NAG Toolbox

nag_lapack_dgesdd (f08kd)

1 Purpose

nag_lapack_dgesdd (f08kd) computes the singular value decomposition (SVD) of a real m by n matrix A, optionally computing the left and/or right singular vectors, by using a divide-and-conquer method.

2 Syntax

```
[a, s, u, vt, info] = nag_lapack_dgesdd(jobz, a, 'm', m, 'n', n)
[a, s, u, vt, info] = f08kd(jobz, a, 'm', m, 'n', n)
```

3 Description

The SVD is written as

$$A = U\Sigma V^{\mathrm{T}},$$

where Σ is an m by n matrix which is zero except for its $\min(m,n)$ diagonal elements, U is an m by m orthogonal matrix, and V is an n by n orthogonal matrix. The diagonal elements of Σ are the singular values of A; they are real and non-negative, and are returned in descending order. The first $\min(m,n)$ columns of U and V are the left and right singular vectors of A.

Note that the function returns V^{T}, not V.

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: **jobz** – CHARACTER(1)

Specifies options for computing all or part of the matrix U.

jobz = 'A'
All m columns of U and all n rows of V^{T} are returned in the arrays **u** and **vt**.

jobz = 'S'
The first $\min(m,n)$ columns of U and the first $\min(m,n)$ rows of V^{T} are returned in the arrays **u** and **vt**.

jobz = 'O'
If **m** $\geq$ **n**, the first n columns of U are overwritten on the array **a** and all rows of V^{T} are returned in the array **vt**. Otherwise, all columns of U are returned in the array **u** and the first m rows of V^{T} are overwritten in the array **vt**.

jobz = 'N'
No columns of U or rows of V^{T} are computed.

Constraint: **jobz** = 'A', 'S', 'O' or 'N'.

2: **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.

5.2 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 matrix A.

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

5.3 Output Parameters

1: **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 $\max(1,\mathbf{n})$.

If **jobz** = 'O', **a** is overwritten with the first n columns of U (the left singular vectors, stored column-wise) if $\mathbf{m} \geq \mathbf{n}$; **a** is overwritten with the first m rows of V^{T} (the right singular vectors, stored row-wise) otherwise.

If **jobz** $\neq$ 'O', the contents of **a** are destroyed.

2: **s**$(\min(\mathbf{m},\mathbf{n}))$ – REAL (KIND=nag_wp) array

The singular values of A, sorted so that $\mathbf{s}(i) \geq \mathbf{s}(i+1)$.

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

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

if **jobz** = 'S' or 'A' or **jobz** = 'O' and $\mathbf{m} < \mathbf{n}$, $ldu = \max(1,\mathbf{m})$;
otherwise $ldu = 1$.

The second dimension of the array **u** will be $\max(1,\mathbf{m})$ if **jobz** = 'A' or **jobz** = 'O' and $\mathbf{m} < \mathbf{n}$, $\max(1,\min(\mathbf{m},\mathbf{n}))$ if **jobz** = 'S' and 1 otherwise.

If **jobz** = 'A' or **jobz** = 'O' and $\mathbf{m} < \mathbf{n}$, **u** contains the m by m orthogonal matrix U.

If **jobz** = 'S', **u** contains the first $\min(m,n)$ columns of U (the left singular vectors, stored column-wise).

If **jobz** = 'O' and $\mathbf{m} \geq \mathbf{n}$, or **jobz** = 'N', **u** is not referenced.

4: **vt**$(ldvt,:)$ – REAL (KIND=nag_wp) array

The first dimension, $ldvt$, of the array **vt** will be

if **jobz** = 'A' or **jobz** = 'O' and **m** $\geq$ **n**, $ldvt = \max(1, \mathbf{n})$;
if **jobz** = 'S', $ldvt = \max(1, \min(\mathbf{m}, \mathbf{n}))$;
otherwise $ldvt = 1$.

The second dimension of the array **vt** will be $\max(1, \mathbf{n})$ if **jobz** = 'A' or 'S' or **jobz** = 'O' and **m** $\geq$ **n** and 1 otherwise.

If **jobz** = 'A' or **jobz** = 'O' and **m** $\geq$ **n**, **vt** contains the n by n orthogonal matrix V^{T}.

If **jobz** = 'S', **vt** contains the first $\min(m, n)$ rows of V^{T} (the right singular vectors, stored row-wise).

If **jobz** = 'O' and **m** $<$ **n**, or **jobz** = 'N', **vt** is not referenced.

5: **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: **jobz**, 2: **m**, 3: **n**, 4: **a**, 5: **lda**, 6: **s**, 7: **u**, 8: **ldu**, 9: **vt**, 10: **ldvt**, 11: **work**, 12: **lwork**, 13: **iwork**, 14: **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 > 0

nag_lapack_dgesdd (f08kd) did not converge, the updating process failed.

7 Accuracy

The computed singular value decomposition is nearly the exact singular value decomposition for a nearby matrix $(A + E)$, where

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

and ϵ is the ***machine precision***. In addition, the computed singular vectors are nearly orthogonal to working precision. See Section 4.9 of Anderson *et al.* (1999) for further details.

8 Further Comments

The total number of floating-point operations is approximately proportional to mn^2 when $m > n$ and m^2n otherwise.

The singular values are returned in descending order.

The complex analogue of this function is nag_lapack_zgesvd (f08kp).

9 Example

This example finds the singular values and left and right singular vectors of the 4 by 6 matrix

$$A = \begin{pmatrix} 2.27 & 0.28 & -0.48 & 1.07 & -2.35 & 0.62 \\ -1.54 & -1.67 & -3.09 & 1.22 & 2.93 & -7.39 \\ 1.15 & 0.94 & 0.99 & 0.79 & -1.45 & 1.03 \\ -1.94 & -0.78 & -0.21 & 0.63 & 2.30 & -2.57 \end{pmatrix},$$

together with approximate error bounds for the computed singular values and vectors.

The example program for nag_lapack_dgesvd (f08kb) illustrates finding a singular value decomposition for the case $m \geq n$.

9.1 Program Text

```
function f08kd_example

fprintf('f08kd example results\n\n');

a = [ 2.27,  0.28, -0.48, 1.07, -2.35,  0.62;
     -1.54, -1.67, -3.09, 1.22,  2.93, -7.39;
      1.15,  0.94,  0.99, 0.79, -1.45,  1.03;
     -1.94, -0.78, -0.21, 0.63,  2.30, -2.57];
m = nag_int(size(a,1));
n = nag_int(size(a,2));

jobz = 'Singular vector parts of U and VT';
[~, s, u, vt, info] = f08kd( ...
                             jobz, a);

disp('Singular values of A');
disp(s');
disp('Left singular vectors');
disp(u);
disp('Right singular vectors (by row)');
disp(vt);

% Singular values error bound
serrbd = x02aj*s(1);
% Reciprocal condition numbers for singular vectors
[rcondu, info] = f08fl( ...
         'Left', m, n, s);
[rcondv, info] = f08fl( ...
         'Right', m, n, s);

% Sigular vector error bounds
uerrbd = serrbd./rcondu;
verrbd = serrbd./rcondv;

disp('Error estimate for the singular values');
fprintf('%12.1e\n',serrbd);
disp('Error estimates for the left singular vectors');
fprintf('%12.1e',uerrbd);
fprintf('\n');
disp('Error estimates for the right singular vectors');
fprintf('%12.1e',verrbd);
fprintf('\n');
```

9.2 Program Results

```
f08kd example results

Singular values of A
    9.9966    3.6831    1.3569    0.5000

Left singular vectors
   -0.1921    0.8030   -0.0041    0.5642
    0.8794    0.3926    0.0752   -0.2587
```

```
   -0.2140    0.2980   -0.7827   -0.5027
    0.3795   -0.3351   -0.6178    0.6017

Right singular vectors (by row)
   -0.2774   -0.2020   -0.2918    0.0938    0.4213   -0.7816
    0.6003    0.0301   -0.3348    0.3699   -0.5266   -0.3353
    0.1277   -0.2805   -0.6453   -0.6781   -0.0413    0.1645
   -0.1323   -0.7034   -0.1906    0.5399    0.0575    0.3957

Error estimate for the singular values
     1.1e-15
Error estimates for the left singular vectors
     1.8e-16     4.8e-16     1.3e-15     1.3e-15
Error estimates for the right singular vectors
     1.8e-16     4.8e-16     1.3e-15     2.2e-15
```