信号与系统实验
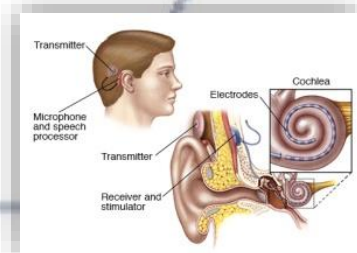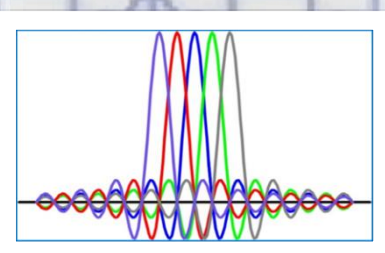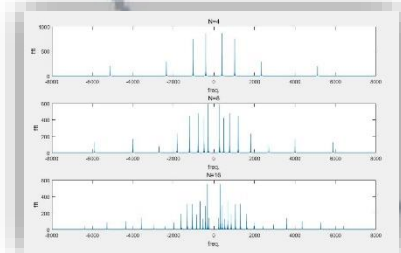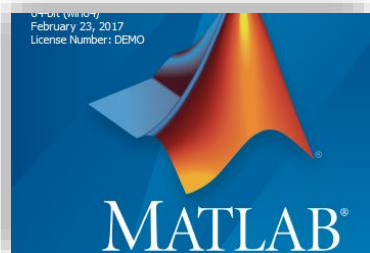
主讲人：吴光 博士
Email: wug@sustech.edu.cn

Signals and Systems (Lab)

Lab 1: MATLAB Programming

Dr. **Wu Guang**

wug@sustc.edu.cn

Electrical & Electronic Engineering

Southern University of Science and Technology

【通信新说】

提取码：yuip

Office: Room 130,
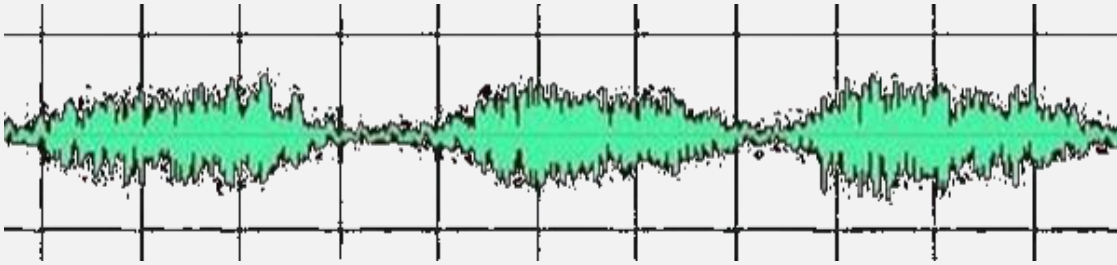　　　SUSTech No.1 Teaching Building
Email: wug@sustc.edu.cn
QQ Study Group: 40737542

https://pan.baidu.com/s/1wSpELGGyqSuhrH-38nOSdw

Part 1: <u>Introduction</u>

Objective: Analysis in frequency-domain

Time-domain

Frequency-domain

Labs in this course

📖 5 Lab assignments+2 Projects

- Lab 1: MATLAB Programming
- Lab 2: Linear Time-Invariant Systems
- Lab 3: Fourier Series Representation of Periodic Signals
- Lab 4: The Continuous-Time Fourier Transform
- Lab 5: System, Transform, Convolution and Filter
- Project 1. Speech synthesis and perception with envelope cue
- Project 2: OFDM Technology
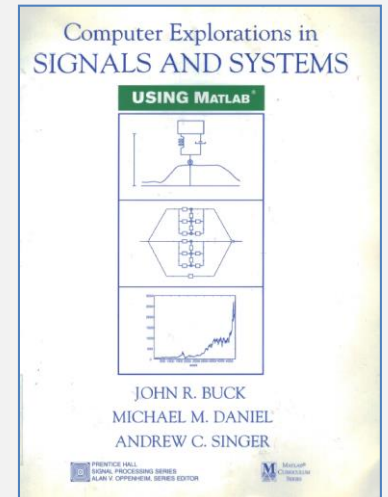
Arrangements

📖 5 lab assignments

- Two students work as a group;
- Two weeks per assignment;
- Evaluation: Report + MATLAB code
- Hard deadline: Please submit your report before the next lab session.

📖 2 projects

- Four students work as a group;
- Three weeks per project;
- Evaluation: Essay + MATLAB code + **presentation**;
- Hard deadline: please submit your report before the next project.

How to write your report ?

✏ Edit your report by word. Use the following format:

Write **an short introduction** to the lab assignment

Type down Question 1
Give you answer to Q1, add the figures if necessary

Type down Question 2
Give you answer to Q2, add the figures if necessary

Type down Question 3
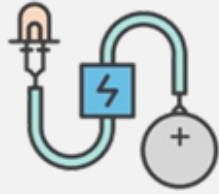Give you answer to Q3, add the figures if necessary
…

✏ Email to wug@sustc.edu.cn

Email Title: (Lab Index)+(Student Name 1)+(Student Name 2)+(Student No. 1)+(Student No. 2)
Example: Lab1+张三+王伟+00001+00002

What's Simulation ?

The proposed algorithm B is better than the existing algorithm A

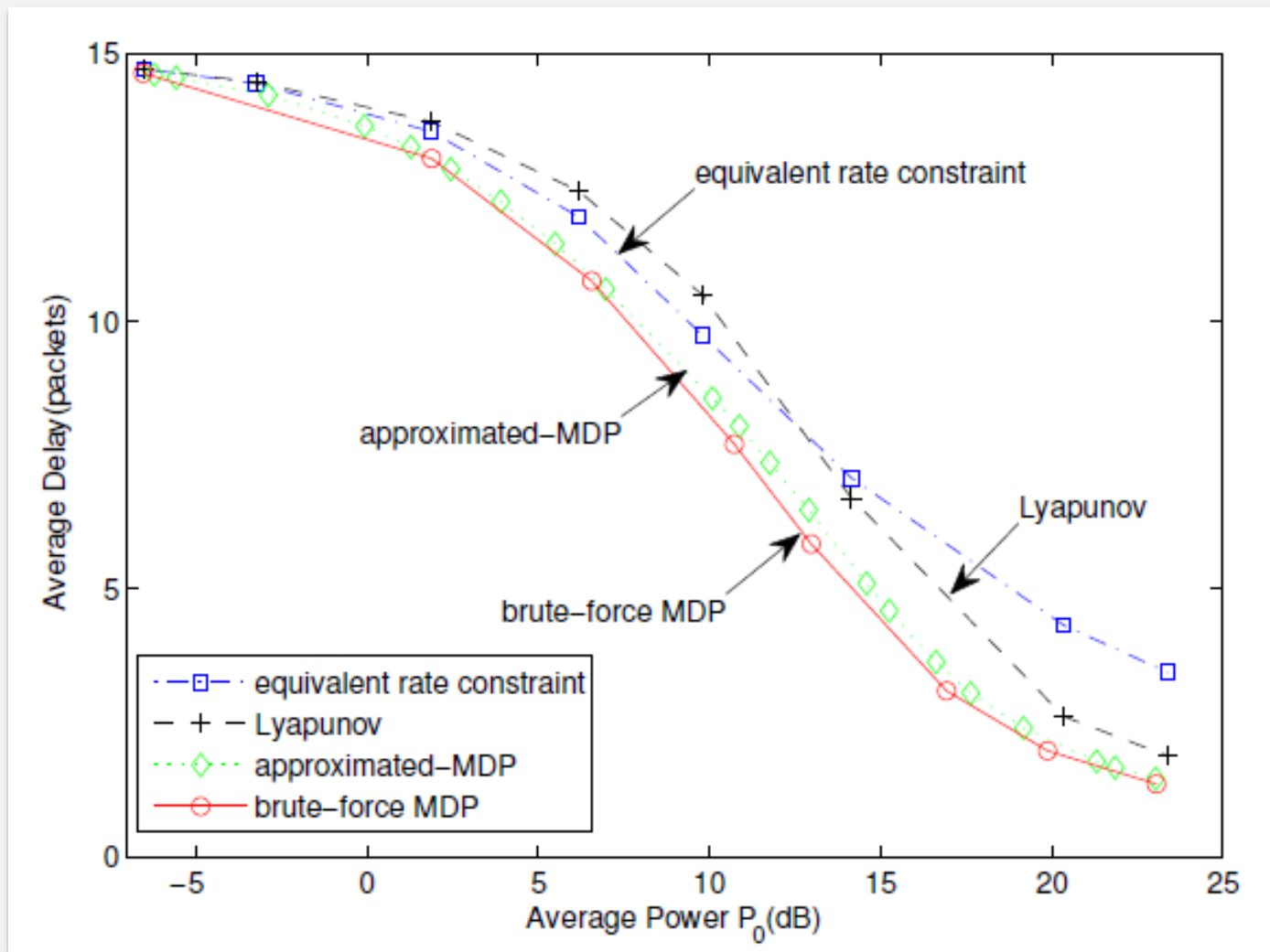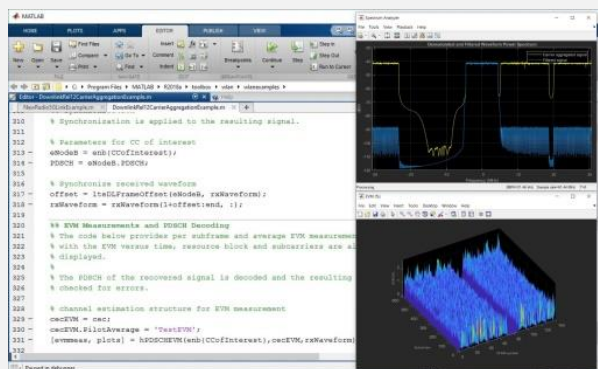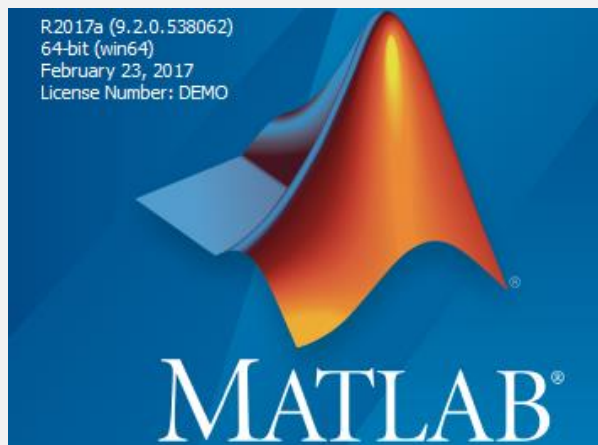Derive the **math expression** of performance for Algorithm A & B

Solid math background is necessary

Test Algorithm A & B **by computer** with typical system configuration

You need to write program

How to express your result ?

Part 2: <u>MATLAB Tutorial</u>

What can you do with MATLAB
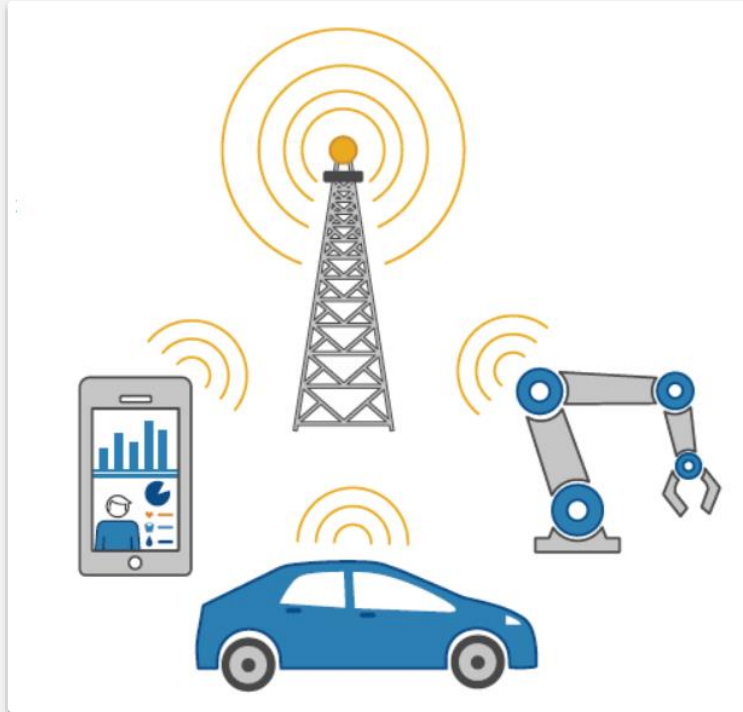
5G Communication
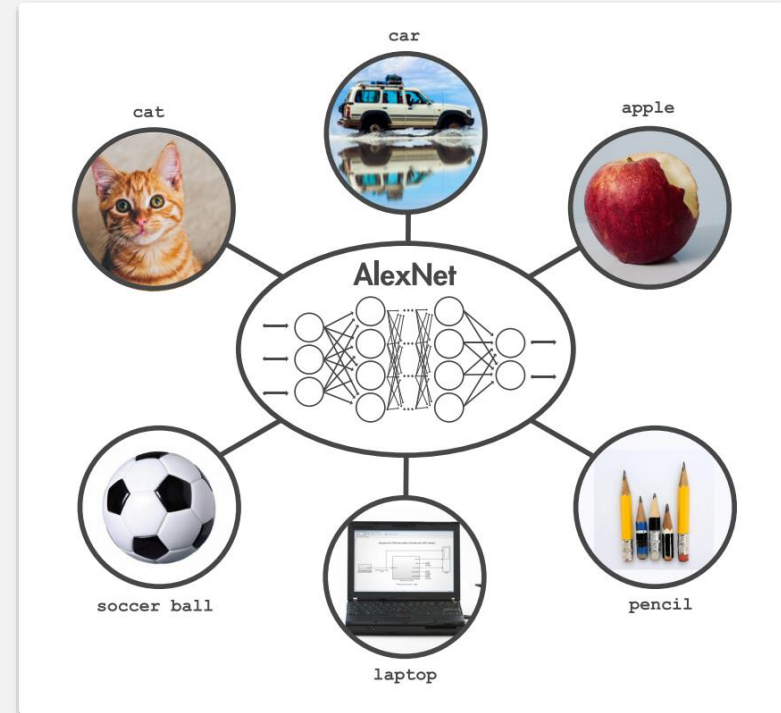
Deep Learning

计算生物学

分析、可视化及对生物数据和系统进行建模

控制系统

设计、测试和实现控制系统

数据科学

形成数据驱动的洞察力，以改进设计和决策

深度学习

设计、构建和可视化卷积神经网络

数字信号处理

从多个来源获取、测量和分析信号

嵌入式系统

设计、编码和验证嵌入式系统

企业和 IT 系统

安全部署 MATLAB 代码至企业 IT 系统

FPGA、ASIC 及 SoC 设计

建模、实现并验证 FPGA、ASIC 和 SoC 设计

图像处理和计算机视觉

采集、处理和分析图像和视频以进行算法开发和系统设计

物联网

连接嵌入式设备与互联网，并从数据中获取洞察力

机器学习

发现规律与建立预测模型

机电一体化

设计、优化和验证机电系统

电力电子器件控制设计

设计和实现电机、功率变换器和电池系统的数字控制

预测性维护

开发和部署状态监控和预测性维护软件

机器人

将机器人构想和概念转变为在真实环境下工作的自主系统

测试和测量

收集、分析和浏览数据并自动化测试

无线通信

创建、设计、测试和验证无线通信系统

MATLAB (MATrix LABoratory)

- Easy-to-use, save your efforts in coding
- Everything in MATLAB is matrix
- Interpreted language
- It has its own working environment
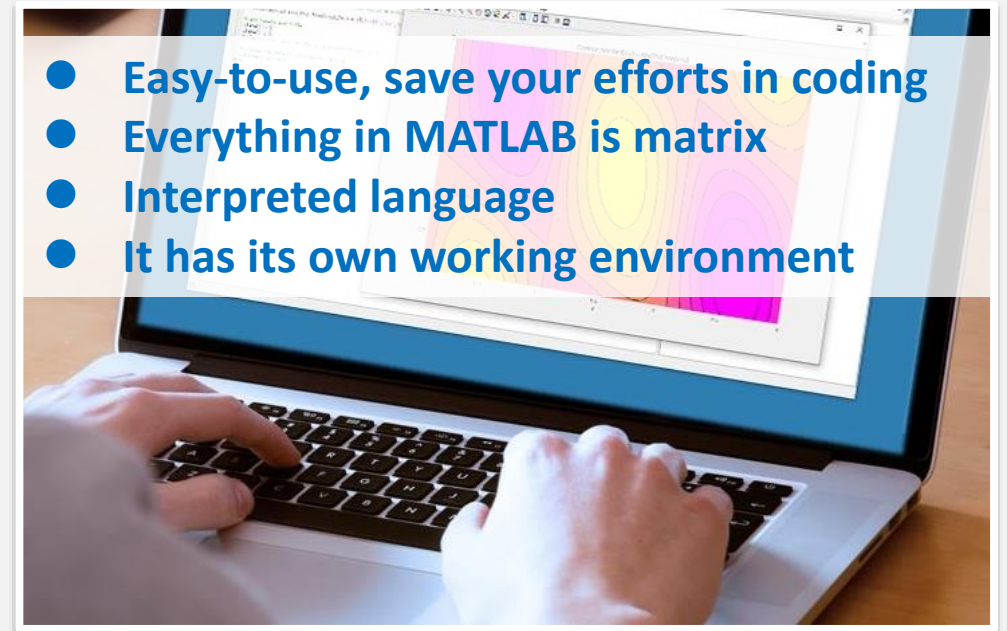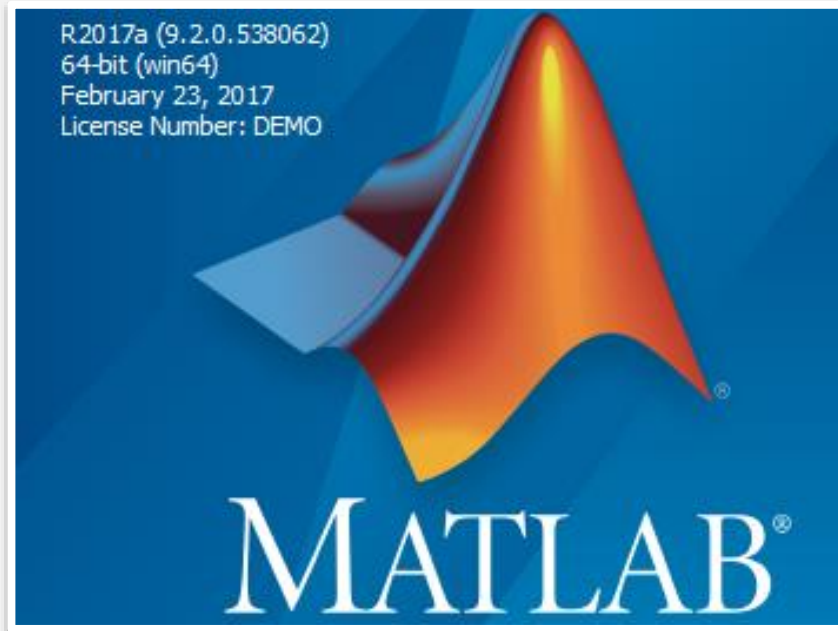
MATLAB Environment

Type your MATLAB statement in **Command Window, you can interact with MATLAB here.**

All currently defined variables will be stored listed out in the **Workspace**.
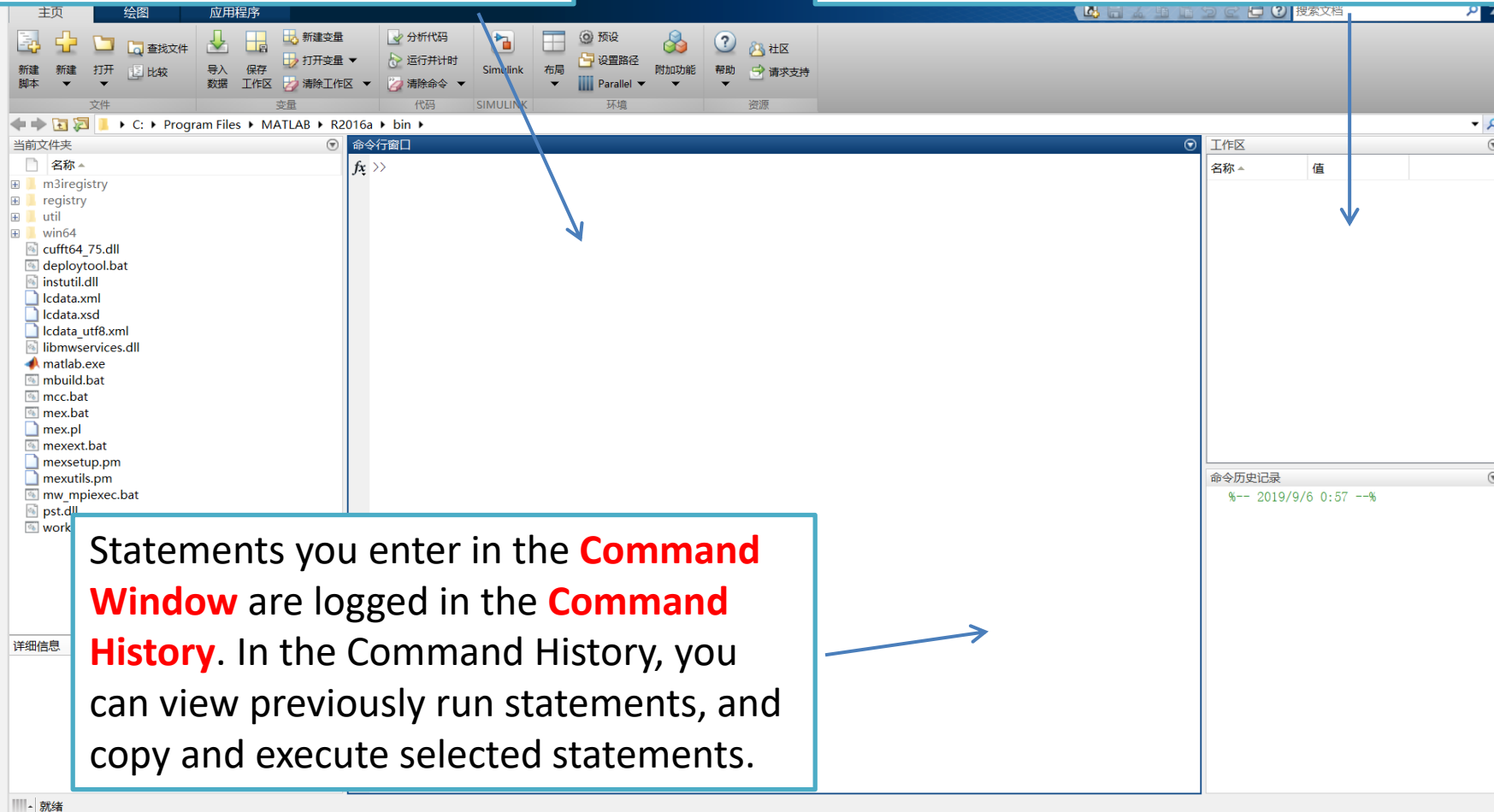
Statements you enter in the **Command Window** are logged in the **Command History**. In the Command History, you can view previously run statements, and copy and execute selected statements.
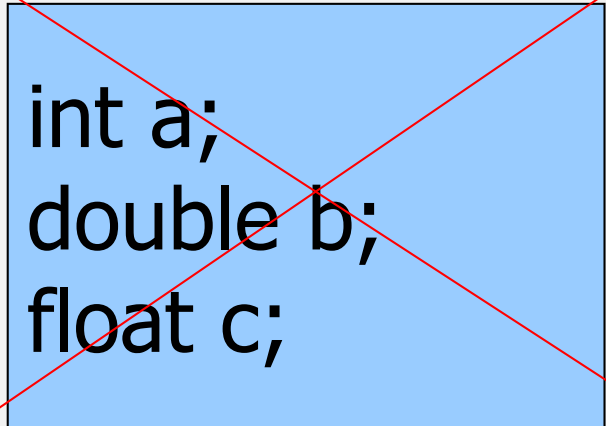
Language Overview
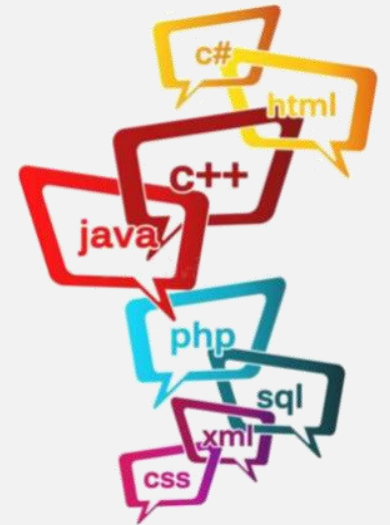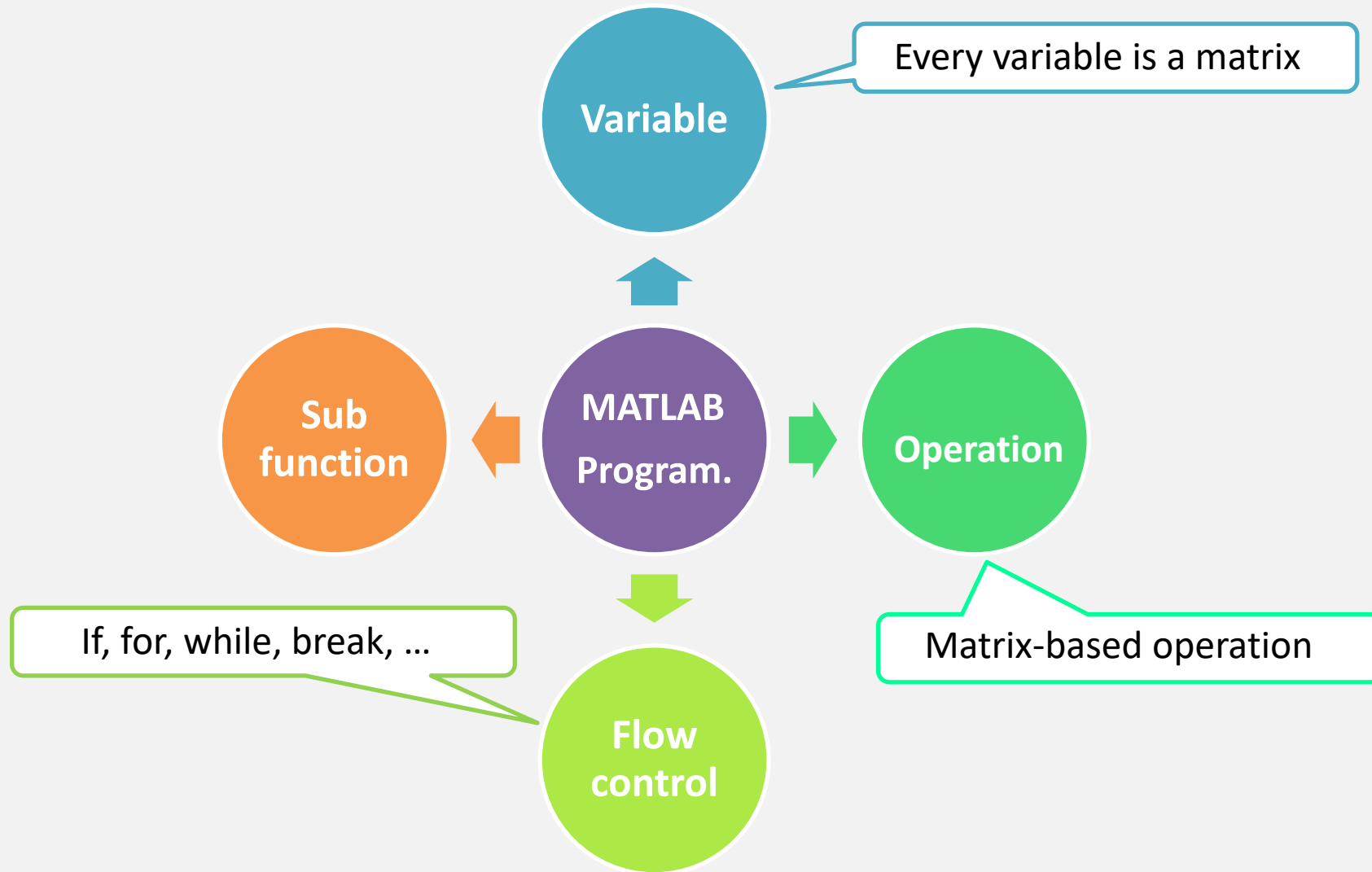
How to Define Variables?

Example:

    a=5
    a='hello world'
    a=5.5
    b=10+1i
    a=a*b
    a=a+b

int a;
double b;
float c;

No need to claim types.

① Latest value of variable will be stored in the workspace
② Semicolon (;) will stop the interaction with command window

Every Variable is a Matrix

- **A vector** **x = [1 2 5 1]**

  `Row Vector`

  ```
  x =
       1    2    5    1
  ```

- **A matrix** **y = [1 2 3; 5 1 4; 3 2 -1]**

  `Matrix`

  ```
  y =
       1        2        3
       5        1        4
       3        2       -1
  ```

- **Transpose** **z = x'**   `z =`

  `Column Vector`

  ```
       1
       2
       5
       1
  ```

Long Vector & Matrix

t =1:10

t =
   1   2   3   4   5   6   7   8   9   10

----

k =2:-0.5:-1

k =
   2 1.5 1 0.5 0 -0.5 -1

B = [1:4; 5:8]

B =

1	2	3	4
5	6	7	8

Generate vectors from functions

- zeros(M,N) | MxN matrix of zeros

```
x = zeros(1,3)
x =
   0    0    0
```

- ones(M,N) | MxN matrix of ones

```
x = ones(1,3)
x =
   1    1    1
```

- rand(M,N) | MxN matrix of uniformly distributed random numbers on (0,1)

```
x = rand(1,3)
x =
   0.9501  0.2311 0.6068
```

Matrix Index

- **The matrix indices begin from 1 (not 0 (as in C))**
- **The matrix indices must be positive integer**

```
A =

      3     5     3
      6     8     2
      2     7     3
```

```
>> A(6)

ans =

      7
```

```
>> A(3,2)

ans =

      7
```

```
>> A(2,:)

ans =

      6     8     2
```

```
>> A(1:2,2)

ans =

      5
      8
```

A(-2), A(0)
Error: ??? Subscript indices must either be real positive integers or logicals.

A(4,2)
Error: ??? Index exceeds matrix dimensions.

Concatenation of Matrices

```
x = [1 2], y = [4 5], z=[ 0 0]

A = [x y]

    1    2    4    5

B = [x ; y]

      1 2
      4 5
```

C = [x y ;z]
Error using vertcat
Dimensions of matrices being concatenated are not consistent.

Operators (Arithmetic)

+ addition

- subtraction

* multiplication

/ division

^ power

' complex conjugate transpose

Matrices Operations

Given A and B:

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

A =

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

```
>> B = [3 5 2; 5 2 8; 3 6 9]

B =

       3       5       2
       5       2       8
       3       6       9
```

How about
  A^2
  A/B
  2*A
  A+1j*B
  (A+1j*B)'

Addition

```
>> X = A + B

X =

       4       7       5
       9       7      14
      10      14      18
```

Subtraction

```
>> Y = A - B

Y =

      -2      -3       1
      -1       3      -2
       4       2       0
```

Product

```
>> Z = A * B

Z =

      22      27      45
      55      66     102
      88     105     159
```

Transpose

```
>> T = A'

T =

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

Element-Wise Operators

In the previous example, please compare

    A^2   v.s.   A.^2
    A*B   v.s.   A.*B
    (A+1j*B)'   v.s.   (A+1j*B).'

.*   element-by-element multiplication

./   element-by-element division

.^   element-by-element power

.'   transpose

```
A = [1 2 3; 5 1 4; 3 4 -1]
   A =

        1    2    3
        5    1    4
        3    4   -1
```

```
x = A(1,:)


x=

    1   2   3
```

```
y = A(3 ,:)


y=

    3  4  -1
```

```
b = x .* y


b=

    3  8 -3
```

```
c = x . / y


c=

   0.33  0.5  -3
```

```
d = x .^2


d=

    1    4    9
```

```
K= x^2
Erorr:
 ??? Error using ==> mpower  Matrix must be square.
B=x*y
Erorr:
??? Error using ==> mtimes Inner matrix dimensions must agree.
```
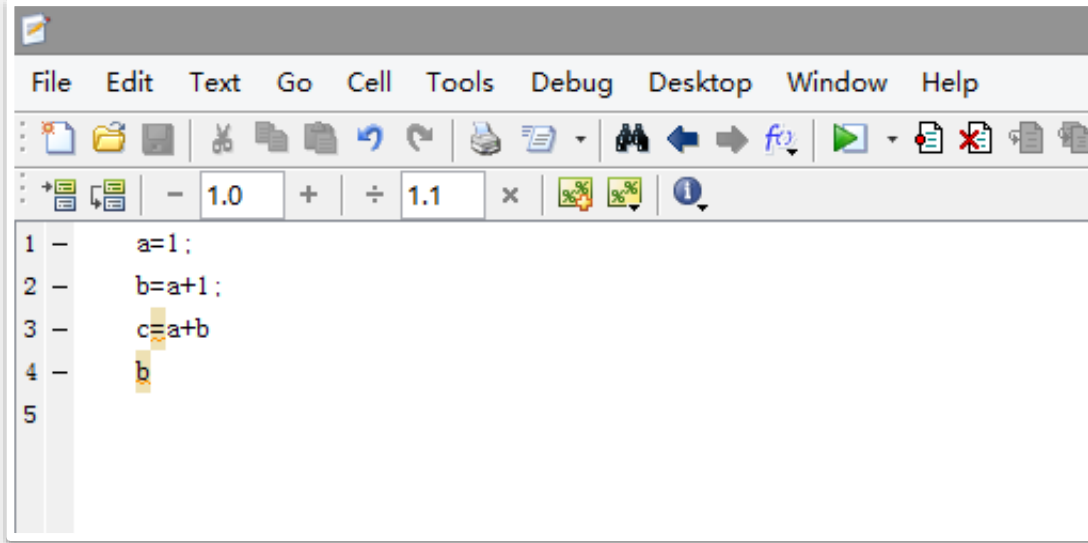
Useful Commands

```
>> who
>> whos
>> clear
>> clc
>> dir
>> help/doc
```
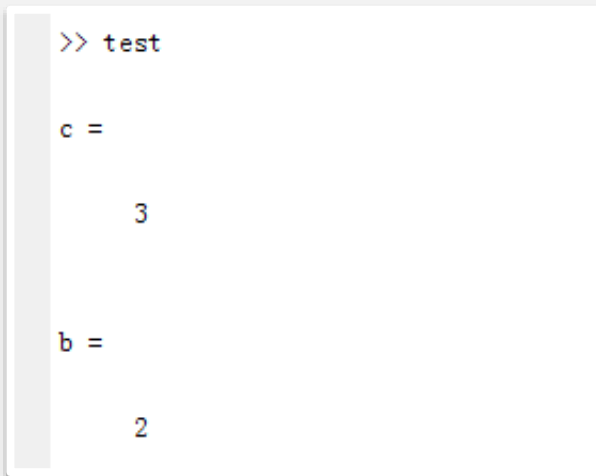
Write a Script

Instead of typing in the **Command Window**, you can write a script in **Editor** and save in a **.m file**, like test.m.

Then, you can run the .m file in **Command Window**. The command in the .m file will be executed one by one (like a batch file).
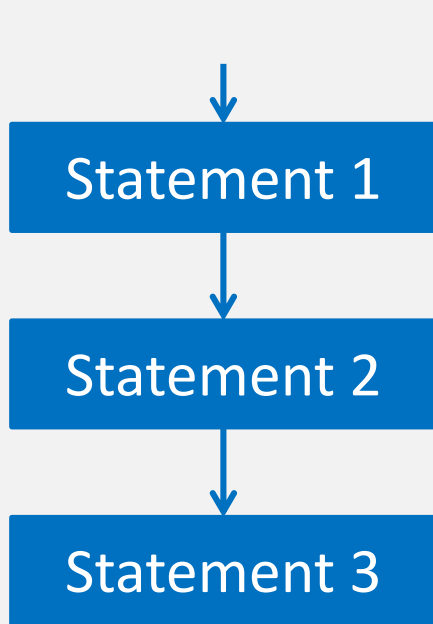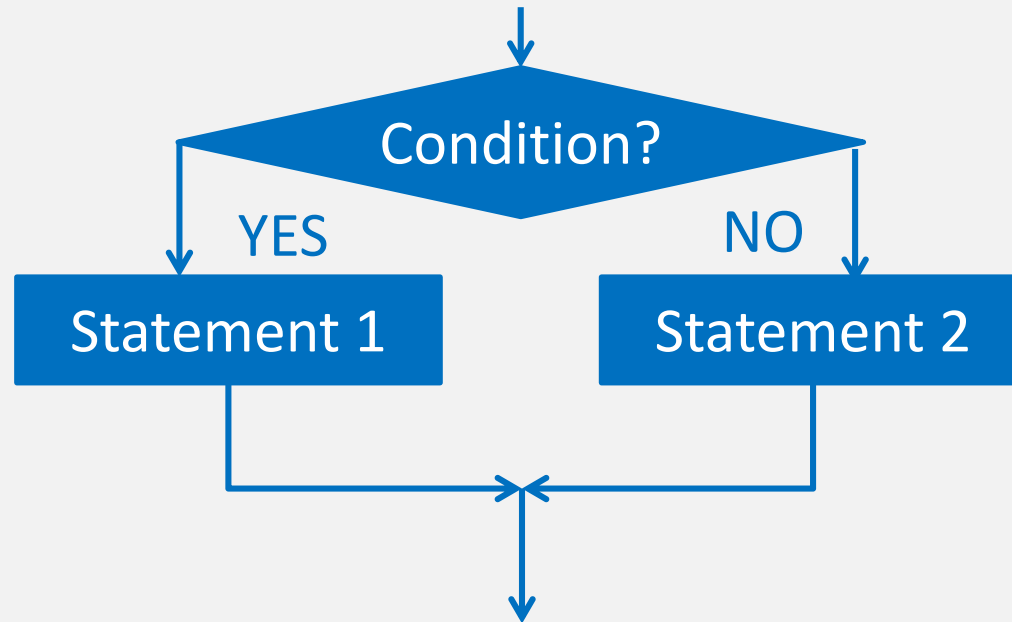
Run and Debug

Flow Control

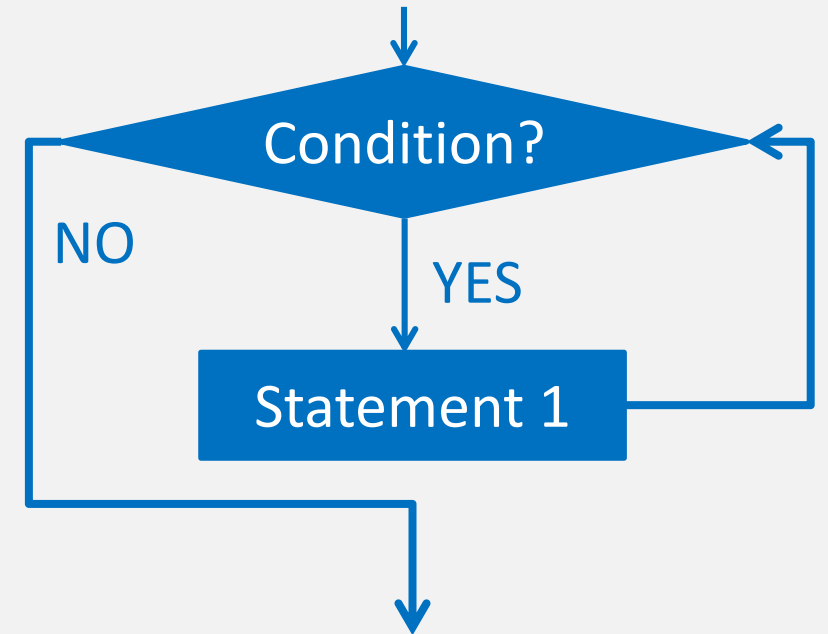Similar to almost all programming language, MATLAB program has three basic structures:

– Sequence, Selection and Repetition

Sequence

Selection: if

Repetition: for & while

```matlab
if (Condition_1)

        MATLAB Commands

elseif (Condition_2)

        MATLAB Commands

elseif (Condition_3)

        MATLAB Commands

else

        MATLAB Commands

end
```

```matlab
if ((a>3) & (b==5))
        Some MATLAB Commands;
end
```

```matlab
if (a<3)
        Some MATLAB Commands;

elseif (b~=5)
        Some MATLAB Commands;
end
```

```matlab
if (a<3)
        Some MATLAB Commands;
else
        Some MATLAB Commands;
end
```

Exercise 1.1 Selector.m

Write a script to add '5' to a number if it is greater than '10', else add '10'.

Repetition -- For

```
for i=Index_Array

    MATLAB Commands

end
```

```
for i=1:100
    Some MATLAB Commands;
end
```

```
for j=1:3:200
    Some MATLAB Commands;
end
```

```
for m=13:-0.2:-21
    Some MATLAB Commands;
end
```

```
for k=[0.1 0.3 -13 12 7 -9.3]
    Some MATLAB Commands;
end
```

Repetition --While

```
while (condition)

    MATLAB Commands

end
```

```
while  ((a>3) & (b==5))

        Some MATLAB Commands;

end
```

Exercise 1.2 Accumulator.m

Write a script to output the sum of numbers from 1 to the number input.

Operators (Logical)

1. == Equal to
2. ~= Not equal to
3. < Strictly smaller
4. > Strictly greater
5. <= Smaller than or equal to
6. >= Greater than equal to
7. & And operator
8. | Or operator
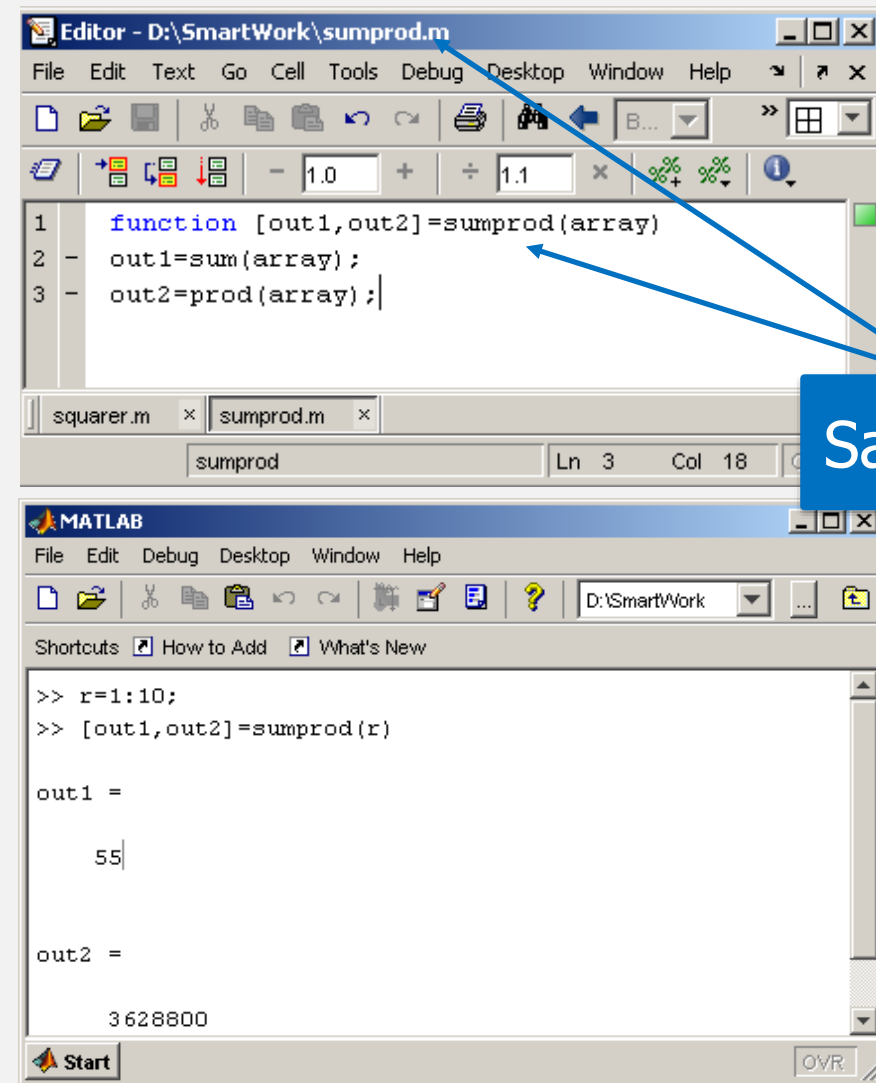9. ~ Not operator

Writing User Defined Functions

- Functions are m-files which can be executed by specifying some inputs and supply some desired outputs.

- The code telling the MATLAB that an m-file is actually a function is

```
function out1=functionname(in1)
function out1=functionname(in1,in2,in3)
function [out1,out2]=functionname(in1,in2)
```

- You should write this command **at the beginning of the m-file** and you should save the m-file **with a file name same as the function name**

Writing User Defined Functions

- Another function which takes an input array and returns the sum and product of its elements as outputs

- The function sumprod(.) can be called from command window or an m-file as

Editor – D:\SmartWork\sumprod.m

File Edit Text Go Cell Tools Debug Desktop Window Help

```
1    function [out1,out2]=sumprod(array)
2 -  out1=sum(array);
3 -  out2=prod(array);
```

squarer.m   sumprod.m

sumprod   Ln 3   Col 18

Same Name

MATLAB

File Edit Debug Desktop Window Help

Shortcuts How to Add What's New

```
>> r=1:10;
>> [out1,out2]=sumprod(r)

out1 =

    55


out2 =

    3628800
```

Start   OVR

Notes

- "%" is the neglect sign for MATLAB (equaivalent of "//" in C). Anything after it on the same line is neglected by MATLAB compiler.
- Sometimes slowing down the execution is done deliberately for observation purposes. You can use the command "pause" for this purpose

```
pause %wait until any key
pause(3) %wait 3 seconds
```

Try to Avoid Loops

- Instead of loops, try to use matrix operators and bulit-in functions.
- Example: let a = [a1 a2 a3] and b=[b1 b2 b3], write a function to calculate

c=[a1+b1 a1+b2 a1+b3;
   a2+b1 a2+b2 a2+b3;
   a3+b1 a3+b2 a3+b3]

- There are two solutions

```
function c = add1(a,b)
c = zeros(3,3);
for m=1:3
  for n=1:3
    c(m,n) = a(m) + b(n);
  end;
end;
```

```
function c = add2(a,b)
c = diag(a) * ones(3) + ones(3) * diag(b);
```

Test 1

IF (The input number N is an **even integer**)

    Sum=2+4+6+…+N;

IF (The input number N is an **odd integer**)

    Sum=1+3+5+…+N;

Basic task: Plot the signal sin(x) between 0≤x≤4π

- Create an x-array of 100 samples between 0 and 4π.

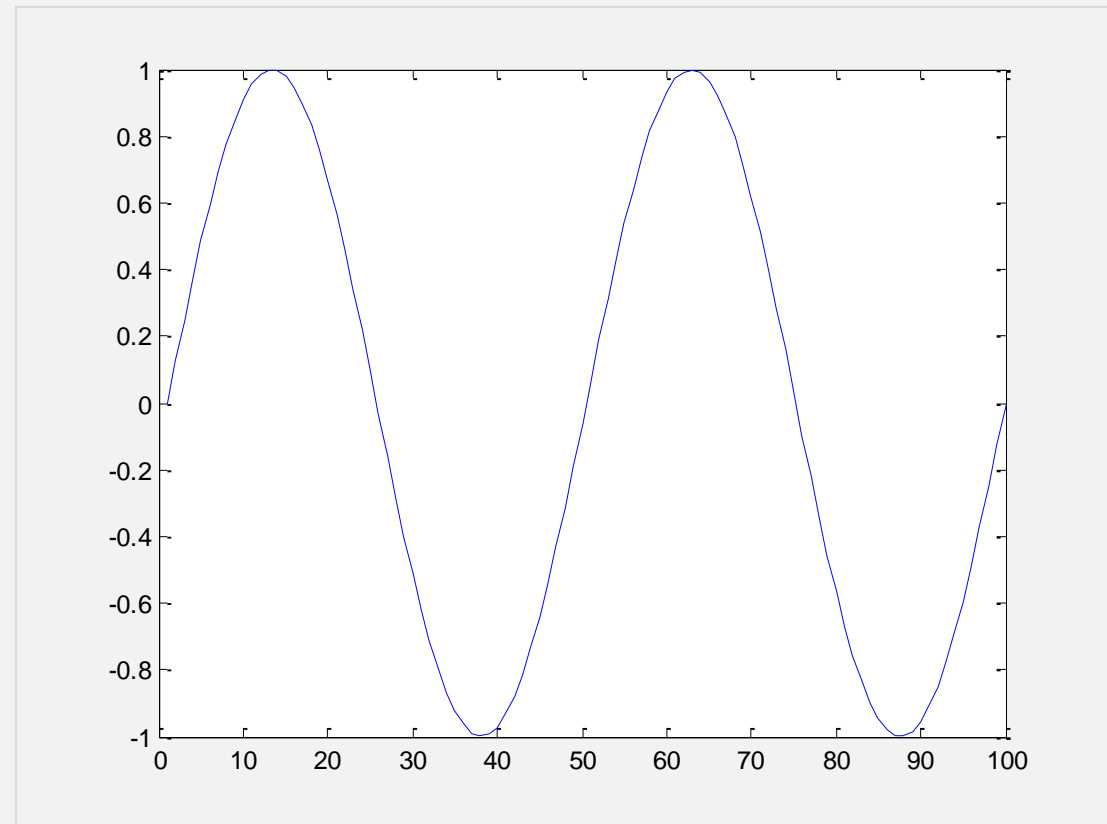>>x=linspace(0,4*pi,100);

- Calculate sin(.) of the x-array

>>y=sin(x);

- Plot the y-array

>>plot(y)
>>plot(x,y)

Plot the signal $e^{-x/3}\sin(x)$ between $0 \leq x \leq 4\pi$

- Create an x-array of 100 samples between 0 and 4π.

```
>>x=linspace(0,4*pi,100);
```

- Calculate sin(.) of the x-array

```
>>y=sin(x);
```

- Calculate $e^{-x/3}$ of the x-array

```
>>y1=exp(-x/3);
```
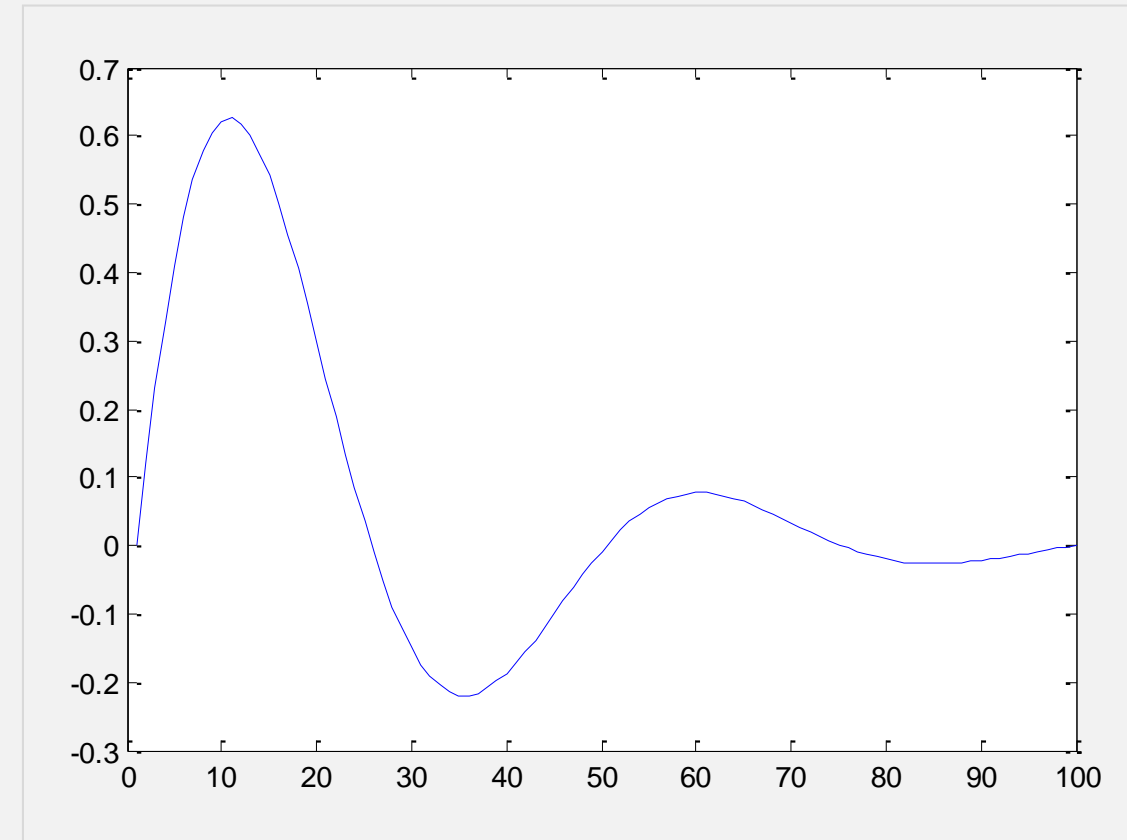
- Multiply the arrays y and y1

```
>>y2=y*y1;
```

Plot the signal e⁻ˣ/³sin(x) between 0≤x≤4π

- Multiply the arrays y and y1 **correctly**

```
>>y2=y.*y1;
```

- Plot the y2-array

```
>>plot(y2)
>>plot(x,y2)
```

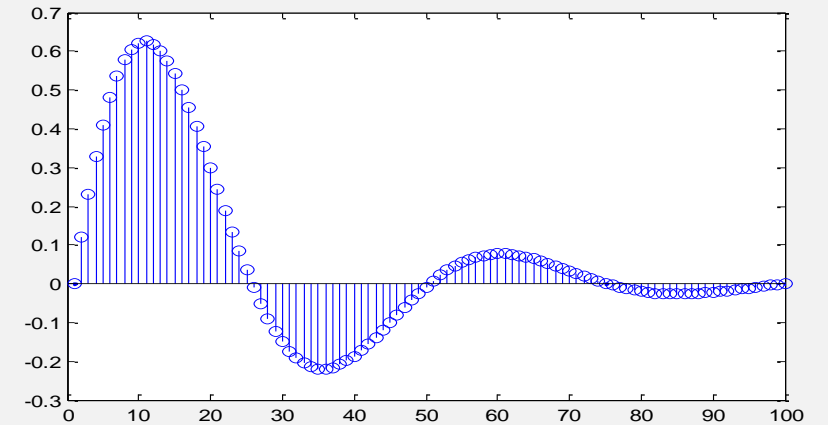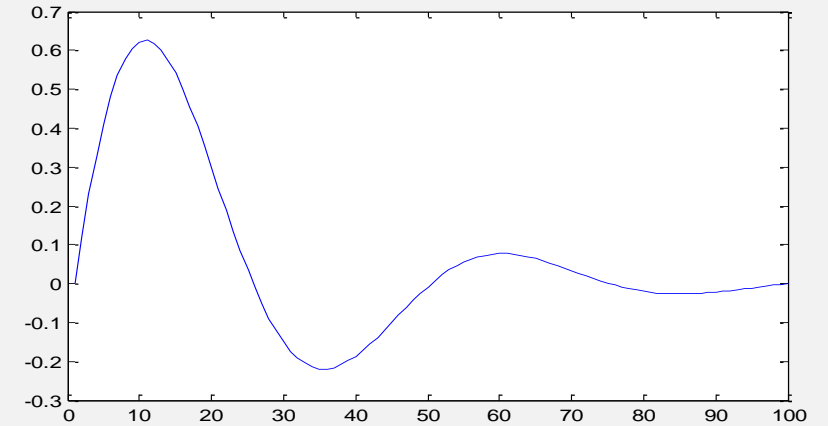# Plot and Stem

Example:
>>x=linspace(0,4*pi,100);
>>y=sin(x);
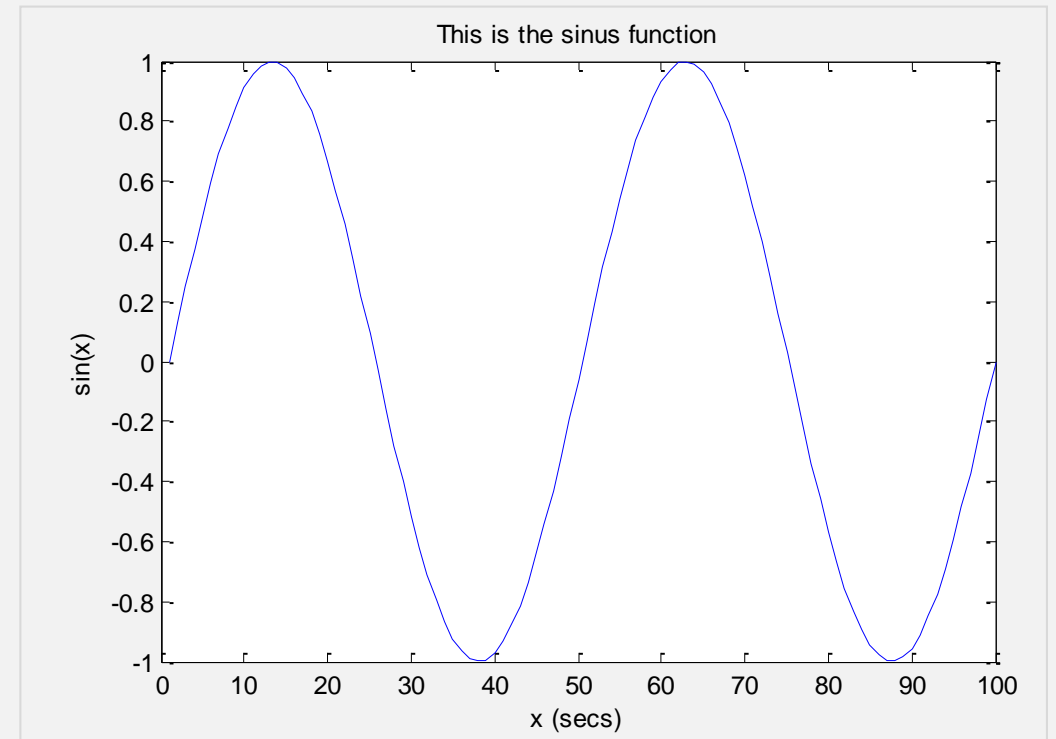>>plot(y)
>>plot(x,y)

Example:
>>stem(y)
>>stem(x,y)

title,xlabel and ylabel

>>title('This is the sinus function')

>>xlabel('x (secs)')

>>ylabel('sin(x)')

Line types,plot symbols and colors

- Plot with various line types, plot symbols and colors

|					
b	blue	.	point	–	solid
g	green	o	circle	:	dotted
r	red	x	x-mark	–.	dashdot
c	cyan	+	plus	––	dashed
m	magenta	*	star	(none)	no line
y	yellow	s	square		
k	black	d	diamond		
w	white	v	triangle (down)		
		^	triangle (up)		
		<	triangle (left)		
		>	triangle (right)		
		p	pentagram		
		h	hexagram		

Colors

Symbols

Line types

Example

>> plot(x, y, 'r--o')

>> title('This is the sinus function','fontsize',20)

>> **grid on**

>> box off

>> axis([0 4*pi ylim])

This is the sinus function
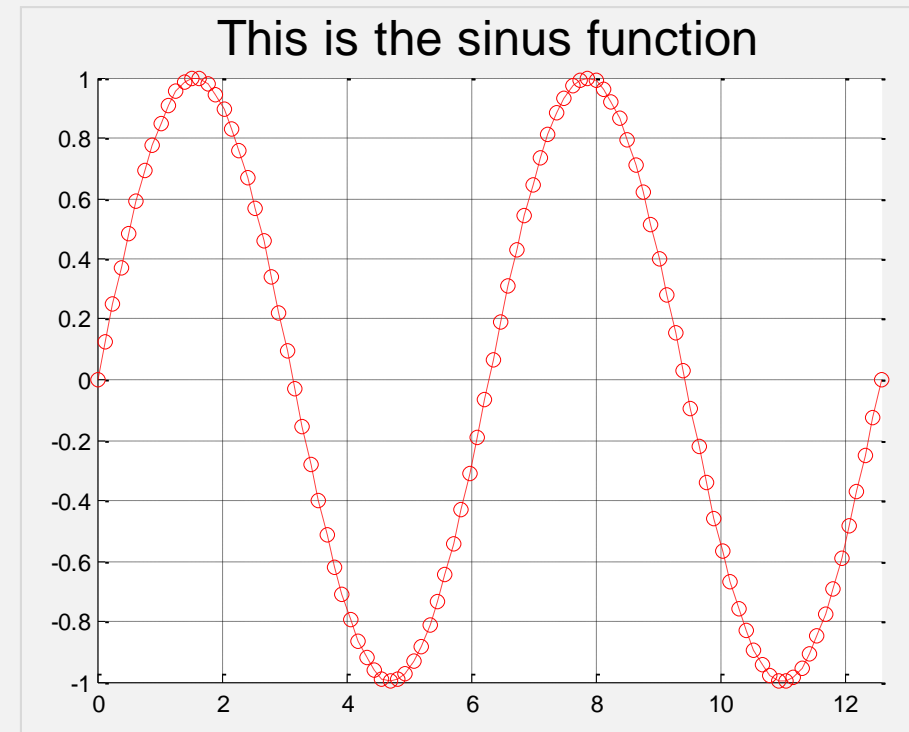
More plots in one panel

\>\> figure(1)

\>\> plot(x, y, 'r--', 'linewidth',2)

\>\> **hold on**

\>\> plot(x, y+1, 'k--', 'linewidth',2)

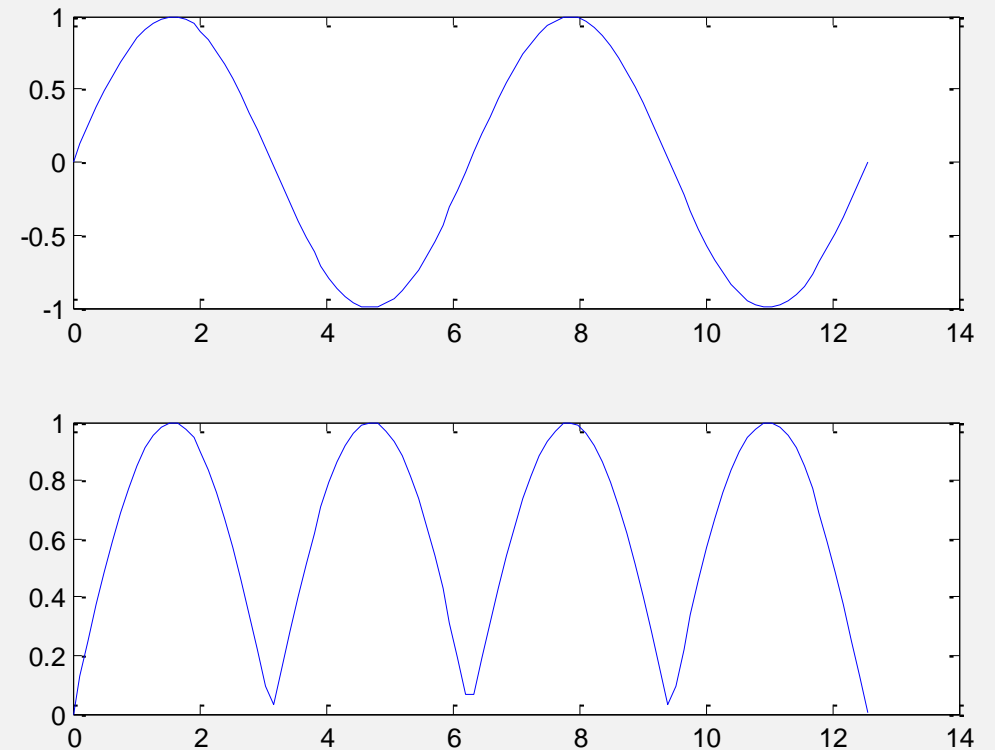\>\> legend('wave 1', 'wave 2')
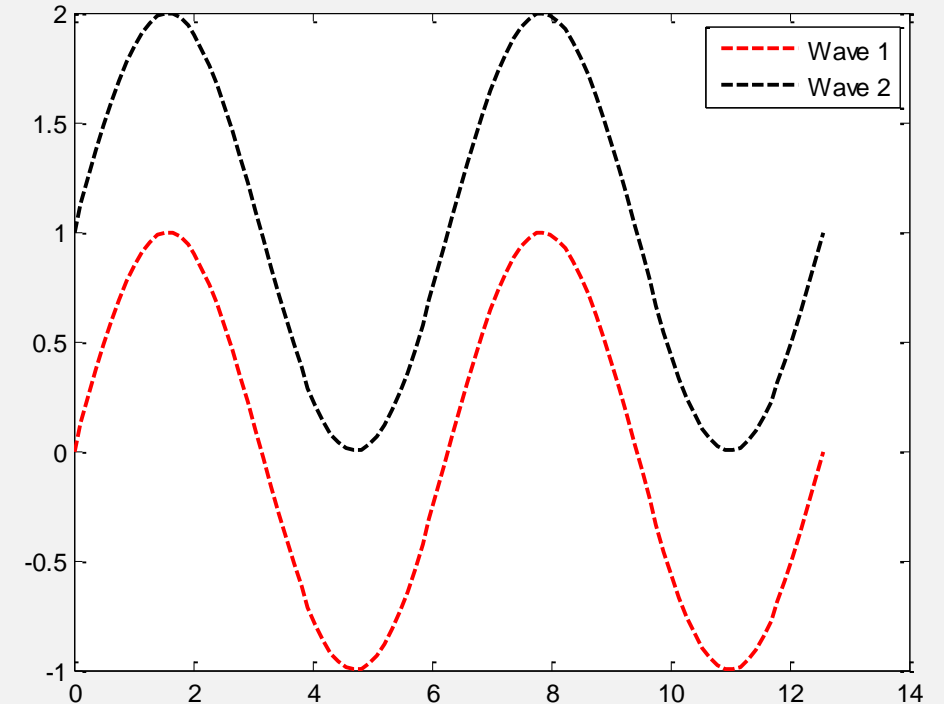
More figures

> ## **Subplot**

>>subplot(2,1,1)

row   column   index

>> figure(1)
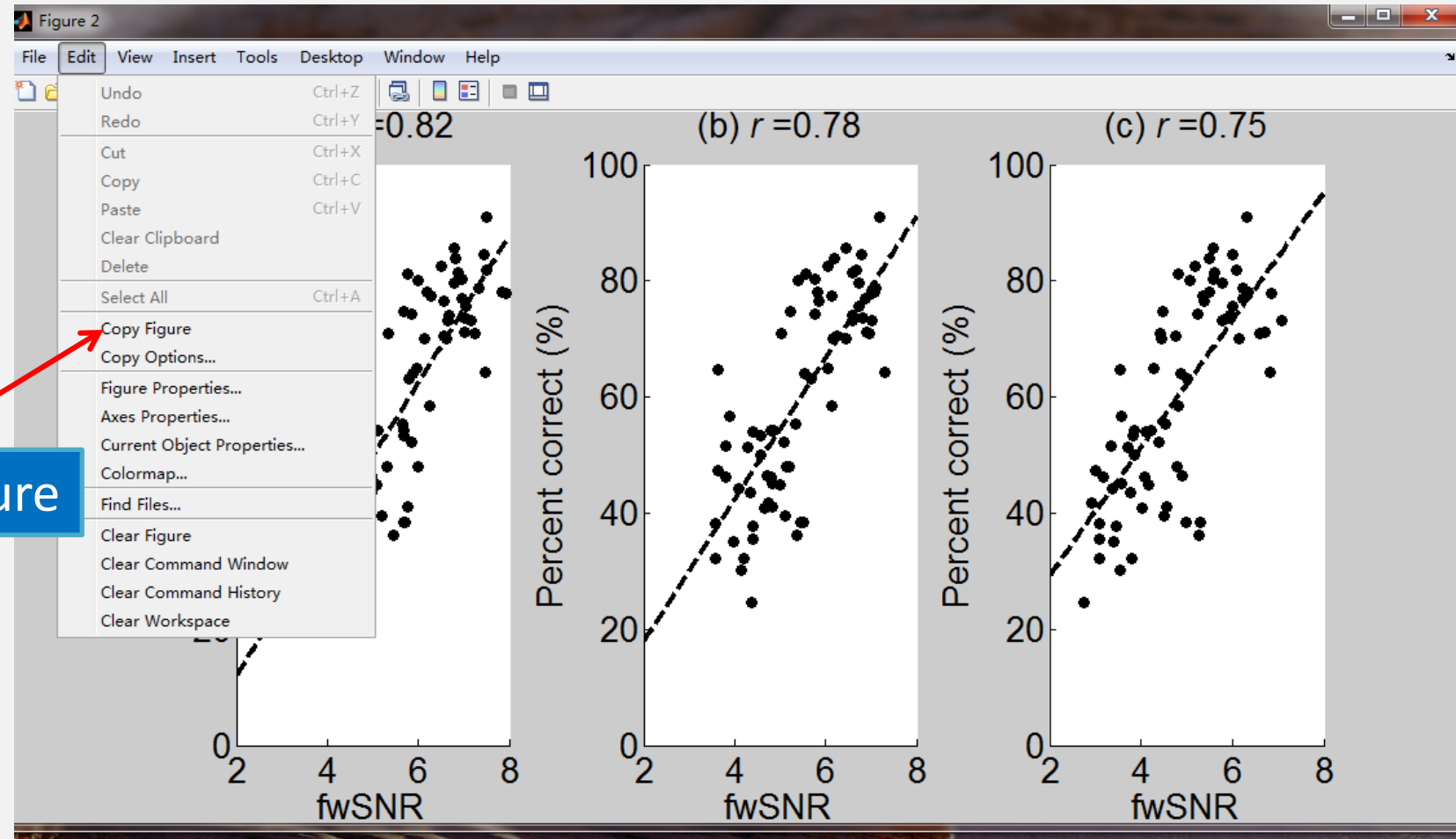
>> subplot(2,1,1),plot(x,y)

>> subplot(2,1,2),plot(x,abs(y))

Copy MATLAB Figure

Some useful functions

➢ **length(.)**

➢ **size(.)**

➢ **abs(.)**

➢ **sum(.)**

➢ **mean(.)**

➢ **std(.)**

➢ **diff(.)**

命令行窗口

```
>> help mean
mean    Average or mean value.
    S = mean(X) is the mean value of the elements in X if X is a vector.
    For matrices, S is a row vector containing the mean value of each
    column.
    For N-D arrays, S is the mean value of the elements along the first
    array dimension whose size does not equal 1.

    mean(X,DIM) takes the mean along the dimension DIM of X.

    S = mean(..., TYPE) specifies the type in which the mean is performed,
    and the type of S. Available options are:

    'double'    -   S has class double for any input X
    'native'    -   S has the same class as X
    'default'   -   If X is floating point, that is double or single,
                    S has the same class as X. If X is not floating point,
                    S has class double.

    S = mean(..., MISSING) specifies how NaN (Not-A-Number) values are
    treated. The default is 'includenan':
```
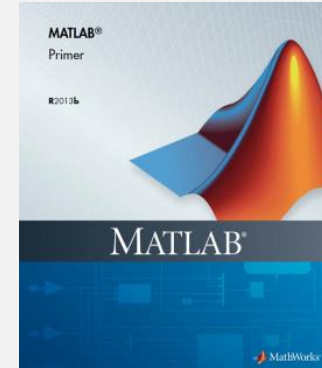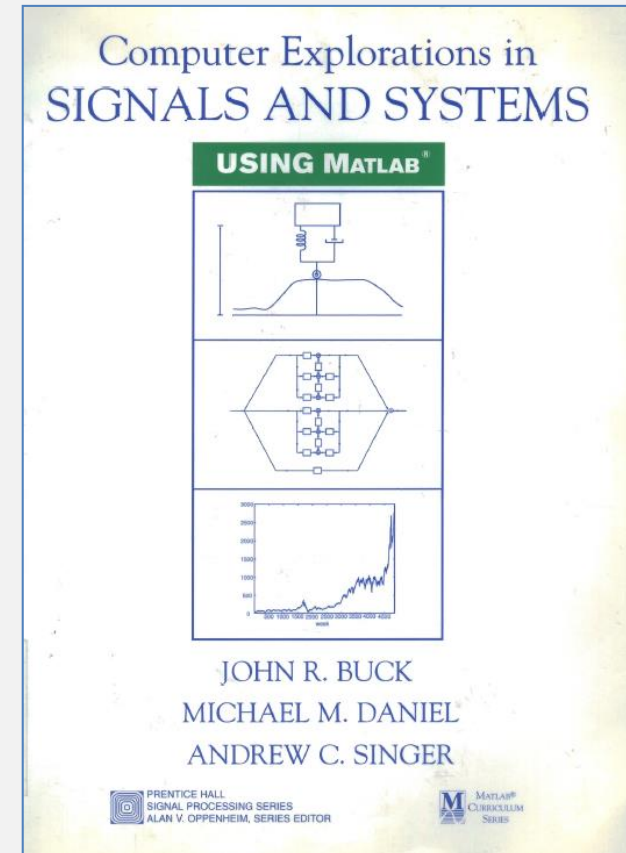
To Start with MATLAB

- Work through the built-in tutorial in MATLAB: "Getting Started"

- A short introduction: getstart.pdf

- A summary of university-authored MATLAB tutorials in http://www.mathworks.com/academia/student_center/tutorials/launchpad.html

Lab Assignments

- 1.4 & 1.5
- Submit your report.

- **Question ?**