NAG Library Routine Document

F08YXF (ZTGEVC)

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

F08YXF (ZTGEVC) computes some or all of the right and/or left generalized eigenvectors of a pair of complex upper triangular matrices (A, B).

2 Specification

```
SUBROUTINE F08YXF (SIDE, HOWMNY, SELECT, N, A, LDA, B, LDB, VL, LDVL,      &
                   VR, LDVR, MM, M, WORK, RWORK, INFO)
INTEGER                N, LDA, LDB, LDVL, LDVR, MM, M, INFO
REAL (KIND=nag_wp)     RWORK(2*N)
COMPLEX (KIND=nag_wp) A(LDA,*), B(LDB,*), VL(LDVL,*), VR(LDVR,*),          &
                       WORK(2*N)
LOGICAL                SELECT(*)
CHARACTER(1)           SIDE, HOWMNY
```

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

3 Description

F08YXF (ZTGEVC) computes some or all of the right and/or left generalized eigenvectors of the matrix pair (A, B) which is assumed to be in upper triangular form. If the matrix pair (A, B) is not upper triangular then the routine F08XSF (ZHGEQZ) should be called before invoking F08YXF (ZTGEVC).

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 F08XSF (ZHGEQZ) but F08YXF (ZTGEVC) 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 F08YXF (ZTGEVC).

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 unitary matrices returned by F08XSF (ZHGEQZ). Equivalently, Q and Z are the left and right Schur vectors of the matrix pair supplied to F08XSF (ZHGEQZ). In that case, QY and ZX are the left and right generalized eigenvectors, respectively, of the matrix pair supplied to F08XSF (ZHGEQZ).

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 Parameters

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..

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: $\mathrm{N} \geq 0$.

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

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

On entry: the matrix A must be in upper triangular form. Usually, this is the matrix A returned by F08XSF (ZHGEQZ).

6: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08YXF (ZTGEVC) is called.

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

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

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

On entry: the matrix B must be in upper triangular form with non-negative real diagonal elements. Usually, this is the matrix B returned by F08XSF (ZHGEQZ).

8: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F08YXF (ZTGEVC) is called.

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

9: $VL(LDVL, *)$ – COMPLEX (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 unitary matrix Q of left Schur vectors returned by F08XSF (ZHGEQZ).

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.

10: LDVL – INTEGER *Input*

On entry: the first dimension of the array VL as declared in the (sub)program from which F08YXF (ZTGEVC) is called.

Constraints:

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

11: $VR(LDVR, *)$ – COMPLEX (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 unitary 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.

12: LDVR – INTEGER *Input*

On entry: the first dimension of the array VR as declared in the (sub)program from which F08YXF (ZTGEVC) 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 eigenvector occupies one column.

15: WORK($2 \times N$) – COMPLEX (KIND=nag_wp) array *Workspace*

16: RWORK($2 \times N$) – REAL (KIND=nag_wp) array *Workspace*

17: 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.

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

F08YXF (ZTGEVC) is not threaded by NAG in any implementation.

F08YXF (ZTGEVC) 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

F08YXF (ZTGEVC) is the sixth step in the solution of the complex generalized eigenvalue problem and is usually called after F08XSF (ZHGEQZ).

The real analogue of this routine is F08YKF (DTGEVC).

10 Example

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

$$A = \begin{pmatrix} 1.0 + 3.0i & 1.0 + 4.0i & 1.0 + 5.0i & 1.0 + 6.0i \\ 2.0 + 2.0i & 4.0 + 3.0i & 8.0 + 4.0i & 16.0 + 5.0i \\ 3.0 + 1.0i & 9.0 + 2.0i & 27.0 + 3.0i & 81.0 + 4.0i \\ 4.0 + 0.0i & 16.0 + 1.0i & 64.0 + 2.0i & 256.0 + 3.0i \end{pmatrix}$$

and

$$B = \begin{pmatrix} 1.0 + 0.0i & 2.0 + 1.0i & 3.0 + 2.0i & 4.0 + 3.0i \\ 1.0 + 1.0i & 4.0 + 2.0i & 9.0 + 3.0i & 16.0 + 4.0i \\ 1.0 + 2.0i & 8.0 + 3.0i & 27.0 + 4.0i & 64.0 + 5.0i \\ 1.0 + 3.0i & 16.0 + 4.0i & 81.0 + 5.0i & 256.0 + 6.0i \end{pmatrix}.$$

To compute generalized eigenvalues, it is required to call five routines: F08WVF (ZGGBAL) to balance the matrix, F08ASF (ZGEQRF) to perform the QR factorization of B, F08AUF (ZUNMQR) to apply Q to A, F08WSF (ZGGHRD) to reduce the matrix pair to the generalized Hessenberg form and F08XSF (ZHGEQZ) to compute the eigenvalues via the QZ algorithm.

The computation of generalized eigenvectors is done by calling F08YXF (ZTGEVC) to compute the eigenvectors of the balanced matrix pair. The routine F08WWF (ZGGBAK) is called to backward transform the eigenvectors to the user-supplied matrix pair. If both left and right eigenvectors are required then F08WWF (ZGGBAK) must be called twice.

10.1 Program Text

```
    Program f08yxfe

!     F08YXF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: f06tff, f06thf, nag_wp, x04dbf, zgeqrf, zggbak,   &
                             zggbal, zgghrd, zhgeqz, ztgevc, zungqr, zunmqr
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Complex (Kind=nag_wp), Parameter :: cone = (1.0E0_nag_wp,0.0E0_nag_wp)
      Complex (Kind=nag_wp), Parameter :: czero = (0.0E0_nag_wp,0.0E0_nag_wp)
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Complex (Kind=nag_wp)            :: e
      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 ..
      Complex (Kind=nag_wp), Allocatable :: a(:,:), alpha(:), b(:,:), beta(:), &
                                            tau(:), vl(:,:), vr(:,:), work(:)
      Real (Kind=nag_wp), Allocatable  :: lscale(:), rscale(:), rwork(:)
      Logical, Allocatable             :: select(:)
      Character (1)                    :: clabs(1), rlabs(1)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: aimag, nint, real
!     .. Executable Statements ..
```

```
      Write (nout,*) 'F08YXF 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),alpha(n),b(ldb,n),beta(n),tau(n),vl(ldvl,ldvl), &
        vr(ldvr,ldvr),work(lwork),lscale(n),rscale(n),rwork(6*n),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 zggbal is f08wvf
      Call zggbal(job,n,a,lda,b,ldb,ilo,ihi,lscale,rscale,rwork,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 x04dbf('General',' ',n,n,a,lda,'Bracketed','F7.4', &
        'Matrix A after balancing','Integer',rlabs,'Integer',clabs,80,0,ifail)
      Write (nout,*)
      Flush (nout)

!     Matrix B after balancing
      ifail = 0
      Call x04dbf('General',' ',n,n,b,ldb,'Bracketed','F7.4', &
        'Matrix B after balancing','Integer',rlabs,'Integer',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 zgeqrf is f08asf
      Call zgeqrf(irows,icols,b(ilo,ilo),ldb,tau,work,lwork,info)

!     Apply the orthogonal transformation to A
!     The NAG name equivalent of zunmqr is f08auf
      Call zunmqr('L','C',irows,icols,irows,b(ilo,ilo),ldb,tau,a(ilo,ilo),lda, &
        work,lwork,info)

!     Initialize VL (for left eigenvectors)
      If (ileft) Then

        Call f06thf('General',n,n,czero,cone,vl,ldvl)
        Call f06tff('Lower',irows-1,irows-1,b(ilo+1,ilo),ldb,vl(ilo+1,ilo), &
          ldvl)
!       The NAG name equivalent of zungqr is f08atf
        Call zungqr(irows,irows,irows,vl(ilo,ilo),ldvl,tau,work,lwork,info)

      End If

!     Initialize VR for right eigenvectors
      If (iright) Call f06thf('General',n,n,czero,cone,vr,ldvr)
```

```
!     Compute the generalized Hessenberg form of (A,B)
      compq = 'V'
      compz = 'V'
!     The NAG name equivalent of zgghrd is f08wsf
      Call zgghrd(compq,compz,n,ilo,ihi,a,lda,b,ldb,vl,ldvl,vr,ldvr,info)

!     Matrix A in generalized Hessenberg form
      ifail = 0
      Call x04dbf('General',' ',n,n,a,lda,'Bracketed','F7.3', &
        'Matrix A in Hessenberg form','Integer',rlabs,'Integer',clabs,80,0, &
        ifail)
      Write (nout,*)
      Flush (nout)

!     Matrix B in generalized Hessenberg form
      ifail = 0
      Call x04dbf('General',' ',n,n,b,ldb,'Bracketed','F7.3', &
        'Matrix B in Hessenberg form','Integer',rlabs,'Integer',clabs,80,0, &
        ifail)

!     Routine ZHGEQZ
!     Workspace query: jwork = -1
      jwork = -1
      job = 'S'
!     The NAG name equivalent of zhgeqz is f08xsf
      Call zhgeqz(job,compq,compz,n,ilo,ihi,a,lda,b,ldb,alpha,beta,vl,ldvl,vr, &
        ldvr,work,jwork,rwork,info)

      Write (nout,*)
      Write (nout,99999) nint(real(work(1)))
      Write (nout,99998) lwork
      Write (nout,*)
      Flush (nout)

!     Compute the generalized Schur form
!     if the workspace lwork is adequate

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

!       The NAG name equivalent of zhgeqz is f08xsf
        Call zhgeqz(job,compq,compz,n,ilo,ihi,a,lda,b,ldb,alpha,beta,vl,ldvl, &
          vr,ldvr,work,lwork,rwork,info)

!       Print the generalized eigenvalues
!       Note: the actual values of beta are real and non-negative

        Write (nout,99997)
        Do i = 1, n
          If (real(beta(i))/=0.0E0_nag_wp) Then
            e = alpha(i)/beta(i)
            Write (nout,99995) i, '(', real(e), ',', aimag(e), ')'
          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 ztgevc is f08yxf
```

```
        Call ztgevc(side,howmny,select,n,a,lda,b,ldb,vl,ldvl,vr,ldvr,n,m,work, &
          rwork,info)

!       Compute right eigenvectors of the original matrix

        If (iright) Then
          job = 'B'
          side = 'R'

!         The NAG name equivalent of zggbak is f08wwf
          Call zggbak(job,side,n,ilo,ihi,lscale,rscale,n,vr,ldvr,info)

!         Normalize the right eigenvectors
          Do i = 1, n
            vr(1:n,i) = vr(1:n,i)/vr(1,i)
          End Do

!         Print the right eigenvectors

          ifail = 0
          Call x04dbf('General',' ',n,n,vr,ldvr,'Bracketed','F7.4', &
            'Right eigenvectors','Integer',rlabs,'Integer',clabs,80,0,ifail)

          Write (nout,*)
          Flush (nout)

        End If

!       Compute left eigenvectors of the original matrix

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

!         The NAG name equivalent of zggbak is f08wwf
          Call zggbak(job,side,n,ilo,ihi,lscale,rscale,n,vl,ldvl,info)

!         Normalize the left eigenvectors
          Do i = 1, n
            vl(1:n,i) = vl(1:n,i)/vl(1,i)
          End Do

!         Print the left eigenvectors

          ifail = 0
          Call x04dbf('General',' ',n,n,vl,ldvl,'Bracketed','F7.4', &
            'Left eigenvectors','Integer',rlabs,'Integer',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,' Infinite eigenvalue')
99995 Format (1X,I4,5X,A,F7.3,A,F7.3,A)
99994 Format (1X,'Insufficient workspace for array WORK'/' in F08XSF/', &
        'ZHGEQZ')
    End Program f08yxfe
```

10.2 Program Data

```
F08YXF Example Program Data
   4                                                                :Value of N
(  1.00, 3.00)  (  1.00, 4.00)  (  1.00, 5.00)  (  1.00, 6.00)
(  2.00, 2.00)  (  4.00, 3.00)  (  8.00, 4.00)  ( 16.00, 5.00)
(  3.00, 1.00)  (  9.00, 2.00)  ( 27.00, 3.00)  ( 81.00, 4.00)
(  4.00, 0.00)  ( 16.00, 1.00)  ( 64.00, 2.00)  (256.00, 3.00)  :End of matrix A
```

```
(   1.00, 0.00)  (   2.00, 1.00)  (   3.00, 2.00)  (   4.00, 3.00)
(   1.00, 1.00)  (   4.00, 2.00)  (   9.00, 3.00)  ( 16.00, 4.00)
(   1.00, 2.00)  (   8.00, 3.00)  ( 27.00, 4.00)  ( 64.00, 5.00)
(   1.00, 3.00)  ( 16.00, 4.00)  ( 81.00, 5.00)  (256.00, 6.00)  :End of matrix B
```

10.3 Program Results

```
 F08YXF Example Program Results
 Matrix A after balancing
                   1                  2                  3                  4
 1  ( 1.0000, 3.0000) ( 1.0000, 4.0000) ( 0.1000, 0.5000) ( 0.1000, 0.6000)
 2  ( 2.0000, 2.0000) ( 4.0000, 3.0000) ( 0.8000, 0.4000) ( 1.6000, 0.5000)
 3  ( 0.3000, 0.1000) ( 0.9000, 0.2000) ( 0.2700, 0.0300) ( 0.8100, 0.0400)
 4  ( 0.4000, 0.0000) ( 1.6000, 0.1000) ( 0.6400, 0.0200) ( 2.5600, 0.0300)

 Matrix B after balancing
                   1                  2                  3                  4
 1  ( 1.0000, 0.0000) ( 2.0000, 1.0000) ( 0.3000, 0.2000) ( 0.4000, 0.3000)
 2  ( 1.0000, 1.0000) ( 4.0000, 2.0000) ( 0.9000, 0.3000) ( 1.6000, 0.4000)
 3  ( 0.1000, 0.2000) ( 0.8000, 0.3000) ( 0.2700, 0.0400) ( 0.6400, 0.0500)
 4  ( 0.1000, 0.3000) ( 1.6000, 0.4000) ( 0.8100, 0.0500) ( 2.5600, 0.0600)

 Matrix A in Hessenberg form
                   1                  2                  3                  4
 1  ( -2.868, -1.595) ( -0.809, -0.328) ( -4.900, -0.987) ( -0.048,  1.163)
 2  ( -2.672,  0.595) ( -0.790,  0.049) ( -4.955, -0.163) ( -0.439, -0.574)
 3  (  0.000,  0.000) ( -0.098, -0.011) ( -1.168, -0.137) ( -1.756, -0.205)
 4  (  0.000,  0.000) (  0.000,  0.000) (  0.087,  0.004) (  0.032,  0.001)

 Matrix B in Hessenberg form
                   1                  2                  3                  4
 1  ( -1.775,  0.000) ( -0.721,  0.043) ( -5.021,  1.190) ( -0.145,  0.726)
 2  (  0.000,  0.000) ( -0.218,  0.035) ( -2.541, -0.146) ( -0.823, -0.418)
 3  (  0.000,  0.000) (  0.000,  0.000) ( -1.396, -0.163) ( -1.747, -0.204)
 4  (  0.000,  0.000) (  0.000,  0.000) (  0.000,  0.000) ( -0.100, -0.004)

 Minimal required LWORK =      4
 Actual value of  LWORK =     24

 Generalized eigenvalues
    1     ( -0.635,  1.653)
    2     (  0.458, -0.843)
    3     (  0.674, -0.050)
    4     (  0.493,  0.910)

 Right eigenvectors
                   1                  2                  3                  4
 1  ( 1.0000,-0.0000) ( 1.0000, 0.0000) ( 1.0000,-0.0000) ( 1.0000, 0.0000)
 2  (-0.8639,-0.2796) (-1.6440,-0.0550) (-1.0246, 0.1861) (-1.9996,-0.1245)
 3  ( 0.3132, 0.1060) ( 0.9119, 0.2267) ( 0.1281,-0.0133) ( 1.1549,-0.0127)
 4  (-0.0518,-0.0122) (-0.1487,-0.1000) ( 0.0067,-0.0022) (-0.2130, 0.0682)

 Left eigenvectors
                   1                  2                  3                  4
 1  ( 1.0000, 0.0000) ( 1.0000,-0.0000) ( 1.0000,-0.0000) ( 1.0000,-0.0000)
 2  (-0.7857, 0.3499) (-2.1501, 0.0091) (-1.6147, 0.8923) (-2.0208, 0.9724)
 3  ( 0.2535,-0.1483) ( 1.2377, 0.0697) ( 0.2769,-0.1860) ( 1.0247,-0.7718)
 4  (-0.0297, 0.0264) (-0.2257,-0.0827) (-0.0344, 0.0282) (-0.1346, 0.2047)
```