Introduction to MATLAB

Lecture Series by CEPSTRUM

Presented by
Pratik Kotkar & Akash Baid

Topics..

- What is MATLAB ??
- Basic Matrix Operations
- Script Files and M-files
- Some more Operations and Functions

APPLICATIONS:

- Plotting functions ..
- Image Processing Basics ..
- Robotics Applications ..
- GUI Design and Programming

Topics..

- What is MATLAB ??
- Basic Matrix Operations
- Script Files and M-files
- Some more Operations and Functions

APPLICATIONS:

- Plotting functions ..
- Image Processing Basics ..
- Robotics Applications ..
- GUI Design and Programming

MATLAB

- MATLAB is a program for doing numerical computation. It was originally designed for solving linear algebra type problems using matrices. It's name is derived from MATrix LABoratory.

MATLAB has since been expanded and now has built-in functions for solving problems requiring data analysis, signal processing, optimization, and several other types of scientific computations. It also contains functions for 2-D and 3-D graphics and animation.

MATLAB

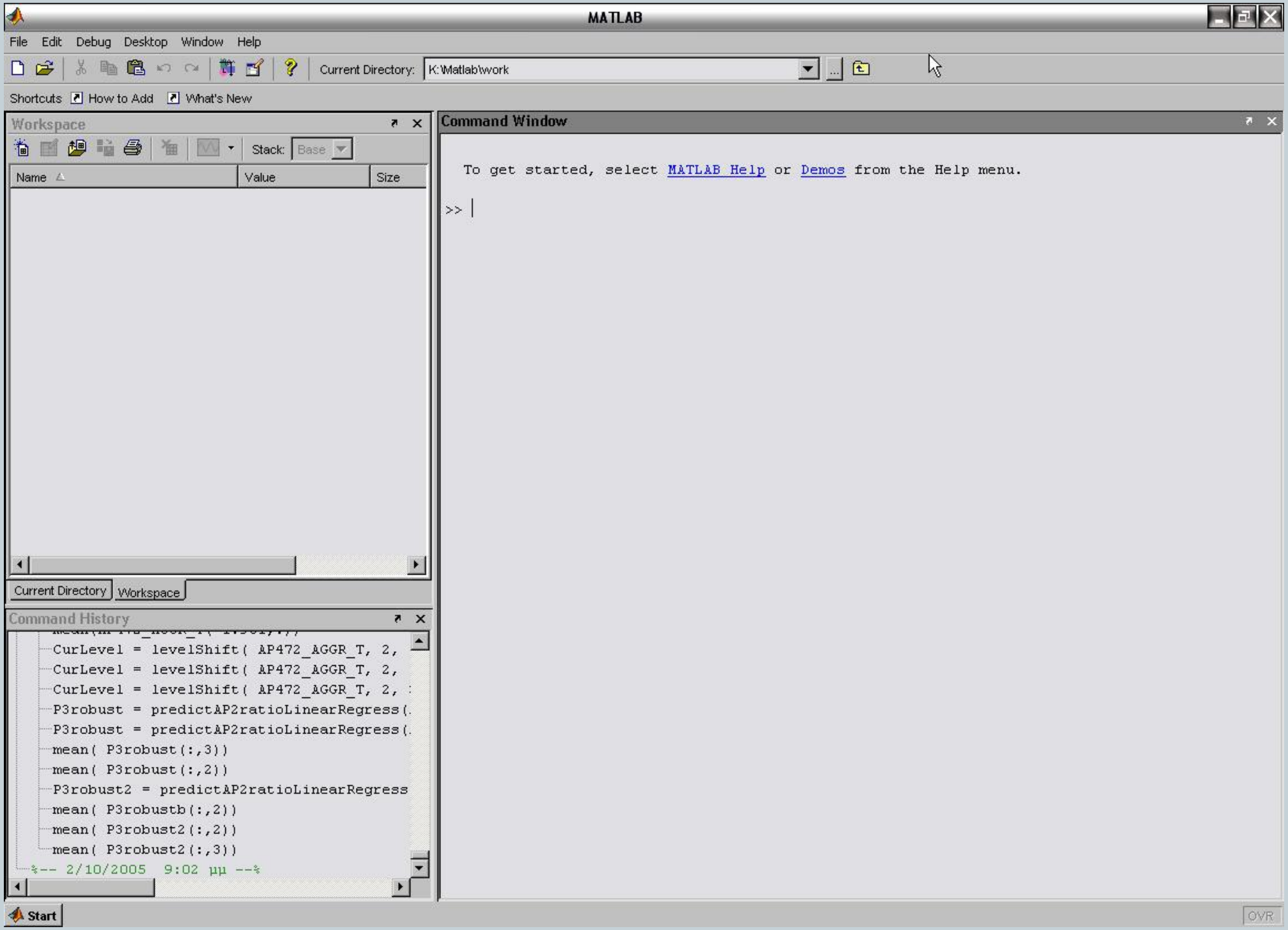
Everything in MATLAB is a matrix !

MATLAB

- The MATLAB environment is command oriented somewhat like UNIX. A prompt appears on the screen and a MATLAB statement can be entered. When the <ENTER> key is pressed, the statement is executed, and another prompt appears.

- If a statement is terminated with a semicolon (;), no results will be displayed. Otherwise results will appear before the next prompt.

The MATLAB User Interface



MATLAB

To get started, type one of these commands: helpwin, helpdesk, or demo

```
» a=5;
» b=a/2

b =

   2.5000

»
```

MATLAB Variable Names

- Variable names ARE case sensitive
- Variable names can contain up to 63 characters (as of MATLAB 6.5 and newer)
- Variable names must start with a letter followed by letters, digits, and underscores.

MATLAB Special Variables

ans	Default variable name for results
pi	Value of π
eps	Smallest incremental number
inf	Infinity
NaN	Not a number e.g. 0/0
i and j	i = j = square root of -1
realmin	The smallest usable positive real number
realmax	The largest usable positive real number

Topics..

- What is MATLAB ??
- Basic Matrix Operations
- Script Files and M-files
- Some more Operations and Functions

APPLICATIONS:

- Plotting functions ..
- Image Processing Basics ..
- Robotics Applications ..
- GUI Design and Programming

Math & Assignment Operators

Power	^	or	.^	a^b	or	a.^b
Multiplication	*	or	.*	a*b	or	a.*b
Division	/	or	./	a/b	or	a./b
or	\	or	.\	b\a	or	b.\a
NOTE:	56/8 = 8\56					

- (unary) + (unary)

Addition	+	a + b	
Subtraction	-	a - b	
Assignment	=	a = b	(assign b to a)

Other MATLAB symbols

>>	prompt
. . .	continue statement on next line
,	separate statements and data
%	start comment which ends at end of line
;	(1) suppress output
	(2) used as a row separator in a matrix
:	specify range

MATLAB Relational Operators

- MATLAB supports six relational operators.

Less Than	<
Less Than or Equal	<=
Greater Than	>
Greater Than or Equal	>=
Equal To	==
Not Equal To	~=

MATLAB Logical Operators

- MATLAB supports three logical operators.

not	~	% highest precedence
and	&	% equal precedence with or
or	|	% equal precedence with and

MATLAB Matrices

- MATLAB treats all variables as matrices. For our purposes a matrix can be thought of as an array, in fact, that is how it is stored.

- Vectors are special forms of matrices and contain only one row OR one column.

- Scalars are matrices with only one row AND one column

MATLAB Matrices

- A matrix with only one row AND one column is a scalar. A scalar can be created in MATLAB as follows:

```
» a_value=23

a_value =

   23
```

MATLAB Matrices

- A matrix with only one row is called a row vector. A row vector can be created in MATLAB as follows (note the commas):

```
» rowvec = [12 , 14 , 63]

rowvec =

   12   14   63
```

MATLAB Matrices

- A matrix with only one column is called a column vector. A column vector can be created in MATLAB as follows (note the semicolons):

```
» colvec = [13 ; 45 ; -2]

colvec =

    13
    45
    -2
```

MATLAB Matrices

- A matrix can be created in MATLAB as follows (note the commas AND semicolons):

» matrix = [1 , 2 , 3 ; 4 , 5 ,6 ; 7 , 8 , 9]

matrix =

1	2	3
4	5	6
7	8	9

Extracting a Sub-Matrix

- A portion of a matrix can be extracted and stored in a smaller matrix by specifying the names of both matrices and the rows and columns to extract. The syntax is:

```
sub_matrix = matrix ( r1 : r2 , c1 : c2 ) ;
```

where r1 and r2 specify the beginning and ending rows and c1 and c2 specify the beginning and ending columns to be extracted to make the new matrix.

MATLAB Matrices

- A column vector can be extracted from a matrix. As an example we create a matrix below:

» matrix=[1,2,3;4,5,6;7,8,9]

matrix =

1	2	3
4	5	6
7	8	9

- Here we extract column 2 of the matrix and make a column vector:

» col_two=matrix(: , 2)

col_two =

2
5
8

MATLAB Matrices

- A row vector can be extracted from a matrix. As an example we create a matrix below:

» matrix=[1,2,3;4,5,6;7,8,9]

matrix =

1	2	3
4	5	6
7	8	9

- Here we extract row 2 of the matrix and make a row vector. Note that the 2:2 specifies the second row and the 1:3 specifies which columns of the row.

» rowvec=matrix(2 : 2 , 1 : 3)

rowvec =

4	5	6

Topics..

- What is MATLAB ??
- Basic Matrix Operations
- Script Files and M-files
- Some more Operations and Functions

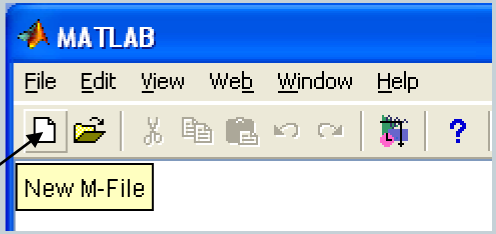
APPLICATIONS:

- Plotting functions ..
- Image Processing Basics ..
- Robotics Applications ..
- GUI Design and Programming

Use of M-File

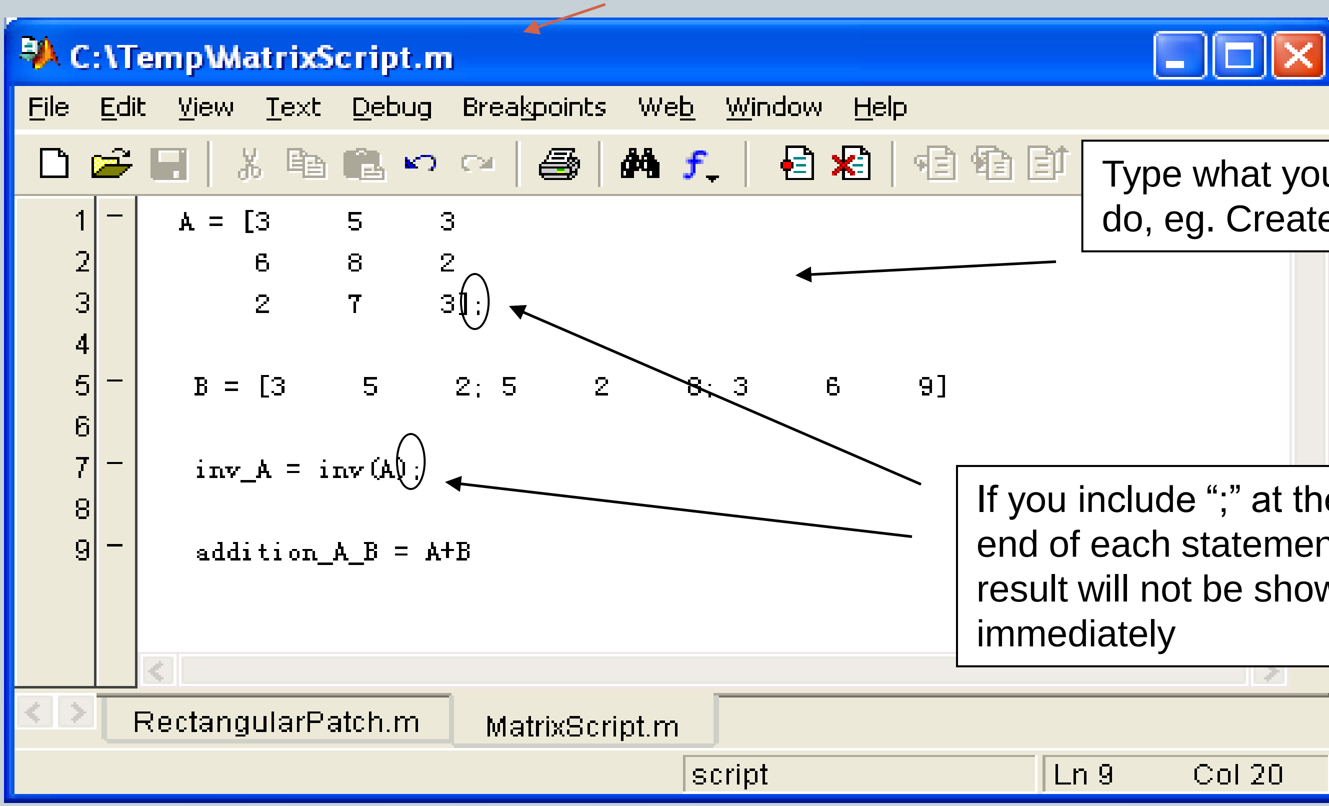
- There are two kinds of M-files:
  - **Scripts**, which do not accept input arguments or return output arguments. They operate on data in the workspace.
  - **Functions**, which can accept input arguments and return output arguments. Internal variables are local to the function.

Click to create a new M-File



M-File as script file

Save file as *filename*.m



Type what you want to do, eg. Create matrices

If you include ";" at the end of each statement, result will not be shown immediately

Run the file by typing the filename in the command window

Reading Data from files

- MATLAB supports reading an entire file and creating a matrix of the data with one statement.

>> load mydata.dat;   % *load*s file into matrix.

% The matrix may be a scalar, a vector, or a

% matrix with multiple rows and columns. The

% matrix will be named mydata.

>> size (mydata)   % *size* will return the number

% of rows and number of

% columns in the matrix

>> length (myvector)   % *length* will return the total

% no. of elements in

myvector

Topics..

- What is MATLAB ??
- Basic Matrix Operations
- Script Files and M-files
- Some more Operations and Functions

APPLICATIONS:

- Plotting functions ..
- Image Processing Basics ..
- Robotics Applications ..
- GUI Design and Programming

Matlab Selection Structures

- An if - elseif - else structure in MATLAB.

Note that elseif is <u>one</u> word.

```
if      expression1              % is true
        % execute these commands
 elseif expression2              % is true
        % execute these commands
 else                        % the default
        % execute these commands
 end
```

MATLAB Repetition Structures

A for loop in MATLAB

for x = array

for **ind = 1:100**

b(ind)=sin(ind/10)

end

while loop in MATLAB

while expression

while **x <= 10**

% execute these commands

end

x=0.1:0.1:10; b=sin(x); - Most of the loops can be avoided!!!

Scalar - Matrix Addition

» a=3;

» b=[1, 2, 3;4, 5, 6]

b =

1	2	3
4	5	6

» c= b+a % Add a to each element of b

c =

4	5	6
7	8	9

Scalar - Matrix Subtraction


```
» a=3;
» b=[1, 2, 3;4, 5, 6]
b =
   1   2   3
   4   5   6
» c = b - a %Subtract a from each element of b
c =
  -2  -1   0
   1   2   3
```

Scalar - Matrix Multiplication


```
» a=3;
» b=[1, 2, 3; 4, 5, 6]
b =
    1    2    3
    4    5    6
» c = a * b  % Multiply each element of b by a
c =
    3    6    9
   12   15   18
```

Scalar - Matrix Division


```
» a=3;
» b=[1, 2, 3; 4, 5, 6]
b =
    1    2    3
    4    5    6
» c = b / a      % Divide each element of b by a
c =
   0.3333   0.6667   1.0000
   1.3333   1.6667   2.0000
```

The use of "." – "Element" Operation

Given A:

```
A =

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

Divide each element of A by 2

```
>> A./2

ans =

    1.5000    2.5000    1.5000
    3.0000    4.0000    1.0000
    1.0000    3.5000    1.5000
```

Multiply each element of A by 3

```
>> A.*3

ans =

     9    15     9
    18    24     6
     6    21     9
```

Square each element of A

```
>> A.^2

ans =

     9    25     9
    36    64     4
     4    49     9
```

Topics..

- What is MATLAB ??
- Basic Matrix Operations
- Script Files and M-files
- Some more Operations and Functions

APPLICATIONS:

- Plotting functions ..
- Image Processing Basics ..
- Robotics Applications ..
- GUI Design and Programming

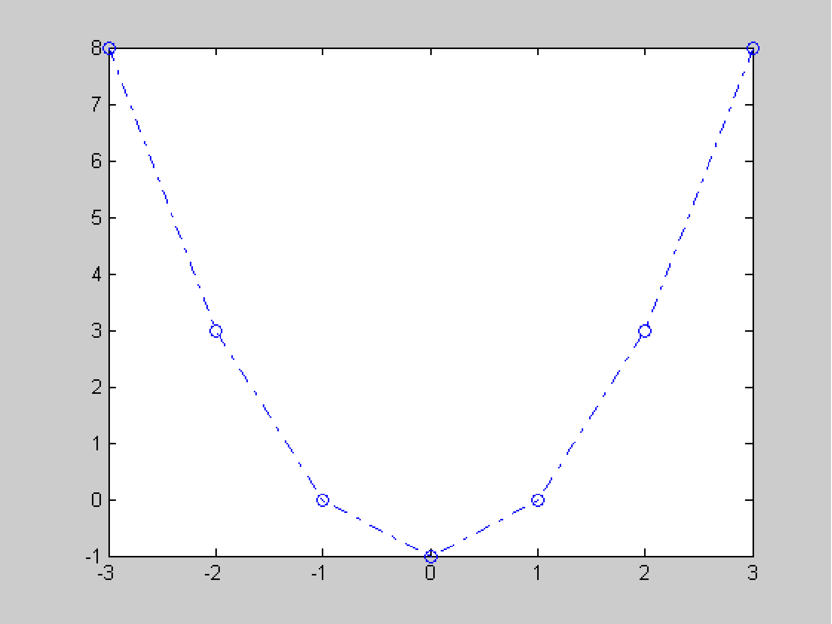
Plot

PLOT Linear plot.

- PLOT(X,Y) plots vector Y versus vector X
- PLOT(Y) plots the columns of Y versus their index
- PLOT(X,Y,S) with plot symbols and colors
- See also SEMILOGX, SEMILOGY, **TITLE**, **XLABEL**, **YLABEL**, AXIS, AXES, **HOLD**, COLORDEF, LEGEND, **SUBPLOT**...

Example

```
x = [-3 -2 -1 0 1 2 3];
y1 = (x.^2) -1;
plot(x, y1,'bo-.');
```



Plot Properties

XLABEL X-axis label.

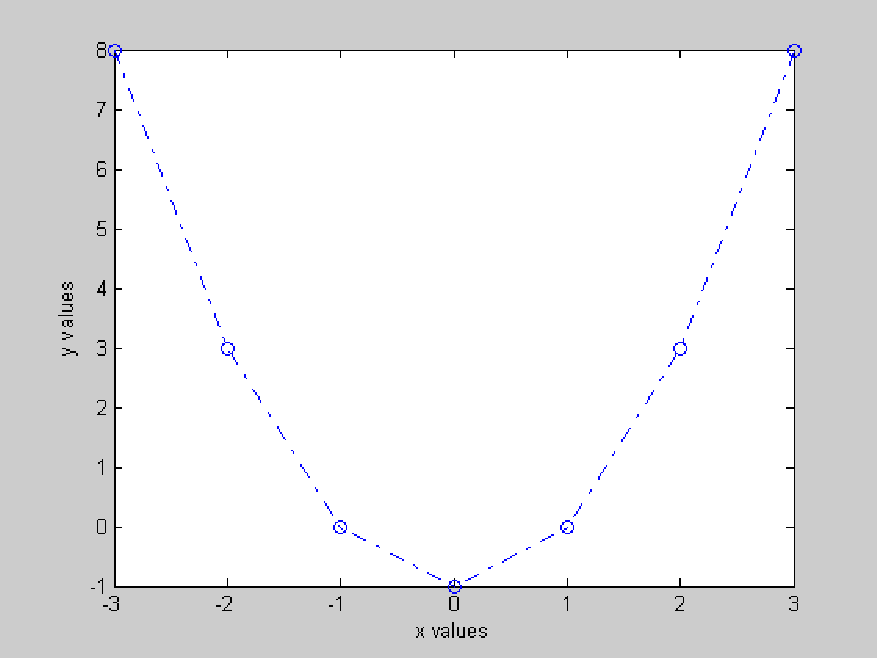
- XLABEL('text') adds text beside the X-axis on the current axis.

YLABEL Y-axis label.

- YLABEL('text') adds text beside the Y-axis on the current axis.

Example

```
...
xlabel('x values');
ylabel('y values');
```



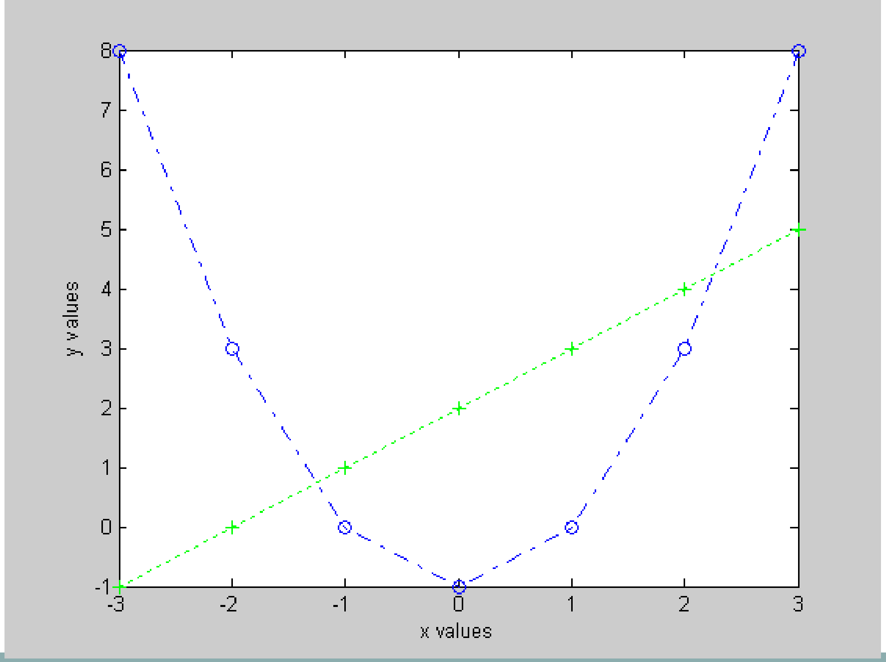
Hold

HOLD Hold current graph.

- HOLD ON holds the current plot and all axis properties so that subsequent graphing commands add to the existing graph.
- HOLD OFF returns to the default mode
- HOLD, by itself, toggles the hold state.

Example

```
...
hold on;
y2 = x + 2;
plot(x, y2, 'g+:');
```



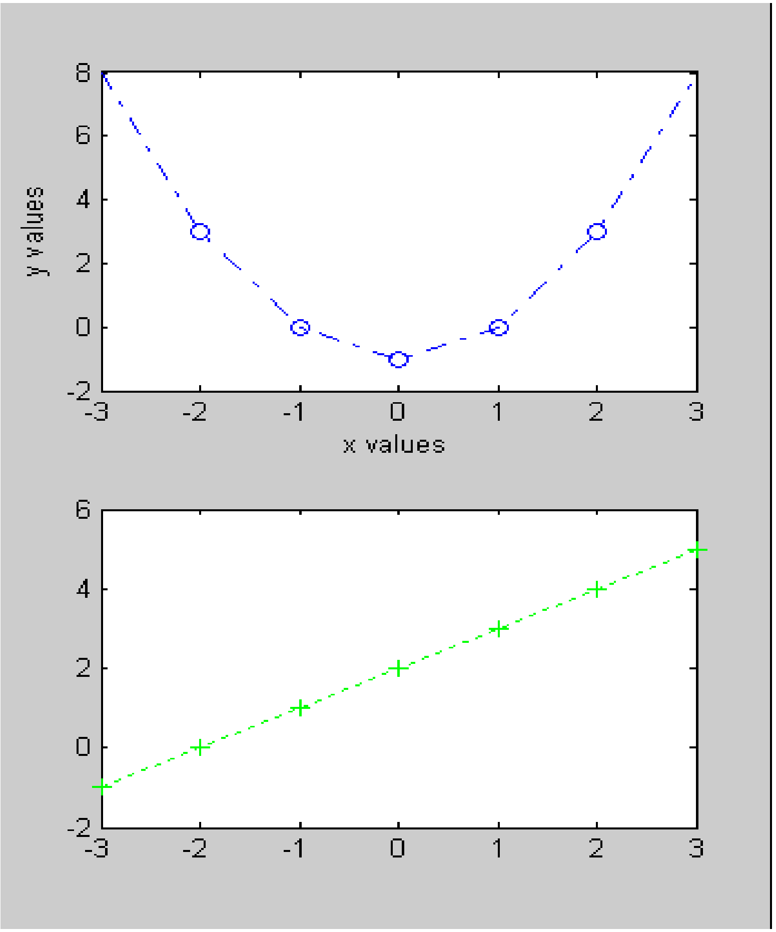
Subplot

SUBPLOT Create axes in tiled positions.

- SUBPLOT(m,n,p), or SUBPLOT(mnp), breaks the Figure window into an m-by-n matrix of small axes

Example

```
x = [-3 -2 -1 0 1 2 3];
y1 = (x.^2) -1;
% Plot y1 on the top
subplot(2,1,1);
plot(x, y1,'bo-.');
xlabel('x values');
ylabel('y values');
% Plot y2 on the bottom
subplot(2,1,2);
y2 = x + 2;
plot(x, y2, 'g+:');
```



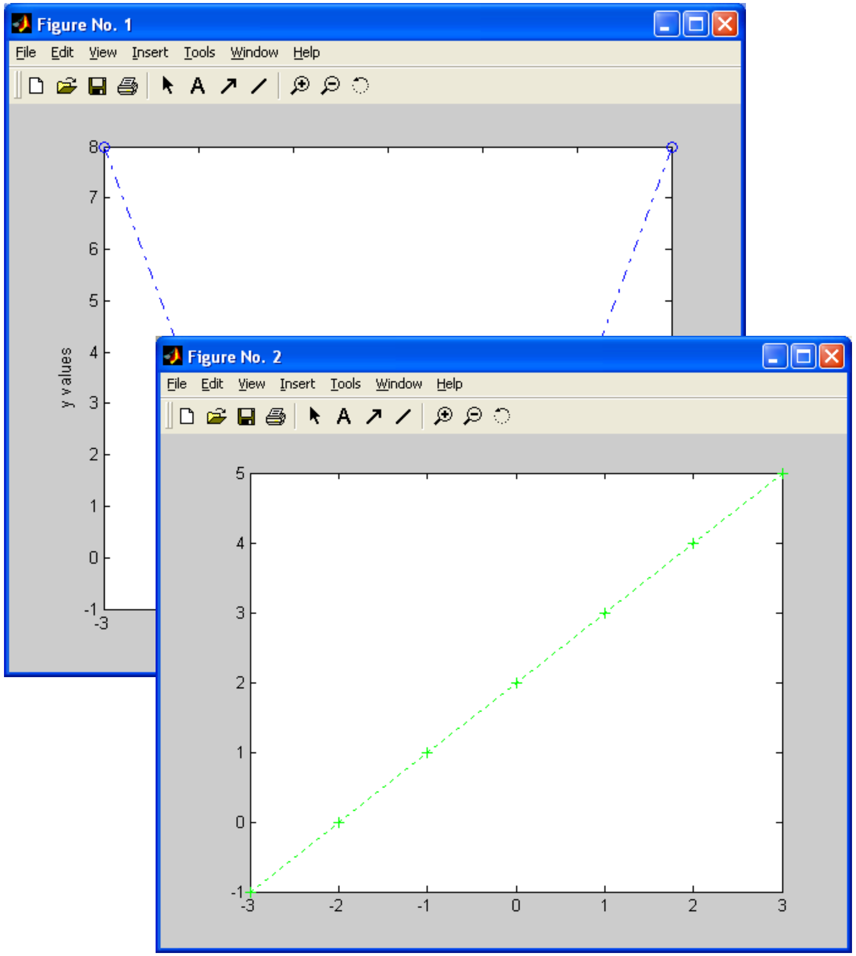
Figure

FIGURE Create figure window.

- FIGURE, by itself, creates a new figure window, and returns its handle.

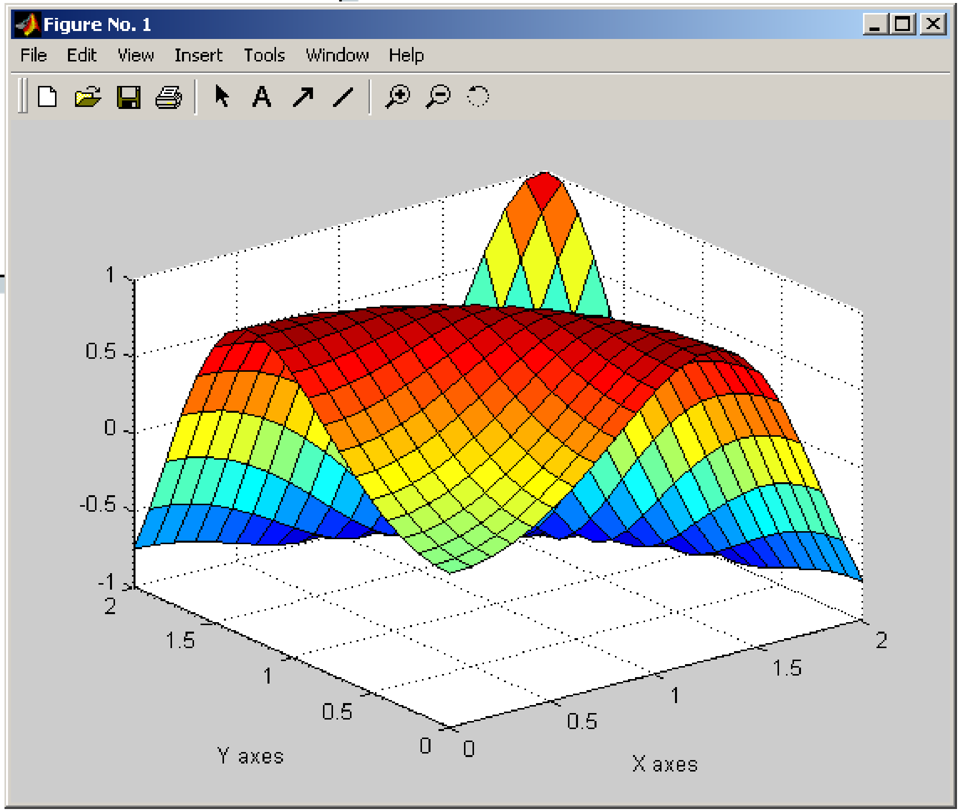
Example

```
x = [-3 -2 -1 0 1 2 3];
y1 = (x.^2) -1;
% Plot y1 in the 1st Figure
plot(x, y1,'bo-.');
xlabel('x values');
ylabel('y values');
% Plot y2 in the 2nd Figure
figure
y2 = x + 2;
plot(x, y2, 'g+:');
```



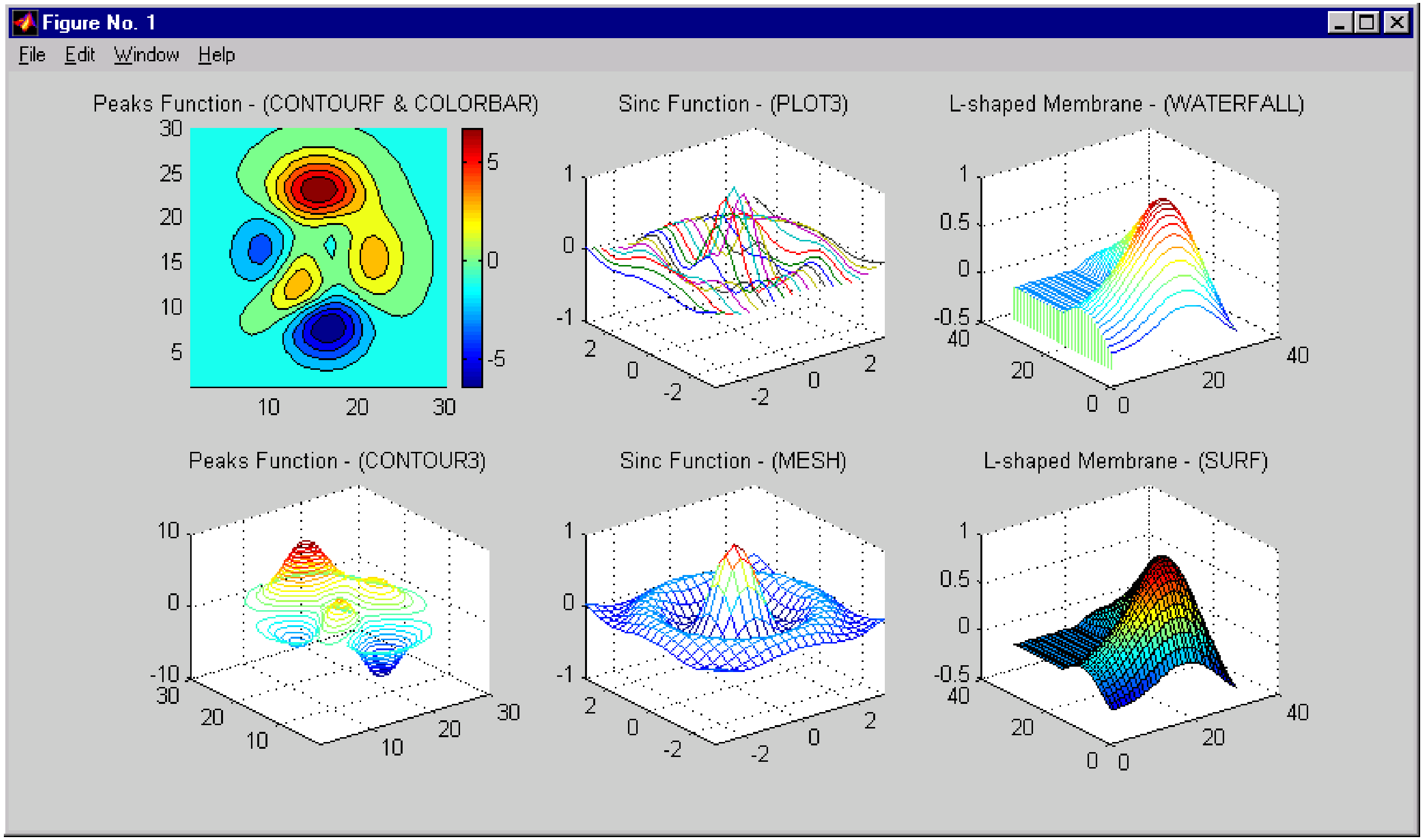
Surface Plot

```
x = 0:0.1:2;
y = 0:0.1:2;
[xx, yy] = meshgrid(x,y);
zz=sin(xx.^2+yy.^2);
surf(xx,yy,zz)
xlabel('X axes')
ylabel('Y axes')
```



3 D Surface Plot

contourf-colorbar-plot3-waterfall-contour3-mesh-surf



Topics..

- What is MATLAB ??
- Basic Matrix Operations
- Script Files and M-files
- Some more Operations and Functions

APPLICATIONS:

- Plotting functions ..
- Image Processing Basics ..
- Robotics Applications ..
- GUI Design and Programming

Image Processing Toolbox

- The Image Processing Toolbox is a collection of functions that extend the capability of the MATLAB ® numeric computing environment. The toolbox supports a wide range of image processing operations, including:
  - Geometric operations
  - Neighborhood and block operations
  - Linear filtering and filter design
  - Transforms
  - Image analysis and enhancement
  - Binary image operations
  - Region of interest operations

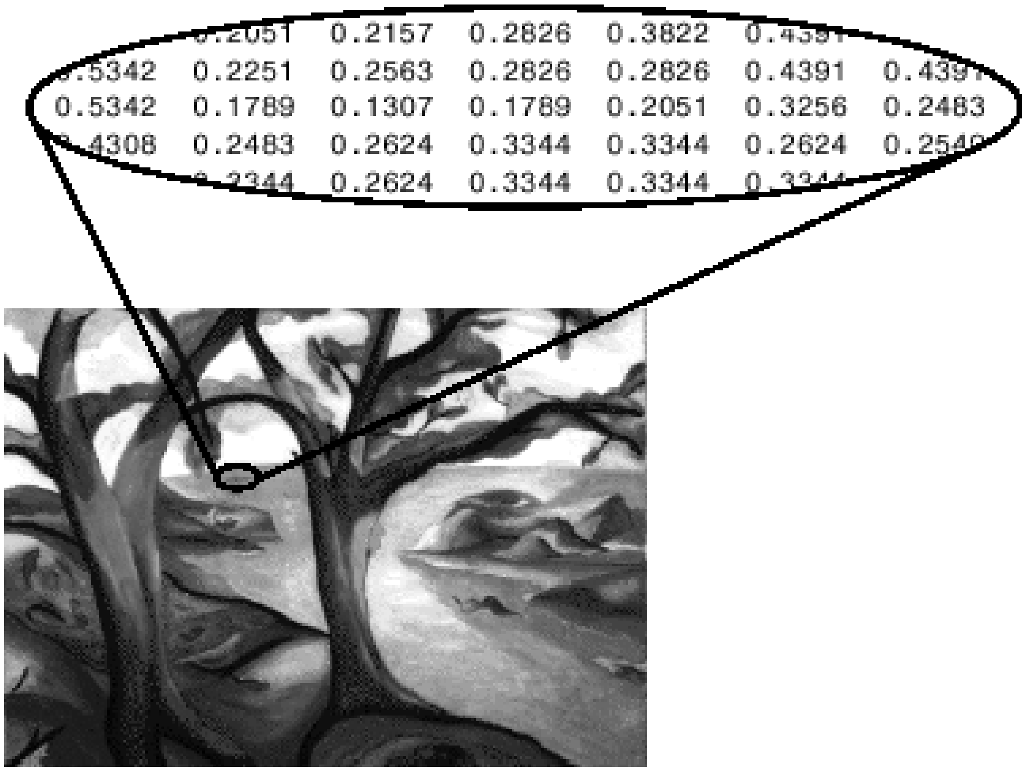
MATLAB Image Types

- Intensity images : [0,1] or uint8
- Binary images : {0,1}
- RGB images : m-by-n-by-3

Intensity Images

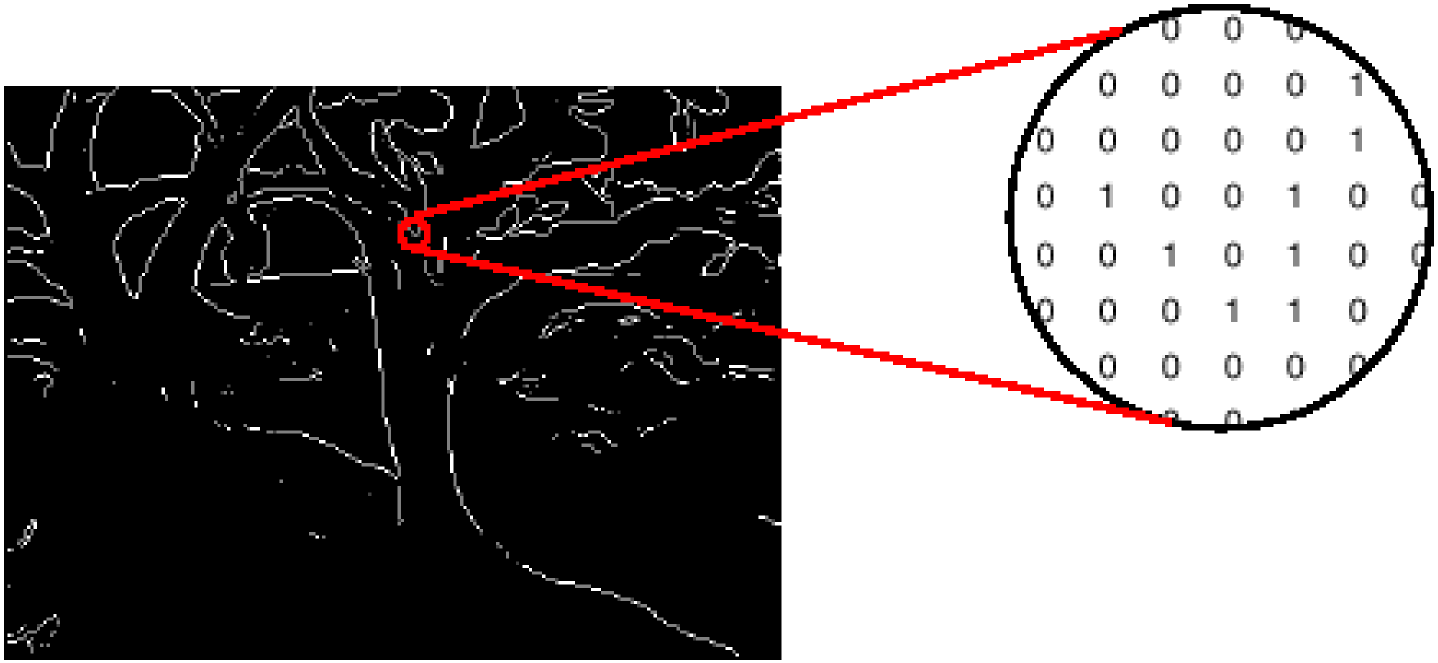
» **image = ind2gray(x,map);**

» **imshow(image);**



Binary Images

» **imshow(edge(image));**



RGB Images

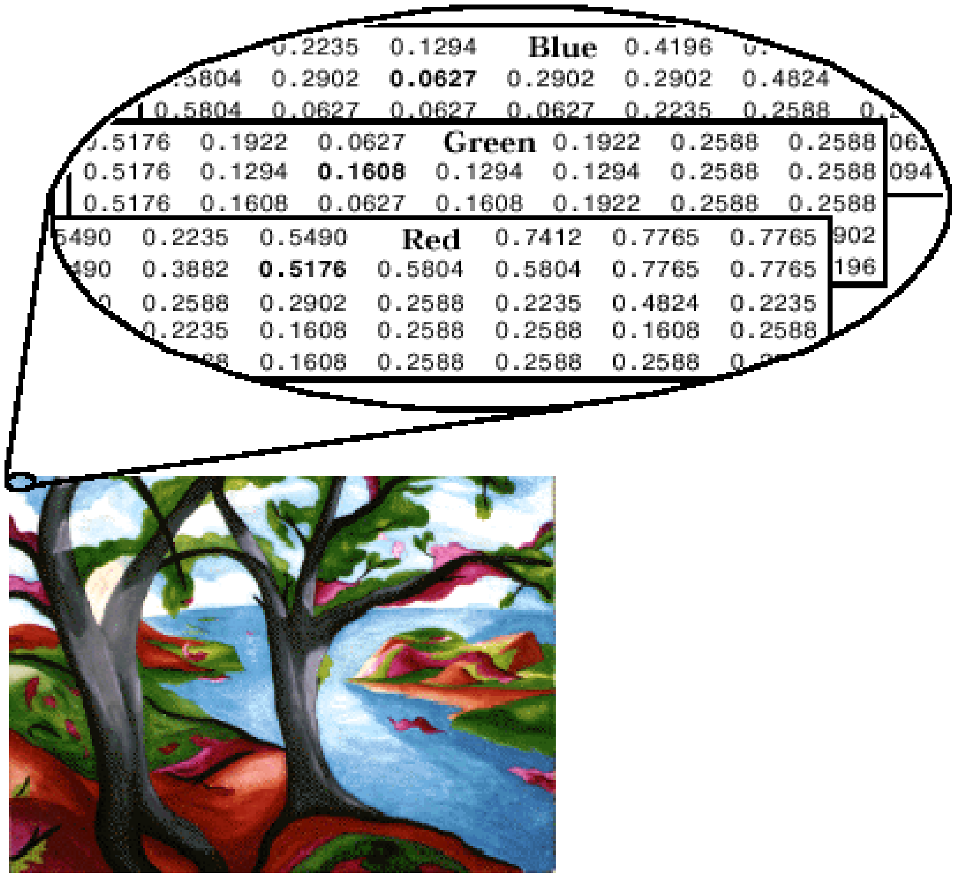

0.1294 Blue 0.4196
0.2902 0.0627 0.2902 0.2902 0.4824
0.5804 0.0627 0.0627 0.0627 0.2235 0.2588
0.1922 0.0627 Green 0.1922 0.2588 0.2588
0.5176 0.1294 0.1608 0.1294 0.1294 0.2588 0.2588
0.5176 0.1608 0.0627 0.1608 0.1922 0.2588 0.2588
0.2235 0.5490 Red 0.7412 0.7765 0.7765
0.3882 0.5176 0.5804 0.5804 0.7765 0.7765
0.2588 0.2902 0.2588 0.2235 0.4824 0.2235
0.1608 0.2588 0.2588 0.1608 0.2588
0.1608 0.2588 0.2588 0.2588

Image Display

- **image** - create and display image object
- **imagesc** - scale and display as image
- **imshow** - display image
- **colorbar** - display colorbar
- **getimage**- get image data from axes
- **truesize** - adjust display size of image
- **zoom** - zoom in and zoom out of 2D plot

Image Conversion

- **Gray2ind** - intensity image to index image
- **im2bw** - image to binary
- **Im2double** - image to double precision
- **Im2uint8** - image to 8-bit unsigned integers
- **Im2uint16** - image to 16-bit unsigned integers
- **Ind2gray** - indexed image to intensity image
- **mat2gray** - matrix to intensity image
- **rgb2gray** - RGB image to grayscale
- **rgb2ind** - RGB image to indexed image

IMAGE ENHANCEMENT

- Adjust intensity
  - imadjust
  - histeq
- Noise removal
  - linear filtering
  - median filtering
  - adaptive filtering

```
>>im2 = histeq(im);
>>imshow(im2)
```