F04JMF – 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

F04JMF solves a real linear equality-constrained least-squares problem.

2 Specification

```
      SUBROUTINE F04JMF(M, N, P, A, LDA, B, LDB, C, D, X, WORK, LWORK,
     1                  IFAIL)
      INTEGER           M, N, P, LDA, LDB, LWORK, IFAIL
      real              A(LDA,*), B(LDB,*), C(*), D(*), X(*), WORK(LWORK)
```

3 Description

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

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

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 $\text{E} = \begin{pmatrix} A \\ B \end{pmatrix}$. 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.

F04JMF is based on the LAPACK routine SGGLSE/DGGLSE, 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

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

5 Parameters

1: M — INTEGER *Input*

On entry: m, the number of rows of the matrix A.

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

2: N — INTEGER *Input*

On entry: n, the number of columns of the matrices A and B.

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

3: P — INTEGER *Input*

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

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

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 F04JMF 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,N).

On entry: the p by n 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 F04JMF is called.

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

8: C(∗) — ***real*** array *Input/Output*

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

On entry: the right-hand side vector c for the least-squares part of the LSE problem.

On exit: the residual sum of squares for the solution vector x is given by the sum of squares of elements C(N−P+1), C(N−P+2),. . .,C(M), provided $m + p > n$; the remaining elements are overwritten.

9: D(∗) — ***real*** array *Input/Output*

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

On entry: the right-hand side vector d for the equality constraints.

On exit: D is overwritten.

10: 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 LSE 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 F04JMF is called.

Suggested value: for optimum performance LWORK should be at least P + min(M,N) + 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 P < 0,
or P > N,
or P < N − M,
or LDA < max(1,M),
or LDB < max(1,P),
or LWORK < max(1,M+N+P).

7 Accuracy

For an error analysis, see [2] and [3].

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 $\frac{2}{3}n^2(3m-n)$.

E04NCF may also be used to solve LSE problems. It differs from F04JMF 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

To solve the least-squares problem

$$\underset{x}{\text{minimize}}\, \|c - Ax\|_2 \;\text{ subject to }\; x_1 = x_3 \;\text{ and }\; x_2 = x_4$$

where

$$c = \begin{pmatrix} -3.15 \\ -0.11 \\ 1.99 \\ -2.70 \\ 0.26 \\ 4.50 \end{pmatrix}$$

and

$$A = \begin{pmatrix} -0.57 & -1.28 & -0.39 & 0.25 \\ -1.93 & 1.08 & -0.31 & -2.14 \\ 2.30 & 0.24 & 0.40 & -0.35 \\ -1.93 & 0.64 & -0.66 & 0.08 \\ 0.15 & 0.30 & 0.15 & -2.13 \\ -0.02 & 1.03 & -1.43 & 0.50 \end{pmatrix};$$

the equality constraints are formulated by setting

$$B = \begin{pmatrix} 1.0 & 0.0 & -1.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & -1.0 \end{pmatrix}$$

and

$$d = \begin{pmatrix} 0.0 \\ 0.0 \end{pmatrix}.$$

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.

```
*     F04JMF Example Program Text.
*     Mark 17 Release. NAG Copyright 1995.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          MMAX, NMAX, PMAX, LDA, LDB, LWORK
      PARAMETER        (MMAX=10,NMAX=10,PMAX=10,LDA=MMAX,LDB=PMAX,
     +                 LWORK=PMAX+NMAX+64*(MMAX+NMAX))
*     .. Local Scalars ..
      real             RSS
      INTEGER          I, IFAIL, J, M, N, P
*     .. Local Arrays ..
      real             A(LDA,NMAX), B(LDB,NMAX), C(MMAX), D(PMAX),
     +                 WORK(LWORK), X(NMAX)
*     .. External Functions ..
      real             sdot
      EXTERNAL         sdot
*     .. External Subroutines ..
      EXTERNAL         F04JMF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F04JMF 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, C and D from data file
*
         READ (NIN,*) ((A(I,J),J=1,N),I=1,M)
         READ (NIN,*) ((B(I,J),J=1,N),I=1,P)
         READ (NIN,*) (C(I),I=1,M)
         READ (NIN,*) (D(I),I=1,P)
*
*        Solve the equality-constrained least-squares problem
*
*        minimize ||C - A*X|| (in the 2-norm) subject to B*X = D
*
         IFAIL = 0
*
         CALL F04JMF(M,N,P,A,LDA,B,LDB,C,D,X,WORK,LWORK,IFAIL)
*
*        Print least-squares solution
*
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Constrained least-squares solution'
         WRITE (NOUT,99999) (X(I),I=1,N)
*
*        Compute the residual sum of squares
*
         WRITE (NOUT,*)
         RSS = sdot(M-N+P,C(N-P+1),1,C(N-P+1),1)
         WRITE (NOUT,99998) 'Residual sum of squares = ', RSS
      END IF
      STOP
```

```
*
99999 FORMAT (1X,8F9.4)
99998 FORMAT (1X,A,1P,e10.2)
      END
```

9.2 Program Data

```
F04JMF Example Program Data
  6  4  2                   :Values of M, N and P
 -0.57  -1.28  -0.39   0.25
 -1.93   1.08  -0.31  -2.14
  2.30   0.24   0.40  -0.35
 -1.93   0.64  -0.66   0.08
  0.15   0.30   0.15  -2.13
 -0.02   1.03  -1.43   0.50 :End of matrix A
  1.00   0.00  -1.00   0.00
  0.00   1.00   0.00  -1.00 :End of matrix B
 -3.15
 -0.11
  1.99
 -2.70
  0.26
  4.50                      :End of C
  0.00
  0.00                      :End of D
```

9.3 Program Results

```
 F04JMF Example Program Results

 Constrained least-squares solution
    0.4857   0.9956   0.4857   0.9956

 Residual sum of squares =   2.95E+01
```