

## NAG Library Routine Document

### F08AHF (DGELQF)

**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

F08AHF (DGELQF) computes the  $LQ$  factorization of a real  $m$  by  $n$  matrix.

#### 2 Specification

```
SUBROUTINE F08AHF (M, N, A, LDA, TAU, WORK, LWORK, INFO)
  INTEGER          M, N, LDA, LWORK, INFO
  REAL (KIND=nag_wp) A(LDA,*), TAU(*), WORK(max(1,LWORK))
```

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

#### 3 Description

F08AHF (DGELQF) forms the  $LQ$  factorization of an arbitrary rectangular real  $m$  by  $n$  matrix. No pivoting is performed.

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

$$A = (L \ 0)Q$$

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

$$A = (L \ 0) \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix}$$

which reduces to

$$A = LQ_1,$$

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

If  $m > n$ ,  $L$  is trapezoidal, and the factorization can be written

$$A = \begin{pmatrix} L_1 \\ L_2 \end{pmatrix} Q$$

where  $L_1$  is lower triangular and  $L_2$  is rectangular.

The  $LQ$  factorization of  $A$  is essentially the same as the  $QR$  factorization of  $A^T$ , since

$$A = (L \ 0)Q \Leftrightarrow A^T = Q^T \begin{pmatrix} L^T \\ 0 \end{pmatrix}.$$

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  $k < m$ , the information returned in the first  $k$  rows of the array  $A$  represents an  $LQ$  factorization of the first  $k$  rows of the original matrix  $A$ .

#### 4 References

None.

## 5 Arguments

- 1: M – INTEGER *Input*  
*On entry:*  $m$ , the number of rows of the matrix  $A$ .  
*Constraint:*  $M \geq 0$ .
- 2: N – INTEGER *Input*  
*On entry:*  $n$ , the number of columns of the matrix  $A$ .  
*Constraint:*  $N \geq 0$ .
- 3: 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:* the  $m$  by  $n$  matrix  $A$ .  
*On exit:* if  $m \leq n$ , the elements above the diagonal are overwritten by details of the orthogonal matrix  $Q$  and the lower triangle is overwritten by the corresponding elements of the  $m$  by  $m$  lower triangular matrix  $L$ .  
 If  $m > n$ , the strictly upper 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  $n$  lower trapezoidal matrix  $L$ .
- 4: LDA – INTEGER *Input*  
*On entry:* the first dimension of the array  $A$  as declared in the (sub)program from which F08AHF (DGELQF) is called.  
*Constraint:*  $LDA \geq \max(1, M)$ .
- 5: TAU(\*) – REAL (KIND=nag\_wp) array *Output*  
**Note:** the dimension of the array  $TAU$  must be at least  $\max(1, \min(M, N))$ .  
*On exit:* further details of the orthogonal matrix  $Q$ .
- 6: 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.
- 7: LWORK – INTEGER *Input*  
*On entry:* the dimension of the array  $WORK$  as declared in the (sub)program from which F08AHF (DGELQF) 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,  $LWORK \geq M \times nb$ , where  $nb$  is the optimal **block size**.  
*Constraint:*  $LWORK \geq \max(1, M)$  or  $LWORK = -1$ .
- 8: 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

F08AHF (DGELQF) 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  $\frac{2}{3}m^2(3n - m)$  if  $m \leq n$  or  $\frac{2}{3}n^2(3m - n)$  if  $m > n$ .

To form the orthogonal matrix  $Q$  F08AHF (DGELQF) may be followed by a call to F08AJF (DORGLQ):

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

but note that the first dimension of the array A, specified by the argument LDA, must be at least N, which may be larger than was required by F08AHF (DGELQF).

When  $m \leq n$ , it is often only the first  $m$  rows of  $Q$  that are required, and they may be formed by the call:

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

To apply  $Q$  to an arbitrary real rectangular matrix  $C$ , F08AHF (DGELQF) may be followed by a call to F08AKF (DORMLQ). For example,

```
CALL DORMLQ('Left','Transpose',M,P,MIN(M,N),A,LDA,TAU,C,LDC, &
           WORK,LWORK,INFO)
```

forms the matrix product  $C = Q^T C$ , where  $C$  is  $m$  by  $p$ .

The complex analogue of this routine is F08AVF (ZGELQF).

## 10 Example

This example finds the minimum norm solutions of the under-determined systems of linear equations

$$Ax_1 = b_1 \quad \text{and} \quad Ax_2 = b_2$$

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

$$A = \begin{pmatrix} -5.42 & 3.28 & -3.68 & 0.27 & 2.06 & 0.46 \\ -1.65 & -3.40 & -3.20 & -1.03 & -4.06 & -0.01 \\ -0.37 & 2.35 & 1.90 & 4.31 & -1.76 & 1.13 \\ -3.15 & -0.11 & 1.99 & -2.70 & 0.26 & 4.50 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} -2.87 & -5.23 \\ 1.63 & 0.29 \\ -3.52 & 4.76 \\ 0.45 & -8.41 \end{pmatrix}.$$

## 10.1 Program Text

```

Program f08ahfe

!      F08AHF Example Program Text

!      Mark 26 Release. NAG Copyright 2016.

!      .. Use Statements ..
Use nag_library, Only: dgelqf, dormlq, dtrsm, nag_wp, x04caf
!      .. 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                 :: nin = 5, nout = 6
!      .. Local Scalars ..
Integer                             :: i, ifail, info, lda, ldb, lwork, m, &
                                     n, nrhs

!      .. Local Arrays ..
Real (Kind=nag_wp), Allocatable    :: a(:,,:), b(:,,:), tau(:), work(:)
!      .. Executable Statements ..
Write (nout,*) 'F08AHF Example Program Results'
Skip heading in data file
Read (nin,*)
Read (nin,*) m, n, nrhs
lda = m
ldb = n
lwork = 64*n
Allocate (a(lda,n),b(ldb,nrhs),tau(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 LQ factorization of A
!      The NAG name equivalent of dgelqf is f08ahf
Call dgelqf(m,n,a,lda,tau,work,lwork,info)

!      Solve L*Y = B, storing the result in B
!      The NAG name equivalent of dtrsm is f06yjf
Call dtrsm('Left','Lower','No transpose','Non-Unit',m,nrhs,one,a,lda,b, &
          ldb)

!      Set rows (M+1) to N of B to zero

If (m<n) Then
  b(m+1:n,1:nrhs) = zero
End If

!      Compute minimum-norm solution X = (Q**T)*B in B
!      The NAG name equivalent of dormlq is f08akf
Call dormlq('Left','Transpose',n,nrhs,m,a,lda,tau,b,ldb,work,lwork,info)

!      Print minimum-norm solution(s)

Write (nout,*)
Flush (nout)

!      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,'Minimum-norm solution(s)',ifail)

End Program f08ahfe

```

**10.2 Program Data**

```

F08AHF Example Program Data
  4  6  2                               :Values of M, N and NRHS
-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
-2.87 -5.23
  1.63  0.29
-3.52  4.76
  0.45 -8.41                               :End of matrix B

```

**10.3 Program Results**

F08AHF Example Program Results

```

Minimum-norm solution(s)
           1           2
1      0.2371      0.7383
2     -0.4575      0.0158
3     -0.0085     -0.0161
4     -0.5192      1.0768
5      0.0239     -0.6436
6     -0.0543     -0.6613

```

---