NAG Library Routine Document

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

2 Specification

```
SUBROUTINE F08AEF(M, N, A, LDA, TAU, WORK, LWORK, INFO)
INTEGER            M, N, LDA, LWORK, INFO
double precision   A(LDA,*), TAU(*), WORK(max(1,LWORK))
```

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

3 Description

F08AEF (DGEQRF) 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: $\mathrm{M} \geq 0$.

2: N – INTEGER *Input*

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

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

3: A(LDA,$*$) – ***double precision*** 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 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 (DGEQRF) is called.

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

5: TAU($*$) – ***double precision*** array *Output*

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

On exit: further details of the orthogonal matrix Q.

6: WORK($\max(1, \mathrm{LWORK})$) – ***double precision*** array *Workspace*

On exit: if $\mathrm{INFO} = 0$, 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 F08AEF (DGEQRF) 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: $\mathrm{LWORK} \geq \max(1, \mathrm{N})$ or $\mathrm{LWORK} = -1$.

8: INFO – INTEGER *Output*

On exit: $\mathrm{INFO} = 0$ 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 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 F08AEF (DGEQRF) may be followed by a call to F08AFF (DORGQR):

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

To apply Q to an arbitrary real rectangular matrix C, F08AEF (DGEQRF) may be followed by a call to F08AGF (DORMQR). For example,

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

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

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

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

9 Example

This example solves the linear least-squares problems

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

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

```
*     F08AEF Example Program Text
*     Mark 21 Release. NAG Copyright 2004.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          MMAX, NB, NMAX
      PARAMETER        (MMAX=8,NB=64,NMAX=8)
      INTEGER          LDA, LDB, NRHSMX, LWORK
      PARAMETER        (LDA=MMAX,LDB=MMAX,NRHSMX=NMAX,LWORK=NB*NMAX)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, INFO, J, M, N, NRHS
*     .. Local Arrays ..
      DOUBLE PRECISION A(LDA,NMAX), B(LDB,NRHSMX), RNORM(NRHSMX),
     +                 TAU(NMAX), WORK(LWORK)
*     .. External Functions ..
      DOUBLE PRECISION DNRM2
      EXTERNAL         DNRM2
*     .. External Subroutines ..
      EXTERNAL         DGEQRF, DORMQR, DTRTRS, X04CAF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08AEF Example Program Results'
      WRITE (NOUT,*)
*     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.NRHSMX)
     +    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 DGEQRF(M,N,A,LDA,TAU,WORK,LWORK,INFO)
*
*        Compute C = (C1) = (Q**T)*B, storing the result in B
*                    (C2)
*
         CALL DORMQR('Left','Transpose',M,NRHS,N,A,LDA,TAU,B,LDB,WORK,
     +               LWORK,INFO)
*
*        Compute least-squares solutions by backsubstitution in
*        R*X = C1
*
         CALL DTRTRS('Upper','No transpose','Non-Unit',N,NRHS,A,LDA,B,
     +               LDB,INFO)
*
         IF (INFO.GT.0) THEN
            WRITE (NOUT,*)
     +        'The upper triangular factor, R, of A is singular, '
            WRITE (NOUT,*)
     +        'the least squares solution could not be computed'
            GO TO 40
         END IF
*
*        Print least-squares solutions
*
         IFAIL = 0
         CALL X04CAF('General',' ',N,NRHS,B,LDB,
     +               'Least-squares solution(s)',IFAIL)
*
*        Compute and print estimates of the square roots of the residual
*        sums of squares
*
         DO 20 J = 1, NRHS
            RNORM(J) = DNRM2(M-N,B(N+1,J),1)
   20    CONTINUE
```

```
*
         WRITE (NOUT,*)
         WRITE (NOUT,*)
     +     'Square root(s) of the residual sum(s) of squares'
         WRITE (NOUT,99999) (RNORM(J),J=1,NRHS)
      ELSE
         WRITE (NOUT,*)
     +     'One or more of MMAX, NMAX or NRHSMX is too small, ',
     +     'and/or M.LT.N'
      END IF
   40 CONTINUE
*
99999 FORMAT (5X,1P,7E11.2)
      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

 -2.67   0.41
 -0.55  -3.10
  3.34  -4.01
 -0.77   2.76
  0.48  -6.17
  4.10   0.21               :End of matrix B
```

9.3 Program Results

```
 F08AEF Example Program Results

 Least-squares solution(s)
             1          2
 1      1.5339    -1.5753
 2      1.8707     0.5559
 3     -1.5241     1.3119
 4      0.0392     2.9585

 Square root(s) of the residual sum(s) of squares
        2.22E-02   1.38E-02
```