NAG Fortran Library Routine Document

F08AEF (SGEQRF/DGEQRF)

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

F08AEF (SGEQRF/DGEQRF) computes the QR factorization of a real m by n matrix.

2 Specification

```
SUBROUTINE F08AEF(M, N, A, LDA, TAU, WORK, LWORK, INFO)
ENTRY      sgeqrf (M, N, A, LDA, TAU, WORK, LWORK, INFO)
INTEGER             M, N, LDA, LWORK, INFO
real                A(LDA,*), TAU(*), WORK(*)
```

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 real 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 and Q is an m by m orthogonal matrix. It is sometimes more convenient to write the factorization as

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

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

$$A = Q\begin{pmatrix} R_1 & R_2 \end{pmatrix},$$

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 F08 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

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 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*** 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 orthogonal 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 orthogonal matrix Q and the remaining elements are overwritten by the corresponding elements of the m by n upper trapezoidal matrix R.

4: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08AEF (SGEQRF/DGEQRF) is called.

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

5: TAU($*$) – ***real*** array *Output*

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

On exit: further details of the orthogonal matrix Q.

6: WORK($*$) – ***real*** array *Workspace*

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

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 subprogram from which F08AEF (SGEQRF/DGEQRF) 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 $N \times nb$, where nb is the ***blocksize***.

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

8: 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 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 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 this routine may be followed by a call to F08AFF (SORGQR/DORGQR):

```
CALL SORGQR (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 F08AEF (SGEQRF/DGEQRF).

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

To apply Q to an arbitrary real rectangular matrix C, this routine may be followed by a call to F08AGF (SORMQR/DORMQR). For example,

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

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

To compute a QR factorization with column pivoting, use F08BEF (SGEQPF/DGEQPF).

The complex analogue of this routine is F08ASF (CGEQRF/ZGEQRF).

9 Example

To solve the linear least-squares problem

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

where b_1 and b_2 are the columns of the matrix B,

$$A = \begin{pmatrix} -0.57 & -1.28 & -0.39 & 0.25 \\ -1.93 & 1.08 & -0.31 & -2.14 \\ 2.30 & 0.24 & 0.40 & -0.35 \\ -1.93 & 0.64 & -0.66 & 0.08 \\ 0.15 & 0.30 & 0.15 & -2.13 \\ -0.02 & 1.03 & -1.43 & 0.50 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} -3.15 & 2.19 \\ -0.11 & -3.64 \\ 1.99 & 0.57 \\ -2.70 & 8.23 \\ 0.26 & -6.35 \\ 4.50 & -1.48 \end{pmatrix}.$$

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.

```
*     F08AEF 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)
      real             ONE
      PARAMETER        (ONE=1.0e0)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, INFO, J, M, N, NRHS
*     .. Local Arrays ..
      real             A(LDA,NMAX), B(LDB,NRHMAX), TAU(NMAX),
     +                 WORK(LWORK)
*     .. External Subroutines ..
      EXTERNAL         sgeqrf, sormqr, strsm, X04CAF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08AEF 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 sgeqrf(M,N,A,LDA,TAU,WORK,LWORK,INFO)
*
*        Compute C = (Q**T)*B, storing the result in B
*
         CALL sormqr('Left','Transpose',M,NRHS,N,A,LDA,TAU,B,LDB,WORK,
     +               LWORK,INFO)
*
*        Compute least-squares solution by backsubstitution in R*X = C
*
         CALL strsm('Left','Upper','No transpose','Non-Unit',N,NRHS,ONE,
     +              A,LDA,B,LDB)
*
*        Print least-squares solution(s)
*
         WRITE (NOUT,*)
         IFAIL = 0
*
         CALL X04CAF('General',' ',N,NRHS,B,LDB,
     +               'Least-squares solution(s)',IFAIL)
*
      END IF
      STOP
      END
```

9.2 Program Data

```
F08AEF 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
 -3.15   2.19
 -0.11  -3.64
  1.99   0.57
 -2.70   8.23
  0.26  -6.35
  4.50  -1.48                  :End of matrix B
```

9.3 Program Results

```
 F08AEF Example Program Results

 Least-squares solution(s)
             1          2
 1      1.5146    -1.5838
 2      1.8621     0.5536
 3     -1.4467     1.3491
 4      0.0396     2.9600
```