

## Introduction of Matlab:

MATLAB stands for 'matrix laboratory'. It is a software package for high performance mathematical computation, visualization, and programming environment. It provides an interactive environment with hundreds of built-in functions for technical computing, graphics and animations. MATLAB is a multi-paradigm. So, it can work with multiple types of programming approaches, such as functional, object-oriented and visual.

## Features of Matlab:

- Matlab is a high level language: Matlab supports ooprogramming. It also supports different types of programming constructs like control flow statements. It also supports input/output statements like `disp()` & `inbuilt()`.
- Interactive graphics: Matlab has inbuilt graphics to enhance user experience. We also supports processing of image and displaying them in 2D or 3D formats.

A large library of mathematical functions: Matlab has a huge inbuilt library of functions required for mathematical analysis of any data. It has common math function like `sqrt`, `factorial` etc.

- Data access and processing : Matlab allows accessing of data from external source like image files, audio and real-time data from JDBC/ ODBC.
- Interactive environment : Matlab offers interactive environment by providing a GUI (Graphical user interface) and different types of tools like signal analyses and tuners.
- Matlab can interface with different languages.
- Matlab and Simulink.
- Used in machine learning, deep learning and computer vision.

### Environment of Matlab :

1. Command window : use to execute the command.
2. Workspace : display all defined variable i.e. who, whos. current variables in workspace.  
Save - save workspace variables to \*.mat file  
load - load variables from \*.mat file.
3. Command history : To display the record command used.
4. File editor : define your function or file work.

### Matlab Matrices:

A matrix is a two-dimensional array of elements.

AJAYA

Teacher's Signature \_\_\_\_\_

the matrix is created by assigning array elements that are delimited by spaces or commas & using semicolon to mark the end of each row.

Syntax :  $a = [\text{Elements}, \text{elements}]$

MATLAB array index start from 1.

### Matlab conditional statements :

conditional statements enable you to select at runtime which block of code to execute. The simplest conditional statement is an if statement.

- if, elseif,  
else  
  
Execute statements if condition is true.
- switch, case, otherwise  
  
Execute one of several groups of statements.
- for  
  
for loops to repeat specified No. of times.
- while  
  
while loop to repeat when condition is true.
- try, catch  
  
Execute statements & catch resulting errors.

- **break** Terminates execution of for or while loop.
  - **return** Returns control to invoking script or function.
  - **continue** Pass control to next iteration of for or while loop.
  - **end** Terminates block of code or indicate last array index.
  - **pause** Stop MATLAB execution temp.
  - **parfor** Parallel for loop.
- \* Loop in MATLAB:

1. **while Loop :** It is used to execute a block of statements repeatedly until a given condition is satisfied.

Syntax - while expression  
statements  
end.

2. **For loop:** used for sequential traversal

~~Syntax : for initial value : step values final values  
statements  
end.~~

⇒ Visualization & graphics in Matlab :

Matlab has extensive facilities for displaying vector & matrices as graphs, as well as annotating & printing these graph. It includes high level functions for two-dimensional & 3D data visualization, image processing, animation & presentation graphics.

⇒ Plot(x,y) ⇒ The plot(x,y) function in MATLAB is used to create a 2D plot of data points. It takes two vectors as i/p arguments 'x' & 'y' which represent the x-coordinates & y-coordinates.

Syntax - Plot(x,y) :

x, y are vectors of same length.

MATLAB creates a line plot of the point with x values on x-axis and y values on y-axis.

Eg. To use the plot(x,y) function

$x = [1, 2, 3, 4, 5];$   
 $y = [2, 4, 6, 8, 10];$   
 $\text{plot}(x, y).$

Plot ( $n \sin(n)$ ) :

plot (n, sin(n)) command in MATLAB will create a 2D plot of the sine function for values of n ranging from minimum value of n to the maximum value of n.

Eg.  $n = (\text{inspace}(0, 2\pi));$   
 $\text{plot}(n, \sin(n))$

AJAYA

1. add axis labels

`xlabel('x')`

`ylabel('sin(x)')`

`title('Plot of the sine function')`

hold on hold off refreshing :

The eg will show that how to show multiple graphs in the same plot by using `hold on` & `hold off` command in MATLAB.

$$f(n) = n;$$

$$g(n) = 2n;$$

$$0 \leq n \leq 10$$

$$n = 0:1:10$$

$$f = n;$$

$$g = 2 * n;$$

`plot(n, f)`

`hold on`

`plot(n, g)`

`hold off`

figure, figure(k) open a new file :

These commands are used to create new figure window or to select an existing fig window to work with.

The figure command without any input argument

AJAYA

Expt. No. \_\_\_\_\_

creates a new, empty figure window.

eg. \* Create a new, empty figure window.  
figure.

If want to create a new figure window with a specific No. then use figure(k) command where 'k' is an integer.

eg. figure(2).

axis ([x<sub>min</sub>, x<sub>max</sub> y<sub>min</sub>, y<sub>max</sub>]) :

axis function is used to set the limits of the axes of a plot. specially the 'axis' function takes a vector of four values as its input argument representing the minimum and maximum values for x-axis & y-axis.

Syntax : axis ([x<sub>min</sub>, x<sub>max</sub>, y<sub>min</sub>, y<sub>max</sub>]) :

~~x~~  
x = -5: 0.1: 5;

y = x.^2;

plot(x, y)

axis([-5, 5, 0, 25])

xlabel('x')

ylabel('y')

title('Plot of y = x^2')

AJAYA

`title ('figure title')` - add title of figure:

% title of figure

`title ('It's a title');`

`surf(x,y,z)` create a three dimensional surface plot, which is three dimensional surface that has solid edge colour and solid face color.

Syntax: `surf(x,y,z)`

`surf(x,y,z,c)`

`surf(z)`

`surf(z,c)`

`surf(ax,-)`

`surf(- Name, value)`

~~`s = surf(-)`~~

Save work

Syntax

`save (filename)`

`save (filename, variable)`

~~`save (filename, variable, fof )`~~

`save ( filename, variable, version )`

`save (filename, variable, version, nocompression )`

`save (filename, variable append )`

~~Save file name.~~

Save all workspace variable to Mac file.

Ajaya

Expt. No. J.

# file name = 'test.mat'  
Save (file name)

# save test.mat

# load ('test.mat').

(A+) DS  
20/3/2020

Object : Introduction of image processing with matlab.

- Image processing toolbox-function : It support the wide range of image processing operation i.e geometric operation : geometric operations applied to images are typically used to transform an image and align it with another image for reconstruction or comparison to line up features for stitching or to simply create an effect such as morphing.

Neighbourhood & block operation : A sliding neighbourhood operation processes an image one pixel at a time, by applying an algorithm to each pixels neighbourhood. In distinct block processing, an image is divided into equally sized blocks without overlap, and the algorithm is applied to each distinct block.

Linear filtering & filter design : Linear filtering is filtering in which the value of an output pixel is a linear combination of the values of the pixels in the input pixel's neighborhood. Filtering is a technique for modifying or enhancing an image.

Transformation : Transformation is a function. A function that maps one set to another set after

performing some operation.

$s = c \log(r+1)$  is the formula of transformation in image processing. where  $s$  and  $r$  are the pixel values of the output and the input image &  $c$  is a constant.

**Image analysis and Enhancement :** image analysis involves processing an image into fundamental components to extract meaningful information. Image enhancement is the process of adjusting digital images so that the results are more suitable for display or further image analysis.

**Region of interest :** It is a portion of image that you want to filter or operate on in some way. You can represent an ROI as a binary mask image.

- Matlab supports various image formats :

BMP (Windows Bitmap).

GIFF (Graphics Interchange File).

JPG or JPEG.

PCX - (Point brush).

PNG - (Portable network graphics).

TIFF - (Tagged image file format).

Raw data of image.

- Data types in Matlab :

int 32 (32 bit single Integer).

int 16 (16 bit single Integer).

int 8 (8 bit single Integer).

double (64 bits).

single (32 bits).

• uint 32 (32 bit unsigned integer.)

• uint 16 (16 bit unsigned integer).

• uint 8 (8 bit unsigned integer)

- Read and write image in matlab:

In order to read images we are going to use the imread() function in MATLAB. The imread() function reads images from the graphics files.

Syntax:  $A = \text{imread}(\text{filename})$

$A = \text{imread}(\text{filename}, \text{fmt})$

$A = \text{imread}(\text{Name}, \text{Value})$

$\text{dim} = \text{size}(\text{img});$

figure;

$\text{imshow}(\text{img});$

$\text{imwrite}(\text{img}, \text{'output.bmp'}, \text{'bmp'})$ ;

writes the indexed image in A and its associated colormap to the file specified by filename.

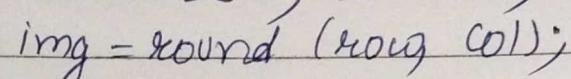
- Build a image(matrix)  
(Intensity Image)

Expt. No. \_\_\_\_\_

```

row = 256;
col = 256;
img = zeros(row, col);
img(100:105, :) = 0.5;
img(:, 100, :105) = 1;
figure;
imshow(img);

```

Binary image : row=256; col=256;  
 = round(row col);  
% img = round(img);  
figure;  
imshow(img);

### Image conversion :

- gray2ind : intensity to index image.
- img2bw : image to binary image.
- img2double : image to double precision.
- img2uint8 : image to 8 bit unsigned integer.
- img2uint16 : image to 16 bit unsigned integer.
- img2gray : index image to intensity image.
- rgb2gray : RGB to gray scale
- rgb2ind : RGB to indexed image.

### Image operation :

- RGB to gray image.

Ayaya

Expt. No. 2.

- image Resize.
- image crop.
- image rotate.
- image histogram.
- image histogram equalization.
- image DCT/FDCT.
- convolution .