# NAG Library Routine Document

# F08ABF (DGEQRT)

**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

F08ABF (DGEQRT) recursively computes, with explicit blocking, the $QR$ factorization of a real $m$ by $n$ matrix.

## 2 Specification

```
SUBROUTINE F08ABF (M, N, NB, A, LDA, T, LDT, WORK, INFO)
INTEGER              M, N, NB, LDA, LDT, INFO
REAL (KIND=nag_wp) A(LDA,*), T(LDT,*), WORK(NB*N)
```

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

## 3 Description

F08ABF (DGEQRT) forms the $QR$ factorization of an arbitrary rectangular real $m$ by $n$ matrix. No pivoting is performed.

It differs from F08AEF (DGEQRF) in that it: requires an explicit block size; stores reflector factors that are upper triangular matrices of the chosen block size (rather than scalars); and recursively computes the $QR$ factorization based on the algorithm of Elmroth and Gustavson (2000).

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

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

where $R$ is an $n$ by $n$ upper triangular matrix and $Q$ is an $m$ by $m$ orthogonal matrix. It is sometimes more convenient to write the factorization as

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

which reduces to

$$A = Q_1R,$$

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

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

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

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 9).

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

## 4 References

Elmroth E and Gustavson F (2000) Applying Recursion to Serial and Parallel $QR$ Factorization Leads to Better Performance *IBM Journal of Research and Development. (Volume 44)* **4** 605–624

Golub G H and Van Loan C F (2012) *Matrix Computations* (4th 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: NB – INTEGER *Input*

*On entry*: the explicitly chosen block size to be used in computing the $QR$ factorization. See Section 9 for details.

*Constraints*:

$\mathrm{NB} \geq 1$;
if $\min(\mathrm{M}, \mathrm{N}) > 0$, $\mathrm{NB} \leq \min(\mathrm{M}, \mathrm{N})$.

4: $\mathrm{A}(\mathrm{LDA}, *)$ – REAL (KIND=nag_wp) 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 orthogonal 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 orthogonal matrix $Q$ and the remaining elements are overwritten by the corresponding elements of the $m$ by $n$ upper trapezoidal matrix $R$.

5: LDA – INTEGER *Input*

*On entry*: the first dimension of the array A as declared in the (sub)program from which F08ABF (DGEQRT) is called.

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

6: $\mathrm{T}(\mathrm{LDT}, *)$ – REAL (KIND=nag_wp) array *Output*

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

*On exit*: further details of the orthogonal matrix $Q$. The number of blocks is $b = \left\lceil \frac{k}{\mathrm{NB}} \right\rceil$, where $k = \min(m, n)$ and each block is of order NB except for the last block, which is of order $k - (b - 1) \times \mathrm{NB}$. For each of the blocks, an upper triangular block reflector factor is computed: $\boldsymbol{T}_1, \boldsymbol{T}_2, \ldots, \boldsymbol{T}_b$. These are stored in the NB by $n$ matrix $T$ as $\boldsymbol{T} = [\boldsymbol{T}_1 | \boldsymbol{T}_2 | \ldots | \boldsymbol{T}_b]$.

7: LDT – INTEGER *Input*

*On entry*: the first dimension of the array T as declared in the (sub)program from which F08ABF (DGEQRT) is called.

*Constraint*: LDT $\geq$ NB.

8: WORK(NB $\times$ N) – REAL (KIND=nag_wp) array *Workspace*

9: INFO – INTEGER *Output*

*On exit*: INFO $= 0$ unless the routine detects an error (see Section 6).

# 6 Error Indicators and Warnings

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

# 8 Parallelism and Performance

F08ABF (DGEQRT) is not threaded by NAG in any implementation.

F08ABF (DGEQRT) 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 total number of floating-point operations is approximately $\frac{2}{3}n^2(3m - n)$ if $m \geq n$ or $\frac{2}{3}m^2(3n - m)$ if $m < n$.

To apply $Q$ to an arbitrary real rectangular matrix $C$, F08ABF (DGEQRT) may be followed by a call to F08ACF (DGEMQRT). For example,

```
CALL DGEMQRT('Left','Transpose',M,P,MIN(M,N),NB,A,LDA,T,LDT,C,LDC, &
             WORK,INFO)
```

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

To form the orthogonal matrix $Q$ explicitly, simply initialize the $m$ by $m$ matrix $C$ to the identity matrix and form $C = QC$ using F08ACF (DGEMQRT) as above.

The block size, NB, used by F08ABF (DGEQRT) is supplied explicitly through the interface. For moderate and large sizes of matrix, the block size can have a marked effect on the efficiency of the algorithm with the optimal value being dependent on problem size and platform. A value of NB $= 64 \ll \min(m, n)$ is likely to achieve good efficiency and it is unlikely that an optimal value would exceed 340.

To compute a $QR$ factorization with column pivoting, use F08BBF (DTPQRT) or F08BEF (DGEQPF).

The complex analogue of this routine is F08APF (ZGEQRT).

## 10 Example

This example solves the linear least squares problems

$$\text{minimize}\,\|Ax_i - b_i\|_2, \quad i = 1, 2$$

where $b_1$ and $b_2$ are the columns of the matrix $B$,

$$A = \begin{pmatrix} -0.57 & -1.28 & -0.39 & 0.25 \\ -1.93 & 1.08 & -0.31 & -2.14 \\ 2.30 & 0.24 & 0.40 & -0.35 \\ -1.93 & 0.64 & -0.66 & 0.08 \\ 0.15 & 0.30 & 0.15 & -2.13 \\ -0.02 & 1.03 & -1.43 & 0.50 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} -2.67 & 0.41 \\ -0.55 & -3.10 \\ 3.34 & -4.01 \\ -0.77 & 2.76 \\ 0.48 & -6.17 \\ 4.10 & 0.21 \end{pmatrix}.$$

### 10.1 Program Text

```
    Program f08abfe

!     F08ABF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: dgemqrt, dgeqrt, dnrm2, dtrtrs, nag_wp, x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nbmax = 64, nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, ifail, info, j, lda, ldb, ldt,    &
                                          lwork, m, n, nb, nrhs
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:,:), b(:,:), rnorm(:), t(:,:),    &
                                          work(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: max, min
!     .. Executable Statements ..
      Write (nout,*) 'F08ABF Example Program Results'
      Write (nout,*)
      Flush (nout)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) m, n, nrhs
      lda = m
      ldb = m
      nb = min(m,n,nbmax)
      ldt = nb
      lwork = nb*max(n,m)
      Allocate (a(lda,n),b(ldb,nrhs),rnorm(nrhs),t(ldt,min(m,n)),work(lwork))

!     Read A and B from data file

      Read (nin,*)(a(i,1:n),i=1,m)
      Read (nin,*)(b(i,1:nrhs),i=1,m)

!     Compute the QR factorization of A
!     The NAG name equivalent of dgeqrt is f08abf
      Call dgeqrt(m,n,nb,a,lda,t,ldt,work,info)

!     Compute C = (C1) = (Q**T)*B, storing the result in B
!                 (C2)
!     The NAG name equivalent of dgemqrt is f08acf
      Call dgemqrt('Left','Transpose',m,nrhs,n,nb,a,lda,t,ldt,b,ldb,work,info)

!     Compute least-squares solutions by backsubstitution in
```

```
!     R*X = C1
!     The NAG name equivalent of dtrtrs is f07tef
      Call dtrtrs('Upper','No transpose','Non-Unit',n,nrhs,a,lda,b,ldb,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

!       Print least-squares solutions

!       ifail: behaviour on error exit
!              =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
        ifail = 0
        Call x04caf('General',' ',n,nrhs,b,ldb,'Least-squares solution(s)', &
          ifail)

!       Compute and print estimates of the square roots of the residual
!       sums of squares

!       The NAG name equivalent of dnrm2 is f06ejf
        Do j = 1, nrhs
          rnorm(j) = dnrm2(m-n,b(n+1,j),1)
        End Do

        Write (nout,*)
        Write (nout,*) 'Square root(s) of the residual sum(s) of squares'
        Write (nout,99999) rnorm(1:nrhs)
      End If

99999 Format (5X,1P,7E11.2)
    End Program f08abfe
```

## 10.2 Program Data

```
F08ABF Example Program Data

  6      4      2           : m, n and nrhs

 -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 : matrix A

 -2.67   0.41
 -0.55  -3.10
  3.34  -4.01
 -0.77   2.76
  0.48  -6.17
  4.10   0.21               : matrix B
```

## 10.3 Program Results

```
 F08ABF Example Program Results

 Least-squares solution(s)
             1          2
 1      1.5339    -1.5753
 2      1.8707     0.5559
 3     -1.5241     1.3119
 4      0.0392     2.9585

 Square root(s) of the residual sum(s) of squares
        2.22E-02   1.38E-02
```