Lecture 4

In last lecture, we looked at some basic MATLAB syntaxes

Vector or array

```
>> x1 = 0

x1 =

     0

>> x2 = 0 ;
```

→ suppress output

```
>> x1 = 1 , x2 = 10 ;
```

→ suppress output

Combine two variable declaration into one using comma

```
>> x1 = 1 , x2 = 10

x1 =

     1

x2 =

     10

>> a = 1 : 10
```

→ to create vector between 1 & 10 with spacing 1

```
a =

     1     2     3     4     5     6     7     8     9    10

>> a = 1 : 2 : 10
```

→ to create vector between 1 & 10 with spacing 2

```
a =

     1     3     5     7     9
```

```
>> a = 0.5 : (-0.1) : 0.1
a =
    0.5000    0.4000    0.3000    0.2000    0.1000
```

← You can have sequence of decreasing number too

```
>> a = x1 : x2
a =
     1     2     3     4    ...    10
```

↳ use variables to create vectors (recall that we have set x1=1, x2=10)

```
>> x1, x2
x1 =
     1
x2 =
    10
```

← check

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

same effect

```
>> size(a)
ans =
     1     3
```

← compute size of vectors or matrix

← row vectors has just 1 row

1 → # row

3 → # columns

```
>> a = [1 2 3 4]
a =
```

tell MATLAB to create new row

1
2
3
4

```
>> size(a)

ans =

     4     1
```

← column vector has just one column

```
>> b = 1:4

b =

     1     2     3     4
```

```
>> a = b'
```

← stands for transpose

row vectors becomes column vectors
column ——"—— row vector

```
a =

     1
     2
     3
     4
```

- Array / matrix

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

A =

     1     2     3
     4     5     6
     7     8     9
```

→ "," for next element in row

";" for next row

```
>> size(A)

ans =

     3     3
```

```
>> A = [1:3; 4:6; 7:9]

A =

     1     2     3
     4     5     6
     7     8     9
```

- **Operations on vectors / matrix**

`+`, `-` are defined following usual principle
↓
elementwise operations

`*`, `/`, `^` — are defined only if these operations are valid between two variables

e.g. if a & b are row vectors

↳ `a*b`, `a/b`, `a^2` are not valid operations

↳ `a.*b`, `a./b`, `a.^2` are *, /, ^ operations applied to each element

Similarly

if a & b are column vectors

↳ `a*b`, `a/b`, `a^2` are not valid operations

↳ `a.*b`, `a./b`, `a.^2` elementwise operations defined.

General rule

(i) treat row vector as matrix with just one row

(ii) treat column ——//—— one column

(iii) Let A be a matrix of size $n \times m$

B ——//—— $l \times k$

Then

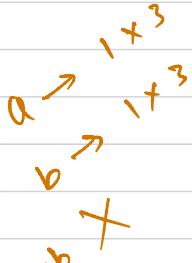
A * B is defined only if

$m = l$

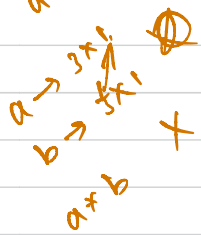
size(A*B) $= n \times k$

a → 1×3, b → 3×1, a*b is valid

(# columns of A should be equal to # rows of B)



With this rule, you can see multiplication of two rows or two columns is invalid



Example

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, \quad B = \begin{bmatrix} 0.1 & 0.2 & 0.3 \\ 0.4 & 0.5 & 0.6 \\ 0.7 & 0.8 & 0.9 \end{bmatrix}$$

Then

$$A * B = \begin{bmatrix} [1,2,3] \times \begin{bmatrix} 0.1 \\ 0.4 \\ 0.7 \end{bmatrix} & [1,2,3] \times \begin{bmatrix} 0.2 \\ 0.5 \\ 0.8 \end{bmatrix} & [1,2,3] \times \begin{bmatrix} 0.3 \\ 0.6 \\ 0.9 \end{bmatrix} \\ [4,5,6] \times \begin{bmatrix} 0.1 \\ 0.4 \\ 0.7 \end{bmatrix} & [4,5,6] \times \begin{bmatrix} 0.2 \\ 0.5 \\ 0.8 \end{bmatrix} & [4,5,6] \times \begin{bmatrix} 0.3 \\ 0.6 \\ 0.9 \end{bmatrix} \\ [7,8,9] \times \begin{bmatrix} 0.1 \\ 0.4 \\ 0.7 \end{bmatrix} & [7,8,9] \times \begin{bmatrix} 0.2 \\ 0.5 \\ 0.8 \end{bmatrix} & [7,8,9] \times \begin{bmatrix} 0.3 \\ 0.6 \\ 0.9 \end{bmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} 3.0 & 3.6 & 4.2 \\ 6.6 & 8.1 & - \\ - & - & - \end{bmatrix}_{3\times 3}$$

↳ From our general rule

- multiplication of row and column vectors
- multiplication of column & row vectors

are valid.

e.g $a = [1, 2, 3]$, $b = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$

Then (i) $a * b = 14$ ← scalar

(ii) $b * a = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \end{bmatrix}$ ← matrix

1×3 ← a
3×1 ← b
1×1 ← a*b

3×1 ← b
1×3 ← a
3×3 ← b*a

size(a), size(b)

In today's lecture

- plotting
- script file
- function file

Plotting:

```
>> t = 0 : 0.1 : 10 ;
```

← "radian"

$t = [0, 0.1, 0.2, \ldots, 10]$

```
>> yt = sin(t) ;
```

$\sin(t) = [\sin(0), \sin(0.1), \sin(0.2), \ldots, \sin(10)]$

cos( ), tan, tanh

```
>> plot(t, yt)
```

t: x points

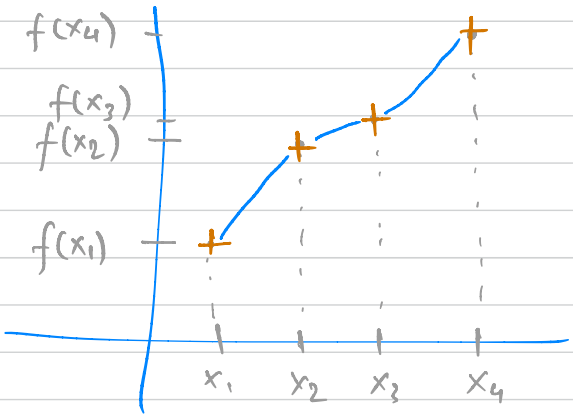
yt: function values at x points

Documentation

>> help plot

help sin

help size



Script file :

```
>> t = 0 : 0.1 : 10 ;      ← row vector
>> yt = sin(t) ;           ← row vector
>> zt = yt.^2 ;            ← row vector

>> plot(t, yt, 'r+:')      ← dotted line
>> hold on                 ← I want to add more plots to the same figure
>> plot(t, zt, 'g*-')      ← solid line
```

→ create a demoPlot.m file

demoPlot.m

```
t = 0:0.1:10;
yt = sin(t);
zt = yt.^2;
plot(t, yt, 'r+:')
hold on
plot(t, zt, 'g*-')
```

20

>> demoPlot ↵

↳ it will put variables in your workspace

↓↓

script files