NAG Toolbox for MATLAB

f04kl

1 Purpose

f04kl solves a complex general Gauss–Markov linear (least-squares) model problem.

2 Syntax

```
[a, b, d, x, y, ifail] = f04kl(a, b, d, 'm', m, 'n', n, 'p', p)
```

3 Description

f04kl solves the complex general Gauss–Markov linear model (GLM) problem

$$\underset{x}{\text{minimize}} \|y\|_2 \qquad \text{subject to} \qquad d = Ax + By$$

where A is an m by n matrix, B is an m by p matrix and d is an m element vector. It is assumed that $n \leq m \leq n+p$, $\text{rank}(A) = n$ and $\text{rank}(E) = m$, where $E = (A \quad B)$. Under these assumptions, the problem has a unique solution x and a minimal 2-norm solution y, which is obtained using a generalized QR factorization of the matrices A and B.

In particular, if the matrix B is square and nonsingular, then the GLM problem is equivalent to the weighted linear least-squares problem

$$\underset{x}{\text{minimize}} \left\|B^{-1}(d - Ax)\right\|_2.$$

f04kl is based on the LAPACK routine CGGGLM/ZGGGLM, see Anderson *et al.* 1999.

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

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 $\max(1, \mathbf{m})$

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

The m by p matrix B.

3: **d**$(*)$ **– complex array**

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

The left-hand side vector d of the GLM equation.

5.2 Optional Input Parameters

1: **m – int32 scalar**

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

m, the number of rows of the matrices A and B.

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: $0 \leq \mathbf{n} \leq \mathbf{m}$.

3: **p – int32 scalar**

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

p, the number of columns of the matrix B.

Constraint: $\mathbf{p} \geq \mathbf{m} - \mathbf{n}$.

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

The array is overwritten.

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

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

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

The array is overwritten.

3: **d**$(*)$ **– complex array**

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

The array is overwritten.

4: **x**$(*)$ **– complex array**

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

The solution vector x of the GLM problem.

5: **y(∗) – complex array**

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

The solution vector y of the GLM problem.

6: **ifail – int32 scalar**

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

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, $\mathbf{m} < 0$,
or $\mathbf{n} < 0$,
or $\mathbf{n} > \mathbf{m}$,
or $\mathbf{p} < 0$,
or $\mathbf{p} < \mathbf{m} - \mathbf{n}$,
or $\mathbf{lda} < \max(1, \mathbf{m})$,
or $\mathbf{ldb} < \max(1, \mathbf{m})$,
or $\mathbf{lwork} < \max(1, \mathbf{m} + \mathbf{n} + \mathbf{p})$ and $\mathbf{lwork} \neq -1$.

7 Accuracy

For an error analysis, see Anderson *et al.* 1992.

8 Further Comments

When $p = m \geq n$, the total number of real floating-point operations is approximately $\frac{8}{3}(2m^3 - n^3) + 16nm^2$; when $p = m = n$, the total number of real floating-point operations is approximately $\frac{56}{3}m^3$.

9 Example

```
a = [complex(0.96, -0.8100000000000001), complex(-0.03, +0.96), complex(-
0.91, +2.06);
     complex(-0.98, +1.98), complex(-1.2, +0.19), complex(-0.66, +0.42);
     complex(0.62, -0.46), complex(1.01, +0.02), complex(0.63, -0.17);
               complex(1.08,   -0.28),   complex(0.2,   -0.12),   complex(-
0.07000000000000001, +1.23)];
b = [complex(0.5, -1), complex(0, +0), complex(0, +0), complex(0, +0);
     complex(0, +0), complex(1, -2), complex(0, +0), complex(0, +0);
     complex(0, +0), complex(0, +0), complex(2, -3), complex(0, +0);
     complex(0, +0), complex(0, +0), complex(0, +0), complex(5, -4)];
d = [complex(6.01, -0.37);
     complex(-5.27, +0.9);
     complex(2.72, -2.13);
     complex(-1.34, -2.8)];
[aOut, bOut, dOut, x, y, ifail] = f04kl(a, b, d)
```

```
aOut =
  -2.8809            -0.5597 - 1.2116i   0.3523 - 1.2597i
  -0.3484 + 0.4420i   1.3009             1.5520 - 0.4961i
   0.1787 - 0.0821i  -0.5245 - 0.0361i   1.8059
   0.2839 - 0.0130i  -0.2338 + 0.4152i  -0.3539 - 0.7964i
bOut =
    1.8936              -1.3320 + 0.3960i    0.1213 - 1.4943i    1.4096 +
0.7140i
```

```
    0.3756 - 0.1350i    1.9049               -2.7958 - 0.0662i    0.4497 -
0.1602i
    0.1262 + 0.0131i    0.1989 + 0.0903i    4.1872               -1.5248 +
1.9624i
  -0.0015 + 0.0595i   0.1014 + 0.1408i   0.3901 + 0.3498i  -3.8214
dOut =
  -1.0000 + 2.0000i
   4.0000 - 5.0000i
  -3.0000 + 1.0000i
         0 + 0.0000i
x =
  -1.0000 + 2.0000i
   4.0000 - 5.0000i
  -3.0000 + 1.0000i
y =
   1.0e-15 *
  -0.0180 - 0.0073i
  -0.0288 - 0.0486i
  -0.0536 - 0.1620i
   0.1303 - 0.0660i
ifail =
           0
```