

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

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

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,      &
                  RWORK, LRWORK, IWORK, LIWORK, INFO)
INTEGER              ITYPE, N, LDA, LDB, LWORK, LRWORK,                &
                    IWORK(max(1,LIWORK)), LIWORK, INFO
REAL (KIND=nag_wp)  W(N), RWORK(max(1,LRWORK))
COMPLEX (KIND=nag_wp) 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 $B = U^H U$, 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

$$Z^H A Z = \Lambda \quad \text{and} \quad Z^H B Z = I,$$

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

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

and for $BAz = \lambda z$ we have

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

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 <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.
 ITYPE = 1
 $Az = \lambda Bz.$
 ITYPE = 2
 $ABz = \lambda z.$
 ITYPE = 3
 $BAz = \lambda z.$
Constraint: ITYPE = 1, 2 or 3.
- 2: JOBZ – CHARACTER(1) *Input*
On entry: indicates whether eigenvectors are computed.
 JOBZ = 'N'
 Only eigenvalues are computed.
 JOBZ = 'V'
 Eigenvalues and eigenvectors are computed.
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: $N \geq 0$.
- 5: A(LDA,*) – COMPLEX (KIND=nag_wp) array *Input/Output*
Note: the second dimension of the array A must be at least $\max(1, N)$.
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', 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 overwritten.
- 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: $LDA \geq \max(1, N)$.

- 7: B(LDB,*) – COMPLEX (KIND=nag_wp) array Input/Output
Note: the second dimension of the array B must be at least $\max(1, N)$.
On entry: the n by n Hermitian matrix B .
 If UPLO = 'U', the upper triangular part of B must be stored and the elements of the array below the diagonal are not referenced.
 If UPLO = 'L', the lower triangular part of B must be stored and the elements of the array above the diagonal are not referenced.
On exit: 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 F08SQF (ZHEGVD) is called.
Constraint: $LDB \geq \max(1, N)$.
- 9: W(N) – REAL (KIND=nag_wp) array Output
On exit: the eigenvalues in ascending order.
- 10: WORK(max(1, LWORK)) – COMPLEX (KIND=nag_wp) 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 $nb \times N$, where nb is the optimal **block size**.
Constraints:
 if $N \leq 1$, LWORK ≥ 1 ;
 if JOBZ = 'N' and $N > 1$, LWORK $\geq N + 1$;
 if JOBZ = 'V' and $N > 1$, LWORK $\geq 2 \times N + N^2$.
- 12: RWORK(max(1, LRWORK)) – REAL (KIND=nag_wp) 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 $N \leq 1$, LRWORK ≥ 1 ;
 if JOBZ = 'N' and $N > 1$, LRWORK $\geq N$;
 if JOBZ = 'V' and $N > 1$, LRWORK $\geq 1 + 5 \times N + 2 \times N^2$.

- 14: IWORK(max(1, LIWORK)) – INTEGER array *Workspace*
On exit: if INFO = 0, 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 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 $N \leq 1$, $LIWORK \geq 1$;
 if JOBZ = 'N' and $N > 1$, $LIWORK \geq 1$;
 if JOBZ = 'V' and $N > 1$, $LIWORK \geq 3 + 5 \times N$.
- 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 , argument i had an illegal value. An explanatory message is output, and execution of the program is terminated.

INFO = 1 to N

If INFO = i , F08FQF (ZHEEVD) failed to converge; i off-diagonal elements of an intermediate tridiagonal form did not converge to zero.

INFO > N

F07FRF (ZPOTRF) returned an error code; i.e., 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 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

```

Program f08sqfe

!      F08SQF Example Program Text

!      Mark 24 Release. NAG Copyright 2012.

!      .. Use Statements ..
Use nag_library, Only: ddisna, nag_wp, x02ajf, x04daf, zhegvd,      &
                        zlanhe => f06ucf, ztrcon

!      .. Implicit None Statement ..
Implicit None

!      .. Parameters ..
Real (Kind=nag_wp), Parameter      :: one = 1.0E+0_nag_wp
Integer, Parameter                  :: nb = 64, nin = 5, nout = 6

!      .. Local Scalars ..
Real (Kind=nag_wp)                  :: anorm, bnorm, eps, rcond, rcondb,      &
                                     t1, t2, t3
Integer                              :: i, ifail, info, lda, ldb, liwork,      &
                                     lrwork, lwork, n

!      .. Local Arrays ..
Complex (Kind=nag_wp), Allocatable :: a(:,,:), b(:,,:), work(:)
Complex (Kind=nag_wp)               :: cdum(1)
Real (Kind=nag_wp), Allocatable     :: eerbnd(:), rcondz(:), rwork(:),      &
                                     w(:), zerbnd(:)
Real (Kind=nag_wp)                  :: rdum(1)
Integer                              :: idum(1)
Integer, Allocatable                 :: iwork(:)

!      .. Intrinsic Procedures ..
Intrinsic                            :: abs, max, nint, real

!      .. Executable Statements ..
Write (nout,*) 'F08SQF Example Program Results'
Write (nout,*)

!      Skip heading in data file
Read (nin,*)
Read (nin,*) n
lda = n
ldb = n
Allocate (a(lda,n),b(ldb,n),eerbnd(n),rcondz(n),w(n),zerbnd(n))

!      Use routine workspace query to get optimal workspace.
lwork = -1
liwork = -1
lrwork = -1

!      The NAG name equivalent of zhegvd is f08sqf
Call zhegvd(2,'Vectors','Upper',n,a,lda,b,ldb,w,cdum,lwork,rdum,lrwork, &
            idum,liwork,info)

```

```

!      Make sure that there is enough workspace for blocksize nb.
      lwork = max((nb+2+n)*n,nint(real(cdum(1))))
      lrwork = max(1+(5+2*n)*n,nint(rdum(1)))
      liwork = max(3+5*n,idum(1))
      Allocate (work(lwork),rwork(lrwork),iwork(liwork))

!      Read the upper triangular parts of the matrices A and B

      Read (nin,*)(a(i,i:n),i=1,n)
      Read (nin,*)(b(i,i:n),i=1,n)

!      Compute the one-norms of the symmetric matrices A and B

!      f06ucf is the NAG name equivalent of the LAPACK auxiliary zlanhe
      anorm = zlanhe('One norm','Upper',n,a,lda,rwork)
      bnorm = zlanhe('One norm','Upper',n,b,ldb,rwork)

!      Solve the generalized Hermitian eigenvalue problem
!      A*B*x = lambda*x (itype = 2)
!      The NAG name equivalent of zhegvd is f08sqf
      Call zhegvd(2,'Vectors','Upper',n,a,lda,b,ldb,w,work,lwork,rwork,lrwork, &
        iwork,liwork,info)

      If (info==0) Then

!      Print solution

      Write (nout,*) 'Eigenvalues'
      Write (nout,99999) w(1:n)
      Flush (nout)

!      Normalize the eigenvectors
      Do i = 1, n
        a(1:n,i) = a(1:n,i)/a(1,i)
      End Do

!      ifail: behaviour on error exit
!      =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
      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
      Flush (nout)

!      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>=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 i = 1, n
        eerbnd(i) = eps*(t3+abs(w(i))/rcondb)
        zerbnd(i) = t2*(t3/rcondz(i)+t1)
    End Do

!      Print the approximate error bounds for the eigenvalues
!      and vectors

    Write (nout,*)
    Write (nout,*) 'Error estimates for the eigenvalues'
    Write (nout,99998) eerbnd(1:n)
    Write (nout,*)
    Write (nout,*) 'Error estimates for the eigenvectors'
    Write (nout,99998) zerbnd(1:n)
    Else
        Write (nout,*)
        Write (nout,*) 'B is very ill-conditioned, error ', &
            'estimates have not been computed'
    End If
    Else If (info>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

99999 Format (3X,(6F11.4))
99998 Format (4X,1P,6E11.1)
99997 Format (1X,A,I4,A)
99996 Format (1X,A,I4)
    End Program f08sqfe

```

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    1.0000    1.0000    1.0000    1.0000
   -0.0000    0.0000   -0.0000   -0.0000

2    -0.4648    1.5001   -0.2201   -1.7750
   0.0291     1.9518   -0.3108    0.6305

3     0.1123   -3.2220   -0.5544    0.0399
   0.0866     0.8343   -0.1985    1.0638

4    -0.5690    1.0723    0.2491   -1.2264
   -0.5820     0.9277    0.5732    0.7828

```

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
