Matrices

MIEET 1º ano

Peter Stallinga UAlg 2011

Matrix

MatLab/Octave is based on matrices.
All operations are done on an entire matrix

$$A = \begin{pmatrix} 1 & 2 & -1 \\ 0 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

How to multiply it by 2?

PASCAL & C

```
for i := 1 to 3
  for j := 1 to 3
    B[i, j] = 2*A[i, j];

for (i=0; i<3; i++) {
  for (j=0; j<3; j++) {
    B[i,j] = 2*A[i,j];
}}
```

Octave

```
B = 2*A
```

Matrix (vector) definitions

Note: A vector is a one-dimensional matrix. An array with one index.

Operations:

```
a = [1, 2, 3]
```

$\begin{pmatrix} 1 & 2 & 3 \end{pmatrix}$

separate elements in row (*)

```
a = [1, 2, 3; 4, 5, 6; 7, 8, 9]
```

or

```
a = [1, 2, 3;
     4, 5, 6;
     7, 8, 9]
```

$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$

separate rows

```
a = [1; 2; 3]
```

or

```
a = [1;
     2;
     3]
```

$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$

*: Note: comma ',' can also be omitted – `[1 2 3]` is equal to `[1, 2, 3]` – but for readability it is better to keep it (in other languages like Mathematica, a space means multiplication)

Scalar - Vector - Matrix - Tensor - Array

All are tensors (when dealing with numbers):
scalar: zeroth order tensor. single value, no index
vector: first order tensor. 1 index per element
matrix: second order tensor. 2 indexes per element
tensor: nth order. n indexes per element

In **programming**: all tensors except scalars are called arrays

In **MatLab**: all (including scalar!) are called array / matrix

Matrix multiplication, matrix vs. array

$$A=\begin{pmatrix}1 & 2 & -1\\ 0 & 2 & 1\\ 1 & 1 & 1\end{pmatrix}$$

How to take the square?
B = A*A, B = A^2

$$B=A^2=\begin{pmatrix}1 & 2 & -1\\ 0 & 2 & 1\\ 1 & 1 & 1\end{pmatrix}^2=A*A=\begin{pmatrix}1 & 2 & -1\\ 0 & 2 & 1\\ 1 & 1 & 1\end{pmatrix}*\begin{pmatrix}1 & 2 & -1\\ 0 & 2 & 1\\ 1 & 1 & 1\end{pmatrix}=\begin{pmatrix}0 & 5 & 0\\ 1 & 5 & 3\\ 2 & 5 & 1\end{pmatrix}$$

1x1 + 2x0 -1x1 = 0, etc.

$$B=A^2=\begin{pmatrix}1^2 & 2^2 & (-1)^2\\ 0^2 & 2^2 & 1^2\\ 1^2 & 1^2 & 1^2\end{pmatrix}=\begin{pmatrix}1 & 4 & 1\\ 0 & 4 & 1\\ 1 & 1 & 1\end{pmatrix}$$

?

The crucial point

$$A = \begin{pmatrix} 1 & 2 & -1 \\ 0 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

How to take the square?
B = A*A, B = A^2

$$\begin{pmatrix} 1 & 2 & -1 \\ 0 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix} * \begin{pmatrix} 1 & 2 & -1 \\ 0 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 5 & 0 \\ 1 & 5 & 3 \\ 2 & 5 & 1 \end{pmatrix}$$

Octave (matrix)

```
B = A*A      B = A^2
```

$$\begin{pmatrix} 1^2 & 2^2 & (-1)^2 \\ 0^2 & 2^2 & 1^2 \\ 1^2 & 1^2 & 1^2 \end{pmatrix} = \begin{pmatrix} 1 & 4 & 1 \\ 0 & 4 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Octave (array)

```
B = A.*A     B = A.^2
```

Matrix and array multiplication

$$\overset{\leftrightarrow n}{} \; m\updownarrow\begin{pmatrix} 1 & 2 & -1 \\ 0 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix} * \overset{\leftrightarrow k}{}\; \begin{pmatrix} 1 & 0 & 1 \\ -1 & 1 & 0 \\ 1 & -1 & 1 \end{pmatrix} n\updownarrow = \overset{\leftrightarrow k}{}\; \begin{pmatrix} -2 & 3 & 0 \\ -1 & 1 & 1 \\ 1 & 0 & 2 \end{pmatrix} \updownarrow m$$

Octave (matrix)

```
C = A*B
```

Width of A has to be equal to **height of B**
C will have width *k* and height *m*

$$\begin{pmatrix} 1 & 2 & -1 \\ 0 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix} .* \begin{pmatrix} 1 & 0 & 1 \\ -1 & 1 & 0 \\ 1 & -1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 2 & 0 \\ 1 & -1 & 1 \end{pmatrix}$$

Octave (array)

```
C = A.*B
```

Matrices have to be of same size in **all** dimensions

The crucial point

Multiplying with scalars: the '.' makes no difference!
(Not very consistent, isn't it?)
With scalars the '.' is always there

$$2*\begin{pmatrix}1 & 2 & -1\\ 0 & 2 & 1\\ 1 & 1 & 1\end{pmatrix}=\begin{pmatrix}2 & 4 & -2\\ 0 & 4 & 2\\ 2 & 2 & 2\end{pmatrix}$$

Octave (matrix)

```
B = a*A
```

$$2.*\begin{pmatrix}1 & 2 & -1\\ 0 & 2 & 1\\ 1 & 1 & 1\end{pmatrix}=\begin{pmatrix}2 & 4 & -2\\ 0 & 4 & 2\\ 2 & 2 & 2\end{pmatrix}$$

Octave (array)

```
B = a.*A
```

Functions on matrices

Using functions: always element by element

$$\sin\begin{pmatrix} -\pi/2 & 0 & \pi/2 \\ 0 & \pi/2 & \pi \\ 0 & -2\pi & \pi/2 \end{pmatrix} = \begin{pmatrix} -1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Octave

```
B = sin(A)
```

$$\begin{pmatrix} -2 & 0 & 2 \\ 0 & 1 & 1 \\ 0 & -2 & 2 \end{pmatrix} < 1 = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

Octave

```
B = A<1
```

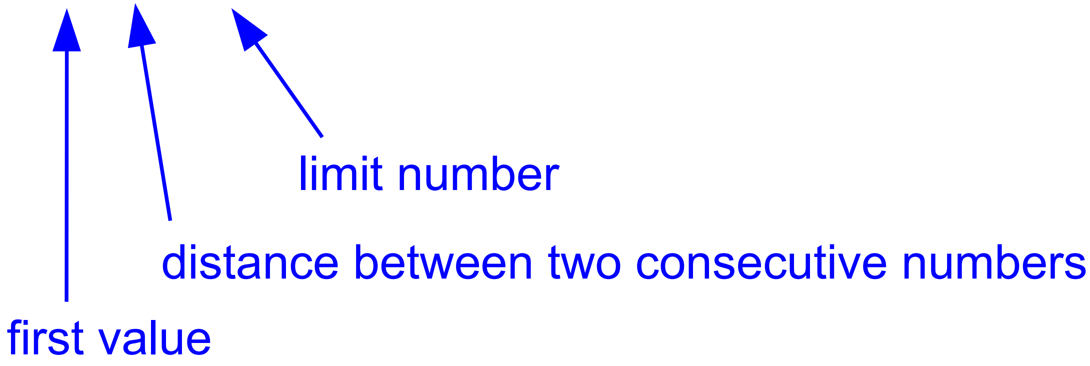
*: '1' means 'true'

Ranges

Ranges can be rapidly defined using the colon, ':'

```
a = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
a = 1:10

a = [1, 3, 5, 7, 9]
a = 1:2:10
```



Access to the values

An element of a matrix can be accessed by its indexes
For example (2 dimensional array a):
a(row, column)

```
Octave:1> a = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];
Octave:2> a(3)
ans = 3
Octave:3> a(1,6)
ans = 6
Octave:1> a = [1; 2; 3; 4; 5; 6; 7; 8; 9; 10];
Octave:2> a(4)
ans = 4
Octave:3> a(7,1)
ans = 7
```

A vector is considered (also) a matrix with 1 line or 1 column
(Not very consistent, isn't it?)

Access to the values

An entire column or row

a(:, column)

a(row, :)

```
Octave:1> a = [1, 2, 3;
               4, 5, 6;
               7, 8, 9];
Octave:2> a(2,:)
ans =
   4  5  6
Octave:3> a(:,1)
ans =
   1
   4
   7
```

Interesting ....

`end` gives last column/row index. Ex.: `a(end, end)` yields `9`

`eye(n)` : unity matrix *n* x *n*
`zeros(m, n)` : matrix *m* x *n* filled with zeros
`ones(m, n)` : matrix *m* x *n* filled with ones
`rand(m, n)` : matrix *m* x *n* filled with random values 0..1

`linspace(start,stop,number)` : vector with **`number`** equally spaced elements from **`start`** to **`stop`**
(equal to **`start:(stop-start)/(number-1):stop`**)

`inv(A)` : inverse matrix
`A'` : transposed matrix

Interesting ....

`max`: largest element of vector. `max(max(A))` is maximum of array

`min`: smallest element of vector. `min(min(A))` is minimum of array

`mean`: average of elements of vector

`std`: standard deviation of elements of vector

`sort`: sorts vector small $\rightarrow$ big

`sum`: sum of elements of vector

`prod`: product of elements of vector

`diff`: difference between consecutive elements