

## F04JLF – NAG Fortran Library Routine Document

**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,
1             IFAIL)
  INTEGER      M, N, P, LDA, LDB, LWORK, IFAIL
  real         A(LDA,*), B(LDB,*), D(*), X(*), Y(*), WORK(LWORK)

```

### 3 Description

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

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

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  $E = (A \ B)$ . 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

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

F04JLF is based on the LAPACK routine SGGGLM/DGGGLM, see [1].

### 4 References

- [1] Anderson E, Bai Z, Bischof C, Demmel J, Dongarra J J, Du Croz J J, Greenbaum A, Hammarling S, McKenney A, Ostrouchov S and Sorensen D (1995) *LAPACK Users' Guide* (2nd Edition) SIAM, Philadelphia
- [2] Anderson E, Bai Z and Dongarra J (1991) Generalized  $QR$  factorization and its applications *LAPACK Working Note No. 31* University of Tennessee, Knoxville

### 5 Parameters

- 1: M — INTEGER *Input*  
*On entry:*  $m$ , the number of rows of the matrices  $A$  and  $B$ .  
*Constraint:*  $M \geq 0$ .
- 2: N — INTEGER *Input*  
*On entry:*  $n$ , the number of columns of the matrix  $A$ .  
*Constraint:*  $0 \leq N \leq M$ .
- 3: P — INTEGER *Input*  
*On entry:*  $p$ , the number of columns of the matrix  $B$ .  
*Constraint:*  $P \geq M - N$ .

- 4:** A(LDA,\*) — *real* 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:*  $A$  is overwritten.
- 5:** LDA — INTEGER *Input*  
*On entry:* the leading 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,\*) — *real* 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:*  $B$  is overwritten.
- 7:** LDB — INTEGER *Input*  
*On entry:* the leading dimension of the array B as declared in the (sub)program from which F04JLF is called.  
*Constraint:*  $LDB \geq \max(1,M)$ .
- 8:** D(\*) — *real* 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:* D is overwritten.
- 9:** X(\*) — *real* 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(\*) — *real* 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(LWORK) — *real* array *Workspace*  
*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.  
*Suggested value:* for optimum performance LWORK should be at least  $N + \min(M,P) + \max(M, N, P) \times nb$ , where  $nb$  is the **blocksize**.  
*Constraint:*  $LWORK \geq \max(1, M+N+P)$ .
- 13:** IFAIL — INTEGER *Input/Output*  
*On entry:* IFAIL must be set to 0, -1 or 1. For users not familiar with this parameter (described in Chapter P01) the recommended value is 0.  
*On exit:* IFAIL = 0 unless the routine detects an error (see Section 6).

## 6 Errors and Warnings

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

Errors detected by the routine:

IFAIL = 1

- On entry, M < 0,
- or N < 0,
- or N > M,
- or P < 0,
- or P < M - N,
- or LDA < max(1,M),
- or LDB < max(1,M),
- or LWORK < max(1,M+N+P).

## 7 Accuracy

For an error analysis, see [2].

## 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

To solve the weighted least-squares problem

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

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

**Note.** The listing of the example program presented below uses bold italicised terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*      F04JLF Example Program Text.
*      Mark 17 Release. NAG Copyright 1995.
*      .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER       (NIN=5, NOUT=6)
      INTEGER          NMAX, MMAX, PMAX, LDA, LDB, LWORK
      PARAMETER       (NMAX=10, MMAX=10, PMAX=10, LDA=NMAX, LDB=NMAX,
+                    LWORK=MMAX+NMAX+64*(NMAX+PMAX))
*      .. Local Scalars ..
      INTEGER          I, IFAIL, J, M, N, P
```

```

*   .. Local Arrays ..
  real          A(LDA,MMAX), B(LDB,PMAX), D(NMAX), WORK(LWORK),
+             X(MMAX), Y(PMAX)
*   .. External Subroutines ..
  EXTERNAL      F04JLF
*   .. Executable Statements ..
  WRITE (NOUT,*) 'F04JLF Example Program Results'
*   Skip heading in data file
  READ (NIN,*)
  READ (NIN,*) N, M, P
  IF (N.LE.NMAX .AND. M.LE.MMAX .AND. P.LE.PMAX) THEN

*
*       Read A, B and D from data file
*
  READ (NIN,*) ((A(I,J),J=1,M),I=1,N)
  READ (NIN,*) ((B(I,J),J=1,P),I=1,N)
  READ (NIN,*) (D(I),I=1,N)

*
*       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,M)
  END IF
  STOP

*
99999 FORMAT (1X,8F9.4)
  END

```

## 9.2 Program Data

```

F04JLF Example Program Data
  4 3 4           :Values of N, M 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

---