

INTRO TO . . .

MATLAB

---

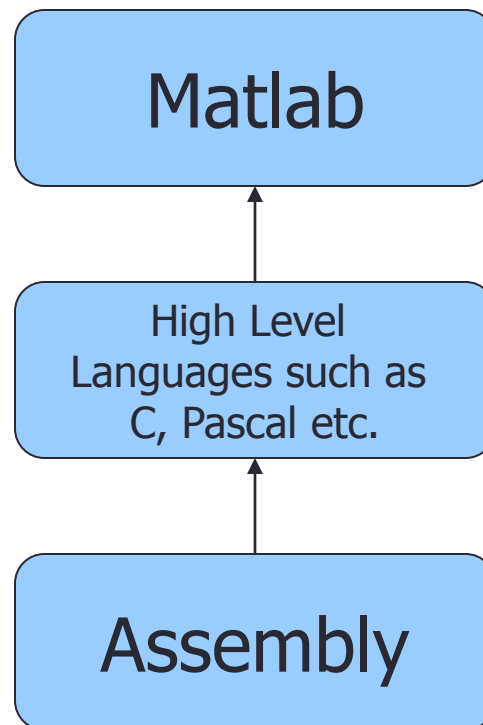
By SARMAD ABBAS  
sarmadkhitran@gmail.com

# Outline:

- What is Matlab?
  - Matlab Screen
  - Variables, array, matrix, indexing
  - Operators (Arithmetic, relational, logical )
  - Display Facilities
  - Flow Control
  - MatLab Functions
  - Using of M-File
  - Writing User Defined Functions
  - Image Processing Toolbox

# What is Matlab?

- Matlab is basically a **high level language** which has many specialized **toolboxes** for making things easier for us
- How high?



# Mat - Lab

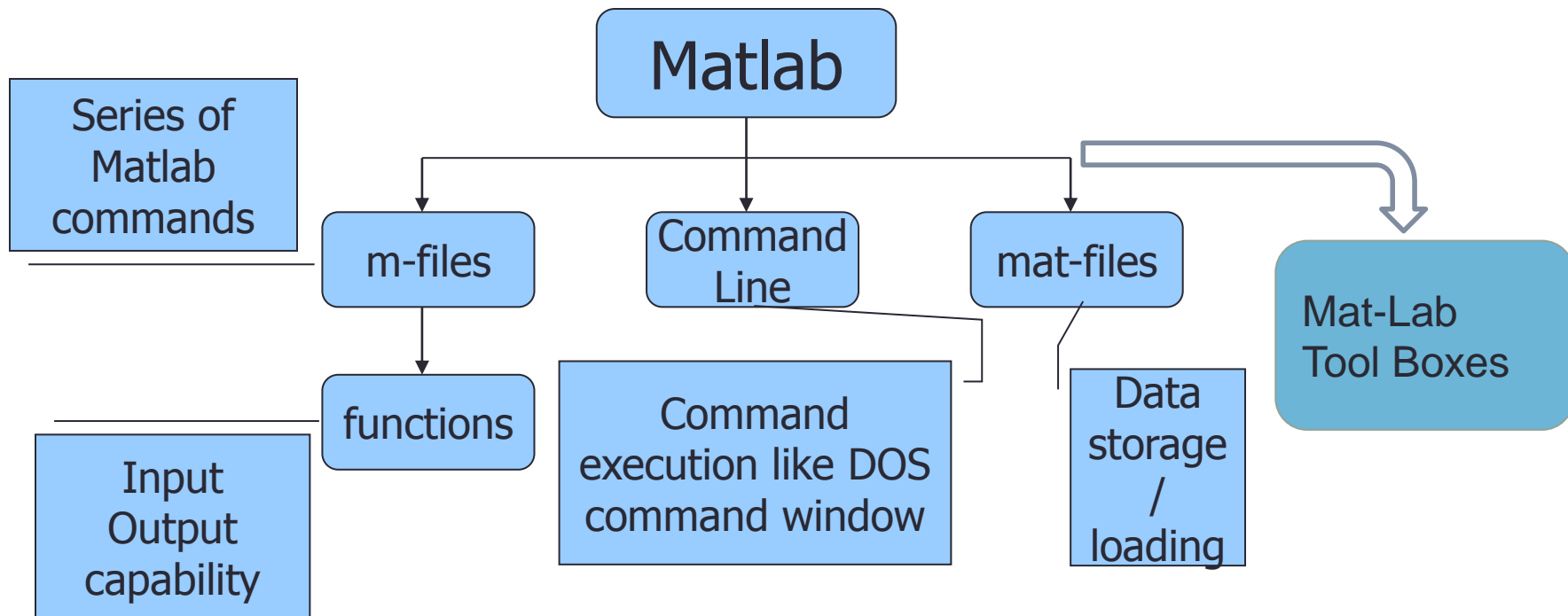
- The MATLAB system is technically called an *interpreter*
- *Unlike other programming languages.*

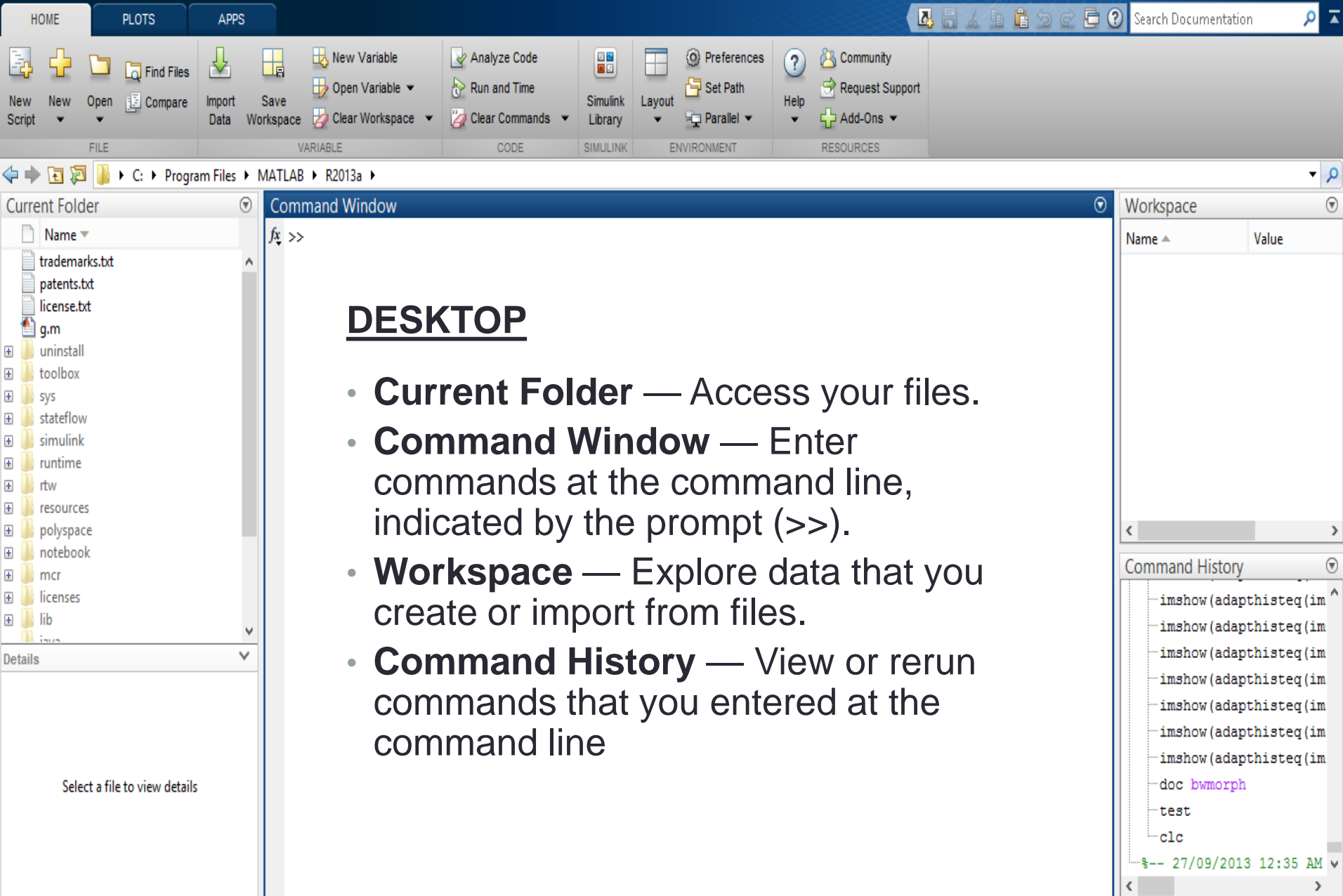
# Why Matlab

- Math and Computations
- Algorithm development
- Simulation
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interfaces building

# What are we interested in?

- Matlab is too broad for our purposes
- The features we are going to require is





The screenshot displays the MATLAB R2013a software interface. The top menu bar includes options like HOME, PLOTS, and APPS. Below it is a toolbar with various icons for file operations, workspace management, and code execution. The main workspace is divided into three panels: Current Folder, Command Window, and Workspace. The Command Window is the central focus, showing a prompt `>>` and a large text overlay. The Current Folder panel on the left shows a directory tree with files like `trademarks.bt` and folders like `toolbox`. The Workspace panel on the right is empty, showing columns for Name and Value. The Command History panel at the bottom right shows a list of executed commands, including `imshow` and `doc bwmorph`.

**DESKTOP**

- **Current Folder** — Access your files.
- **Command Window** — Enter commands at the command line, indicated by the prompt (`>>`).
- **Workspace** — Explore data that you create or import from files.
- **Command History** — View or rerun commands that you entered at the command line

# Some Basic Commands

- Creating Variables

- $a = 1$
- $b = 2$
- $c = a + b$
- $d = \cos(a)$
- $\sin(a)$
- $e = a * b;$

# Matlab Special Variables

- ans            Default variable name for results
- pi             Value of  $\pi$
- inf            Infinity
- NaN          Not a number e.g. 0/0
- realmin      The smallest usable positive real number
- realmax      The largest usable positive real number

# Case sensitivity

# Matrices and Arrays

- MATLAB - "matrix laboratory."
- other programming languages.
- MATLAB - matrices and arrays.
- All MATLAB variables are multidimensional arrays, no matter what type of data.
- A matrix is a two-dimensional array often used for linear algebra.

# Matrixes/Arrays

- a row vector.

- `a = [1 2 3 4]`
- `B = 0:10:100`
- `z = zeros(5,1)`
- `B = rand(3,5);` (0,1)
- `linspace(0, pi/2, 10)`

- a matrix

- `a = [1 2 3; 4 5 6; 7 8 10]`
- `A = ones (5,5)`
- `Z = zeros(2,5)`
- `true(2, 4)`
- `false (5,6)`
- `rand (5,6)`
- `magic(n)` **`nXn matrix of integers (1 through n^2)`**

# Array Indexing

- `A = magic(4)`
- `A(4,2)`
- `A(8)`                    **traverses down each column in order**
- `t = A(4,5)`
- `A(4,5) = 17`
- `A(1:3,2)`                    **vector notation**
- `A(3,:)`
- `B = 0:10:100`

# Some basic Operations with Matrixes

- $a + 10$
- $\sin(a)$
- $a'$
- $p = a * \text{inv}(a)$
- $A = [a, a]$
- $A = [a; a]$
- $\text{sqrt}(-1)$
- $c = [3+4i, 4+3j, -i, 10j]$
- `format long`
- `format short`

transpose

Inverse of Matrix

Matrix Concatenation

Matrix Concatenation

Complex Numbers

# Data Types

Name	Description
double	Double-precision, floating-point numbers in the approximate range $-10^{308}$ to $10^{308}$ (8 bytes per element).
uint8	Unsigned 8-bit integers in the range [0, 255] (1 byte per element).
uint16	Unsigned 16-bit integers in the range [0, 65535] (2 bytes per element).
uint32	Unsigned 32-bit integers in the range [0, 4294967295] (4 bytes per element).
int8	Signed 8-bit integers in the range [-128, 127] (1 byte per element).
int16	Signed 16-bit integers in the range [-32768, 32767] (2 bytes per element).
int32	Signed 32-bit integers in the range [-2147483648, 2147483647] (4 bytes per element).
single	Single-precision floating-point numbers with values in the approximate range $-10^{38}$ to $10^{38}$ (4 bytes per element).
char	Characters (2 bytes per element).
logical	Values are 0 or 1 (1 byte per element).

# Loops / if else

```
A = 2 : 2 : 20;
```

```
i = 1;
```

```
While (i < 10)
```

```
    disp(A(i));
```

```
    i = i+1;
```

```
end
```

```
////////////////////////////////////
```

```
for i = 1:10
```

```
    disp (A(i))
```

```
end
```

```
////////////////////////////////////
```

```
If (i == 10)
```

```
    disp 'I = 10';
```

```
end
```

```
////////////////////////////////////
```

```
If (i == 10)
```

```
    disp 'I = 10';
```

```
elseif(i == 11)
```

```
    - - - - -;
```

```
else
```

```
    - - - - -;
```

```
end
```

# Character Strings

- `myText = 'Hello, world';`
- `otherText = 'You"re right'`
- `longText = [myText, ' - ', otherText]`
  - `f = 71;`
  - `c = (f-32)/1.8;`
  - `tempText = ['Temperature is ', num2str(c), 'C']`

# Calling Functions

- `A = [1 3 5]; B = [10 6 4];`
- `[maxA, location] = max(A);`
- `disp('hello world');`
- `m = mean(A);`
- `length (A)`
- `Whos`
- `clc`

# HELP

- doc **mean**
- mean(
- help **mean**
- Online documentation for Matlab at the MathWorks website
- <http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.html>

# Data Type conversions

- Char, int , uint, cell, double, long
- Uint8(A);
- Num2str
- Str2num
- Num2cell

# Plotting

- `x = 0:pi/100:2*pi;`
- `y = sin(x);`
- `plot(x,y)`
- `xlabel('x')`
- `ylabel('sin(x)')`
- `title('Plot of the Sine Function')`
- `plot(x,y,'r--')`
- **3-D Plots**
- `[X,Y] = meshgrid(-2:.2:2);`
- `Z = X .* exp(-X.^2 - Y.^2);`
- `surf(X,Y,Z)`
- `subplot(2,2,1);`
- `hold on`

# Some More Functions

- `A = 1 : 0.2 : 10;`
- `length (A)`
- `Size (A)`
- `Isempty (A)`
- `find (A == 2)`
- `Sort (A)`
- [Diff](#) A
- `Sum`
- `round`
- `Ceil`
- `Fix`
- `Floor`
- `Strcomp`
- `Mean`
- `Median`
- `Min`
- `Max`
- `Std`
- `Sum`
- `var`

`round toward zero`  
`round toward -ive inf`

# The input statement

- `balance = input( 'Enter bank balance: ' );`
- `disp( 'Bank balance:' );`
- `disp( balance );`

# Trigonometric/ Logarithmic

- Sin
- sind
- Cos
- cosd
- Tan
- tand
- Asin
- Atan
- Acos
- Asind
- Sinh
- Exp
- Log
- Log10
- Log2
- sqrt

# Complex Numbers

- Abs
- Angle
- Conj
- $i, j$
- Imag
- Real

# Constants

- `i, j`
- `inf`
- `pi`
- `NaN`
- `-inf`

# Arithmetic Operators

- $A+B$
- $A-B$
- $A*B$
- $A.*B$
- $A/B$
- $A./B$
- $A\backslash B$
- $A.\backslash B$
- $A^B$
- $A.^B$
- $A'$

# Relational Operators

- $A < B$
- $A > B$
- $A \leq B$
- $A \geq B$
- $A == B$
- $A \sim B$

# Logical Operators: Elementwise & | ~

- `expr1 & expr2`
- `expr1 | expr2`
- `~expr`

# Bit-Wise Operations

- `bitand` Bit-wise AND
- `bitcmp` Bit-wise complement
- `bitget` Get bit at specified position
- `bitor` Bit-wise OR
- `bitset` Set bit at specific location
- `bitxor` Bit-wise XOR

# M-files / function M-files

- **function** [out1,out2, ...] = FunctionName (in1, int, ....)

Starement 1 ;

Starement 2 ;

----- ;

- **end**

# Fourier Analysis Filtering

- `fft`
- `ifft`
- `conv`
- `fft2`
- `fftshift`
- `ifft2`
- `conv2`
- `imfilter` (Image, Mask)

# Digital Image Processing Using Matlab

- Image processing ToolBox.
- `im=imread('image.jpg');` Load image
- `R=im(:, :, 1);` red plane
- `G=im(:, :, 2);` green plane
- `B=im(:, :, 3);` blue plane
- `[r c v]=size(im);` rows, columns, planes
- `imshow(im);` **display** image
- `imwrite( im,'image.jpg');`
- `rgb2gray(im);` RGB to Grayscale Conversation
- `Im=uint8(zeros (40, 20))`  
(double,uint8,uint16,uint32,int8,int16,int32,single,char,logical)
- `im=uint8(im);`

- `zeros(r,c,'uint8');`
- `imadd (im1,im2);`                      to add two images.
- `imsubtract (im1,im2);`
- `imcomplement (im);`                      image negative.
- `subplot(x,y,n)`                              **n from [1, x\*y]**
- `figure`
- `title ('add title of the figure here.');`

- `im2bw(im_gray, T)` ; to convert grayscale image into BW.
- `T=graythresh(im_g)`; to set the gray threshold.
- `imcrop(im)`; to crop an image.
- `Imcrop(im, dim)`; to crop with dimension given to it
- `imtool(im_g)`; image tool.
- `imrotate()`; to rotate an image.
- `help imcrop`; use matlab help.