NAG Fortran Library Routine Document

F08YXF (CTGEVC/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 (CTGEVC/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,
1                  LDVR, MM, M, WORK, RWORK, INFO)
 ENTRY      ctgevc (SIDE, HOWMNY, SELECT, N, A, LDA, B, LDB, VL, LDVL, VR,
1                  LDVR, MM, M, WORK, RWORK, INFO)
 INTEGER           N, LDA, LDB, LDVL, LDVR, MM, M, INFO
 real              RWORK(*)
 complex           A(LDA,*), B(LDB,*), VL(LDVL,*), VR(LDVR,*), WORK(*)
 LOGICAL           SELECT(*)
 CHARACTER*1       SIDE, HOWMNY
```

The ENTRY statement enables the routine to be called by its LAPACK name.

3 Description

F08YXF (CTGEVC/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 (CHGEQZ/ZHGEQZ) should be called before invoking F08YXF (CTGEVC/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

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

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 (CHGEQZ/ZHGEQZ) but F08YXF (CTGEVC/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 (CTGEVC/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 the user. Usually, Q and Z are chosen as the unitary matrices returned by F08XSF (CHGEQZ/ZHGEQZ). Equivalently, Q and Z are the left and right Schur vectors of the matrix pair supplied to F08XSF (CHGEQZ/ZHGEQZ). In that case, QY and ZX are the left and right generalized eigenvectors, respectively, of the matrix pair supplied to F08XSF (CHGEQZ/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:

if SIDE = 'R', only right eigenvectors are computed;

if SIDE = 'L', only left eigenvectors are computed;

if 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:

if HOWMNY = 'A', all right and/or left eigenvectors are computed;

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

if 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, N) if HOWMNY = 'S'; otherwise, SELECT is not referenced.

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..

Constraints: SELECT(j) = .TRUE. or .FALSE., for $j = 1, \ldots, n$

4: N – INTEGER *Input*

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

Constraint: N $\geq$ 0.

5: A(LDA,*) – ***complex*** 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 (CHGEQZ/ZHGEQZ).

6: LDA – INTEGER *Input*

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

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

7: B(LDB,*) – ***complex*** array *Input*

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

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 (CHGEQZ/ZHGEQZ)

8: LDB – INTEGER *Input*

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

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

9: VL(LDVL,*) – ***complex*** array *Input/Output*

Note: the second dimension of the array VL must be at least $\max(1, \mathrm{MM})$ 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 initialised to an n by n matrix Q. Usually, this is the unitary matrix Q of left Schur vectors returned by F08XSF (CHGEQZ/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 (CTGEVC/ZTGEVC) is called.

Constraints:

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

11: VR(LDVR,*) – ***complex*** array *Input/Output*

Note: the second dimension of the array VR must be at least $\max(1, \mathrm{MM})$ 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 initialised to an n by n matrix Z. Usually, this is the unitary matrix Z of right Schur vectors returned by F08XEF (SHGEQZ/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 (CTGEVC/ZTGEVC) is called.

Constraints:

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

13: MM – INTEGER *Input*

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

Constraints:

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

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(*) – ***complex*** array *Workspace*

Note: the dimension of the array WORK must be at least $\max(1, 2 * \mathrm{N})$.

16: RWORK(*) – ***real*** array *Workspace*

Note: the dimension of the array RWORK must be at least $\max(1, 2 * \mathrm{N})$.

17: INFO – INTEGER *Output*

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

6 Error Indicators and Warnings

Errors or warnings detected by the routine:

INFO < 0

If INFO $= -i$, the ith parameter 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 summarize the accuracy of the solution of the generalized eigenvalue problem. Interested readers should consult section 4.11 of the LAPACK Users' Guide Anderson *et al.* (1999) and Chapter 6 of Stewart and Sun (1990).

8 Further Comments

This routine is the sixth step in the solution of the complex generalized eigenvalue problem and is usually called after F08XSF (CHGEQZ/ZHGEQZ).

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

9 Example

The example program 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

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

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

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

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

9.1 Program Text

Note: the listing of the example program presented below uses ***bold italicised*** terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     F08YXF Example Program Text
*     Mark 20 Release. NAG Copyright 2001.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, LDA, LDB, LDQ, LDZ, LWORK
      PARAMETER        (NMAX=10,LDA=NMAX,LDB=NMAX,LDQ=NMAX,LDZ=NMAX,
     +                 LWORK=6*NMAX)
      complex          CONE, CZERO
      PARAMETER        (CONE=1.0e0,CZERO=0.0e0)
*     .. Local Scalars ..
      complex          E
      INTEGER          I, ICOLS, IFAIL, IHI, ILO, INFO, IROWS, J, JWORK,
     +                 M, N
      LOGICAL          ILEFT, IRIGHT
      CHARACTER        COMPQ, COMPZ, HOWMNY, JOB, SIDE
*     .. Local Arrays ..
      complex          A(LDA,NMAX), ALPHA(NMAX), B(LDB,NMAX),
     +                 BETA(NMAX), Q(LDQ,LDQ), TAU(NMAX), WORK(LWORK),
     +                 Z(LDZ,LDZ)
      real             LSCALE(NMAX), RSCALE(NMAX), RWORK(6*NMAX)
      LOGICAL          SELECT(NMAX)
      CHARACTER        CLABS(1), RLABS(1)
*     .. External Subroutines ..
      EXTERNAL         F06TFF, F06THF, X04DBF, cgeqrf, cggbak, cggbal,
     +                 cgghrd, chgeqz, ctgevc, cungqr, cunmqr
*     .. Intrinsic Functions ..
      INTRINSIC        real, imag, NINT
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08YXF Example Program Results'
*
*     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
      IF (N.LE.NMAX) THEN
*
*        READ matrix A from data file
*
         READ (NIN,*) ((A(I,J),J=1,N),I=1,N)
*
*        READ matrix B from data file
*
         READ (NIN,*) ((B(I,J),J=1,N),I=1,N)
*
*        Balance matrix pair (A,B)
*
         JOB = 'B'
         CALL cggbal(JOB,N,A,LDA,B,LDB,ILO,IHI,LSCALE,RSCALE,RWORK,INFO)
*
*        Matrix A after balancing
*
         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,*)
*
*        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,*)
*
*        Reduce B to triangular form using QR
*
         IROWS = IHI + 1 - ILO
         ICOLS = N + 1 - ILO
         CALL cgeqrf(IROWS,ICOLS,B(ILO,ILO),LDB,TAU,WORK,LWORK,INFO)
*
*        Apply the orthogonal transformation to A
*
         CALL cunmqr('L','C',IROWS,ICOLS,IROWS,B(ILO,ILO),LDB,TAU,
     +               A(ILO,ILO),LDA,WORK,LWORK,INFO)
*
*        Initialize Q (for left eigenvectors)
*
         IF (ILEFT) THEN
*
            CALL F06THF('General',N,N,CZERO,CONE,Q,LDQ)
            CALL F06TFF('Lower',IROWS-1,IROWS-1,B(ILO+1,ILO),LDB,
     +                  Q(ILO+1,ILO),LDQ)
            CALL cungqr(IROWS,IROWS,IROWS,Q(ILO,ILO),LDQ,TAU,WORK,LWORK,
     +                  INFO)
         END IF
*
*        Initialize Z for right eigenvectors
*
         IF (IRIGHT) CALL F06THF('General',N,N,CZERO,CONE,Z,LDZ)
*
*        Compute the generalized Hessenberg form of (A,B)
*
         COMPQ = 'V'
         COMPZ = 'V'
         CALL cgghrd(COMPQ,COMPZ,N,ILO,IHI,A,LDA,B,LDB,Q,LDQ,Z,LDZ,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,*)
*
*        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 chgeqz
*        Workspace query: JWORK = -1
*
         JWORK = -1
         JOB = 'S'
         CALL chgeqz(JOB,COMPQ,COMPZ,N,ILO,IHI,A,LDA,B,LDB,ALPHA,BETA,Q,
     +               LDQ,Z,LDZ,WORK,JWORK,RWORK,INFO)
         WRITE (NOUT,*)
         WRITE (NOUT,99999) NINT(real(WORK(1)))
         WRITE (NOUT,99998) LWORK
         WRITE (NOUT,*)
*
*        Compute the generalized Schur form
*        if the workspace LWORK is adequate
*
         IF (NINT(real(WORK(1))).LE.LWORK) THEN
            CALL chgeqz(JOB,COMPQ,COMPZ,N,ILO,IHI,A,LDA,B,LDB,ALPHA,
     +                  BETA,Q,LDQ,Z,LDZ,WORK,LWORK,RWORK,INFO)
*
*        Print the generalized eigenvalues
*        Note: the actual values of beta are real and non-negative
*
            WRITE (NOUT,99997)
            DO 20 I = 1, N
               IF (real(BETA(I)).NE.0.0e0) THEN
                  E = ALPHA(I)/BETA(I)
                  WRITE (NOUT,99995) I, '(', real(E), ',', imag(E), ')'
               ELSE
                  WRITE (NOUT,99996) I
               END IF
   20       CONTINUE
            WRITE (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
*
            CALL ctgevc(SIDE,HOWMNY,SELECT,N,A,LDA,B,LDB,Q,LDQ,Z,LDZ,N,
     +                  M,WORK,RWORK,INFO)
*
*        Compute right eigenvectors of the original matrix
*
            IF (IRIGHT) THEN
               JOB = 'B'
               SIDE = 'R'
*
               CALL cggbak(JOB,SIDE,N,ILO,IHI,LSCALE,RSCALE,N,Z,LDZ,
     +                     INFO)
*
*        Print the right eigenvectors
*
               IFAIL = 0
               CALL X04DBF('General',' ',N,N,Z,LDZ,'Bracketed','F7.4',
     +                     'Right eigenvectors','Integer',RLABS,
     +                     'Integer',CLABS,80,0,IFAIL)
               WRITE (NOUT,*)
            END IF
*
*        Compute left eigenvectors of the original matrix
*
            IF (IRIGHT) THEN
               JOB = 'B'
               SIDE = 'L'
*
               CALL cggbak(JOB,SIDE,N,ILO,IHI,LSCALE,RSCALE,N,Q,LDQ,
     +                     INFO)
*
*        Print the left eigenvectors
*
               IFAIL = 0
               CALL X04DBF('General',' ',N,N,Q,LDQ,'Bracketed','F7.4',
     +                     'Left eigenvectors','Integer',RLABS,
     +                     'Integer',CLABS,80,0,IFAIL)
            END IF
         ELSE
            WRITE (NOUT,99994)
         END IF
      END IF
      STOP
*
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/',
     +       'chgeqz')
      END
```

9.2 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
```

9.3 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 =     60

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

 Right eigenvectors
                    1                 2                 3                 4
 1  ( 0.0870,-0.1955) ( 0.0550, 0.0318) (-0.5392,-0.2697) ( 0.0467,-0.0597)
 2  (-0.1298, 0.1446) (-0.1060,-0.0705) ( 0.6027, 0.1760) (-0.0801, 0.0956)
 3  ( 0.0480,-0.0520) ( 0.0639, 0.0361) (-0.0726,-0.0274) ( 0.0562,-0.0438)
 4  (-0.0069, 0.0091) (-0.0139,-0.0030) (-0.0042,-0.0007) (-0.0129, 0.0042)

 Left eigenvectors
                    1                 2                 3                 4
 1  (-0.2725,-0.1776) ( 0.0474, 0.0490) (-0.1146,-0.1935) ( 0.0765,-0.0082)
 2  ( 0.2762, 0.0441) (-0.1435,-0.0529) ( 0.3578, 0.2103) (-0.1643, 0.0183)
 3  (-0.0954,-0.0046) ( 0.0864, 0.0136) (-0.0677,-0.0323) ( 0.0952,-0.0048)
 4  ( 0.0128,-0.0019) (-0.0164, 0.0031) ( 0.0094, 0.0034) (-0.0179,-0.0045)
```