NAG Library Routine Document

F08SQF (ZHEGVD)

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

F08SQF (ZHEGVD) computes all the eigenvalues and, optionally, the eigenvectors of a complex generalized Hermitian-definite eigenproblem, of the form

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

where A and B are Hermitian and B is also positive-definite. If eigenvectors are desired, it uses a divide-and-conquer algorithm.

2 Specification

```
 SUBROUTINE F08SQF(ITYPE, JOBZ, UPLO, N, A, LDA, B, LDB, W, WORK, LWORK,
1                  RWORK, LRWORK, IWORK, LIWORK, INFO)
 INTEGER           ITYPE, N, LDA, LDB, LWORK, LRWORK,
1                  IWORK(max(1,LIWORK)), LIWORK, INFO
 double precision  W(N), RWORK(max(1,LRWORK))
 complex*16        A(LDA,*), B(LDB,*), WORK(max(1,LWORK))
 CHARACTER*1       JOBZ, UPLO
```

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

3 Description

F08SQF (ZHEGVD) first performs a Cholesky factorization of the matrix B as $B = U^{\mathrm{H}}U$, when UPLO = 'U' or $B = LL^{\mathrm{H}}$, 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

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

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

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

and for $BAz = \lambda z$ we have

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

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, Philadelphiahttp://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.

ITYPE $= 1$
$Az = \lambda Bz$.

ITYPE $= 2$
$ABz = \lambda z$.

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.

Constraint: UPLO = 'U' or 'L'.

4: N – INTEGER *Input*

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

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

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 n by n Hermitian matrix A.

If UPLO = 'U', the upper triangular part of A must be stored and the elements of the array below the diagonal are not referenced.

If UPLO = 'L', the lower triangular part of A must be stored and the elements of the array above the diagonal are not referenced.

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, $Z^{\mathrm{H}}BZ = I$;

if ITYPE $= 3$, $Z^{\mathrm{H}}B^{-1}Z = I$.

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 F08SQF (ZHEGVD) is called.

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

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 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 $\mathrm{B} = U^{\mathrm{H}}U$ or $\mathrm{B} = LL^{\mathrm{H}}$.

8: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F08SQF (ZHEGVD) is called.

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

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

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

10: WORK($\max(1, \mathrm{LWORK})$) – ***complex*16*** array *Workspace*

On exit: if INFO $= 0$, the real part of WORK(1) contains the minimum value of LWORK required for optimal performance.

11: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08SQF (ZHEGVD) is called.

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

Suggested value: for optimal performance, LWORK should usually be larger than the minimum, try increasing by $nb \times \mathrm{N}$, where nb is the optimal ***block size***.

Constraints:

if $\mathrm{N} \leq 1$, LWORK ≥ 1;
if JOBZ = 'N' and $\mathrm{N} > 1$, LWORK $\geq \mathrm{N} + 1$;
if JOBZ = 'V' and $\mathrm{N} > 1$, LWORK $\geq 2 \times \mathrm{N} + \mathrm{N}^2$.

12: RWORK($\max(1, \mathrm{LRWORK})$) – ***double precision*** array *Workspace*

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

13: LRWORK – INTEGER *Input*

On entry: the dimension of the array RWORK as declared in the (sub)program from which F08SQF (ZHEGVD) is called.

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

Constraints:

if $\mathrm{N} \leq 1$, $\mathrm{LRWORK} \geq 1$;
if $\mathrm{JOBZ} = \text{'N'}$ and $\mathrm{N} > 1$, $\mathrm{LRWORK} \geq \mathrm{N}$;
if $\mathrm{JOBZ} = \text{'V'}$ and $\mathrm{N} > 1$, $\mathrm{LRWORK} \geq 1 + 5 \times \mathrm{N} + 2 \times \mathrm{N}^2$.

14: IWORK$(\max(1, \mathrm{LIWORK}))$ – INTEGER array *Workspace*

On exit: if $\mathrm{INFO} = 0$, $\mathrm{IWORK}(1)$ returns the optimal LIWORK.

15: LIWORK – INTEGER *Input*

On entry: the dimension of the array IWORK as declared in the (sub)program from which F08SQF (ZHEGVD) is called.

If $\mathrm{LIWORK} = -1$, a workspace query is assumed; the routine only calculates the optimal sizes of the WORK, RWORK and IWORK arrays, returns these values as the first entries of the WORK, RWORK and IWORK arrays, and no error message related to LWORK, LRWORK or LIWORK is issued.

Constraints:

if $\mathrm{N} \leq 1$, $\mathrm{LIWORK} \geq 1$;
if $\mathrm{JOBZ} = \text{'N'}$ and $\mathrm{N} > 1$, $\mathrm{LIWORK} \geq 1$;
if $\mathrm{JOBZ} = \text{'V'}$ and $\mathrm{N} > 1$, $\mathrm{LIWORK} \geq 3 + 5 \times \mathrm{N}$.

16: INFO – INTEGER *Output*

On exit: $\mathrm{INFO} = 0$ unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the routine:

$\mathrm{INFO} < 0$

If $\mathrm{INFO} = -i$, argument i had an illegal value. An explanatory message is output, and execution of the program is terminated.

$\mathrm{INFO} > 0$

F07FRF (ZPOTRF) or F08FQF (ZHEEVD) returned an error code:

$\leq \mathrm{N}$ if $\mathrm{INFO} = i$, F08FQF (ZHEEVD) failed to converge; i off-diagonal elements of an intermediate tridiagonal form did not converge to zero;

$> \mathrm{N}$ if $\mathrm{INFO} = \mathrm{N} + i$, for $1 \leq i \leq \mathrm{N}$, 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 F08SCF (DSYGVD).

9 Example

This example finds all the eigenvalues and eigenvectors of the generalized Hermitian eigenproblem $ABz = \lambda z$, 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 an estimate of the condition number of B, and approximate error bounds for the computed eigenvalues and eigenvectors.

The example program for F08SNF (ZHEGV) illustrates solving a generalized Hermitian eigenproblem of the form $Az = \lambda Bz$.

9.1 Program Text

```
*     F08SQF 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, LIWORK, LRWORK, LWORK
      PARAMETER        (LDA=NMAX,LDB=NMAX,LIWORK=3+5*NMAX,
     +                 LRWORK=1+(5+2*NMAX)*NMAX,LWORK=(NB+2+NMAX)*NMAX)
      DOUBLE PRECISION ONE
      PARAMETER        (ONE=1.0D+0)
*     .. Local Scalars ..
      DOUBLE PRECISION ANORM, BNORM, EPS, RCOND, RCONDB, T1, T2, T3
      INTEGER          I, IFAIL, INFO, J, LIWOPT, LRWOPT, LWOPT, N
*     .. Local Arrays ..
      COMPLEX *16      A(LDA,NMAX), B(LDB,NMAX), WORK(LWORK)
      DOUBLE PRECISION EERBND(NMAX), RCONDZ(NMAX), RWORK(LRWORK),
     +                 W(NMAX), ZERBND(NMAX)
      INTEGER          IWORK(LIWORK)
*     .. External Functions ..
      DOUBLE PRECISION F06UCF, X02AJF
      EXTERNAL         F06UCF, X02AJF
*     .. External Subroutines ..
      EXTERNAL         DDISNA, X04DAF, ZHEGVD, ZTRCON
*     .. Intrinsic Functions ..
      INTRINSIC        ABS, INT
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08SQF 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*B*x = lambda*x (ITYPE = 2)
*
         CALL ZHEGVD(2,'Vectors','Upper',N,A,LDA,B,LDB,W,WORK,LWORK,
     +               RWORK,LRWORK,IWORK,LIWORK,INFO)
         LWOPT = INT(WORK(1))
         LIWOPT = IWORK(1)
         LRWOPT = INT(RWORK(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*B - lambda*I)
*
               CALL DDISNA('Eigenvectors',N,N,W,RCONDZ,INFO)
*
*              Compute the error estimates for the eigenvalues and
*              eigenvectors
*
               T1 = ONE/RCOND
               T2 = EPS*T1
               T3 = ANORM*BNORM
               DO 20 I = 1, N
                  EERBND(I) = EPS*(T3+ABS(W(I))/RCONDB)
                  ZERBND(I) = T2*(T3/RCONDZ(I)+T1)
   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 ZHEGVD. INFO =', INFO
         END IF
*
*        Print workspace information
*
         IF (LWORK.LT.LWOPT) THEN
            WRITE (NOUT,*)
            WRITE (NOUT,99995) 'Optimum workspace required = ', LWOPT,
     +        'Workspace provided         = ', LWORK
         END IF
         IF (LRWORK.LT.LRWOPT) THEN
            WRITE (NOUT,*)
            WRITE (NOUT,99995) 'Real workspace required = ', LRWOPT,
     +        'Real workspace provided = ', LRWORK
         END IF
         IF (LIWORK.LT.LIWOPT) THEN
            WRITE (NOUT,*)
            WRITE (NOUT,99995) 'Integer workspace required = ', LIWOPT,
     +        'Integer workspace provided = ', LIWORK
         END IF
      ELSE
         WRITE (NOUT,*) 'NMAX too small'
      END IF
*
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

```
F08SQF 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

```
 F08SQF Example Program Results

 Eigenvalues
      -61.7321    -6.6195     0.0725    43.1883
 Eigenvectors
            1          2          3          4
 1     -0.2729    -0.1219     0.9130     0.1738
        0.2791     0.1054    -2.1011     0.1110

 2      0.1187    -0.3886    -0.8539    -0.3785
       -0.1377    -0.0797     0.1787    -0.0873

 3     -0.0548     0.3047    -0.9232    -0.1111
        0.0077    -0.4414     0.9837     0.1894
```

```
 4      0.3177    -0.2285     1.4319    -0.3000
        0.0000     0.0000     0.0000     0.0000

 Estimate of reciprocal condition number for B
        2.5E-03

 Error estimates for the eigenvalues
        2.7E-12    3.1E-13    2.6E-14    1.9E-12

 Error estimates for the eigenvectors
        5.2E-14    1.1E-13    1.1E-13    5.4E-14
```