NAG 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,       &
                   INFO)

INTEGER            N, M, P, LDA, LDB, LWORK, INFO
REAL (KIND=nag_wp) A(LDA,*), TAUA(min(N,M)), B(LDB,*), TAUB(min(N,P)),     &
                   WORK(max(1,LWORK))
```

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

$$R = \begin{cases} \begin{array}{cc} & m \\ \begin{array}{r} m \\ n-m \end{array} & \begin{pmatrix} R_{11} \\ 0 \end{pmatrix} \end{array}, & \text{if } n \geq m; \\ \begin{array}{cc} & \begin{array}{cc} n & m-n \end{array} \\ n & \begin{pmatrix} R_{11} & R_{12} \end{pmatrix} \end{array}, & \text{if } n < m, \end{cases}$$

with R_{11} upper triangular,

$$T = \begin{cases} \begin{array}{cc} & \begin{array}{cc} p-n & n \end{array} \\ n & \begin{pmatrix} 0 & T_{12} \end{pmatrix} \end{array}, & \text{if } n \leq p, \\ \begin{array}{cc} & p \\ \begin{array}{r} n-p \\ p \end{array} & \begin{pmatrix} T_{11} \\ T_{21} \end{pmatrix} \end{array}, & \text{if } n > p, \end{cases}$$

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 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* (eds 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 . *In Reliable Numerical Computation* (eds M G Cox and S Hammarling) 73–91 Oxford University Press

5 Arguments

1: N – INTEGER *Input*

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

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

2: M – INTEGER *Input*

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

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

3: P – INTEGER *Input*

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

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

4: A(LDA, $*$) – REAL (KIND=nag_wp) array *Input/Output*

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

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

6: TAUA($\min(\mathrm{N}, \mathrm{M})$) – REAL (KIND=nag_wp) array *Output*

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

7: B(LDB, $*$) – REAL (KIND=nag_wp) array *Input/Output*

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

On entry: the n by p matrix B.

On exit: if $n \leq p$, the upper triangle of the subarray $\mathrm{B}(1:n, p-n+1:p)$ 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($\min(\text{N}, \text{P})$) – REAL (KIND=nag_wp) array *Output*

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

10: WORK($\max(1, \text{LWORK})$) – REAL (KIND=nag_wp) array *Workspace*

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

12: INFO – INTEGER *Output*

On exit: $\text{INFO} = 0$ unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

$\text{INFO} < 0$

If $\text{INFO} = -i$, argument i 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 Parallelism and Performance

F08ZEF (DGGQRF) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F08ZEF (DGGQRF) makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 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).

10 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.

10.1 Program Text

```
    Program f08zefe

!     F08ZEF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: dgemv, dggqrf, dnrm2, dormqr, dormrq, dtrtrs,     &
                             nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Real (Kind=nag_wp), Parameter    :: one = 1.0E0_nag_wp
      Real (Kind=nag_wp), Parameter    :: zero = 0.0E0_nag_wp
      Integer, Parameter               :: nb = 64, nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: rnorm
      Integer                          :: i, info, lda, ldb, lwork, m, n, p
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:,:), b(:,:), d(:), taua(:),       &
                                          taub(:), work(:), y(:)
!     .. Intrinsic Procedures ..
      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
      lda = n
      ldb = n
      lwork = nb*(m+p)
      Allocate (a(lda,m),b(ldb,p),d(n),taua(m),taub(m+p),work(lwork),y(p))

!     Read A, B and D from data file
      Read (nin,*)(a(i,1:m),i=1,n)
      Read (nin,*)(b(i,1:p),i=1,n)
```

```
      Read (nin,*) d(1:n)

!     Compute the generalized QR factorization of (A,B) as
!     A = Q*(R),   B = Q*(T11 T12)*Z
!           (0)          ( 0  T22)
!     The NAG name equivalent of dggqrf is f08zef
      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)
!     The NAG name equivalent of dormqr is f08agf
      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)
      y(1:m+p-n) = zero

      If (n>m) Then

!       Copy c2 into Y2
        y(m+p-n+1:p) = d(m+1:n)

!       Solve T22*w2 = c2 for w2, storing result in Y2
!       The NAG name equivalent of dtrtrs is f07tef
        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>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 100
        End If

!       Compute estimate of the square root of the residual sum of squares
!       norm(y) = norm(w2)

!       The NAG name equivalent of dnrm2 is f06ejf
        rnorm = dnrm2(n-m,y(m+p-n+1),1)

!       Form c1 - T12*w2 in D
!       The NAG name equivalent of dgemv is f06paf
        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
!     The NAG name equivalent of dtrtrs is f07tef
      Call dtrtrs('Upper','No transpose','Non-unit',m,1,a,lda,d,m,info)

      If (info>0) Then
        Write (nout,*) 'The upper triangular factor, R, of A is singular, '
        Write (nout,*) 'the least squares solution could not be computed'
      Else

!       Compute y = (Z**T)*w
!       The NAG name equivalent of dormrq is f08ckf
        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(1:m)

!       Print residual vector y

        Write (nout,*)
        Write (nout,*) 'Residual vector'
        Write (nout,99998) y(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
      End If
100   Continue

99999 Format (1X,7F11.4)
99998 Format (3X,1P,7E11.2)
    End Program f08zefe
```

10.2 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
```

10.3 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
```