NAG Library Routine Document

F08ATF (ZUNGQR)

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

F08ATF (ZUNGQR) generates all or part of the complex unitary matrix Q from a QR factorization computed by F08ASF (ZGEQRF), F08BSF (ZGEQPF) or F08BTF (ZGEQP3).

2 Specification

```
SUBROUTINE F08ATF (M, N, K, A, LDA, TAU, WORK, LWORK, INFO)

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

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

3 Description

F08ATF (ZUNGQR) is intended to be used after a call to F08ASF (ZGEQRF), F08BSF (ZGEQPF) or F08BTF (ZGEQP3), which perform a QR 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 columns.

Usually Q is determined from the QR factorization of an m by p matrix A with $m \geq p$. The whole of Q may be computed by:

```
CALL ZUNGQR(M,M,P,A,LDA,TAU,WORK,LWORK,INFO)
```

(note that the array A must have at least m columns) or its leading p columns by:

```
CALL ZUNGQR(M,P,P,A,LDA,TAU,WORK,LWORK,INFO)
```

The columns of Q returned by the last call form an orthonormal basis for the space spanned by the columns of A; thus F08ASF (ZGEQRF) followed by F08ATF (ZUNGQR) can be used to orthogonalize the columns of A.

The information returned by the QR factorization routines also yields the QR factorization of the leading k columns of A, where $k < p$. The unitary matrix arising from this factorization can be computed by:

```
CALL ZUNGQR(M,M,K,A,LDA,TAU,WORK,LWORK,INFO)
```

or its leading k columns by:

```
CALL ZUNGQR(M,K,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 order of the unitary matrix Q.

Constraint: $\text{M} \geq 0$.

2: N – INTEGER *Input*

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

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

3: K – INTEGER *Input*

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

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

4: A(LDA,$*$) – COMPLEX (KIND=nag_wp) 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 F08ASF (ZGEQRF), F08BSF (ZGEQPF) or F08BTF (ZGEQP3).

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 F08ATF (ZUNGQR) is called.

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

6: TAU($*$) – COMPLEX (KIND=nag_wp) 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 F08ASF (ZGEQRF), F08BSF (ZGEQPF) or F08BTF (ZGEQP3).

7: WORK($\max(1, LWORK)$) – COMPLEX (KIND=nag_wp) array *Workspace*

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

8: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08ATF (ZUNGQR) 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 $\geq \max(1, N)$ 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 $= -i$, argument i 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 $n = k$, the number is approximately $\frac{8}{3}n^2(3m - n)$.

The real analogue of this routine is F08AFF (DORGQR).

9 Example

This example forms the leading 4 columns of the unitary matrix Q from the QR factorization of the matrix A, where

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

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

9.1 Program Text

```
    Program f08atfe

!     F08ATF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, x04dbf, zgeqrf, zungqr
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, ifail, info, lda, lwork, m, n
      Character (30)                   :: title
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: a(:,:), tau(:), work(:)
      Character (1)                    :: clabs(1), rlabs(1)
!     .. Executable Statements ..
      Write (nout,*) 'F08ATF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) m, n
      lda = m
      lwork = 64*n
      Allocate (a(lda,n),tau(n),work(lwork))

!     Read A from data file

      Read (nin,*)(a(i,1:n),i=1,m)

!     Compute the QR factorization of A
!     The NAG name equivalent of zgeqrf is f08asf
      Call zgeqrf(m,n,a,lda,tau,work,lwork,info)
```

```
!     Form the leading N columns of Q explicitly
!     The NAG name equivalent of zungqr is f08atf
      Call zungqr(m,n,n,a,lda,tau,work,lwork,info)

!     Print the leading N columns of Q only

      Write (nout,*)
      Write (title,99999) n
      Flush (nout)

!     ifail: behaviour on error exit
!             =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
      ifail = 0
      Call x04dbf('General',' ',m,n,a,lda,'Bracketed','F7.4',title,'Integer', &
        rlabs,'Integer',clabs,80,0,ifail)

99999 Format ('The leading ',I2,' columns of Q')
    End Program f08atfe
```

9.2 Program Data

```
F08ATF Example Program Data
  6  4                                                    :Values of M and N
 ( 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
```

9.3 Program Results

```
 F08ATF Example Program Results

 The leading  4 columns of Q
                    1                  2                  3                  4
 1  (-0.3110, 0.2624) (-0.3175, 0.4835) ( 0.4966,-0.2997) (-0.0072,-0.3718)
 2  ( 0.3175,-0.6414) (-0.2062, 0.1577) (-0.0793,-0.3094) (-0.0282,-0.1491)
 3  (-0.2008, 0.1490) ( 0.4892,-0.0900) ( 0.0357,-0.0219) ( 0.5625,-0.0710)
 4  ( 0.1199,-0.1231) ( 0.2566,-0.3055) ( 0.4489,-0.2141) (-0.1651, 0.1800)
 5  (-0.2689,-0.1652) ( 0.1697,-0.2491) (-0.0496, 0.1158) (-0.4885,-0.4540)
 6  (-0.3499, 0.0907) (-0.0491,-0.3133) (-0.1256,-0.5300) ( 0.1039, 0.0450)
```