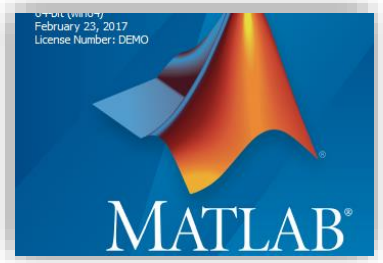
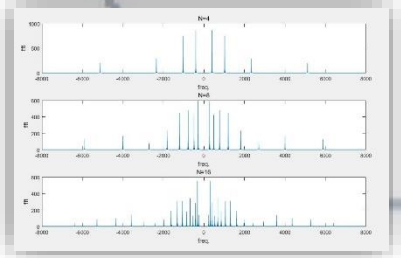
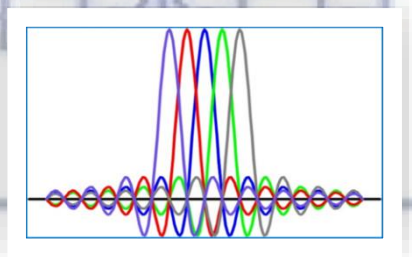
信号与系统实验

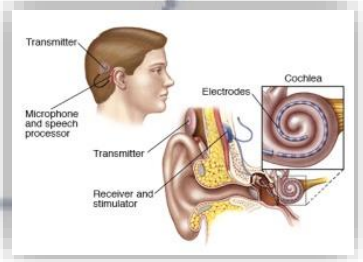
主讲人：吴光 博士

Email: wug@sustech.edu.cn









南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

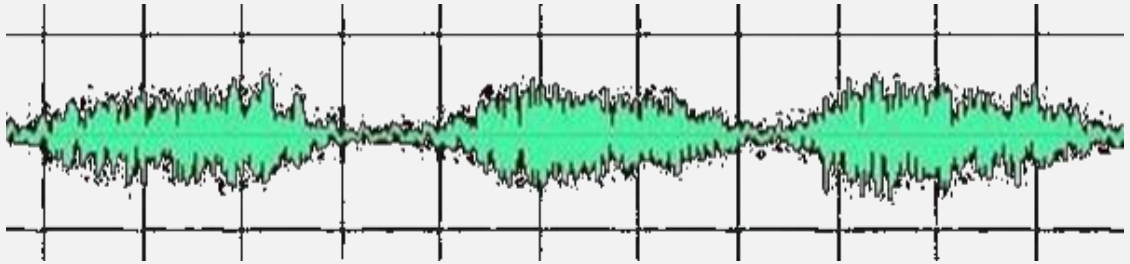
Signals and Systems (Lab)

Lab 1: MATLAB Programming

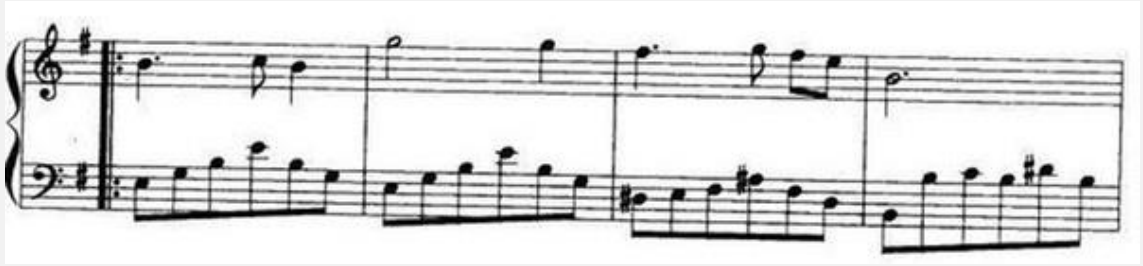
Dr. Wu Guang
wug@sustech.edu.cn
Electrical & Electronic Engineering
Southern University of Science and Technology

Part 1: Introduction

Objective: Analysis in frequency-domain



Time-domain

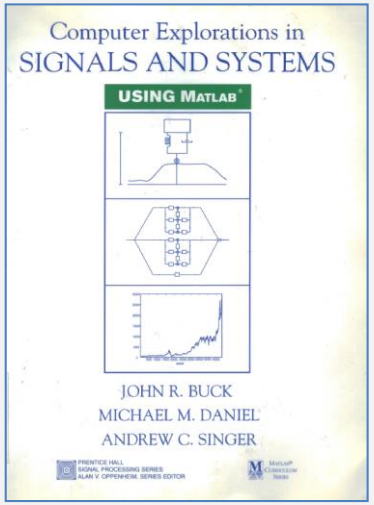


Frequency-domain

Labs in this course

📖 5 Lab assignments+2 Projects

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



Arrangements

📖 5 lab assignments

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

📖 2 projects

- Four students work as a group;
- Three weeks per project;
- Evaluation: Project Report + **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:

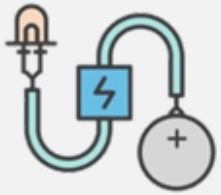
> Write **a short introduction** to the lab assignment
>
> Type down Question 1
> Give you answer to Q1, add the figures if necessary
>
> Type down Question 2
> Give you answer to Q2, add the figures if necessary
>
> Type down Question 3
> Give you answer to Q3, add the figures if necessary
> …

- Email to wug@sustech.edu.cn

File name: (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

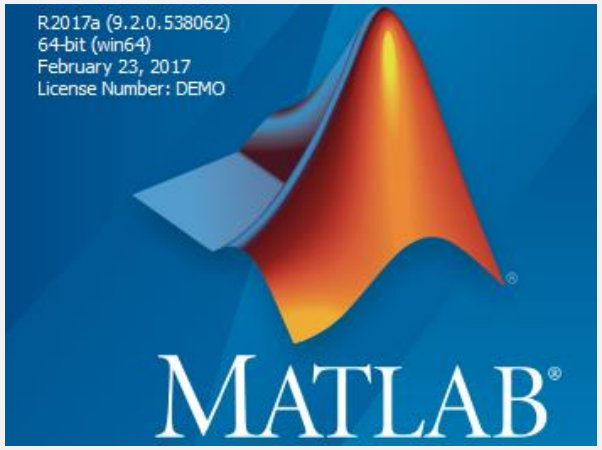
Derive the **math expression** of performance for Algorithm A & B

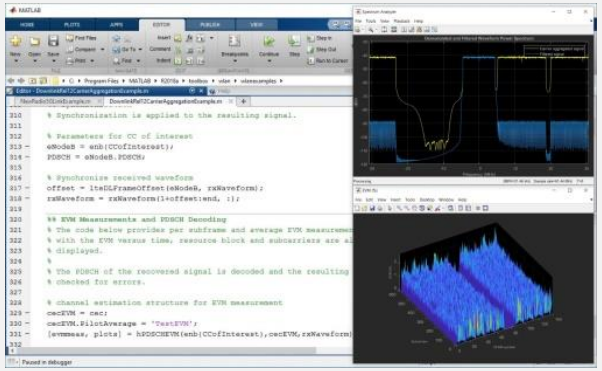
Solid math background is necessary

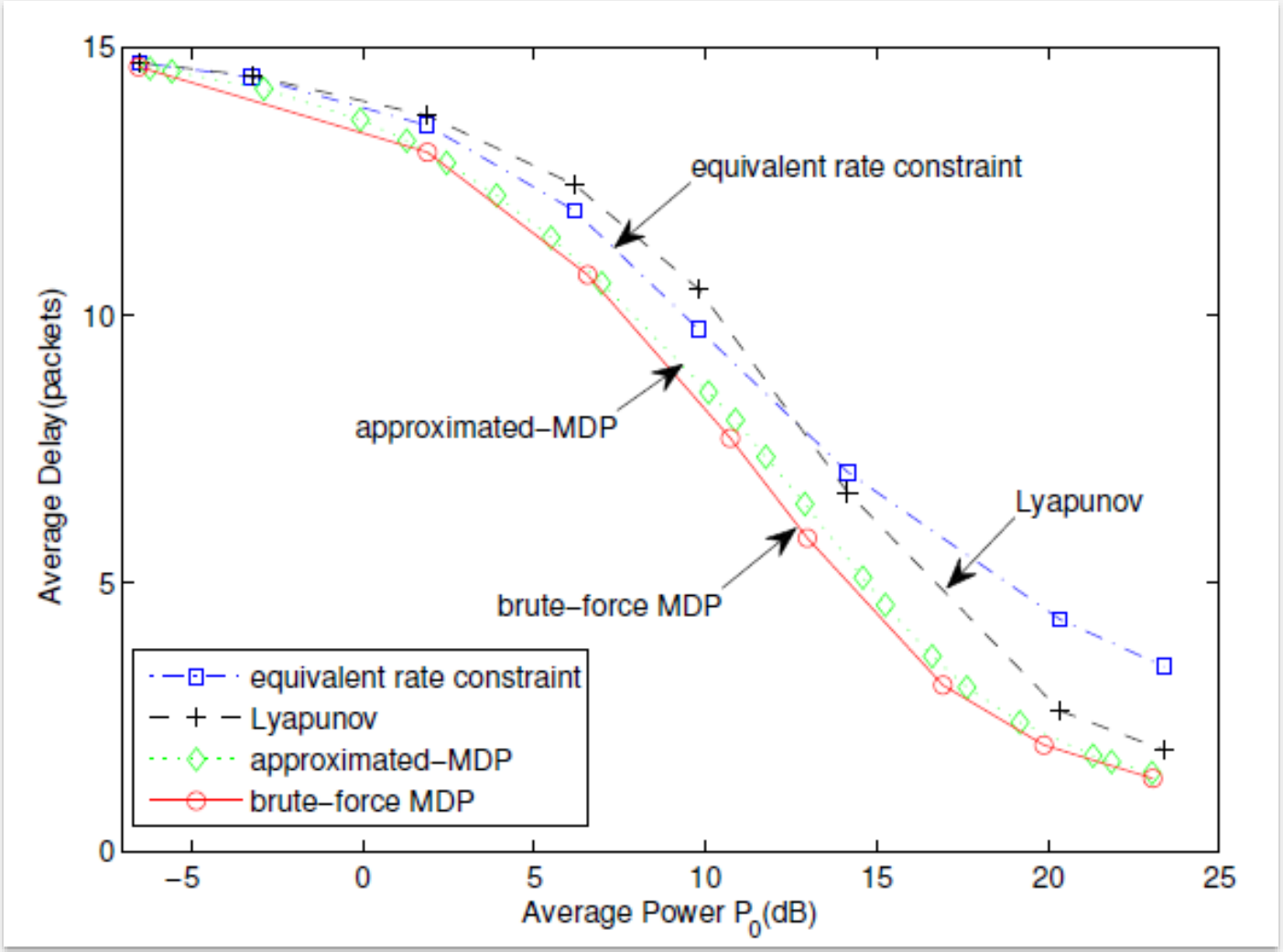
Test Algorithm A & B **by computer** with typical system configuration

You need to write program

How to express your result ?







南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Part 2: MATLAB Tutorial

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

What can you do with MATLAB ?

Computational Biology

Analyze, visualize, and model biological data and systems

Control Systems

Design, test, and implement control systems

Data Science

Explore data; build machine learning models; do predictive analytics

Deep Learning

Data preparation, design, simulation, and deployment for deep neural networks

Embedded Systems

Design, code, and verify embedded systems

Enterprise and IT Systems

Use MATLAB with your IT systems

FPGA, ASIC, and SoC Development

Automate your workflow — from algorithm development to hardware design and verification

Image Processing and Computer Vision

Acquire, process, and analyze images and video for algorithm development and system design

Internet of Things

Connect embedded devices to the Internet and gain insight from your data

Machine Learning

Train models, tune parameters, and deploy to production or the edge

Mechatronics

Design, optimize, and verify mechatronic systems

Power Electronics Control Design

Design and implement digital control for motors, power converters, and battery systems

Predictive Maintenance

Develop and deploy condition monitoring and predictive maintenance software

Robotics

Convert your robotics ideas and concepts into autonomous systems that work seamlessly in real-world environments.

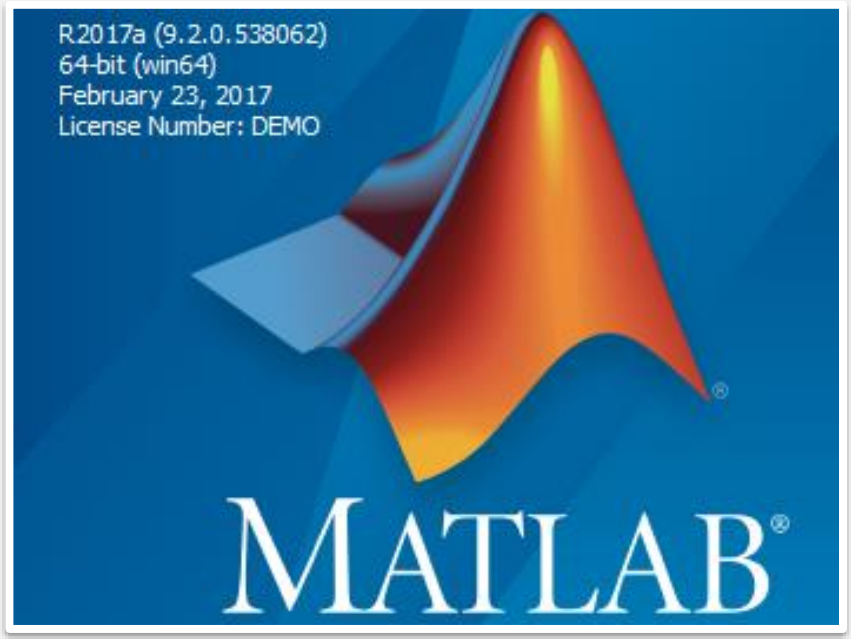
Signal Processing

Analyze signals and time-series data. Model, design, and simulate signal processing systems.

Test and Measurement

Acquire, analyze, and explore data and automate tests

MATLAB (MATrix LABoratory)

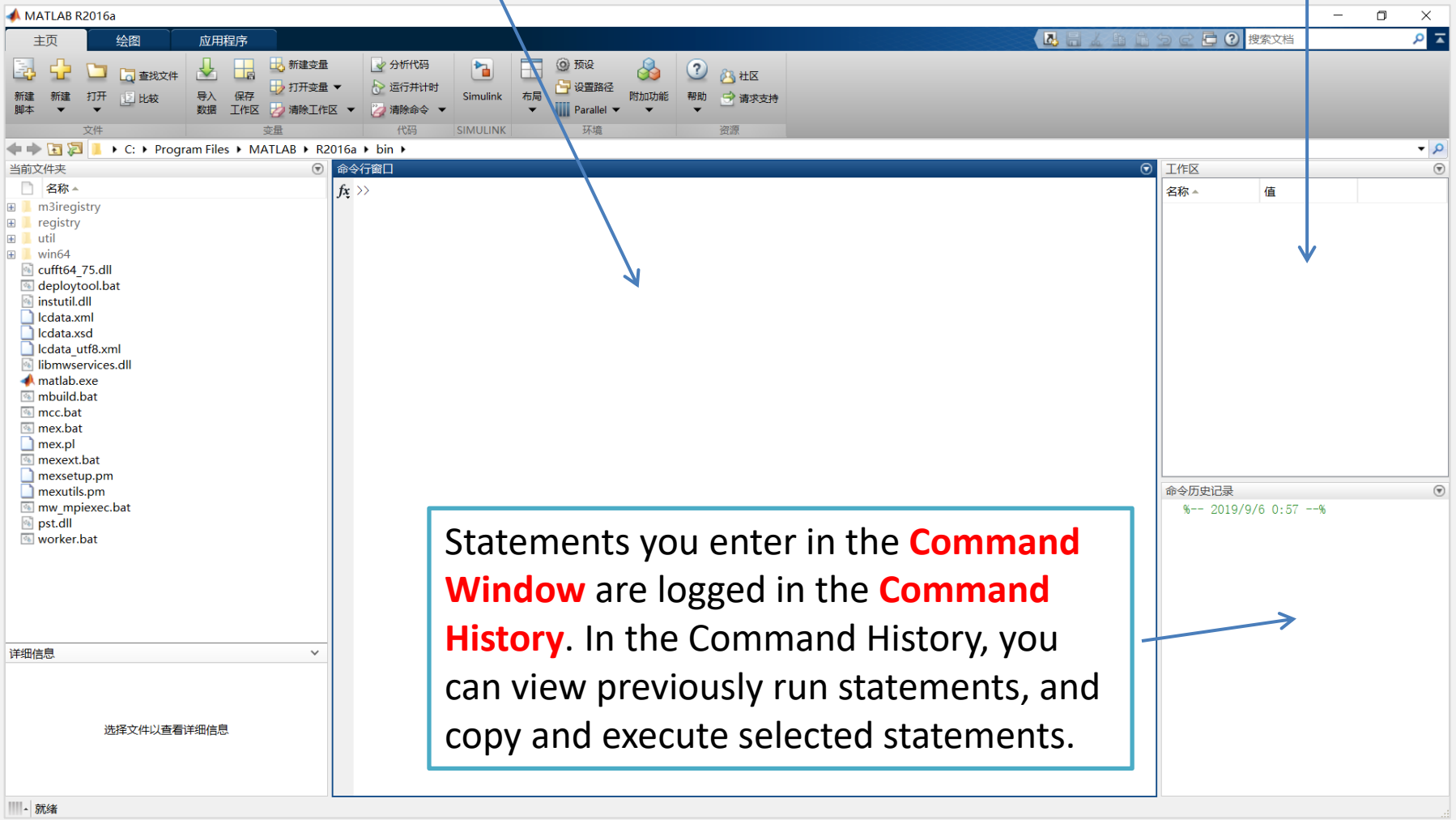


MathWorks®

IEEE acm

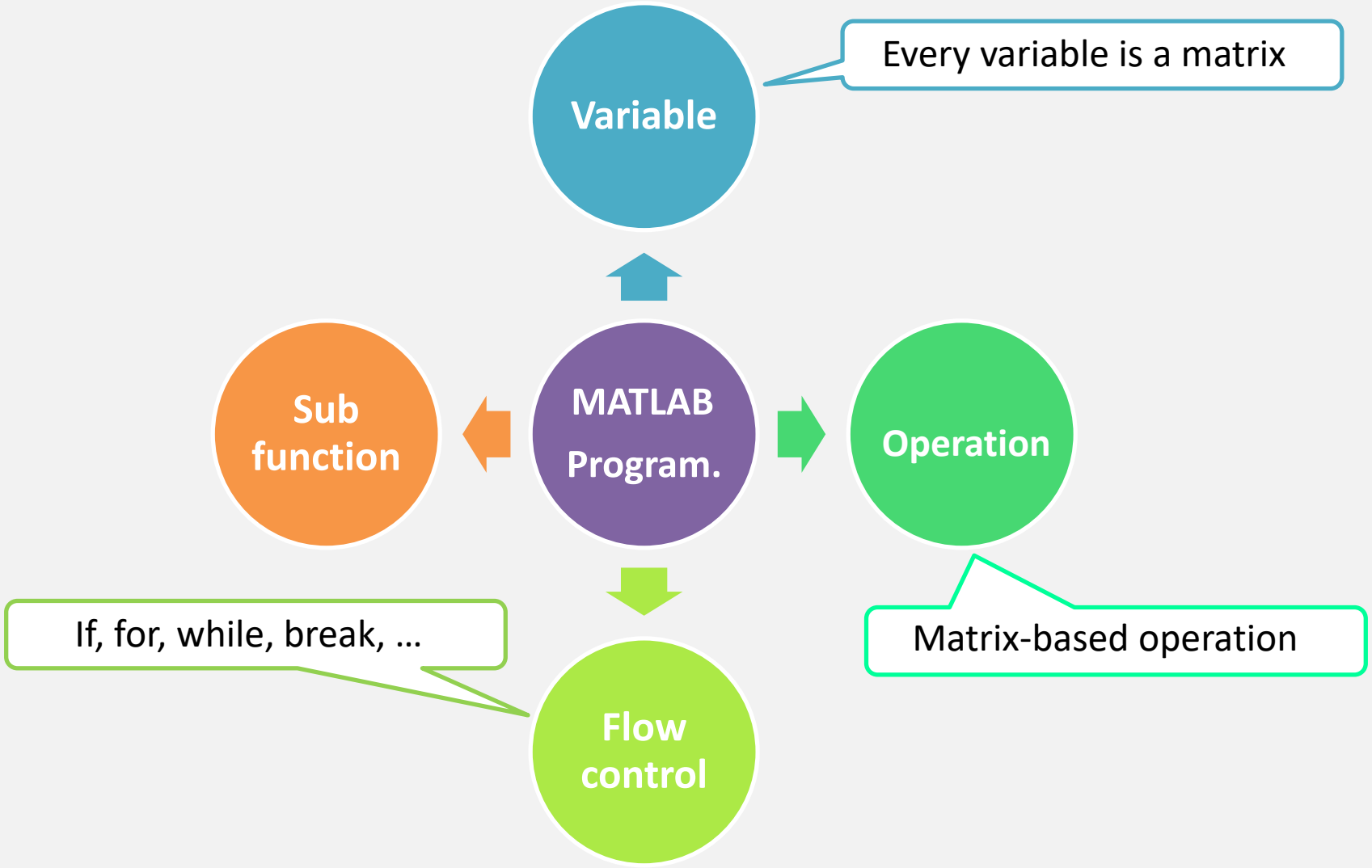
Type your MATLAB statement in **Command Window, you can interact with MATLAB here.**

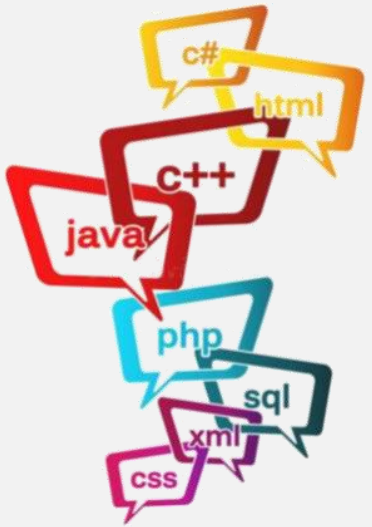
All currently defined variables will be stored listed out in the **Workspace**.



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

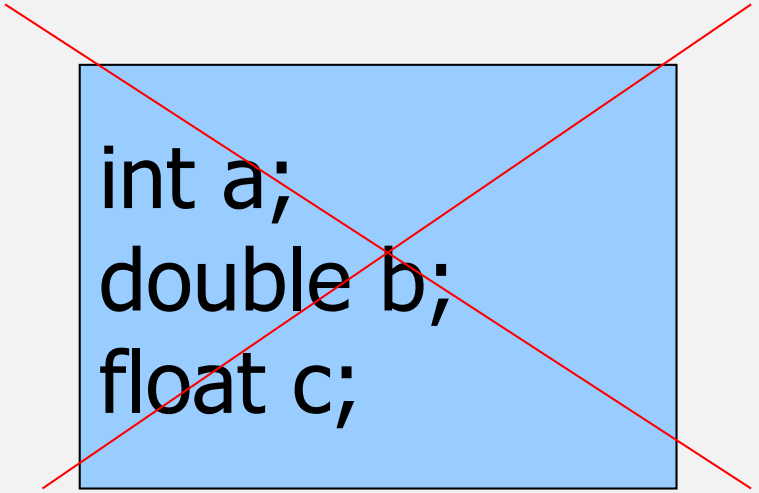
a='hello world'

a=5.5

b=10+1i

a=a*b

a=a+b



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]`

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

Row Vector

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

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

Matrix

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

```
z =
     1
     2
     5
     1
```

Column Vector

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
```

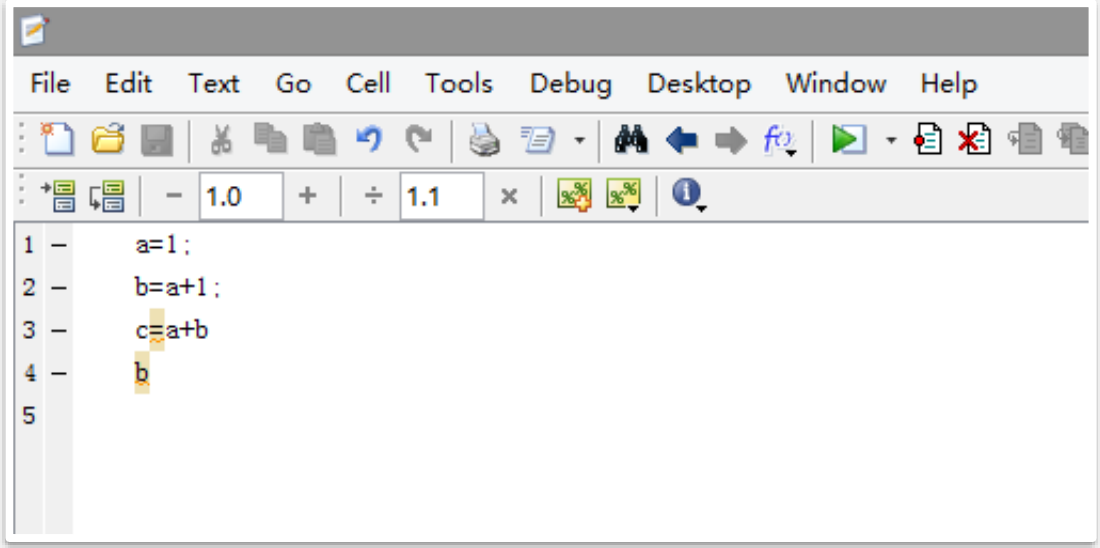
```
d = x .^2

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



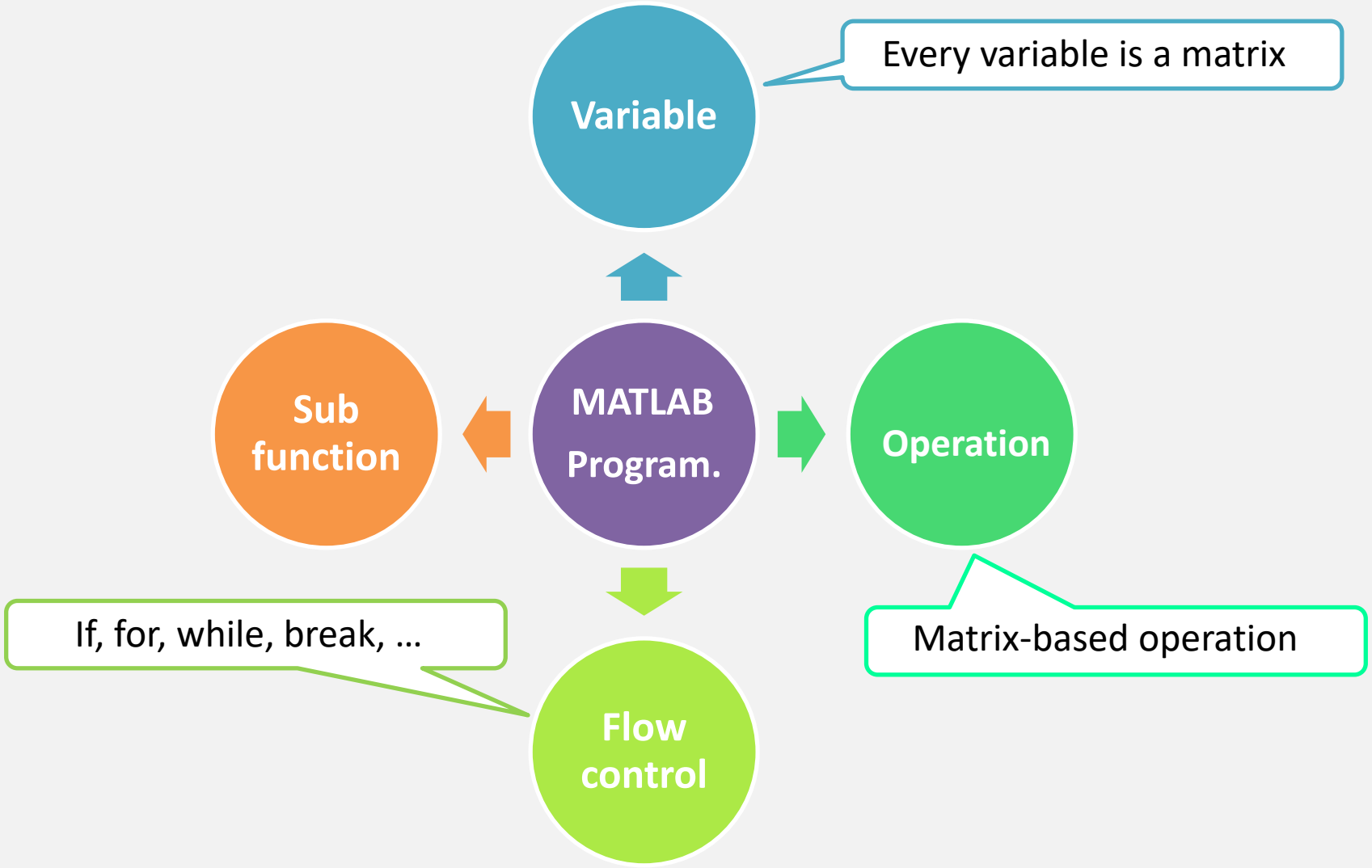
Instead of typing in the **Command Window**, you can write a script in **Editor** and save in a **.m file**, like test.m.

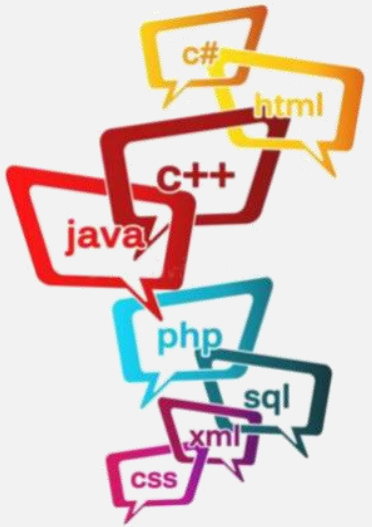
```
>> test

c =

     3


b =

     2
```

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

Language Overview

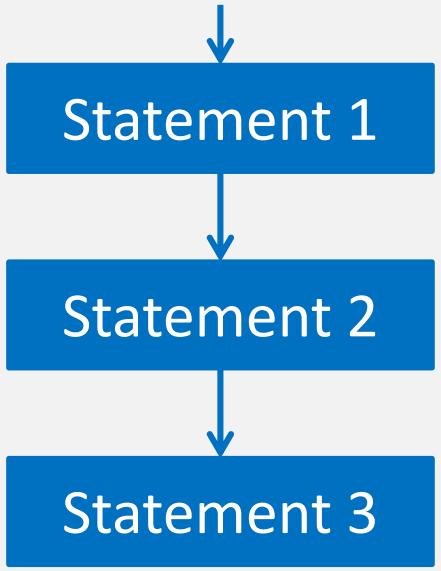




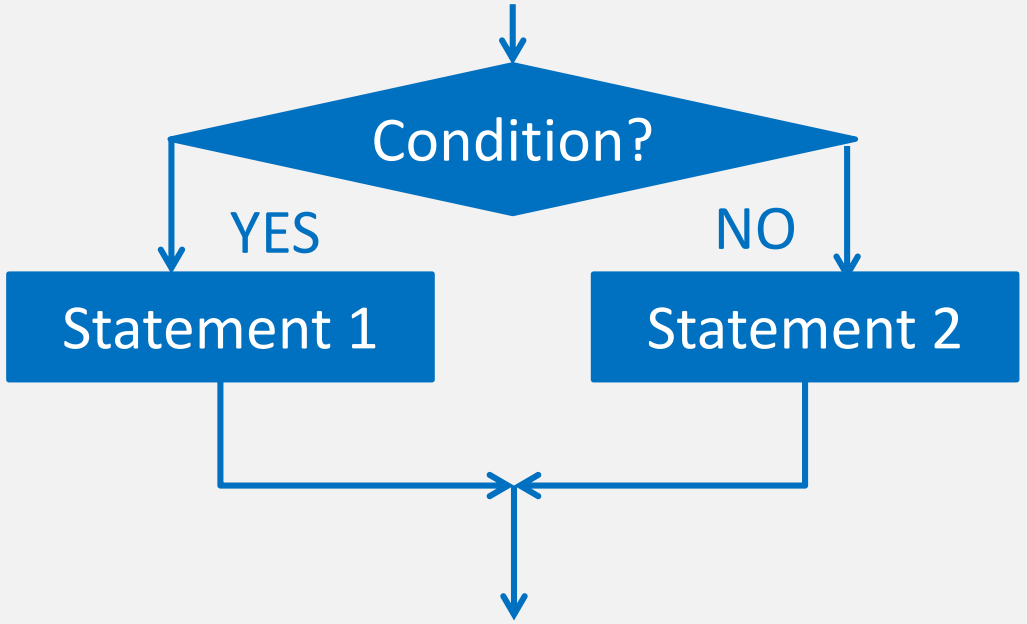
Flow Control

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

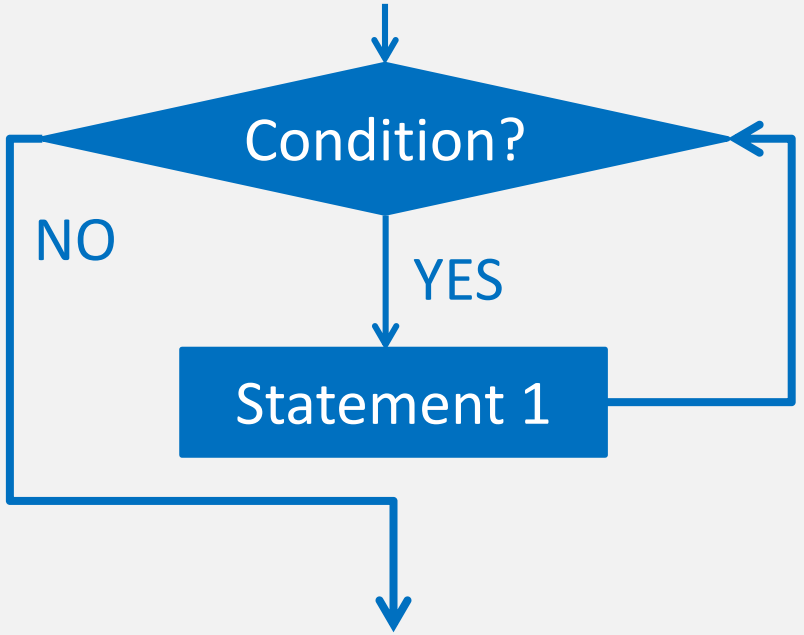
- Sequence, Selection and Repetition



Sequence



Selection: if



Repetition: for & while

```
if (Condition_1)
        MATLAB Commands
elseif (Condition_2)
        MATLAB Commands
elseif (Condition_3)
        MATLAB Commands
else
        MATLAB Commands
end
```

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

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

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

```
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

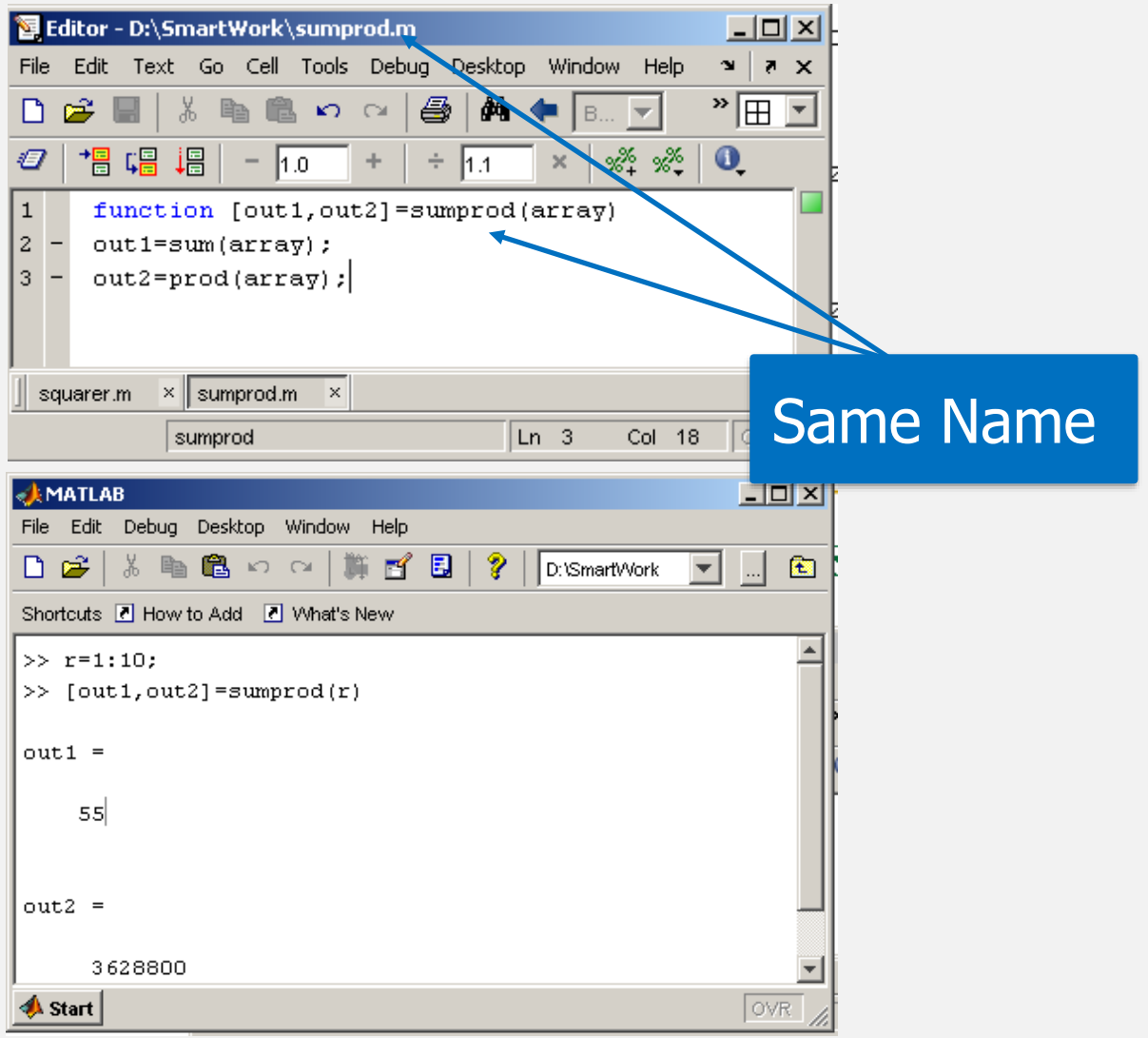
```
function out1=functionname(in1)
function out1=functionname(in1,in2,in3)
function [out1,out2]=functionname(in1,in2)
```

- 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



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 $0 \leq x \leq 4\pi$

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

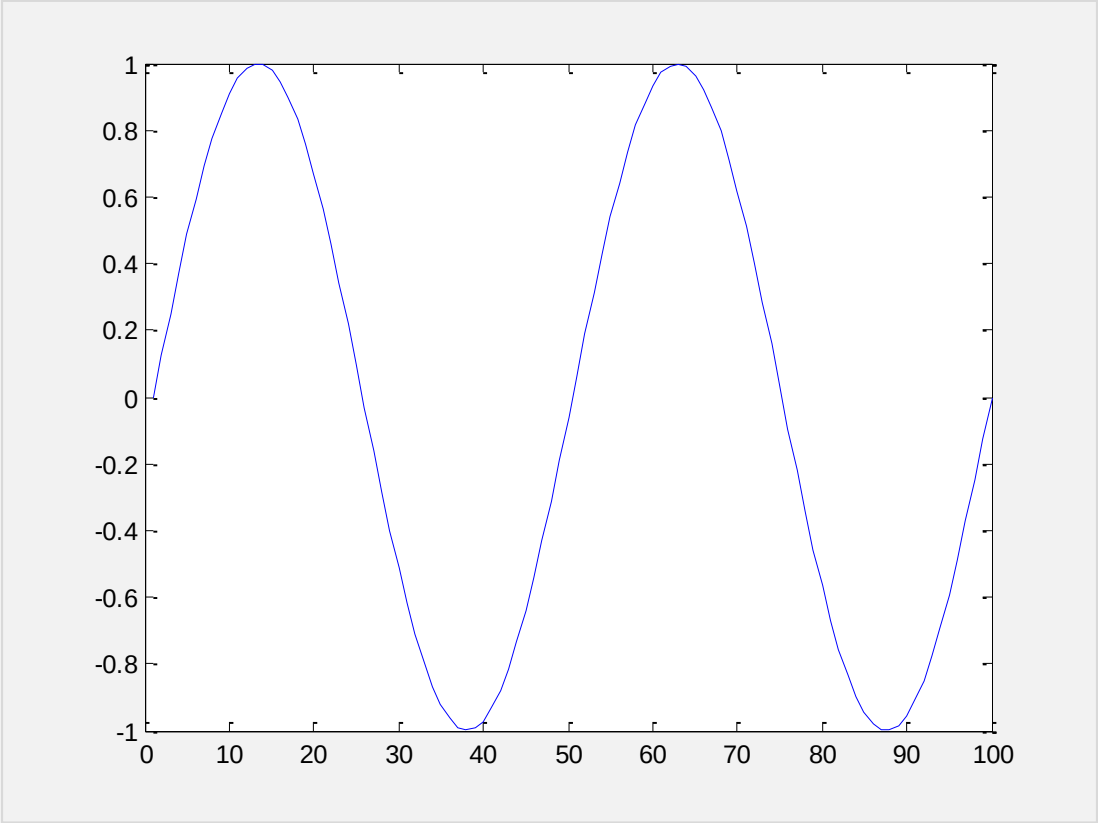
```
>>x=linspace(0,4*pi,100);
```

- Calculate sin(.) of the x-array

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

- Plot the y-array

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



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

- 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;
```

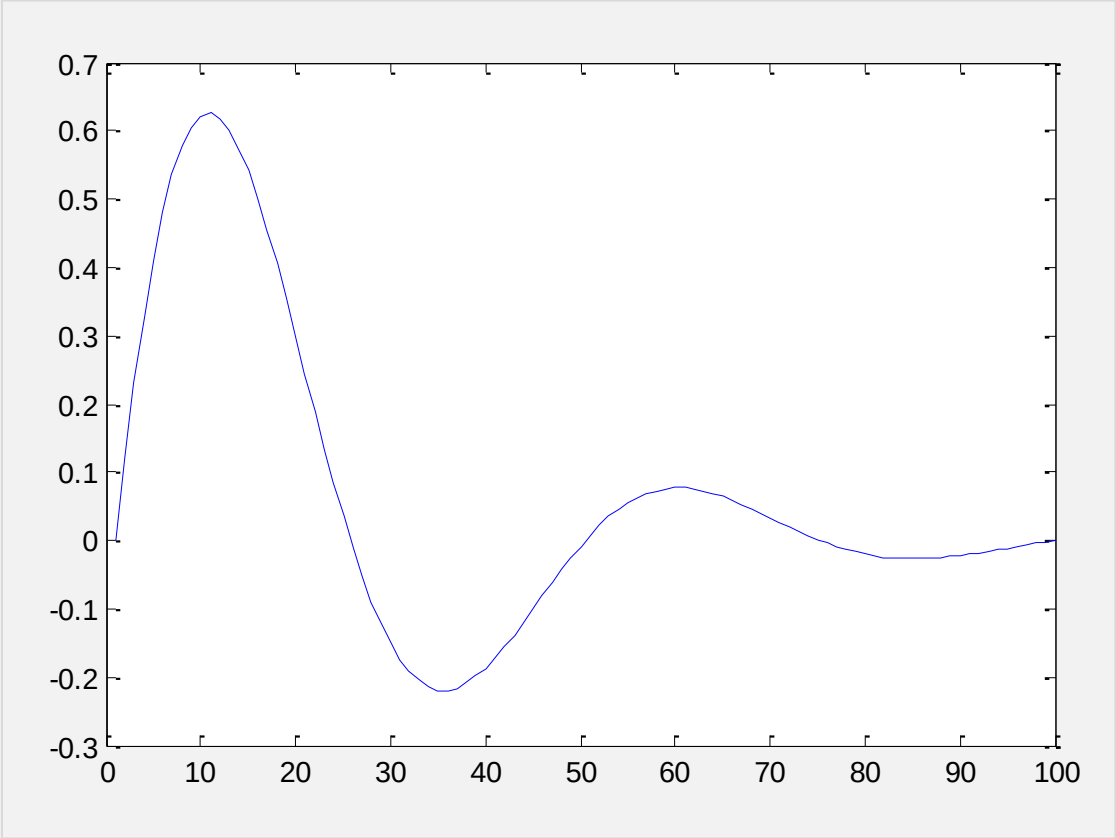
Plot the signal $e^{-x/3}\sin(x)$ between $0 \le x \le 4\pi$

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

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

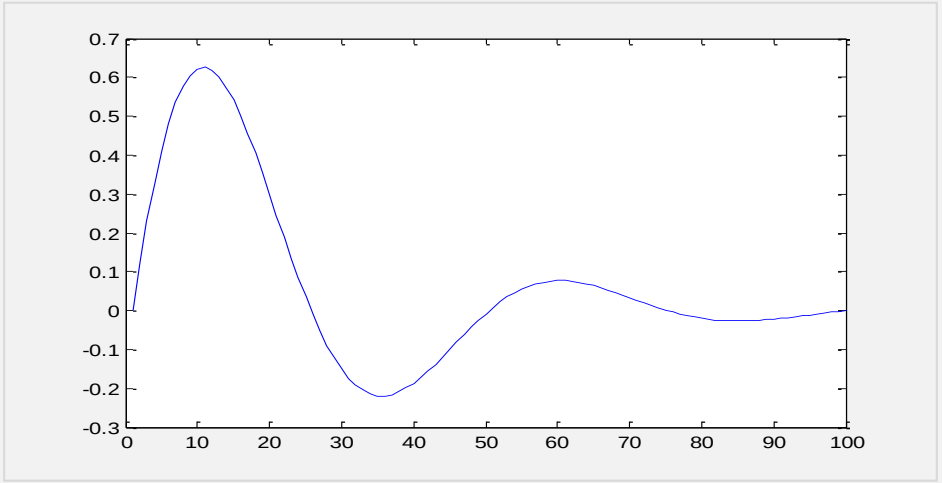
- Plot the y2-array

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

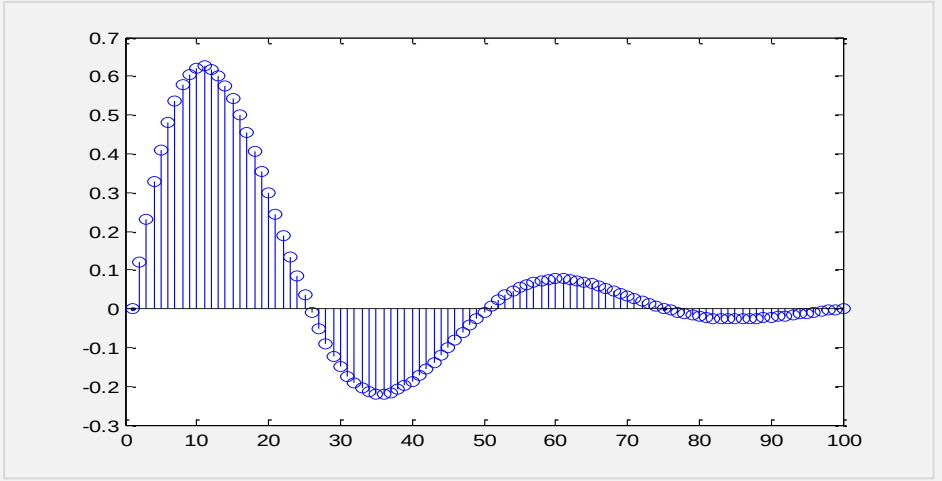


Plot and Stem

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



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

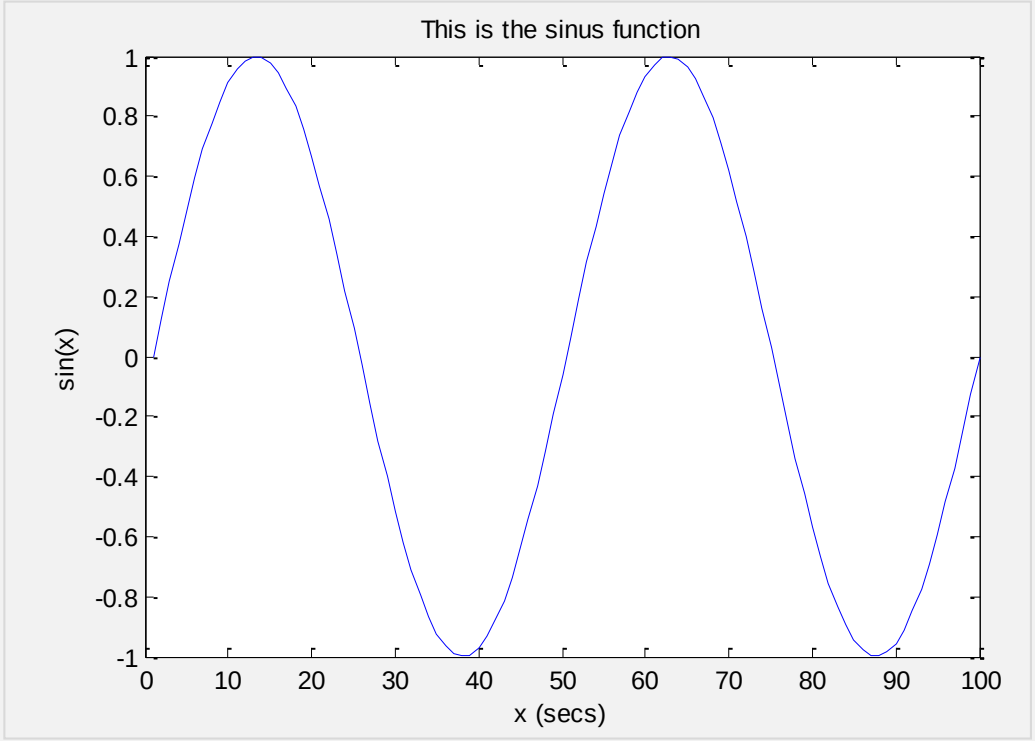


title,xlabel and ylabel

```
>>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		

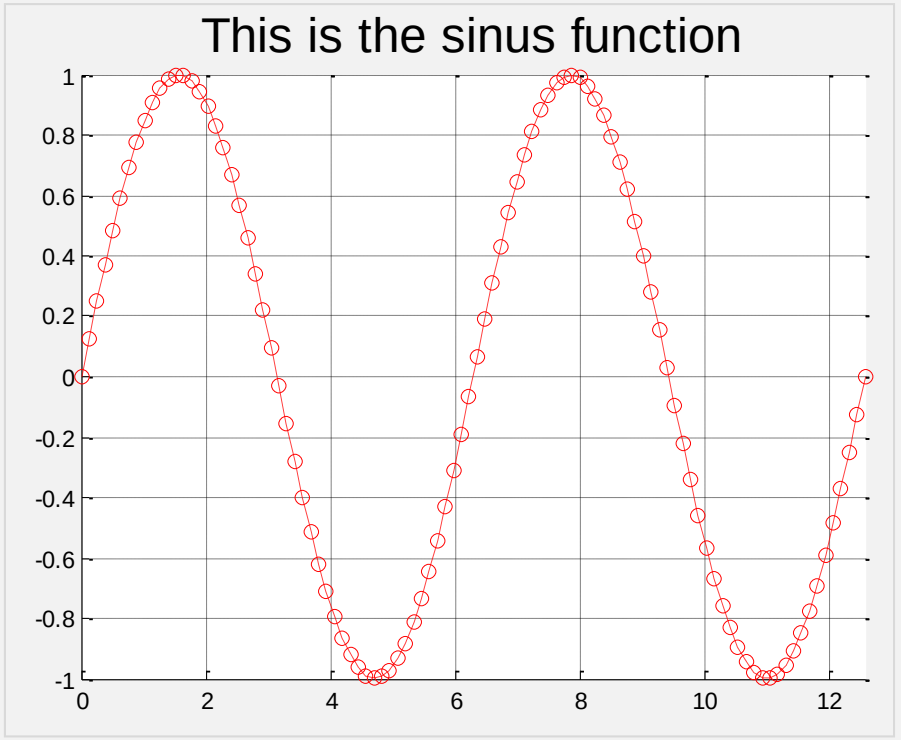
Colors

Symbols

Line types

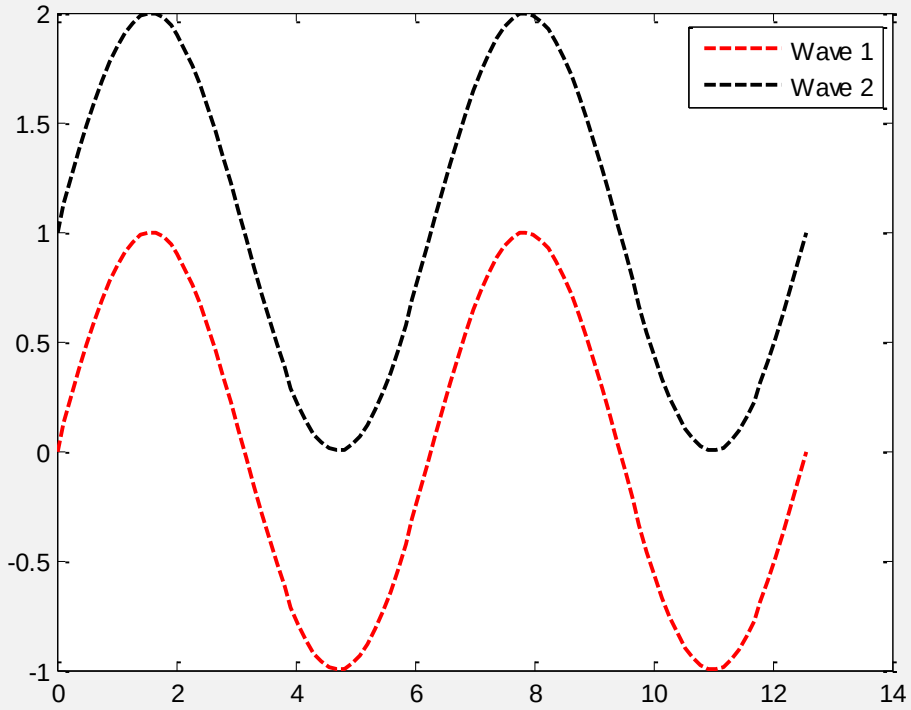
Example

```
>> plot(x, y, ‘r--o’)
>> title('This is the sinus function','fontsize',20)
>> grid on
>> box off
>> axis([0 4*pi ylim])
```



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’)
```



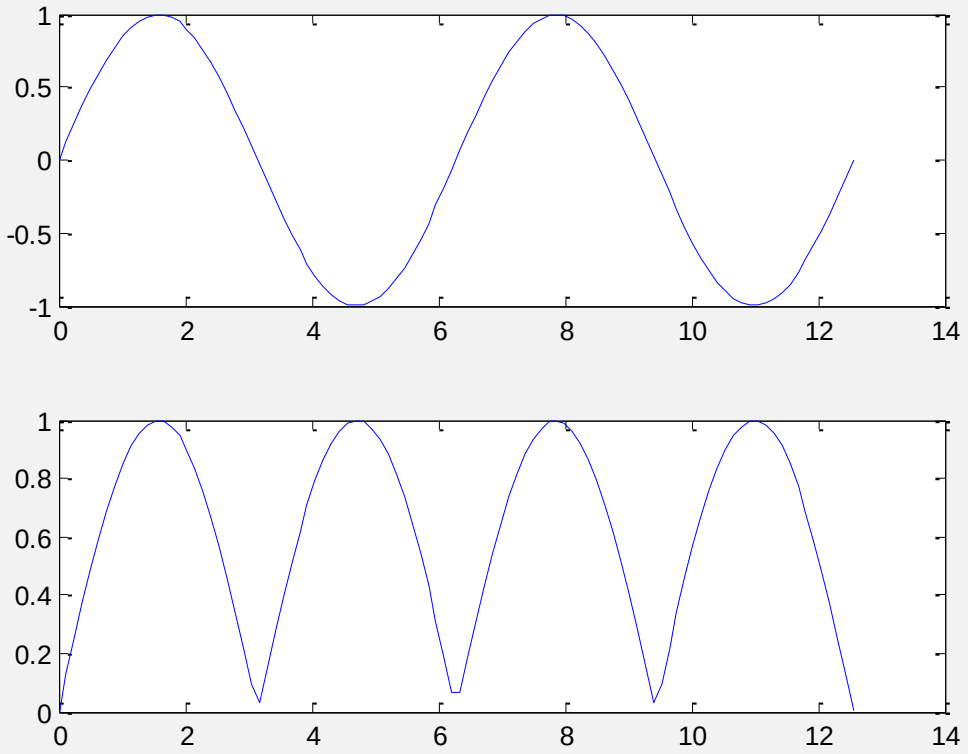
More figures

- **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))
```



Some useful functions

- length(.)
- size(.)
- abs(.)
- sum(.)
- mean(.)
- std(.)
- diff(.)

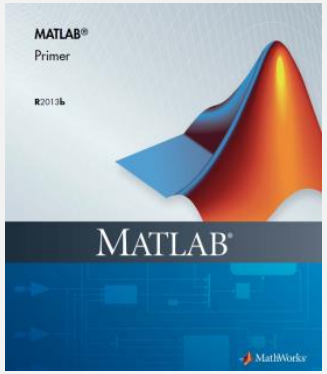
命令行窗口

```
>> 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”

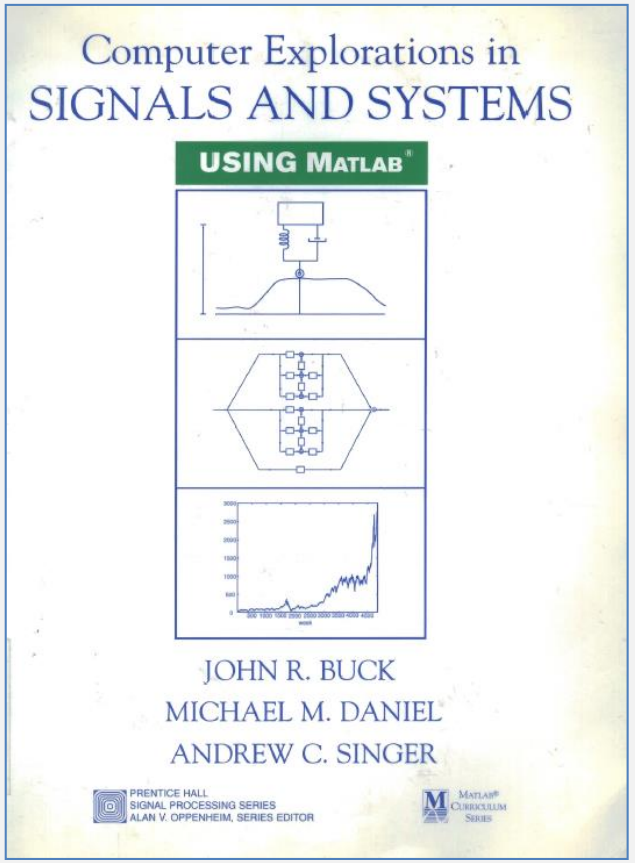
- A short introduction: getstart.pdf



- 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 ?