NAG Library Routine Document

F08UQF (ZHBGVD)

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

F08UQF (ZHBGVD) computes all the eigenvalues and, optionally, the eigenvectors of a complex generalized Hermitian-definite banded eigenproblem, of the form

$$Az = \lambda Bz,$$

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

2 Specification

```
SUBROUTINE F08UQF (JOBZ, UPLO, N, KA, KB, AB, LDAB, BB, LDBB, W, Z, LDZ,        &
                   WORK, LWORK, RWORK, LRWORK, IWORK, LIWORK, INFO)

INTEGER                 N, KA, KB, LDAB, LDBB, LDZ, LWORK, LRWORK,              &
                        IWORK(max(1,LIWORK)), LIWORK, INFO
REAL (KIND=nag_wp)      W(N), RWORK(max(1,LRWORK))
COMPLEX (KIND=nag_wp) AB(LDAB,*), BB(LDBB,*), Z(LDZ,*), WORK(max(1,LWORK))
CHARACTER(1)            JOBZ, UPLO
```

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

3 Description

The generalized Hermitian-definite band problem

$$Az = \lambda Bz$$

is first reduced to a standard band Hermitian problem

$$Cx = \lambda x,$$

where C is a Hermitian band matrix, using Wilkinson's modification to Crawford's algorithm (see Crawford (1973) and Wilkinson (1977)). The Hermitian eigenvalue problem is then solved for the eigenvalues and the eigenvectors, if required, which are then backtransformed to the eigenvectors of the original problem.

The eigenvectors are normalized so that the matrix of eigenvectors, Z, satisfies

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

where Λ is the diagonal matrix whose diagonal elements are the eigenvalues.

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

Crawford C R (1973) Reduction of a band-symmetric generalized eigenvalue problem *Comm. ACM* **16** 41–44

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

Wilkinson J H (1977) Some recent advances in numerical linear algebra *The State of the Art in Numerical Analysis* (ed D A H Jacobs) Academic Press

5 Parameters

1: 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'.

2: 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'.

3: N – INTEGER *Input*

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

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

4: KA – INTEGER *Input*

On entry: if UPLO = 'U', the number of superdiagonals, k_a, of the matrix A.

If UPLO = 'L', the number of subdiagonals, k_a, of the matrix A.

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

5: KB – INTEGER *Input*

On entry: if UPLO = 'U', the number of superdiagonals, k_b, of the matrix B.

If UPLO = 'L', the number of subdiagonals, k_b, of the matrix B.

Constraint: $\mathrm{KA} \geq \mathrm{KB} \geq 0$.

6: AB(LDAB,$*$) – COMPLEX (KIND=nag_wp) array *Input/Output*

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

On entry: the upper or lower triangle of the n by n Hermitian band matrix A.

The matrix is stored in rows 1 to $k_a + 1$, more precisely,

if UPLO = 'U', the elements of the upper triangle of A within the band must be stored with element A_{ij} in $\mathrm{AB}(k_a + 1 + i - j, j)$ for $\max(1, j - k_a) \leq i \leq j$;

if UPLO = 'L', the elements of the lower triangle of A within the band must be stored with element A_{ij} in $\mathrm{AB}(1 + i - j, j)$ for $j \leq i \leq \min(n, j + k_a)$.

On exit: the contents of AB are overwritten.

7: LDAB – INTEGER *Input*

On entry: the first dimension of the array AB as declared in the (sub)program from which F08UQF (ZHBGVD) is called.

Constraint: LDAB $\geq$ KA + 1.

8: BB(LDBB,$*$) – COMPLEX (KIND=nag_wp) array *Input/Output*

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

On entry: the upper or lower triangle of the n by n Hermitian band matrix B.

The matrix is stored in rows 1 to $k_b + 1$, more precisely,

if UPLO = 'U', the elements of the upper triangle of B within the band must be stored with element B_{ij} in $\mathrm{BB}(k_b + 1 + i - j, j)$ for $\max(1, j - k_b) \leq i \leq j$;

if UPLO = 'L', the elements of the lower triangle of B within the band must be stored with element B_{ij} in $\mathrm{BB}(1 + i - j, j)$ for $j \leq i \leq \min(n, j + k_b)$.

On exit: the factor S from the split Cholesky factorization $B = S^{\mathrm{H}}S$, as returned by F08UTF (ZPBSTF).

9: LDBB – INTEGER *Input*

On entry: the first dimension of the array BB as declared in the (sub)program from which F08UQF (ZHBGVD) is called.

Constraint: LDBB $\geq$ KB + 1.

10: W(N) – REAL (KIND=nag_wp) array *Output*

On exit: the eigenvalues in ascending order.

11: Z(LDZ,$*$) – COMPLEX (KIND=nag_wp) array *Output*

Note: the second dimension of the array Z must be at least $\max(1, \mathrm{N})$ if JOBZ = 'V', and at least 1 otherwise.

On exit: if JOBZ = 'V', Z contains the matrix Z of eigenvectors, with the ith column of Z holding the eigenvector associated with $\mathrm{W}(i)$. The eigenvectors are normalized so that $Z^{\mathrm{H}}BZ = I$.

If JOBZ = 'N', Z is not referenced.

12: LDZ – INTEGER *Input*

On entry: the first dimension of the array Z as declared in the (sub)program from which F08UQF (ZHBGVD) is called.

Constraints:

if JOBZ = 'V', LDZ $\geq \max(1, \mathrm{N})$;
otherwise LDZ $\geq$ 1.

13: WORK($\max(1, \mathrm{LWORK})$) – 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.

14: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08UQF (ZHBGVD) 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.

Constraints:

if $\mathrm{N} \leq 1$, $\mathrm{LWORK} \geq 1$;
if JOBZ = 'N' and $\mathrm{N} > 1$, $\mathrm{LWORK} \geq \max(1, \mathrm{N})$;
if JOBZ = 'V' and $\mathrm{N} > 1$, $\mathrm{LWORK} \geq \max\left(1, \mathrm{N}^2\right)$.

15: RWORK$(\max(1, \mathrm{LRWORK}))$ – REAL (KIND=nag_wp) array *Workspace*

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

16: LRWORK – INTEGER *Input*

On entry: the first dimension of the array RWORK as declared in the (sub)program from which F08UQF (ZHBGVD) 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 JOBZ = 'N' and $\mathrm{N} > 1$, $\mathrm{LRWORK} \geq \max(1, \mathrm{N})$;
if JOBZ = 'V' and $\mathrm{N} > 1$, $\mathrm{LRWORK} \geq 1 + 5 \times \mathrm{N} + 2 \times \mathrm{N}^2$.

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

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

18: LIWORK – INTEGER *Input*

On entry: the dimension of the array IWORK as declared in the (sub)program from which F08UQF (ZHBGVD) 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 JOBZ = 'N' or $\mathrm{N} \leq 1$, $\mathrm{LIWORK} \geq 1$;
if JOBZ = 'V' and $\mathrm{N} > 1$, $\mathrm{LIWORK} \geq 3 + 5 \times \mathrm{N}$.

19: 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 > 0

If INFO $= i$ and $i \leq \mathrm{N}$, the algorithm failed to converge; i off-diagonal elements of an intermediate tridiagonal form did not converge to zero.

If INFO $= i$ and $i > \mathrm{N}$, if INFO $= \mathrm{N} + i$, for $1 \le i \le \mathrm{N}$, then F08UTF (ZPBSTF) returned INFO $= i$: 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.

8 Further Comments

The total number of floating point operations is proportional to n^3 if JOBZ = 'V' and, assuming that $n \gg k_a$, is approximately proportional to $n^2 k_a$ otherwise.

The real analogue of this routine is F08UCF (DSBGVD).

9 Example

This example finds all the eigenvalues of the generalized band Hermitian eigenproblem $Az = \lambda Bz$, where

$$A = \begin{pmatrix} -1.13 & 1.94 - 2.10i & -1.40 + 0.25i & 0 \\ 1.94 + 2.10i & -1.91 & -0.82 - 0.89i & -0.67 + 0.34i \\ -1.40 - 0.25i & -0.82 + 0.89i & -1.87 & -1.10 - 0.16i \\ 0 & -0.67 - 0.34i & -1.10 + 0.16i & 0.50 \end{pmatrix}$$

and

$$B = \begin{pmatrix} 9.89 & 1.08 - 1.73i & 0 & 0 \\ 1.08 + 1.73i & 1.69 & -0.04 + 0.29i & 0 \\ 0 & -0.04 - 0.29i & 2.65 & -0.33 + 2.24i \\ 0 & 0 & -0.33 - 2.24i & 2.17 \end{pmatrix}.$$

9.1 Program Text

```
    Program f08uqfe

!     F08UQF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, zhbgvd
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
      Character (1), Parameter         :: uplo = 'U'
!     .. Local Scalars ..
      Integer                          :: i, info, j, ka, kb, ldab, ldbb,      &
                                          liwork, lrwork, lwork, n
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: ab(:,:), bb(:,:), work(:)
      Complex (Kind=nag_wp)            :: dummy(1,1)
      Real (Kind=nag_wp), Allocatable  :: rwork(:), w(:)
      Integer, Allocatable             :: iwork(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: max, min
!     .. Executable Statements ..
      Write (nout,*) 'F08UQF Example Program Results'
      Write (nout,*)
!     Skip heading in data file
      Read (nin,*)
```

```
      Read (nin,*) n, ka, kb
      ldab = ka + 1
      ldbb = kb + 1
      lrwork = n
      lwork = n
      liwork = 1
      Allocate (ab(ldab,n),bb(ldbb,n),work(lwork),rwork(lrwork),w(n), &
        iwork(liwork))

!     Read the upper or lower triangular parts of the matrices A and
!     B from data file

      If (uplo=='U') Then
        Read (nin,*)((ab(ka+1+i-j,j),j=i,min(n,i+ka)),i=1,n)
        Read (nin,*)((bb(kb+1+i-j,j),j=i,min(n,i+kb)),i=1,n)
      Else If (uplo=='L') Then
        Read (nin,*)((ab(1+i-j,j),j=max(1,i-ka),i),i=1,n)
        Read (nin,*)((bb(1+i-j,j),j=max(1,i-kb),i),i=1,n)
      End If

!     Solve the generalized Hermitian band eigenvalue problem
!     A*x = lambda*B*x

!     The NAG name equivalent of zhbgvd is f08uqf
      Call zhbgvd('No vectors',uplo,n,ka,kb,ab,ldab,bb,ldbb,w,dummy,1,work, &
        lwork,rwork,lrwork,iwork,liwork,info)

      If (info==0) Then

!       Print solution

        Write (nout,*) 'Eigenvalues'
        Write (nout,99999) w(1:n)
      Else If (info>n .And. info<=2*n) Then
        i = info - n
        Write (nout,99998) 'The leading minor of order ', i, &
          ' of B is not positive definite'
      Else
        Write (nout,99997) 'Failure in ZHBGVD. INFO =', info
      End If

99999 Format (3X,(6F11.4))
99998 Format (1X,A,I4,A)
99997 Format (1X,A,I4)
    End Program f08uqfe
```

9.2 Program Data

```
F08UQF Example Program Data

   4            2            1                          :Values of N, KA and KB

 (-1.13, 0.00) ( 1.94,-2.10) (-1.40, 0.25)
               (-1.91, 0.00) (-0.82,-0.89) (-0.67, 0.34)
                             (-1.87, 0.00) (-1.10,-0.16)
                                           ( 0.50, 0.00) :End of matrix A

 ( 9.89, 0.00) ( 1.08,-1.73)
               ( 1.69, 0.00) (-0.04, 0.29)
                             ( 2.65, 0.00) (-0.33, 2.24)
                                           ( 2.17, 0.00) :End of matrix B
```

9.3 Program Results

```
 F08UQF Example Program Results

 Eigenvalues
       -6.6089    -2.0416     0.1603     1.7712
```