NAG Toolbox for MATLAB

f04jd

1 Purpose

f04jd 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 Syntax

```
[a, b, sigma, irank, work, ifail] = f04jd(m, a, b, tol, lwork, 'n', n)
```

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

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

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

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

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

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

$r^{\mathrm{T}} r$ being the residual sum of squares.

4 References

Lawson C L and Hanson R J 1974 *Solving Least-squares Problems* Prentice–Hall

5 Parameters

5.1 Compulsory Input Parameters

1: **m – int32 scalar**

m, the number of rows of **a**.

Constraint: $1 \leq \mathbf{m} \leq \mathbf{n}$.

2: **a(lda,n) – double array**

lda, the first dimension of the array, must be at least **m**.

The m by n matrix A.

3: **b(n) – double array**

The first m elements must contain the right-hand side vector b.

4: **tol – double scalar**

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.

5: **lwork – int32 scalar**

Constraint: $\mathbf{lwork} \geq \mathbf{m} \times (\mathbf{m} + 4)$.

5.2 Optional Input Parameters

1: **n – int32 scalar**

Default: The dimension of the arrays **a**, **b**. (An error is raised if these dimensions are not equal.)

n, the number of columns of **a**.

Constraint: $\mathbf{n} \geq \mathbf{m}$.

5.3 Input Parameters Omitted from the MATLAB Interface

lda

5.4 Output Parameters

1: **a(lda,n) – double array**

The array contains the first m rows of the n by n matrix P^{T}, i.e., the right-hand singular vectors, stored by rows.

2: **b(n) – double array**

The n element solution vector x.

3: **sigma – double scalar**

The standard error, i.e., the value $\sqrt{r^{\mathrm{T}}r/(m-k)}$ 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.

4: **irank – int32 scalar**

k, the rank of the matrix A.

5: **work(lwork) – double array**

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.

6: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, **m** < 1,
or **n** $<$ **m**,
or **lda** $<$ **m**,
or **lwork** $<$ **m** $\times$ (**m** $+ 4$).

ifail $= 2$

The *QR* algorithm has failed to converge to the singular values in $50 \times$ **n** iterations. This failure is not likely to occur.

7 Accuracy

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

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

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 f04jd is approximately proportional to $m^2(n + m)$.

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

f04ja 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

$$\sigma_k > \textbf{tol} \times \sigma_1,$$

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

9 Example

```
m = int32(4);
a = [0.05, 0.25, 0.35, 1.75, 0.3, 0.4;
     0.05, 0.25, 0.35, 1.75, -0.3, -0.4;
     0.25, 0.05, 1.75, 0.35, 0.3, 0.4;
     -0.25, -0.05, -1.75, -0.35, 0.3, 0.4];
b = [1;
     2;
     3;
     4;
     0;
     0];
tol = 0.0005;
lwork = int32(96);
[aOut, bOut, sigma, irank, work, ifail] = f04jd(m, a, b, tol, lwork)
```

```
aOut =
```

```
    0.1000    0.1000    0.7000    0.7000    0.0000    0.0000
    0.1000   -0.1000    0.7000   -0.7000         0    0.0000
   -0.0000   -0.0000    0.0000    0.0000   -0.6000   -0.8000
   -0.0000   -0.0000    0.0000    0.0000    0.8000   -0.6000
bOut =
   -0.0667
    0.1333
   -0.4667
    0.9333
    1.8000
    2.4000
sigma =
     4
irank =
           3
work =
     array elided
ifail =
           0
```