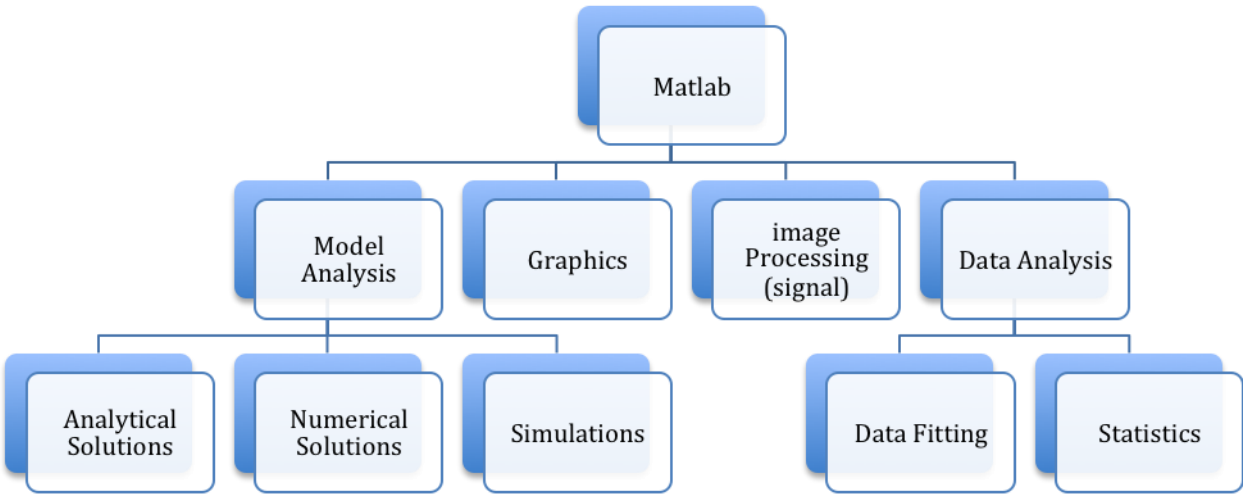
Matlab
The language of technical computing

- Matlab is basically a high level language that has many specialized toolboxes for making things easier.
- Matlab means **MATrix LABoratory**
- What is Matlab used for?



- Command window:
  It is used for writing commands and operations.

  - Commands: (examples)

    >> clc : Clear the command window.
    >> clear : clear the workspace from variables.
    >> who : displays all variables in workspace.

- Workspace Window:
  This window shows all variables written in the command window or from running codes.

- Command History Window:
  - It shows all previous commands
  - Double click on any command and it will be executed.

- Current Directory Window:
  - It shows all files stored in the current directory of Matlab, which may be in C partition.

- Help Window:
  How to call **help** in Matlab?

  >> help : opens general help for Matlab.
  >> help +space +command : shows how to use this command
  >> lookfor+string expressing the topic you want a help in. In this case, you don't know the command.

Ex. >>lookfor inverse

❖ Types of Matlab Variables:

1) Integers (1, 2, 3, … etc.)
2) Real (5.36, 7.2, … etc.)
3) Complex (4+2i, … etc.)
4) Vector ([1 5 7 3], … etc.)
5) Matrix ([1 5 7; 2 3 4], ... etc.)

$$\begin{bmatrix} 1 & 5 & 7 \\ 2 & 3 & 4 \end{bmatrix}$$

6) String ('Mohamed', … etc.)

- Variables should start with **letter**
- Variables must have no **special character**
- Variables must NOT be a **special word**
- Variables can be named using maximum of 32 characters.

❖ Special Words: (examples)

- ans

- pi ($\pi = 3.14 = \frac{22}{7}$)
- eps ($\epsilon = 2.2204 * 10^{-16}$, very tiny number)
- inf (∞, infinity) = $\frac{x}{0}$
- nan (undefined value) = $\frac{0}{0}$

❖ Trigonometric Functions:

- sin(X), cos(X), tan(X), sec(X), csc(X), cot(X).
  Note: the angle "X" should be in **radians**.
- sind(X), cosd(X), …
  Here the angle should be in **degrees**.
- To get inverse functions: asin(…) or asind(…)

❖ Exponential functions:

>> exp(x)	e^X
>> log(X)	$\ln(X)$
>> log10(X)	$\log_{10}(X)$
>> sqrt(X)	$\sqrt{X}$
>> nthroot(X, n)	$\sqrt[n]{X}$

❖ How to define a vector?

1) X=1:10 (here, the step is 1)
   i.e., X = 1,2,3,4,5,6,7,8,9,10

2) X=1:2:10 (here, the step is 2)
   i.e., X = 1,3,5,7,9

3) X=linspace(0,10,30)
   i.e., X = 0, ……....., 10
   ⟷
   30 points

4) X=[1 4 3 2 −9]
   X = these values

5) X=[1 2 3; 4 5 6]

i.e., X = $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$ (matrix)

6) X=[1; 2; 3; 5; −10]

i.e., X = $\begin{bmatrix} 1 \\ 2 \\ 3 \\ 5 \\ -10 \end{bmatrix}$

Semicolon means → new line

❖ <u>Vectors calling:</u>

X=[1 2 3 4 7 9 − 15 3]

>> X(7)= -15
>> X(2:7)= 2 3 4 7 9 -15
>> X(2:end) = 2 3 4 7 9 -15 3
>> length(X) = 8 (number of elements)
>> sum(X) = 14
>> mean(X) = 1.75 (average)
>> max(X) = 9
>> min(X) = -15
>> X(1) = 1
>> X(7) = -15
>> X([2 4]) = 2 4
>> X(1:2:8) = 1 3 7 -15
>> Size(X) = 1 8 (dimension)

rows ← 1 8 → columns

>> i=find(X>3) } find the position of elements of X whose values are greater than 3

i = 4 5 6

- To get their values → >> X(i)= 4 7 9

To do the previous in one step → X(find(X>3)) = 4 7 9

>> X(find(X==max(X)))

To get number of elements greater than the average:
>> average=mean(X)
m=find(X>average)
L=length(m)

=> Try other methods…

- How to define a matrix? (Matrix algebra)

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

i.e., a = $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$

>> a=ones(2,2) i.e., a = $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$

row column

>> a=ones(3,3)*7 i.e., a = $\begin{bmatrix} 7 & 7 & 7 \\ 7 & 7 & 7 \\ 7 & 7 & 7 \end{bmatrix}$

>> a=zeros(2,2) i.e., a = $\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$

>> a=eye(3) i.e., a = $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

- Algebra:

>> det(a) → to get the determinant of (a)
>> inv(a) → to get the inverse matrix of a

If a and b are matrices,
- a*b = matrix expressing the matrix multiplication result
- a.*b = matrix expressing element by element multiplication

Ex:

→ a = $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ b = $\begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$

- a*b = $\begin{bmatrix} 14 & 32 & 50 \\ 32 & 77 & 122 \\ 50 & 122 & 194 \end{bmatrix}$

BUT

- a.*b = $\begin{bmatrix} 1 & 8 & 21 \\ 8 & 25 & 48 \\ 21 & 48 & 81 \end{bmatrix}$

→ a = $\begin{bmatrix} 3 & 4 & 6 \\ 9 & 5 & 2 \\ -1 & 5 & 9 \end{bmatrix}$

- a/b = a * b^{-1}
- a\b = a^{-1} * b
- a` = $\begin{bmatrix} 3 & 9 & -1 \\ 4 & 5 & 5 \\ 6 & 2 & 9 \end{bmatrix}$ → Transpose

❖ Some important notes:

>> X = 3; → this semicolon is to prevent printing

>> X=3; >> Y=5; >> Z=20; >> clc >> X → X=3	>> X=3; >> Y=5; >> Z=5; >> clear >> X → Error "Undefined function of variable 'X`"

>> Y=5.*X → this point is preferable to be put before any operation except (+ & -) to ensure "element by element" operation and NOT a matrix operation

>> Y==10
ans=0 if No
ans=1 if Yes

➔ **Try to write the following equations:**

1) $Y = X^3 \sqrt{X}$

2) $Y = \frac{\sin X}{X}$

3) $Y = \frac{(\sin X)^{x^2}}{2+tanX}$

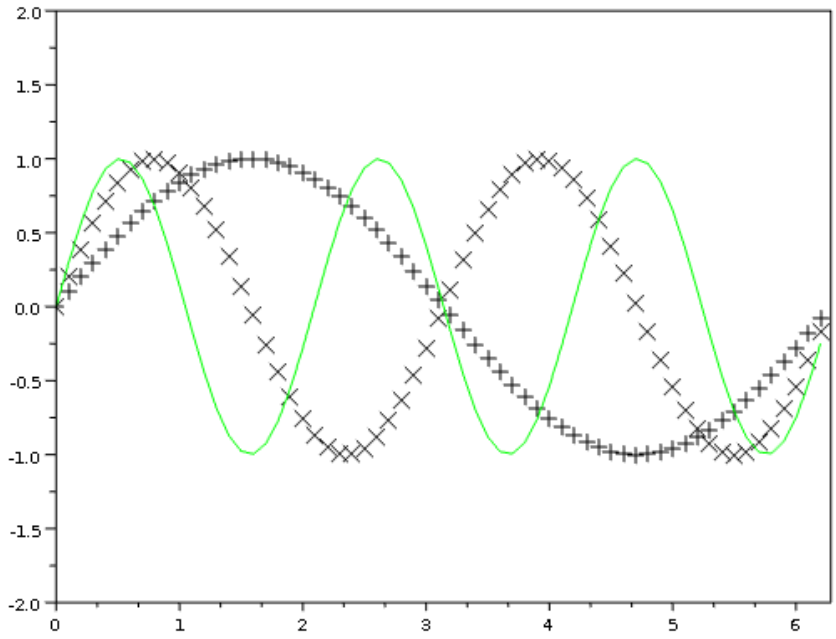
4) $Y = \sqrt{e^{X/2}} \sin X$

5) $Y = \sin X \, e^{-0.3X}$

Two-dimensional plotting
2D plotting

- Matlab has a powerful graphics system for presenting and visualized data….
- 2D graphics is plotting in 2D plane. It means that plotting using 2 variables (X, Y):

y=f(x)

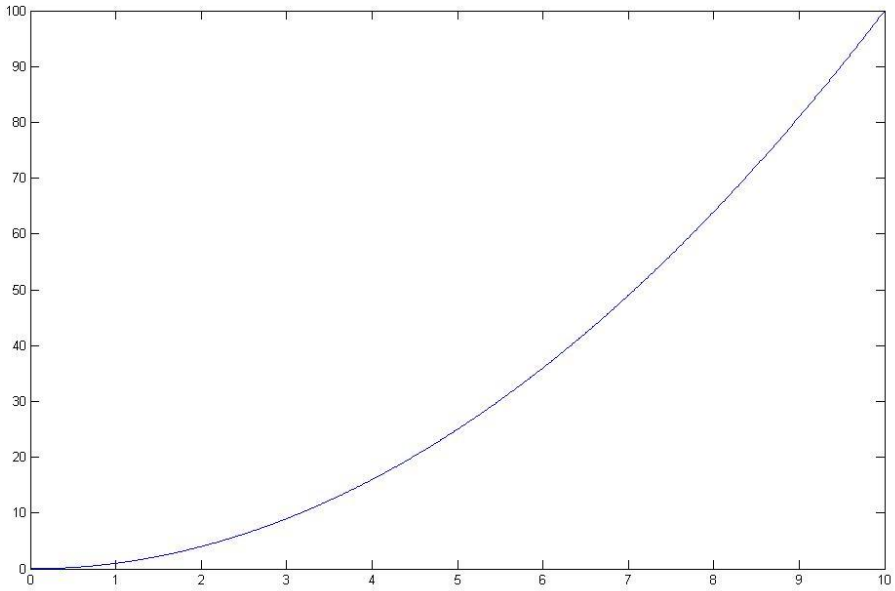


- Therefore, it is needed to define (x) and then the function f(x).

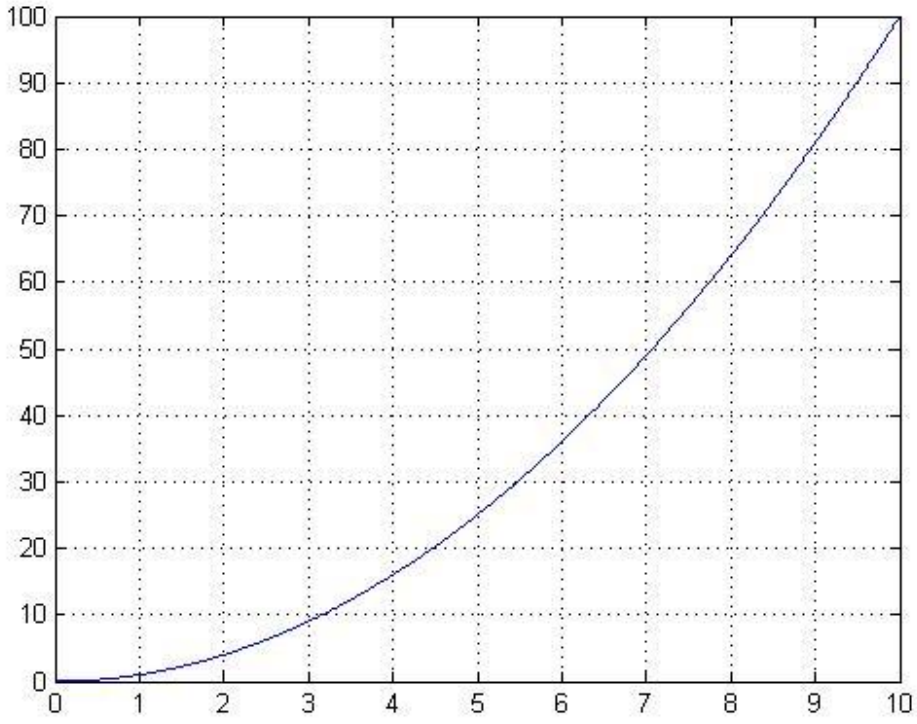
- Example:

We want to plot the function $y = x^2$

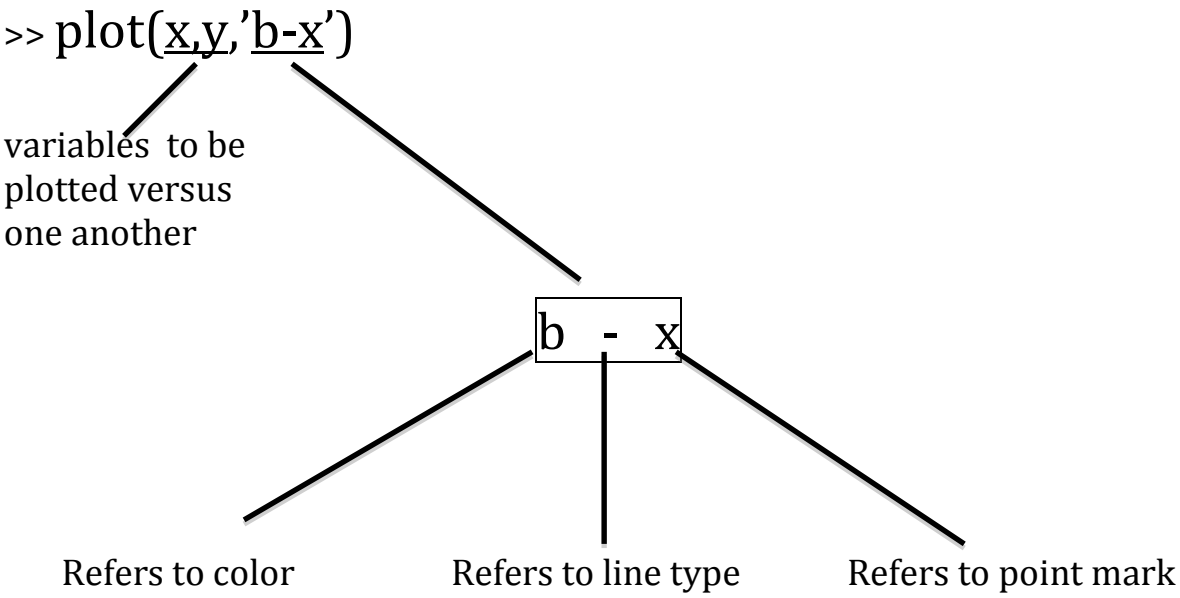
```
>> x=linspace(0,10,1000);
>> y=x.^2;
>> plot(x,y)
```



>> grid (→ this command is to make grids on the plot)



❖ Generally speaking:



r : red g : green b : blue c : cyan m : magenta y : yellow k : black w : white	- solid line -- dashed : dotted -. dash-dot line	x : cross + : plus sign o : circle s : square ^ : upward pointing triangle p : 5-pointed star h : 6-pointed star

❖ Graph annotations:

>> title('…….')
Creates a title for the figure.

>> xlabel('…')
Creates x-axis label.

>> ylabel('…')
Creates y-axis label.

>> text(x_1,y_1,'…')
Creates a certain text at coordinates (x_1,y_1).

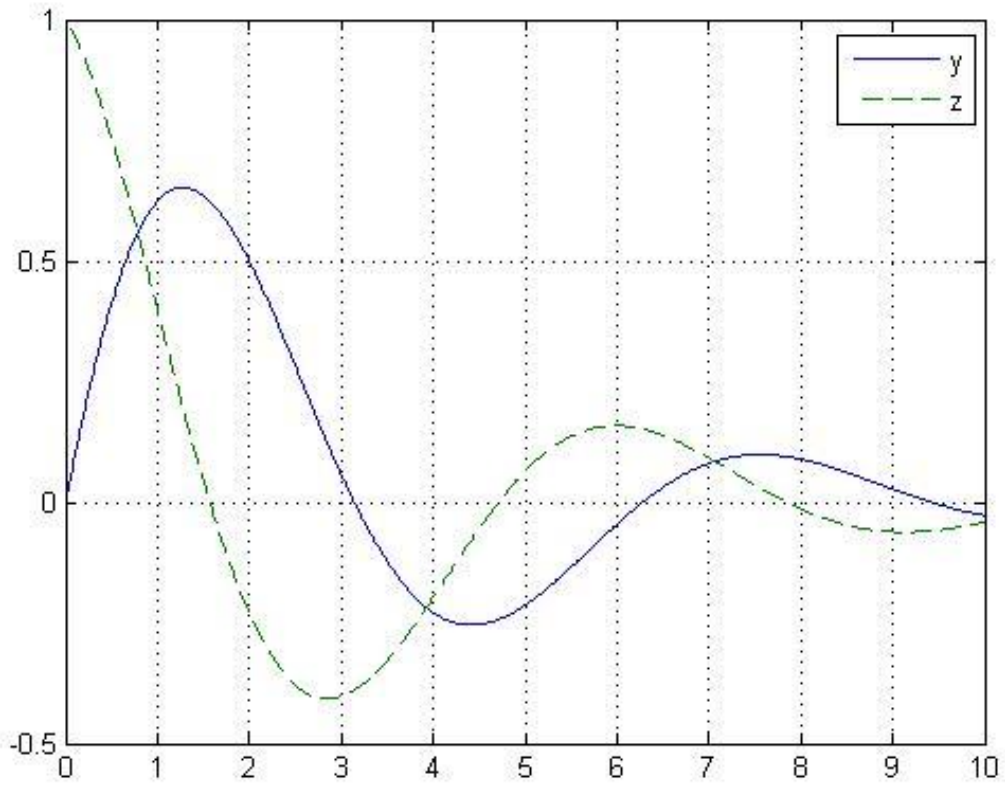
>> gtext('……')
Creates a certain text using the mouse cursor.

>> legend('…','…','…')
Creates legend for the graph.

>> hold on
Retrains the current plot and axes properties so that the subsequent graphing commands will be added to the existing one.

❖ To plot 2 functions on one graph:

```
>> x=linspace(0,10,1000);
>> y=sin(x).*exp(-0.3.*x);
>> z=cos(x).*exp(-0.3.*x);
>> plot(x,y,'-',x,z,'--')
>> grid
>> legend('y','z')
```



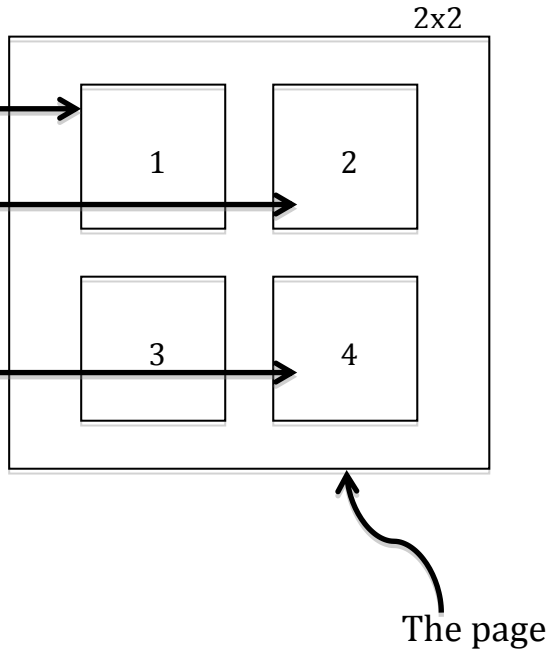
Note that:

To distinguish between y and z
>> plot(x,y,'-',x,z,'--')

solid line dashed line

❖ For more plots in one page:

```
>> subplot(2,2,1)
>> plot(x,y)
>> subplot(2,2,2)
>> plot(x,z)
.
.
.
.
.
>> subplot(2,2,4)
>> plot(x,y,x,z)
```



How to draw straight line connecting between two points?

>> plot([x1 x2], [y1 y2])