NAG Toolbox for MATLAB

f08zn

1 Purpose

f08zn solves a complex linear equality-constrained least-squares problem.

2 Syntax

```
[a, b, c, d, x, info] = f08zn(a, b, c, d, 'm', m, 'n', n, 'p', p)
```

3 Description

f08zn solves the complex 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 $p \leq n \leq m + p$, $\mathrm{rank}(B) = p$ and $\mathrm{rank}(\mathrm{E}) = n$, 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,$*$) – 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 $\max(1, \mathbf{p})$

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

The p by n matrix B.

3: **c($*$) – complex 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**$(*)$ **– complex 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**,$*$) **– 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 is overwritten.

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

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

d is overwritten.

5: **x(∗) – complex 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 $\mathrm{rank}(B) < P$; 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 $\mathrm{rank}(BA) < N$; 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 real floating-point operations is approximately $\frac{8}{3}n^2(6m + n)$; if $p \ll n$, the number reduces to approximately $\frac{8}{3}n^2(3m - n)$.

9 Example

```
a = [complex(0.96, -0.81), complex(-0.03, +0.96), complex(-0.91, +2.06),
complex(-0.05, +0.41);
      complex(-0.98, +1.98), complex(-1.2, +0.19), complex(-0.66, +0.42),
complex(-0.81, +0.56);
       complex(0.62, -0.46), complex(1.01, +0.02), complex(0.63, -0.17),
complex(-1.11, +0.6);
      complex(0.37, +0.38), complex(0.19, -0.54), complex(-0.98, -0.36),
complex(0.22, -0.2);
       complex(0.83, +0.51), complex(0.2, +0.01), complex(-0.17, -0.46),
complex(1.47, +1.59);
       complex(1.08, -0.28), complex(0.2, -0.12), complex(-0.07, +1.23),
complex(0.26, +0.26)];
b = [complex(1, +0), complex(0, +0), complex(-1, +0), complex(0, +0);
     complex(0, +0), complex(1, +0), complex(0, +0), complex(-1, +0)];
```

```
c = [complex(-2.54, +0.09);
     complex(1.65, -2.26);
     complex(-2.11, -3.96);
     complex(1.82, +3.3);
     complex(-6.41, +3.77);
     complex(2.07, +0.66)];
d = [complex(0, +0);
     complex(0, +0)];
[aOut, bOut, cOut, dOut, x, info] = f08zn(a, b, c, d)
```

```
aOut =
  -2.7118               -1.4390 - 1.0315i  -0.1054 + 1.3176i  -0.3924 -
0.1950i
  -0.2024 + 0.6829i  -1.8583               -0.9453 + 0.1928i   1.4355 +
0.2631i
   0.2443 - 0.2408i  -0.4279 + 0.1339i   2.9079               -0.2395 +
0.1886i
  -0.1408 + 0.0504i   0.2817 - 0.1483i  -0.4394 - 0.1145i  -1.5759
   0.1577 - 0.0379i   0.4242 + 0.4187i  -0.0867 - 0.7048i  -0.0589 +
0.0550i
   0.3069 + 0.1458i   0.0940 - 0.3436i  -0.1277 + 0.1658i   0.1632 -
0.4069i
bOut =
   -0.4142         0    1.4142         0
         0   -0.4142         0    1.4142
cOut =
   1.5379 - 2.7748i
  -1.0478 + 5.2746i
  -0.1373 + 0.0467i
  -0.0471 - 0.0105i
   0.0001 - 0.0025i
  -0.0418 - 0.0099i
dOut =
     0
     0
x =
   1.0874 - 1.9621i
  -0.7409 + 3.7297i
   1.0874 - 1.9621i
  -0.7409 + 3.7297i
info =
           0
```