Lecture (01)
An Overview of MATLAB

By:
Dr. Ahmed ElShafee

What Is MATLAB?

- MATLAB (MATrix LABoratory)
- high-performance language for technical computing
- computation, visualization, and programming in an easy-to-use environment
Typical uses include:

- Math and computation
- Algorithm development
- Modelling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including Graphical User Interface building

Advantages
A good choice for vision program development because:

- Easy to do very rapid prototyping
- Quick to learn, and good documentation
- A good library of image processing functions
- Excellent display capabilities
- Widely used for teaching and research in universities and industry
- Another language to impress your manager with!
drawbacks:
- Slow for some kinds of processes
- Not geared to the web
- Not designed for large-scale system development

MATLAB consists of:
**The MATLAB language**
- a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features.
The MATLAB working environment

- the set of tools and facilities that you work with as the MATLAB user or programmer, including tools for developing, managing, debugging, and profiling.

Handle Graphics

- the MATLAB graphics system.
- It includes high-level commands for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics.
The MATLAB function library.

• a vast collection of computational algorithms ranging from elementary functions like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigenvalues, Bessel functions, and fast Fourier transforms as well as special image processing related functions

The MATLAB Application Program Interface (API)

• a library that allows you to write C and Fortran programs that interact with MATLAB.

• It include facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.
Some facts for a first impression

• Everything in MATLAB is a matrix!
• MATLAB is an interpreted language, no compilation needed (but possible)
• MATLAB does not need any variable declarations, no dimension statements, has no packaging, no storage allocation, no pointers
• Programs can be run step by step, with full access to all variables, functions etc.

MATLAB Interactive Session

• To start MATLAB on a MS Windows system, double-click on the MATLAB icon.
• You will then see the MATLAB Desktop
• You can manipulate other windows in the same way.
• To restore the default configuration, click on the Desktop menu, then click on Desktop Layout, and select Default.

• Work with matlab as simple calculator
• Exploring (return-up arrow- down arrow-left arrow – right arrow)

![Command Window and Workspace](image)

• Note that the variable $\text{ans}$ now has the value 4.
• You can use variables to write mathematical expressions. Instead of using the default variable $\text{ans}$, you can assign the result to a variable of your own choosing, say, $r$,  

![Command Window and Workspace](image)
Note:

- you can put a space before and after the = sign if you want. MATLAB ignores these spaces when making its calculations.
- It also ignores spaces surrounding % and ! signs.

Calling and reusing variables:
MATLAB has hundreds of functions available. One of these is the square root function, sqrt.
• Note: consider the difference between left and right division

```
Command Window
>> 7/2
ans =
    3.5000
>> 2 \ 7
ans =
    3.5000
fs >>
```

• Order of precedence

<table>
<thead>
<tr>
<th>Precedence</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>First</td>
<td>Parentheses, evaluated starting with the innermost pair.</td>
</tr>
<tr>
<td>Second</td>
<td>Exponentiation, evaluated from left to right.</td>
</tr>
<tr>
<td>Third</td>
<td>Multiplication and division with equal precedence, evaluated from left to right.</td>
</tr>
<tr>
<td>Fourth</td>
<td>Addition and subtraction with equal precedence, evaluated from left to right.</td>
</tr>
</tbody>
</table>
Example 01

Use MATLAB to compute the following expressions.

a. \( 6 \left( \frac{10}{13} \right) + \frac{18}{5(7)} + 5(9) \)

b. \( 6(35^{1/4}) + 14^{0.35} \)
Dr. Ahmed ElShafee, ACU: Spring 2016, Introduction To Engineering

Command Window

New to MATLAB? Watch this Video, see Demos, or read Getting Started.

```
>> (6*(10/13)+(18/(5*7))+(5*(9^2))
ans =
410.1297
```

```
>> (6*(35^(1/4)))+(14^0.35)
ans =
17.1123
```
The Assignment Operator

- The = sign in MATLAB is called the assignment or replacement operator.
- When you type \( x = 3 \), you tell MATLAB to assign the value 3 to the variable \( x \).
- When you type : \( x = x + 2 \). This tells MATLAB to add 2 to the current value of \( x \), and to replace the current value of \( x \) with this new value.
- If you type: \( >> x = 5 + y \) if the variable \( y \) has not been assigned a value, then the following will generate an error message in

Example 02

- The volume of a circular cylinder of height \( h \) and radius \( r \) is given by \( V = \pi r^2 h \).
- A particular cylindrical tank is 15 m tall and has a radius of 8 m. We want to construct another cylindrical tank with a volume 20 percent greater but having the same height. How large must its radius be?
Example 02

- First solve the cylinder equation for the radius $r$.

\[
    r = \sqrt{\frac{V}{\pi h}}
\]

```matlab
>> r = 8;
>> h = 15;
>> V = pi*r^2*h;
>> V = V + 0.2*V;
>> r = sqrt(V/(pi*h))
r =
     8.7636
```

Variable Names

- Variable names must begin with a letter; the rest of the name can contain letters, digits, and underscore characters. MATLAB is case sensitive.
- speed, Speed, SPEED, Speed_1, and Speed_2.
Commands for managing the work session

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>clc</td>
<td>Clears the Command window.</td>
</tr>
<tr>
<td>clear</td>
<td>Removes all variables from memory.</td>
</tr>
<tr>
<td>clear var1 var2</td>
<td>Removes the variables var1 and var2 from memory.</td>
</tr>
<tr>
<td>exist('name')</td>
<td>Determines if a file or variable exists having the name 'name'.</td>
</tr>
<tr>
<td>quit</td>
<td>Stops MATLAB.</td>
</tr>
<tr>
<td>who</td>
<td>Lists the variables currently in memory.</td>
</tr>
<tr>
<td>whos</td>
<td>Lists the current variables and sizes and indicate if they have imaginary parts.</td>
</tr>
<tr>
<td>:</td>
<td>Colon; generates an array having regularly spaced elements.</td>
</tr>
<tr>
<td>,</td>
<td>Comma; separates elements of an array.</td>
</tr>
<tr>
<td>;</td>
<td>Semicolon; suppresses screen printing; also denotes a new row in an array.</td>
</tr>
<tr>
<td>...</td>
<td>Ellipsis; continues a line.</td>
</tr>
</tbody>
</table>

```matlab
>> exist('r')
ans =
1
>> exist('ans')
ans =
1
>> exist('r')
ans =
1
>> exist('s')
ans =
1
>> exist('k')
ans =
0
```
Command Window

New to MATLAB? Watch this Video.

>> exist('s');
>> exist('s')

ans =

1
>> clear s
>> exist('s')

ans =

0

>> x = 2; y = 6 + x, x = y + 7

Y =

8
x =

15
Constants

- MATLAB has several special constants, such as the built-in constant \( \pi \)
- \( \infty \) stands for infinity
- The symbol \( \text{NaN} \) stands for “not a number.”
• The symbols $i$ and $j$ denote the imaginary unit,

```matlab
>> sqrt(-1)
ans =
     0 + 1.0000i
```

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ans</td>
<td>Temporary variable containing the most recent answer.</td>
</tr>
<tr>
<td>$i, j$</td>
<td>The imaginary unit $\sqrt{-1}$.</td>
</tr>
<tr>
<td>Inf</td>
<td>Infinity.</td>
</tr>
<tr>
<td>NaN</td>
<td>Indicates an undefined numerical result.</td>
</tr>
<tr>
<td>pi</td>
<td>The number $\pi$.</td>
</tr>
</tbody>
</table>
Complex Number Operations

• complex number algebra automatically. For example, the number $c_1 = 1 + 2i$ is entered as follows: $c1 = 1-2i$. You can also type $c1 = \text{Complex}(1, -2)$.
• Note that an asterisk is not needed between i or j and a number

```matlab
>> s = 3+7i; w = 5-9i;
>> w+s
ans =
     8.0000 - 2.0000i
>> w*s
ans =
   78.0000 + 8.0000i
>> w/s
ans =
  -0.8276 - 1.0690i
```

Example 03

Given $x = -5 + 9i$ and $y = 6 - 2i$, use MATLAB to show that $x + y = 1 + 7i$, $xy = -12 + 64i$, and $x/y = -1.2 + 1.1i$. 
Formatting Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description and example</th>
</tr>
</thead>
<tbody>
<tr>
<td>format short</td>
<td>Four decimal digits (the default); 13.6745.</td>
</tr>
<tr>
<td>format long</td>
<td>16 digits; 17.27484029463547.</td>
</tr>
<tr>
<td>format short e</td>
<td>Five digits (four decimals) plus exponent; 6.3792e+03.</td>
</tr>
<tr>
<td>format long e</td>
<td>16 digits (15 decimals) plus exponent; 6.379243784781294e-04.</td>
</tr>
<tr>
<td>format bank</td>
<td>Two decimal digits; 126.73.</td>
</tr>
<tr>
<td>format +</td>
<td>Positive, negative, or zero; +.</td>
</tr>
<tr>
<td>format rat</td>
<td>Rational approximation; 43/7.</td>
</tr>
<tr>
<td>format compact</td>
<td>Suppresses some blank lines.</td>
</tr>
<tr>
<td>format loose</td>
<td>Resets to less compact display mode.</td>
</tr>
</tbody>
</table>
Dr. Ahmed ElShafee, ACU: Spring 2016, Introduction To Engineering

\[
\begin{align*}
\text{Command Window} \\
\text{New to MATLAB? Watch this Video, see Demos, or read Getting Started.} \\
\text{>> x=5.3} \\
\text{x} = \\
5.3000 \\
\text{>> format short} \\
\text{x} = \\
5.3000 \\
\text{>> format long} \\
\text{x} = \\
5.300000000000000000000000000000000000
\end{align*}
\]

\[
\begin{align*}
\text{Command Window} \\
\text{New to MATLAB? Watch this Video, see Demos, or read Getting Started.} \\
\text{>> format short e} \\
\text{>> x} \\
\text{x} = \\
5.3000e+000 \\
\text{>> format long e} \\
\text{x} = \\
5.300000000000000000000000000000000000e+000 \\
\end{align*}
\]

\[
\begin{align*}
\text{Command Window} \\
\text{New to MATLAB? Watch this Video, see Demos, or read Getting Started.} \\
\text{>> format bank} \\
\text{>> x} \\
\text{x} = \\
5.30 \\
\text{>> y=1256.5} \\
\text{y} = \\
1256.50 \\
\text{>> z=-55} \\
\text{z} = \\
-55
\end{align*}
\]
Arrays, Files, and Plots

Built in functions

- MATLAB has hundreds of functions
  - to compute \( \sin x \), where \( x \) has a value in radians, you type \( \sin(x) \) {radians}, \( \sin(x) \) {degree}.
  - To compute \( \cos x \), type \( \cos(x) \) {radians}, \( \cos(x) \) {degree}.
  - The exponential function \( e^x \) is computed from \( \exp(x) \).
  - The natural logarithm, \( \ln x \), is computed by typing \( \log(x) \).
  - You compute the base-10 logarithm by typing \( \log10(x) \).
  - The inverse sine, or arcsine, is obtained by typing \( \arcsin(x) \) {radian} \( \arcsin(x) \) {degree}.
<table>
<thead>
<tr>
<th>Function</th>
<th>MATLAB syntax*</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e^x$</td>
<td>exp (x)</td>
</tr>
<tr>
<td>$\sqrt{x}$</td>
<td>sqrt (x)</td>
</tr>
<tr>
<td>$\ln x$</td>
<td>log (x)</td>
</tr>
<tr>
<td>$\log_{10} x$</td>
<td>log10 (x)</td>
</tr>
<tr>
<td>$\cos x$</td>
<td>cos (x)</td>
</tr>
<tr>
<td>$\sin x$</td>
<td>sin (x)</td>
</tr>
<tr>
<td>$\tan x$</td>
<td>tan (x)</td>
</tr>
<tr>
<td>$\cos^{-1} x$</td>
<td>acos (x)</td>
</tr>
<tr>
<td>$\sin^{-1} x$</td>
<td>asin (x)</td>
</tr>
<tr>
<td>$\tan^{-1} x$</td>
<td>atan (x)</td>
</tr>
</tbody>
</table>

### Elementary math functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>sqrt (x)</td>
<td>Square root.</td>
<td>&gt;&gt; sqrt(81) ans = 9</td>
</tr>
<tr>
<td>exp (x)</td>
<td>Exponential ($e^x$).</td>
<td>&gt;&gt; exp(5) ans = 148.4132</td>
</tr>
<tr>
<td>abs (x)</td>
<td>Absolute value.</td>
<td>&gt;&gt; abs(-24) ans = 24</td>
</tr>
<tr>
<td>log (x)</td>
<td>Natural logarithm.</td>
<td>&gt;&gt; log(1000) ans = 6.9078</td>
</tr>
<tr>
<td>log10 (x)</td>
<td>Base 10 logarithm.</td>
<td>&gt;&gt; log10(1000) ans = 3.0000</td>
</tr>
<tr>
<td>factorial (x)</td>
<td>The factorial function $x!$ (x must be a positive integer.)</td>
<td>&gt;&gt; factorial(5) ans = 120</td>
</tr>
</tbody>
</table>
### Trigonometric math functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\sin(x))</td>
<td>Sine of angle (x) ((x) in radians)</td>
<td>(\gg \sin(\pi/6))\n(\text{ans} = 0.5000)</td>
</tr>
<tr>
<td>(\text{sind}(x))</td>
<td>Sine of angle (x) ((x) in degrees).</td>
<td></td>
</tr>
<tr>
<td>(\cos(x))</td>
<td>Cosine of angle (x) ((x) in radians).</td>
<td>(\gg \cosd(30))\n(\text{ans} = 0.8660)</td>
</tr>
<tr>
<td>(\text{cosd}(x))</td>
<td>Cosine of angle (x) ((x) in degrees).</td>
<td></td>
</tr>
<tr>
<td>(\tan(x))</td>
<td>Tangent of angle (x) ((x) in radians).</td>
<td>(\gg \tan(\pi/6))\n(\text{ans} = 0.5774)</td>
</tr>
<tr>
<td>(\text{tand}(x))</td>
<td>Tangent of angle (x) ((x) in degrees).</td>
<td></td>
</tr>
<tr>
<td>(\cot(x))</td>
<td>Cotangent of angle (x) ((x) in radians).</td>
<td>(\gg \cotd(30))\n(\text{ans} = 1.7321)</td>
</tr>
<tr>
<td>(\text{cotd}(x))</td>
<td>Cotangent of angle (x) ((x) in degrees).</td>
<td></td>
</tr>
</tbody>
</table>

### Rounding functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\text{round}(x))</td>
<td>Round to the nearest integer.</td>
<td>(\gg \text{round}(17/5))\n(\text{ans} = 3)</td>
</tr>
<tr>
<td>(\text{fix}(x))</td>
<td>Round toward zero.</td>
<td>(\gg \text{fix}(13/5))\n(\text{ans} = 2)</td>
</tr>
<tr>
<td>(\text{ceil}(x))</td>
<td>Round toward infinity.</td>
<td>(\gg \text{ceil}(11/5))\n(\text{ans} = 3)</td>
</tr>
<tr>
<td>(\text{floor}(x))</td>
<td>Round toward minus infinity.</td>
<td>(\gg \text{floor}(-9/4))\n(\text{ans} = -3)</td>
</tr>
<tr>
<td>(\text{rem}(x,y))</td>
<td>Returns the remainder after (x) is divided by (y).</td>
<td>(\gg \text{rem}(13,5))\n(\text{ans} = 3)</td>
</tr>
<tr>
<td>(\text{sign}(x))</td>
<td>Signum function. Returns 1 if (x &gt; 0), -1 if (x &lt; 0), and 0 if (x = 0).</td>
<td>(\gg \text{sign}(5))\n(\text{ans} = 1)</td>
</tr>
</tbody>
</table>
\begin{verbatim}
>> exp(1)
ans =
2.7183
>> sqrt(16)
ans =
4
>> log(exp(1))
ans =
1
>> log10(100)
ans =
2
\end{verbatim}

\begin{verbatim}
>> cosd(45)
ans =
0.7071
>> cos(45/180*pi)
ans =
0.7071
>> acosd(ans)
ans =
45
>> acos(0.7071)*180/pi
ans =
45.0005
\end{verbatim}
Arrays

- MATLAB is able to handle collections of numbers, called arrays, as if they were a single variable.
- A numerical array is an ordered collection of numbers.
- Use square brackets to define next the variable $x$ to contain this collection by typing $x = [0 \ 4 \ 3 \ 6]$
- The elements separated by spaces, or commas are $y = [6, 3, 4, 0]$
- The array $[0, 4, 3, 6]$ can be considered to have one row and four columns.

Arithmetic operations:
- We can add the two arrays $x$ and $y$ to produce another array $z$ by typing the single line $z = x + y$
Regularly spaced matrix

```matlab
>> x=1:10
x =
 1 2 3 4 5 6 7 8 9 10

>> y=0:2:10
y =
 0 2 4 6 8 10
```
Command Window

New to MATLAB? Watch this Video, see Demos, or read Getting Started.

```matlab
>> m=10:-1:0
m =
   10   9   8   7   6   5   4   3   2   1   0

>> m=10:-2:0
m =
   10   8   6   4   2   0
```

Command Window

New to MATLAB? Watch this Video, see Demos, or read Getting Started.

```matlab
>> theta=0:45:360
theta =
   0   45   90  135  180  225  270  315  360

>> A=sind(theta)
A =
   0   0.7071   1.0000   0.7071   0  -0.7071  -1.0000  -0.7071   0
```
Calling element from array

```
>> x
x =
   0   4   3   6
>> x(1)
ans =
   0
>> x(2)
ans =
   4
>> x(2:3)
ans =
   4   3
```
Number of Elements

Command Window

```
>> A
A =
0   0.7071    1.0000    0.7071     0  -0.7071   -1.0000   -0.7071     0

>> length(A)
ans =
9
```

Row array and Column Array

Command Window

```
>> phi=[0;45;90;135;180]

phi =
    0
    45
    90
   135
   180

>> B=cosd(phi)

B =
    1.0000
    0.7071
    0
   -0.7071
  -1.0000

>>
```
Polynomial Roots

- We can describe a polynomial in MATLAB with an array whose elements are the polynomial’s coefficients,

\[ 4x^3 - 8x^2 + 7x - 5 \]

\[ \text{array}[4, -8, 7, -5] \]

\[ \text{roots}(a) \]

```
x =
1.3880
0.3060 ± 0.8983i
```
\[(x - 4)(x + 2) = x^2 + 2x - 8\]

\[x = \begin{cases} 
1.0 \\
3.0 \pm 5.0i 
\end{cases}\]
Example 04

- Use MATLAB to determine how many elements are in the array \( \cos(0):0.02:\log_{10}(100) \).
- Use MATLAB to determine the 25th element.
Example 05

Use MATLAB to find the roots of the polynomial $290 - 11x + 6x^2 + x^3$. 

```
>> f=[1, 6, -11, 290]
f =
    1     6    -11    290
>> roots(f)
ans =
   -10.0000
    2.0000 + 5.0000i
    2.0000 - 5.0000i
```
Working with Files

- *.m files are used to save matlab code
- *.mat files are used to save matlab variables
\[
\begin{align*}
\theta &= 0:10:360; \\
A &= 5 \cos(d(\theta))
\end{align*}
\]
Dr. Ahmed ElShafee, ACU: Spring 2016, Introduction To Engineering

MATLAB 7.8.0 (R2009a)

Command Window

New to MATLAB? Watch this Video, see Demos, or read Getting Started.

>> program1

A =

Columns 1 through 9
5.0000 4.9240 4.6985 4.3301 3.8302 3.2139 2.5000 1.7101 0.8682

Columns 10 through 18
-0.8682 -1.7101 -2.5000 -3.2139 -3.8302 -4.3301 -4.6985 -4.9240

Columns 19 through 27
-5.0000 -4.9240 -4.6985 -4.3301 -3.8302 -3.2139 -2.5000 -1.7101 -0.8682

Columns 28 through 36
0.8682 1.7101 2.5000 3.2139 3.8302 4.3301 4.6985 4.9240

Column 37
5.0000

>> A(5)

ans =

3.8302

>>
Dr. Ahmed ElShafee, ACU : Spring 2016, Introduction To Engineering

Command Window

>> clear
f(2) >>

Current Directory

Name          Date Modified
10CosTheta.mat 09:39 02/03/16
program1.m     09:36 02/03/16

Command Window

New to MATLAB? Watch this Video, see Demos, or read Getting Started.

>> clear
>> theta=0:30:360
theta =
   0     30     60     90    120    150    180    210    240    270    300    330    360
>> A=10*cos(theta)
A =
    Columns 1 through 9
           10.0000    0.6603    5.0000         0   -3.0000   -5.6603   -10.0000   -8.6603   -5.0000
    Columns 10 through 13
              0   5.0000    8.6603    10.0000
>> save 10CosTheta
f(2) >>
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>addpath dirame</td>
<td>Adds the directory dirame to the search path.</td>
</tr>
<tr>
<td>cd dirame</td>
<td>Changes the current directory to dirame.</td>
</tr>
<tr>
<td>dir</td>
<td>Lists all files in the current directory.</td>
</tr>
<tr>
<td>dir dirame</td>
<td>Lists all the files in the directory dirame.</td>
</tr>
<tr>
<td>path</td>
<td>Displays the MATLAB search path.</td>
</tr>
<tr>
<td>pathtool</td>
<td>Starts the Set Path tool.</td>
</tr>
<tr>
<td>pwd</td>
<td>Displays the current directory.</td>
</tr>
<tr>
<td>rmpath dirame</td>
<td>Removes the directory dirame from the search path.</td>
</tr>
<tr>
<td>what</td>
<td>Lists the MATLAB-specific files found in the current working directory.</td>
</tr>
<tr>
<td>what dirame</td>
<td>Lists the MATLAB-specific files in directory dirame.</td>
</tr>
<tr>
<td>which item</td>
<td>Displays the path name of item if item is a function or le.</td>
</tr>
</tbody>
</table>

**Command Window**

```
>> pwd
ans =
C:\Users\AEliShafee\Documents\MATLAB

>> dir
.
..    1OCosTheta.mat program1.m

>> what
M-files in the current directory C:\Users\AEliShafee\Documents\MATLAB
program1
MAT-files in the current directory C:\Users\AEliShafee\Documents\MATLAB
1OCosTheta

>>
```
Plotting with MATLAB

```
theta=0:30:360
theta =
    0   30   60   90  120  150  180  210  240  270  300  330  360
A=sind(theta)
A =
Columns 1 through 9
    0   0.5000   0.8660  1.0000   0.8660   0.5000    0  -0.5000  -0.8660
Columns 10 through 13
   -1.0000  -0.8660  -0.5000    0
>> plot(theta,A)
>> xlabel('Theta')
>> ylabel('Amplitude')
```

Figure 1

Note new toolbar buttons: data brushing & linked plots

Dr. Ahmed ElShafee, ACU : Spring 2016, Introduction To Engineering
Dr. Ahmed ElShafee, ACU: Spring 2016, Introduction To Engineering

Command Window

New to MATLAB? Watch this Video, see Demos, or read Getting Started.

```matlab
>> x=0:1:10
x =
     0     1     2     3     4     5     6     7     8     9    10
>> y=2*x + 3
y =
     3     5     7     9    11    13    15    17    19    21    23
>> plot(x,y)
>> xlabel('x')
>> ylabel('y')
>> title('first order equation')
```

Figure 1
Dr. Ahmed ElShafee, ACU: Spring 2016, Introduction To Engineering

Command Window

New to MATLAB? Watch this Video, see Demos, or read Getting Started.

```matlab
>> x=0:0.1:10;
>> y=x.*x;
>> plot(x,y)
>> xlabel('x')
>> title('second order equation')
>> ylabel('y')
```

Figure 1

second order equation

![Graph of a second order equation](image)
Figure 1

Second Order equation

Command Window

New to MATLAB? Watch this Video, see Demos, or read Getting Started.

```matlab
>> x=0:0.1:10;
>> y1=2*x+3;
>> y2=x.*x;
>> plot(x,y1,x,y2);
>> xlabel('x'),gtext('y1'),gtext('y2')
>> title('First Order Vs. Second Order equation');
```
Example 06

Use MATLAB to plot the function \( s = 2 \sin(3t + 2) + \sqrt{5t + 1} \) over the interval \( 0 \leq t \leq 5 \). Put a title on the plot, and properly label the axes. The variable \( s \) represents speed in feet per second; the variable \( t \) represents time in seconds.
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

```matlab
>> t=0:0.1:5;
>> S=2*(sind((2*pi*t)+2)+sqrt((5*pi*t)+1));
>> plot(t,S);
>> xlabel('sec');
>> ylabel('Speed Feet/sec');
```
Example 07

Use MATLAB to plot the functions \( y = 4\sqrt{6x + 1} \) and \( z = 5e^{0.3x} - 2x \) over the interval \( 0 \leq x \leq 1.5 \). Properly label the plot and each curve. The variables \( y \) and \( z \) represent force in newtons; the variable \( x \) represents distance in meters.

Command Window

New to MATLAB? Watch this Video, see Demos, or read Getting Started.

\[
\begin{align*}
\text{>> } x &= 0:0.1:1.5; \\
\text{>> } y &= (4\text{sqrt}(6*x)+1); \\
\text{>> } z &= (5\text{exp}(0.3*x))-(2*x); \\
\text{>> } \text{plot}(x,y,x,z) \\
\text{>> } \text{xlabel('Distance (m)' );} \\
\text{>> } \text{ylabel('Y Force (N)' ),gtext('Z Force (n)' )}
\end{align*}
\]
Thanks,..

See you next week (ISA),...