F08ASF (CGEQRF/ZGEQRF) – NAG Fortran Library Routine Document

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

F08ASF (CGEQRF/ZGEQRF) computes the QR factorization of a complex m by n matrix.

2 Specification

```
SUBROUTINE F08ASF(M, N, A, LDA, TAU, WORK, LWORK, INFO)
ENTRY      cgeqrf(M, N, A, LDA, TAU, WORK, LWORK, INFO)
INTEGER           M, N, LDA, LWORK, INFO
complex           A(LDA,*), TAU(*), WORK(LWORK)
```

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

3 Description

This routine forms the QR factorization of an arbitrary rectangular complex m by n matrix. No pivoting is performed.

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

$$A = Q \begin{pmatrix} R \\ 0 \end{pmatrix}$$

where R is an n by n upper triangular matrix (with real diagonal elements) and Q is an m by m unitary matrix. It is sometimes more convenient to write the factorization as

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

which reduces to

$$A = Q_1 R,$$

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

If $m < n$, R is trapezoidal, and the factorization can be written

$$A = Q(R_1 R_2),$$

where R_1 is upper triangular and R_2 is rectangular.

The matrix Q is not formed explicitly but is represented as a product of $\min(m, n)$ elementary reflectors (see the Chapter Introduction for details). Routines are provided to work with Q in this representation (see Section 8).

Note also that for any $k < n$, the information returned in the first k columns of the array A represents a QR factorization of the first k columns of the original matrix A.

4 References

[1] Golub G H and van Loan C F (1996) *Matrix Computations* Johns Hopkins University Press (3rd Edition), Baltimore

5 Parameters

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,$*$) — ***complex*** 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 elements below the diagonal are overwritten by details of the unitary matrix Q and the upper triangle is overwritten by the corresponding elements of the n by n upper triangular matrix R.

If $m < n$, the strictly lower triangular part is overwritten by details of the unitary matrix Q and the remaining elements are overwritten by the corresponding elements of the m by n upper trapezoidal matrix R.

The diagonal elements of R are real.

4: LDA — INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08ASF (CGEQRF/ZGEQRF) is called.

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

5: TAU($*$) — ***complex*** array *Output*

Note: the dimension of the array TAU must be at least max(1,min(M,N)).

On exit: further details of the unitary matrix Q.

6: WORK(LWORK) — ***complex*** array *Workspace*

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

7: LWORK — INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08ASF (CGEQRF/ZGEQRF) is called.

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

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 $= -i$, the ith parameter 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 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 this routine may be followed by a call to F08ATF (CUNGQR/ZUNGQR):

```
CALL CUNGQR (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 F08ASF.

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

To apply Q to an arbitrary complex rectangular matrix C, this routine may be followed by a call to F08AUF (CUNMQR/ZUNMQR). For example,

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

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

To compute a QR factorization with column pivoting, use F08BSF (CGEQPF/ZGEQPF).

The real analogue of this routine is F08AEF (SGEQRF/DGEQRF).

9 Example

To solve the linear least-squares problem

$$\text{minimize } \| Ax_i - b_i \|_2 \text{ for } i = 1, 2$$

where b_1 and b_2 are the columns 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

$$B = \begin{pmatrix} -1.54 + 0.76i & 3.17 - 2.09i \\ 0.12 - 1.92i & -6.53 + 4.18i \\ -9.08 - 4.31i & 7.28 + 0.73i \\ 7.49 + 3.65i & 0.91 - 3.97i \\ -5.63 - 2.12i & -5.46 - 1.64i \\ 2.37 + 8.03i & -2.84 - 5.86i \end{pmatrix}.$$

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.

```
*     F08ASF Example Program Text
*     Mark 16 Release. NAG Copyright 1992.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          MMAX, NMAX, LDA, LDB, NRHMAX, LWORK
      PARAMETER        (MMAX=8,NMAX=8,LDA=MMAX,LDB=MMAX,NRHMAX=NMAX,
     +                 LWORK=64*NMAX)
      complex          ONE
      PARAMETER        (ONE=(1.0e0,0.0e0))
*     .. Local Scalars ..
      INTEGER          I, IFAIL, INFO, J, M, N, NRHS
*     .. Local Arrays ..
      complex          A(LDA,NMAX), B(LDB,NRHMAX), TAU(NMAX),
     +                 WORK(LWORK)
      CHARACTER        CLABS(1), RLABS(1)
*     .. External Subroutines ..
      EXTERNAL         X04DBF, cgeqrf, ctrsm, cunmqr
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08ASF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) M, N, NRHS
      IF (M.LE.MMAX .AND. N.LE.NMAX .AND. M.GE.N .AND. NRHS.LE.NRHMAX)
     +    THEN
*
*        Read A and B from data file
*
         READ (NIN,*) ((A(I,J),J=1,N),I=1,M)
         READ (NIN,*) ((B(I,J),J=1,NRHS),I=1,M)
*
*        Compute the QR factorization of A
*
         CALL cgeqrf(M,N,A,LDA,TAU,WORK,LWORK,INFO)
*
*        Compute C = (Q**H)*B, storing the result in B
*
         CALL cunmqr('Left','Conjugate transpose',M,NRHS,N,A,LDA,TAU,B,
     +               LDB,WORK,LWORK,INFO)
*
*        Compute least-squares solution by backsubstitution in R*X = C
*
         CALL ctrsm('Left','Upper','No transpose','Non-Unit',N,NRHS,ONE,
     +              A,LDA,B,LDB)
*
*        Print least-squares solution(s)
*
         WRITE (NOUT,*)
         IFAIL = 0
*
         CALL X04DBF('General',' ',N,NRHS,B,LDB,'Bracketed','F7.4',
     +               'Least-squares solution(s)','Integer',RLABS,
     +               'Integer',CLABS,80,0,IFAIL)
*
```

```
      END IF
      STOP
      END
```

9.2 Program Data

```
 F08ASF 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
  (-1.54, 0.76) ( 3.17,-2.09)
  ( 0.12,-1.92) (-6.53, 4.18)
  (-9.08,-4.31) ( 7.28, 0.73)
  ( 7.49, 3.65) ( 0.91,-3.97)
  (-5.63,-2.12) (-5.46,-1.64)
  ( 2.37, 8.03) (-2.84,-5.86)                               :End of matrix B
```

9.3 Program Results

```
 F08ASF Example Program Results

 Least-squares solution(s)
                    1                 2
 1  (-0.4936,-1.1993) ( 0.7535, 1.4404)
 2  (-2.4708, 2.8373) ( 5.1726,-3.6235)
 3  ( 1.5060,-2.1830) (-2.6609, 2.1334)
 4  ( 0.4459, 2.6848) (-2.6966, 0.2711)
```