NAG Fortran Library Routine Document

F08ZEF (DGGQRF)

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

F08ZEF (DGGQRF) computes a generalized QR factorization of a real matrix pair (A, B), where A is an n by m matrix and B is an n by p matrix.

2   Specification

```
      SUBROUTINE F08ZEF (N, M, P, A, LDA, TAUA, B, LDB, TAUB, WORK, LWORK,
     1                   INFO)
      INTEGER          N, M, P, LDA, LDB, LWORK, INFO
      double precision A(LDA,*), TAUA(*), B(LDB,*), TAUB(*), WORK(*)
```

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

3   Description

F08ZEF (DGGQRF) forms the generalized QR factorization of an n by m matrix A and an n by p matrix B

$$A = QR, \quad B = QTZ,$$

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

$$R = \begin{cases} \begin{matrix} & m \\ m \\ n-m \end{matrix} \begin{pmatrix} R_{11} \\ 0 \end{pmatrix}, & \text{if } n \geq m; \\[2em] \begin{matrix} & n & m-n \\ n \end{matrix} \begin{pmatrix} R_{11} & R_{12} \end{pmatrix}, & \text{if } n < m, \end{cases}$$

with R_{11} upper triangular,

$$T = \begin{cases} \begin{matrix} & p-n & n \\ n \end{matrix} \begin{pmatrix} 0 & T_{12} \end{pmatrix}, & \text{if } n \leq p, \\[2em] \begin{matrix} & & p \\ n-p \\ p \end{matrix} \begin{pmatrix} T_{11} \\ T_{21} \end{pmatrix}, & \text{if } n > p, \end{cases}$$

with T_{12} or T_{21} upper triangular.

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

$$B^{-1}A = Z^{\mathrm{T}}\left(T^{-1}R\right).$$

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:      N – INTEGER   *Input*

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

     Constraint: $N \geq 0$.

2:      M – INTEGER   *Input*

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

     Constraint: $M \geq 0$.

3:      P – INTEGER   *Input*

     On entry: p, the number of columns of the matrix B.

     Constraint: $P \geq 0$.

4:      A(LDA,∗) – ***double precision*** array   *Input/Output*

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

     On entry: the n by m matrix A.

     On exit: the elements on and above the diagonal of the array contain the $\min(n, m)$ by m upper trapezoidal matrix R (R is upper triangular if $n \geq m$); the elements below the diagonal, with the array TAUA, represent the orthogonal matrix Q as a product of $\min(n, m)$ 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 F08ZEF (DGGQRF) is called.

     Constraint: $LDA \geq \max(1, N)$.

6:      TAUA(∗) – ***double precision*** array   *Output*

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

     On exit: the scalar factors of the elementary reflectors which represent the orthogonal matrix Q.

7:      B(LDB,∗) – ***double precision*** array   *Input/Output*

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

     On entry: the n by p matrix B.

     On exit: if $n \leq p$, the upper triangle of the subarray $B(1 : n, p - n + 1 : p)$ contains the n by n upper triangular matrix T_{12}.

     If $n > p$, the elements on and above the $(n - p)$th subdiagonal contain the n by p upper trapezoidal matrix T; the remaining elements, with the array TAUB, represent the orthogonal 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 F08ZEF (DGGQRF) is called.

   Constraint: $\text{LDB} \geq \max(1, \text{N})$.

9:   TAUB($*$) – **double precision** array   *Output*

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

   On exit: the scalar factors of the elementary reflectors which represent the orthogonal matrix Z.

10:   WORK($*$) – **double precision** array   *Workspace*

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

   On exit: if INFO = 0, 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 F08ZEF (DGGQRF) 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, $\text{LWORK} \geq \max(\text{N}, \text{M}, \text{P}) \times \max(nb1, nb2, nb3)$, where $nb1$ is the optimal **block size** for the QR factorization of an n by m matrix, $nb2$ is the optimal **block size** for the RQ factorization of an n by p matrix, and $nb3$ is the optimal **block size** for a call of F08AGF (DORMQR).

   Constraint: $\text{LWORK} \geq \max(1, \text{N}, \text{M}, \text{P})$ 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 ith parameter had an illegal value. An explanatory message is output, and execution of the program is terminated.

7   Accuracy

The computed generalized QR 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 ϵ is the **machine precision**.

8   Further Comments

The orthogonal matrices Q and Z may be formed explicitly by calls to F08AFF (DORGQR) and F08CJF (DORGRQ) respectively. F08AGF (DORMQR) may be used to multiply Q by another matrix and F08CKF (DORMRQ) may be used to multiply Z by another matrix.

The complex analogue of this routine is F08ZSF (ZGGQRF).

9 Example

This example solves the general Gauss-Markov linear model problem

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

where

$$A = \begin{pmatrix} -0.57 & -1.28 & -0.39 \\ -1.93 & 1.08 & -0.31 \\ 2.30 & 0.24 & -0.40 \\ -0.02 & 1.03 & -1.43 \end{pmatrix}, \quad B = \begin{pmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 \\ 0 & 0 & 2.0 & 0 \\ 0 & 0 & 0 & 5.0 \end{pmatrix} \quad \text{and} \quad d = \begin{pmatrix} 1.32 \\ -4.00 \\ 5.52 \\ 3.24 \end{pmatrix}.$$

The solution is obtained by first computing 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

```
*       F08ZEF 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=NMAX,LDB=NMAX,LWORK=NB*(MMAX+PMAX))
        DOUBLE PRECISION ONE, ZERO
        PARAMETER         (ONE=1.0D0,ZERO=0.0D0)
*       .. Local Scalars ..
        DOUBLE PRECISION RNORM
        INTEGER           I, INFO, J, M, N, P
*       .. Local Arrays ..
        DOUBLE PRECISION A(LDA,MMAX), B(LDB,PMAX), D(NMAX), TAUA(MMAX),
       +                 TAUB(MMAX+PMAX), WORK(LWORK), Y(PMAX)
*       .. External Functions ..
        DOUBLE PRECISION DNRM2
        EXTERNAL          DNRM2
*       .. External Subroutines ..
        EXTERNAL          DCOPY, DGEMV, DGGQRF, DORMQR, DORMRQ, DTRTRS,
       +                 F06FBF
*       .. Intrinsic Functions ..
        INTRINSIC         MAX, MIN
*       .. Executable Statements ..
        WRITE (NOUT,*) 'F08ZEF Example Program Results'
        WRITE (NOUT,*)
*       Skip heading in data file
        READ (NIN,*)
        READ (NIN,*) N, M, P
        IF (M.LE.MMAX .AND. N.LE.NMAX .AND. P.LE.PMAX .AND. M.LE.N .AND.
       +    N.LE.(M+P)) THEN
*
*       Read A, B and D from data file
*
        READ (NIN,*) ((A(I,J),J=1,M),I=1,N)
        READ (NIN,*) ((B(I,J),J=1,P),I=1,N)
        READ (NIN,*) (D(I),I=1,N)
*
*       Compute the generalized QR factorization of (A,B) as
*       A = Q*(R),   B = Q*(T11 T12)*Z
*           (0)          ( 0   T22)
*
        CALL DGGQRF(N,M,P,A,LDA,TAUA,B,LDB,TAUB,WORK,LWORK,INFO)
*
*       Compute c = (c1) = (Q**T)*d, storing the result in D
```

```
*                  (c2)
*
         CALL DORMQR('Left','Transpose',N,1,M,A,LDA,TAUA,D,N,WORK,LWORK,
     +               INFO)
*
*        Putting Z*y = w = (w1), set w1 = 0, storing the result in Y1
*                          (w2)
*
         CALL F06FBF(M+P-N,ZERO,Y,1)
*
         IF (N.GT.M) THEN
*
*           Copy c2 into Y2
*
            CALL DCOPY(N-M,D(M+1),1,Y(M+P-N+1),1)
*
*           Solve T22*w2 = c2 for w2, storing result in Y2
*
            CALL DTRTRS('Upper','No transpose','Non-unit',N-M,1,
     +                  B(M+1,M+P-N+1),LDB,Y(M+P-N+1),N-M,INFO)
*
            IF (INFO.GT.0) THEN
               WRITE (NOUT,*)
     +           'The upper triangular factor, T22, of B is singular, '
               WRITE (NOUT,*)
     +           'the least squares solution could not be computed'
               GO TO 20
            END IF
*
*           Compute estimate of the square root of the residual sum of
*           squares norm(y) = norm(w2)
*
            RNORM = DNRM2(N-M,Y(M+P-N+1),1)
*
*           Form c1 - T12*w2 in D
*
            CALL DGEMV('No transpose',M,N-M,-ONE,B(1,M+P-N+1),LDB,
     +                 Y(M+P-N+1),1,ONE,D,1)
         END IF
*
*        Solve R*x = c1 - T12*w2 for x
*
         CALL DTRTRS('Upper','No transpose','Non-unit',M,1,A,LDA,D,M,
     +               INFO)
*
         IF (INFO.GT.0) THEN
            WRITE (NOUT,*)
     +        'The upper triangular factor, R, of A is singular, '
            WRITE (NOUT,*)
     +        'the least squares solution could not be computed'
            GO TO 20
         END IF
*
*        Compute y = (Z**T)*w
*
         CALL DORMRQ('Left','Transpose',P,1,MIN(N,P),B(MAX(1,N-P+1),1),
     +               LDB,TAUB,Y,P,WORK,LWORK,INFO)
*
*        Print least squares solution x
*
         WRITE (NOUT,*) 'Generalized least squares solution'
         WRITE (NOUT,99999) (D(I),I=1,M)
*
*        Print residual vector y
*
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Residual vector'
         WRITE (NOUT,99998) (Y(I),I=1,P)
*
*        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 M.LT.N or N.GT.(M+P)'
      END IF
   20 CONTINUE
      STOP
*
99999 FORMAT (1X,7F11.4)
99998 FORMAT (3X,1P,7E11.2)
      END
```

9.2   Program Data

```
F08ZEF Example Program Data

  4       3       4            :Values of N, M and P

 -0.57  -1.28  -0.39
 -1.93   1.08  -0.31
  2.30   0.24  -0.40
 -0.02   1.03  -1.43          :End of matrix A

  0.50   0.00   0.00   0.00
  0.00   1.00   0.00   0.00
  0.00   0.00   2.00   0.00
  0.00   0.00   0.00   5.00 :End of matrix B

  1.32
 -4.00
  5.52
  3.24                       :End of vector d
```

9.3   Program Results

```
F08ZEF Example Program Results

Generalized least squares solution
     1.9889    -1.0058    -2.9911

Residual vector
    -6.37E-04  -2.45E-03  -4.72E-03   7.70E-03

Square root of the residual sum of squares
     9.38E-03
```