NAG Fortran Library Routine Document

F08XBF (DGGESX)

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

F08XBF (DGGESX) computes the generalized eigenvalues, the generalized real Schur form (S,T), and, optionally the left and/or right generalized Schur vectors for a pair of n by n real nonsymmetric matrices (A,B).

Estimates of condition numbers for selected generalized eigenvalue clusters and Schur vectors are also computed.

2 Specification

```
      SUBROUTINE F08XBF (JOBVSL, JOBVSR, SORT, SELCTG, SENSE, N, A, LDA, B,
     1                   LDB, SDIM, ALPHAR, ALPHAI, BETA, VSL, LDVSL, VSR,
     2                   LDVSR, RCONDE, RCONDV, WORK, LWORK, IWORK, LIWORK,
     3                   BWORK, INFO)
      INTEGER            N, LDA, LDB, SDIM, LDVSL, LDVSR, LWORK, IWORK(*),
     1                   LIWORK, INFO
      double precision   A(LDA,*), B(LDB,*), ALPHAR(*), ALPHAI(*), BETA(*),
     1                   VSL(LDVSL,*), VSR(LDVSR,*), RCONDE(2), RCONDV(2),
     2                   WORK(*)
      LOGICAL            SELCTG, BWORK(*)
      CHARACTER*1        JOBVSL, JOBVSR, SORT, SENSE
      EXTERNAL           SELCTG
```

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

3 Description

The generalized real Schur factorization of (A,B) is given by

$$A = QSZ^T, \quad B = QTZ^T,$$

where Q and Z are orthogonal, T is upper triangular and S is quasi-upper triangular with 1 by 1 and 2 by 2 diagonal blocks. The generalized eigenvalues, λ, of (A,B) are computed from the diagonals of S and T and satisfy

$$Az = \lambda Bz,$$

where z is the corresponding generalized eigenvector. λ is actually returned as the pair (α,β) such that

$$\lambda = \alpha/\beta$$

since β, or even both α and β can be zero. The columns of Q and Z are the left and right generalized Schur vectors of (A,B).

Optionally, F08XAF (DGGES) can order the generalized eigenvalues on the diagonals of (S,T) so that selected eigenvalues are at the top left. The leading columns of Q and Z then form an orthonormal basis for the corresponding eigenspaces, the deflating subspaces.

F08XAF (DGGES) computes T to have non-negative diagonal elements, and the 2 by 2 blocks of S correspond to complex conjugate pairs of generalized eigenvalues. The generalized Schur factorization, before reordering, is computed by the QZ algorithm.

The reciprocals of the condition estimates, the reciprocal values of the left and right projection norms, are returned in RCONDE(1) and RCONDE(2) respectively, for the selected generalized eigenvalues, together with reciprocal condition estimates for the corresponding left and right deflating subspaces, in $\mathrm{RCONDV}(1)$ and $\mathrm{RCONDV}(2)$. See Section 4.11 of Anderson *et al.* (1999) for further information.

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 URL: http://www.netlib.org/lapack/lug

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

5 Parameters

1: JOBVSL – CHARACTER*1 *Input*

On entry: if JOBVSL = 'N', do not compute the left Schur vectors.

If JOBVSL = 'V', compute the left Schur vectors.

2: JOBVSR – CHARACTER*1 *Input*

On entry: if JOBVSR = 'N', do not compute the right Schur vectors.

If JOBVSR = 'V', compute the right Schur vectors.

3: SORT – CHARACTER*1 *Input*

On entry: specifies whether or not to order the eigenvalues on the diagonal of the generalized Schur form:

if SORT = 'N', eigenvalues are not ordered;
if SORT = 'S', eigenvalues are ordered (see SELCTG).

4: SELCTG – LOGICAL FUNCTION, supplied by the user. *External Procedure*

If SORT = 'S', SELCTG is used to select generalized eigenvalues to the top left of the generalized Schur form.

If SORT = 'N', SELCTG is not referenced and F08XBF (DGGESX) may be called with the dummy function F08XAZ.

Its specification is:

```
LOGICAL FUNCTION SELCTG (AR, AI, B)
double precision         AR, AI, B
```

1: AR – ***double precision*** *Input*
2: AI – ***double precision*** *Input*
3: B – ***double precision*** *Input*

On entry: an eigenvalue $\left(\mathrm{AR}(j) + \sqrt{-1} \times \mathrm{AI}(j)\right)/\mathrm{B}(j)$ is selected if $\mathrm{SELCTG}(\mathrm{AR}(j), \mathrm{AI}(j), \mathrm{B}(j))$ is true. If either one of a complex conjugate pair is selected, then both complex generalized eigenvalues are selected.

Note that in the ill-conditioned case, a selected complex generalized eigenvalue may no longer satisfy $\mathrm{SELCTG}(\mathrm{AR}(j), \mathrm{AI}(j), \mathrm{B}(j)) =$ .TRUE. after ordering. INFO is set to $\mathrm{N} + 2$ in this case. (See INFO below).

SELCTG must be declared as EXTERNAL in the (sub)program from which F08XBF (DGGESX) is called. Parameters denoted as *Input* must **not** be changed by this procedure.

5: SENSE – CHARACTER*1 *Input*

On entry: determines which reciprocal condition numbers are computed:

if SENSE = 'N', none are computed;
if SENSE = 'E', computed for average of selected eigenvalues only;
if SENSE = 'V', computed for selected deflating subspaces only;
if SENSE = 'B', computed for both.

If SENSE = 'E', 'V' or 'B', SORT must equal 'S'.

6: N – INTEGER *Input*

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

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

7: A(LDA,$*$) – ***double precision*** array *Input/Output*

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

On entry: the first of the pair of matrices, A.

On exit: has been overwritten by its generalized Schur form S.

8: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08XBF (DGGESX) is called.

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

9: B(LDB,$*$) – ***double precision*** array *Input/Output*

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

On entry: the second of the pair of matrices, B.

On exit: has been overwritten by its generalized Schur form T.

10: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F08XBF (DGGESX) is called.

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

11: SDIM – INTEGER *Output*

On exit: if SORT = 'N', SDIM = 0.

If SORT = 'S', SDIM = number of eigenvalues (after sorting) and for which SELCTG is true. (Complex conjugate pairs for which SELCTG is true for either eigenvalue count as 2.)

12: ALPHAR($*$) – ***double precision*** array *Output*

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

On exit: see the description of BETA below.

13: ALPHAI($*$) – ***double precision*** array *Output*

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

On exit: see the description of BETA below.

14: BETA($*$) – ***double precision*** array *Output*

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

On exit: $(\mathrm{ALPHAR}(j) + \mathrm{ALPHAI}(j) \times i)/\mathrm{BETA}(j)$, $j = 1, \ldots, \mathrm{N}$, will be the generalized eigenvalues. $\mathrm{ALPHAR}(j) + \mathrm{ALPHAI}(j) \times i$, and $\mathrm{BETA}(j)$, $j = 1, \ldots, \mathrm{N}$ are the diagonals of the complex Schur form (S, T) that would result if the 2 by 2 diagonal blocks of the real Schur form of (A, B) were further reduced to triangular form using 2 by 2 complex unitary transformations.

If $\mathrm{ALPHAI}(j)$ is zero, then the jth eigenvalue is real; if positive, then the jth and $(j+1)$st eigenvalues are a complex conjugate pair, with $\mathrm{ALPHAI}(j+1)$ negative.

Note: the quotients $\mathrm{ALPHAR}(j)/\mathrm{BETA}(j)$ and $\mathrm{ALPHAI}(j)/\mathrm{BETA}(j)$ may easily over- or underflow, and $\mathrm{BETA}(j)$ may even be zero. Thus, the user should avoid naively computing the ratio α/β. However, ALPHAR and ALPHAI will be always less than and usually comparable with $\|\mathrm{A}\|_2$ in magnitude, and BETA always less than and usually comparable with $\|\mathrm{B}\|_2$.

15: VSL(LDVSL,$*$) – ***double precision*** array *Output*

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

On exit: if JOBVSL = 'V', VSL will contain the left Schur vectors, Q.

If JOBVSL = 'N', VSL is not referenced.

16: LDVSL – INTEGER *Input*

On entry: the first dimension of the array VSL as declared in the (sub)program from which F08XBF (DGGESX) is called.

Constraints:

if JOBVSL = 'V', LDVSL $\geq \max(1, \mathrm{N})$;
LDVSL ≥ 1 otherwise.

17: VSR(LDVSR,$*$) – ***double precision*** array *Output*

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

On exit: if JOBVSR = 'V', VSR will contain the right Schur vectors, Z.

If JOBVSR = 'N', VSR is not referenced.

18: LDVSR – INTEGER *Input*

On entry: the first dimension of the array VSR as declared in the (sub)program from which F08XBF (DGGESX) is called.

Constraints:

if JOBVSR = 'V', LDVSR $\geq \max(1, \mathrm{N})$;
LDVSR ≥ 1 otherwise.

19: RCONDE(2) – ***double precision*** array *Output*

On exit: if SENSE = 'E' or 'B', RCONDE(1) and RCONDE(2) contain the reciprocal condition numbers for the average of the selected eigenvalues.

If SENSE = 'N' or 'V', RCONDE is not referenced.

20: RCONDV(2) – ***double precision*** array *Output*

On exit: if SENSE = 'V' or 'B', RCONDV(1) and RCONDV(2) contain the reciprocal condition numbers for the selected deflating subspaces.

if SENSE = 'N' or 'E', RCONDV is not referenced.

21: WORK($*$) – ***double precision*** array *Workspace*

Note: the dimension of the array WORK must be at least $\max(1, \text{LWORK})$.

On exit: if INFO $= 0$, WORK(1) returns the optimal LWORK.

22: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08XBF (DGGESX) is called.

If LWORK $= -1$, a workspace query is assumed; the routine only calculates the bound on the optimal size of the WORK array and the minimum size of the IWORK array, returns these values as the first entries of the WORK and IWORK arrays, and no error message related to LWORK or LIWORK is issued.

Constraints:

if N $= 0$, LWORK $\geq 8 \times (\text{N} + 1) + 16$;
if SENSE $=$ 'E', 'V' or 'B', LWORK $\geq \max(8 \times (\text{N} + 1) + 16, 2 \times \text{SDIM} \times (\text{N} - \text{SDIM}))$.

Note: that $2 \times \text{SDIM} \times (\text{N} - \text{SDIM}) \leq \text{N} \times \text{N}/2$. Note also that an error is only returned if LWORK $< 8 \times (\text{N} + 1) + 16$, but if SENSE $=$ 'E', 'V' or 'B' this may not be large enough. Consider increasing LWORK by nb, where nb is the block size.

23: IWORK($*$) – INTEGER array *Output*

Note: the dimension of the array IWORK must be at least $\max(1, \text{LIWORK})$.

On exit: if INFO $= 0$, IWORK(1) returns the minimum LIWORK.

24: LIWORK – INTEGER *Input*

On entry: the dimension of the array IWORK as declared in the (sub)program from which F08XBF (DGGESX) is called.

If LIWORK $= -1$, a workspace query is assumed; the routine only calculates the bound on the optimal size of the WORK array and the minimum size of the IWORK array, returns these values as the first entries of the WORK and IWORK arrays, and no error message related to LWORK or LIWORK is issued.

Constraints:

if SENSE $=$ 'N' or N $= 0$, LIWORK ≥ 1;
LIWORK $\geq \text{N} + 6$ otherwise.

25: BWORK($*$) – LOGICAL array *Workspace*

Note: the dimension of the array BWORK must be at least 1 if SORT $=$ 'N' and at least $\max(1, \text{N})$ otherwise.

If SORT $=$ 'N', BWORK is not referenced.

26: 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 argument had an illegal value.

INFO = 1 to N

> The QZ iteration failed. (A, B) are not in Schur form, but ALPHAR(j), ALPHAI(j), and BETA(j) should be correct for $j = \text{INFO} + 1, \ldots, \text{N}$.

INFO > N

> $= \text{N} + 1$: other than QZ iteration failed in F08XEF (DHGEQZ)
>
> $= \text{N} + 2$: after reordering, roundoff changed values of some complex eigenvalues so that leading eigenvalues in the generalized Schur form no longer satisfy SELCTG = .TRUE.. This could also be caused due to scaling.
>
> $= \text{N} + 3$: reordering failed because some eigenvalues were too close to separate (the problem is very ill-conditioned).

7 Accuracy

The computed generalized Schur factorization satisfies

$$A + E = QSZ^T, \quad B + F = QTZ^T,$$

where

$$\|(E, F)\|_F = \mathrm{O}(\epsilon)\|(A, B)\|_F$$

and ϵ is the ***machine precision***. See Section 4.11 of Anderson *et al.* (1999) for further details.

8 Further Comments

The total number of floating-point operations is proportional to n^3.

The complex analogue of this routine is F08XPF (ZGGESX).

9 Example

To find the generalized Schur factorization of the matrix pair (A, B), where

$$A = \begin{pmatrix} 3.9 & 12.5 & -34.5 & -0.5 \\ 4.3 & 21.5 & -47.5 & 7.5 \\ 4.3 & 21.5 & -43.5 & 3.5 \\ 4.4 & 26.0 & -46.0 & 6.0 \end{pmatrix}$$

and

$$B = \begin{pmatrix} 1.0 & 2.0 & -3.0 & 1.0 \\ 1.0 & 3.0 & -5.0 & 4.0 \\ 1.0 & 3.0 & -4.0 & 3.0 \\ 1.0 & 3.0 & -4.0 & 4.0 \end{pmatrix},$$

such that the real eigenvalues of (A, B) correspond to the top left diagonal elements of the generalized Schur form, (S, T). Estimates of the condition numbers for the selected eigenvalue cluster and corresponding deflating subspaces are also returned.

Note that the block size (NB) of 64 assumed in this example is not realistic for such a small problem, but should be suitable for large problems.

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.

```
*     F08XBF Example Program Text
*     Mark 21 Release. NAG Copyright 2004.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NB, NMAX
      PARAMETER        (NB=64,NMAX=10)
      INTEGER          LDA, LDB, LDVSL, LDVSR, LIWORK, LWORK
      PARAMETER        (LDA=NMAX,LDB=NMAX,LDVSL=NMAX,LDVSR=NMAX,
     +                 LIWORK=NMAX+6,LWORK=8*(NMAX+1)
     +                 +16+NMAX*NB+NMAX*NMAX/2)
*     .. Local Scalars ..
      DOUBLE PRECISION ABNORM, ANORM, BNORM, EPS, TOL
      INTEGER          I, IFAIL, INFO, J, LWKOPT, N, SDIM
*     .. Local Arrays ..
      DOUBLE PRECISION A(LDA,NMAX), ALPHAI(NMAX), ALPHAR(NMAX),
     +                 B(LDB,NMAX), BETA(NMAX), RCONDE(2), RCONDV(2),
     +                 VSL(LDVSL,NMAX), VSR(LDVSR,NMAX), WORK(LWORK)
      INTEGER          IWORK(LIWORK)
      LOGICAL          BWORK(NMAX)
*     .. External Functions ..
      DOUBLE PRECISION F06BNF, F06RAF, X02AJF
      LOGICAL          DELCTG
      EXTERNAL         F06BNF, F06RAF, X02AJF, DELCTG
*     .. External Subroutines ..
      EXTERNAL         DGGESX, X04CAF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08XBF Example Program Results'
      WRITE (NOUT,*)
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      IF (N.LE.NMAX) THEN
*
*        Read in the matrices A and B
*
         READ (NIN,*) ((A(I,J),J=1,N),I=1,N)
         READ (NIN,*) ((B(I,J),J=1,N),I=1,N)
*
*        Find the Frobenius norms of A and B
*
         ANORM = F06RAF('Frobenius',N,N,A,LDA,WORK)
         BNORM = F06RAF('Frobenius',N,N,B,LDB,WORK)
*
*        Find the generalized Schur form
*
         CALL DGGESX('Vectors (left)','Vectors (right)','Sort',DELCTG,
     +               'Both reciprocal condition numbers',N,A,LDA,B,LDB,
     +               SDIM,ALPHAR,ALPHAI,BETA,VSL,LDVSL,VSR,LDVSR,RCONDE,
     +               RCONDV,WORK,LWORK,IWORK,LIWORK,BWORK,INFO)
*
         IF (INFO.GT.0 .AND. INFO.NE.(N+2)) THEN
            WRITE (NOUT,99999) 'Failure in DGGESX. INFO =', INFO
         ELSE
            WRITE (NOUT,99999)
     +        'Number of eigenvalues for which DELCTG is true = ', SDIM,
     +        '(dimension of deflating subspaces)'
            WRITE (NOUT,*)
            IF (INFO.EQ.(N+2)) THEN
               WRITE (NOUT,99998) '***Note that rounding errors mean ',
     +           'that leading eigenvalues in the generalized',
     +           'Schur form no longer satisfy DELCTG = .TRUE.'
               WRITE (NOUT,*)
            END IF
*
```

```
*           Print out the factors of the generalized Schur factorization
*
            IFAIL = 0
            CALL X04CAF('General',' ',N,N,A,LDA,
     +                  'Generalized Schur matrix S',IFAIL)
*
            WRITE (NOUT,*)
            CALL X04CAF('General',' ',N,N,B,LDB,
     +                  'Generalized Schur matrix T',IFAIL)
*
            WRITE (NOUT,*)
            CALL X04CAF('General',' ',N,N,VSL,LDVSL,
     +                  'Matrix of left generalized Schur vectors',
     +                  IFAIL)
*
            WRITE (NOUT,*)
            CALL X04CAF('General',' ',N,N,VSR,LDVSR,
     +                  'Matrix of right generalized Schur vectors',
     +                  IFAIL)
*
*           Print out the reciprocal condition numbers
*
            WRITE (NOUT,*)
            WRITE (NOUT,99997)
     +        'Reciprocals of left and right projection norms onto',
     +        'the deflating subspaces for the selected eigenvalues',
     +        'RCONDE(1) = ', RCONDE(1), ', RCONDE(2) = ', RCONDE(2)
            WRITE (NOUT,*)
            WRITE (NOUT,99997)
     +        'Reciprocal condition numbers for the left and right',
     +        'deflating subspaces', 'RCONDV(1) = ', RCONDV(1),
     +        ', RCONDV(2) = ', RCONDV(2)
*
*           Compute the machine precision and sqrt(ANORM**2+BNORM**2)
*
            EPS = X02AJF()
            ABNORM = F06BNF(ANORM,BNORM)
            TOL = EPS*ABNORM
*
*           Print out the approximate asymptotic error bound on the
*           average absolute error of the selected eigenvalues given by
*
*              eps*norm((A, B))/PL,   where PL = RCONDE(1)
*
            WRITE (NOUT,*)
            WRITE (NOUT,99996)
     +        'Approximate asymptotic error bound for selected ',
     +        'eigenvalues    = ', TOL/RCONDE(1)
*
*           Print out an approximate asymptotic bound on the maximum
*           angular error in the computed deflating subspaces given by
*
*              eps*norm((A, B))/DIF(2),   where DIF(2) = RCONDV(2)
*
            WRITE (NOUT,99996)
     +        'Approximate asymptotic error bound for the deflating ',
     +        'subspaces = ', TOL/RCONDV(2)
*
            LWKOPT = WORK(1)
            IF (LWORK.LT.LWKOPT) THEN
               WRITE (NOUT,*)
               WRITE (NOUT,99995) 'Optimum workspace required = ',
     +           LWKOPT, 'Workspace provided         = ', LWORK
            END IF
         END IF
      ELSE
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'NMAX too small'
      END IF
      STOP
*
```

```
99999 FORMAT (1X,A,I4,/1X,A)
99998 FORMAT (1X,2A,/1X,A)
99997 FORMAT (1X,A,/1X,A,/1X,2(A,1P,E8.1))
99996 FORMAT (1X,2A,1P,E8.1)
99995 FORMAT (1X,A,I5,/1X,A,I5)
      END

      LOGICAL FUNCTION DELCTG(AR,AI,B)
*     .. Scalar Arguments ..
*
*     Logical function DELCTG for use with DGGESX (F08XBF)
*
*     Returns the value .TRUE. if the imaginary part of the eigenvalue
*     (AR + AI*i)/B is zero, i.e. the eigenvalue is real
*
      DOUBLE PRECISION        AI, AR, B
*     .. Local Scalars ..
      LOGICAL                 D
*     .. Executable Statements ..
      IF (AI.EQ.0.0D0) THEN
         D = .TRUE.
      ELSE
         D = .FALSE.
      END IF
*
      DELCTG = D
*
      RETURN
      END
```

9.2 Program Data

```
F08XBF Example Program Data
   4                     :Value of N
   3.9  12.5 -34.5  -0.5
   4.3  21.5 -47.5   7.5
   4.3  21.5 -43.5   3.5
   4.4  26.0 -46.0   6.0 :End of matrix A
   1.0   2.0  -3.0   1.0
   1.0   3.0  -5.0   4.0
   1.0   3.0  -4.0   3.0
   1.0   3.0  -4.0   4.0 :End of matrix B
```

9.3 Program Results

```
 F08XBF Example Program Results

 Number of eigenvalues for which DELCTG is true =    2
 (dimension of deflating subspaces)

 Generalized Schur matrix S
             1          2          3          4
 1      3.8009   -69.4505    50.3135   -43.2884
 2      0.0000     9.2033    -0.2001     5.9881
 3      0.0000     0.0000     1.4279     4.4453
 4      0.0000     0.0000     0.9019    -1.1962

 Generalized Schur matrix T
             1          2          3          4
 1      1.9005   -10.2285     0.8658    -5.2134
 2      0.0000     2.3008     0.7915     0.4262
 3      0.0000     0.0000     0.8101     0.0000
 4      0.0000     0.0000     0.0000    -0.2823

 Matrix of left generalized Schur vectors
          1       2       3       4
 1   0.4642  0.7886  0.2915 -0.2786
 2   0.5002 -0.5986  0.5638 -0.2713
 3   0.5002  0.0154 -0.0107  0.8657
 4   0.5331 -0.1395 -0.7727 -0.3151
 Matrix of right generalized Schur vectors
          1       2       3       4
 1   0.9961 -0.0014  0.0887 -0.0026
 2   0.0057 -0.0404 -0.0938 -0.9948
 3   0.0626  0.7194 -0.6908  0.0363
 4   0.0626 -0.6934 -0.7114  0.0956

 Reciprocals of left and right projection norms onto
 the deflating subspaces for the selected eigenvalues
 RCONDE(1) =  1.9E-01, RCONDE(2) =  1.8E-02

 Reciprocal condition numbers for the left and right
 deflating subspaces
 RCONDV(1) =  5.4E-02, RCONDV(2) =  9.0E-02

 Approximate asymptotic error bound for selected eigenvalues    =  5.7E-14
 Approximate asymptotic error bound for the deflating subspaces =  1.2E-13
```