Lecture Two
Multimedia Systems

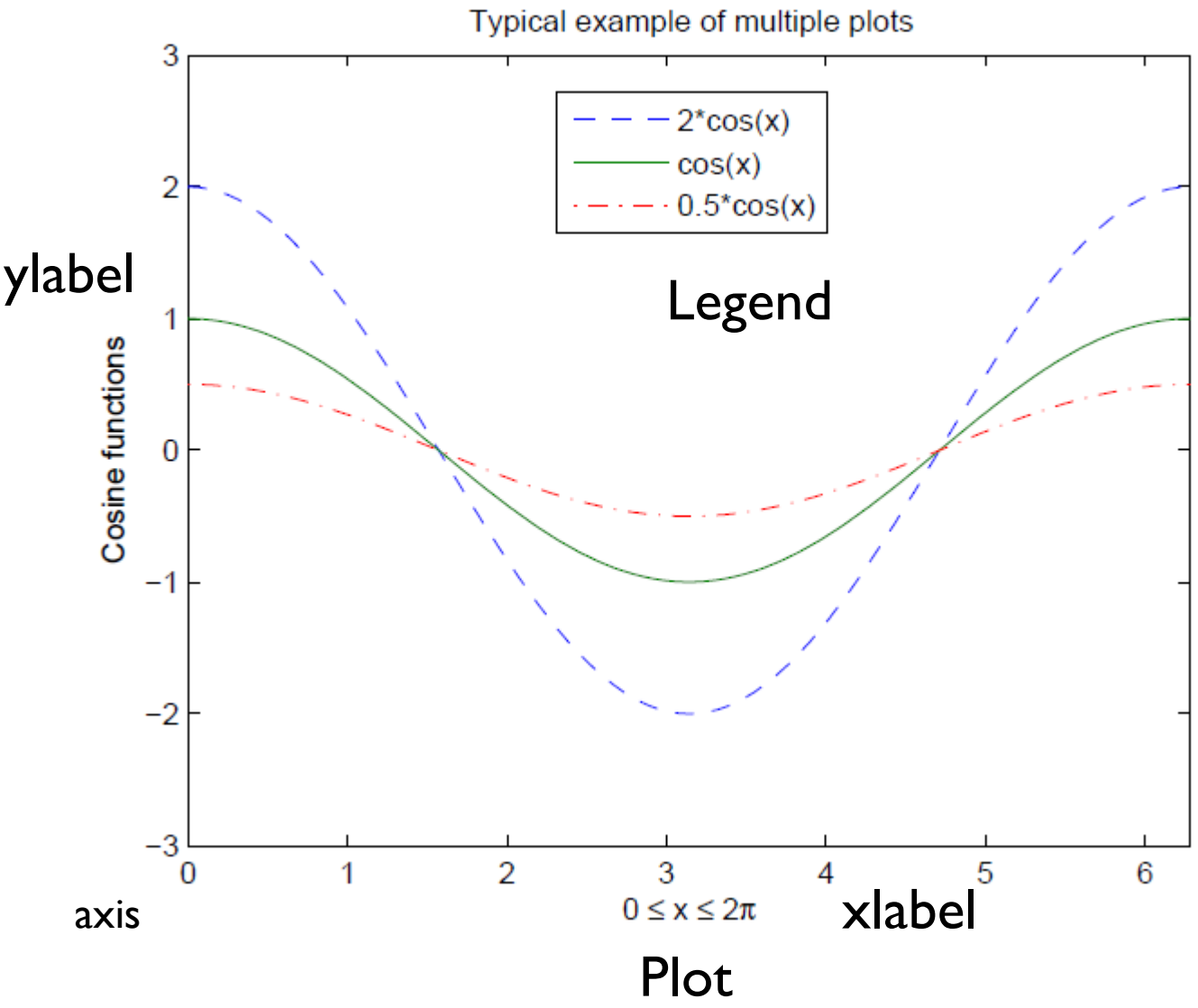
Introduction to MATLAB

Part two

By: Eng. Haydara Mahmoud.

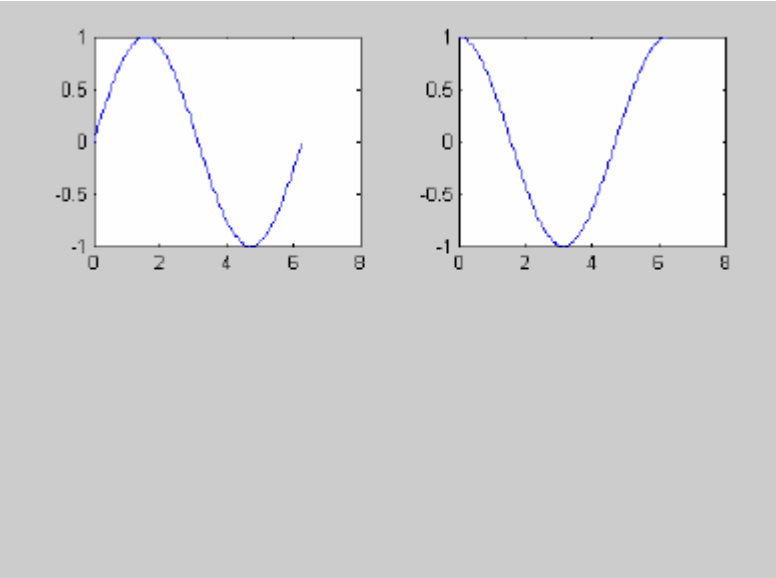
Damascus University – Faculty of Informatics - 2019

Review:


Title
Typical example of multiple plots
ylabel
Legend
2*cos(x)
cos(x)
0.5*cos(x)
Cosine functions
3
2
1
0
−1
−2
−3
0
1
2
3
4
5
6
axis
0 ≤ x ≤ 2π
xlabel
Plot

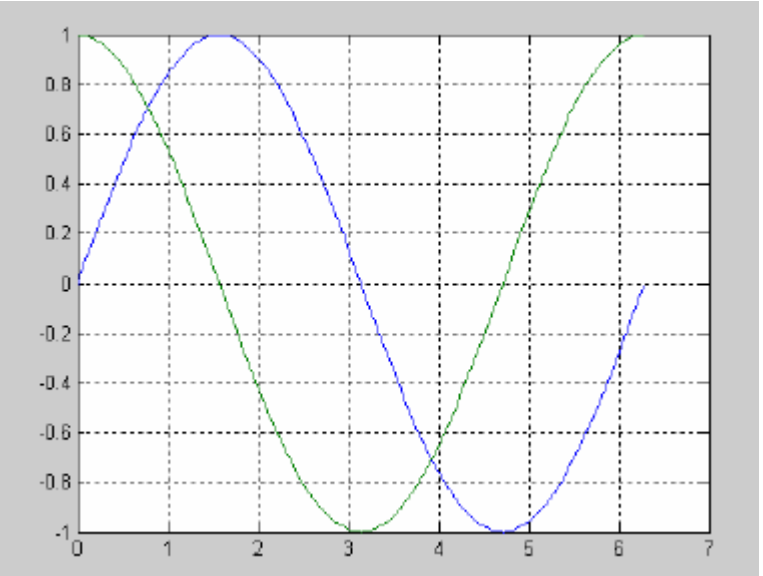
Subplot

```
t = 0:pi/100:2*pi;
y1=sin(t);
y2=sin(t+pi/2);
subplot(2,2,1)
plot(t,y1)
subplot(2,2,2)
plot(t,y2)
```



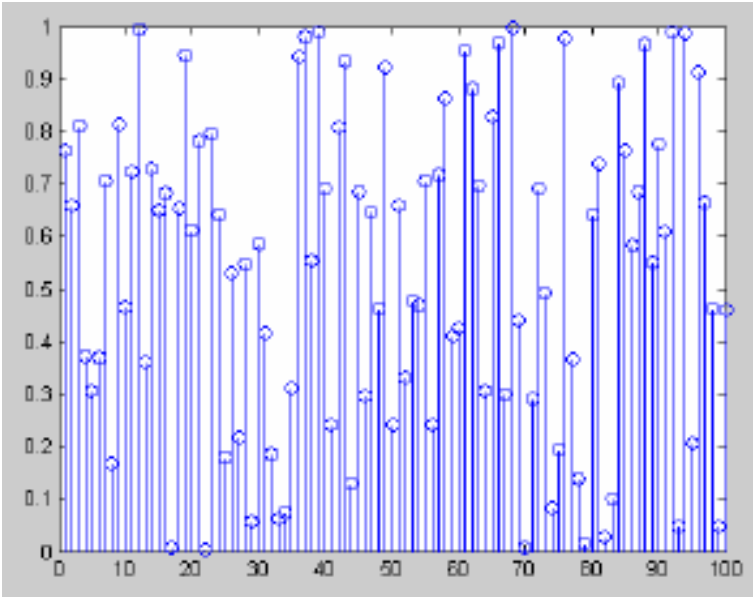
Another way to plot multiple graphs:

```
t = 0:pi/100:2*pi;
y1=sin(t);
y2=sin(t+pi/2);
plot(t,y1,t,y2)
grid on
```

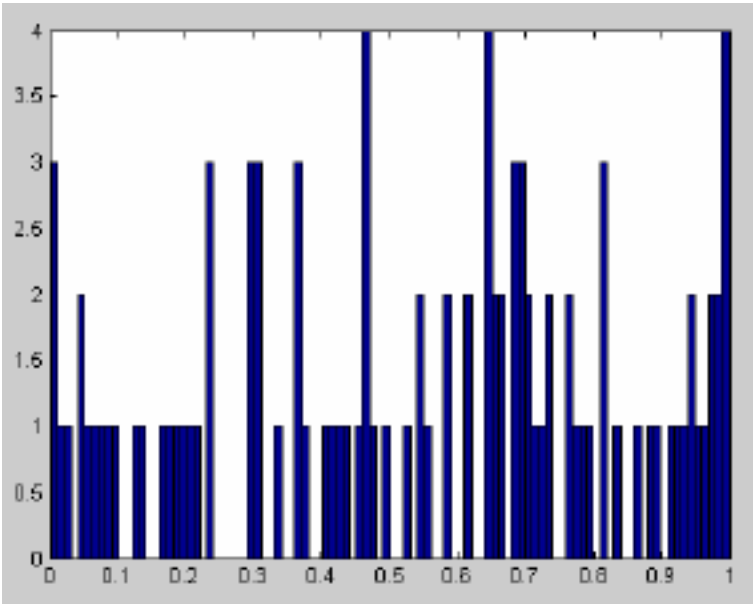


Stem & hist:

```
x=rand(100,1);
stem(x);
```




```
hist(x,100)
```



M-Files:

M-files can be *scripts* that simply execute a series of MATLAB statements, or they can be *functions* that can accept arguments and can produce one or more outputs.

to create a file: **File** → **New** → **M-file.**

Functions:

(File → New → Function)

```
function f=myfunction(x,y)
   f=x+y;
```

- save it in myfunction.m
- call it with `y=myfunction(x,y)`

```
function [outputs] = function_name(inputs)
```

Functions – Cont.

`function C=FtoC(F)`	One input argument and one output argument
`function area=TrapArea(a,b,h)`	Three inputs and one output
`function [h,d]=motion(v,angle)`	Two inputs and two outputs

Example of input and output arguments

Input commands:

```
game1 = input('Enter the points scored in the first game ');
game2 = input('Enter the points scored in the second game ');
game3 = input('Enter the points scored in the third game ');
average = (game1+game2+game3)/3
```

Output commands:

Two commands that are frequently used to generate output are: `disp` and `fprintf`.

`disp`
- Simple to use.
- Provide limited control over the appearance of output

`fprintf`
- Slightly more complicated than `disp`.
- Provide total control over the appearance of output

Exercise:

Write a function file that converts temperature in degrees Fahrenheit (°F) to degrees Centigrade (°C). Use `input` and `fprintf` commands to display a mix of text and numbers.

conversion formulation, $C = 5/9 * (F - 32)$.

Flow Control (if .. end)

```
if A > B
   'greater'
elseif A < B
   'less'
else
   'equal'
end
```

MATLAB supports the variants of "`if`" construct.

- `if ... end`
- `if ... else ... end`
- `if ... elseif ... else ... end`

It should be noted that:

- `elseif` has no space between `else` and `if` (one word)
- no semicolon (`;`) is needed at the end of lines containing `if`, `else`, `end`
- indentation of `if` block is not required, but facilitate the reading.
- the `end` statement is required

Flow Control (for .. end)

```
for variable = expression
    statements
end
```

```
for x = 1:10
   r(x) = x;
end
```

Flow Control (while .. end)

```
while expression
      statements
end
```

The `statements` are executed as long as `expression` is true.

```
x = 1
while x <= 10
      x = 3*x
end
```

It is important to note that if the condition inside the looping is not well defined, the looping will continue *indefinitely*. If this happens, we can stop the execution by pressing **Ctrl-C**.

Break .. Continue

- The `break` statement. A `while` loop can be terminated with the `break` statement, which passes control to the first statement after the corresponding `end`. The `break` statement can also be used to exit a `for` loop.

- The `continue` statement can also be used to exit a `for` loop to pass immediately to the next iteration of the loop, skipping the remaining statements in the loop.

Save output to a file (.txt):

1. Open a file using `fopen`
2. Write the output using `fprintf`
3. Close the file using `fclose`

```
% write some variable length strings to a file
op = fopen('weekdays.txt','wt');
fprintf(op,'Sunday\nMonday\nTuesday\nWednesday\n');
fprintf(op,'Thursday\nFriday\nSaturday\n');
fclose(op);
```

Examples:

- Simulate the outcomes of 100 fair coin tosses

```
x=rand(100,1);
p=sum(x<0.5)/100

p =
    0.5400
```

- Simulate the outcomes of 1000 fair coin tosses

```
x=rand(1000,1);
p=sum(x<0.5)/1000

p =
    0.5110
```

- Simulate the outcomes of 1000 biased coin tosses with p[Head]=0.4

```
x=rand(1000,1);
p=sum(x<0.4)/1000


p =

    0.4160
```

Exercise:

- Write a **function** in MATLAB (mid.m) to calculate the average of (n) entered numbers?
- Write a **script** which includes:
  - Calling the function (mid.m)?
  - Saving the output to a file (result.txt)?

    - ***Hint:*** use the function (num2str) to convert the number into string.