NAG Toolbox for MATLAB

f08bn

1 Purpose

f08bn 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] = f08bn(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.

f08bn 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 $1/\mathbf{rcond}$. 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

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

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 URL: 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,**$*$) **– complex array**

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

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

The m by n matrix A.

2: **b**(**ldb**,$*$) – **complex array**

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

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

The m by r right-hand side matrix B.

3: **jpvt**($*$) – **int32 array**

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

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

4: **rcond** – **double scalar**

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** – **int32 scalar**

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

m, the number of rows of the matrix A.

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

2: **n** – **int32 scalar**

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** – **int32 scalar**

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 Input Parameters Omitted from the MATLAB Interface

lda, ldb, work, lwork, rwork

5.4 Output Parameters

1: **a**(**lda**,$*$) – **complex array**

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

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

a has been overwritten by details of its complete orthogonal factorization.

2: **b**(**ldb**,$*$) – **complex array**

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

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

the n by r solution matrix X.

3: **jpvt**$(*)$ **– int32 array**

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

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

4: **rank – int32 scalar**

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 – int32 scalar**

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

6 Error Indicators and Warnings

Errors or warnings detected by the function:

info $= -i$

If **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: **rwork**, 14: **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 real analogue of this function is f08ba.

9 Example

```
a = [complex(0.47, -0.34), complex(-0.4, +0.54), complex(0.6, +0.01),
complex(0.8, -1.02);
     complex(-0.32, -0.23), complex(-0.05, +0.2), complex(-0.26, -0.44),
complex(-0.43, +0.17);
      complex(0.35, -0.6), complex(-0.52, -0.34), complex(0.87, -0.11),
complex(-0.34, -0.09);
     complex(0.89, +0.71), complex(-0.45, -0.45), complex(-0.02, -0.57),
complex(1.14, -0.78);
      complex(-0.19, +0.06), complex(0.11, -0.85), complex(1.44, +0.8),
complex(0.07000000000000001, +1.14)];
b = [complex(-1.08, -2.59);
     complex(-2.61, -1.49);
     complex(3.13, -3.61);
     complex(7.33, -8.01);
     complex(9.119999999999999, +7.63)];
jpvt = [int32(0);
     int32(0);
     int32(0);
     int32(0)];
rcond = 0.01;
```

```
[aOut, bOut, jpvtOut, rank, info] = f08bn(a, b, jpvt, rcond)
```

```
aOut =
    2.5031                0.4809 - 0.6802i   -0.3516 - 1.0496i    0.0243 -
0.2110i
   -0.1420 + 0.0082i   -1.9697                 0.6274 + 0.2048i   -0.0617 +
0.1246i
   -0.0906 - 0.0591i   -0.4734 + 0.1684i   -1.2204                -0.2462 -
0.2637i
    0.4082 - 0.1178i    0.2585 + 0.0267i    0.0451 + 0.5815i   -0.0094
   -0.0896 + 0.3397i   -0.6872 + 0.0317i   -0.1171 - 0.0121i    0.8616 -
0.2123i
bOut =
    1.1669 - 3.3224i
    1.3486 + 5.5027i
    4.1764 + 2.3435i
    0.6467 + 0.0107i
   -0.1224 - 0.0933i
jpvtOut =
           4
           3
           2
           1
rank =
           3
info =
           0
```