NAG Toolbox for MATLAB

f08va

1 Purpose

f08va computes the generalized singular value decomposition (GSVD) of an m by n real matrix A and a p by n real matrix B.

2 Syntax

```
[k, l, a, b, alpha, beta, u, v, q, info] = f08va(jobu, jobv, jobq, a, b,
'm', m, 'n', n, 'p', p)
```

3 Description

The generalized singular value decomposition is given by

$$U^{\mathrm{T}}AQ = D_1(\,0 \quad R\,), \qquad V^{\mathrm{T}}BQ = D_2(\,0 \quad R\,),$$

where U, V and Q are orthogonal matrices. Let $(k+l)$ be the effective numerical rank of the matrix $\left(\,A^{\mathrm{T}} \quad B^{\mathrm{T}}\,\right)^{\mathrm{T}}$, then R is a $(k+l)$ by $(k+l)$ nonsingular upper triangular matrix, D_1 and D_2 are m by $(k+l)$ and p by $(k+l)$ 'diagonal' matrices structured as follows:

if $m-k-l \geq 0$,

$$D_1 = \begin{array}{r c} & \begin{array}{cc} k & l \end{array} \\ \begin{array}{r} k \\ l-l \\ m-k-l \end{array} & \begin{pmatrix} I & 0 \\ 0 & C \\ 0 & 0 \end{pmatrix} \end{array}$$

$$D_2 = \begin{array}{r c} & \begin{array}{cc} k & l \end{array} \\ \begin{array}{r} l \\ p-l \end{array} & \begin{pmatrix} 0 & S \\ 0 & 0 \end{pmatrix} \end{array}$$

$$(\,0 \quad R\,) = \begin{array}{r c} & \begin{array}{ccc} n-k-l & k & l \end{array} \\ \begin{array}{r} k \\ l \end{array} & \begin{pmatrix} 0 & R_{11} & R_{12} \\ 0 & 0 & R_{22} \end{pmatrix} \end{array}$$

where

$$C = \mathrm{diag}(\alpha_{k+1}, \ldots, \alpha_{k+l}),$$

$$S = \mathrm{diag}(\beta_{k+1}, \ldots, \beta_{k+l}),$$

and

$$C^2 + S^2 = I.$$

R is stored in $\mathbf{a}(1:, k+1, n-k-l+1:, \mathrm{n})$ on exit.

If $m-k-l < 0$,

$$D_1 = \begin{array}{r c} & \begin{array}{ccc} k & m-k & k+l-m \end{array} \\ \begin{array}{r} k \\ m-k \end{array} & \begin{pmatrix} I & 0 & 0 \\ 0 & C & 0 \end{pmatrix} \end{array}$$

$$D_2 = \begin{array}{r} \\ m-k \\ k+l-m \\ p-l \end{array} \begin{array}{c} \begin{array}{ccc} k & m-k & k+l-m \end{array} \\ \begin{pmatrix} 0 & S & 0 \\ 0 & 0 & I \\ 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$\begin{pmatrix} 0 & R \end{pmatrix} = \begin{array}{r} \\ k \\ m-k \\ k+l-m \end{array} \begin{array}{c} \begin{array}{cccc} n-k-l & k & m-k & k+l-m \end{array} \\ \begin{pmatrix} 0 & R_{11} & R_{12} & R_{13} \\ 0 & 0 & R_{22} & R_{23} \\ 0 & 0 & 0 & R_{33} \end{pmatrix} \end{array}$$

where

$$C = \mathrm{diag}(\alpha_{k+1}, \dots, \alpha_m),$$

$$S = \mathrm{diag}(\beta_{k+1}, \dots, \beta_m),$$

and

$$C^2 + S^2 = I.$$

$\begin{pmatrix} R_{11} & R_{12} & R_{13} \\ 0 & R_{22} & R_{23} \end{pmatrix}$ is stored in $\mathbf{a}(1:, m, n-k-l+1:, n)$, and R_{33} is stored in $\mathbf{b}(m-k+1:, l, n+m-k-l+1:, n)$ on exit.

The function computes C, S, R and, optionally, the orthogonal transformation matrices U, V and Q.

In particular, if B is an n by n nonsingular matrix, then the GSVD of A and B implicitly gives the SVD of AB^{-1}:

$$AB^{-1} = U\left(D_1 D_2^{-1}\right)V^{\mathrm{T}}.$$

If $\begin{pmatrix} A^{\mathrm{T}} & B^{\mathrm{T}} \end{pmatrix}^{\mathrm{T}}$ has orthonormal columns, then the GSVD of A and B is also equal to the CS decomposition of A and B. Furthermore, the GSVD can be used to derive the solution of the eigenvalue problem:

$$A^{\mathrm{T}}Ax = \lambda B^{\mathrm{T}}Bx.$$

In some literature, the GSVD of A and B is presented in the form

$$U^{\mathrm{T}}AX = \begin{pmatrix} 0 & D_1 \end{pmatrix}, \qquad V^{\mathrm{T}}BX = \begin{pmatrix} 0 & D_2 \end{pmatrix},$$

where U and V are orthogonal and X is nonsingular, D_1 and D_2 are 'diagonal'. The former GSVD form can be converted to the latter form by taking the nonsingular matrix X as

$$X = Q\begin{pmatrix} I & 0 \\ 0 & R^{-1} \end{pmatrix}.$$

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.

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: **n – int32 scalar**

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

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

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

3: **p – int32 scalar**

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

p, the number of rows of the matrix B.

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

5.3 Input Parameters Omitted from the MATLAB Interface

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

5.4 Output Parameters

1: **k – int32 scalar**

2: **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 $\begin{pmatrix} \mathbf{a}^{\mathrm{T}} & \mathbf{b}^{\mathrm{T}} \end{pmatrix}^{\mathrm{T}}$.

3: **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 matrix R, or part of R. See Section 3 for details.

4: **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 R if $m-k-l<0$. See Section 3 for details.

5: **alpha($*$) – double array**

Note: the dimension of the array **alpha** must be at least $\max(1,\mathbf{n})$.

See the description of **beta**.

6: **beta($*$) – double array**

Note: the dimension of the array **beta** must be at least $\max(1,\mathbf{n})$.

alpha and **beta** contain the generalized singular value pairs of A and B, α_i and β_i;

$\mathbf{alpha}(1:\mathbf{k})=1$,

$\mathbf{beta}(1:\mathbf{k})=0$,

and if $m-k-l\geq 0$,

$\mathbf{alpha}(\mathbf{k}+1:\mathbf{k}+\mathbf{l})=C$,

$\mathbf{beta}(\mathbf{k}+1:\mathbf{k}+\mathbf{l})=S$,

or if $m-k-l<0$,

$\mathbf{alpha}(\mathbf{k}+1:\mathbf{m})=C$,

$\mathbf{alpha}(\mathbf{m}+1:\mathbf{k}+\mathbf{l})=0$,

$\mathbf{beta}(\mathbf{k}+1:\mathbf{m})=S$,

$\mathbf{beta}(\mathbf{m}+1:\mathbf{k}+\mathbf{l})=1$, and

$\mathbf{alpha}(\mathbf{k}+\mathbf{l}+1:\mathbf{n})=0$,

$\mathbf{beta}(\mathbf{k}+\mathbf{l}+1:\mathbf{n})=0$.

7: **u(ldu,$*$) – double array**

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

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

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

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

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

8: **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 $\max(1, \mathbf{p})$

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

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

9: **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 n by n orthogonal matrix Q.

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

10: **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: **n**, 6: **p**, 7: **k**, 8: **l**, 9: **a**, 10: **lda**, 11: **b**, 12: **ldb**, 13: **alpha**, 14: **beta**, 15: **u**, 16: **ldu**, 17: **v**, 18: **ldv**, 19: **q**, 20: **ldq**, 21: **work**, 22: **iwork**, 23: **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.

info $= 1$

If **info** $= 1$, the Jacobi-type procedure failed to converge.

7 Accuracy

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

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

and ϵ is the ***machine precision***. See Section 4.12 of Anderson *et al.* 1999 for further details.

8 Further Comments

The complex analogue of this function is f08vn.

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];
[k, l, aOut, bOut, alpha, beta, u, v, q, info] = f08va(jobu, jobv, jobq,
a, b)
```

```
k =
           1
l =
           2
aOut =
   -2.0569   -9.0121   -9.3705
         0  -10.8822   -7.2688
         0         0   -6.0405
         0         0         0
bOut =
         0   -6.5869   -4.3998
         0         0   -6.0405
alpha =
    1.0000
    0.7960
    0.0799
beta =
         0
    0.6053
    0.9968
u =
   -0.1348    0.5252   -0.2092    0.8137
    0.6742   -0.5221   -0.3889    0.3487
    0.2697    0.5276   -0.6578   -0.4650
    0.6742    0.4161    0.6101    0.0000
v =
    0.3554   -0.9347
    0.9347    0.3554
q =
   -0.8321   -0.0946   -0.5466
    0.5547   -0.1419   -0.8199
         0   -0.9853    0.1706
info =
           0
```