NAG Toolbox

nag_lapack_dgelqf (f08ah)

1 Purpose

nag_lapack_dgelqf (f08ah) computes the LQ factorization of a real m by n matrix.

2 Syntax

```
[a, tau, info] = nag_lapack_dgelqf(a, 'm', m, 'n', n)
[a, tau, info] = f08ah(a, 'm', m, 'n', n)
```

3 Description

nag_lapack_dgelqf (f08ah) forms the LQ factorization of an arbitrary rectangular real m by n matrix. No pivoting is performed.

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

$$A = \begin{pmatrix} L & 0 \end{pmatrix} Q$$

where L is an m by m lower triangular matrix and Q is an n by n orthogonal matrix. It is sometimes more convenient to write the factorization as

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

which reduces to

$$A = LQ_1,$$

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

If $m > n$, L is trapezoidal, and the factorization can be written

$$A = \begin{pmatrix} L_1 \\ L_2 \end{pmatrix} Q$$

where L_1 is lower triangular and L_2 is rectangular.

The LQ factorization of A is essentially the same as the QR factorization of A^{T}, since

$$A = \begin{pmatrix} L & 0 \end{pmatrix} Q \Leftrightarrow A^{\mathrm{T}} = Q^{\mathrm{T}} \begin{pmatrix} L^{\mathrm{T}} \\ 0 \end{pmatrix}.$$

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). Functions are provided to work with Q in this representation (see Section 9).

Note also that for any $k < m$, the information returned in the first k rows of the array **a** represents an LQ factorization of the first k rows of the original matrix A.

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

1: **a**$(lda,:)$ – REAL (KIND=nag_wp) array

The first dimension of the array **a** must be at least $\max(1,\mathbf{m})$.

The second dimension of the array **a** must be at least $\max(1,\mathbf{n})$.

The m by n matrix A.

5.2 Optional Input Parameters

1: **m** – INTEGER

Default: the first dimension of the array **a**.

m, the number of rows of the matrix A.

Constraint: $\mathbf{m} \geq 0$.

2: **n** – INTEGER

Default: the second dimension of the array **a**.

n, the number of columns of the matrix A.

Constraint: $\mathbf{n} \geq 0$.

5.3 Output Parameters

1: **a**$(lda,:)$ – REAL (KIND=nag_wp) array

The first dimension of the array **a** will be $\max(1,\mathbf{m})$.

The second dimension of the array **a** will be $\max(1,\mathbf{n})$.

If $m \leq n$, the elements above the diagonal store details of the orthogonal matrix Q and the lower triangle stores the corresponding elements of the m by m lower triangular matrix L.

If $m > n$, the strictly upper triangular part stores details of the orthogonal matrix Q and the remaining elements store the corresponding elements of the m by n lower trapezoidal matrix L.

2: **tau**$(:)$ – REAL (KIND=nag_wp) array

The dimension of the array **tau** will be $\max(1,\min(\mathbf{m},\mathbf{n}))$

Further details of the orthogonal matrix Q.

3: **info** – INTEGER

info $= 0$ unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

info $= -i$

If **info** $= -i$, parameter i had an illegal value on entry. The parameters are numbered as follows:

1: **m**, 2: **n**, 3: **a**, 4: **lda**, 5: **tau**, 6: **work**, 7: **lwork**, 8: **info**.

It is possible that **info** refers to a parameter that is omitted from the MATLAB interface. This usually indicates that an error in one of the other input parameters has caused an incorrect value to be inferred.

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 $\frac{2}{3}m^2(3n - m)$ if $m \leq n$ or $\frac{2}{3}n^2(3m - n)$ if $m > n$.

To form the orthogonal matrix Q nag_lapack_dgelqf (f08ah) may be followed by a call to nag_lapack_dorglq (f08aj):

```
[a, info] = f08aj(a, tau, 'k', min(m,n));
```

but note that the first dimension of the array **a**, specified by the argument lda, must be at least **n**, which may be larger than was required by nag_lapack_dgelqf (f08ah).

When $m \leq n$, it is often only the first m rows of Q that are required, and they may be formed by the call:

```
[a, info] = f08aj(a, tau, 'k', m);
```

To apply Q to an arbitrary real rectangular matrix C, nag_lapack_dgelqf (f08ah) may be followed by a call to nag_lapack_dormlq (f08ak). For example,

```
[c, info] = f08ak('Left', 'Transpose', a, tau, c, 'k', min(m, n));
```

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

The complex analogue of this function is nag_lapack_zgelqf (f08av).

9 Example

This example finds the minimum norm solutions of the under-determined systems of linear equations

$$Ax_1 = b_1 \quad \text{and} \quad Ax_2 = b_2$$

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

$$A = \begin{pmatrix} -5.42 & 3.28 & -3.68 & 0.27 & 2.06 & 0.46 \\ -1.65 & -3.40 & -3.20 & -1.03 & -4.06 & -0.01 \\ -0.37 & 2.35 & 1.90 & 4.31 & -1.76 & 1.13 \\ -3.15 & -0.11 & 1.99 & -2.70 & 0.26 & 4.50 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} -2.87 & -5.23 \\ 1.63 & 0.29 \\ -3.52 & 4.76 \\ 0.45 & -8.41 \end{pmatrix}.$$

9.1 Program Text

```
     function f08ah_example

fprintf('f08ah example results\n\n');

a = [-5.42, 3.28, -3.68, 0.27, 2.06, 0.46;
     -1.65, -3.4, -3.2, -1.03, -4.06, -0.01;
     -0.37, 2.35, 1.9, 4.31, -1.76, 1.13;
     -3.15, -0.11, 1.99, -2.7, 0.26, 4.5];
b = [-2.87, -5.23;
      1.63,  0.29;
     -3.52,  4.76;
      0.45, -8.41;
      0,     0;
      0,     0];
% Compute the LQ factorization of a
[a, tau, info] = f08ah(a);

% solve l*y=b
l = tril(a(:, 1:4));
```

```
b(1:4,:) = inv(l)*b(1:4,:);

% Compute minimum-norm solution x = (q^t)*b
[x, info] = f08ak( ...
    'Left', 'Transpose', a, tau, b);

mtitle = 'Minimum-norm solution(s)';
[ifail] = x04ca( ...
                 'General', ' ', x, mtitle);
```

9.2 Program Results

```
     f08ah example results

 Minimum-norm solution(s)
             1          2
 1      0.2371     0.7383
 2     -0.4575     0.0158
 3     -0.0085    -0.0161
 4     -0.5192     1.0768
 5      0.0239    -0.6436
 6     -0.0543    -0.6613
```