NAG Fortran Library Routine Document

F08SNF (ZHEGV)

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

F08SNF (ZHEGV) computes all the eigenvalues, and optionally, the eigenvectors of a complex generalized Hermitian-definite eigenproblem, of the form

$$Az = \lambda Bz, \quad ABz = \lambda z \quad \text{or} \quad BAz = \lambda z,$$

where A and B are Hermitian and B is also positive-definite.

2 Specification

```
      SUBROUTINE F08SNF (ITYPE, JOBZ, UPLO, N, A, LDA, B, LDB, W, WORK, LWORK,
     1                   RWORK, INFO)
      INTEGER            ITYPE, N, LDA, LDB, LWORK, INFO
      double precision   W(*), RWORK(*)
      complex*16         A(LDA,*), B(LDB,*), WORK(*)
      CHARACTER*1        JOBZ, UPLO
```

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

3 Description

F08SNF (ZHEGV) first performs a Cholesky factorization of the matrix B as $B = U^H U$, when UPLO = 'U' or $B = LL^H$, when UPLO = 'L'. The generalized problem is then reduced to a standard symmetric eigenvalue problem

$$Cx = \lambda x,$$

which is solved for the eigenvalues and, optionally, the eigenvectors; the eigenvectors are then backtransformed to give the eigenvectors of the original problem.

For the problem $Az = \lambda Bz$, the eigenvectors are normalized so that the matrix of eigenvectors, z, satisfies

$$Z^H AZ = \Lambda \quad \text{and} \quad Z^H BZ = I,$$

where Λ is the diagonal matrix whose diagonal elements are the eigenvalues. For the problem $ABz = \lambda z$ we correspondingly have

$$Z^{-1} AZ^{-H} = \Lambda \quad \text{and} \quad Z^H BZ = I,$$

and for $BAz = \lambda z$ we have

$$Z^H AZ = \Lambda \quad \text{and} \quad Z^H B^{-1} Z = I.$$

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: ITYPE – INTEGER *Input*

On entry: specifies the problem type to be solved:

if ITYPE = 1, $Az = \lambda Bz$;
if ITYPE = 2, $ABz = \lambda z$;
if ITYPE = 3, $BAz = \lambda z$.

2: JOBZ – CHARACTER*1 *Input*

On entry: if JOBZ = 'N', compute eigenvalues only.

If JOBZ = 'V', compute eigenvalues and eigenvectors.

Constraint: JOBZ = 'N' or 'V'.

3: UPLO – CHARACTER*1 *Input*

On entry: if UPLO = 'U', the upper triangles of A and B are stored.

If UPLO = 'L', the lower triangles of A and B are stored.

4: N – INTEGER *Input*

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

Constraint: N $\geq$ 0.

5: A(LDA,$*$) – ***complex*16*** array *Input/Output*

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

On entry: the Hermitian matrix A.

If UPLO = 'U', the leading n by n upper triangular part of A contains the upper triangular part of the matrix A.

If UPLO = 'L', the leading n by n lower triangular part of A contains the lower triangular part of the matrix A.

On exit: if JOBZ = 'V', then if INFO = 0, A contains the matrix Z of eigenvectors. The eigenvectors are normalized as follows:

if ITYPE = 1 or 2, $Z^H BZ = I$;
if ITYPE = 3, $Z^H B^{-1} Z = I$.

If JOBZ = 'N', the upper triangle (if UPLO = 'U') or the lower triangle (if UPLO = 'L') of A, including the diagonal, is destroyed.

6: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08SNF (ZHEGV) is called.

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

7: B(LDB,$*$) – ***complex*16*** array *Input/Output*

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

On entry: the Hermitian positive-definite matrix B:

if UPLO = 'U', the leading n by n upper triangular part of B contains the upper triangular part of the matrix B;
if UPLO = 'L', the leading n by n lower triangular part of B contains the lower triangular part of the matrix B.

On exit: if INFO $\leq$ N, the part of B containing the matrix is overwritten by the triangular factor U or L from the Cholesky factorization $B = U^H U$ or $B = LL^H$.

8: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F08SNF (ZHEGV) is called.

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

9: W($*$) – ***double precision*** array *Output*

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

On exit: if INFO $= 0$, the eigenvalues in ascending order.

10: WORK($*$) – ***complex*16*** 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.

11: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08SNF (ZHEGV) is called.

For optimal efficiency, LWORK $\geq (nb + 1) \times N$, where nb is the optimal block size for F08FSF (ZHETRD).

If LWORK $= -1$, a workspace query is assumed; the routine only calculates the optimal size of the WORK array, returns this value as the first entry of the WORK array, and no error message related to LWORK is issued.

Constraint: LWORK $\geq \max(1, 2 \times N)$.

12: RWORK($*$) – ***double precision*** array *Workspace*

Note: the dimension of the array RWORK must be at least $\max(1, 3 \times N - 2)$.

13: 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 > 0

F07FRF (ZPOTRF) or F08FNF (ZHEEV) returned an error code:

$\leq$ N if INFO $= i$, F08FNF (ZHEEV) failed to converge; i off-diagonal elements of an intermediate tridiagonal form did not converge to zero;

$>$ N if INFO $= N + i$, for $1 \leq i \leq N$, then the leading minor of order i of B is not positive-definite. The factorization of B could not be completed and no eigenvalues or eigenvectors were computed.

7 Accuracy

If B is ill-conditioned with respect to inversion, then the error bounds for the computed eigenvalues and vectors may be large, although when the diagonal elements of B differ widely in magnitude the eigenvalues and eigenvectors may be less sensitive than the condition of B would suggest. See Section 4.10 of Anderson *et al.* (1999) for details of the error bounds.

The example program below illustrates the computation of approximate error bounds.

8 Further Comments

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

The real analogue of this routine is F08SAF (DSYGV).

9 Example

To find all the eigenvalues and eigenvectors of the generalized Hermitian eigenproblem $Az = \lambda Bz$, where

$$A = \begin{pmatrix} -7.36 & 0.77 - 0.43i & -0.64 - 0.92i & 3.01 - 6.97i \\ 0.77 + 0.43i & 3.49 & 2.19 + 4.45i & 1.90 + 3.73i \\ -0.64 + 0.92i & 2.19 - 4.45i & 0.12 & 2.88 - 3.17i \\ 3.01 + 6.97i & 1.90 - 3.73i & 2.88 + 3.17i & -2.54 \end{pmatrix}$$

and

$$B = \begin{pmatrix} 3.23 & 1.51 - 1.92i & 1.90 + 0.84i & 0.42 + 2.50i \\ 1.51 + 1.92i & 3.58 & -0.23 + 1.11i & -1.18 + 1.37i \\ 1.90 - 0.84i & -0.23 - 1.11i & 4.09 & 2.33 - 0.14i \\ 0.42 - 2.50i & -1.18 - 1.37i & 2.33 + 0.14i & 4.29 \end{pmatrix},$$

together with and estimate of the condition number of B, and approximate error bounds for the computed eigenvalues and eigenvectors.

The example program for F08SQF (ZHEGVD) illustrates solving a generalized Hermitian eigenproblem of the form $ABz = \lambda z$.

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.

```
*     F08SNF Example Program Text
*     Mark 21.  NAG Copyright 2004.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NB, NMAX
      PARAMETER        (NB=64,NMAX=10)
      INTEGER          LDA, LDB, LWORK
      PARAMETER        (LDA=NMAX,LDB=NMAX,LWORK=(NB+1)*NMAX)
*     .. Local Scalars ..
      DOUBLE PRECISION ANORM, BNORM, EPS, RCOND, RCONDB, T1, T2, T3
      INTEGER          I, IFAIL, INFO, J, LWKOPT, N
*     .. Local Arrays ..
      COMPLEX *16      A(LDA,NMAX), B(LDB,NMAX), WORK(LWORK)
      DOUBLE PRECISION EERBND(NMAX), RCONDZ(NMAX), RWORK(3*NMAX-2),
     +                 W(NMAX), ZERBND(NMAX)
*     .. External Functions ..
      DOUBLE PRECISION F06UCF, X02AJF
      EXTERNAL         F06UCF, X02AJF
*     .. Intrinsic Functions ..
      INTRINSIC        ABS
*     .. External Subroutines ..
      EXTERNAL         DDISNA, X04DAF, ZHEGV, ZTRCON
```

```
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08SNF Example Program Results'
      WRITE (NOUT,*)
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      IF (N.LE.NMAX) THEN
*
*        Read the upper triangular parts of the matrices A and B
*
         READ (NIN,*) ((A(I,J),J=I,N),I=1,N)
         READ (NIN,*) ((B(I,J),J=I,N),I=1,N)
*
*        Compute the one-norms of the symmetric matrices A and B
*
         ANORM = F06UCF('One norm','Upper',N,A,LDA,RWORK)
         BNORM = F06UCF('One norm','Upper',N,B,LDB,RWORK)
*
*        Solve the generalized Hermitian eigenvalue problem
*        A*x = lambda*B*x (ITYPE = 1)
*
         CALL ZHEGV(1,'Vectors','Upper',N,A,LDA,B,LDB,W,WORK,LWORK,
     +              RWORK,INFO)
         LWKOPT = WORK(1)
*
         IF (INFO.EQ.0) THEN
*
*           Print solution
*
            WRITE (NOUT,*) 'Eigenvalues'
            WRITE (NOUT,99999) (W(J),J=1,N)
*
            IFAIL = 0
            CALL X04DAF('General',' ',N,N,A,LDA,'Eigenvectors',IFAIL)
*
*           Call ZTRCON (F07TUF) to estimate the reciprocal condition
*           number of the Cholesky factor of B.  Note that:
*           cond(B) = 1/RCOND**2
*
            CALL ZTRCON('One norm','Upper','Non-unit',N,B,LDB,RCOND,
     +                  WORK,RWORK,INFO)
*
*           Print the reciprocal condition number of B
*
            RCONDB = RCOND**2
            WRITE (NOUT,*)
            WRITE (NOUT,*)
     +        'Estimate of reciprocal condition number for B'
            WRITE (NOUT,99998) RCONDB
*
*           Get the machine precision, EPS, and if RCONDB is not less
*           than EPS**2, compute error estimates for the eigenvalues and
*           eigenvectors
*
            EPS = X02AJF()
            IF (RCOND.GE.EPS) THEN
*
*              Call DDISNA (F08FLF) to estimate reciprocal condition
*              numbers for the eigenvectors of (A - lambda*B)
*
               CALL DDISNA('Eigenvectors',N,N,W,RCONDZ,INFO)
*
*              Compute the error estimates for the eigenvalues and
*              eigenvectors
*
               T1 = EPS/RCONDB
               T2 = ANORM/BNORM
               T3 = T2/RCOND
               DO 20 I = 1, N
                  EERBND(I) = T1*(T2+ABS(W(I)))
                  ZERBND(I) = T1*(T3+ABS(W(I)))/RCONDZ(I)
```

```
   20          CONTINUE
*
*              Print the approximate error bounds for the eigenvalues
*              and vectors
*
               WRITE (NOUT,*)
               WRITE (NOUT,*) 'Error estimates for the eigenvalues'
               WRITE (NOUT,99998) (EERBND(I),I=1,N)
               WRITE (NOUT,*)
               WRITE (NOUT,*) 'Error estimates for the eigenvectors'
               WRITE (NOUT,99998) (ZERBND(I),I=1,N)
            ELSE
               WRITE (NOUT,*)
               WRITE (NOUT,*) 'B is very ill-conditioned, error ',
     +           'estimates have not been computed'
            END IF
         ELSE IF (INFO.GT.N) THEN
            I = INFO - N
            WRITE (NOUT,99997) 'The leading minor of order ', I,
     +        ' of B is not positive definite'
         ELSE
            WRITE (NOUT,99996) 'Failure in ZHEGV. INFO =', INFO
         END IF
*
*        Print workspace information
*
         IF (LWORK.LT.LWKOPT) THEN
            WRITE (NOUT,*)
            WRITE (NOUT,99995) 'Optimum workspace required = ', LWKOPT,
     +        'Workspace provided         = ', LWORK
         END IF
      ELSE
         WRITE (NOUT,*) 'NMAX too small'
      END IF
      STOP
*
99999 FORMAT (3X,(6F11.4))
99998 FORMAT (4X,1P,6E11.1)
99997 FORMAT (1X,A,I4,A)
99996 FORMAT (1X,A,I4)
99995 FORMAT (1X,A,I5,/1X,A,I5)
      END
```

9.2 Program Data

```
F08SNF Example Program Data

   4                                                       :Value of N

 (-7.36, 0.00) ( 0.77, -0.43) (-0.64, -0.92) ( 3.01, -6.97)
               ( 3.49,  0.00) ( 2.19,  4.45) ( 1.90,  3.73)
                              ( 0.12,  0.00) ( 2.88, -3.17)
                                             (-2.54,  0.00) :End of matrix A

 ( 3.23, 0.00) ( 1.51, -1.92) ( 1.90,  0.84) ( 0.42,  2.50)
               ( 3.58,  0.00) (-0.23,  1.11) (-1.18,  1.37)
                              ( 4.09,  0.00) ( 2.33, -0.14)
                                             ( 4.29,  0.00) :End of matrix B
```

9.3 Program Results

```
 F08SNF Example Program Results

 Eigenvalues
      -5.9990    -2.9936     0.5047     3.9990
 Eigenvectors
            1          2          3          4
 1     1.7372     0.4889     0.6164     0.2310
       0.1062    -0.5010     0.1937    -1.2161
```

```
 2      -0.3843     0.1118     0.2596    -0.4710
        -0.4933    -0.0367    -0.4203     0.4814

 3      -0.8237    -0.8115    -0.0365    -0.2242
        -0.2991     0.4114    -0.3321     0.6335

 4       0.2643     0.7877     0.0994     0.8515
         0.6276     0.2002     0.6588     0.0000

 Estimate of reciprocal condition number for B
         2.5E-03

 Error estimates for the eigenvalues
         3.4E-13    2.0E-13    9.6E-14    2.5E-13

 Error estimates for the eigenvectors
         5.8E-13    5.3E-13    4.3E-13    4.7E-13
```