NAG Fortran Library Routine Document

F08AWF (CUNGLQ/ZUNGLQ)

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

F08AWF (CUNGLQ/ZUNGLQ) generates all or part of the complex unitary matrix Q from an LQ factorization computed by F08AVF (CGELQF/ZGELQF).

2 Specification

```
SUBROUTINE F08AWF(M, N, K, A, LDA, TAU, WORK, LWORK, INFO)
ENTRY      cunglq (M, N, K, A, LDA, TAU, WORK, LWORK, INFO)
INTEGER           M, N, K, LDA, LWORK, INFO
complex           A(LDA,*), TAU(*), WORK(*)
```

The ENTRY statement enables the routine to be called by its LAPACK name.

3 Description

This routine is intended to be used after a call to F08AVF (CGELQF/ZGELQF), which performs an LQ factorization of a complex matrix A. The unitary 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 CUNGLQ (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 CUNGLQ (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 F08AVF (CGELQF/ZGELQF) followed by F08AWF (CUNGLQ/ZUNGLQ) can be used to orthogonalise 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 unitary matrix arising from this factorization can be computed by:

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

or its leading k rows by:

```
CALL CUNGLQ (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: $M \geq 0$.

2: N – INTEGER *Input*

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

Constraint: $N \geq M$.

3: K – INTEGER *Input*

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

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

4: A(LDA,*) – ***complex*** 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 F08AVF (CGELQF/ZGELQF).

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 F08AWF (CUNGLQ/ZUNGLQ) is called.

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

6: TAU(*) – ***complex*** 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 F08AVF (CGELQF/ZGELQF).

7: WORK(*) – ***complex*** array *Workspace*

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

On exit: if INFO = 0, the real part of WORK(1) contains the minimum value of LWORK required for optimum performance.

8: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08AWF (CUNGLQ/ZUNGLQ) is called, unless $LWORK = -1$, in which case a workspace query is assumed and the routine only calculates the optimal dimension of WORK (using the formula given below).

Suggested value: for optimum performance LWORK should be at least $M \times nb$, where nb is the ***blocksize***.

Constraint: $LWORK \geq \max(1, M)$ or $LWORK = -1$.

9: INFO – INTEGER *Output*

On exit: INFO = 0 unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the routine:

INFO < 0

If INFO $= -i$, the ith parameter 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 unitary matrix by a matrix E such that

$$\|E\|_2 = O(\epsilon),$$

where ϵ is the ***machine precision***.

8 Further Comments

The total number of real floating-point operations is approximately $16mnk - 8(m+n)k^2 + \frac{16}{3}k^3$; when $m = k$, the number is approximately $\frac{8}{3}m^2(3n - m)$.

The real analogue of this routine is F08AJF (SORGLQ/DORGLQ).

9 Example

To form the leading 4 rows of the unitary matrix Q from the LQ factorization of the matrix A, where

$$A = \begin{pmatrix} 0.28 - 0.36i & 0.50 - 0.86i & -0.77 - 0.48i & 1.58 + 0.66i \\ -0.50 - 1.10i & -1.21 + 0.76i & -0.32 - 0.24i & -0.27 - 1.15i \\ 0.36 - 0.51i & -0.07 + 1.33i & -0.75 + 0.47i & -0.08 + 1.01i \end{pmatrix}.$$

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

9.1 Program Text

Note: the listing of the example program presented below uses ***bold italicised*** terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     F08AWF Example Program Text
*     Mark 16 Release. NAG Copyright 1992.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          MMAX, NMAX, LDA, LWORK
      PARAMETER        (MMAX=8,NMAX=8,LDA=MMAX,LWORK=64*MMAX)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, INFO, J, M, N
      CHARACTER*30     TITLE
*     .. Local Arrays ..
      complex          A(LDA,NMAX), TAU(NMAX), WORK(LWORK)
      CHARACTER        CLABS(1), RLABS(1)
*     .. External Subroutines ..
      EXTERNAL         X04DBF, cgelqf, cunglq
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08AWF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) M, N
      IF (M.LE.MMAX .AND. N.LE.NMAX .AND. M.LE.N) THEN
*
*        Read A from data file
*
         READ (NIN,*) ((A(I,J),J=1,N),I=1,M)
*
```

```
*        Compute the LQ factorization of A
*
         CALL cgelqf(M,N,A,LDA,TAU,WORK,LWORK,INFO)
*
*        Form the leading M rows of Q explicitly
*
         CALL cunglq(M,N,M,A,LDA,TAU,WORK,LWORK,INFO)
*
*        Print the leading M rows of Q only
*
         WRITE (NOUT,*)
         WRITE (TITLE,99999) M
         IFAIL = 0
*
         CALL X04DBF('General',' ',M,N,A,LDA,'Bracketed','F7.4',TITLE,
     +               'Integer',RLABS,'Integer',CLABS,80,0,IFAIL)
*
      END IF
      STOP
*
99999 FORMAT ('The leading ',I2,' rows of Q')
      END
```

9.2 Program Data

```
F08AWF Example Program Data
  3  4                                                      :Values of M and N
 ( 0.28,-0.36) ( 0.50,-0.86) (-0.77,-0.48) ( 1.58, 0.66)
 (-0.50,-1.10) (-1.21, 0.76) (-0.32,-0.24) (-0.27,-1.15)
 ( 0.36,-0.51) (-0.07, 1.33) (-0.75, 0.47) (-0.08, 1.01)   :End of matrix A
```

9.3 Program Results

```
 F08AWF Example Program Results

 The leading  3 rows of Q
                   1                 2                 3                 4
 1  (-0.1258, 0.1618) (-0.2247, 0.3864) ( 0.3460, 0.2157) (-0.7099,-0.2966)
 2  (-0.1163,-0.6380) (-0.3240, 0.4272) (-0.1995,-0.5009) (-0.0323,-0.0162)
 3  (-0.4607, 0.1090) ( 0.2171,-0.4062) ( 0.2733,-0.6106) (-0.0994,-0.3261)
```