

NAG Toolbox

nag_lapack_dorcscd (f08ra)

1 Purpose

nag_lapack_dorcscd (f08ra) computes the CS decomposition of a real m by m orthogonal matrix X , partitioned into a 2 by 2 array of submatrices.

2 Syntax

```
[x11, x12, x21, x22, theta, u1, u2, v1t, v2t, info] = nag_lapack_dorcscd(m, p, q,
x11, x12, x21, x22, 'jobu1', jobu1, 'jobu2', jobu2, 'jobv1t', jobv1t, 'jobv2t',
jobv2t, 'trans', trans, 'signs', signs)
```

```
[x11, x12, x21, x22, theta, u1, u2, v1t, v2t, info] = f08ra(m, p, q, x11, x12,
x21, x22, 'jobu1', jobu1, 'jobu2', jobu2, 'jobv1t', jobv1t, 'jobv2t', jobv2t,
'trans', trans, 'signs', signs)
```

3 Description

The m by m orthogonal matrix X is partitioned as

$$X = \begin{pmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{pmatrix}$$

where X_{11} is a p by q submatrix and the dimensions of the other submatrices X_{12} , X_{21} and X_{22} are such that X remains m by m .

The CS decomposition of X is $X = U\Sigma_p V^T$ where U , V and Σ_p are m by m matrices, such that

$$U = \begin{pmatrix} U_1 & \mathbf{0} \\ \mathbf{0} & U_2 \end{pmatrix}$$

is an orthogonal matrix containing the p by p orthogonal matrix U_1 and the $(m-p)$ by $(m-p)$ orthogonal matrix U_2 ;

$$V = \begin{pmatrix} V_1 & \mathbf{0} \\ \mathbf{0} & V_2 \end{pmatrix}$$

is an orthogonal matrix containing the q by q orthogonal matrix V_1 and the $(m-q)$ by $(m-q)$ orthogonal matrix V_2 ; and

$$\Sigma_p = \left(\begin{array}{cc|cc} I_{11} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ & C & \mathbf{0} & -S \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & -I_{12} \\ & \mathbf{0} & \mathbf{0} & I_{22} \\ \hline \mathbf{0} & S & & C \\ \mathbf{0} & & I_{21} & \mathbf{0} \end{array} \right)$$

contains the r by r non-negative diagonal submatrices C and S satisfying $C^2 + S^2 = I$, where $r = \min(p, m-p, q, m-q)$ and the top left partition is p by q .

The identity matrix I_{11} is of order $\min(p, q) - r$ and vanishes if $\min(p, q) = r$.

The identity matrix I_{12} is of order $\min(p, m-q) - r$ and vanishes if $\min(p, m-q) = r$.

The identity matrix I_{21} is of order $\min(m-p, q) - r$ and vanishes if $\min(m-p, q) = r$.

The identity matrix I_{22} is of order $\min(m-p, m-q) - r$ and vanishes if $\min(m-p, m-q) = r$.

In each of the four cases $r = p, q, m - p, m - q$ at least two of the identity matrices vanish.

The indicated zeros represent augmentations by additional rows or columns (but not both) to the square diagonal matrices formed by I_{ij} and C or S .

Σ_p does not need to be stored in full; it is sufficient to return only the values θ_i for $i = 1, 2, \dots, r$ where $C_{ii} = \cos(\theta_i)$ and $S_{ii} = \sin(\theta_i)$.

The algorithm used to perform the complete CS decomposition is described fully in Sutton (2009) including discussions of the stability and accuracy of the algorithm.

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 (2012) *Matrix Computations* (4th Edition) Johns Hopkins University Press, Baltimore

Sutton B D (2009) Computing the complete CS decomposition *Numerical Algorithms (Volume 50)* 1017–1398 Springer US 33–65 <http://dx.doi.org/10.1007/s11075-008-9215-6>

5 Parameters

5.1 Compulsory Input Parameters

1: **m** – INTEGER

m , the number of rows and columns in the orthogonal matrix X .

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

2: **p** – INTEGER

p , the number of rows in X_{11} and X_{12} .

Constraint: $0 \leq \mathbf{p} \leq \mathbf{m}$.

3: **q** – INTEGER

q , the number of columns in X_{11} and X_{21} .

Constraint: $0 \leq \mathbf{q} \leq \mathbf{m}$.

4: **x11**($ldx11, :$) – REAL (KIND=nag_wp) array

The first dimension, $ldx11$, of the array **x11** must satisfy

if **trans** = 'T', $ldx11 \geq \max(1, \mathbf{q})$;
otherwise $ldx11 \geq \max(1, \mathbf{p})$.

The second dimension of the array **x11** must be at least $\max(1, \mathbf{p})$ if **trans** = 'T', and at least \mathbf{q} otherwise.

The upper left partition of the orthogonal matrix X whose CSD is desired.

5: **x12**($ldx12, :$) – REAL (KIND=nag_wp) array

The first dimension, $ldx12$, of the array **x12** must satisfy

if **trans** = 'T', $ldx12 \geq \max(1, \mathbf{m} - \mathbf{q})$;
otherwise $ldx12 \geq \max(1, \mathbf{p})$.

The second dimension of the array **x12** must be at least $\max(1, \mathbf{p})$ if **trans** = 'T', and at least $\mathbf{m} - \mathbf{q}$ otherwise.

The upper right partition of the orthogonal matrix X whose CSD is desired.

- 6: **x21**(*ldx21*, :) – REAL (KIND=nag_wp) array

The first dimension, *ldx21*, of the array **x21** must satisfy

if **trans** = 'T', $ldx21 \geq \max(1, \mathbf{q})$;
otherwise $ldx21 \geq \max(1, \mathbf{m} - \mathbf{p})$.

The second dimension of the array **x21** must be at least $\max(1, \mathbf{m} - \mathbf{p})$ if **trans** = 'T', and at least **q** otherwise.

The lower left partition of the orthogonal matrix X whose CSD is desired.

- 7: **x22**(*ldx22*, :) – REAL (KIND=nag_wp) array

The first dimension, *ldx22*, of the array **x22** must satisfy

if **trans** = 'T', $ldx22 \geq \max(1, \mathbf{m} - \mathbf{q})$;
otherwise $ldx22 \geq \max(1, \mathbf{m} - \mathbf{p})$.

The second dimension of the array **x22** must be at least $\max(1, \mathbf{m} - \mathbf{p})$ if **trans** = 'T', and at least $\mathbf{m} - \mathbf{q}$ otherwise.

The lower right partition of the orthogonal matrix X CSD is desired.

5.2 Optional Input Parameters

- 1: **jobu1** – CHARACTER(1)

Default: 'Y'

if **jobu1** = 'Y', U_1 is computed;
otherwise, U_1 is not computed.

- 2: **jobu2** – CHARACTER(1)

Default: 'Y'

if **jobu2** = 'Y', U_2 is computed;
otherwise, U_2 is not computed.

- 3: **jobv1t** – CHARACTER(1)

Default: 'Y'

if **jobv1t** = 'Y', V_1^T is computed;
otherwise, V_1^T is not computed.

- 4: **jobv2t** – CHARACTER(1)

Default: 'Y'

if **jobv2t** = 'Y', V_2^T is computed;
otherwise, V_2^T is not computed.

- 5: **trans** – CHARACTER(1)

Default: 'N'

if **trans** = 'T', X , U_1 , U_2 , V_1^T and V_2^T are stored in row-major order;
otherwise, X , U_1 , U_2 , V_1^T and V_2^T are stored in column-major order.

6: **signs** – CHARACTER(1)

Default: 'D'

if **signs** = 'O', the lower-left block is made nonpositive (the other convention);
otherwise, the upper-right block is made nonpositive (the default convention).

5.3 Output Parameters

1: **x11**(*ldx11*, :) – REAL (KIND=nag_wp) array

The first dimension, *ldx11*, of the array **x11** will be

if **trans** = 'T', $ldx11 = \max(1, \mathbf{q})$;
otherwise $ldx11 = \max(1, \mathbf{p})$.

The second dimension of the array **x11** will be $\max(1, \mathbf{p})$ if **trans** = 'T' and **q** otherwise.

Contains details of the orthogonal matrix used in a simultaneous bidiagonalization process.

2: **x12**(*ldx12*, :) – REAL (KIND=nag_wp) array

The first dimension, *ldx12*, of the array **x12** will be

if **trans** = 'T', $ldx12 = \max(1, \mathbf{m} - \mathbf{q})$;
otherwise $ldx12 = \max(1, \mathbf{p})$.

The second dimension of the array **x12** will be $\max(1, \mathbf{p})$ if **trans** = 'T' and $\mathbf{m} - \mathbf{q}$ otherwise.

Contains details of the orthogonal matrix used in a simultaneous bidiagonalization process.

3: **x21**(*ldx21*, :) – REAL (KIND=nag_wp) array

The first dimension, *ldx21*, of the array **x21** will be

if **trans** = 'T', $ldx21 = \max(1, \mathbf{q})$;
otherwise $ldx21 = \max(1, \mathbf{m} - \mathbf{p})$.

The second dimension of the array **x21** will be $\max(1, \mathbf{m} - \mathbf{p})$ if **trans** = 'T' and **q** otherwise.

Contains details of the orthogonal matrix used in a simultaneous bidiagonalization process.

4: **x22**(*ldx22*, :) – REAL (KIND=nag_wp) array

The first dimension, *ldx22*, of the array **x22** will be

if **trans** = 'T', $ldx22 = \max(1, \mathbf{m} - \mathbf{q})$;
otherwise $ldx22 = \max(1, \mathbf{m} - \mathbf{p})$.

The second dimension of the array **x22** will be $\max(1, \mathbf{m} - \mathbf{p})$ if **trans** = 'T' and $\mathbf{m} - \mathbf{q}$ otherwise.

Contains details of the orthogonal matrix used in a simultaneous bidiagonalization process.

5: **theta**($\min(\mathbf{p}, \mathbf{m} - \mathbf{p}, \mathbf{q}, \mathbf{m} - \mathbf{q})$) – REAL (KIND=nag_wp) array

The values θ_i for $i = 1, 2, \dots, r$ where $r = \min(p, m - p, q, m - q)$. The diagonal submatrices *C* and *S* of Σ_p are constructed from these values as

$$C = \text{diag}(\cos(\mathbf{theta}(1)), \dots, \cos(\mathbf{theta}(r))) \text{ and}$$

$$S = \text{diag}(\sin(\mathbf{theta}(1)), \dots, \sin(\mathbf{theta}(r))).$$

6: **u1**(*ldu1*, :) – REAL (KIND=nag_wp) array

The first dimension of the array **u1** will be if **jobu1** = 'Y', $ldu1 = \max(1, \mathbf{p})$.

The second dimension of the array **u1** will be $\max(1, \mathbf{p})$ if **jobu1** = 'Y' and 1 otherwise.

If **jobu1** = 'Y', **u1** contains the *p* by *p* orthogonal matrix U_1 .

- 7: **u2**(*ldu2*,:) – REAL (KIND=nag_wp) array
 The first dimension of the array **u2** will be if **jobu2** = 'Y', $ldu2 = \max(1, \mathbf{m} - \mathbf{p})$.
 The second dimension of the array **u2** will be $\max(1, \mathbf{m} - \mathbf{p})$ if **jobu2** = 'Y' and 1 otherwise.
 If **jobu2** = 'Y', **u2** contains the $m - p$ by $m - p$ orthogonal matrix U_2 .
- 8: **v1t**(*ldv1t*,:) – REAL (KIND=nag_wp) array
 The first dimension of the array **v1t** will be if **jobv1t** = 'Y', $ldv1t = \max(1, \mathbf{q})$.
 The second dimension of the array **v1t** will be $\max(1, \mathbf{q})$ if **jobv1t** = 'Y' and 1 otherwise.
 If **jobv1t** = 'Y', **v1t** contains the q by q orthogonal matrix V_1^T .
- 9: **v2t**(*ldv2t*,:) – REAL (KIND=nag_wp) array
 The first dimension of the array **v2t** will be if **jobv2t** = 'Y', $ldv2t = \max(1, \mathbf{m} - \mathbf{q})$.
 The second dimension of the array **v2t** will be $\max(1, \mathbf{m} - \mathbf{q})$ if **jobv2t** = 'Y' and 1 otherwise.
 If **jobv2t** = 'Y', **v2t** contains the $m - q$ by $m - q$ orthogonal matrix V_2^T .
- 10: **info** – INTEGER
info = 0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

info < 0

If **info** = $-i$, argument i had an illegal value. An explanatory message is output, and execution of the program is terminated.

info > 0

The Jacobi-type procedure failed to converge during an internal reduction to bidiagonal-block form. The process requires convergence to $\min(\mathbf{p}, \mathbf{m} - \mathbf{p}, \mathbf{q}, \mathbf{m} - \mathbf{q})$ values, the value of **info** gives the number of converged values.

7 Accuracy

The computed CS decomposition is nearly the exact CS decomposition for the nearby matrix $(X + E)$, where

$$\|E\|_2 = O(\epsilon),$$

and ϵ is the *machine precision*.

8 Further Comments

The total number of floating-point operations required to perform the full CS decomposition is approximately $2m^3$.

The complex analogue of this function is `nag_lapack_zuncsd` (f08rn).

9 Example

This example finds the full CS decomposition of

$$X = \begin{pmatrix} -0.13484 & 0.52524 & -0.20924 & 0.81373 \\ 0.67420 & -0.52213 & -0.38886 & 0.34874 \\ 0.26968 & 0.52757 & -0.65782 & -0.46499 \\ 0.67420 & 0.41615 & 0.61014 & 0.00000 \end{pmatrix}$$

partitioned in 2 by 2 blocks.

The decomposition is performed both on submatrices of the orthogonal matrix X and on separated partition matrices. Code is also provided to perform a recombining check if required.

9.1 Program Text

```
function f08ra_example

fprintf('f08ra example results\n\n');

% Find the full CS decomposition of X:
m = nag_int(5);
p = nag_int(3);
q = nag_int(2);
x = [-0.7576  0.3697  0.3838  0.2126 -0.3112;
     -0.4077 -0.1552 -0.1129  0.2676  0.8517;
     -0.0488  0.7240 -0.6730 -0.1301  0.0602;
     -0.2287  0.0088  0.2235 -0.9235  0.2120;
     0.4530  0.5612  0.5806  0.1162  0.3595];

% Partition matrix
x11 = x(1:p, 1:q);   x12 = x(1:p, q+1:m);
x21 = x(p+1:m,1:q); x22 = x(p+1:m,q+1:m);

[x11, x12, x21, x22, theta, u1, u2, v1t, v2t, info] = ...
f08ra( ...
    m, p, q, x11, x12, x21, x22);

disp('Components of CS factorization of X:');
disp('   Theta');
disp(theta);
disp('   U1');
disp(u1);
disp('   U2');
disp(u2);
disp('   V1');
disp(v1t);
disp('   V2');
disp(v2t);
```

9.2 Program Results

```
f08ra example results

Components of CS factorization of X:
   Theta
   0.1811
   0.8255

   U1
-0.8249  -0.3370  -0.4538
-0.2042  -0.5710   0.7952
-0.5271   0.7486   0.4022

   U2
-0.9802  -0.1982
-0.1982   0.9802

   V1
```

0.7461	0.6658	
-0.6658	0.7461	
V2		
-0.3397	0.7738	0.5346
0.8967	0.4379	-0.0640
-0.2837	0.4576	-0.8427
