F04JGF – 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

F04JGF finds the solution of a linear least-squares problem, $Ax = b$, where A is a real m by n $(m \geq n)$ matrix and b is an m element vector. If the matrix of observations is not of full rank, then the minimal least-squares solution is returned.

2 Specification

```
 SUBROUTINE F04JGF(M, N, A, NRA, B, TOL, SVD, SIGMA, IRANK, WORK,
1                  LWORK, IFAIL)
 INTEGER           M, N, NRA, IRANK, LWORK, IFAIL
 real              A(NRA,N), B(M), TOL, SIGMA, WORK(LWORK)
 LOGICAL           SVD
```

3 Description

The minimal least-squares solution of the problem $Ax = b$ is the vector x of minimum (Euclidean) length which minimizes the length of the residual vector $r = b - Ax$.

The real m by n $(m \geq n)$ matrix A is factorized as

$$A = Q \begin{pmatrix} U \\ 0 \end{pmatrix}$$

where Q is an m by m orthogonal matrix and U is an n by n upper triangular matrix. If U is of full rank, then the least-squares solution is given by

$$x = (U^{-1}\ 0)Q^T b.$$

If U is not of full rank, then the singular value decomposition of U is obtained so that U is factorized as

$$U = RDP^T,$$

where R and P are n by n orthogonal matrices and D is the n by n diagonal matrix

$$D = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n),$$

with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$, these being the singular values of A. If the singular values $\sigma_{k+1}, \ldots, \sigma_n$ are negligible, but σ_k is not negligible, relative to the data errors in A, then the rank of A is taken to be k and the minimal least-squares solution is given by

$$x = P \begin{pmatrix} S^{-1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} R^T & 0 \\ 0 & I \end{pmatrix} Q^T b,$$

where $S = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_k)$.

This routine obtains the factorizations by a call to F02WDF.

The routine also returns the value of the standard error

$$\begin{aligned} \sigma &= \sqrt{\frac{r^T r}{m-k}}, \quad \text{if } m > k, \\ &= 0, \qquad\qquad \text{if } m = k, \end{aligned}$$

$r^T r$ being the residual sum of squares and k the rank of A.

4 References

[1] Lawson C L and Hanson R J (1974) *Solving Least-squares Problems* Prentice–Hall

5 Parameters

1: M — INTEGER *Input*

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

Constraint: M $\geq$ N.

2: N — INTEGER *Input*

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

Constraint: $1 \leq$ N $\leq$ M.

3: A(NRA,N) — ***real*** array *Input/Output*

On entry: the m by n matrix A.

On exit: if SVD is returned as .FALSE., A is overwritten by details of the QU factorization of A (see F02WDF for further details). If SVD is returned as .TRUE., the first n rows of A are overwritten by the right-hand singular vectors, stored by rows; and the remaining rows of the array are used as workspace.

4: NRA — INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F04JGF is called.

Constraint: NRA $\geq$ M.

5: B(M) — ***real*** array *Input/Output*

On entry: the right-hand side vector b.

On exit: the first n elements of B contain the minimal least-squares solution vector x. The remaining $m - n$ elements are used for workspace.

6: TOL — ***real*** *Input*

On entry: a relative tolerance to be used to determine the rank of A. TOL should be chosen as approximately the largest relative error in the elements of A. For example, if the elements of A are correct to about 4 significant figures then TOL should be set to about 5×10^{-4}. See Section 8 for a description of how TOL is used to determine rank. If TOL is outside the range $(\epsilon, 1.0)$, where ϵ is the ***machine precision***, then the value ϵ is used in place of TOL. For most problems this is unreasonably small.

7: SVD — LOGICAL *Output*

On exit: SVD is returned as .FALSE. if the least-squares solution has been obtained from the QU factorization of A. In this case A is of full rank. SVD is returned as .TRUE. if the least-squares solution has been obtained from the singular value decomposition of A.

8: SIGMA — ***real*** *Output*

On exit: the standard error, i.e., the value $\sqrt{r^T r/(m-k)}$ when $m > k$, and the value zero when $m = k$. Here r is the residual vector $b - Ax$ and k is the rank of A.

9: IRANK — INTEGER *Output*

On exit: k, the rank of the matrix A. It should be noted that it is possible for IRANK to be returned as n and SVD to be returned as .TRUE.. This means that the matrix U only just failed the test for non-singularity.

10: WORK(LWORK) — ***real*** array *Output*

On exit: if SVD is returned as .FALSE., then the first n elements of WORK contain information on the QU factorization of A (see parameter A above and F02WDF), and WORK$(n+1)$ contains the condition number $\|U\|_E\|U^{-1}\|_E$ of the upper triangular matrix U.

If SVD is returned as .TRUE., then the first n elements of WORK contain the singular values of A arranged in descending order and WORK$(n+1)$ contains the total number of iterations taken by the QR algorithm. The rest of WORK is used as workspace.

11: LWORK — INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F04JGF is called.

Constraint: LWORK $\geq 4 \times$ N.

12: 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 Error Indicators and Warnings

Errors detected by the routine:

IFAIL = 1

On entry, N < 1,
or M < N,
or NRA < M,
or LWORK < 4 × N.

IFAIL = 2

The QR algorithm has failed to converge to the singular values in 50 × N iterations. This failure can only happen when the singular value decomposition is employed, but even then it is not likely to occur.

7 Accuracy

The computed factors Q, U, R, D and P^T satisfy the relations

$$Q\begin{pmatrix} U \\ 0 \end{pmatrix} = A + E\ ,\ Q\begin{pmatrix} R & 0 \\ 0 & I \end{pmatrix}\begin{pmatrix} D \\ 0 \end{pmatrix} P^T = A + F,$$

where

$$\|E\|_2 \leq c_1\epsilon\|A\|_2, \|F\|_2 \leq c_2\epsilon\|A\|_2,$$

ϵ being the ***machine precision***, and c_1 and c_2 being modest functions of m and n. Note that $\|A\|_2 = \sigma_1$.

For a fuller discussion, covering the accuracy of the solution x see Lawson and Hanson [1], especially pp 50 and 95.

8 Further Comments

If the least-squares solution is obtained from the QU factorization then the time taken by the routine is approximately proportional to $n^2(3m-n)$. If the least-squares solution is obtained from the singular value decomposition then the time taken is approximately proportional to $n^2(3m+19n)$. The approximate proportionality factor is the same in each case.

This routine is column biased and so is suitable for use in paged environments.

Following the QU factorization of A the condition number

$$c(U) = \|U\|_E \|U^{-1}\|_E$$

is determined and if $c(U)$ is such that

$$c(U) \times \mathrm{TOL} > 1.0$$

then U is regarded as singular and the singular values of A are computed. If this test is not satisfied, U is regarded as non-singular and the rank of A is set to n. When the singular values are computed the rank of A, say k, is returned as the largest integer such that

$$\sigma_k > \mathrm{TOL} \times \sigma_1,$$

unless $\sigma_1 = 0$ in which case k is returned as zero. That is, singular values which satisfy $\sigma_i \leq \mathrm{TOL} \times \sigma_1$ are regarded as negligible because relative perturbations of order TOL can make such singular values zero.

9 Example

To obtain a least-squares solution for $r = b - Ax$, where

$$A = \begin{pmatrix} 0.05 & 0.05 & 0.25 & -0.25 \\ 0.25 & 0.25 & 0.05 & -0.05 \\ 0.35 & 0.35 & 1.75 & -1.75 \\ 1.75 & 1.75 & 0.35 & -0.35 \\ 0.30 & -0.30 & 0.30 & 0.30 \\ 0.40 & -0.40 & 0.40 & 0.40 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{pmatrix}$$

and the value TOL is to be taken as 5×10^{-4}.

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.

```
*     F04JGF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          MMAX, NMAX, NRA, LWORK
      PARAMETER        (MMAX=8,NMAX=MMAX,NRA=MMAX,LWORK=4*NMAX)
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*     .. Local Scalars ..
      real             SIGMA, TOL
      INTEGER          I, IFAIL, IRANK, J, M, N
      LOGICAL          SVD
*     .. Local Arrays ..
      real             A(NRA,NMAX), B(MMAX), WORK(LWORK)
*     .. External Subroutines ..
      EXTERNAL         F04JGF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F04JGF Example Program Results'
```

```
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) M, N
      TOL = 5.0e-4
      WRITE (NOUT,*)
      IF (M.GT.0 .AND. M.LE.MMAX .AND. N.GT.0 .AND. N.LE.NMAX) THEN
         READ (NIN,*) ((A(I,J),J=1,N),I=1,M)
         READ (NIN,*) (B(I),I=1,M)
         IFAIL = 0
*
         CALL F04JGF(M,N,A,NRA,B,TOL,SVD,SIGMA,IRANK,WORK,LWORK,IFAIL)
*
         WRITE (NOUT,*) 'Solution vector'
         WRITE (NOUT,99996) (B(I),I=1,N)
         WRITE (NOUT,*)
         WRITE (NOUT,99998) 'Standard error = ', SIGMA, '    Rank = ',
     +     IRANK
         WRITE (NOUT,*)
         WRITE (NOUT,99997) 'SVD = ', SVD
      ELSE
         WRITE (NOUT,99999) 'M or N out of range: M = ', M, '  N = ', N
      END IF
      STOP
*
99999 FORMAT (1X,A,I5,A,I5)
99998 FORMAT (1X,A,F6.3,A,I2)
99997 FORMAT (1X,A,L2)
99996 FORMAT (1X,8F9.3)
      END
```

9.2 Program Data

```
F04JGF Example Program Data
  6  4
  0.05  0.05  0.25 -0.25
  0.25  0.25  0.05 -0.05
  0.35  0.35  1.75 -1.75
  1.75  1.75  0.35 -0.35
  0.30 -0.30  0.30  0.30
  0.40 -0.40  0.40  0.40
  1.0   2.0   3.0   4.0   5.0   6.0
```

9.3 Program Results

```
 F04JGF Example Program Results

 Solution vector
     4.967   -2.833    4.567    3.233

 Standard error =  0.909    Rank =  3

 SVD =  T
```