NAG Fortran 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,
1                   RWORK, LRWORK, IWORK, LIWORK, INFO)
 INTEGER            ITYPE, N, LDZ, LWORK, LRWORK, IWORK(*), LIWORK, INFO
 double precision   W(*), RWORK(*)
 complex*16         AP(*), BP(*), Z(LDZ,*), WORK(*)
 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 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: $\mathrm{N} \geq 0$.

5: AP($*$) – ***complex*16*** 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 Hermitian matrix A, packed columnwise in a linear array. The jth column of A is stored in the array AP as follows:

if UPLO = 'U', $\mathrm{AP}(i + (j-1) \times j/2) = a_{ij}$ for $1 \leq i \leq j$;
if UPLO = 'L', $\mathrm{AP}(i + (j-1) \times (2 \times n - j)/2) = a_{ij}$ for $j \leq i \leq n$.

On exit: the contents of AP are destroyed.

6: BP($*$) – ***complex*16*** 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 Hermitian matrix B, packed columnwise in a linear array. The jth column of B is stored in the array BP as follows:

if UPLO = 'U', $\mathrm{BP}(i + (j-1) \times j/2) = b_{ij}$ for $1 \leq i \leq j$;
if UPLO = 'L', $\mathrm{BP}(i + (j-1) \times (2 \times n - j)/2) = b_{ij}$ for $j \leq i \leq n$.

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

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

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

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

8: Z(LDZ,$*$) – ***complex*16*** array *Output*

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

On exit: if JOBZ = 'V', then if $\text{INFO} = 0$, Z 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', 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', $\text{LDZ} \geq \max(1, N)$;
$\text{LDZ} \geq 1$ otherwise.

10: WORK($*$) – ***complex*16*** array *Workspace*

Note: the dimension of the array WORK must be at least $\max(1, \text{LWORK})$.

On exit: if $\text{INFO} = 0$, $\text{WORK}(1)$ returns the optimal LWORK.

11: LWORK – INTEGER *Input*

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

If $\text{LWORK} = -1$, 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 $N \leq 1$, $\text{LWORK} \geq 1$;
if JOBZ = 'N' and $N > 1$, $\text{LWORK} \geq N$;
if JOBZ = 'V' and $N > 1$, $\text{LWORK} \geq 2 \times N$.

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

Note: the dimension of the array RWORK must be at least $\max(1, \text{LRWORK})$.

On exit: if $\text{INFO} = 0$, $\text{RWORK}(1)$ 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 $\text{LRWORK} = -1$, 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 $N \leq 1$, $\text{LRWORK} \geq 1$;
if JOBZ = 'N' and $N > 1$, $\text{LRWORK} \geq N$;
if JOBZ = 'V' and $N > 1$, $\text{LRWORK} \geq 1 + 5 \times N + 2 \times N^2$.

14: IWORK($*$) – INTEGER array *Workspace*

Note: the dimension of the array IWORK must be at least $\max(1, \text{LIWORK})$.

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

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 $= -i$, the ith argument had an illegal value.

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 $= \text{N} + i$, for $1 \leq i \leq \text{N}$, 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 F08TCF (DSPGVD).

9 Example

To find 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 $Az = \lambda Bz$.

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.

```
*     F08TQF Example Program Text
*     Mark 21.  NAG Copyright 2004.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX
      PARAMETER        (NMAX=10)
      INTEGER          LIWORK, LRWORK, LWORK
      PARAMETER        (LIWORK=1,LRWORK=NMAX,LWORK=2*NMAX)
      CHARACTER        UPLO
      PARAMETER        (UPLO='U')
*     .. Local Scalars ..
      DOUBLE PRECISION ANORM, BNORM, EPS, RCOND, RCONDB, T1
      INTEGER          I, INFO, J, LIWOPT, LRWOPT, LWOPT, N
*     .. Local Arrays ..
      COMPLEX *16      AP((NMAX*(NMAX+1))/2), BP((NMAX*(NMAX+1))/2),
     +                 DUMMY(1,1), WORK(LWORK)
      DOUBLE PRECISION EERBND(NMAX), RWORK(LRWORK), W(NMAX)
      INTEGER          IWORK(LIWORK)
*     .. External Functions ..
      DOUBLE PRECISION F06UDF, X02AJF
      EXTERNAL         F06UDF, X02AJF
*     .. External Subroutines ..
      EXTERNAL         ZHPGVD, ZTPCON
*     .. Intrinsic Functions ..
      INTRINSIC        ABS
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08TQF Example Program Results'
      WRITE (NOUT,*)
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      IF (N.LE.NMAX) THEN
*
*        Read the upper or lower triangular parts of the matrices A and
*        B from data file
*
         IF (UPLO.EQ.'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.EQ.'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)
*
         CALL ZHPGVD(2,'No vectors',UPLO,N,AP,BP,W,DUMMY,1,WORK,LWORK,
     +               RWORK,LRWORK,IWORK,LIWORK,INFO)
         LWOPT = WORK(1)
         LRWOPT = RWORK(1)
         LIWOPT = IWORK(1)
*
         IF (INFO.EQ.0) THEN
*
*           Print solution
*
            WRITE (NOUT,*) 'Eigenvalues'
            WRITE (NOUT,99999) (W(J),J=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.GE.EPS) THEN
               T1 = ANORM*BNORM
               DO 20 I = 1, N
                  EERBND(I) = EPS*(T1+ABS(W(I))/RCONDB)
   20          CONTINUE
*
*              Print the approximate error bounds for the eigenvalues
*
               WRITE (NOUT,*)
               WRITE (NOUT,*) 'Error estimates for the eigenvalues'
               WRITE (NOUT,99998) (EERBND(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 .AND. INFO.LE.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
*
*        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
      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

```
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
```

9.3 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 for the eigenvalues
        2.7E-12    3.1E-13    2.6E-14    1.9E-12
```