# NAG Library Routine Document

# F08CSF (ZGEQLF)

**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

F08CSF (ZGEQLF) computes a $QL$ factorization of a complex $m$ by $n$ matrix $A$.

## 2 Specification

```
SUBROUTINE F08CSF (M, N, A, LDA, TAU, WORK, LWORK, INFO)
INTEGER                 M, N, LDA, LWORK, INFO
COMPLEX (KIND=nag_wp) A(LDA,*), TAU(*), WORK(max(1,LWORK))
```

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

## 3 Description

F08CSF (ZGEQLF) forms the $QL$ factorization of an arbitrary rectangular complex $m$ 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 $m$ unitary 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 $m$ unitary matrix. In the case where $m > n$ the factorization can be expressed as

$$A = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix}\begin{pmatrix} 0 \\ L \end{pmatrix} = Q_2 L,$$

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 kendgroup 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*: $\mathrm{M} \geq 0$.

2: N – INTEGER *Input*

*On entry*: $n$, the number of columns of the matrix $A$.

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

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

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

*On entry*: the $m$ by $n$ matrix $A$.

*On exit*: if $m \geq n$, the lower triangle of the subarray $\mathrm{A}(m-n+1:m, 1:n)$ 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 unitary 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 F08CSF (ZGEQLF) is called.

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

5: TAU($*$) – COMPLEX (KIND=nag_wp) array *Output*

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

*On exit*: the scalar factors of the elementary reflectors (see Section 9).

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

*On exit*: if $\mathrm{INFO} = 0$, the real part of $\mathrm{WORK}(1)$ 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 F08CSF (ZGEQLF) 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*: $\mathrm{LWORK} \geq \max(1, \mathrm{N})$.

8: 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 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

F08CSF (ZGEQLF) 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 real floating-point operations is approximately $\frac{8}{3}n^2(3m - n)$ if $m \geq n$ or $\frac{8}{3}m^2(3n - m)$ if $m < n$.

To form the unitary matrix $Q$ F08CSF (ZGEQLF) may be followed by a call to F08CTF (ZUNGQL):

```
CALL ZUNGQL(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 F08CSF (ZGEQLF).

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

To apply $Q$ to an arbitrary complex rectangular matrix $C$, F08CSF (ZGEQLF) may be followed by a call to F08CUF (ZUNMQL). For example,

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

forms $C = Q^{\mathrm{H}}C$, where $C$ is $m$ by $p$.

The real analogue of this routine is F08CEF (DGEQLF).

## 10 Example

This example solves the linear least squares problems

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

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

$$A = \begin{pmatrix} 0.96 - 0.81i & -0.03 + 0.96i & -0.91 + 2.06i & -0.05 + 0.41i \\ -0.98 + 1.98i & -1.20 + 0.19i & -0.66 + 0.42i & -0.81 + 0.56i \\ 0.62 - 0.46i & 1.01 + 0.02i & 0.63 - 0.17i & -1.11 + 0.60i \\ -0.37 + 0.38i & 0.19 - 0.54i & -0.98 - 0.36i & 0.22 - 0.20i \\ 0.83 + 0.51i & 0.20 + 0.01i & -0.17 - 0.46i & 1.47 + 1.59i \\ 1.08 - 0.28i & 0.20 - 0.12i & -0.07 + 1.23i & 0.26 + 0.26i \end{pmatrix}$$

and

$$B = \begin{pmatrix} -2.09 + 1.93i & 3.26 - 2.70i \\ 3.34 - 3.53i & -6.22 + 1.16i \\ -4.94 - 2.04i & 7.94 - 3.13i \\ 0.17 + 4.23i & 1.04 - 4.26i \\ -5.19 + 3.63i & -2.31 - 2.12i \\ 0.98 + 2.53i & -1.39 - 4.05i \end{pmatrix}.$$

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 f08csfe

!     F08CSF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: dznrm2, nag_wp, x04dbf, zgeqlf, ztrtrs, zunmql
!     .. 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 ..
      Complex (Kind=nag_wp), Allocatable :: a(:,:), b(:,:), tau(:), work(:)
      Real (Kind=nag_wp), Allocatable   :: rnorm(:)
      Character (1)                     :: clabs(1), rlabs(1)
!     .. Executable Statements ..
      Write (nout,*) 'F08CSF 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),tau(n),work(lwork),rnorm(nrhs))

!     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 zgeqlf is f08csf
      Call zgeqlf(m,n,a,lda,tau,work,lwork,info)

!     Compute C = (C1) = (Q**H)*B, storing the result in B
!                 (C2)
!     The NAG name equivalent of zunmql is f08cuf
      Call zunmql('Left','Conjugate 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 ztrtrs is f07tsf
      Call ztrtrs('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 x04dbf('General',' ',n,nrhs,b(m-n+1,1),ldb,'Bracketed','F7.4',    &
          'Least squares solution(s)','Integer',rlabs,'Integer',clabs,80,0,    &
          ifail)

!       Compute and print estimates of the square roots of the residual
!       sums of squares
!       The NAG name equivalent of dznrm2 is f06jjf
        Do j = 1, nrhs
          rnorm(j) = dznrm2(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 (3X,1P,7E11.2)
    End Program f08csfe
```

## 10.2 Program Data

```
F08CSF Example Program Data
   6            4           2                 :Values of M, N and NRHS

 ( 0.96,-0.81) (-0.03, 0.96) (-0.91, 2.06) (-0.05, 0.41)
 (-0.98, 1.98) (-1.20, 0.19) (-0.66, 0.42) (-0.81, 0.56)
 ( 0.62,-0.46) ( 1.01, 0.02) ( 0.63,-0.17) (-1.11, 0.60)
 (-0.37, 0.38) ( 0.19,-0.54) (-0.98,-0.36) ( 0.22,-0.20)
 ( 0.83, 0.51) ( 0.20, 0.01) (-0.17,-0.46) ( 1.47, 1.59)
 ( 1.08,-0.28) ( 0.20,-0.12) (-0.07, 1.23) ( 0.26, 0.26) :End of matrix A

 (-2.09, 1.93) ( 3.26,-2.70)
 ( 3.34,-3.53) (-6.22, 1.16)
 (-4.94,-2.04) ( 7.94,-3.13)
 ( 0.17, 4.23) ( 1.04,-4.26)
 (-5.19, 3.63) (-2.31,-2.12)
 ( 0.98, 2.53) (-1.39,-4.05)                              :End of matrix B
```

## 10.3 Program Results

```
 F08CSF Example Program Results

 Least squares solution(s)
                     1                 2
 1  (-0.5044,-1.2179) ( 0.7629, 1.4529)
 2  (-2.4281, 2.8574) ( 5.1570,-3.6089)
```

```
 3   ( 1.4872,-2.1955) (-2.6518, 2.1203)
 4   ( 0.4537, 2.6904) (-2.7606, 0.3318)

 Square root(s) of the residual sum(s) of squares
      6.88E-02   1.87E-01
```