svd

Singular value decomposition

Syntax

```
s = svd(A)
```
example

```
[U,S,V] = svd(A)
```
example
```
[U,S,V] = svd(A,'econ')
```
example
```
[U,S,V] = svd(A,0)
```
example

Description

`s = svd(A)` returns the singular values of matrix `A` in descending order. example

`[U,S,V] = svd(A)` performs a singular value decomposition of matrix `A`, such that `A = U*S*V'`. example

`[U,S,V] = svd(A,'econ')` produces an economy-size decomposition of `m`-by-`n` matrix `A`: example

- `m > n` — Only the first `n` columns of `U` are computed, and `S` is `n`-by-`n`.
- `m = n` — `svd(A,'econ')` is equivalent to `svd(A)`.
- `m < n` — Only the first `m` columns of `V` are computed, and `S` is `m`-by-`m`.

`[U,S,V] = svd(A,0)` produces a different economy-size decomposition of `m`-by-`n` matrix `A`: example

- `m > n` — `svd(A,0)` is equivalent to `svd(A,'econ')`.
- `m <= n` — `svd(A,0)` is equivalent to `svd(A)`.

Examples

collapse all

Singular Values of Matrix

Compute the singular values of a full rank matrix.

Open This Example

```
A = [1 0 1; -1 -2 0; 0 1 -1]
```

```
A =

     1     0     1
    -1    -2     0
     0     1    -1
```

```
s = svd(A)
```

```
s =

    2.4605
    1.6996
    0.2391
```

Singular Value Decomposition

Find the singular value decomposition of a rectangular matrix A.

Open This Example

```
A = [1 2; 3 4; 5 6; 7 8]
```

```
A =

     1     2
     3     4
     5     6
     7     8
```

```
[U,S,V] = svd(A)
```

```
U =

   -0.1525   -0.8226   -0.3945   -0.3800
   -0.3499   -0.4214    0.2428    0.8007
   -0.5474   -0.0201    0.6979   -0.4614
   -0.7448    0.3812   -0.5462    0.0407


S =

   14.2691         0
         0    0.6268
         0         0
         0         0


V =

   -0.6414    0.7672
   -0.7672   -0.6414
```

Confirm the relation `A = U*S*V'`, within machine precision.

```
U*S*V'
```

```
ans =

    1.0000    2.0000
    3.0000    4.0000
    5.0000    6.0000
    7.0000    8.0000
```

Economy-Size Decomposition

Find the economy-size decomposition of a rectangular matrix.

Open This Example

```
A = [1 2; 3 4; 5 6; 7 8]
```

```
A =

     1     2
     3     4
     5     6
     7     8
```

```
[U,S,V] = svd(A,0)
```

```
U =

   -0.1525   -0.8226
   -0.3499   -0.4214
   -0.5474   -0.0201
   -0.7448    0.3812


S =

   14.2691         0
         0    0.6268


V =

   -0.6414    0.7672
   -0.7672   -0.6414
```

Since `A` is 4-by-2, `svd(A,'econ')` produces the same results.

Rank, Column Space, and Null Space of Matrix

Use the results of the singular value decomposition to determine the rank, column space, and null space of a matrix.

Open This Example

```
A = [2 0 2; 0 1 0; 0 0 0]
```

```
A =

     2     0     2
     0     1     0
     0     0     0
```

```
[U,S,V] = svd(A)
```

```
U =

     1     0     0
     0     1     0
     0     0     1


S =

    2.8284         0         0
         0    1.0000         0
         0         0         0


V =

    0.7071         0   -0.7071
         0    1.0000         0
    0.7071         0    0.7071
```

Calculate the rank using the number of nonzero singular values.

```
s = diag(S);
rank_A = nnz(s)
```

```
rank_A =

     2
```

Compute an orthonormal basis for the column space of `A` using the columns of `U` that correspond to nonzero singular values.

```
column_basis = U(:,logical(s))
```

```
column_basis =

     1     0
     0     1
     0     0
```

Compute an orthonormal basis for the null space of `A` using the columns of `V` that correspond to singular values equal to zero.

```
null_basis = V(:,~s)
```

```
null_basis =

   -0.7071
         0
    0.7071
```

The functions `rank`, `orth`, and `null` provide convenient ways to calculate these quantities.

Input Arguments

collapse all

A — Input matrix
matrix

Input matrix. `A` can be either square or rectangular in size.

Data Types: `single` | `double`
Complex Number Support: Yes

Output Arguments

collapse all

s — Singular values
column vector

Singular values, returned as a column vector. The singular values are nonnegative real numbers listed in decreasing order.

U — Left singular vectors
matrix

Left singular vectors, returned as the columns of a matrix.

- For an `m`-by-`n` matrix `A` with `m > n`, the economy-sized decompositions `svd(A,'econ')` and `svd(A,0)` compute only the first `n` columns of `U`. In this case, the columns of `U` are orthogonal and `U` is an `m`-by-`n` matrix that satisfies $U^H U = I_n$.
- For full decompositions, `svd(A)` returns `U` as an `m`-by-`m` unitary matrix satisfying $U U^H = U^H U = I_m$. The columns of `U` that correspond to nonzero singular values form a set of orthonormal basis vectors for the range of `A`.

S — Singular values
diagonal matrix

Singular values, returned as a diagonal matrix. The diagonal elements of `S` are nonnegative singular values in decreasing order. The size of `S` is as follows:

- For an `m`-by-`n` matrix `A`, the economy-sized decomposition `svd(A,'econ')` returns `S` as a square matrix of order `min([m,n])`.
- For full decompositions, `svd(A)` returns `S` with the same size as `A`.
- If `m > n`, then `svd(A,0)` returns `S` as a square matrix of order `min([m,n])`.
- If `m < n`, then `svd(A,0)` returns `S` with the same size as `A`.

V — Right singular vectors
matrix

Right singular vectors, returned as the columns of a matrix.

- For an `m`-by-`n` matrix `A` with `m < n`, the economy decomposition `svd(A,'econ')` computes only the first `m` columns of `V`. In this case, the columns of `V` are orthogonal and `V` is an `n`-by-`m` matrix that satisfies $V^H V = I_m$.

- For full decompositions, `svd(A)` returns `V` as an `n`-by-`n` unitary matrix satisfying $VV^H = V^HV = I_n$. The columns of `V` that do *not* correspond to nonzero singular values form a set of orthonormal basis vectors for the null space of `A`.

More About

- Singular Values

See Also

`gsvd` | `null` | `orth` | `rank` | `svds`

Introduced before R2006a