Introduction to MATLAB

Presented by

Preeti Kharwar

What is MATLAB?

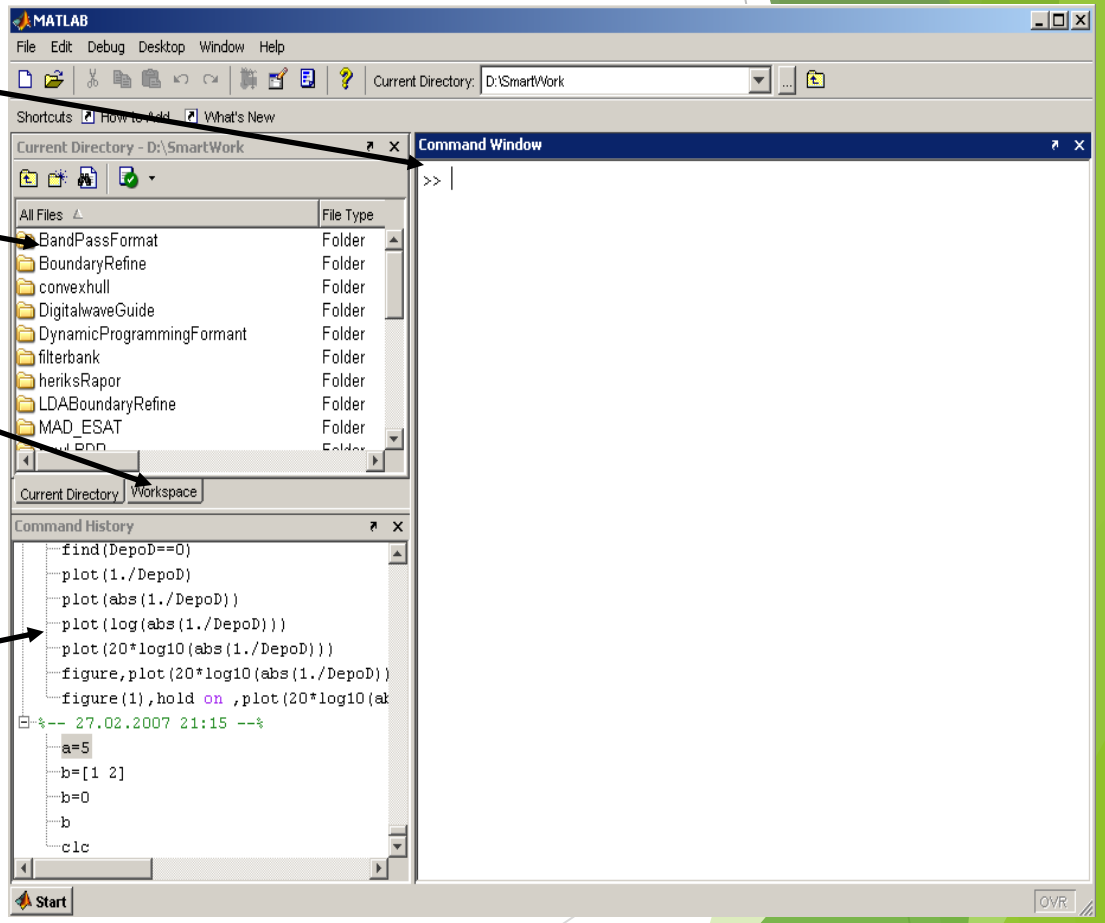
- Stands for **MAT**rix **LAB**oratory.
- High level language for technical computing with great visualization capabilities.
- Highly optimized for matrix operations.
- Highly interactive, interpreted programming language.

MATLAB in default view has four smaller windows:

- **Command Window-**
  - type commands
- **Current Directory Window**
  - View folders and m-files
- **Workspace Window**
  - View program variables
  - Double click on a variable

  to see it in the Array Editor
- **Command History Window**
  - view past commands
  - save a whole session

  using diary



Note: Besides the above windows there are: Editor Window, Help Window, Figure Window.

Variables

- No need for types. i.e.,

```
int a;
double b;
float c;
```

- All variables are created with double precision unless specified and they are vectors or matrices.

```
Example:
>>x=5;
>>x1=2;
```

Create Arrays and Matrices in MATLAB-

- Creation of One dimensional Array: **Syntax-** var_name=[elements].

  Example- x = 5.

- For row vector separate the elements with comma or space and for column vector use semicolon or press enter after each element.

  Syntax -var_name=[m:q:n] or m: q: n.

  Example- Row vector-y = [1 2 3] (or y = [1, 2, 3])

  Column vector >> z = [1; 2; 3]

- Creation of Matrices:

  Syntax-var_name=[first row elements; second row elements ;.....; last row]

  Example- w=[1 2 3 ;6 8 9; 0 1 0];

- Accessing element-

  For Vector:

  Syntax- var name[element indices]

  Example- y(1,2)=2;

  For Matrix:

  Syntax- var name[row, column]

  Example- w(2,3)=9;

- Two vectors (matrices) are appended horizontally and vertically by using [A, B] and [A; B], respectively.
- **Example-**a = [1 ,2], b = [4 ,5],

  >> A = [a, b]

  A=1 2 4 5

  >> B = [a ; b]

  B= 1 2

  4 5

- Special Commands: zeros(m,n), ones(m,n), eye(n).

Adding Elements to a Vector or a Matrix-

```
>> A=[1 2 3]
A=
    1   2   3
>> A(4:6)=[5:2:9]
A=
    1   2   3   5   7   9

>> B=[1 2]
B=
    1   2
>> B(5)=7;
B=
    1   2   0   0   7
```

```
>> C=[1 2; 3 4]
C=
    1   2
    3   4
>> C(3,:)=[5 6];
C=
    1   2
    3   4
    5   6

>> D=linspace(4,12,3);
>> E=[C D']
E=
    1   2   4
    3   4   8
    5   6   12
```

Arithmetic Operations & Variable Naming-

- Arithmetic operations: Addition(+), Subtraction(-), Multiplication(*), Right division (/) , Left division (\) and Exponentiation (^).
- Semicolon is used to suppress command output.
- clc command clears the command window.
- MATLAB operations on numeric arrays are matrix operations.
- Prepend "." for element-wise operations.
- Rules about Variable names-

  Must begin with a letter.

  Maximum 63 characters. No space allowed.

  Can have letters, digits, and underscore characters.

  Case sensitive.

Examples-

- **A=[1 2 ;3 4] B=[1 0; 1 0]**
- **A+B=[2 2; 4 4]**
- **A-B=[0 2; 2 4]**
- **A*B=[3 0;7 0]**
- **A.*B=[1 0; 3 0]**

Operators (relational, logical)

- == Equal to
- ~= Not equal to
- < Strictly smaller
- > Strictly greater
- <= Smaller than or equal to
- >= Greater than equal to
- & - And operator
- | - Or operator

Conditionals Statements-

To perform conditional operations we have if-end, if-else-end, if-elseif-else-end. Structures along with the switch-case statements.

- Syntax for if –else-end:

  if condition

  commands

  else

  commands

  end

- Syntax for Switch :

  switch switch expression

  case value1

  commands

  case value2

  commands

  otherwise

  commands

  end

Loops in MATLAB-

- **Syntax for for loop:**

```
for k= initial: increment: final
 Statement1
 Statement2
 end
```

Example-

```
 s = 2;
 H = 0;
for c = 1:s
   H(c) = 1/(c-1)
end
```

Ans- [2.0 0.6667]

- **Syntax for while loop:**

```
while conditional  expression
Statement1
Statement2
end
```

Example-

```
n = 50;
while n < 100
        n=n*1.05+50;
end
```

Ans- 102.5

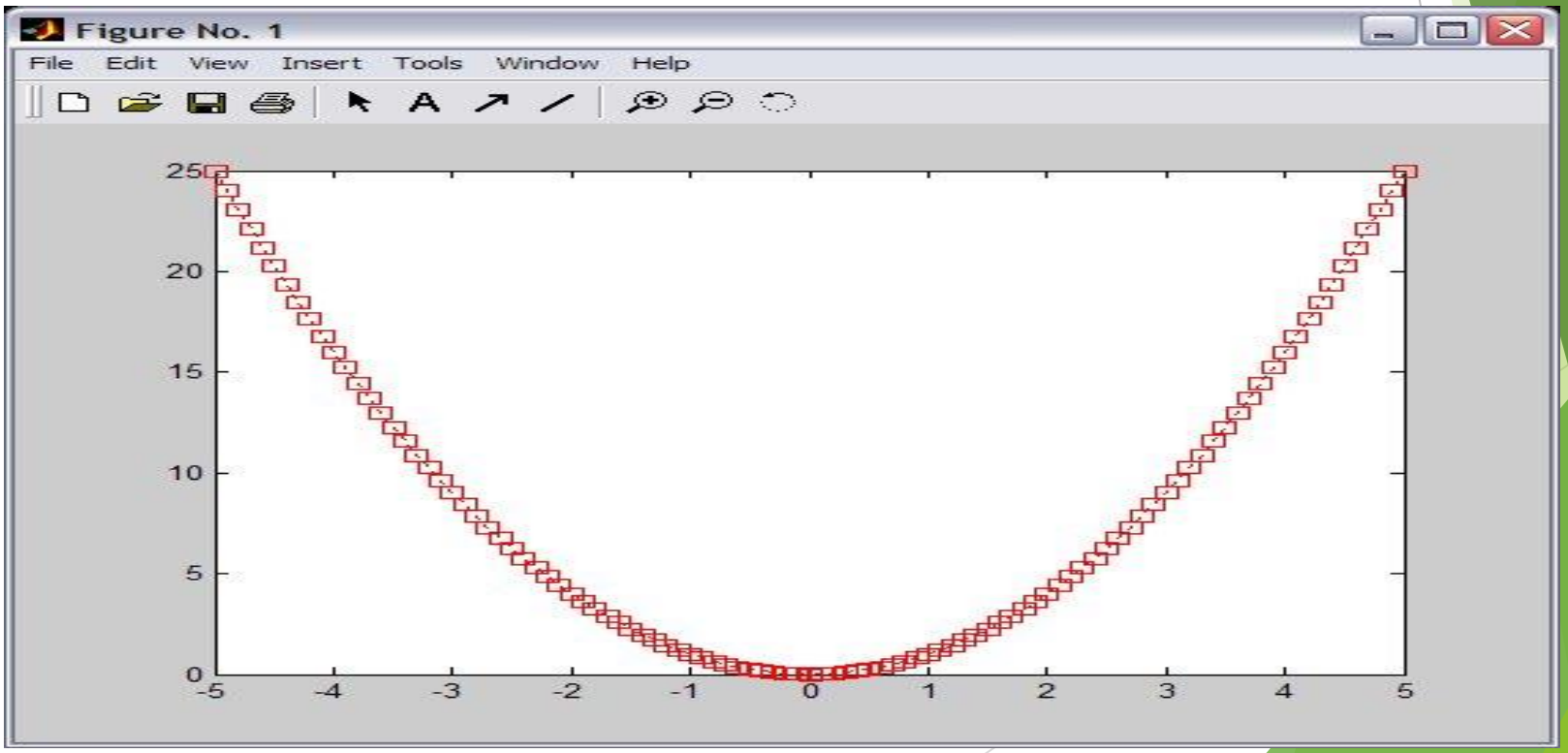
Some Built-in functions-

- mean(A):mean value of a vector
- max(A), min (A): maximum and minimum.
- sum(A): summation.
- sort(A): sorted vector
- median(A): median value
- std(A): standard deviation.
- det(A) : determinant of a square matrix
- dot(a,b): dot product of two vectors
- Cross(a,b): cross product of two vectors
- Inv(A): Inverse of a matrix A

Graphics - 2D Plots

Syntax-plot(xdata, ydata, ‘marker_style’);

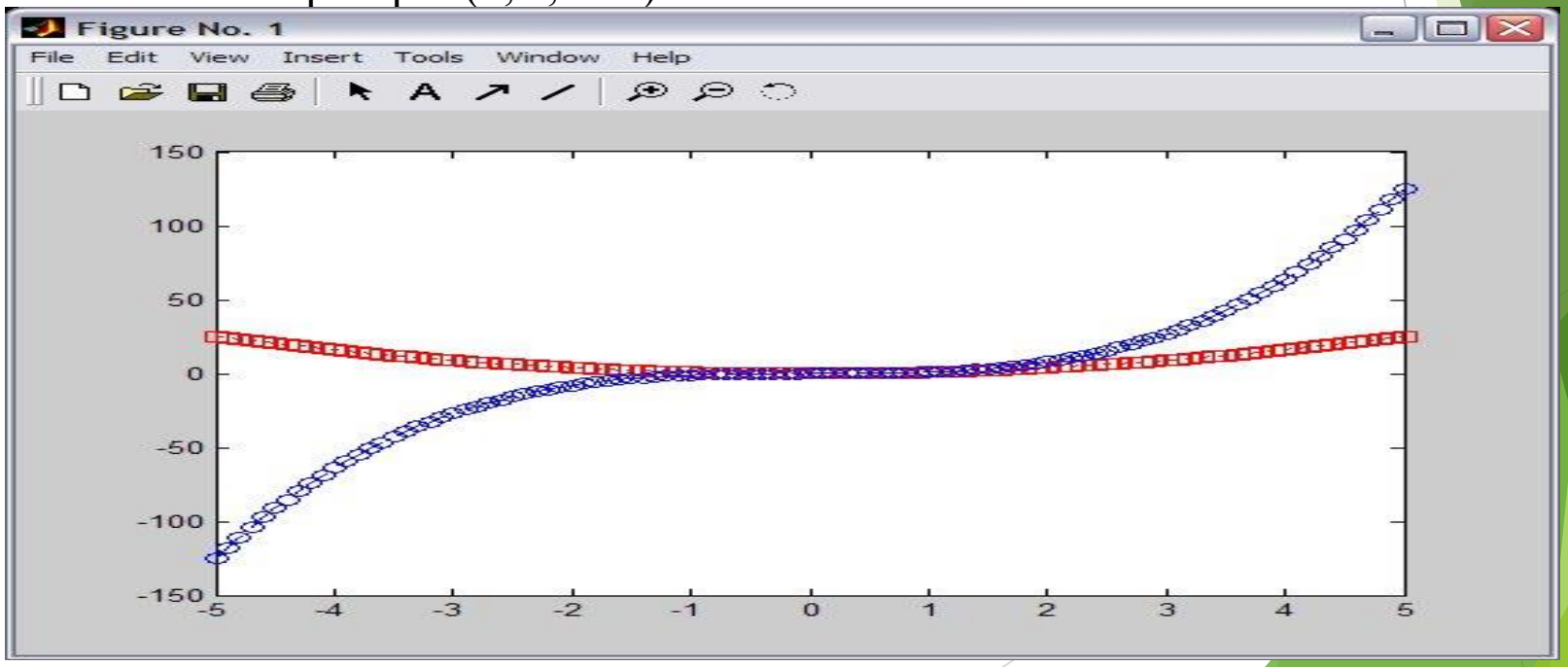
For example:
```
>> x=-5:0.1:5;
>> y=x.^2;
>> pl1=plot(x, y, ‘--rs');
```



Graphics - Overlay Plots

Use "**hold on** " for overlaying graphs.

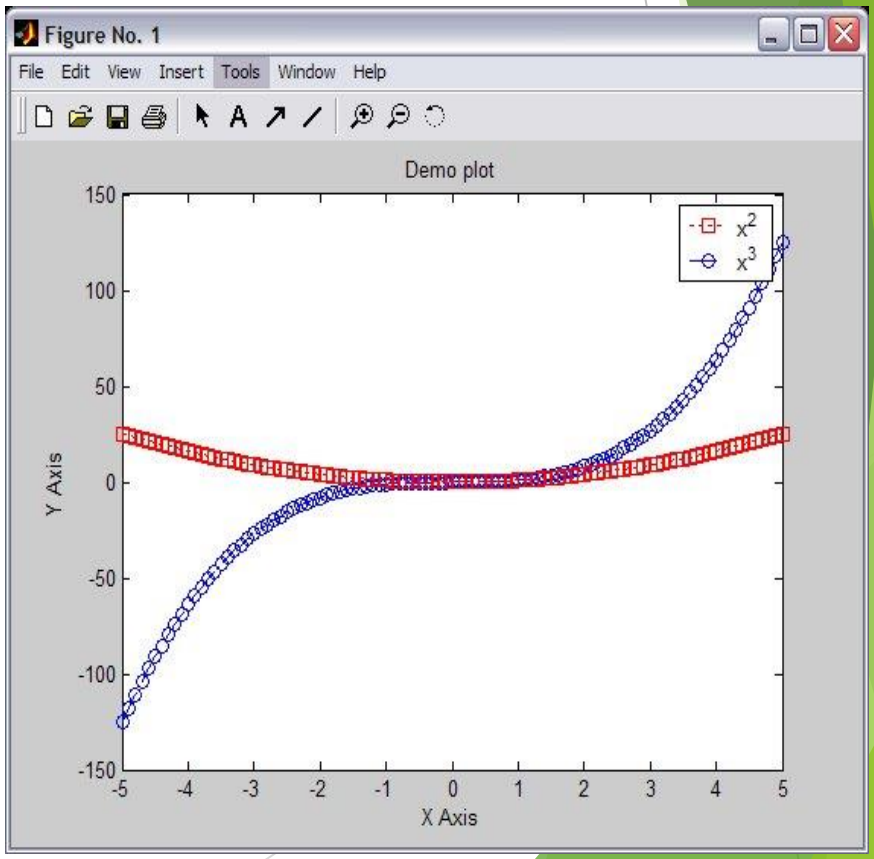
- Example:>> x=-5:0.1:5;
  >> y=x.^2;
  >> pl1=plot(x, y, '--rs');
  >> hold on;
  >> z=x.^3;
  >> pl2=plot(x, z,':bo')



Graphics – Annotation-

Use title, xlabel, ylabel and legend for annotation

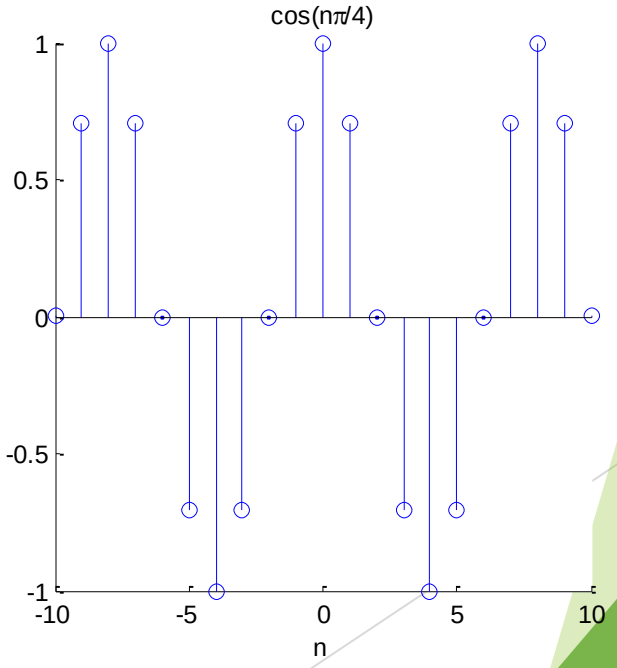
- >> title('Demo plot');
- >> xlabel('X Axis');
- >> ylabel('Y Axis');
- >> legend('x^2', 'x^3');



Graphics-Stem()

- stem()is to plot discrete sequence data

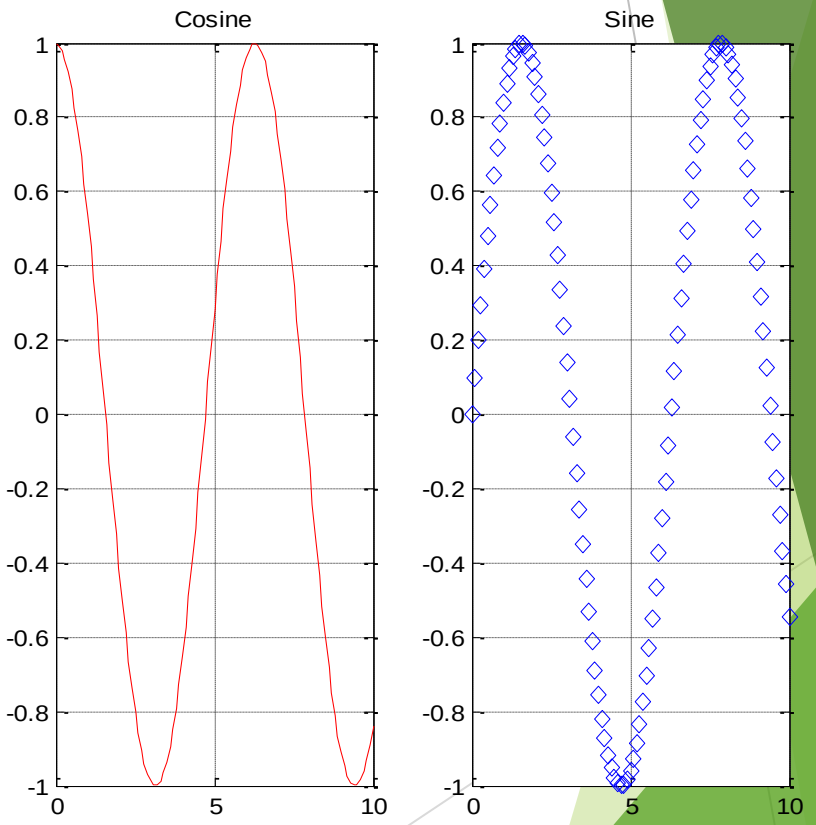
```
>> n=-10:10;
>> x=cos(pi/4*n);
>> f=stem(n,x)
>> title('cos(n\pi/4)')
>> xlabel('n')
```



subplots

- Use subplots to divide a plotting window into several panes.

```
>> x=0:0.1:10;
>> f=figure;
>> f1=subplot(1,2,1);
>> plot(x,cos(x),'r');
>> grid on;
>> title('Cosine')
>> f2=subplot(1,2,2);
>> plot(x,sin(x),'d');
>> grid on;
>> title('Sine');
```



Problems-

- Plot the following signals in linear scale

$$x(t) = \sin(3t) \qquad -5 < t < 5$$
$$y(t) = e^{2t+3} \qquad 0 < t < 5$$

- Plot the following signals, use log scale for y-axis

$$x(t) = e^{t+2}\,(2t+1) \quad 0 < t < 10$$

- Plot the real part and imaginary part of the following signal

$$x(t) = e^{0.5t + j(t+\pi/3)} \qquad 0 < t < 10$$

- For the signal in previous question, plot its phase and magnitude .