NAG Library Routine Document

F08TQF (ZHPGVD)

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

F08TQF (ZHPGVD) 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, stored in packed format, and B is also positive definite. If eigenvectors are desired, it uses a divide-and-conquer algorithm.

2 Specification

```
SUBROUTINE F08TQF (ITYPE, JOBZ, UPLO, N, AP, BP, W, Z, LDZ, WORK, LWORK,    &
                   RWORK, LRWORK, IWORK, LIWORK, INFO)
INTEGER                ITYPE, N, LDZ, LWORK, LRWORK,                        &
                       IWORK(max(1,LIWORK)), LIWORK, INFO
REAL (KIND=nag_wp)     W(N), RWORK(max(1,LRWORK))
COMPLEX (KIND=nag_wp) AP(*), BP(*), Z(LDZ,*), WORK(max(1,LWORK))
CHARACTER(1)           JOBZ, UPLO
```

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

3 Description

F08TQF (ZHPGVD) 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 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 Arguments

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

5: AP($*$) – COMPLEX (KIND=nag_wp) array *Input/Output*

Note: the dimension of the array AP must be at least $\max(1, \mathrm{N} \times (\mathrm{N}+1)/2)$.

On entry: the upper or lower triangle of the n by n Hermitian matrix A, packed by columns.

More precisely,

if UPLO = 'U', the upper triangle of A must be stored with element A_{ij} in $\mathrm{AP}(i + j(j-1)/2)$ for $i \leq j$;

if UPLO = 'L', the lower triangle of A must be stored with element A_{ij} in $\mathrm{AP}(i + (2n-j)(j-1)/2)$ for $i \geq j$.

On exit: the contents of AP are destroyed.

6: BP($*$) – COMPLEX (KIND=nag_wp) array *Input/Output*

Note: the dimension of the array BP must be at least $\max(1, \mathrm{N} \times (\mathrm{N}+1)/2)$.

On entry: the upper or lower triangle of the n by n Hermitian matrix B, packed by columns.

More precisely,

if UPLO = 'U', the upper triangle of B must be stored with element B_{ij} in $\text{BP}(i + j(j-1)/2)$ for $i \le j$;

if UPLO = 'L', the lower triangle of B must be stored with element B_{ij} in $\text{BP}(i + (2n-j)(j-1)/2)$ for $i \ge j$.

On exit: the triangular factor U or L from the Cholesky factorization $B = U^{\text{H}}U$ or $B = LL^{\text{H}}$, in the same storage format as B.

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

On exit: the eigenvalues in ascending order.

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

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

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

if ITYPE = 1 or 2, $Z^{\text{H}}BZ = I$;

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

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

9: LDZ – INTEGER *Input*

On entry: the first dimension of the array Z as declared in the (sub)program from which F08TQF (ZHPGVD) is called.

Constraints:

if JOBZ = 'V', LDZ $\ge$ max(1, N);
otherwise LDZ $\ge$ 1.

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 F08TQF (ZHPGVD) 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 N $\le$ 1, LWORK $\ge$ 1;
if JOBZ = 'N' and N > 1, LWORK $\ge$ N;
if JOBZ = 'V' and N > 1, LWORK $\ge 2 \times$ N.

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 F08TQF (ZHPGVD) 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 F08TQF (ZHPGVD) 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 $N \leq 1$, LIWORK ≥ 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

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

F07GRF (ZPPTRF) or F08GQF (ZHPEVD) returned an error code:

$\leq$ N if INFO $= i$, F08GQF (ZHPEVD) 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 Parallelism and Performance

F08TQF (ZHPGVD) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F08TQF (ZHPGVD) makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

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

The real analogue of this routine is F08TCF (DSPGVD).

10 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 F08TNF (ZHPGV) illustrates solving a generalized Hermitian eigenproblem of the form $Az = \lambda Bz$.

10.1 Program Text

```
    Program f08tqfe

!     F08TQF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: f06udf, nag_wp, x02ajf, zhpgvd, ztpcon
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
      Character (1), Parameter         :: uplo = 'U'
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: anorm, bnorm, eps, rcond, rcondb, t1
      Integer                          :: aplen, i, info, j, liwork, lrwork,   &
                                          lwork, n
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: ap(:), bp(:), work(:)
      Complex (Kind=nag_wp)            :: dummy(1,1)
      Real (Kind=nag_wp), Allocatable  :: eerbnd(:), rwork(:), w(:)
      Real (Kind=nag_wp)               :: rdum(1)
```

```
      Integer                          :: idum(1)
      Integer, Allocatable             :: iwork(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: abs, max, nint, real
!     .. Executable Statements ..
      Write (nout,*) 'F08TQF Example Program Results'
      Write (nout,*)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      aplen = (n*(n+1))/2
      Allocate (ap(aplen),bp(aplen),eerbnd(n),w(n))

!     Use routine workspace query to get optimal workspace.
      lwork = -1
      liwork = -1
      lrwork = -1
!     The NAG name equivalent of zhpgvd is f08tqf
      Call zhpgvd(2,'No vectors',uplo,n,ap,bp,w,dummy,1,dummy,lwork,rdum,      &
        lrwork,idum,liwork,info)

!     Make sure that there is at least the minimum workspace
      lwork = max(2*n,nint(real(dummy(1,1))))
      lrwork = max(n,nint(rdum(1)))
      liwork = max(1,idum(1))
      Allocate (work(lwork),rwork(lrwork),iwork(liwork))

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

      If (uplo=='U') Then
        Read (nin,*)((ap(i+(j*(j-1))/2),j=i,n),i=1,n)
        Read (nin,*)((bp(i+(j*(j-1))/2),j=i,n),i=1,n)
      Else If (uplo=='L') Then
        Read (nin,*)((ap(i+((2*n-j)*(j-1))/2),j=1,i),i=1,n)
        Read (nin,*)((bp(i+((2*n-j)*(j-1))/2),j=1,i),i=1,n)
      End If

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

      anorm = f06udf('One norm',uplo,n,ap,rwork)
      bnorm = f06udf('One norm',uplo,n,bp,rwork)

!     Solve the generalized symmetric eigenvalue problem
!     A*B*x = lambda*x (itype = 2)

!     The NAG name equivalent of zhpgvd is f08tqf
      Call zhpgvd(2,'No vectors',uplo,n,ap,bp,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)

!       Call ZTPCON (F07UUF) to estimate the reciprocal condition
!       number of the Cholesky factor of B.  Note that:
!       cond(B) = 1/rcond**2.  ZTPCON requires WORK and RWORK to be
!       of length at least 2*n and n respectively

        Call ztpcon('One norm',uplo,'Non-unit',n,bp,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

        eps = x02ajf()
        If (rcond>=eps) Then
          t1 = anorm*bnorm
          Do i = 1, n
            eerbnd(i) = t1 + abs(w(i))/rcondb
          End Do

!         Print the approximate error bounds for the eigenvalues

          Write (nout,*)
          Write (nout,*) 'Error estimates (relative to machine precision)'
          Write (nout,*) 'for the eigenvalues:'
          Write (nout,99998) eerbnd(1:n)
        Else
          Write (nout,*)
          Write (nout,*) 'B is very ill-conditioned, error ',                  &
            'estimates have not been computed'
        End If
      Else If (info>n .And. info<=2*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 ZHPGVD. 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 f08tqfe
```

10.2 Program Data

```
F08TQF 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
```

10.3 Program Results

```
 F08TQF Example Program Results

 Eigenvalues
      -61.7321    -6.6195     0.0725    43.1883

 Estimate of reciprocal condition number for B
        2.5E-03

 Error estimates (relative to machine precision)
 for the eigenvalues:
        2.4E+04    2.8E+03    2.3E+02    1.7E+04
```