NAG Library Routine Document

F08AJF (DORGLQ)

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

F08AJF (DORGLQ) generates all or part of the real orthogonal matrix Q from an LQ factorization computed by F08AHF (DGELQF).

2 Specification

```
SUBROUTINE F08AJF (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 ***dorglq***.

3 Description

F08AJF (DORGLQ) is intended to be used after a call to F08AHF (DGELQF), which performs an LQ 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 rows.

Usually Q is determined from the LQ factorization of a p by n matrix A with $p \leq n$. The whole of Q may be computed by:

```
CALL DORGLQ(N,N,P,A,LDA,TAU,WORK,LWORK,INFO)
```

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

```
CALL DORGLQ(P,N,P,A,LDA,TAU,WORK,LWORK,INFO)
```

The rows of Q returned by the last call form an orthonormal basis for the space spanned by the rows of A; thus F08AHF (DGELQF) followed by F08AJF (DORGLQ) can be used to orthogonalize the rows of A.

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

```
CALL DORGLQ(N,N,K,A,LDA,TAU,WORK,LWORK,INFO)
```

or its leading k rows by:

```
CALL DORGLQ(K,N,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 Parameters

1: M – INTEGER *Input*

On entry: m, the number of rows of the matrix Q.

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

2: N – INTEGER *Input*

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

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

3: K – INTEGER *Input*

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

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

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: details of the vectors which define the elementary reflectors, as returned by F08AHF (DGELQF).

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 F08AJF (DORGLQ) is called.

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

6: $\mathrm{TAU}(*)$ – 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 F08AHF (DGELQF).

7: $\mathrm{WORK}(\max(1, \mathrm{LWORK}))$ – 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 F08AJF (DORGLQ) 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, $\mathrm{LWORK} \geq \mathrm{M} \times nb$, where nb is the optimal ***block size***.

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

$\mathrm{INFO} < 0$

If $\mathrm{INFO} = -i$, 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

F08AJF (DORGLQ) is not threaded by NAG in any implementation.

F08AJF (DORGLQ) 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 $m = k$, the number is approximately $\frac{2}{3}m^2(3n - m)$.

The complex analogue of this routine is F08AWF (ZUNGLQ).

10 Example

This example forms the leading 4 rows of the orthogonal matrix Q from the LQ factorization of the matrix A, where

$$A = \begin{pmatrix} -5.42 & 3.28 & -3.68 & 0.27 & 2.06 & 0.46 \\ -1.65 & -3.40 & -3.20 & -1.03 & -4.06 & -0.01 \\ -0.37 & 2.35 & 1.90 & 4.31 & -1.76 & 1.13 \\ -3.15 & -0.11 & 1.99 & -2.70 & 0.26 & 4.50 \end{pmatrix}.$$

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

10.1 Program Text

```
    Program f08ajfe

!     F08AJF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: dgelqf, dorglq, 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,*) 'F08AJF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) m, n
      lda = m
      lwork = 64*m
      Allocate (a(lda,n),tau(n),work(lwork))

!     Read A from data file

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

!     Compute the LQ factorization of A
!     The NAG name equivalent of dgelqf is f08ahf
      Call dgelqf(m,n,a,lda,tau,work,lwork,info)

!     Form the leading M rows of Q explicitly
!     The NAG name equivalent of dorglq is f08ajf
      Call dorglq(m,n,m,a,lda,tau,work,lwork,info)

!     Print the leading M rows of Q only

      Write (nout,*)
      Write (title,99999) m
      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,' rows of Q')
    End Program f08ajfe
```

10.2 Program Data

```
F08AJF Example Program Data
  4  6                                      :Values of M and N
 -5.42   3.28  -3.68   0.27   2.06   0.46
 -1.65  -3.40  -3.20  -1.03  -4.06  -0.01
 -0.37   2.35   1.90   4.31  -1.76   1.13
 -3.15  -0.11   1.99  -2.70   0.26   4.50   :End of matrix A
```

10.3 Program Results

```
 F08AJF Example Program Results

 The leading  4 rows of Q
          1       2       3       4       5       6
 1  -0.7104  0.4299 -0.4824  0.0354  0.2700  0.0603
 2  -0.2412 -0.5323 -0.4845 -0.1595 -0.6311 -0.0027
 3   0.1287 -0.2619 -0.2108 -0.7447  0.5227 -0.2063
 4  -0.3403 -0.0921  0.4546 -0.3869 -0.0465  0.7191
```