NAG Toolbox

nag_lapack_dgelsy (f08ba)

1 Purpose

nag_lapack_dgelsy (f08ba) computes the minimum norm solution to a real linear least squares problem

$$\min_x \|b - Ax\|_2$$

using a complete orthogonal factorization of A. A is an m by n matrix which may be rank-deficient. Several right-hand side vectors b and solution vectors x can be handled in a single call.

2 Syntax

```
[a, b, jpvt, rank, info] = nag_lapack_dgelsy(a, b, jpvt, rcond, 'm', m, 'n', n,
'nrhs_p', nrhs_p)
```

```
[a, b, jpvt, rank, info] = f08ba(a, b, jpvt, rcond, 'm', m, 'n', n, 'nrhs_p',
nrhs_p)
```

3 Description

The right-hand side vectors are stored as the columns of the m by r matrix B and the solution vectors in the n by r matrix X.

nag_lapack_dgelsy (f08ba) first computes a QR factorization with column pivoting

$$AP = Q\begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix},$$

with R_{11} defined as the largest leading sub-matrix whose estimated condition number is less than $1/$**rcond**. The order of R_{11}, **rank**, is the effective rank of A.

Then, R_{22} is considered to be negligible, and R_{12} is annihilated by orthogonal transformations from the right, arriving at the complete orthogonal factorization

$$AP = Q\begin{pmatrix} T_{11} & 0 \\ 0 & 0 \end{pmatrix}Z.$$

The minimum norm solution is then

$$X = PZ^{\mathrm{T}}\begin{pmatrix} T_{11}^{-1}Q_1^{\mathrm{T}}b \\ 0 \end{pmatrix}$$

where Q_1 consists of the first **rank** columns of Q.

4 References

Anderson E, Bai Z, Bischof C, Blackford S, Demmel J, Dongarra J J, Du Croz J J, Greenbaum A, Hammarling S, McKenney A and Sorensen D (1999) *LAPACK Users' Guide* (3rd Edition) SIAM, Philadelphia http://www.netlib.org/lapack/lug

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

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.

2: **b**$(ldb,:)$ – REAL (KIND=nag_wp) array

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

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

The m by r right-hand side matrix B.

3: **jpvt**$(:)$ – INTEGER array

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

If **jpvt**$(i) \neq 0$, the ith column of A is permuted to the front of AP, otherwise column i is a free column.

4: **rcond** – REAL (KIND=nag_wp)

Suggested value: if the condition number of **a** is not known then $\mathbf{rcond} = \sqrt{(\epsilon)/2}$ (where ϵ is ***machine precision***, see nag_machine_precision (x02aj)) is a good choice. Negative values or values less than ***machine precision*** should be avoided since this will cause **a** to have an effective rank $= \min(\mathbf{m},\mathbf{n})$ that could be larger than its actual rank, leading to meaningless results.

Used to determine the effective rank of A, which is defined as the order of the largest leading triangular sub-matrix R_{11} in the QR factorization of A, whose estimated condition number is $< 1/\mathbf{rcond}$.

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

3: **nrhs_p** – INTEGER

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

r, the number of right-hand sides, i.e., the number of columns of the matrices B and X.

Constraint: $\mathbf{nrhs\_p} \geq 0$.

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 $\max(1, \mathbf{n})$.

a stores details of its complete orthogonal factorization.

2: **b**(ldb, :) – REAL (KIND=nag_wp) array

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

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

The n by r solution matrix X.

3: **jpvt**(:) – INTEGER array

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

If $\mathbf{jpvt}(i) = k$, then the ith column of AP was the kth column of A.

4: **rank** – INTEGER

The effective rank of A, i.e., the order of the sub-matrix R_{11}. This is the same as the order of the sub-matrix T_{11} in the complete orthogonal factorization of A.

5: **info** – INTEGER

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

6 Error Indicators and Warnings

$\mathbf{info} = -i$

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

1: **m**, 2: **n**, 3: **nrhs_p**, 4: **a**, 5: **lda**, 6: **b**, 7: **ldb**, 8: **jpvt**, 9: **rcond**, 10: **rank**, 11: **work**, 12: **lwork**, 13: **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

See Section 4.5 of Anderson *et al.* (1999) for details of error bounds.

8 Further Comments

The complex analogue of this function is nag_lapack_zgelsy (f08bn).

9 Example

This example solves the linear least squares problem

$$\min_x \|b - Ax\|_2$$

for the solution, x, of minimum norm, where

$$A = \begin{pmatrix} -0.09 & 0.14 & -0.46 & 0.68 & 1.29 \\ -1.56 & 0.20 & 0.29 & 1.09 & 0.51 \\ -1.48 & -0.43 & 0.89 & -0.71 & -0.96 \\ -1.09 & 0.84 & 0.77 & 2.11 & -1.27 \\ 0.08 & 0.55 & -1.13 & 0.14 & 1.74 \\ -1.59 & -0.72 & 1.06 & 1.24 & 0.34 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} 7.4 \\ 4.2 \\ -8.3 \\ 1.8 \\ 8.6 \\ 2.1 \end{pmatrix}.$$

A tolerance of 0.01 is used to determine the effective rank of A.

Note that the block size (NB) of 64 assumed in this example is not realistic for such a small problem, but should be suitable for large problems.

9.1 Program Text

```
function f08ba_example

fprintf('f08ba example results\n\n');

a = [-0.09,  0.14, -0.46,  0.68,  1.29;
     -1.56,  0.20,  0.29,  1.09,  0.51;
     -1.48, -0.43,  0.89, -0.71, -0.96;
     -1.09,  0.84,  0.77,  2.11, -1.27;
      0.08,  0.55, -1.13,  0.14,  1.74;
     -1.59, -0.72,  1.06,  1.24,  0.34];
b = [ 7.4;
      4.2;
     -8.3;
      1.8;
      8.6;
      2.1];
[m,n] = size(a);

jpvt = zeros(n,1,nag_int_name);
rcond = 0.01;

[af, x, jpvt, rank, info] = f08ba( ...
    a, b, jpvt, rcond);

disp('Least squares solution');
disp(x');
disp('Tolerance used to estimate the rank of A');
disp(rcond);
disp('Estimated rank of A');
disp(rank);
```

9.2 Program Results

```
f08ba example results

Least squares solution
    0.6344    0.9699   -1.4402    3.3678    3.3992   -0.0035

Tolerance used to estimate the rank of A
    0.0100

Estimated rank of A
                    4
```
