NAG Toolbox

nag_lapack_dggsvd (f08va)

1 Purpose

nag_lapack_dggsvd (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, iwork, info] = nag_lapack_dggsvd(jobu, jobv,
jobq, a, b, 'm', m, 'n', n, 'p', p)
```

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

Note: the interface to this routine has changed since earlier releases of the toolbox:

At Mark 23: **iwork** was made an output parameter.

3 Description

The generalized singular value decomposition is given by

$$U^{\mathrm{T}}AQ = D_1\begin{pmatrix}0 & R\end{pmatrix}, \quad V^{\mathrm{T}}BQ = D_2\begin{pmatrix}0 & R\end{pmatrix},$$

where U, V and Q are orthogonal matrices. Let $(k+l)$ be the effective numerical rank of the matrix $\begin{pmatrix}A\\B\end{pmatrix}$, 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 $m-k-l \geq 0$,

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

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

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

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 as a submatrix of A with elements R_{ij} stored as $A_{i,n-k-l+j}$ on exit.

If $m - k - l < 0$,

$$D_1 = \begin{matrix} & \begin{matrix} k & m-k & k+l-m \end{matrix} \\ \begin{matrix} k \\ m-k \end{matrix} & \begin{pmatrix} I & 0 & 0 \\ 0 & C & 0 \end{pmatrix} \end{matrix}$$

$$D_2 = \begin{matrix} & \begin{matrix} k & m-k & k+l-m \end{matrix} \\ \begin{matrix} m-k \\ k+l-m \\ p-l \end{matrix} & \begin{pmatrix} 0 & S & 0 \\ 0 & 0 & I \\ 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

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

where

$$C = \mathrm{diag}(\alpha_{k+1}, \ldots, \alpha_m),$$
$$S = \mathrm{diag}(\beta_{k+1}, \ldots, \beta_m),$$

and

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

$\begin{pmatrix} R_{11} & R_{12} & R_{13} \\ 0 & R_{22} & R_{23} \end{pmatrix}$ is stored as a submatrix of A with R_{ij} stored as $A_{i,n-k-l+j}$, and R_{33} is stored as a submatrix of B with $(R_{33})_{ij}$ stored as $B_{m-k+i,n+m-k-l+j}$.

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 $\begin{pmatrix} A \\ B \end{pmatrix}$ 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

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

where U and V are orthogonal and X is nonsingular, and 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 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** – CHARACTER(1)

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

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

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

2: **jobv** – CHARACTER(1)

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

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

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

3: **jobq** – CHARACTER(1)

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

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

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

4: **a**$(lda,:)$ – REAL (KIND=nag_wp) array

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

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

The m by n matrix A.

5: **b**$(ldb,:)$ – REAL (KIND=nag_wp) array

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

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

The p by n matrix B.

5.2 Optional Input Parameters

1: **m** – INTEGER

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

m, the number of rows of the matrix A.

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

2: **n** – INTEGER

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** – INTEGER

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

p, the number of rows of the matrix B.

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

5.3 Output Parameters

1: **k** – INTEGER

2: **l** – INTEGER

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 $\begin{pmatrix} A \\ B \end{pmatrix}$.

3: **a**(lda, :) – REAL (KIND=nag_wp) array

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

The second dimension of the array **a** will be $\max(1, \mathbf{n})$.

Contains the triangular matrix R, or part of R. See Section 3 for details.

4: **b**(ldb, :) – REAL (KIND=nag_wp) array

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

The second dimension of the array **b** will be $\max(1, \mathbf{n})$.

Contains the triangular matrix R if $m - k - l < 0$. See Section 3 for details.

5: **alpha**(**n**) – REAL (KIND=nag_wp) array

See the description of **beta**.

6: **beta**(**n**) – REAL (KIND=nag_wp) array

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

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

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

The notation $\mathbf{alpha}(\mathbf{k} : \mathbf{n})$ above refers to consecutive elements $\mathbf{alpha}(i)$, for $i = \mathbf{k}, \ldots, \mathbf{n}$.

7: **u**(ldu, :) – REAL (KIND=nag_wp) array

The first dimension, ldu, of the array **u** will be

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

The second dimension of the array **u** will be $\max(1, \mathbf{m})$ if **jobu** = 'U' and 1 otherwise.

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

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

8: **v**(ldv, :) – REAL (KIND=nag_wp) array

The first dimension, ldv, of the array **v** will be

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

The second dimension of the array **v** will be $\max(1, \mathbf{p})$ if **jobv** = 'V' and 1 otherwise.

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

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

9: **q**(ldq, :) – REAL (KIND=nag_wp) array

The first dimension, ldq, of the array **q** will be

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

The second dimension of the array **q** will be $\max(1, \mathbf{n})$ if **jobq** = 'Q' and 1 otherwise.

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

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

10: **iwork**(**n**) – INTEGER array

Stores the sorting information. More precisely, the following loop will sort **alpha**

```
for i=k+1:min(m,k+l)
 % swap alpha(i) and alpha(iwork(i))
 end
```

such that **alpha**$(1) \geq$ **alpha**$(2) \geq \cdots \geq$ **alpha**(**n**).

11: **info** – INTEGER

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

6 Error Indicators and Warnings

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 nag_lapack_zggsvd (f08vn).

9 Example

This example finds the generalized singular value decomposition

$$A = U\Sigma_1 \begin{pmatrix} 0 & R \end{pmatrix} Q^{\mathrm{T}}, \quad B = V\Sigma_2 \begin{pmatrix} 0 & R \end{pmatrix} Q^{\mathrm{T}},$$

where

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \\ 4 & 5 & 6 \\ 7 & 8 & 8 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} -2 & -3 & 3 \\ 4 & 6 & 5 \end{pmatrix},$$

together with estimates for the condition number of R and the error bound for the computed generalized singular values.

The example program assumes that $m \geq n$, and would need slight modification if this is not the case.

9.1 Program Text

```
     function f08va_example

fprintf('f08va example results\n\n');

% Generalized SVD of (A, B), where
m = 4;
n = 3;
p = 2;
a = [ 1, 2, 3;
      3, 2, 1;
      4, 5, 6;
      7, 8, 8];
b = [-2, -3, 3;
      4,  6, 5];

% Compute the GSVD of matrix pair A, B
% (a, b) = (U*D1, V*D2)*(0 R)*(Q^T), m>=n
jobu = 'U';
jobv = 'V';
jobq = 'Q';
[k, l, DR, b, alpha, beta, U, V, Q, iwork, info] = ...
  f08va( ...
  jobu, jobv, jobq, a, b);

irank = k + l;
fprintf('\nInfinite generalized singular values = %5d\n', k);
fprintf('  Finite generalized singular values = %5d\n', l);
fprintf('  Numerical rank of (a'' b'')''         = %5d\n', irank);

fprintf('\nFinite generalized singular values\n');
w = alpha(k+1:irank)./beta(k+1:irank);
disp(transpose(w));

fprintf('Orthogonal matrix U\n');
disp(U);

fprintf('Orthogonal matrix V\n');
disp(V);

fprintf('Orthogonal matrix Q\n');
disp(Q);

fprintf('Non singular upper triangular matrix R\n');
R = DR(1:n,n-(irank-1):n);
```

```
disp(R);

% Estimate the reciprocal condition number of R
[rcond, info] = f07tg( ...
        'Inf-norm','Upper','Non-unit', R);

fprintf('Condition number for R:\n%11.1e\n', 1/rcond);

% So long as irank = n, compute error bound for singular values
if (irank==n)
  serrbd = x02aj/rcond;
  fprintf('\nError bound for the generalized singular values\n%11.1e\n', ...
          serrbd);
else
  fprintf('\n(A'' B'')'' is not of full rank\n');
end
```

9.2 Program Results

```
     f08va example results

Infinite generalized singular values =     1
  Finite generalized singular values =     2
  Numerical rank of (a' b')'         =     3

Finite generalized singular values
    1.3151    0.0802

Orthogonal matrix 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

Orthogonal matrix V
    0.3554   -0.9347
    0.9347    0.3554

Orthogonal matrix Q
   -0.8321   -0.0946   -0.5466
    0.5547   -0.1419   -0.8199
         0   -0.9853    0.1706

Non singular upper triangular matrix R
   -2.0569   -9.0121   -9.3705
         0  -10.8822   -7.2688
         0         0   -6.0405

Condition number for R:
    2.4e+01

Error bound for the generalized singular values
    2.6e-15
```