NAG Fortran Library Routine Document

F04JDF

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

F04JDF finds the minimal solution of a linear least-squares problem, $Ax = b$, where A is a real m by $n(m \leq n)$ matrix and b is an m element vector.

2 Specification

```
      SUBROUTINE F04JDF(M, N, A, NRA, B, TOL, SIGMA, IRANK, WORK, LWORK,
     1                  IFAIL)
      INTEGER           M, N, NRA, IRANK, LWORK, IFAIL
      real              A(NRA,N), B(N), TOL, SIGMA, WORK(LWORK)
```

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 $n(m \leq n)$ matrix A may be factorized as the singular value decomposition (SVD) into

$$A = Q(\,D \quad 0\,)P^T,$$

where Q is an m by m orthogonal matrix, P is an n by n orthogonal matrix and D is the m by m diagonal matrix

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

with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_m \geq 0$, these being the singular values of A. The first m columns of P are the right-hand singular vectors of A.

If the singular values $\sigma_{k+1}, \ldots, \sigma_m$ 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

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

where $S = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_k)$. The routine also returns the value of the standard error

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

$r^T r$ being the residual sum of squares.

4 References

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: $1 \leq \mathrm{M} \leq \mathrm{N}$.

2: N – INTEGER *Input*

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

Constraint: N $\geq$ M.

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

On entry: the m by n matrix A.

On exit: A is overwritten by the first m rows of the n by n matrix P^T, i.e., the right-hand singular vectors, stored by rows.

4: NRA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F04JDF is called.

Constraint: NRA $\geq$ M.

5: B(N) – ***real*** array *Input/Output*

On entry: the first m elements must contain the right-hand side vector b.

On exit: the n element solution vector x.

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: 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.

8: IRANK – INTEGER *Output*

On exit: k, the rank of the matrix A.

9: WORK(LWORK) – ***real*** array *Output*

On exit: the first m elements of WORK contain the singular values of A arranged in descending order. WORK$(m+1)$ contains the total number of iterations taken by the QR algorithm. The rest of WORK is used as workspace.

10: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F04JDF is called.

Constraint: LWORK $\geq$ M $\times$ (M + 4).

11: IFAIL – INTEGER *Input/Output*

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

On exit: 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, for users 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 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:

IFAIL $= 1$

On entry, M < 1,
or N $<$ M,
or NRA $<$ M,
or LWORK $<$ M $\times$ (M $+ 4$).

IFAIL $= 2$

The QR algorithm has failed to converge to the singular values in 50 × N iterations. This failure is not likely to occur.

7 Accuracy

The computed factors Q, D and P^T satisfy the relation

$$Q(\,D \quad 0\,)P^T = A + E,$$

where

$$\|E\|_2 \leq c\epsilon\|A\|_2,$$

ϵ being the ***machine precision*** and c being a most function 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 (1974), especially pages 50 and 95.

8 Further Comments

The time taken by the routine is approximately proportional to $m^2(n+m)$.

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

F04JAF gives the minimal least-squares solution for the case $m > n$.

The rank of A, say k, is returned as the largest integer such that

$$\sigma_k >$$

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

9 Example

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

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

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.

```
*     F04JDF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NMAX, MMAX, NRA, LWORK
      PARAMETER        (NMAX=8,MMAX=NMAX,NRA=MMAX,LWORK=MMAX*(MMAX+4))
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*     .. Local Scalars ..
      real             SIGMA, TOL
      INTEGER          I, IFAIL, IRANK, J, M, N
*     .. Local Arrays ..
      real             A(NRA,NMAX), B(NMAX), WORK(LWORK)
*     .. External Subroutines ..
      EXTERNAL         F04JDF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F04JDF 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 F04JDF(M,N,A,NRA,B,TOL,SIGMA,IRANK,WORK,LWORK,IFAIL)
*
         WRITE (NOUT,*) 'Solution vector'
         WRITE (NOUT,99997) (B(I),I=1,N)
         WRITE (NOUT,*)
         WRITE (NOUT,99998) 'Standard error = ', SIGMA, '    Rank = ',
     +     IRANK
      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,8F9.3)
      END
```

9.2 Program Data

```
F04JDF Example Program Data
  4  6
  0.05  0.25  0.35  1.75  0.30  0.40
  0.05  0.25  0.35  1.75 -0.30 -0.40
  0.25  0.05  1.75  0.35  0.30  0.40
 -0.25 -0.05 -1.75 -0.35  0.30  0.40
   1.0   2.0   3.0   4.0
```

9.3 Program Results

```
 F04JDF Example Program Results

 Solution vector
    -0.067    0.133   -0.467    0.933    1.800    2.400

 Standard error =  4.000    Rank =  3
```