# NAG Fortran Library Routine Document

# F08ZTF (ZGGRQF)

**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

F08ZTF (ZGGRQF) computes a generalized $RQ$ factorization of a complex matrix pair $(A, B)$, where $A$ is an $m$ by $n$ matrix and $B$ is a $p$ by $n$ matrix.

## 2 Specification

```
      SUBROUTINE F08ZTF (M, P, N, A, LDA, TAUA, B, LDB, TAUB, WORK, LWORK,
     1                   INFO)
      INTEGER            M, P, N, LDA, LDB, LWORK, INFO
      complex*16         A(LDA,*), TAUA(*), B(LDB,*), TAUB(*), WORK(*)
```

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

## 3 Description

F08ZTF (ZGGRQF) forms the generalized $RQ$ factorization of an $m$ by $n$ matrix $A$ and a $p$ by $n$ matrix $B$

$$A = RQ, \quad B = ZTQ,$$

where $Q$ is an $n$ by $n$ unitary matrix, $Z$ is a $p$ by $p$ unitary matrix and $R$ and $T$ are of the form

$$R = \begin{cases} \begin{array}{c} \\ m \end{array}\!\!\begin{array}{c} \begin{array}{cc} n-m & m \end{array} \\ \begin{pmatrix} 0 & R_{12} \end{pmatrix} \end{array}; & \text{if } m \leq n, \\[2ex] \begin{array}{c} \\ m-n \\ n \end{array}\!\!\begin{array}{c} n \\ \begin{pmatrix} R_{11} \\ R_{21} \end{pmatrix} \end{array}; & \text{if } m > n, \end{cases}$$

with $R_{12}$ or $R_{21}$ upper triangular,

$$T = \begin{cases} \begin{array}{c} \\ n \\ p-n \end{array}\!\!\begin{array}{c} n \\ \begin{pmatrix} T_{11} \\ 0 \end{pmatrix} \end{array}; & \text{if } p \geq n, \\[2ex] \begin{array}{c} \\ p \end{array}\!\!\begin{array}{c} \begin{array}{cc} p & n-p \end{array} \\ \begin{pmatrix} T_{11} & T_{12} \end{pmatrix} \end{array}; & \text{if } p < n, \end{cases}$$

with $T_{11}$ upper triangular.

In particular, if $B$ is square and nonsingular, the generalized $RQ$ factorization of $A$ and $B$ implicitly gives the $RQ$ factorization of $AB^{-1}$ as

$$AB^{-1} = \left(RT^{-1}\right)Z^{\mathrm{H}}.$$

## 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

Anderson E, Bai Z and Dongarra J (1992) Generalized *QR* factorization and its applications *Linear Algebra Appl. (Volume 162–164)* 243–271

Hammarling S (1987) The numerical solution of the general Gauss-Markov linear model *Mathematics in Signal Processing* (ed T S Durrani, J B Abbiss, J E Hudson, R N Madan, J G McWhirter, and T A Moore) 441–456 Oxford University Press

Paige C C (1990) Some aspects of generalized *QR* factorizations *Mathematics in Signal Processing* (ed M G Cox and S Hammarling) 73–91 Oxford University Press

## 5 Parameters

1: M – INTEGER *Input*

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

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

2: P – INTEGER *Input*

*On entry*: $p$, the number of rows of the matrix $B$.

*Constraint*: $\mathrm{P} \geq 0$.

3: N – INTEGER *Input*

*On entry*: $n$, the number of columns of the matrices $A$ and $B$.

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

4: 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 $m \leq n$, the upper triangle of the subarray $\mathrm{A}(1:m, n-m+1:n)$ contains the $m$ by $m$ upper triangular matrix $R_{12}$.

If $m \geq n$, the elements on and above the $(m-n)$th subdiagonal contain the $m$ by $n$ upper trapezoidal matrix $R$; the remaining elements, with the array TAUA, represent the unitary matrix $Q$ as a product of $\min(m, n)$ elementary reflectors (see Section 3.3.6 in the F08 Chapter Introduction).

5: LDA – INTEGER *Input*

*On entry*: the first dimension of the array A as declared in the (sub)program from which F08ZTF (ZGGRQF) is called.

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

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

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

*On exit*: the scalar factors of the elementary reflectors which represent the unitary matrix $Q$.

7: B(LDB,$*$) – ***complex*16*** array *Input/Output*

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

*On entry*: the $p$ by $n$ matrix $B$.

*On exit*: the elements on and above the diagonal of the array contain the $\min(p, n)$ by $n$ upper trapezoidal matrix $T$ ($T$ is upper triangular if $p \geq n$); the elements below the diagonal, with the array TAUB, represent the unitary matrix $Z$ as a product of elementary reflectors (see Section 3.3.6 in the F08 Chapter Introduction).

8: LDB – INTEGER *Input*

*On entry*: the first dimension of the array B as declared in the (sub)program from which F08ZTF (ZGGRQF) is called.

*Constraint*: LDB $\geq \max(1, \mathrm{P})$.

9: TAUB($*$) – ***complex*16*** array *Output*

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

*On exit*: the scalar factors of the elementary reflectors which represent the unitary matrix $Z$.

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

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

*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 F08ZTF (ZGGRQF) 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 $\geq \max(\mathrm{N}, \mathrm{M}, \mathrm{P}) \times \max(nb1, nb2, nb3)$, where $nb1$ is the optimal ***block size*** for the *RQ* factorization of an $m$ by $n$ matrix by F08CVF (ZGERQF), $nb2$ is the optimal ***block size*** for the *QR* factorization of a $p$ by $n$ matrix by F08ASF (ZGEQRF), and $nb3$ is the optimal ***block size*** for a call of F08CKF (DORMRQ).

*Constraint*: LWORK $\geq \max(1, \mathrm{N}, \mathrm{M}, \mathrm{P})$ or LWORK $= -1$.

12: 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 $i$th parameter had an illegal value. An explanatory message is output, and execution of the program is terminated.

## 7 Accuracy

The computed generalized *RQ* factorization is the exact factorization for nearby matrices $(A + E)$ and $(B + F)$, where

$$\|E\|_2 = O\,\epsilon\|A\|_2 \quad \text{and} \quad \|F\|_2 = O\,\epsilon\|B\|_2,$$

and $\epsilon$ is the ***machine precision***.

## 8 Further Comments

The unitary matrices $Q$ and $Z$ may be formed explicitly by calls to F08CWF (ZUNGRQ) and F08ATF (ZUNGQR) respectively. F08CXF (ZUNMRQ) may be used to multiply $Q$ by another matrix and F08AUF (ZUNMQR) may be used to multiply $Z$ by another matrix.

The real analogue of this routine is F08ZFF (DGGRQF).

## 9 Example

This example solves the general Gauss-Markov linear model problem

$$\min_x \|y\|_2 \quad \text{subject to } d = Ax + By$$

where

$$A = \begin{pmatrix} 0.96 - 0.81i & -0.03 + 0.96i & -0.91 + 2.06i & -0.05 + 0.41i \\ -0.98 + 1.98i & -1.20 + 0.19i & -0.66 + 0.42i & -0.81 + 0.56i \\ 0.62 - 0.46i & 1.01 + 0.02i & 0.63 - 0.17i & -1.11 + 0.60i \\ 0.37 + 0.38i & 0.19 - 0.54i & -0.98 - 0.36i & 0.22 - 0.20i \\ 0.83 + 0.51i & 0.20 + 0.01i & -0.17 - 0.46i & 1.47 + 1.59i \\ 1.08 - 0.28i & 0.20 - 0.12i & -0.07 + 1.23i & 0.26 + 0.26i \end{pmatrix},$$

$$B = \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix}, \quad c = \begin{pmatrix} -2.54 + 0.09i \\ 1.65 - 2.26i \\ -2.11 - 3.96i \\ 1.82 + 3.30i \\ -6.41 + 3.77i \\ 2.07 + 0.66i \end{pmatrix} \quad \text{and} \quad d = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

The constraints $Bx = d$ correspond to $x_1 = x_3$ and $x_2 = x_4$.

The solution is obtained by first obtaining a generalized $QR$ factorization of the matrix pair $(A, B)$. The example illustrates the general solution process, although the above data corresponds to a simple weighted least squares problem.

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

```
*     F08ZTF Example Program Text
*     Mark 21 Release. NAG Copyright 2004.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          MMAX, NB, NMAX, PMAX
      PARAMETER        (MMAX=10,NB=64,NMAX=10,PMAX=10)
      INTEGER          LDA, LDB, LWORK
      PARAMETER        (LDA=MMAX,LDB=PMAX,LWORK=NB*(MMAX+NMAX))
      COMPLEX *16      ONE
      PARAMETER        (ONE=(1.0D0,0.0D0))
*     .. Local Scalars ..
      DOUBLE PRECISION RNORM
      INTEGER          I, INFO, J, M, N, P
*     .. Local Arrays ..
      COMPLEX *16      A(LDA,NMAX), B(LDB,NMAX), C(MMAX), D(PMAX),
     +                 TAUA(MMAX+NMAX), TAUB(PMAX), WORK(LWORK), X(NMAX)
*     .. External Functions ..
      DOUBLE PRECISION DZNRM2
      EXTERNAL         DZNRM2
*     .. External Subroutines ..
      EXTERNAL         ZCOPY, ZGEMV, ZGGRQF, ZTRMV, ZTRTRS, ZUNMQR,
     +                 ZUNMRQ
*     .. Intrinsic Functions ..
      INTRINSIC        MIN
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08ZTF Example Program Results'
      WRITE (NOUT,*)
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) M, N, P
      IF (M.LE.MMAX .AND. N.LE.NMAX .AND. P.LE.PMAX .AND. P.LE.N .AND.
     +    N.LE.(M+P)) THEN
```

```
*
*        Read A, B, c and d from data file
*
         READ (NIN,*) ((A(I,J),J=1,N),I=1,M)
         READ (NIN,*) ((B(I,J),J=1,N),I=1,P)
         READ (NIN,*) (C(I),I=1,M)
         READ (NIN,*) (D(I),I=1,P)
*
*        Compute the generalized RQ factorization of (A,B) as
*        B = (0 R12)*Q,   A = Z*(T11 T12 T13)*Q, where R12, T11 and T22
*                               ( 0  T22 T23)
*        are upper triangular
*
         CALL ZGGRQF(P,M,N,B,LDB,TAUB,A,LDA,TAUA,WORK,LWORK,INFO)
*
*        Compute (f1) = (Z**H)*c, storing the result in C
*                (f2)
*
         CALL ZUNMQR('Left','Conjugate transpose',M,1,MIN(M,N),A,LDA,
     +               TAUA,C,M,WORK,LWORK,INFO)
*
*        Putting Q*x = (y1), solve R12*w = d for w, storing result in D
*                      (w )
*
         CALL ZTRTRS('Upper','No transpose','Non-unit',P,1,B(1,N-P+1),
     +               LDB,D,P,INFO)
*
         IF (INFO.GT.0) THEN
            WRITE (NOUT,*)
     +        'The upper triangular factor, R12, of B is singular, '
            WRITE (NOUT,*)
     +        'the least squares solution could not be computed'
            GO TO 40
         END IF
*
*        Form f1 - T1*w, T1 = (T12 T13), in C
*
         CALL ZGEMV('No transpose',N-P,P,-ONE,A(1,N-P+1),LDA,D,1,ONE,C,
     +              1)
*
*        Solve T11*y1 = f1 - T1*w for y1, storing result in C
*
         CALL ZTRTRS('Upper','No transpose','Non-unit',N-P,1,A,LDA,C,
     +               N-P,INFO)
*
         IF (INFO.GT.0) THEN
            WRITE (NOUT,*)
     +        'The upper triangular factor, T11, of A is singular, '
            WRITE (NOUT,*)
     +        'the least squares solution could not be computed'
            GO TO 40
         END IF
*
*        Copy y into X (first y1, then w)
*
         CALL ZCOPY(N-P,C,1,X,1)
         CALL ZCOPY(P,D,1,X(N-P+1),1)
*
*        Compute x = (Q**H)*y
*
         CALL ZUNMRQ('Left','Conjugate transpose',N,1,P,B,LDB,TAUB,X,N,
     +               WORK,LWORK,INFO)
*
*        Putting w = (y2), form f2 - T22*y2 - T23*y3
*                    (y3)
*
*        T22*y2
*
         CALL ZTRMV('Upper','No transpose','Non-unit',MIN(M,N)-N+P,
     +              A(N-P+1,N-P+1),LDA,D,1)
*
```

```
*        f2 - T22*y2
*
         DO 20 I = 1, MIN(M,N) - N + P
            C(N-P+I) = C(N-P+I) - D(I)
   20    CONTINUE
         IF (M.LT.N) THEN
*
*           f2 - T22*y2 - T23*y3
*
            CALL ZGEMV('No transpose',M-N+P,N-M,-ONE,A(N-P+1,M+1),LDA,
     +                 D(M-N+P+1),1,ONE,C(N-P+1),1)
         END IF
*
*        Compute estimate of the square root of the residual sum of
*        squares norm(r) = norm(f2 - T22*y2 - T23*y3)
*
         RNORM = DZNRM2(M-(N-P),C(N-P+1),1)
*
*        Print least squares solution x
*
         WRITE (NOUT,*) 'Constrained least squares solution'
         WRITE (NOUT,99999) (X(I),I=1,N)
*
*        Print estimate of the square root of the residual sum of
*        squares
*
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Square root of the residual sum of squares'
         WRITE (NOUT,99998) RNORM
      ELSE
         WRITE (NOUT,*)
     +     'One or more of MMAX, NMAX or PMAX is too small, ',
     +     'and/or N.LT.P or N.GT.(M+P)'
      END IF
   40 CONTINUE
      STOP
*
99999 FORMAT (4(' (',F7.4,',',F7.4,')',:))
99998 FORMAT (1X,1P,E10.2)
      END
```

## 9.2 Program Data

```
F08ZTF Example Program Data

  6             4             2                          :Values of M, N and P

( 0.96,-0.81) (-0.03, 0.96) (-0.91, 2.06) (-0.05, 0.41)
(-0.98, 1.98) (-1.20, 0.19) (-0.66, 0.42) (-0.81, 0.56)
( 0.62,-0.46) ( 1.01, 0.02) ( 0.63,-0.17) (-1.11, 0.60)
( 0.37, 0.38) ( 0.19,-0.54) (-0.98,-0.36) ( 0.22,-0.20)
( 0.83, 0.51) ( 0.20, 0.01) (-0.17,-0.46) ( 1.47, 1.59)
( 1.08,-0.28) ( 0.20,-0.12) (-0.07, 1.23) ( 0.26, 0.26) :End of matrix A

( 1.00, 0.00) ( 0.00, 0.00) (-1.00, 0.00) ( 0.00, 0.00)
( 0.00, 0.00) ( 1.00, 0.00) ( 0.00, 0.00) (-1.00, 0.00) :End of matrix B

(-2.54, 0.09)
( 1.65,-2.26)
(-2.11,-3.96)
( 1.82, 3.30)
(-6.41, 3.77)
( 2.07, 0.66)                                            :End of vector c

( 0.00, 0.00)
( 0.00, 0.00)                                            :End of vector d
```

## 9.3 Program Results

```
F08ZTF Example Program Results

Constrained least squares solution
( 1.0874,-1.9621) (-0.7409, 3.7297) ( 1.0874,-1.9621) (-0.7409, 3.7297)

Square root of the residual sum of squares
  1.59E-01
```