NAG Fortran Library Routine Document

F04JLF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

F04JLF solves a real general Gauss–Markov linear (least-squares) model problem.

2 Specification

```
SUBROUTINE F04JLF (M, N, P, A, LDA, B, LDB, D, X, Y, WORK, LWORK, IFAIL)
INTEGER            M, N, P, LDA, LDB, LWORK, IFAIL
double precision   A(LDA,*), B(LDB,*), D(*), X(*), Y(*), WORK(*)
```

3 Description

F04JLF solves the real general Gauss–Markov linear model (GLM) problem

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

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 $E = \begin{pmatrix} A & B \end{pmatrix}$. 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 non-singular, then the GLM problem is equivalent to the weighted linear least-squares problem

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

F04JLF is based on the LAPACK routine SGGGLM/DGGGLM, 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

1: M – INTEGER *Input*

On entry: m, the number of rows of the matrices A and B.

Constraint: $\text{M} \geq 0$.

2: N – INTEGER *Input*

On entry: n, the number of columns of the matrix A.

Constraint: $0 \leq \text{N} \leq \text{M}$.

3: P – INTEGER *Input*

On entry: p, the number of columns of the matrix B.

Constraint: $P \geq M - N$.

4: A(LDA,$*$) – ***double precision*** array *Input/Output*

Note: the second dimension of the array A must be at least $\max(1, N)$.

On entry: the m by n matrix A.

On exit: is overwritten.

5: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F04JLF is called.

Constraint: $LDA \geq \max(1, M)$.

6: B(LDB,$*$) – ***double precision*** array *Input/Output*

Note: the second dimension of the array B must be at least $\max(1, P)$.

On entry: the m by p matrix B.

On exit: is overwritten.

7: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F04JLF is called.

Constraint: $LDB \geq \max(1, M)$.

8: D($*$) – ***double precision*** array *Input/Output*

Note: the dimension of the array D must be at least $\max(1, M)$.

On entry: the left-hand side vector d of the GLM equation.

On exit: is overwritten.

9: X($*$) – ***double precision*** array *Output*

Note: the dimension of the array X must be at least $\max(1, N)$.

On exit: the solution vector x of the GLM problem.

10: Y($*$) – ***double precision*** array *Output*

Note: the dimension of the array Y must be at least $\max(1, P)$.

On exit: the solution vector y of the GLM problem.

11: WORK($*$) – ***double precision*** array *Workspace*

Note: the dimension of the array WORK must be at least $\max(1, LWORK)$.

On exit: if IFAIL $= 0$, WORK(1) contains the minimum value of LWORK required for optimum performance.

12: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the subprogram from which F04JLF is called unless LWORK $= -1$, in which case a workspace query is assumed and the routine only calculates the optimal dimension of WORK (using the formula given below).

Suggested value: for optimum performance LWORK should be at least $\mathrm{N} + \min(\mathrm{M}, \mathrm{P}) + \max(\mathrm{M}, \mathrm{P}) \times nb$, where nb is the ***blocksize***.

Constraint: $\mathrm{LWORK} \geq \max(1, \mathrm{M} + \mathrm{N} + \mathrm{P})$ or $\mathrm{LWORK} = -1$.

13: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Chapter P01 for details.

On exit: $\mathrm{IFAIL} = 0$ unless the routine detects an error (see Section 6).

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this parameter the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

6 Error Indicators and Warnings

If on entry $\mathrm{IFAIL} = 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

$\mathrm{IFAIL} = 1$

On entry, $\mathrm{M} < 0$,
or $\mathrm{N} < 0$,
or $\mathrm{N} > \mathrm{M}$,
or $\mathrm{P} < 0$,
or $\mathrm{P} < \mathrm{M} - \mathrm{N}$,
or $\mathrm{LDA} < \max(1, \mathrm{M})$,
or $\mathrm{LDB} < \max(1, \mathrm{M})$,
or $\mathrm{LWORK} < \max(1, \mathrm{M} + \mathrm{N} + \mathrm{P})$ and $\mathrm{LWORK} \neq -1$.

7 Accuracy

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

8 Further Comments

When $p = m \geq n$, the total number of floating-point operations is approximately $\frac{2}{3}(2m^3 - n^3) + 4nm^2$; when $p = m = n$, the total number of floating-point operations is approximately $\frac{14}{3}m^3$.

9 Example

This example solves the weighted least-squares problem

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

where

$$B = \begin{pmatrix} 0.5 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 2.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 5.0 \end{pmatrix}, \quad d = \begin{pmatrix} 1.31 \\ -4.01 \\ 5.56 \\ 3.22 \end{pmatrix} \quad \text{and} \quad A = \begin{pmatrix} -0.57 & -1.28 & -0.39 \\ -1.93 & 1.08 & -0.31 \\ 2.30 & 0.24 & -0.40 \\ -0.02 & 1.03 & -1.43 \end{pmatrix}.$$

9.1 Program Text

```
*     F04JLF Example Program Text
*     Mark 20 Revised. NAG Copyright 2001.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, MMAX, PMAX, LDA, LDB, LWORK
      PARAMETER        (NMAX=10,MMAX=10,PMAX=10,LDA=MMAX,LDB=MMAX,
     +                 LWORK=NMAX+MMAX+64*(MMAX+PMAX))
*     .. Local Scalars ..
      INTEGER          I, IFAIL, J, M, N, P
*     .. Local Arrays ..
      DOUBLE PRECISION A(LDA,NMAX), B(LDB,PMAX), D(MMAX), WORK(LWORK),
     +                 X(NMAX), Y(PMAX)
*     .. External Subroutines ..
      EXTERNAL         F04JLF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F04JLF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) M, N, P
      IF (M.LE.MMAX .AND. N.LE.NMAX .AND. P.LE.PMAX) THEN
*
*        Read A, B and D from data file
*
         READ (NIN,*) ((A(I,J),J=1,N),I=1,M)
         READ (NIN,*) ((B(I,J),J=1,P),I=1,M)
         READ (NIN,*) (D(I),I=1,M)
*
*        Solve the weighted least-squares problem
*
*        minimize ||inv(B)*(D-A*X)|| (in the 2-norm)
*
         IFAIL = 0
*
         CALL F04JLF(M,N,P,A,LDA,B,LDB,D,X,Y,WORK,LWORK,IFAIL)
*
*        Print least-squares solution
*
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Least-squares solution'
         WRITE (NOUT,99999) (X(I),I=1,N)
      END IF
      STOP
*
99999 FORMAT (1X,8F9.4)
      END
```

9.2 Program Data

```
F04JLF Example Program Data
  4  3  4                     :Values of M, N and P
-0.57  -1.28  -0.39
-1.93   1.08  -0.31
 2.30   0.24  -0.40
-0.02   1.03  -1.43           :End of matrix A
 0.50   0.00   0.00   0.00
 0.00   1.00   0.00   0.00
 0.00   0.00   2.00   0.00
 0.00   0.00   0.00   5.00 :End of matrix B
 1.31
-4.01
 5.56
 3.22                     :End of D
```

9.3 Program Results

```
 F04JLF Example Program Results

 Least-squares solution
    2.0000  -1.0000  -3.0000
```