 1 by 1 and 2 by 2 diagonal blocks. Corresponding to each 2 by 2 diagonal block is a complex conjugate pair of eigenvalues and eigenvectors; only one eigenvector of the pair is computed, namely the one corresponding to the eigenvalue with positive imaginary part. Each 1 by 1 block gives a real generalized eigenvalue and a corresponding eigenvector.

4 References

Anderson E, Bai Z, Bischof C, Blackford S, Demmel J, Dongarra J J, Du Croz J J, Greenbaum A, Hammarling S, McKenney A and Sorensen D (1999) *LAPACK Users' Guide* (3rd Edition) SIAM, Philadelphia

Golub G H and Van Loan C F (1996) *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

Moler C B and Stewart G W (1973) An algorithm for generalized matrix eigenproblems *SIAM J. Numer. Anal.* **10** 241–256

Stewart G W and Sun J-G (1990) *Matrix Perturbation Theory* Academic Press, London

5 Arguments

1: SIDE – CHARACTER(1) *Input*

On entry: specifies the required sets of generalized eigenvectors.

SIDE = 'R'
Only right eigenvectors are computed.

SIDE = 'L'
Only left eigenvectors are computed.

SIDE = 'B'
Both left and right eigenvectors are computed.

Constraint: SIDE = 'B', 'L' or 'R'.

2: HOWMNY – CHARACTER(1) *Input*

On entry: specifies further details of the required generalized eigenvectors.

HOWMNY = 'A'
All right and/or left eigenvectors are computed.

HOWMNY = 'B'
All right and/or left eigenvectors are computed; they are backtransformed using the input matrices supplied in arrays VR and/or VL.

HOWMNY = 'S'
Selected right and/or left eigenvectors, defined by the array SELECT, are computed.

Constraint: HOWMNY = 'A', 'B' or 'S'.

3: SELECT($*$) – LOGICAL array *Input*

Note: the dimension of the array SELECT must be at least $\max(1, \mathrm{N})$ if HOWMNY = 'S', and at least 1 otherwise.

On entry: specifies the eigenvectors to be computed if HOWMNY = 'S'. To select the generalized eigenvector corresponding to the jth generalized eigenvalue, the jth element of SELECT should be set to .TRUE.; if the eigenvalue corresponds to a complex conjugate pair, then real and imaginary parts of eigenvectors corresponding to the complex conjugate eigenvalue pair will be computed.

If HOWMNY = 'A' or 'B', SELECT is not referenced.

Constraint: if HOWMNY = 'S', SELECT(j) = .TRUE. or .FALSE., for $j = 1, 2, \ldots, n$.

4: N – INTEGER *Input*

On entry: n, the order of the matrices A and B.

Constraint: N ≥ 0.

5: A(LDA, $*$) – REAL (KIND=nag_wp) array *Input*

Note: the second dimension of the array A must be at least max(1, N).

On entry: the matrix pair (A, B) must be in the generalized Schur form. Usually, this is the matrix A returned by F08XEF (DHGEQZ).

6: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08YKF (DTGEVC) is called.

Constraint: LDA $\geq$ max(1, N).

7: B(LDB, $*$) – REAL (KIND=nag_wp) array *Input*

Note: the second dimension of the array B must be at least max(1, N).

On entry: the matrix pair (A, B) must be in the generalized Schur form. If A has a 2 by 2 diagonal block then the corresponding 2 by 2 block of B must be diagonal with positive elements. Usually, this is the matrix B returned by F08XEF (DHGEQZ).

8: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F08YKF (DTGEVC) is called.

Constraint: LDB $\geq$ max(1, N).

9: VL(LDVL, $*$) – REAL (KIND=nag_wp) array *Input/Output*

Note: the second dimension of the array VL must be at least max(1, MM) if SIDE = 'L' or 'B' and at least 1 if SIDE = 'R'.

On entry: if HOWMNY = 'B' and SIDE = 'L' or 'B', VL must be initialized to an n by n matrix Q. Usually, this is the orthogonal matrix Q of left Schur vectors returned by F08XEF (DHGEQZ).

On exit: if SIDE = 'L' or 'B', VL contains:

if HOWMNY = 'A', the matrix Y of left eigenvectors of (A, B);

if HOWMNY = 'B', the matrix QY;

if HOWMNY = 'S', the left eigenvectors of (A, B) specified by SELECT, stored consecutively in the columns of the array VL, in the same order as their corresponding eigenvalues.

A complex eigenvector corresponding to a complex eigenvalue is stored in two consecutive columns, the first holding the real part, and the second the imaginary part.

If SIDE = 'R', VL is not referenced.

10: LDVL – INTEGER *Input*

On entry: the first dimension of the array VL as declared in the (sub)program from which F08YKF (DTGEVC) is called.

Constraints:

if SIDE = 'L' or 'B', LDVL $\geq$ max(1, N);
if SIDE = 'R', LDVL $\geq$ 1.

11: VR(LDVR, $*$) – REAL (KIND=nag_wp) array *Input/Output*

Note: the second dimension of the array VR must be at least max(1, MM) if SIDE = 'R' or 'B' and at least 1 if SIDE = 'L'.

On entry: if HOWMNY = 'B' and SIDE = 'R' or 'B', VR must be initialized to an n by n matrix Z. Usually, this is the orthogonal matrix Z of right Schur vectors returned by F08XEF (DHGEQZ).

On exit: if SIDE = 'R' or 'B', VR contains:

if HOWMNY = 'A', the matrix X of right eigenvectors of (A, B);

if HOWMNY = 'B', the matrix ZX;

if HOWMNY = 'S', the right eigenvectors of (A, B) specified by SELECT, stored consecutively in the columns of the array VR, in the same order as their corresponding eigenvalues.

A complex eigenvector corresponding to a complex eigenvalue is stored in two consecutive columns, the first holding the real part, and the second the imaginary part.

If SIDE = 'L', VR is not referenced.

12: LDVR – INTEGER *Input*

On entry: the first dimension of the array VR as declared in the (sub)program from which F08YKF (DTGEVC) is called.

Constraints:

if SIDE = 'R' or 'B', LDVR $\geq$ max(1, N);
if SIDE = 'L', LDVR ≥ 1.

13: MM – INTEGER *Input*

On entry: the number of columns in the arrays VL and/or VR.

Constraints:

if HOWMNY = 'A' or 'B', MM $\geq$ N;
if HOWMNY = 'S', MM must not be less than the number of requested eigenvectors.

14: M – INTEGER *Output*

On exit: the number of columns in the arrays VL and/or VR actually used to store the eigenvectors. If HOWMNY = 'A' or 'B', M is set to N. Each selected real eigenvector occupies one column and each selected complex eigenvector occupies two columns.

15: WORK($6 \times$ N) – REAL (KIND=nag_wp) array *Workspace*

16: INFO – INTEGER *Output*

On exit: INFO = 0 unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

INFO < 0

If INFO $= -i$, argument i had an illegal value. An explanatory message is output, and execution of the program is terminated.

INFO > 0

If INFO $= i$, the 2 by 2 block (INFO : INFO + 1) does not have complex eigenvalues.

7 Accuracy

It is beyond the scope of this manual to summarise the accuracy of the solution of the generalized eigenvalue problem. Interested readers should consult Section 4.11 of the LAPACK Users' Guide (see Anderson *et al.* (1999)) and Chapter 6 of Stewart and Sun (1990).

8 Parallelism and Performance

F08YKF (DTGEVC) makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

F08YKF (DTGEVC) is the sixth step in the solution of the real generalized eigenvalue problem and is called after F08XEF (DHGEQZ).

The complex analogue of this routine is F08YXF (ZTGEVC).

10 Example

This example computes the α and β arguments, which defines the generalized eigenvalues and the corresponding left and right eigenvectors, of the matrix pair (A, B) given by

$$A = \begin{pmatrix} 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\ 2.0 & 4.0 & 8.0 & 16.0 & 32.0 \\ 3.0 & 9.0 & 27.0 & 81.0 & 243.0 \\ 4.0 & 16.0 & 64.0 & 256.0 & 1024.0 \\ 5.0 & 25.0 & 125.0 & 625.0 & 3125.0 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 1.0 & 2.0 & 3.0 & 4.0 & 5.0 \\ 1.0 & 4.0 & 9.0 & 16.0 & 25.0 \\ 1.0 & 8.0 & 27.0 & 64.0 & 125.0 \\ 1.0 & 16.0 & 81.0 & 256.0 & 625.0 \\ 1.0 & 32.0 & 243.0 & 1024.0 & 3125.0 \end{pmatrix}.$$

To compute generalized eigenvalues, it is required to call five routines: F08WHF (DGGBAL) to balance the matrix, F08AEF (DGEQRF) to perform the QR factorization of B, F08AGF (DORMQR) to apply Q to A, F08WEF (DGGHRD) to reduce the matrix pair to the generalized Hessenberg form and F08XEF (DHGEQZ) to compute the eigenvalues via the QZ algorithm.

The computation of generalized eigenvectors is done by calling F08YKF (DTGEVC) to compute the eigenvectors of the balanced matrix pair. The routine F08WJF (DGGBAK) is called to backward transform the eigenvectors to the user-supplied matrix pair. If both left and right eigenvectors are required then F08WJF (DGGBAK) must be called twice.

10.1 Program Text

```
!   F08YKF Example Program Text
!   Mark 26 Release. NAG Copyright 2016.

    Module f08ykfe_mod

!     F08YKF Example Program Module:
!            Parameters and User-defined Routines

!     .. Use Statements ..
      Use nag_library, Only: nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Accessibility Statements ..
      Private
      Public                             :: normalize
!     .. Parameters ..
      Real (Kind=nag_wp), Parameter, Public :: one = 1.0_nag_wp
      Real (Kind=nag_wp), Parameter, Public :: zero = 0.0_nag_wp
```

```
      Integer, Parameter, Public       :: nin = 5, nout = 6
    Contains
      Subroutine normalize(n,alphai,v,ldv)

!       .. Use Statements ..
        Use nag_library, Only: dnrm2
!       .. Implicit None Statement ..
        Implicit None
!       .. Scalar Arguments ..
        Integer, Intent (In)           :: ldv, n
!       .. Array Arguments ..
        Real (Kind=nag_wp), Intent (In) :: alphai(n)
        Real (Kind=nag_wp), Intent (Inout) :: v(ldv,*)
!       .. Local Scalars ..
        Real (Kind=nag_wp)             :: a, b, r, r1, r2, v1, v2
        Integer                        :: i, j, k
!       .. Intrinsic Procedures ..
        Intrinsic                      :: sqrt
!       .. Executable Statements ..

        Do j = 1, n

          If (alphai(j)>=0.0_nag_wp) Then
            If (alphai(j)==0.0_nag_wp) Then
!             Real eigenvalue
!             The 2-norm of Q is calculated using dnrm2 (f06ejf).
              r = dnrm2(n,v(1,j),1)
              v(1:n,j) = v(1:n,j)/r
            Else
!             Complex eigenvalue (positive imaginary part)
!             Make largest element real and positive
              r1 = dnrm2(n,v(1,j),1)
              r2 = dnrm2(n,v(1,j+1),1)
              r1 = sqrt(r1**2+r2**2)
              r2 = -1.0_nag_wp
              Do i = 1, n
                r = v(i,j)**2 + v(i,j+1)**2
                If (r>r2) Then
                  r2 = r
                  k = i
                End If
              End Do
              r = r1*sqrt(r2)
              a = v(k,j)/r
              b = v(k,j+1)/r
              Do i = 1, n
                v1 = v(i,j)
                v2 = v(i,j+1)
                v(i,j) = v1*a + v2*b
                v(i,j+1) = v2*a - v1*b
              End Do
            End If
          End If
        End Do
      End Subroutine normalize
    End Module f08ykfe_mod
    Program f08ykfe

!     F08YKF Example Main Program

!     .. Use Statements ..
      Use nag_library, Only: dgeqrf, dggbak, dggbal, dgghrd, dhgeqz, dorgqr,   &
                             dormqr, dtgevc, f06qff, f06qhf, nag_wp, x04cbf
      Use f08ykfe_mod, Only: nin, normalize, nout, one, zero
!     .. Implicit None Statement ..
      Implicit None
!     .. Local Scalars ..
      Integer                        :: i, icols, ifail, ihi, ilo, info,     &
                                        irows, jwork, lda, ldb, ldvl, ldvr,  &
                                        lwork, m, n
      Logical                        :: ileft, iright
```

```
      Character (1)                    :: compq, compz, howmny, job, side
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:,:), alphai(:), alphar(:),        &
                                          b(:,:), beta(:), lscale(:),          &
                                          rscale(:), tau(:), vl(:,:), vr(:,:), &
                                          work(:)
      Logical, Allocatable             :: select(:)
      Character (0)                    :: clabs(1), rlabs(1)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: nint
!     .. Executable Statements ..
      Write (nout,*) 'F08YKF Example Program Results'
      Flush (nout)

!     ileft  is TRUE if left  eigenvectors are required
!     iright is TRUE if right eigenvectors are required

      ileft = .True.
      iright = .True.

!     Skip heading in data file

      Read (nin,*)
      Read (nin,*) n
      lda = n
      ldb = n
      ldvl = n
      ldvr = n
      lwork = 6*n
      Allocate (a(lda,n),alphai(n),alphar(n),b(ldb,n),beta(n),lscale(n),       &
        rscale(n),tau(n),vl(ldvl,ldvl),vr(ldvr,ldvr),work(lwork),select(n))

!     READ matrix A from data file
      Read (nin,*)(a(i,1:n),i=1,n)

!     READ matrix B from data file
      Read (nin,*)(b(i,1:n),i=1,n)

!     Balance matrix pair (A,B)
      job = 'B'
!     The NAG name equivalent of dggbal is f08whf
      Call dggbal(job,n,a,lda,b,ldb,ilo,ihi,lscale,rscale,work,info)

!     Matrix A after balancing
!     ifail: behaviour on error exit
!             =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
      ifail = 0
      Call x04cbf('General',' ',n,n,a,lda,'F8.4','Matrix A after balancing',   &
        'I',rlabs,'I',clabs,80,0,ifail)
      Write (nout,*)
      Flush (nout)

!     Matrix B after balancing
      ifail = 0
      Call x04cbf('General',' ',n,n,b,ldb,'F8.4','Matrix B after balancing',   &
        'I',rlabs,'I',clabs,80,0,ifail)
      Write (nout,*)
      Flush (nout)

!     Reduce B to triangular form using QR
      irows = ihi + 1 - ilo
      icols = n + 1 - ilo
!     The NAG name equivalent of dgeqrf is f08aef
      Call dgeqrf(irows,icols,b(ilo,ilo),ldb,tau,work,lwork,info)

!     Apply the orthogonal transformation to matrix A
!     The NAG name equivalent of dormqr is f08agf
      Call dormqr('L','T',irows,icols,irows,b(ilo,ilo),ldb,tau,a(ilo,ilo),lda, &
        work,lwork,info)

!     Initialize VL (if left eigenvectors are required)
```

```
      If (ileft) Then

        Call f06qhf('General',n,n,zero,one,vl,ldvl)
        Call f06qff('Lower',irows-1,irows-1,b(ilo+1,ilo),ldb,vl(ilo+1,ilo),    &
          ldvl)

!       The NAG name equivalent of dorgqr is f08aff
        Call dorgqr(irows,irows,irows,vl(ilo,ilo),ldvl,tau,work,lwork,info)
      End If

!     Initialize VR (if right eigenvectors are required)
      If (iright) Then
        Call f06qhf('General',n,n,zero,one,vr,ldvr)
      End If

!     Compute the generalized Hessenberg form of (A,B)
      compq = 'V'
      compz = 'V'
!     The NAG name equivalent of dgghrd is f08wef
      Call dgghrd(compq,compz,n,ilo,ihi,a,lda,b,ldb,vl,ldvl,vr,ldvr,info)

!     Matrix A in generalized Hessenberg form
      ifail = 0
      Call x04cbf('General',' ',n,n,a,lda,'F8.4','Matrix A in Hessenberg form' &
        ,'I',rlabs,'I',clabs,80,0,ifail)
      Write (nout,*)
      Flush (nout)

!     Matrix B in generalized Hessenberg form
      ifail = 0
      Call x04cbf('General',' ',n,n,b,ldb,'F8.4','Matrix B in Hessenberg form' &
        ,'I',rlabs,'I',clabs,80,0,ifail)

!     Routine DHGEQZ
!     Workspace query: jwork = -1

      jwork = -1
      job = 'S'

!     The NAG name equivalent of dhgeqz is f08xef
      Call dhgeqz(job,compq,compz,n,ilo,ihi,a,lda,b,ldb,alphar,alphai,beta,vl, &
        ldvl,vr,ldvr,work,jwork,info)

      Write (nout,*)
      Write (nout,99999) nint(work(1))
      Write (nout,99998) lwork
      Write (nout,*)
      Write (nout,99997)
      Write (nout,99996)

!     Compute the generalized Schur form
!     if the workspace lwork is adequate
!     The Schur form also gives parameters
!     required to compute generalized eigenvalues

      If (nint(work(1))<=lwork) Then

!       The NAG name equivalent of dhgeqz is f08xef
        Call dhgeqz(job,compq,compz,n,ilo,ihi,a,lda,b,ldb,alphar,alphai,beta,  &
          vl,ldvl,vr,ldvr,work,lwork,info)

!       Print the generalized eigenvalues

        Do i = 1, n
          If (beta(i)/=0.0E0_nag_wp) Then
            Write (nout,99995) i, '(', alphar(i)/beta(i), ',',                 &
              alphai(i)/beta(i), ')'
          Else
            Write (nout,99996) i
          End If
        End Do
```

```
        Write (nout,*)
        Flush (nout)

!       Compute left and right generalized eigenvectors
!       of the balanced matrix

        howmny = 'B'
        If (ileft .And. iright) Then
          side = 'B'
        Else If (ileft) Then
          side = 'L'
        Else If (iright) Then
          side = 'R'
        End If

!       The NAG name equivalent of dtgevc is f08ykf
        Call dtgevc(side,howmny,select,n,a,lda,b,ldb,vl,ldvl,vr,ldvr,n,m,work, &
          info)

        If (iright) Then

!         Compute right eigenvectors of the original matrix

          job = 'B'
          side = 'R'

!         The NAG name equivalent of dggbak is f08wjf
          Call dggbak(job,side,n,ilo,ihi,lscale,rscale,n,vr,ldvr,info)

          Call normalize(n,alphai,vr,ldvr)
!         Print the right eigenvectors

          ifail = 0
          Call x04cbf('General',' ',n,n,vr,ldvr,'F8.4','Right eigenvectors',   &
            'I',rlabs,'I',clabs,80,0,ifail)

          Write (nout,*)
          Flush (nout)
        End If

!       Compute left eigenvectors of the original matrix

        If (ileft) Then
          job = 'B'
          side = 'L'

!         The NAG name equivalent of dggbak is f08wjf
          Call dggbak(job,side,n,ilo,ihi,lscale,rscale,n,vl,ldvl,info)

          Call normalize(n,alphai,vl,ldvl)
!         Print the left eigenvectors

          ifail = 0
          Call x04cbf('General',' ',n,n,vl,ldvl,'F8.4','Left eigenvectors',    &
            'I',rlabs,'I',clabs,80,0,ifail)

        End If
      Else
        Write (nout,99994)
      End If

99999 Format (1X,'Minimal required LWORK = ',I6)
99998 Format (1X,'Actual value of  LWORK = ',I6)
99997 Format (1X,'Generalized eigenvalues')
99996 Format (1X,I4,5X,'Infinite eigenvalue')
99995 Format (1X,I4,5X,A,F7.3,A,F7.3,A)
99994 Format (1X,'Insufficient workspace for array WORK',/,' in F08XEF/',      &
        'DHGEQZ')
    End Program f08ykfe
```

10.2 Program Data

```
F08YKF Example Program Data
   5                                                          :Value of N
1.00        1.00        1.00        1.00        1.00
2.00        4.00        8.00       16.00       32.00
3.00        9.00       27.00       81.00      243.00
4.00       16.00       64.00      256.00     1024.00
5.00       25.00      125.00      625.00     3125.00          :End of matrix A
1.00        2.00        3.00        4.00        5.00
1.00        4.00        9.00       16.00       25.00
1.00        8.00       27.00       64.00      125.00
1.00       16.00       81.00      256.00      625.00
1.00       32.00      243.00     1024.00     3125.00          :End of matrix B
```

10.3 Program Results

```
 F08YKF Example Program Results
 Matrix A after balancing
         1       2       3       4       5
1   1.0000  1.0000  0.1000  0.1000  0.1000
2   2.0000  4.0000  0.8000  1.6000  3.2000
3   0.3000  0.9000  0.2700  0.8100  2.4300
4   0.4000  1.6000  0.6400  2.5600 10.2400
5   0.5000  2.5000  1.2500  6.2500 31.2500

 Matrix B after balancing
         1       2       3       4       5
1   1.0000  2.0000  0.3000  0.4000  0.5000
2   1.0000  4.0000  0.9000  1.6000  2.5000
3   0.1000  0.8000  0.2700  0.6400  1.2500
4   0.1000  1.6000  0.8100  2.5600  6.2500
5   0.1000  3.2000  2.4300 10.2400 31.2500

 Matrix A in Hessenberg form
         1       2       3       4       5
1  -2.1898 -0.3181  2.0547  4.7371 -4.6249
2  -0.8395 -0.0426  1.7132  7.5194-17.1850
3   0.0000 -0.2846 -1.0101 -7.5927 26.4499
4   0.0000  0.0000  0.0376  1.4070 -3.3643
5   0.0000  0.0000  0.0000  0.3813 -0.9937

 Matrix B in Hessenberg form
         1       2       3       4       5
1  -1.4248 -0.3476  2.1175  5.5813 -3.9269
2   0.0000 -0.0782  0.1189  8.0940-15.2928
3   0.0000  0.0000  1.0021-10.9356 26.5971
4   0.0000  0.0000  0.0000  0.5820 -0.0730
5   0.0000  0.0000  0.0000  0.0000  0.5321

 Minimal required LWORK =      5
 Actual value of  LWORK =     30

 Generalized eigenvalues

    1     ( -2.437,  0.000)
    2     (  0.607,  0.795)
    3     (  0.607, -0.795)
    4     (  1.000,  0.000)
    5     ( -0.410,  0.000)

 Right eigenvectors
         1       2       3       4       5
1  -0.3083  0.7026  0.0000 -0.3985 -0.3747
2   0.6622 -0.5582 -0.3678  0.7287  0.7339
3  -0.6244  0.1600  0.1763 -0.5380 -0.5394
4   0.2732 -0.0211 -0.0492  0.1423  0.1720
5  -0.0438  0.0010  0.0072 -0.0199 -0.0192

 Left eigenvectors
```

```
          1       2       3       4       5
 1  -0.3747  0.7026  0.0000 -0.3985  0.3083
 2   0.7339 -0.5582 -0.3678  0.7287 -0.6622
 3  -0.5394  0.1600  0.1763 -0.5380  0.6244
 4   0.1720 -0.0211 -0.0492  0.1423 -0.2732
 5  -0.0192  0.0010  0.0072 -0.0199  0.0438
```