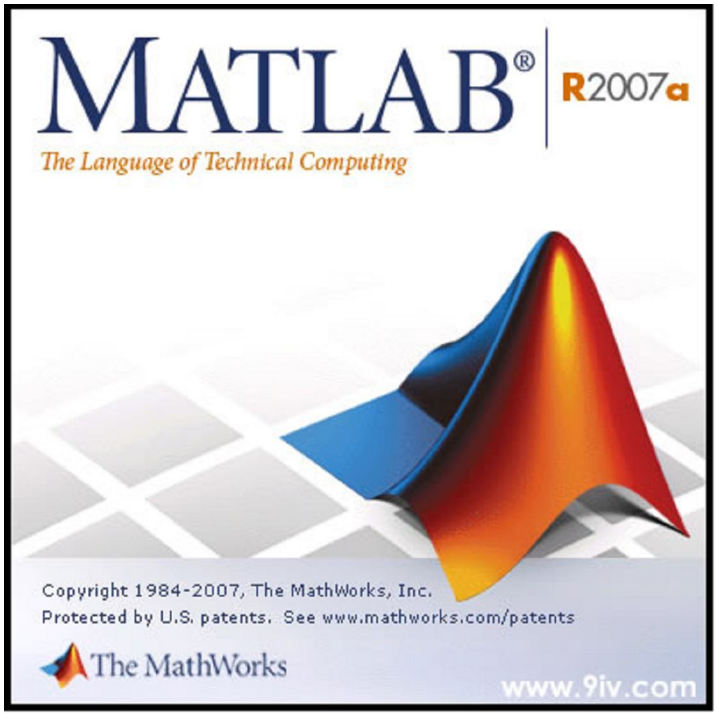
Introduction to MATLAB



Outline:

- ✓What is Matlab?
- ✓Matlab Screen
- ✓Variables, array, matrix
- ✓ Operators
- ✓Display Facilities

What is MATLAB

✓**MATLAB is** a platform for the scientific calculation, data processing and their display

✓**www.mathworks.it**

✓The name MATLAB is an abbreviation for **MATrix LABoratory**.

✓The **basic data structure is the matrix** which means that, when processing, any amount from the calculation is treated as an array of size nxm.

✓MATLAB is a high-level interactive environment that allows you to easily build and manage **arrays**, and, as special cases, **vectors and scalars**.

(a vector is a matrix 1xn, while a scalar is treated as a 1x1 matrix).

Why Use MATLAB?

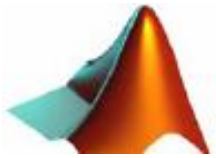
➢It is used in scientific research and resolution of engineering problems

➢MATLAB is a useful tool for:

✓Developing Algorithms
✓**Data Analysis** (Curve fitting)
✓Data Visualization
✓Designing Graphical User Interfaces
✓**Numeric Computation** (Optimization and numerical integration, Ordinary differenzial equations (ODEs), Partial differential eqautions (PDEs))

➢There are versions for all platforms:

Windows, Macintosh, UNIX, CRAY.
There is also a Student Edition.



How to Start MATLAB?

- **To start MATLAB** from Windows just double-click the mouse on MATLAB icon.

- The window that appears when you run is called the **MATLAB desktop**.

- The **prompt symbol >>** indicates that the computer is ready to receive instructions and to execute them.

- To exit, just type
  >> Quit

MATLAB screen

Toolbar (menu and icons)

Selecting the folder

Current Directory

✓View folders and m-files

Workspace

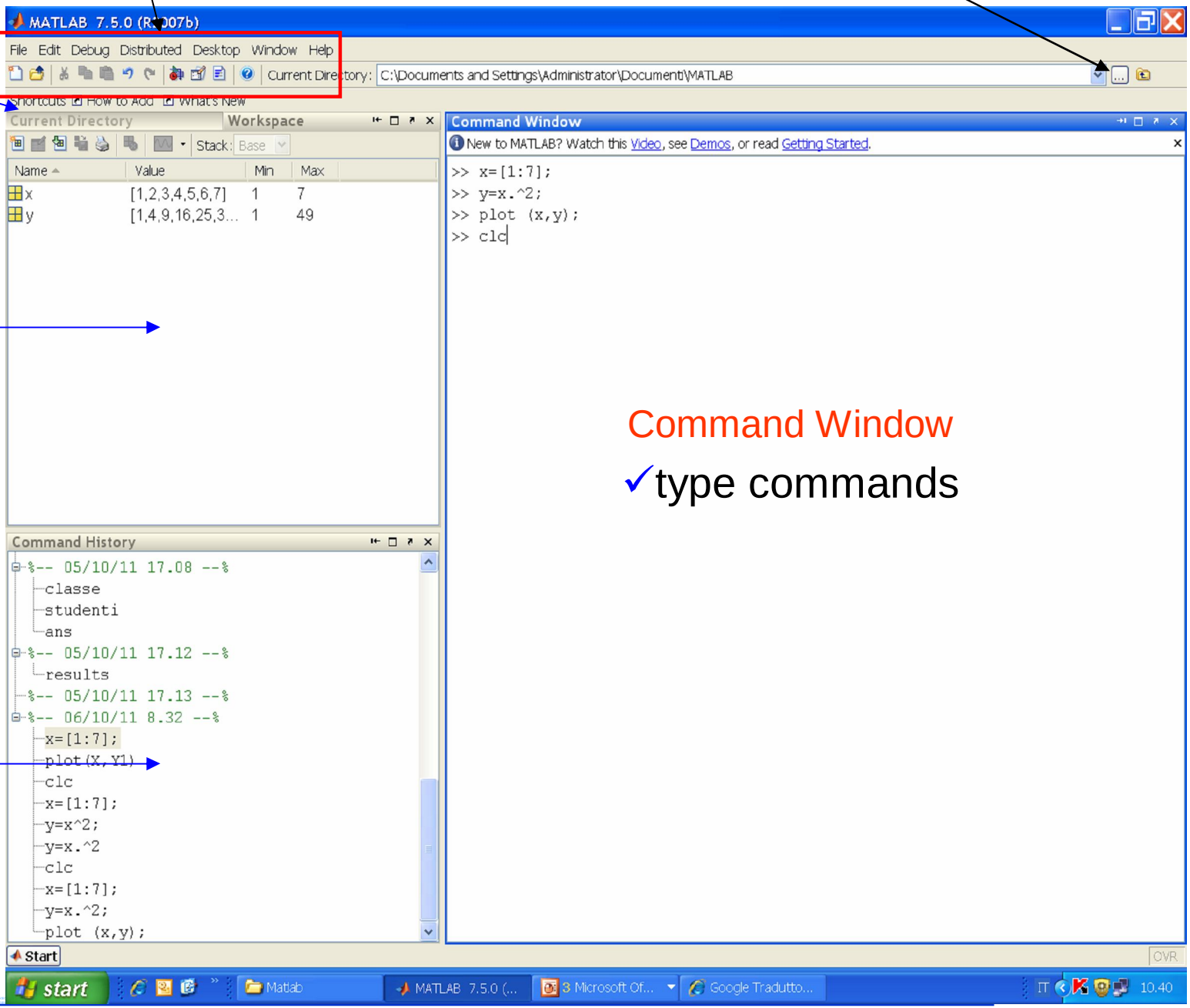
✓View program variables

✓Double click on a variable to see it in the Array Editor

Command History

✓view past commands

✓save a whole session using diary



Command Window

✓type commands

Entering commands

✦The first mode to use MATLAB is to enter commands directly into the Command Window after the prompt (>>)

Example:

>> Fifteen
the puzzle game to 15 numbers

>> Wrldtrv
calculates the air distances between cities in the world

Try the command **demo**!

✦ MATLAB as a calculator

From the prompt you can also enter expressions

```
>> 2 +1
ans =
3

>> Log (4)
ans =
1.3863
>>
```

Using the command “format” is possible to select the display format of the result

Entering commands

✓**Use the arrows** (↑ or ↓) to scroll (up or down) through the commands previously typed on the command window

✓Use of **Command History** to repeat commands you have already typed (select the text of the command with the mouse, and holding the left mouse button drag it to the command window

✓**Use the Command clc** to clear the command window

The display format

- **Example:**

```
>> y = 9 / 7
```

>>**format short**: 5 digits (4 decimal) (standard or default)

```
 1.2857
```

>>**format long**: 16 digits

```
 1.285714285714290
```

>>**format short e**: 5 digits (4 decimal) plus the exponent

```
1.2857e +000
```

>>**format long e**: 16 digits (15 decimal) plus the exponent

```
 1.285714285714286e +000
```

>>**format bank**: 2 decimal places

```
1.29
```

>>**format rat**: Rational approximation

```
9/7
```

The variables

- The result of an expression is always assigned to a variable

- A variable is a symbol used to contain a given data

```
>> 2 +1
ans =
3
```

- When we use MATLAB as a calculator it assigns the result of an expression to a variable called **ans** (abbreviation of answer)

The variables

✣We can now use variables to store a data and use them in subsequent calculations

Example:

```
>> 4 / 2
ans =
2


>> ans * 3
ans =
6
```

Definitions of variables

➢**We can define variables** with names chosen by us to which we assign values

Example:

```
>> r = 8 / 10
r =
0.8000
```

➢The variables that we define remain in memory until we do not finish the MATLAB session (by closing the program).

➢The workspace shows the names and values of all variables used in the current work session with Matlab

Example:

```
>> r
r =
0.8000
```

Definitions of variables

- In general the name of a variable is associated to an entity (scalar, vector, matrix) that contains data.

- MATLAB does not require any declaration of the variables (i.e.: type, size, ...)

- The names chosen must comply with the following **syntax rules**:

  - ✓It can contain only letters, numbers and the underscore character ("_");

  - ✓Variable names must start with a letter and can not be longer than 32 characters;

  - ✓You can not use reserved words of MATLAB.

MATLAB is "case-sensitive language", which distinguishes between uppercase and lowercase:
A is a different variable from a.

Possible values for a variable

✦What values can be assigned to a variable?

- single number (even complex)
- string
- vector
- matrix
- ...

The assignment operator

✦ When we define a variable, we assign a value to it through the **symbol =**

```
>> a = 6
```

✦ The value in A does not change until we change the value with another assignment

```
>> a
a =
        6

>> a * 5
ans =
        30

>> a = 7
a =
        7
```

The assignment operator

variable = expression

✦ The variable on the left side of the operator is replaced by the value created by the expression written on the right side

Example:

```
>> s = 3 + 4-2
s =
        5
```

Correct assignments

```
x = 3
x = x + 3
y = x + 2
z = y * 4
```

Incorrect assignments

```
3 = x
x + 2 = 20
```

x = 5 + y (if y has not been previously defined)

Basic operators

Symbol	Operation	Format
^	Exponentiation: a^b	a ^ b
*	Multiplication: ab	a*b
/	Division on the right: a / b = a:b	a/ b
\	Left division: a \ b = b:a	a \ b
+	Addition: a + b	a + b
-	Subtraction: a - b	a-b

Priority Rules

✦ MATLAB follows the same rules used by the major programming languages

Level of priority	Operation
1	Brackets: they are evaluated starting from the most internal couple
2	Exponentiation:
3	Multiplication and Division with the same priority: evaluated from left to right
4	Addition and Subtraction with the same priority: evaluated from left to right

Exercise

Use Matlab to calculate the following expressions:

a) $x = 6\frac{10}{13} + \frac{18}{5(7)} + 5\left(9^2\right)$

b) $y = 6\left(35^{1/4}\right) + 14^{0.35}$

Solutions:

➡ a): 412.2835
➡ b): 17.1124

Example: Volume of a circular cylinder

✦The volume of a cylinder of height h and radius r is given by $V = \pi r^2 h$.

✦ A cylindrical tank is 15 meters high and has a radius of 8 meters.

✦ We want to build a new cylindrical tank with a volume 20% greater than the previous but with the same height as the existing one.

✦ What radius will have the new tank?

Solution

✦ The value of the radius is given by: $r = \sqrt{\frac{V}{\pi \cdot h}}$

✦ assigns the values to the variables that represent the radius and height

>> r = 8

>>h = 15

✦ calculates the volume of the first cylinder

>>V = 3.14 * r ^ 2 * h (3.0159 e+003)

✦ increases the volume V by 20%

>>V= V + 0.2 *V

(3.6191 e+003)

✦We obtain the radius of the new cylinder

>>r = sqrt (h * V/3.14) (131.4534)

Commands to manage a work session

Command	Description
clear	Delete all variables from memory
clear var1, var2	Delete the variables var1 and var2 from memory
exist ('name')	Determine if a file or a variable have the 'name' specified
quit	Closes matlab
who	List the variables that reside in the workspace
whos	List the variables and the space in memory It is what is shown graphically in the window Workspace
,	Separate instructions
;	Does not display the result of an instruction and separate the rows of an array
:	Generates a vector of elements at regular intervals
...	Allows to continue an instruction in the next line

Example

```
>> clc

>>a = 3
a =
        3

>> b = 5
b =
        5

>> who
Your variables are:
        a  b

>> Whos
Name   Size     Bytes   Class    Attributes
  a      1x1        8    double
  b      1x1        8    double
```

Examples (2)

```
>> Clear a

>> Whos
Name   Size     Bytes   Class    Attributes
   b      1x1         8       double

>> Exist ('b')
ans =
        1

>> Exist ('a')
ans =
        0
>>
```

How to Save a Work Session

✓Each session can be saved in a binary file (filename.mat) using the command:

>>**save <filename>**

and loaded in the MATLAB environment using

>>**load <filename>**

The variables used in each workspace can be found simply by typing the name of the variable itself.

Predefined Constants

- **eps:** Specifies the precision of decimal numbers
- **i, j:** imaginary unit
- **Inf:** infinity
- **NaN (Not a Number):** Indicates an undefined numeric result
- **pi:** the number π (3.1416)

Commands, Instructions, Functions

✦ **Instructions.** They can not have arguments

❖ quit, clc

✦ **Commands.** They do not require arguments, but may have, and when they argumets, they are not enclosed in brackets

❖ clear x

✦ **Functions.** They require arguments that must be enclosed in brackets

❖ sqrt (x)

MATLAB has a large number of standard mathematical functions such as:

Abs, sqrt, exp, sin, Log, Log10,...

For a list of elementary mathematical functions, type:

>>**help elfun**

For a list of more advanced mathematical and matrix functions, type:

>>**help specfun**
>>**help elmat**

How to build a vector?

Row vector

```
>> r = [2 4 10]
r=
2.0000 4.0000 10.0000

>>  s= [2, 4,10]
s=
2.0000 4.0000 10.0000
```

Coloumn vector

```
>> g = [3; 7; 9]
g=
        3
        7
        9

>>g = [3, 7, 9]'   (transposed vector)
g=
        3
        7
        9
```

How to build a vector?

✓The operator "**:**" allow to generate an extended array of elements at regular intervals

>>x = [m:q:n]
m is the first element of the vector
q is the constant increment
n is the last element of the vector if m-n is an integer multiple of q

>>x=[0:2:8]
x is the following vector: x= [0, 2, 4, 6, 8]

✓Command **linspace** **create a row vector with elements liearly separate**

>>linspace = (x1:x2:n)

x1 is the first element of the vector
x2 is the last element of the vector
N is the number of element

>>linspace = (5:8:31)
Is equivalent to x=[5:.1:8]

>>linspace = (5:8)
Is equivalent to x=[5:8:100]

Size, length and sum of a Vector

```
a=[2:3:25];


>>size(a)

ans=
        1    8

>>length(a)
ans=
        8

>>sum(a)

ans=
        100
```

How to Build a Matrix?

```
>>A = [2,4,10;16,3,7]
ans=
        2  4  10
       16   3   7
```

NOTE: " space" or "," separate the elements of each coloumn, while ";" separate the elements of each row.

To refer to a specific element of a matrix:

```
>>A(2,2)
ans=
        3
```

Use ":" to refer all the row or all the coloumn

```
>>A(1,:) first row, all coloumn
ans=
        2  4  10
>>A(:,2)  all row, second coloumn
ans:
        4
        3
>>A(:,2:3)   all row, coloumns from second to third
Ans=
        4  10
        3   7
```

Matrix Index

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

Given:

```
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.

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
```

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
```

Operators (Element by Element)

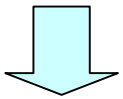
.* element-by-element multiplication

./ element-by-element division

.^ element-by-element power

The use of “.” – “Element” Operation

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

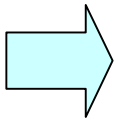


```
x = A(1,:)

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
```

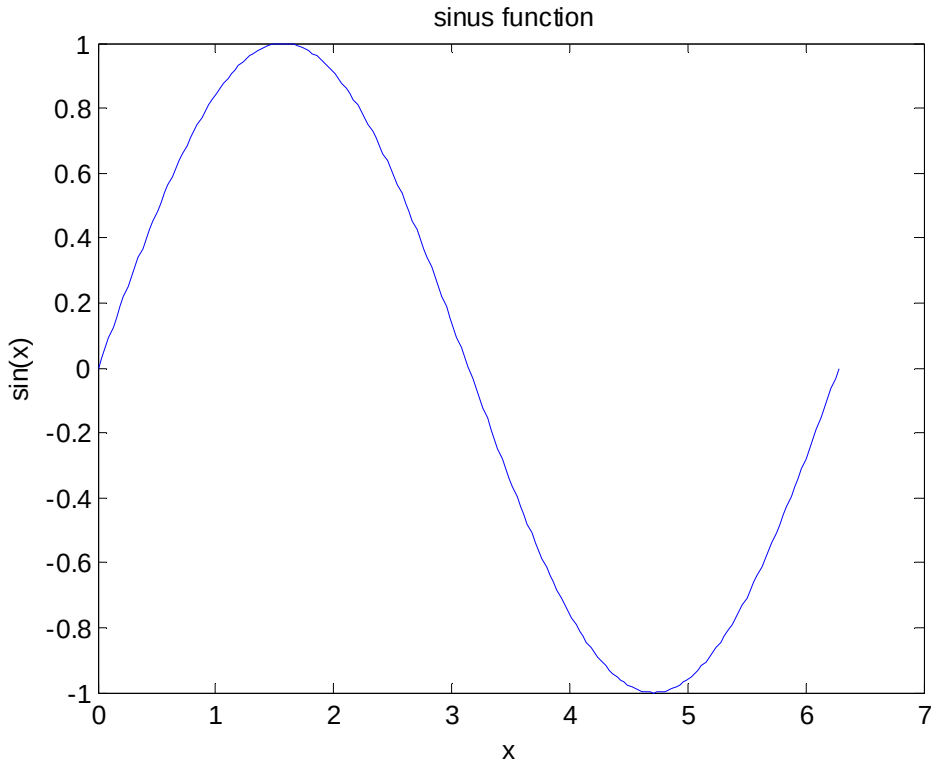
```
K= x^2
Erorr:
 ??? Error using ==> mpower  Matrix must be square.
B=x*y
Erorr:
??? Error using ==> mtimes Inner matrix dimensions must agree.
```

Example: Plot the function sin(x) between 0≤x≤2π

```
>>x=[0:pi/100:2*pi];
>> y1=sin(x);
>> plot(x,y1)
```

How to add lebels to the graph?

```
>> plot(x,y1), xlabel('x'), ylabel('sin(x)'), title('sinus function')
```



Plot the function sin(x) between 0≤x≤2π

How to add **symbol-style-color** to a serie of data in a graph?

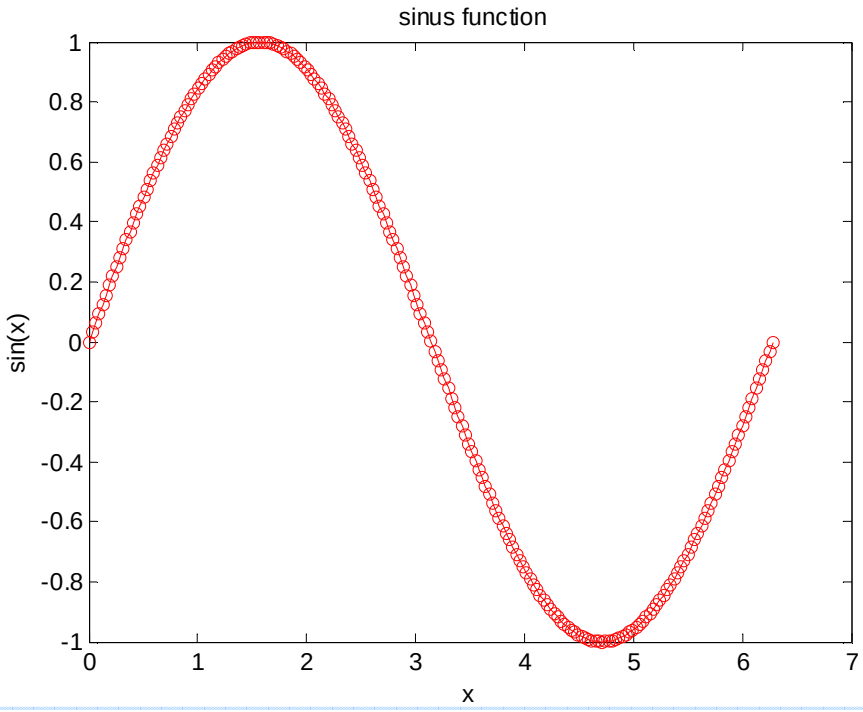
-Symbol: ‘+’, ‘o’, ‘*’, ‘x’

-Style: ‘-’, ‘—’, ‘:’, ‘-.’

-Color: ‘’c’, ‘m’, ‘y’, ‘g’, ‘b’, ‘w’, ‘e’, ‘k’

Plot(x,y1,’ symbol-style-color’)

```
>> plot(x,y1,'o-r'), xlabel('x'), ylabel('sin(x)'), title('sinus function')
>>
```



Plot the functions sin(x) and cos(x) between 0≤x≤2π

```
>>x=[0:pi/100:2*pi];
>> y1=sin(x);
>>y2=cos(x);
>> plot(x,y1, ‘o-r’,x,y2, ‘*:g’), xlabel('x'), ylabel('sin(x), cos(x)'),
title('sinus function')
>>
```