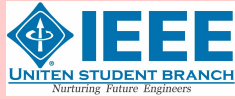


MATLAB: GETTING STARTED

WELCOME AND GOODLUCK



Syed Khaleel Ahmed

Dept. of Electronics and Communication Engg.,
Universiti Tenaga Nasional
syedkhaleel@uniten.edu.my syedkhaleel2000@gmail.com
http://metalab.uniten.edu.my/~syedkhaleel/

November 5, 2016

INTRODUCTION TO MATLAB

- 1 What is MATLAB?
- 2 What is available in MATLAB?
- 3 Why MATLAB?
- 4 Where MATLAB?

INTRODUCTION TO MATLAB



What is MATLAB?

A high-level software for

- *numerical computation,*
- *data analysis,* and
- *visualization.*

INTRODUCTION TO MATLAB



What is available in MATLAB?

- Pre-defined functions.
- Toolboxes.
- SIMULINK.
- Blocksets.

Why MATLAB?

- de-facto industry standard, especially in engineering,
- easy to use, and
- availability of toolboxes and blocksets.

Where MATLAB?

- Automotive,
- Signal Processing,
- Communication,
- Aerospace,
- Finance and Economics,
- Computer, and
- *many more.*

Course Contents

- 1 MATLAB User Interface Layout.
- 2 Working with Variables.
- 3 Visualizing Data.
- 4 Programming.

- Basic Components of MATLAB
- MATLAB as a Calculator
- Operators and Operator Precedence
- Pre-defined Functions

Basic Components of MATLAB

- To start
 - *Double-click MATLAB on desktop* or
 - *click start menu.*
- MATLAB user interface or desktop environment
 - *Command Window*
 - *Command History*
 - *Current Directory browser*
 - *Workspace Browser*

MATLAB as a Calculator

```
>> 2*2 - 4/3
```

```
ans =
    2.6667
```

```
>> 16^(1/4) + 3*sin( pi/ 4 )
```

```
ans =
    4.1213
```

```
>> sqrt( 4 ) + exp( j*pi/6 )
```

```
ans =
    2.8660 + 0.5000i
```

Pre-defined Constants

π	pi
i, j	i, j
smallest value	eps
largest real number	realmax
smallest real number	realmin
largest integer	intmax
smallest integer	intmin

Operators and operator precedence

- | | |
|---------------------------------|------------|
| ① () parenthesis | >> 2*3+2 |
| ② ´ complex conjugate transpose | >> 2*(3+2) |
| ③ ^ power | >> 2/3 |
| ④ * multiplication; / division | >> 2\3 |
| ⑤ \ left division | >> 2*3^4 |
| ⑥ + addition; - subtraction | >> (2*3)^4 |

Pre-defined Functions

- Trigonometric – COS, ACOS, EXP, SIN, ASIN
- Exponential – LOG, EXP, SQRT
- Complex – ABS, ANGLE, CONJ
- Discrete Maths – FACTOR, PRIMES, GCD

Want more information? Type

```
>> help elfun
>>
>> doc elfun
>>
>> help angle
```

- Creating and Manipulating Variables.
- Accessing and Manipulating Elements in a Matrix.
- Computations with Matrices.

Creating and Manipulating Variables

All variables in MATLAB are arrays (matrices).

- A *scalar* is a 1×1 array.

```
>> a = 1
a =
    1
```

- A (*column*) *vector* is an $n \times 1$ array.

```
>> b = [ 1; 2 ]
b =
    1
    2
```

- A *row vector* is a $1 \times m$ array.

```
>> c = [ 1 2 3 ]
c =
    1    2    3
```

Creating and Manipulating Variables (contd.)

All variables in MATLAB are arrays (matrices).

- A *matrix* is an $n \times m$ array.

```
>> d = [ 1 2; 3 4 ]
```

```
d =
    1    2
    3    4
```

- *Strings* are arrays of characters.

```
>> str = 'Hello World!'
```

```
str =
Hello World!
```

Creating and Manipulating Variables (contd.)

```
>> x = 1
```

```
x =
    1
```

```
>> y = 4;
```

```
>> r = sqrt( x^2 + y^2 )
```

```
r =
    4.1231
```

```
>> fx = cos( 2*x + pi/4 )
```

```
fx =
   -0.9372
```

Creating and Manipulating Variables (contd.)

```
>> x = [ 1 2 3 4 5 ]
>> x = 1:5
>> x = 0:0.25:1
>> x = linspace( 0, 1, 5 )
>> x = logspace( -1, 2, 4 )
```

Exercises:

- Create the vector $x = [10 \ \pi \ \sin(30^\circ) \ \sqrt{2}]$.
- Create the vector

$$y = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 10 \end{bmatrix}.$$

- Grid the interval from 1 to 5 using 11 points.
- Create a vector w with first element 0, last element 4 & increment 0.5

Creating and Manipulating Variables (contd.)

Special Matrices

```
>> A = zeros( 3 )
>> B = ones( 2, 4 )
>> C = rand( 1, 4 )
>> D = magic( 4 )
>> E = eye( 2 )
>> X = [ ones( 2 ) zeros( 2, 3 ) rand( 2, 1 ) ]
>> sparse( X )
```

Accessing and Manipulating Elements in a Matrix

Array elements are accessed through indices.

- A single matrix element.
- A sub-matrix.
- Re-order elements.

Accessing and Manipulating Elements in a Matrix (contd.)

```
>> A = rand( 4, 5 )
>> A( 2:3, 3:4 )
>> A( 2, 3 )
>> A( 2, 3 ) = 5
>> A( 1, : )
>> A( :, 2 )
>> A( 2:end, : )
>> A( end:-1:1, : )
>> A( [ 1 3 ], : )
>> A( 3, [ 2 4 ] )
>> A( [ 1 3 ], [ 4 2 ] )
```

Exercises:

- Create a random 2×3 matrix A .
- Modify a_{23} to π .
- Invert the order of the columns of A .

Computations with Matrices

Two types of computations

Suppose

A_1 and A_2 are two matrices of order $m \times n$,

B is of order $n \times p$,

C is $n \times n$, and

α is a scalar.

- Matrix computations – *Mathematically defined.*

Examples: $A_1 + A_2$, $A_1 B$, C^{-1} , αA , ...

- Element-wise computations – *useful for speeding up computations.*

Examples: $a_{kl} b_{kl}$, ...

Computations with Matrices (contd.)

Examples: Create the following matrices

- a random 2×3 matrix A ,
- a random 3×2 matrix B ,
- $C = AB$,
- $D = C^{-1}$
- $E = \epsilon D$
- $F = D + 2I$
- H such that $h_{kl} = a_{kl}^2$.

Want to know more matrix functions, type

```
>> help matfun
```

```
>>
```

```
>> doc matfun
```

VISUALIZING DATA

- Importing data into MATLAB.
- Basic plotting commands.
- Customizing plots.
- Types of Plots.
- Exporting Plots to other applications.
- Saving and loading data.

VISUALIZING DATA (CONTD.)

Importing Data into MATLAB

- Using the *Import Wizard*.
 - Click *HOME* tab.
 - Click *Import Data* icon.
 - Browse to folder and choose file.
- Using MATLAB commands – *importdata*.


```
>> mydata = importdata('studmarks.txt');
```
- Alternate commands *xlsread*, *csvread*, *dlmread*.
- To see supported file formats, type


```
>> doc fileformats
```

Basic plotting commands

- Plotting a sinusoidal function $y = \sin(x)$.

```
>> x = 0:0.2:2*pi;
>> y = sin( x );
>> plot( x, y )
```
- Different looks

```
>> plot( x, y, 'r' )
>> plot( x, y, ':' )
>> plot( x, y, 'x' )
```
- Standard form: `plot(xdata, ydata, ' <color><linestyle><marker>')`.

```
>> plot( x, y, 'g-.o' )
```
- For more information

```
>> help plot
```

Basic plotting commands (contd.)

- Drawing multiple plots on the same graph:

$$y = \sin(x) \text{ and } z = \cos(x).$$

```
>> x = 0:0.2:2*pi;
>> y = sin( x );
>> z = cos( x );
```

- Does this work?

```
>> plot( x, y )
>> plot( x, z )
```

Basic plotting commands (contd.)

- Drawing multiple plots on the same graph:

$$y = \sin(x) \text{ and } z = \cos(x).$$

- What about this?

```
>> plot( x, y, x, z )
>> plot( x, y, 'r:o', x, z, 'm--s' )
```

- Or this?

```
>> plot( x, y, 'r:o' )
>> hold on
>> plot( x, z, 'm--s' )
>> hold off
```

Customizing Plots

```
>> x = 0:0.2:2*pi;
>> y = sin( x );
>> z = cos( x );
>> plot( x, y, 'r:o', x, z, 'm--s' )
```

- Adding a grid

```
>> grid
```
- Label the axes

```
>> xlabel('x values')
>> ylabel('y values')
```
- A title

```
>> title('Plot of sinusoidal functions')
```
- Legend for multiple graphs

```
>> legend('sin(x)', 'cos(x)')
```

Basic plotting commands (contd.)

- Drawing multiple graphs in the same window:

$$y = \sin(x) \text{ and } z = \cos(x).$$

```
>> x = 0:0.2:2*pi;
>> y = sin( x );
>> z = cos( x );

>> subplot(221), plot( x, y, 'r:o' )
>> subplot(222), plot( x, z, 'm--s' )
>> subplot(223), plot( x, y+z, 'b--s' )
>> subplot(224), plot( x, y.*z, 'm-.h' )
```

Types of Plots

```
>> x = 0:0.2:2*pi;

>> y = sin( x );

>> plot( x, y, 'r:o' )

>> stem( x, y )

>> bar( x, y )

>> stairs( x, y )

>> area( x, y )
```

Exporting to other applications

In Figure window,

- choose *File* → *Save As ...*,
and select desired file type, or
- choose *Edit* → *Copy Figure*,
then paste in desired file.

Saving and Loading Data**Saving Data.**

```
>> x = 0:0.2:2*pi;
>> y = sin( x );
>> z = cos( x );
>> save
```

Saving to: matlab.mat

```
>> save mydata
>> save mydata2 x y
>> save mydat3 x y -ascii
```

Loading Data.

```
>> clear
>> load mydata2
>>
>> clear
```


- The MATLAB editor.
- Script m-files.
- Function m-files.

The MATLAB Editor

- For writing MATLAB programs.
- Works like any normal text editor.
- Two types of programs
 - Script m-files.
 - Function m-files.
- To open the editor,
 - Click *File* → *New* → *Blank M-file*, or
 - Click *New M-File* icon.

Script m-files

Type the following in the editor.

```

%% This is my first MATLAB program
% First clear the mess
close all % Closes all figure windows
clear % clear the workspace
clc % clear the command window

%% Display message
disp('Look mommy! Now I can write script m-files!!!')

%% Determine the variables
x = 0:0.2:2*pi; y = sin( x ); z = cos( x );

%% Plot the figures
plot( x, y, 'r:o', x, z, 'm--s' ), grid
xlabel('x values'), ylabel('y values')
title('Plot of sinusoidal functions')
legend('sin(x)', 'cos(x)')

```

Script m-files (contd.)

- To save the program (use *myfirst.m* as the name).
 - click on *File* → *Save*, or
 - click the *Save icon*, or
 - type *Ctrl-s*.
- To run the program
 - Type the file name (*myfirst*) in the command window, or
 - click the *Run myfirst.m* icon.

Script m-files (contd.)

- For clarity of the program, add the following (comments) at the top.

```
%% MYFIRST
%
% This program plots the sinusoidal functions sin(x) and cos(x)
% for x = 0 to pi in increments of 0.2 radians.
%
% Written by ...
% June 22, 2011
% Last modified ...
%
```

Function m-files

- Type the following in the editor,


```
function y = mymean( x )
% Calculates the average of the numbers in a vector
[ m, n ] = size( x ); % No of rows & columns in x.
if m==1 | n==1
    k = max( m, n ); % Number of elements
    y = sum( x )/k;
else
    disp('x must be a vector.')
```
- The first line of a function must be


```
function [outputs] = function_name( inputs )
```
- Save the file as *mymean.m*

Function m-files (contd.)

- To execute the program, type


```
>> x = rand( 1, 4 )
>> mymean( x )
>> y = rand( 4, 1 )
>> mymean( y )
>> z = rand( 2, 4 )
>> mymean( z )
```

Script m-files & Function m-files Comparison

Script m-files	Function m-files
No restriction on structure	First line must be the function definition.
Variables are transparent with the workspace. Workspace variables can be used. script file variables are available in the workspace.	Variables are local. Variables are exchanged through input and output arguments.
No restriction on name	File name must be same as function name.