# NAG Fortran Library Routine Document

# F08YKF (STGEVC/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 (STGEVC/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,
1                  LDVR, MM, M, WORK, INFO)
 ENTRY      stgevc (SIDE, HOWMNY, SELECT, N, A, LDA, B, LDB, VL, LDVL, VR,
1                  LDVR, MM, M, WORK, INFO)
 INTEGER           N, LDA, LDB, LDVL, LDVR, MM, M, INFO
 real              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

F08YKF (STGEVC/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 (SHGEQZ/DHGEQZ) should be called before invoking F08YKF (STGEVC/DTGEVC).

The right generalized eigenvector $x$ and the left generalized eigenvector $y$ of $(A, B)$ corresponding to a generalized eigenvalue $\lambda$ 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 F08XEF (SHGEQZ/DHGEQZ) but F08YKF (STGEVC/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 (STGEVC/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 the user. Usually, $Q$ and $Z$ are chosen as the orthogonal matrices returned by F08XEF (SHGEQZ/DHGEQZ). Equivalently, $Q$ and $Z$ are the left and right Schur vectors of the matrix pair supplied to F08XEF (SHGEQZ/DHGEQZ). In that case, $QY$ and $ZX$ are the left and right generalized eigenvectors, respectively, of the matrix pair supplied to F08XEF (SHGEQZ/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 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 $\max(1, \mathrm{N})$ if HOWMNY = 'S'; otherwise, this array is not referenced.

*On entry*: specifies the eigenvectors to be computed if HOWMNY = 'S'. To select the generalized eigenvector corresponding to the $j$th generalized eigenvalue, the $j$th 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.

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

5: A(LDA,*) – ***real*** array *Input*

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

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

6: LDA – INTEGER *Input*

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

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

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

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

*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 (SHGEQZ/DHGEQZ).

8: LDB – INTEGER *Input*

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

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

9: VL(LDVL,*) – ***real*** 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 orthogonal matrix $Q$ of left Schur vectors returned by F08XEF (SHGEQZ/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.

10: LDVL – INTEGER *Input*

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

*Constraints*:

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

11: VR(LDVR,*) – ***real*** 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 orthogonal 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.

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.

12: LDVR – INTEGER *Input*

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

*Constraints*:

LDVR $\geq \max(1, \mathrm{N})$ if SIDE = 'R' or 'B';
LDVR $\geq 1$ 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 real eigenvector occupies one column and each selected complex eigenvector occupies two columns.

15: WORK(*) – ***real*** array *Workspace*

**Note:** the dimension of the array WORK must be at least $\max(1, 6 * \mathrm{N})$.

16: 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 $i$th parameter 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 $(\mathrm{INFO} : \mathrm{INFO} + 1)$ does not have complex eigenvalues.

## 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 real generalized eigenvalue problem and is called after F08XEF (SHGEQZ/DHGEQZ).

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

# 9 Example

The example program computes the $\alpha$ and $\beta$ parameters, which defines the generalized eigenvalues and the corresponding left and right eigenvectors, of the matrix pair $(A, B)$ given by

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

and

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

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

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

```
*     F08YKF Example Program Text
*     Mark 20 Release. NAG Copyright 2001.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, LDA, LDB, LWORK
      PARAMETER        (NMAX=10,LDA=NMAX,LDB=NMAX,LWORK=6*NMAX)
      INTEGER          LDQ, LDZ
      PARAMETER        (LDQ=NMAX,LDZ=NMAX)
      real             ONE, ZERO
      PARAMETER        (ONE=1.0e0,ZERO=0.0e0)
*     .. Local Scalars ..
      INTEGER          I, ICOLS, IFAIL, IHI, ILO, INFO, IROWS, J, JWORK,
     +                 M, N
      LOGICAL          ILEFT, IRIGHT
      CHARACTER        COMPQ, COMPZ, HOWMNY, JOB, SIDE
*     .. Local Arrays ..
      real             A(LDA,NMAX), ALPHAI(NMAX), ALPHAR(NMAX),
     +                 B(LDB,NMAX), BETA(NMAX), LSCALE(NMAX),
     +                 Q(LDQ,LDQ), RSCALE(NMAX), TAU(NMAX), WORK(LWORK),
     +                 Z(LDZ,LDZ)
      LOGICAL          SELECT(NMAX)
*     .. External Subroutines ..
      EXTERNAL         sgeqrf, sggbak, sggbal, sgghrd, shgeqz, sorgqr,
     +                 sormqr, stgevc, F06QFF, F06QHF, X04CAF
*     .. Intrinsic Functions ..
      INTRINSIC        NINT
*     .. Executable Statements ..
```

```
      WRITE (NOUT,*) 'F08YKF 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 sggbal(JOB,N,A,LDA,B,LDB,ILO,IHI,LSCALE,RSCALE,WORK,INFO)
*
*        Matrix A after balancing
*
         IFAIL = 0
         CALL X04CAF('General',' ',N,N,A,LDA,'Matrix A after balancing',
     +               IFAIL)
         WRITE (NOUT,*)
*
*        Matrix B after balancing
*
         IFAIL = 0
         CALL X04CAF('General',' ',N,N,B,LDB,'Matrix B after balancing',
     +               IFAIL)
         WRITE (NOUT,*)
*
*        Reduce B to triangular form using QR
*
         IROWS = IHI + 1 - ILO
         ICOLS = N + 1 - ILO
         CALL sgeqrf(IROWS,ICOLS,B(ILO,ILO),LDB,TAU,WORK,LWORK,INFO)
*
*        Apply the orthogonal transformation to matrix A
*
         CALL sormqr('L','T',IROWS,ICOLS,IROWS,B(ILO,ILO),LDB,TAU,
     +               A(ILO,ILO),LDA,WORK,LWORK,INFO)
*
*        Initialize Q (if left eigenvectors are required)
*
         IF (ILEFT) THEN
            CALL F06QHF('General',N,N,ZERO,ONE,Q,LDQ)
            CALL F06QFF('Lower',IROWS-1,IROWS-1,B(ILO+1,ILO),LDB,
     +                  Q(ILO+1,ILO),LDQ)
*
            CALL sorgqr(IROWS,IROWS,IROWS,Q(ILO,ILO),LDQ,TAU,WORK,LWORK,
     +                  INFO)
         END IF
*
*        Initialize Z (if right eigenvectors are required)
*
         IF (IRIGHT) CALL F06QHF('General',N,N,ZERO,ONE,Z,LDZ)
*
*        Compute the generalized Hessenberg form of (A,B)
*
         COMPQ = 'V'
         COMPZ = 'V'
```

```
         CALL sgghrd(COMPQ,COMPZ,N,ILO,IHI,A,LDA,B,LDB,Q,LDQ,Z,LDZ,INFO)
*
*        Matrix A in generalized Hessenberg form
*
         IFAIL = 0
         CALL X04CAF('General',' ',N,N,A,LDA,
     +               'Matrix A in Hessenberg form',IFAIL)
         WRITE (NOUT,*)
*
*        Matrix B in generalized Hessenberg form
*
         IFAIL = 0
         CALL X04CAF('General',' ',N,N,B,LDB,
     +               'Matrix B in Hessenberg form',IFAIL)
*
*        Routine shgeqz
*        Workspace query: JWORK = -1
*
         JWORK = -1
         JOB = 'S'
         CALL shgeqz(JOB,COMPQ,COMPZ,N,ILO,IHI,A,LDA,B,LDB,ALPHAR,
     +               ALPHAI,BETA,Q,LDQ,Z,LDZ,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)).LE.LWORK) THEN
            CALL shgeqz(JOB,COMPQ,COMPZ,N,ILO,IHI,A,LDA,B,LDB,ALPHAR,
     +                  ALPHAI,BETA,Q,LDQ,Z,LDZ,WORK,LWORK,INFO)
*
*        Print the generalized eigenvalues
*
            DO 20 I = 1, N
               IF (BETA(I).NE.0.0e0) THEN
                  WRITE (NOUT,99995) I, '(', ALPHAR(I)/BETA(I), ',',
     +              ALPHAI(I)/BETA(I), ')'
               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 stgevc(SIDE,HOWMNY,SELECT,N,A,LDA,B,LDB,Q,LDQ,Z,LDZ,N,
     +                  M,WORK,INFO)
            IF (IRIGHT) THEN
*
*        Compute right eigenvectors of the original matrix
*
               JOB = 'B'
               SIDE = 'R'
*
               CALL sggbak(JOB,SIDE,N,ILO,IHI,LSCALE,RSCALE,N,Z,LDZ,
```

```
     +                     INFO)
*
*        Print the right eigenvectors
*
               IFAIL = 0
               CALL X04CAF('General',' ',N,N,Z,LDZ,'Right eigenvectors',
     +                     IFAIL)
               WRITE (NOUT,*)
            END IF
*
*        Compute left eigenvectors of the original matrix
*
            IF (ILEFT) THEN
               JOB = 'B'
               SIDE = 'L'
*
               CALL sggbak(JOB,SIDE,N,ILO,IHI,LSCALE,RSCALE,N,Q,LDQ,
     +                     INFO)
*
*        Print the left eigenvectors
*
               IFAIL = 0
               CALL X04CAF('General',' ',N,N,Q,LDQ,'Left eigenvectors',
     +                     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,I4,5X,'Infinite eigenvalue')
99995 FORMAT (1X,I4,5X,A,F7.3,A,F7.3,A)
99994 FORMAT (1X,'Insufficient workspace for array WORK',/' in F08XEF/',
     +       'shgeqz')
      END
```

## 9.2 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
```

## 9.3 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 =     60

 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   -4.9374E-02  -2.0772E-01   2.5702E-02  -7.4074E-02  -6.9466E-02
 2    1.0606E-01   1.7848E-01   8.8325E-02   1.3545E-01   1.3605E-01
 3   -1.0000E-01  -5.3742E-02  -4.6258E-02  -1.0000E-01  -1.0000E-01
 4    4.3761E-02   8.0277E-03   1.3765E-02   2.6455E-02   3.1879E-02
 5   -7.0192E-03  -5.5974E-04  -2.0807E-03  -3.7037E-03  -3.5534E-03

 Left eigenvectors
               1            2            3            4            5
 1   -6.9466E-02  -2.0922E-01  -5.2678E-03  -7.4074E-02   4.9374E-02
 2    1.3605E-01   1.6346E-01   1.1371E-01   1.3545E-01  -1.0606E-01
 3   -1.0000E-01  -4.6314E-02  -5.3686E-02  -1.0000E-01   1.0000E-01
 4    3.1879E-02   5.9054E-03   1.4799E-02   2.6455E-02  -4.3761E-02
 5   -3.5534E-03  -2.4617E-04  -2.1404E-03  -3.7037E-03   7.0192E-03
```