NAG Library Routine Document

F08CFF (DORGQL)

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

F08CFF (DORGQL) generates all or part of the real m by m orthogonal matrix Q from a QL factorization computed by F08CEF (DGEQLF).

2 Specification

```
SUBROUTINE F08CFF (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 ***dorgql***.

3 Description

F08CFF (DORGQL) is intended to be used after a call to F08CEF (DGEQLF), which performs a QL 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 trailing columns.

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

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

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

```
CALL DORGQL(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 F08CEF (DGEQLF) followed by F08CFF (DORGQL) can be used to orthogonalize the columns of A.

The information returned by F08CEF (DGEQLF) also yields the QL factorization of the trailing k columns of A, where $k < p$. The orthogonal matrix arising from this factorization can be computed by:

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

or its trailing k columns by:

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

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

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 number of rows of the matrix Q.

Constraint: $M \geq 0$.

2: N – INTEGER *Input*

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

Constraint: $M \geq N \geq 0$.

3: K – INTEGER *Input*

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

Constraint: $N \geq K \geq 0$.

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

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

On entry: details of the vectors which define the elementary reflectors, as returned by F08CEF (DGEQLF).

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 F08CFF (DORGQL) is called.

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

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

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

On entry: further details of the elementary reflectors, as returned by F08CEF (DGEQLF).

7: WORK(max(1, LWORK)) – REAL (KIND=nag_wp) array *Workspace*

On exit: if INFO $= 0$, 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 F08CFF (DORGQL) 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, $LWORK \geq N \times nb$, where nb is the optimal ***block size***.

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

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 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

F08CFF (DORGQL) 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 F08CTF (ZUNGQL).

10 Example

This example generates the first four columns of the matrix Q of the QL factorization of A as returned by F08CEF (DGEQLF), where

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

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 f08cffe

!     F08CFF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: dgeqlf, dorgql, nag_wp, x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nb = 64, 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,*) 'F08CFF Example Program Results'
      Write (nout,*)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) m, n
      lda = m
      lwork = nb*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 QL factorization of A

!     The NAG name equivalent of dgeqlf is f08cef
      Call dgeqlf(m,n,a,lda,tau,work,lwork,info)

!     Form the leading N columns of Q explicitly

!     The NAG name equivalent of dorgql is f08cff
      Call dorgql(m,n,n,a,lda,tau,work,lwork,info)

!     Form the heading for X04CAF

      Write (title,99999) n
      Flush (nout)

!     Print the leading N columns of Q

!     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 ',I4,' columns of Q')
    End Program f08cffe
```

10.2 Program Data

```
F08CFF 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

```
 F08CFF Example Program Results

 The leading    4 columns of Q
          1       2       3       4
 1  -0.0833  0.9100 -0.2202 -0.0809
 2   0.2972 -0.1080 -0.2706  0.6922
 3  -0.6404 -0.2351  0.2220  0.1132
 4   0.4461 -0.1620 -0.3866 -0.0259
 5  -0.2938  0.2022  0.0015  0.6890
 6  -0.4575 -0.1946 -0.8243 -0.1617
```