

NAG Toolbox for MATLAB

f04ja

1 Purpose

f04ja finds the minimal 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.

2 Syntax

```
[a, b, sigma, irank, work, ifail] = f04ja(a, b, tol, lwork, 'm', m, '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 ($m \geq n$) matrix A may be factorized by the singular value decomposition (SVD) as

$$A = Q \begin{pmatrix} D \\ 0 \end{pmatrix} P^T,$$

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

$$D = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$$

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

If the singular values $\sigma_{k+1}, \dots, \sigma_n$ 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

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

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

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

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

4 References

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

5 Parameters

5.1 Compulsory Input Parameters

1: **a(lda,n)** – double array

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

The m by n matrix A .

2: **b(m) – double array**

The right-hand side vector b .

3: **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 instead of **tol**. For most problems this is unreasonably small.

4: **lwork – int32 scalar**

Constraint: $\text{lwork} \geq 4 \times n$.

5.2 Optional Input Parameters1: **m – int32 scalar**

Default: The dimension of the array **b**.

m , the number of rows of the matrix A .

Constraint: $m \geq n$.

2: **n – int32 scalar**

Default: The dimension of the array **a**.

n , the number of columns of the matrix A .

Constraint: $1 \leq n \leq m$.

5.3 Input Parameters Omitted from the MATLAB Interface

lda

5.4 Output Parameters1: **a(lda,n) – double array**

The first n rows of **a** are overwritten by the n by n matrix P^T , i.e., the right-hand singular vectors, stored by **rows**. The rest of the array is used as workspace.

2: **b(m) – double array**

The first n elements of **b** contain the minimal least-squares solution vector x . The remaining $m - n$ elements are used for workspace.

3: **sigma – double scalar**

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 .

4: **irank – int32 scalar**

k , the rank of the matrix A .

5: **work(lwork) – double array**

The first n elements of **work** contain the singular values of A arranged in descending order. **work**($n + 1$) 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, **n** < 1,
or **m** < **n**,
or **lda** < **m**,
or **lwork** < 4 × **n**.

ifail = 2

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

7 Accuracy

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

$$Q \begin{pmatrix} D \\ 0 \end{pmatrix} P^T = A + E,$$

where $\|E\|_2 \leq c\epsilon\|A\|_2$, ϵ being the *machine precision* and c being a modest 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 f04ja is approximately proportional to $n^2(m + 6n)$.

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

f04jd 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

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

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

9 Example

```
a = [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;
      0.3, -0.3, 0.3, 0.3;
      0.4, -0.4, 0.4, 0.4];
b = [1;
      2;
      3;
      4;
      5;
      6];
tol = 0.0005;
```

```
lwork = int32(32);
[aOut, bOut, sigma, irank, work, ifail] = f04ja(a, b, tol, lwork)
```