NAG Toolbox for MATLAB

f02we

1 Purpose

f02we returns all, or part, of the singular value decomposition of a general real matrix.

2 Syntax

```
[a, b, q, sv, pt, work, ifail] = f02we(a, b, wantq, wantp, 'm', m, 'n',
n, 'ncolb', ncolb)
```

3 Description

The m by n matrix A is factorized as

$$A = QDP^{\mathrm{T}},$$

where

$$D = \begin{pmatrix} S \\ 0 \end{pmatrix}, \quad m > n,$$

$$D = S, \quad m = n,$$

$$D = \begin{pmatrix} S & 0 \end{pmatrix}, \quad m < n,$$

Q is an m by m orthogonal matrix, P is an n by n orthogonal matrix, and S is a $\min(m,n)$ by $\min(m,n)$ diagonal matrix with nonnegative diagonal elements, $sv_1, sv_2, \ldots, sv_{\min(m,n)}$, ordered such that

$$sv_1 \geq sv_2 \geq \cdots \geq sv_{\min(m,n)} \geq 0.$$

The first $\min(m,n)$ columns of Q are the left-hand singular vectors of A, the diagonal elements of S are the singular values of A and the first $\min(m,n)$ columns of P are the right-hand singular vectors of A.

Either or both of the left-hand and right-hand singular vectors of A may be requested and the matrix C given by

$$C = Q^{\mathrm{T}}B,$$

where B is an m by *ncolb* given matrix, may also be requested.

f02we obtains the singular value decomposition by first reducing A to upper triangular form by means of Householder transformations, from the left when $m \geq n$ and from the right when $m < n$. The upper triangular form is then reduced to bidiagonal form by Givens plane rotations and finally the QR algorithm is used to obtain the singular value decomposition of the bidiagonal form.

Good background descriptions to the singular value decomposition are given in Dongarra *et al.* 1979, Hammarling 1985 and Wilkinson 1978. Note that this function is not based on the LINPACK function SSVDC/DSVDC.

Note that if K is any orthogonal diagonal matrix so that

$$KK^{\mathrm{T}} = I$$

(so that K has elements $+1$ or -1 on the diagonal), then

$$A = (QK)D(PK)^{\mathrm{T}}$$

is also a singular value decomposition of A.

4 References

Dongarra J J, Moler C B, Bunch J R and Stewart G W 1979 *LINPACK Users' Guide* SIAM, Philadelphia

Hammarling S 1985 The singular value decomposition in multivariate statistics *SIGNUM Newsl.* **20 (3)** 2–25

Wilkinson J H 1978 Singular Value Decomposition – Basic Aspects *Numerical Software – Needs and Availability* (ed D A H Jacobs) Academic Press

5 Parameters

5.1 Compulsory Input 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 $\max(1, \mathbf{n})$

The leading m by n part of the array **a** must contain the matrix A whose singular value decomposition is required.

2: **b(ldb,$*$) – double array**

The first dimension, **ldb**, of the array **b** must satisfy

if **ncolb** > 0, **ldb** $\geq \max(1, \mathbf{m})$;
ldb ≥ 1 otherwise.

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

If **ncolb** > 0, the leading m by $ncolb$ part of the array **b** must contain the matrix to be transformed.

3: **wantq – logical scalar**

Must be **true**, if the left-hand singular vectors are required.

If **wantq** $=$ **false**, the array **q** is not referenced.

4: **wantp – logical scalar**

Must be **true** if the right-hand singular vectors are required.

If **wantp** $=$ **false**, the array **pt** is not referenced.

5.2 Optional Input Parameters

1: **m – int32 scalar**

m, the number of rows of the matrix A.

Constraint: **m** ≥ 0.

If **m** $= 0$, an immediate return is effected

2: **n – int32 scalar**

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

n, the number of columns of the matrix A.

Constraint: **n** ≥ 0.

If **n** $= 0$, an immediate return is effected

3: **ncolb – int32 scalar**

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

ncolb, the number of columns of the matrix B.

If **ncolb** $= 0$, the array **b** is not referenced.

Constraint: **ncolb** ≥ 0.

5.3 Input Parameters Omitted from the MATLAB Interface

lda, ldb, ldq, ldpt

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$(1, \mathbf{n})$

If $\mathbf{m} \geq \mathbf{n}$ and **wantq** $=$ **true**, the leading m by n part of **a** will contain the first n columns of the orthogonal matrix Q.

If $\mathbf{m} < \mathbf{n}$ and **wantp** $=$ **true**, the leading m by n part of **a** will contain the first m rows of the orthogonal matrix P^{T}.

If $\mathbf{m} \geq \mathbf{n}$ and **wantq** $=$ **false** and **wantp** $=$ **true**, the leading n by n part of **a** will contain the first n rows of the orthogonal matrix P^{T}.

Otherwise the leading m by n part of **a** is used as internal workspace.

2: **b(ldb,$*$) – double array**

The first dimension, **ldb**, of the array **b** must satisfy

if **ncolb** > 0, **ldb** $\geq$ max$(1, \mathbf{m})$;
ldb ≥ 1 otherwise.

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

Contains the m by *ncolb* matrix $\mathbf{q}^{\mathrm{T}}\mathbf{b}$.

3: **q(ldq,$*$) – double array**

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

if $\mathbf{m} < \mathbf{n}$ and **wantq** $=$ **true**, **ldq** $\geq$ max$(1, \mathbf{m})$;
ldq ≥ 1 otherwise.

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

If $\mathbf{m} < \mathbf{n}$ and **wantq** $=$ **true**, the leading m by m part of the array **q** will contain the orthogonal matrix Q. Otherwise the array **q** is not referenced.

4: **sv($*$) – double array**

Note: the dimension of the array **sv** must be at least min$(\mathbf{m}, \mathbf{n})$.

The min(m, n) diagonal elements of the matrix S.

5: **pt(ldpt,$*$) – double array**

The first dimension, **ldpt**, of the array **pt** must satisfy

if $\mathbf{m} \geq \mathbf{n}$ and **wantq** $=$ **true** and **wantp** $=$ **true**, **ldpt** $\geq$ max$(1, \mathbf{n})$;
ldpt ≥ 1 otherwise.

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

If $\mathbf{m} \geq \mathbf{n}$ and **wantq** and **wantp** are **true**, the leading n by n part of the array **pt** will contain the orthogonal matrix P^{T}. Otherwise the array **pt** is not referenced.

6: **work**($*$) – **double array**

Note: the dimension of the array **work** must be at least $\max(1, lwork)$, where $lwork$ must be as given as follows:

$\mathbf{m} \geq \mathbf{n}$

wantq = **true** and **wantp** = **true**

$$lwork = \max\left(\mathbf{n}^2 + 5 \times (\mathbf{n} - 1), \mathbf{n} + \mathbf{ncolb}, 4\right)$$

wantq = **true** and **wantp** = **false**

$$lwork = \max\left(\mathbf{n}^2, \mathbf{ncolb}\right) + \max(4 \times (\mathbf{n} - 1), 5) + 1$$

wantq = **false** and **wantp** = **true**

$$lwork = \max(3 \times (\mathbf{n} - 1), 2) \quad \text{when } \mathbf{ncolb} = 0$$
$$lwork = \max(5 \times (\mathbf{n} - 1), 2) \quad \text{when } \mathbf{ncolb} > 0$$

wantq = **false** and **wantp** = **false**

$$lwork = \max(2 \times (\mathbf{n} - 1), 2) \quad \text{when } \mathbf{ncolb} = 0$$
$$lwork = \max(3 \times (\mathbf{n} - 1), 2) \quad \text{when } \mathbf{ncolb} > 0$$

$\mathbf{m} < \mathbf{n}$

wantq = **true** and **wantp** = **true**

$$lwork = \max\left(\mathbf{m}^2 + 5 \times (\mathbf{m} - 1), 2\right)$$

wantq = **true** and **wantp** = **false**

$$lwork = \max(3 \times (\mathbf{m} - 1), 1)$$

wantq = **false** and **wantp** = **true**

$$lwork = \max\left(\mathbf{m}^2 + 3 \times (\mathbf{m} - 1), 2\right) \quad \text{when } \mathbf{ncolb} = 0$$
$$lwork = \max\left(\mathbf{m}^2 + 5 \times (\mathbf{m} - 1), 2\right) \quad \text{when } \mathbf{ncolb} > 0$$

wantq = **false** and **wantp** = **false**

$$lwork = \max(2 \times (\mathbf{m} - 1), 1) \quad \text{when } \mathbf{ncolb} = 0$$
$$lwork = \max(3 \times (\mathbf{m} - 1), 1) \quad \text{when } \mathbf{ncolb} > 0$$

work$(\min(\mathbf{m}, \mathbf{n}))$ contains the total number of iterations taken by the QR algorithm.

The rest of the array is used as workspace.

7: **ifail** – **int32 scalar**

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

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= -1$

One or more of the following conditions hold:

$\mathbf{m} < 0$;
$\mathbf{n} < 0$;
$\mathbf{lda} < \mathbf{m}$;
$\mathbf{ncolb} < 0$;
$\mathbf{ldb} < \mathbf{m}$ and $\mathbf{ncolb} > 0$;
$\mathbf{ldq} < \mathbf{m}$ and $\mathbf{m} < \mathbf{n}$ and **wantq** = **true**;
$\mathbf{ldpt} < \mathbf{n}$ and $\mathbf{m} \geq \mathbf{n}$ and **wantq** = **true**, and **wantp** = **true**.

ifail > 0

The QR algorithm has failed to converge in $50 \times \min(m, n)$ iterations. In this case $\mathbf{sv}(1), \mathbf{sv}(2), \ldots, \mathbf{sv}(\mathbf{ifail})$ may not have been found correctly and the remaining singular values may not be the smallest. The matrix A will nevertheless have been factorized as $A = QEP^{\mathrm{T}}$, where the leading $\min(m, n)$ by $\min(m, n)$ part of E is a bidiagonal matrix with $\mathbf{sv}(1), \mathbf{sv}(2), \ldots, \mathbf{sv}(\min(m, n))$ as the diagonal elements and $\mathbf{work}(1), \mathbf{work}(2), \ldots, \mathbf{work}(\min(m, n) - 1)$ as the superdiagonal elements.

This failure is not likely to occur.

7 Accuracy

The computed factors Q, D and P satisfy the relation

$$QDP^{\mathrm{T}} = A + E,$$

where

$$\|E\| \leq c\epsilon\|A\|,$$

ϵ is the ***machine precision***, c is a modest function of m and n and $\|.\|$ denotes the spectral (two) norm. Note that $\|A\| = sv_1$.

8 Further Comments

>Following the use of f02we the rank of A may be estimated as follows:

```
tol = eps;
irank = 1;
while (irank <= numel(sv) && sv(irank) >= tol*sv(1) )
  irank = irank + 1;
end
```

returns the value k in `irank`, where k is the smallest integer for which $\mathbf{sv}(k) < tol \times \mathbf{sv}(1)$, where tol is typically the machine precision, so that `irank` is an estimate of the rank of S and thus also of A.

9 Example

```
a = [2, 2.5, 2.5;
     2, 2.5, 2.5;
     1.6, -0.4, 2.8;
     2, -0.5, 0.5;
     1.2, -0.3, -2.9];
b = [1.1; 0.9; 0.6; 0; -0.8];
wantq = true;
wantp = true;
[aOut, bOut, q, sv, pt, work, ifail] = f02we(a, b, wantq, wantp)
```

```
aOut =
    0.6011   -0.1961   -0.3165
    0.6011   -0.1961   -0.3165
    0.4166    0.1569    0.6941
    0.1688   -0.3922    0.5636
   -0.2742   -0.8629    0.0139
bOut =
    1.6716
    0.3922
   -0.2276
   -0.1000
   -0.1000
q =
     0     0     0     0     0
sv =
```

```
    6.5616
    3.0000
    2.4384
pt =
    0.4694    0.4324    0.7699
   -0.7845   -0.1961    0.5883
    0.4054   -0.8801    0.2471
work =
    0.0000
    0.0000
    5.0000
    0.5883
   -0.1961
   -0.7845
   -0.2471
    0.8801
   -0.4054
    0.0000
    0.0000
    5.0000
   -0.3000
    0.3000
    0.8000
    1.0000
    1.0000
    0.0008
    0.0000
ifail =
           0
```