NAG Fortran Library Routine Document

F08ZFF (DGGRQF)

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

F08ZFF (DGGRQF) computes a generalized RQ factorization of a real matrix pair (A, B), where A is an m by n matrix and B is a p by n matrix.

2 Specification

```
      SUBROUTINE F08ZFF (M, P, N, A, LDA, TAUA, B, LDB, TAUB, WORK, LWORK,
     1                   INFO)
      INTEGER            M, P, N, LDA, LDB, LWORK, INFO
      double precision   A(LDA,*), TAUA(*), B(LDB,*), TAUB(*), WORK(*)
```

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

3 Description

F08ZFF (DGGRQF) 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 orthogonal matrix, Z is a p by p orthogonal matrix and R and T are of the form

$$R = \begin{cases} \begin{array}{c} \\ m \end{array} \begin{pmatrix} \overset{n-m}{0} & \overset{m}{R_{12}} \end{pmatrix}; & \text{if } m \le n, \\ \begin{array}{c} m-n \\ n \end{array} \overset{n}{\begin{pmatrix} R_{11} \\ R_{21} \end{pmatrix}}; & \text{if } m > n, \end{cases}$$

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

$$T = \begin{cases} \begin{array}{c} n \\ p-n \end{array} \overset{n}{\begin{pmatrix} T_{11} \\ 0 \end{pmatrix}}; & \text{if } p \ge n, \\ p \begin{pmatrix} \overset{p}{T_{11}} & \overset{n-p}{T_{12}} \end{pmatrix}; & \text{if } p < n, \end{cases}$$

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

$$AB^{-1} = (RT^{-1})Z^{\mathrm{T}}.$$

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,$*$) – ***double precision*** 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 orthogonal 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 F08ZFF (DGGRQF) is called.

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

6: TAUA($*$) – ***double precision*** 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 orthogonal matrix Q.

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

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

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 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 F08ZFF (DGGRQF) is called.

Constraint: LDB $\geq \max(1, P)$.

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

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

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 F08ZFF (DGGRQF) 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 $\geq \max(N, M, P) \times \max(nb1, nb2, nb3)$, where $nb1$ is the optimal ***block size*** for the *RQ* factorization of an m by n matrix by F08CHF (DGERQF), $nb2$ is the optimal ***block size*** for the *QR* factorization of a p by n matrix by F08AEF (DGEQRF), and $nb3$ is the optimal ***block size*** for a call of F08CKF (DORMRQ).

Constraint: LWORK $\geq \max(1, N, M, P)$ 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 *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 ϵ is the ***machine precision***.

8 Further Comments

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

The complex analogue of this routine is F08ZTF (ZGGRQF).

9 Example

This example solves the linear equality constrained least squares problem

$$\min_x \|c - Ax\|_2 \quad \text{subject to} \quad Bx = d$$

where

$$A = \begin{pmatrix} -0.57 & -1.28 & -0.39 & 0.25 \\ -1.93 & 1.08 & -0.31 & -2.14 \\ 2.30 & 0.24 & 0.40 & -0.35 \\ -1.93 & 0.64 & -0.66 & 0.08 \\ 0.15 & 0.30 & 0.15 & -2.13 \\ -0.02 & 1.03 & -1.43 & 0.50 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix},$$

$$c = \begin{pmatrix} -1.50 \\ -2.14 \\ 1.23 \\ -0.54 \\ -1.68 \\ 0.82 \end{pmatrix} \quad \text{and} \quad d = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

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

The solution is obtained by first computing a generalized RQ factorization of the matrix pair (B, A). The example illustrates the general solution process.

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

```
*     F08ZFF 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))
      DOUBLE PRECISION ONE
      PARAMETER        (ONE=1.0D0)
*     .. Local Scalars ..
      DOUBLE PRECISION RNORM
      INTEGER          I, INFO, J, M, N, P
*     .. Local Arrays ..
      DOUBLE PRECISION A(LDA,NMAX), B(LDB,NMAX), C(MMAX), D(PMAX),
     +                 TAUA(MMAX+NMAX), TAUB(PMAX), WORK(LWORK), X(NMAX)
*     .. External Functions ..
      DOUBLE PRECISION DNRM2
      EXTERNAL         DNRM2
*     .. External Subroutines ..
      EXTERNAL         DCOPY, DGEMV, DGGRQF, DORMQR, DORMRQ, DTRMV,
     +                 DTRTRS
*     .. Intrinsic Functions ..
      INTRINSIC        MIN
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08ZFF 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 DGGRQF(P,M,N,B,LDB,TAUB,A,LDA,TAUA,WORK,LWORK,INFO)
*
*        Compute (f1) = (Z**T)*c, storing the result in C
*                (f2)
*
         CALL DORMQR('Left','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 DTRTRS('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 DGEMV('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 DTRTRS('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 DCOPY(N-P,C,1,X,1)
         CALL DCOPY(P,D,1,X(N-P+1),1)
*
*        Compute x = (Q**T)*y
*
         CALL DORMRQ('Left','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 DTRMV('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 DGEMV('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 = DNRM2(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 (1X,7F11.4)
99998 FORMAT (3X,1P,E11.2)
      END
```

9.2 Program Data

```
F08ZFF Example Program Data

  6      4      2           :Values of M, N and P

 -0.57  -1.28  -0.39   0.25
 -1.93   1.08  -0.31  -2.14
  2.30   0.24   0.40  -0.35
 -1.93   0.64  -0.66   0.08
  0.15   0.30   0.15  -2.13
 -0.02   1.03  -1.43   0.50 :End of matrix A

  1.00   0.00  -1.00   0.00
  0.00   1.00   0.00  -1.00 :End of matrix B

 -1.50
 -2.14
  1.23
 -0.54
 -1.68
  0.82                      :End of vector c

  0.00
  0.00                      :End of vector d
```

9.3 Program Results

```
F08ZFF Example Program Results

Constrained least squares solution
     0.4890     0.9975     0.4890     0.9975

Square root of the residual sum of squares
     2.51E-02
```