NAG Toolbox for MATLAB

f08ve

1 Purpose

f08ve uses orthogonal transformations to simultaneously reduce the m by n matrix A and the p by n matrix B to upper triangular form. This factorization is usually used as a preprocessing step for computing the generalized singular value decomposition (GSVD).

2 Syntax

```
[a, b, k, l, u, v, q, info] = f08ve(jobu, jobv, jobq, a, b, tola, tolb,
'm', m, 'p', p, 'n', n)
```

3 Description

f08ve computes orthogonal matrices U, V and Q such that

$$U^{\mathrm{T}}AQ = \begin{cases} \begin{array}{r} \\ k \\ l \\ m-k-l \end{array} \begin{array}{c} \begin{array}{ccc} n-k-l & k & l \end{array} \\ \begin{pmatrix} 0 & A_{12} & A_{13} \\ 0 & 0 & A_{23} \\ 0 & 0 & 0 \end{pmatrix} \end{array} & \text{if } m-k-l \geq 0; \\ \begin{array}{r} \\ k \\ m-k \end{array} \begin{array}{c} \begin{array}{ccc} n-k-l & k & l \end{array} \\ \begin{pmatrix} 0 & A_{12} & A_{13} \\ 0 & 0 & A_{23} \end{pmatrix} \end{array} & \text{if } m-k-l < 0; \end{cases}$$

$$V^{\mathrm{T}}BQ = \begin{array}{r} \\ l \\ p-l \end{array} \begin{array}{c} \begin{array}{ccc} n-k-l & k & l \end{array} \\ \begin{pmatrix} 0 & 0 & B_{13} \\ 0 & 0 & 0 \end{pmatrix} \end{array}$$

where the k by k matrix A_{12} and l by l matrix B_{13} are nonsingular upper triangular; A_{23} is l by l upper triangular if $m-k-l \geq 0$ and is $(m-k)$ by l upper trapezoidal otherwise. $(k+l)$ is the effective numerical rank of the $(m+p)$ by n matrix $\begin{pmatrix} A^{\mathrm{T}} & B^{\mathrm{T}} \end{pmatrix}^{\mathrm{T}}$.

This decomposition is usually used as the preprocessing step for computing the Generalized Singular Value Decomposition (GSVD), see function f08va.

4 References

Anderson E, Bai Z, Bischof C, Blackford S, Demmel J, Dongarra J J, Du Croz J J, Greenbaum A, Hammarling S, McKenney A and Sorensen D 1999 *LAPACK Users' Guide* (3rd Edition) SIAM, Philadelphia URL: http://www.netlib.org/lapack/lug

Golub G H and Van Loan C F 1996 *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

5 Parameters

5.1 Compulsory Input Parameters

1: **jobu – string**

If **jobu** = 'U', the orthogonal matrix U is computed.

If **jobu** = 'N', **u** is not computed.

Constraint: **jobu** = 'U' or 'N'.

2: **jobv – string**

If **jobv** = 'V', the orthogonal matrix V is computed.

If **jobv** = 'N', **v** is not computed.

Constraint: **jobv** = 'V' or 'N'.

3: **jobq – string**

If **jobq** = 'Q', the orthogonal matrix Q is computed.

If **jobq** = 'N', **q** is not computed.

Constraint: **jobq** = 'Q' or 'N'.

4: **a**(**lda**,$*$) **– double array**

The first dimension of the array **a** must be at least $\max(1, \mathbf{m})$

The second dimension of the array must be at least $\max(1, \mathbf{n})$

The m by n matrix A.

5: **b**(**ldb**,$*$) **– double array**

The first dimension of the array **b** must be at least $\max(1, \mathbf{p})$

The second dimension of the array must be at least $\max(1, \mathbf{n})$

The p by n matrix B.

6: **tola – double scalar**

7: **tolb – double scalar**

tola and **tolb** are the thresholds to determine the effective numerical rank of matrix B and a subblock of A. Generally, they are set to

$$\begin{aligned} \mathbf{tola} &= \max(\mathbf{m}, \mathbf{n})\|A\|\epsilon, \\ \mathbf{tolb} &= \max(\mathbf{p}, \mathbf{n})\|B\|\epsilon, \end{aligned}$$

where ϵ is the ***machine precision***.

The size of **tola** and **tolb** may affect the size of backward errors of the decomposition.

5.2 Optional Input Parameters

1: **m – int32 scalar**

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

m, the number of rows of the matrix A.

Constraint: $\mathbf{m} \geq 0$.

2: **p – int32 scalar**

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

p, the number of rows of the matrix B.

Constraint: $\mathbf{p} \geq 0$.

3: **n – int32 scalar**

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

n, the number of columns of the matrices A and B.

Constraint: $\mathbf{n} \geq 0$.

5.3 Input Parameters Omitted from the MATLAB Interface

lda, ldb, ldu, ldv, ldq, iwork, tau, work

5.4 Output Parameters

1: **a**(**lda**,$*$) – **double array**

The first dimension of the array **a** must be at least $\max(1, \mathbf{m})$

The second dimension of the array must be at least $\max(1, \mathbf{n})$

Contains the triangular (or trapezoidal) matrix described in Section 3.

2: **b**(**ldb**,$*$) – **double array**

The first dimension of the array **b** must be at least $\max(1, \mathbf{p})$

The second dimension of the array must be at least $\max(1, \mathbf{n})$

contains the triangular matrix described in Section 3.

3: **k – int32 scalar**

4: **l – int32 scalar**

k and **l** specify the dimension of the subblocks k and l as described in Section 3; $(k + l)$ is the effective numerical rank of $\begin{pmatrix} \mathbf{a}^{\mathrm{T}} & \mathbf{b}^{\mathrm{T}} \end{pmatrix}^{\mathrm{T}}$.

5: **u**(**ldu**,$*$) – **double array**

The first dimension, **ldu**, of the array **u** must satisfy

if **jobu** = 'U', **ldu** $\geq \max(1, \mathbf{m})$;
ldu ≥ 1 otherwise.

The second dimension of the array must be at least $\max(1, \mathbf{m})$

If **jobu** = 'U', **u** contains the orthogonal matrix U.

If **jobu** = 'N', **u** is not referenced.

6: **v**(**ldv**,$*$) – **double array**

The first dimension, **ldv**, of the array **v** must satisfy

if **jobv** = 'V', **ldv** $\geq \max(1, \mathbf{p})$;
ldv ≥ 1 otherwise.

The second dimension of the array must be at least $\max(1, \mathbf{m})$

If **jobv** = 'V', **v** contains the orthogonal matrix V.

If **jobv** = 'N', **v** is not referenced.

7: **q**(**ldq**,$*$) – **double array**

The first dimension, **ldq**, of the array **q** must satisfy

if **jobq** = 'Q', **ldq** $\geq \max(1, \mathbf{n})$;
ldq ≥ 1 otherwise.

The second dimension of the array must be at least $\max(1, \mathbf{n})$

If **jobq** = 'Q', **q** contains the orthogonal matrix Q.

If **jobq** = 'N', **q** is not referenced.

8: **info – int32 scalar**

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

6 Error Indicators and Warnings

Errors or warnings detected by the function:

info $= -i$

If **info** $= -i$, parameter i had an illegal value on entry. The parameters are numbered as follows:

1: **jobu**, 2: **jobv**, 3: **jobq**, 4: **m**, 5: **p**, 6: **n**, 7: **a**, 8: **lda**, 9: **b**, 10: **ldb**, 11: **tola**, 12: **tolb**, 13: **k**, 14: **l**, 15: **u**, 16: **ldu**, 17: **v**, 18: **ldv**, 19: **q**, 20: **ldq**, 21: **iwork**, 22: **tau**, 23: **work**, 24: **info**.

It is possible that **info** refers to a parameter that is omitted from the MATLAB interface. This usually indicates that an error in one of the other input parameters has caused an incorrect value to be inferred.

7 Accuracy

The computed factorization is nearly the exact factorization for nearby matrices $(A + E)$ and $(B + F)$, where

$$\|E\|_2 = O(\epsilon)\|A\|_2 \qquad \text{and} \qquad \|F\|_2 = O(\epsilon)\|B\|_2,$$

and ϵ is the ***machine precision***.

8 Further Comments

The complex analogue of this function is f08vs.

9 Example

```
jobu = 'U';
jobv = 'V';
jobq = 'Q';
a = [1, 2, 3;
     3, 2, 1;
     4, 5, 6;
     7, 8, 8];
b = [-2, -3, 3;
     4, 6, 5];
tola = 8.001412032943023e-15;
tolb = 3.000529512353634e-15;
[aOut, bOut, k, l, u, v, q, info] = f08ve(jobu, jobv, jobq, a, b, tola,
tolb)
```

```
aOut =
   -2.0569   10.7706   -7.2814
         0    7.1947   -7.5262
         0         0    0.5813
         0         0         0
bOut =
         0    8.0623   -3.1305
         0         0   -4.9193
k =
          1
l =
          2
u =
   -0.1348    0.5103   -0.2435    0.8137
    0.6742   -0.5467   -0.3535    0.3487
```

```
    0.2697    0.4829   -0.6913   -0.4650
    0.6742    0.4556    0.5813    0.0000
v =
   -0.4472    0.8944         0         0
    0.8944    0.4472         0         0
q =
   -0.8321    0.5547         0
    0.5547    0.8321         0
         0         0   -1.0000
info =
           0
```