

# Learning MATLAB: Two-Dimensional Plots & Introduction to Programming

Dr. Waleed Al-Hanafy

waleed\_alhanafy@yahoo.com

Faculty of Electronic Engineering, Menoufia Univ., Egypt

Learning MATLAB & its Applications — Lecture no. 4

August 2, 2011

# Overview

- 1 Two-Dimensional Plots
- 2 Introduction to Programming
  - Control Statements
- 3 Conclusions

## References:

- [1] Desmond J. Higham and Nicholas J. Higham, MATLAB Guide, 2nd ed. Society for Industrial and Applied Mathematics, 2005.
- [2] Amos Gilat, MATLAB An Introduction with Applications. John Wiley & Sons Inc., 2004.

# The Plot Command

`plot(x,y)`

- The plot “command plot” in its simplest shape plots the dependent variable  $y$  as a function of the independent variable  $x$ . Both  $x$  and  $y$  should have the same dimension.
- Another detailed form of the “plot command” is:

`plot (x,y, 'line specifiers', 'PropertyName', PropertyValue)]`

- Line specifiers are optional and can be used to define the style and color of the line and the type of markers (if markers are desired). The line style specifiers are:

# The Plot Command (cont'd)

		Line color	Specifier	Marker Type	Specifier
		red	r	plus sign	+
Line Style	Specifier	green	g	circle	o
solid (default)	-	blue	b	asterisk	*
dashed	- -	cyan	c	point	.
dotted	:	magenta	m	square	s
dash-dot	-.	yellow	y	diamond	d
		black	k	five-pointed star	p
		white	w	six-pointed star	h

# Examples

- `plot(x,y, 'r' )` — A red solid line connects the points
- `plot(x,y, '-y' )` — A yellow dashed line connects the points
- `plot(x,y, '*' )` — The points are marked with \* (no line between the points)
- `plot(x,y,'g:d')` A green dotted line connects the points that are marked with diamond markers

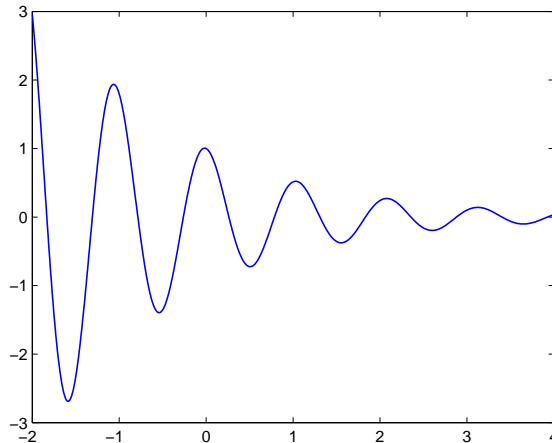
# Plot of a Function

In order to plot a function  $y = f(x)$  with the “plot command”, the user needs to first create a vector of values of  $x$  for the domain that the function will be plotted. Then, a vector  $y$  is created with the corresponding values of  $f(x)$  by using element-by-element calculations. Once the two vectors exist, they can be used in the plot command.

**Example:** The plot command is used to plot the function  $y = 3.5^{-0.5x} \cos(6x)$  for  $-2 \leq x \leq 4$ . A program that plots this function is shown in the following script file.

```
x=[-2:0.01:4];  
y=3.5.^(-0.5*x).*cos(6*x);  
plot(x,y)
```

# The Result



# Formatting a Plot

- The “xlabel” and “ylabel” commands: Labels can be placed next to the axes the xlabel and ylabel commands which have the form:

```
xlabel('text as string')  
ylabel('text as string')
```

- The “title” command: A title can be added to the plot with the command:

```
title('text as string')
```

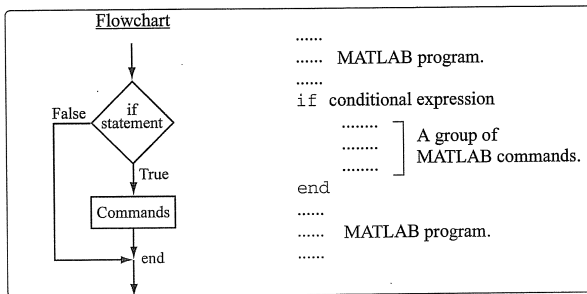
- The “legend” command: Places a legend on the plot:

```
legend( 'string1', 'string2' )
```



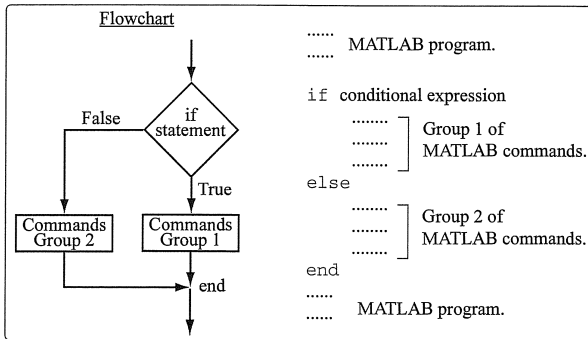
# The IF Statement

## ■ The **if-end** Structure:



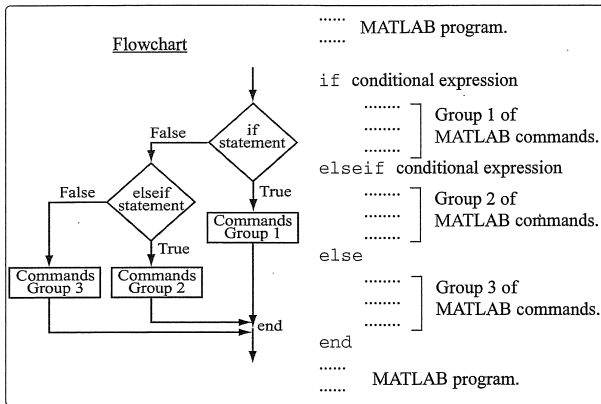
# The IF Statement (cont'd)

## ■ The **if-else-end** Structure:



# The IF Statement (cont'd)

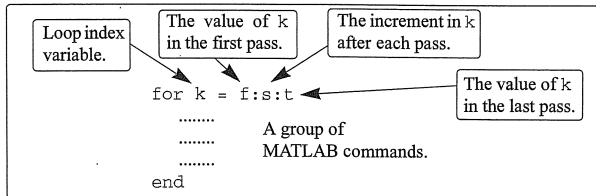
## ■ The **if-elseif-else-end** Structure:



# For Loop

The **for-end** Loops:

In for-end loops the execution of a command, or a group of commands, is repeated a predetermined number of times. The form of the loop is shown below.



# While Loop

The **while-end** loops:

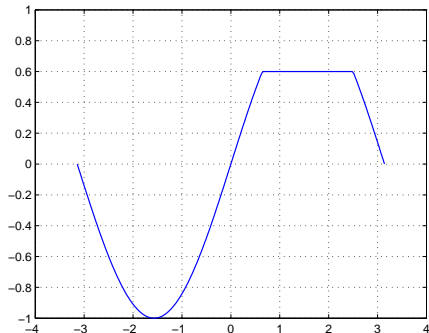
Are used in situations when looping is needed but the number of passes is not known ahead of time. In **while-end** loops the number of passes is not specified when the looping process starts. Instead, the looping process continues until a stated condition is satisfied. The structure of a **while-end** loop is shown below.

```
while conditional expression
    .....
    .....      A group of
    .....      MATLAB commands.
end
```

# Example

Suppose that we want to draw a clipped sine wave  $y = \sin(x)$  such that if  $y \geq 0.6$  the output, i.e.  $y$  will equal to 0.6. We can do this using the following code, and the result will be as shown in the Fig. below:

```
x=-pi:pi/100:pi;  
y=sin(x);  
for i=1:length(y);  
    if y(i)>=.6  
        y(i)=.6;  
    end  
end  
plot(x,y);  
axis([-4 4 -1 1])  
grid on
```



# Conclusions

## Concluding remarks

- Two-Dimensional Plots have been given
- An Introduction to MATLAB programming is considered