

NAG Library Routine Document

F08CEF (DGEQLF)

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

F08CEF (DGEQLF) computes a QL factorization of a real m by n matrix A .

2 Specification

```
SUBROUTINE F08CEF (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 *dgeqlf*.

3 Description

F08CEF (DGEQLF) forms the QL factorization of an arbitrary rectangular real m by n matrix.

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

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

where L is an n by n lower triangular matrix and Q is an m by m orthogonal matrix. If $m < n$ the factorization is given by

$$A = QL,$$

where L is an m by n lower trapezoidal matrix and Q is again an m by m orthogonal matrix. In the case where $m > n$ the factorization can be expressed as

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

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

The matrix Q is not formed explicitly but is represented as a product of $\min(m, n)$ elementary reflectors (see Section 3.3.6 in 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 in the last k columns of the array A represents a QL factorization of the last k columns of the original matrix A .

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 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 $m \geq n$, the lower triangle of the subarray $A(m - n + 1 : m, 1 : n)$ contains the n by n lower triangular matrix L .
 If $m \leq n$, the elements on and below the $(n - m)$ th superdiagonal contain the m by n lower trapezoidal matrix L . The remaining elements, with the array TAU, represent the orthogonal matrix Q as a product of elementary reflectors (see Section 3.3.6 in the F08 Chapter Introduction).
- 4: LDA – INTEGER *Input*
On entry: the first dimension of the array A as declared in the (sub)program from which F08CEF (DGEQLF) 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: the scalar factors of the elementary reflectors (see Section 9).
- 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 F08CEF (DGEQLF) 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: $LWORK \geq \max(1, N)$.
- 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 ϵ is the *machine precision*.

8 Parallelism and Performance

F08CEF (DGEQLF) 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 form the orthogonal matrix Q F08CEF (DGEQLF) may be followed by a call to F08CFF (DORGQL):

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

but note that the second dimension of the array A must be at least M , which may be larger than was required by F08CEF (DGEQLF).

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

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

To apply Q to an arbitrary real rectangular matrix C , F08CEF (DGEQLF) may be followed by a call to F08CGF (DORMQL). For example,

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

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

The complex analogue of this routine is F08CSF (ZGEQLF).

10 Example

This example solves the linear least squares problems

$$\min_x \|b_j - Ax_j\|_2, \quad j = 1, 2$$

for x_1 and x_2 , where b_j is the j th column of the matrix B ,

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

The solution is obtained by first obtaining a *QL* factorization of the matrix *A*.

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 f08cefe

```

!      F08CEF Example Program Text
!
!      Mark 26 Release. NAG Copyright 2016.
!
!      .. Use Statements ..
      Use nag_library, Only: dgeqlf, dnrnm2, dormql, dtrtrs, nag_wp, x04caf
!      .. Implicit None Statement ..
      Implicit None
!      .. Parameters ..
      Integer, Parameter          :: nb = 64, nin = 5, nout = 6
!      .. Local Scalars ..
      Integer                     :: i, ifail, info, j, lda, ldb, lwork, &
                                m, n, nrhs
!      .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable :: a(:,,:), b(:,,:), rnorm(:), tau(:), &
                                work(:)
!      .. Executable Statements ..
      Write (nout,*) 'F08CEF Example Program Results'
      Write (nout,*)
      Flush (nout)
!      Skip heading in data file
      Read (nin,*)
      Read (nin,*) m, n, nrhs
      lda = m
      ldb = m
      lwork = nb*n
      Allocate (a(lda,n),b(ldb,nrhs),rnorm(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 QL factorization of A
!      The NAG name equivalent of dgeqlf is f08cef
      Call dgeqlf(m,n,a,lda,tau,work,lwork,info)
!
!      Compute C = (C1) = (Q**T)*B, storing the result in B
!      (C2)
!      The NAG name equivalent of dormql is f08cgf
      Call dormql('Left','Transpose',m,nrhs,n,a,lda,tau,b,ldb,work,lwork,info)
!
!      Compute least squares solutions by back-substitution in
!      L*X = C2
!      The NAG name equivalent of dtrtrs is f07tef
      Call dtrtrs('Lower','No transpose','Non-Unit',n,nrhs,a(m-n+1,1),lda, &
                b(m-n+1,1),ldb,info)
!
      If (info>0) Then
         Write (nout,*) 'The lower triangular factor, L, of A is singular, '
         Write (nout,*) 'the least squares solution could not be computed'
      Else

```

```

!      Print least squares solution(s)

!      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(m-n+1,1),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 dnorm2 is f06ejf
      Do j = 1, nrhs
        rnorm(j) = dnorm2(m-n,b(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 f08cefe

```

10.2 Program Data

F08CEF Example Program Data

```

  6      4      2      :Values of 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 :End of 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      :End of matrix B

```

10.3 Program Results

F08CEF 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

```
