NAG Library Routine Document

F08AFF (DORGQR)

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

F08AFF (DORGQR) generates all or part of the real orthogonal matrix Q from a QR factorization computed by F08AEF (DGEQRF), F08BEF (DGEQPF) or F08BFF (DGEQP3).

2 Specification

```
SUBROUTINE F08AFF (M, N, K, A, LDA, TAU, WORK, LWORK, INFO)
INTEGER            M, N, K, LDA, LWORK, INFO
REAL (KIND=nag_wp) A(LDA,*), TAU(*), WORK(max(1,LWORK))
```

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

3 Description

F08AFF (DORGQR) is intended to be used after a call to F08AEF (DGEQRF), F08BEF (DGEQPF) or F08BFF (DGEQP3). which perform a QR factorization of a real matrix A. The orthogonal matrix Q is represented as a product of elementary reflectors.

This routine may be used to generate Q explicitly as a square matrix, or to form only its leading columns.

Usually Q is determined from the QR factorization of an m by p matrix A with $m \geq p$. The whole of Q may be computed by:

```
CALL DORGQR(M,M,P,A,LDA,TAU,WORK,LWORK,INFO)
```

(note that the array A must have at least m columns) or its leading p columns by:

```
CALL DORGQR(M,P,P,A,LDA,TAU,WORK,LWORK,INFO)
```

The columns of Q returned by the last call form an orthonormal basis for the space spanned by the columns of A; thus F08AEF (DGEQRF) followed by F08AFF (DORGQR) can be used to orthogonalize the columns of A.

The information returned by the QR factorization routines also yields the QR factorization of the leading k columns of A, where $k < p$. The orthogonal matrix arising from this factorization can be computed by:

```
CALL DORGQR(M,M,K,A,LDA,TAU,WORK,LWORK,INFO)
```

or its leading k columns by:

```
CALL DORGQR(M,K,K,A,LDA,TAU,WORK,LWORK,INFO)
```

4 References

Golub G H and Van Loan C F (1996) *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

5 Arguments

1: M – INTEGER *Input*

On entry: m, the order of the orthogonal matrix Q.

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

2: N – INTEGER *Input*

On entry: n, the number of columns of the matrix Q.

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

3: K – INTEGER *Input*

On entry: k, the number of elementary reflectors whose product defines the matrix Q.

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

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

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

On entry: details of the vectors which define the elementary reflectors, as returned by F08AEF (DGEQRF), F08BEF (DGEQPF) or F08BFF (DGEQP3).

On exit: the m by n matrix Q.

5: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08AFF (DORGQR) is called.

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

6: TAU($*$) – REAL (KIND=nag_wp) array *Input*

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

On entry: further details of the elementary reflectors, as returned by F08AEF (DGEQRF), F08BEF (DGEQPF) or F08BFF (DGEQP3).

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

On exit: if $\mathrm{INFO} = 0$, $\mathrm{WORK}(1)$ contains the minimum value of LWORK required for optimal performance.

8: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08AFF (DORGQR) is called.

If $\mathrm{LWORK} = -1$, 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, $\mathrm{LWORK} \geq \mathrm{N} \times nb$, where nb is the optimal ***block size***.

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

9: INFO – INTEGER *Output*

On exit: $\mathrm{INFO} = 0$ 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 matrix Q differs from an exactly orthogonal matrix by a matrix E such that

$$\|E\|_2 = O(\epsilon),$$

where ϵ is the ***machine precision***.

8 Parallelism and Performance

F08AFF (DORGQR) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F08AFF (DORGQR) 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 $4mnk - 2(m+n)k^2 + \frac{4}{3}k^3$; when $n = k$, the number is approximately $\frac{2}{3}n^2(3m-n)$.

The complex analogue of this routine is F08ATF (ZUNGQR).

10 Example

This example forms the leading 4 columns of the orthogonal matrix Q from the QR factorization of the matrix A, where

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

The columns of Q form an orthonormal basis for the space spanned by the columns of A.

10.1 Program Text

```
    Program f08affe

!     F08AFF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: dgeqrf, dorgqr, nag_wp, x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
```

```
!     .. Local Scalars ..
      Integer                          :: i, ifail, info, lda, lwork, m, n
      Character (30)                   :: title
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:,:), tau(:), work(:)
!     .. Executable Statements ..
      Write (nout,*) 'F08AFF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) m, n
      lda = m
      lwork = 64*n
      Allocate (a(lda,n),tau(n),work(lwork))

!     Read A from data file

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

!     Compute the QR factorization of A
!     The NAG name equivalent of dgeqrf is f08aef
      Call dgeqrf(m,n,a,lda,tau,work,lwork,info)

!     Form the leading N columns of Q explicitly
!     The NAG name equivalent of dorgqr is f08aff
      Call dorgqr(m,n,n,a,lda,tau,work,lwork,info)

!     Print the leading N columns of Q only

      Write (nout,*)
      Write (title,99999) n
      Flush (nout)

!     ifail: behaviour on error exit
!             =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
      ifail = 0
      Call x04caf('General',' ',m,n,a,lda,title,ifail)

99999 Format ('The leading ',I2,' columns of Q')
    End Program f08affe
```

10.2 Program Data

```
F08AFF Example Program Data
  6  4                          :Values of M and N
 -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
```

10.3 Program Results

```
 F08AFF Example Program Results

 The leading  4 columns of Q
          1       2       3       4
 1  -0.1576  0.6744 -0.4571  0.4489
 2  -0.5335 -0.3861  0.2583  0.3898
 3   0.6358 -0.2928  0.0165  0.1930
 4  -0.5335 -0.1692 -0.0834 -0.2350
 5   0.0415 -0.1593  0.1475  0.7436
 6  -0.0055 -0.5064 -0.8339  0.0335
```