NAG Toolbox for MATLAB

f08za

1 Purpose

f08za solves a real linear equality-constrained least-squares problem.

2 Syntax

```
[a, b, c, d, x, info] = f08za(a, b, c, d, 'm', m, 'n', n, 'p', p)
```

3 Description

f08za solves the real linear equality-constrained least-squares (LSE) problem

$$\underset{x}{\text{minimize}} \|c - Ax\|_2 \qquad \text{subject to} \qquad Bx = d$$

where A is an m by n matrix, B is a p by n matrix, c is an m element vector and d is a p element vector. It is assumed that $p \leq n \leq m + p$, $\text{rank}(B) = p$ and $\text{rank}(E) = n$, where $E = \begin{pmatrix} A \\ B \end{pmatrix}$. These conditions ensure that the LSE problem has a unique solution, which is obtained using a generalized RQ factorization of the matrices B and A.

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

Anderson E, Bai Z and Dongarra J 1992 Generalized *QR* factorization and its applications *Linear Algebra Appl. (Volume 162–164)* 243–271

Eldèn L 1980 Perturbation theory for the least-squares problem with linear equality constraints *SIAM J. Numer. Anal.* **17** 338–350

5 Parameters

5.1 Compulsory Input Parameters

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

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

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

The p by n matrix B.

3: **c**($*$) – **double array**

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

The right-hand side vector c for the least-squares part of the LSE problem.

4: **d**$(*)$ **– double array**

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

The right-hand side vector d for the equality constraints.

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** The second dimension of the array **b**.

n, the number of columns of the matrices A and B.

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

3: **p – int32 scalar**

Default: The dimension of the array **d**.

p, the number of rows of the matrix B.

Constraint: $0 \leq \mathbf{p} \leq \mathbf{n} \leq \mathbf{m} + \mathbf{p}$.

5.3 Input Parameters Omitted from the MATLAB Interface

lda, ldb, work, lwork

5.4 Output Parameters

1: **a**(**lda,***$*$*) **– double 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 is overwritten.

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

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

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

b is overwritten.

3: **c**$(*)$ **– double array**

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

The residual sum of squares for the solution vector x is given by the sum of squares of elements $\mathbf{c}(\mathbf{n} - \mathbf{p} + 1), \mathbf{c}(\mathbf{n} - \mathbf{p} + 2), \ldots, \mathbf{c}(\mathbf{m})$, provided $m + p > n$; the remaining elements are overwritten.

4: **d**$(*)$ **– double array**

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

d is overwritten.

5: **x(∗) – double array**

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

The solution vector x of the LSE problem.

6: **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: **p**, 4: **a**, 5: **lda**, 6: **b**, 7: **ldb**, 8: **c**, 9: **d**, 10: **x**, 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.

info $= 1$

The upper triangular factor R associated with B in the generalized RQ factorization of the pair (B, A) is singular, so that $\text{rank}(B) < P$; the least squares solution could not be computed.

info $= 2$

The $(N - P)$ by $(N - P)$ part of the upper trapezoidal factor T associated with A in the generalised RQ factorization of the pair (B, A) is singular, so that $\text{rank}(BA) < N$; the least squares solutions could not be computed.

7 Accuracy

For an error analysis, see Anderson *et al.* 1992 and Eldèn 1980. See also Section 4.6 of Anderson *et al.* 1999.

8 Further Comments

When $m \geq n = p$, the total number of floating-point operations is approximately $\frac{2}{3}n^2(6m + n)$; if $p \ll n$, the number reduces to approximately $\frac{2}{3}n^2(3m - n)$.

e04nc may also be used to solve LSE problems. It differs from f08za in that it uses an iterative (rather than direct) method, and that it allows general upper and lower bounds to be specified for the variables x and the linear constraints Bx.

9 Example

```
a = [-0.57, -1.28, -0.39, 0.25;
     -1.93, 1.08, -0.31, -2.14;
     2.3, 0.24, 0.4, -0.35;
     -1.93, 0.64, -0.66, 0.08;
     0.15, 0.3, 0.15, -2.13;
     -0.02, 1.03, -1.43, 0.5];
b = [1, 0, -1, 0;
     0, 1, 0, -1];
c = [-1.5;
     -2.14;
```

```
     1.23;
     -0.54;
     -1.68;
     0.82];
d = [0;
     0];
[aOut, bOut, cOut, dOut, x, info] = f08za(a, b, c, d)
```

```
aOut =
    3.3264   -0.2354    1.5296   -0.8479
    0.3955    2.0364    0.8978   -1.3588
   -0.4767    0.1207   -1.3372   -0.4674
    0.4573   -0.2835   -0.2594   -2.6447
   -0.0530    0.5095    0.3009   -0.0678
    0.2560   -0.4662    0.3760    0.4784
bOut =
   -0.4142         0    1.4142         0
         0   -0.4142         0    1.4142
cOut =
    0.6916
    1.4107
   -0.0052
   -0.0153
   -0.0126
   -0.0144
dOut =
     0
     0
x =
    0.4890
    0.9975
    0.4890
    0.9975
info =
           0
```