NAG Library Routine Document

F08BTF (ZGEQP3)

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

F08BTF (ZGEQP3) computes the QR factorization, with column pivoting, of a complex m by n matrix.

2 Specification

```
SUBROUTINE F08BTF(M, N, A, LDA, JPVT, TAU, WORK, LWORK, RWORK, INFO)
INTEGER            M, N, LDA, JPVT(*), LWORK, INFO
double precision   RWORK(*)
complex*16         A(LDA,*), TAU(*), WORK(max(1,LWORK))
```

The routine may be called by its LAPACK name ***zgeqp3***.

3 Description

F08BTF (ZGEQP3) forms the QR factorization, with column pivoting, of an arbitrary rectangular complex m by n matrix.

If $m \geq n$, the factorization is given by:

$$AP = Q\begin{pmatrix} R \\ 0 \end{pmatrix},$$

where R is an n by n upper triangular matrix (with real diagonal elements), Q is an m by m unitary matrix and P is an n by n permutation matrix. It is sometimes more convenient to write the factorization as

$$AP = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix}\begin{pmatrix} R \\ 0 \end{pmatrix},$$

which reduces to

$$AP = Q_1R,$$

where Q_1 consists of the first n columns of Q, and Q_2 the remaining $m - n$ columns.

If $m < n$, R is trapezoidal, and the factorization can be written

$$AP = Q\begin{pmatrix} R_1 & R_2 \end{pmatrix},$$

where R_1 is upper triangular and R_2 is rectangular.

The matrix Q is not formed explicitly but is represented as a product of $\min(m,n)$ elementary reflectors (see the F08 Chapter Introduction for details). Routines are provided to work with Q in this representation (see Section 8).

Note also that for any $k < n$, the information returned in the first k columns of the array A represents a QR factorization of the first k columns of the permuted matrix AP.

The routine allows specified columns of A to be moved to the leading columns of AP at the start of the factorization and fixed there. The remaining columns are free to be interchanged so that at the ith stage the pivot column is chosen to be the column which maximizes the 2-norm of elements i to m over columns i to n.

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: M – INTEGER *Input*

On entry: m, the number of rows of the matrix A.

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

2: N – INTEGER *Input*

On entry: n, the number of columns of the matrix A.

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

3: 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 m by n matrix A.

On exit: if $m \geq n$, the elements below the diagonal are overwritten by details of the unitary matrix Q and the upper triangle is overwritten by the corresponding elements of the n by n upper triangular matrix R.

If $m < n$, the strictly lower triangular part is overwritten by details of the unitary matrix Q and the remaining elements are overwritten by the corresponding elements of the m by n upper trapezoidal matrix R.

The diagonal elements of R are real.

4: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08BTF (ZGEQP3) is called.

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

5: JPVT($*$) – INTEGER array *Input/Output*

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

On entry: if $\mathrm{JPVT}(j) \neq 0$, then the jth column of A is moved to the beginning of AP before the decomposition is computed and is fixed in place during the computation. Otherwise, the jth column of A is a free column (i.e., one which may be interchanged during the computation with any other free column).

On exit: details of the permutation matrix P. More precisely, if $\mathrm{JPVT}(j) = k$, then the kth column of A is moved to become the jth column of AP; in other words, the columns of AP are the columns of A in the order $\mathrm{JPVT}(1), \mathrm{JPVT}(2), \ldots, \mathrm{JPVT}(n)$.

6: TAU($*$) – ***complex*16*** array *Output*

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

On exit: further details of the unitary matrix Q.

7: WORK(max(1, 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.

8: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08BTF (ZGEQP3) is called.

If LWORK $= -1$, a workspace query is assumed; the routine only calculates the optimal size of the WORK array, returns this value as the first entry of the WORK array, and no error message related to LWORK is issued.

Suggested value: for optimal performance, LWORK $\geq (\mathrm{N}+1) \times nb$, where *nb* is the optimal ***block size***.

Constraint: LWORK $\geq \mathrm{N}+1$ or LWORK $= -1$.

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

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

10: 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.

7 Accuracy

The computed factorization is the exact factorization of a nearby matrix $(A + E)$, where

$$\|E\|_2 = O(\epsilon)\|A\|_2,$$

and ϵ is the ***machine precision***.

8 Further Comments

The total number of real floating-point operations is approximately $\frac{8}{3}n^2(3m-n)$ if $m \geq n$ or $\frac{8}{3}m^2(3n-m)$ if $m < n$.

To form the unitary matrix Q F08BTF (ZGEQP3) may be followed by a call to F08ATF (ZUNGQR):

```
CALL ZUNGQR(M,M,MIN(M,N),A,LDA,TAU,WORK,LWORK,INFO)
```

but note that the second dimension of the array A must be at least M, which may be larger than was required by F08BTF (ZGEQP3).

When $m \geq n$, it is often only the first n columns of Q that are required, and they may be formed by the call:

```
CALL ZUNGQR(M,N,N,A,LDA,TAU,WORK,LWORK,INFO)
```

To apply Q to an arbitrary complex rectangular matrix C, F08BTF (ZGEQP3) may be followed by a call to F08AUF (ZUNMQR). For example,

```
 CALL ZUNMQR('Left','Conjugate Transpose',M,P,MIN(M,N),A,LDA,TAU,
+            C,LDC,WORK,LWORK,INFO)
```

forms $C = Q^{\mathrm{H}}C$, where C is m by p.

To compute a QR factorization without column pivoting, use F08ASF (ZGEQRF).

The real analogue of this routine is F08BFF (DGEQP3).

9 Example

This example solves the linear least-squares problems

$$\min_x \|b_j - Ax_j\|_2, \qquad j = 1,2$$

for the basic solutions x_1 and x_2, where

$$A = \begin{pmatrix} 0.47 - 0.34i & -0.40 + 0.54i & 0.60 + 0.01i & 0.80 - 1.02i \\ -0.32 - 0.23i & -0.05 + 0.20i & -0.26 - 0.44i & -0.43 + 0.17i \\ 0.35 - 0.60i & -0.52 - 0.34i & 0.87 - 0.11i & -0.34 - 0.09i \\ 0.89 + 0.71i & -0.45 - 0.45i & -0.02 - 0.57i & 1.14 - 0.78i \\ -0.19 + 0.06i & 0.11 - 0.85i & 1.44 + 0.80i & 0.07 + 1.14i \end{pmatrix}$$

$$\text{and} \qquad B = \begin{pmatrix} -1.08 - 2.59i & 2.22 + 2.35i \\ -2.61 - 1.49i & 1.62 - 1.48i \\ 3.13 - 3.61i & 1.65 + 3.43i \\ 7.33 - 8.01i & -0.98 + 3.08i \\ 9.12 + 7.63i & -2.84 + 2.78i \end{pmatrix}.$$

and b_j is the jth column of the matrix B. The solution is obtained by first obtaining a QR factorization with column pivoting of the matrix A. A tolerance of 0.01 is used to estimate the rank of A from the upper triangular factor, R.

Note that the block size (NB) of 64 assumed in this example is not realistic for such a small problem, but should be suitable for large problems.

9.1 Program Text

```
*     F08BTF Example Program Text
*     Mark 21 Release. NAG Copyright 2004.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          MMAX, NB, NMAX, NRHSMX
      PARAMETER        (MMAX=8,NB=64,NMAX=8,NRHSMX=2)
      INTEGER          LDA, LDB, LWORK
      PARAMETER        (LDA=MMAX,LDB=MMAX,LWORK=(NMAX+1)*NB)
      COMPLEX *16      ONE, ZERO
      PARAMETER        (ONE=(1.0D0,0.0D0),ZERO=(0.0D0,0.0D0))
*     .. Local Scalars ..
      DOUBLE PRECISION TOL
      INTEGER          I, IFAIL, INFO, J, K, M, N, NRHS
*     .. Local Arrays ..
      COMPLEX *16      A(LDA,NMAX), B(LDB,NRHSMX), TAU(NMAX),
     +                 WORK(LWORK)
      DOUBLE PRECISION RNORM(NRHSMX), RWORK(2*NMAX)
      INTEGER          JPVT(NMAX)
      CHARACTER        CLABS(1), RLABS(1)
*     .. External Functions ..
      DOUBLE PRECISION DZNRM2
      EXTERNAL         DZNRM2
*     .. External Subroutines ..
      EXTERNAL         F06DBF, F06THF, X04DBF, ZCOPY, ZGEQP3, ZTRSM,
     +                 ZUNMQR
*     .. Intrinsic Functions ..
      INTRINSIC        ABS
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08BTF Example Program Results'
```

```
      WRITE (NOUT,*)
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) M, N, NRHS
      IF (M.LE.MMAX .AND. N.LE.NMAX .AND. M.GE.N .AND. NRHS.LE.NRHSMX)
     +    THEN
*
*        Read A and B from data file
*
         READ (NIN,*) ((A(I,J),J=1,N),I=1,M)
         READ (NIN,*) ((B(I,J),J=1,NRHS),I=1,M)
*
*        Initialize JPVT to be zero so that all columns are free
*
         CALL F06DBF(N,0,JPVT,1)
*
*        Compute the QR factorization of A
*
         CALL ZGEQP3(M,N,A,LDA,JPVT,TAU,WORK,LWORK,RWORK,INFO)
*
*        Compute C = (C1) = (Q**H)*B, storing the result in B
*                    (C2)
*
         CALL ZUNMQR('Left','Conjugate Transpose',M,NRHS,N,A,LDA,TAU,B,
     +               LDB,WORK,LWORK,INFO)
*
*        Choose TOL to reflect the relative accuracy of the input data
*
         TOL = 0.01D0
*
*        Determine and print the rank, K, of R relative to TOL
*
         DO 20 K = 1, N
            IF (ABS(A(K,K)).LE.TOL*ABS(A(1,1))) GO TO 40
   20    CONTINUE
   40    K = K - 1
*
         WRITE (NOUT,*) 'Tolerance used to estimate the rank of A'
         WRITE (NOUT,99999) TOL
         WRITE (NOUT,*) 'Estimated rank of A'
         WRITE (NOUT,99998) K
         WRITE (NOUT,*)
*
*        Compute least-squares solutions by backsubstitution in
*        R(1:K,1:K)*Y = C1, storing the result in B
*
         CALL ZTRSM('Left','Upper','No transpose','Non-Unit',K,NRHS,ONE,
     +              A,LDA,B,LDB)
*
*        Compute estimates of the square roots of the residual sums of
*        squares (2-norm of each of the columns of C2)
*
         DO 60 J = 1, NRHS
            RNORM(J) = DZNRM2(M-K,B(K+1,J),1)
   60    CONTINUE
*
*        Set the remaining elements of the solutions to zero (to give
*        the basic solutions)
*
         CALL F06THF('General',N-K,NRHS,ZERO,ZERO,B(K+1,1),LDB)
*
*        Permute the least-squares solutions stored in B to give X = P*Y
*
         DO 100 J = 1, NRHS
            DO 80 I = 1, N
               WORK(JPVT(I)) = B(I,J)
   80       CONTINUE
            CALL ZCOPY(N,WORK,1,B(1,J),1)
  100    CONTINUE
*
*        Print least-squares solutions
```

```
*
         IFAIL = 0
         CALL X04DBF('General',' ',N,NRHS,B,LDB,'Bracketed','F7.4',
     +               'Least-squares solution(s)','Integer',RLABS,
     +               'Integer',CLABS,80,0,IFAIL)
*
*        Print the square roots of the residual sums of squares
*
         WRITE (NOUT,*)
         WRITE (NOUT,*)
     +     'Square root(s) of the residual sum(s) of squares'
         WRITE (NOUT,99999) (RNORM(J),J=1,NRHS)
*
      ELSE
         WRITE (NOUT,*)
     +     'One or more of MMAX, NMAX and NRHSMX is too small, ',
     +     'and/or M.LT.N'
      END IF
*
99999 FORMAT (3X,1P,7E11.2)
99998 FORMAT (1X,I8)
      END
```

9.2 Program Data

```
F08BTF Example Program Data

   5             4             2                 :Values of M, N and NRHS

 ( 0.47,-0.34) (-0.40, 0.54) ( 0.60, 0.01) ( 0.80,-1.02)
 (-0.32,-0.23) (-0.05, 0.20) (-0.26,-0.44) (-0.43, 0.17)
 ( 0.35,-0.60) (-0.52,-0.34) ( 0.87,-0.11) (-0.34,-0.09)
 ( 0.89, 0.71) (-0.45,-0.45) (-0.02,-0.57) ( 1.14,-0.78)
 (-0.19, 0.06) ( 0.11,-0.85) ( 1.44, 0.80) ( 0.07, 1.14) :End of matrix A

 (-1.08,-2.59) ( 2.22, 2.35)
 (-2.61,-1.49) ( 1.62,-1.48)
 ( 3.13,-3.61) ( 1.65, 3.43)
 ( 7.33,-8.01) (-0.98, 3.08)
 ( 9.12, 7.63) (-2.84, 2.78)                              :End of matrix B
```

9.3 Program Results

```
 F08BTF Example Program Results

 Tolerance used to estimate the rank of A
      1.00E-02
 Estimated rank of A
        3

 Least-squares solution(s)
                   1                 2
 1  ( 0.0000, 0.0000) ( 0.0000, 0.0000)
 2  ( 2.7020, 8.0911) (-2.2682,-2.9884)
 3  ( 2.8888, 2.5012) ( 0.9779, 1.3565)
 4  ( 2.7100, 0.4791) (-1.3734, 0.2212)

 Square root(s) of the residual sum(s) of squares
      2.51E-01   8.10E-02
```