LEBANESE AMERICAN UNIVERSITY
School of Engineering
Department of Electrical and Computer Engineering

ELE443 Control System LAB
Fall 2013

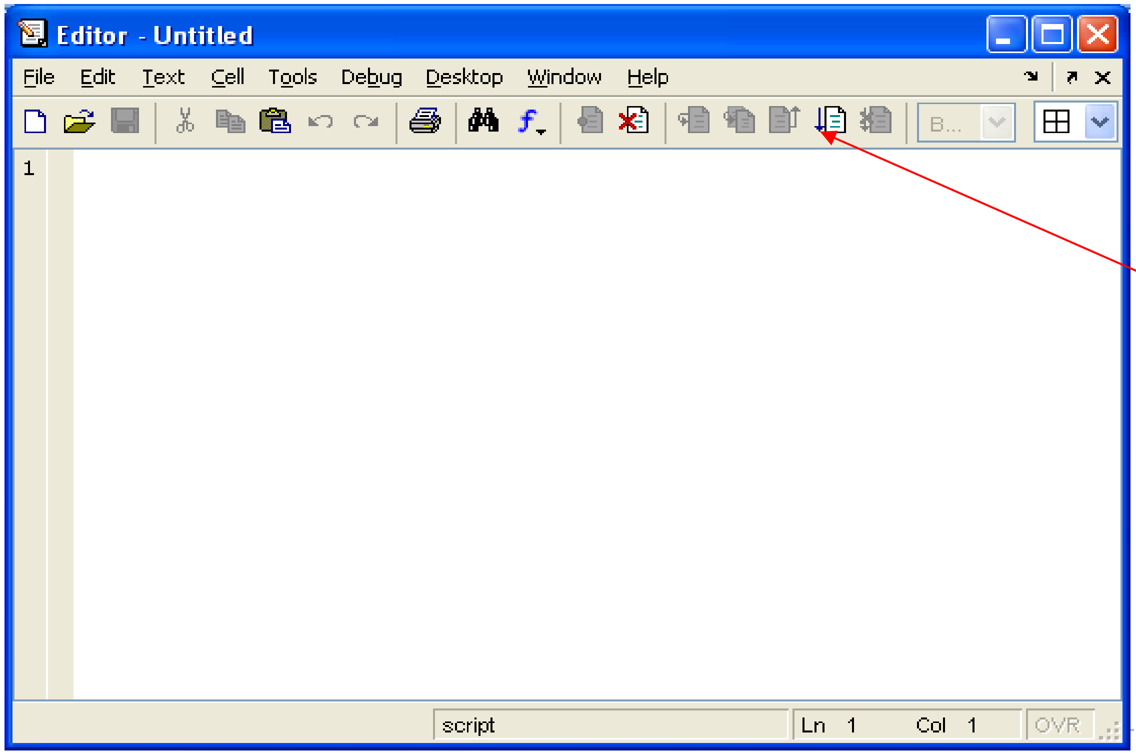
Lecture 3: Function files & Graphs

Introduction

- Script files are used to write programs, to save and to run them using MATLAB commands.
- Script files contain list of MATLAB commands.
- Script are saved in "filename.m"
- Select File menu in the MATLAB toolbar:
  - File>New>M-file
- The Editor window will open.

Creating Script Files

- Commands in M-files are executed in the order they are listed.
- Script files can be edited and executed many times.
- The program is automatically saved when Run button is pressed.



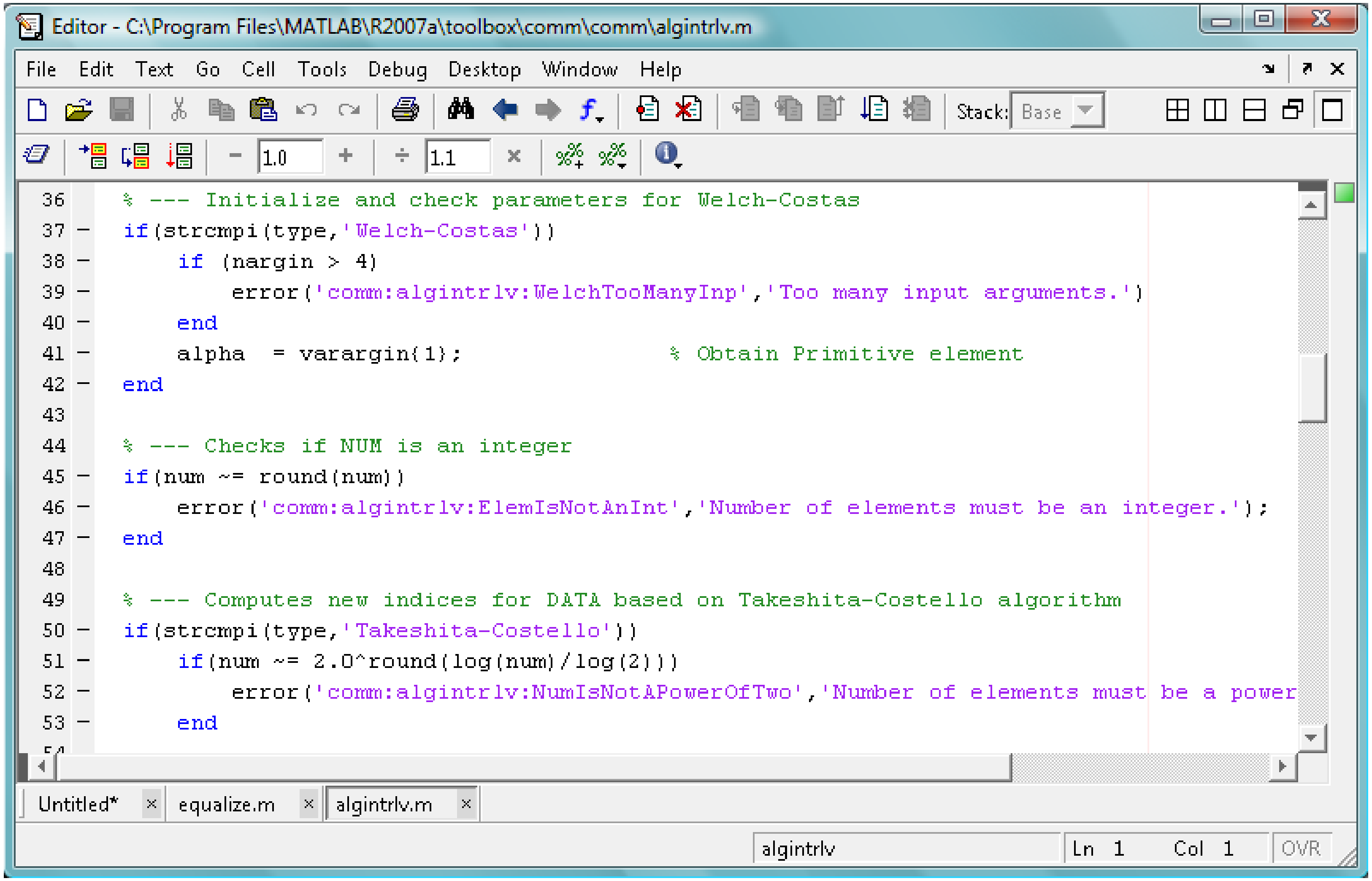
Run command

Breakpoints

- Breakpoints can be placed in script files for debugging.
- A breakpoint can be set for every line.
  - Place the cursor on the desired line.
  - Press F12 or choose Debug>Set/Clear Breakpoint from Editor toolbar.

- Program stops and Workspace variables are updated once the program reaches a breakpoint
- Press Continue button to resume the program

Example of M-file


Editor - C:\Program Files\MATLAB\R2007a\toolbox\comm\comm\algintrlv.m
File Edit Text Go Cell Tools Debug Desktop Window Help
Stack: Base
1.0
1.1
Untitled*
equalize.m
algintrlv.m
algintrlv
Ln 1 Col 1
OVR

```
% --- Initialize and check parameters for Welch-Costas
if(strcmpi(type,'Welch-Costas'))
    if (nargin > 4)
        error('comm:algintrlv:WelchTooManyInp','Too many input arguments.')
    end
    alpha  = varargin{1};              % Obtain Primitive element
end

% --- Checks if NUM is an integer
if(num ~= round(num))
    error('comm:algintrlv:ElemIsNotAnInt','Number of elements must be an integer.');
end

% --- Computes new indices for DATA based on Takeshita-Costello algorithm
if(strcmpi(type,'Takeshita-Costello'))
    if(num ~= 2.0^round(log(num)/log(2)))
        error('comm:algintrlv:NumIsNotAPowerOfTwo','Number of elements must be a power
    end
```

Function files

- MATLAB has its own set of built-in and toolbox functions
  - Example: sin, exp, rand, plot…
- User can write his own function and executes it.
- Function files are written using the Editor Window.
- A function has a name, can have arguments and output arguments.
- The first line in Function files has the following format:
  - function [out_1,…,out_m] =function-name(arg_1,…,arg_n)
- [out_1,…,out_m] is an array that returns m variables
- (arg_1,…,arg_n) is a list of n arguments that the function takes.

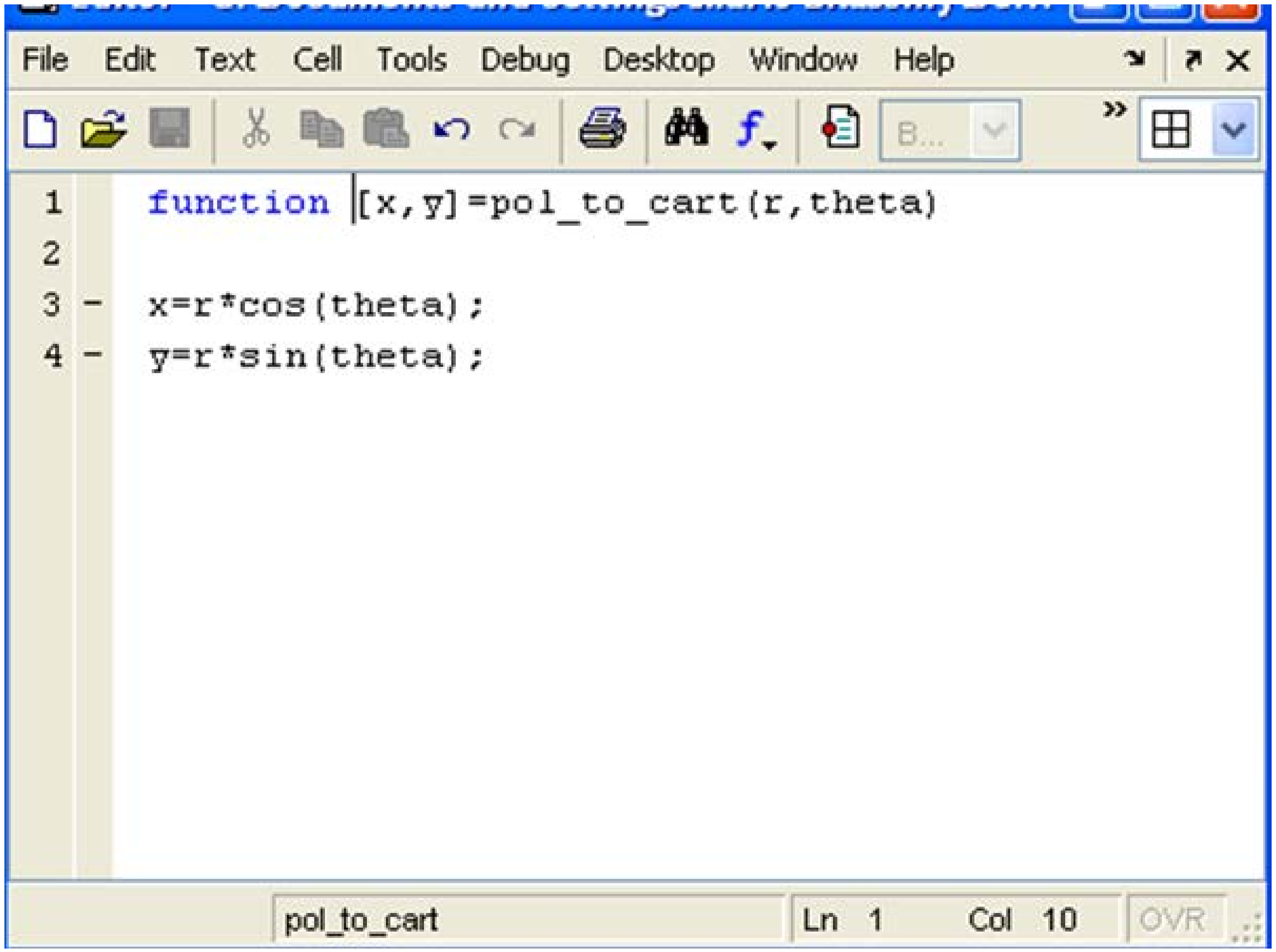
Function Files

- function[out_1,…,out_m]=function-name(arg_1,…,arg_n)
- Function-name
  - It's the function name.
  - The M-file should have the same name as function-name
- M-files that contain functions are saved as
  - Function-name.m
- In order to call a user defined function, function-name.m should exist in the Current Directory.

Example of function files

- Let's define a function that transforms the polar coordinates into their corresponding Cartesian coordinates.
- Let's name the function pol_to_cart.
- This function has 2 input arguments:
  - **r** and **theta**
- This function has 2 output arguments:
  - **x** and **y**

Example of function files

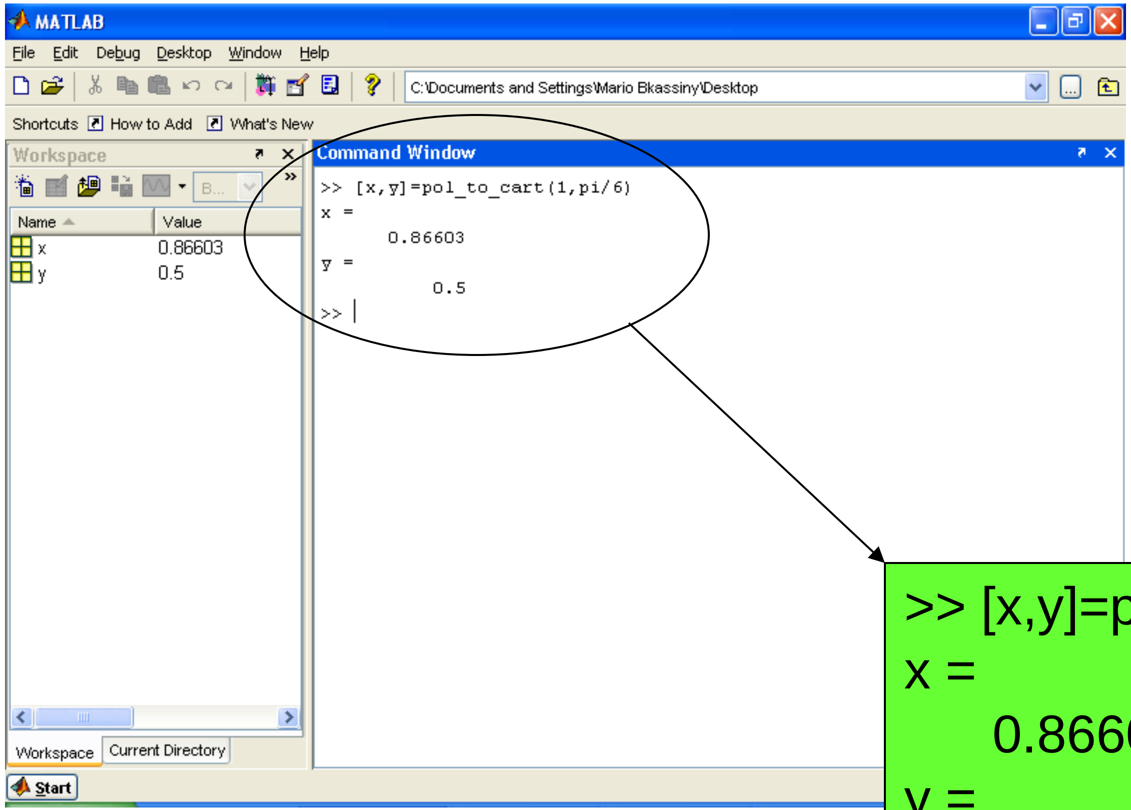


```
function [x,y]=pol_to_cart(r,theta)

x=r*cos(theta);
y=r*sin(theta);
```

This function is saved as pol_to_cart.m file

Calling User defined functions


MATLAB
File Edit Debug Desktop Window Help
C:\Documents and Settings\Mario Bkassiny\Desktop
Shortcuts How to Add What's New
Workspace
Command Window
Start
Workspace Current Directory

Name	Value
x	0.86603
y	0.5

```
>> [x,y]=pol_to_cart(1,pi/6)
x =
      0.86603
y =
          0.5
>>
```

```
>> [x,y]=pol_to_cart(1,pi/6)
x =
      0.86603
y =
          0.5
```

User defined functions can run in Command Window, or in Script Files or in other Function Files.

Function Overloading

- The behavior of a function can be modified depending on the number of input and output arguments.
- Commands:
  - nargin: Number of Input arguments
  - nargout: Number of Output arguments
- Example:

```
function [P]=integral(t,x,flag)
          if nargin==2
          dt=t(2)-t(1);
          P=sum(x)*dt;
    elseif nargin==3
          dt=t(2)-t(1);
          P=cumsum(x).*dt;
end
```

Saving Data

- MATLAB uses its own platform independent file format for saving data files.
  - Files have a ".mat" extension
  - **save** is used to save variables from the workspace to a named file (by default: matlab.mat if no filename is given)
    - **save filename** – saves entire workspace to *filename.mat*
    - **save var1 var2 … filename** – saves named variables to *filename.mat*
  - By default **save** overwrites an existing file of the same name, use **–append** to append data to an existing file
    - Variables of the same name are always overwritten!!!
    - **save var1 var2 … filename -append**
- Data is recovered using **load** command
  - **load filename** – loads entire *.mat* file
  - **load filename var1 var2 …** - loads named vars

Prompting for User input

- The **input** function can be used to prompt the user for numeric or string input.

```
>>x = input('Enter a value for x');
>>YourName = input('Enter your name','s');
```

MatLab Programming

- **Program Control Statements:**
  - Conditional Control (if, switch)
  - Loop Control (for, while, continue, break)
  - Error Control (try, catch)
  - Program Termination (return)

Conditional Control (if statement)

```
clear;
x=-2;
y=10;
if(x<0)
    angle=180+atand(y/x)
else
    angle=atand(y/x)
end
```

Loop Control (For loop)

- Loop Expression Format:

```
for index = start:increment:end
    statements
end
```

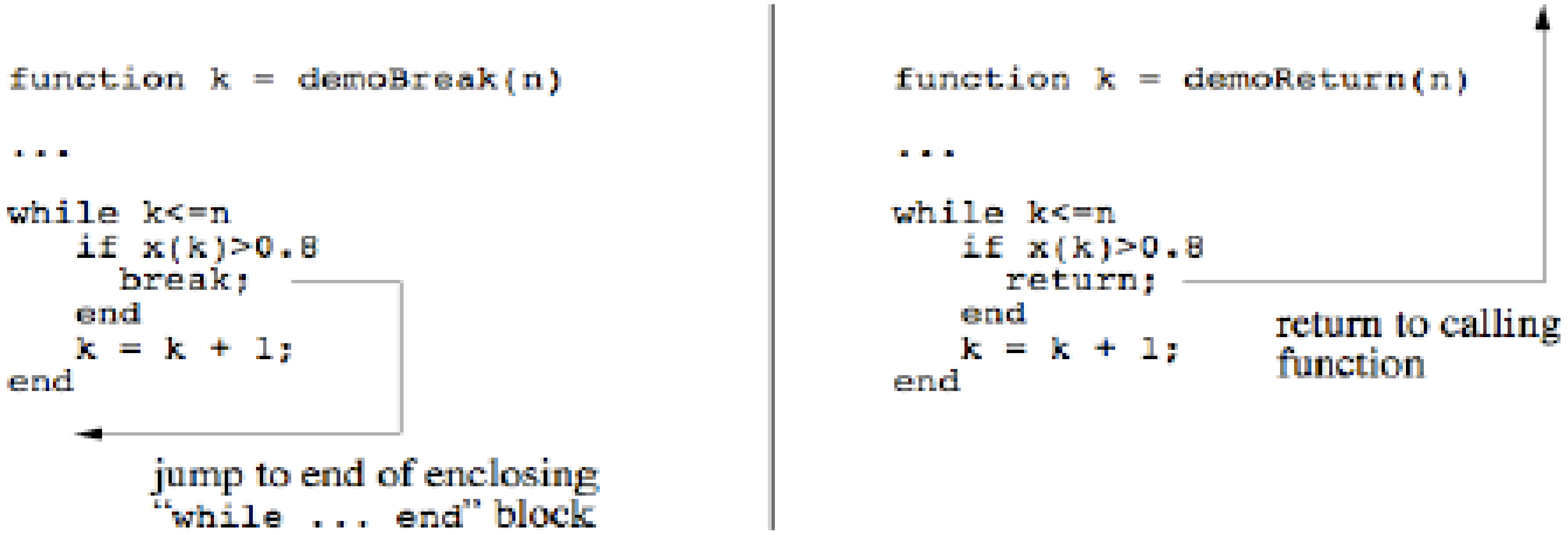
- Example:

```
for n = 2:size(x,2)
    x(n) = 2 * x(n - 1);
end
```

- The command `break` exits a loop

Comparison of break and return

- **break** is used to escape the current loop.
- **return** is used to escape the current function.



```
function k = demoBreak(n)

...

while k<=n
    if x(k)>0.8
       break;
    end
    k = k + 1;
end
```

```
function k = demoReturn(n)

...

while k<=n
    if x(k)>0.8
       return;
    end
    k = k + 1;
end
```

jump to end of enclosing "while ... end" block
return to calling function

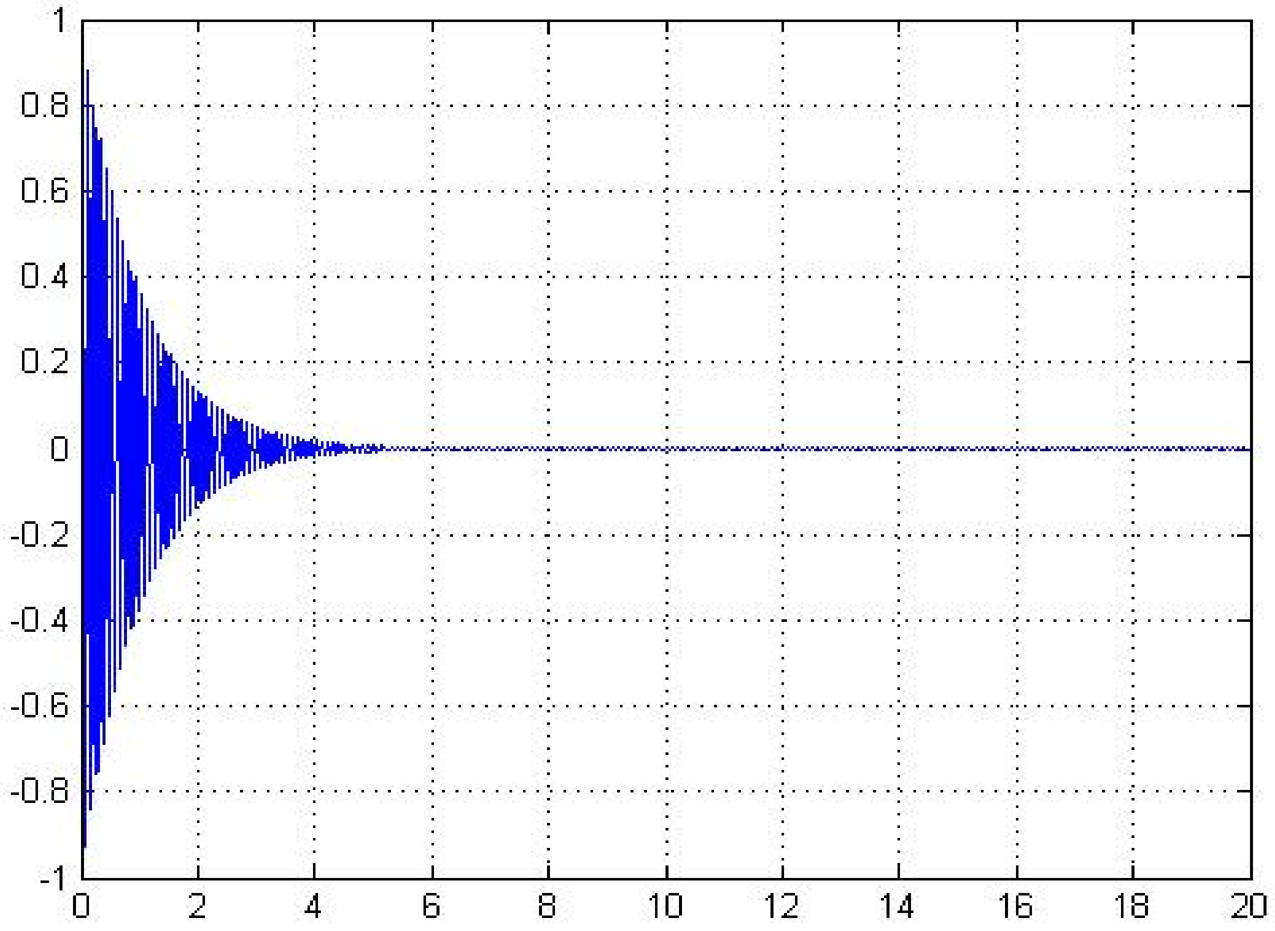
- Remark: an *infinite loop* can be broken by typing "Ctrl+C"

Plotting

- MATLAB is used extensively to plot graphs.
- Different parameters can be modified in MATLAB figures:
  - Number of subplots in a figure
  - Scale (i.e. Linear or Logarithmic)
  - Grid, colors, labels and legends.

- Consider the plotting example:
  - t=0:0.0001:3;
  - f=exp(-t).*sin(2*pi*10*t);
  - plot(t,f), grid
- Note: t and f have the same size

Plot command

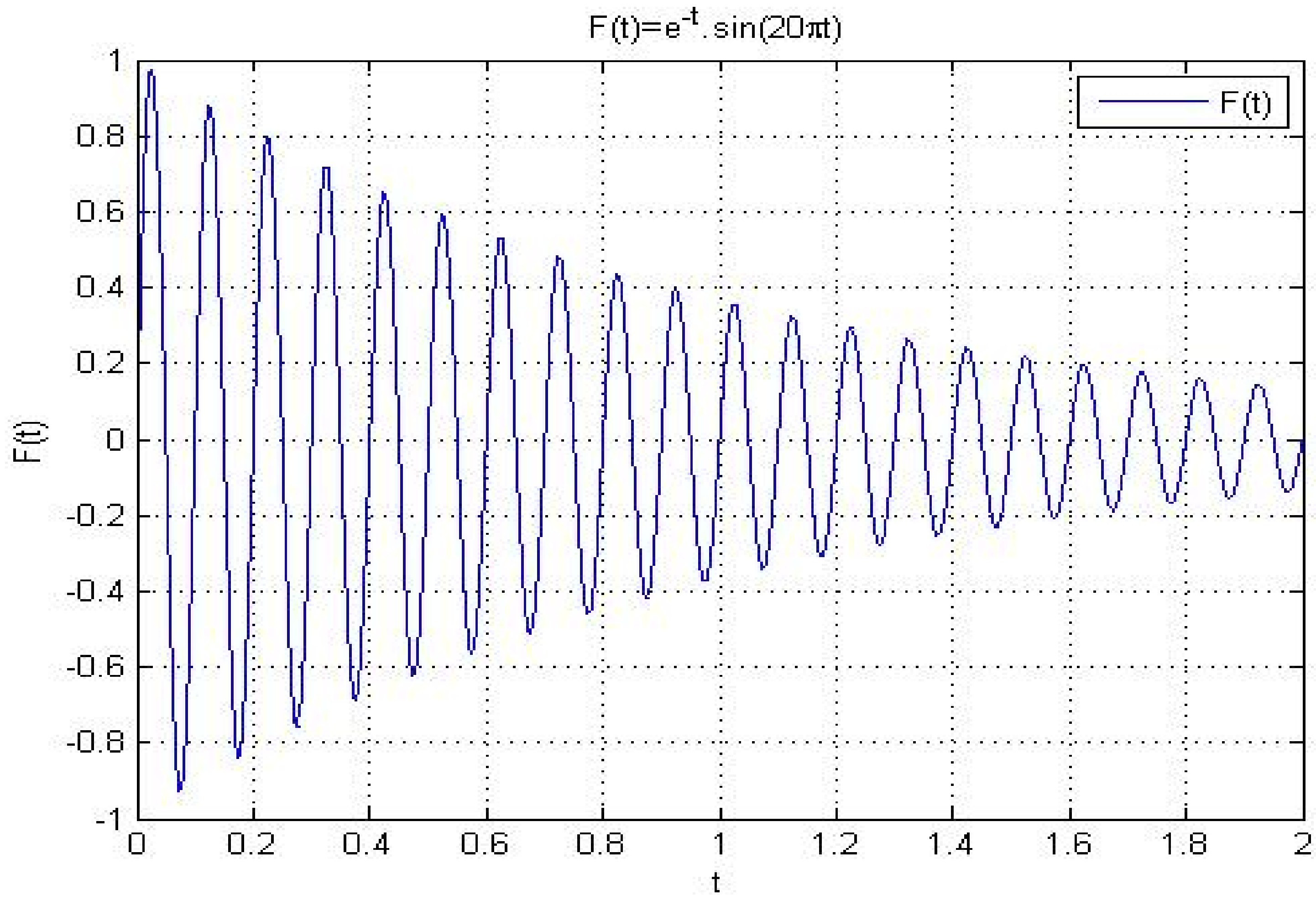


Plotting a function

- The Figure can be edited and labeled such that:

```
ylabel('F(t)')
xlabel('t')
title('F(t)=e^{-t}.sin(20\pit)')
axis([0,2,-1,1])  % axis([xmin, xmax, ymin, ymax])
legend('F(t)')
```

Plotting a function



Plotting 2 curves on the same graph

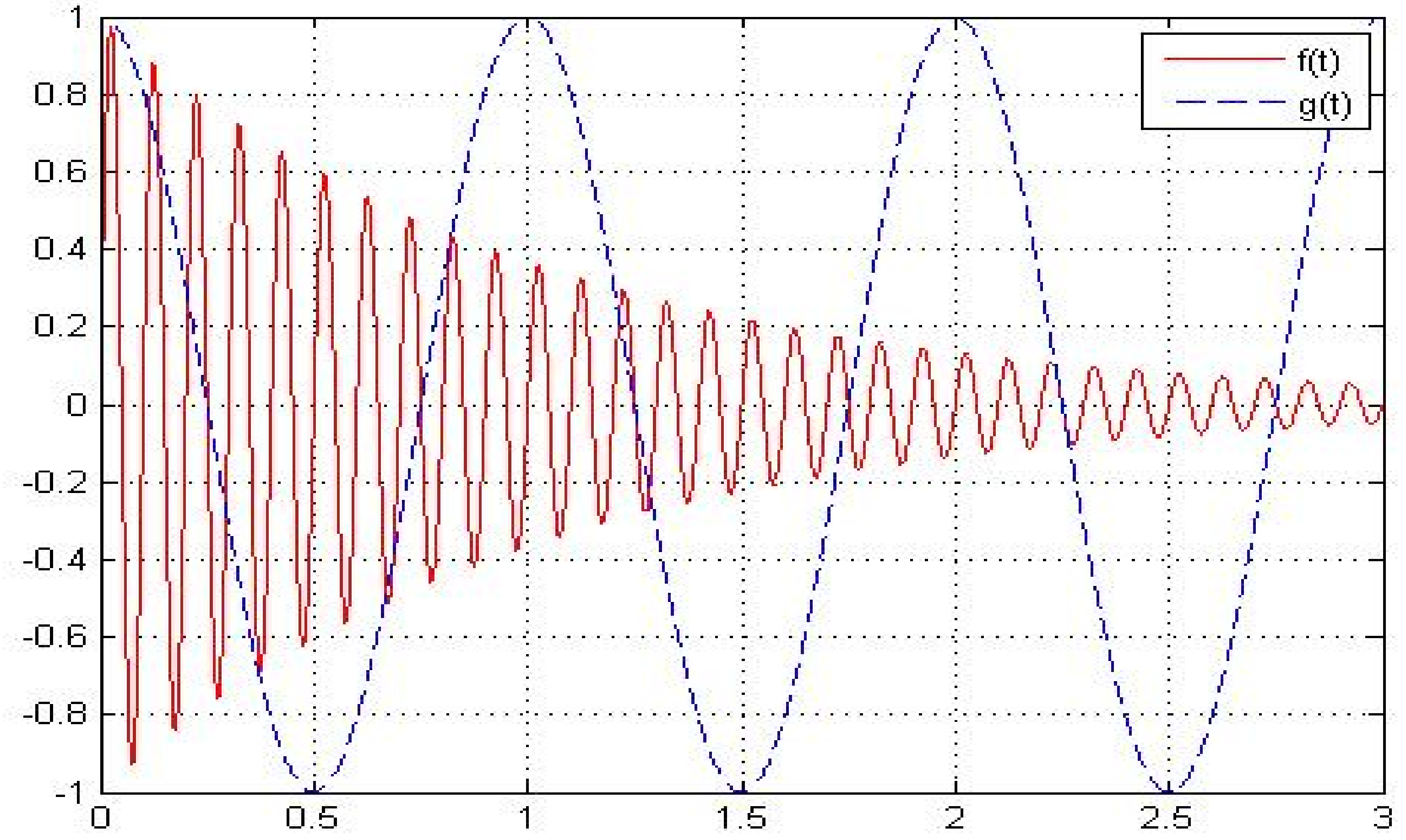
```
t=0:0.0001:3;
f=exp(-
t).*sin(2*pi*10*t);
g=cos(2*pi*t);
plot(t,f,'r',t,g,'b--')
legend('f(t)','g(t)')
```

2 ways

```
t=0:0.0001:3;
f=exp(-
t).*sin(2*pi*10*t);
g=cos(2*pi*t);
plot(t,f,'r,)
hold
plot(t,g,'b--')
legend('f(t)','g(t)')
```

- Use the help command to get more information on the plot command.

Plotting 2 curves on the same graph

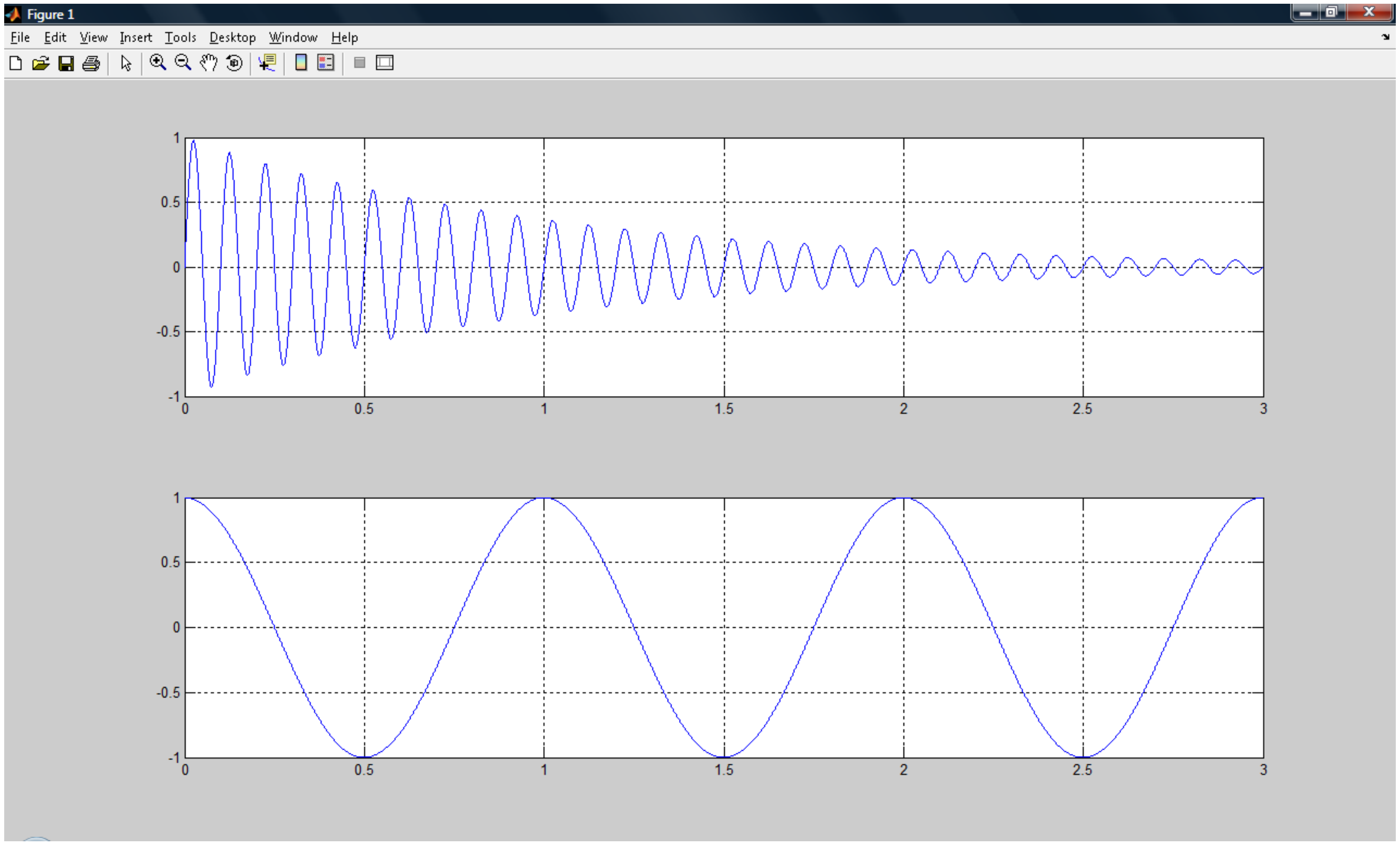


Plotting 2 curves in 1 graph window

```
t=0:1e-4:3;
f=exp(-t).*sin(2*pi*10*t);
g=cos(2*pi*t);
subplot(2,1,1),plot(t,f),grid
subplot(2,1,2),plot(t,g),grid
```

- subplot(M,N,n) creates an array of M-by-N graphs in a figure window, where n is the number of a selected graph in the array.

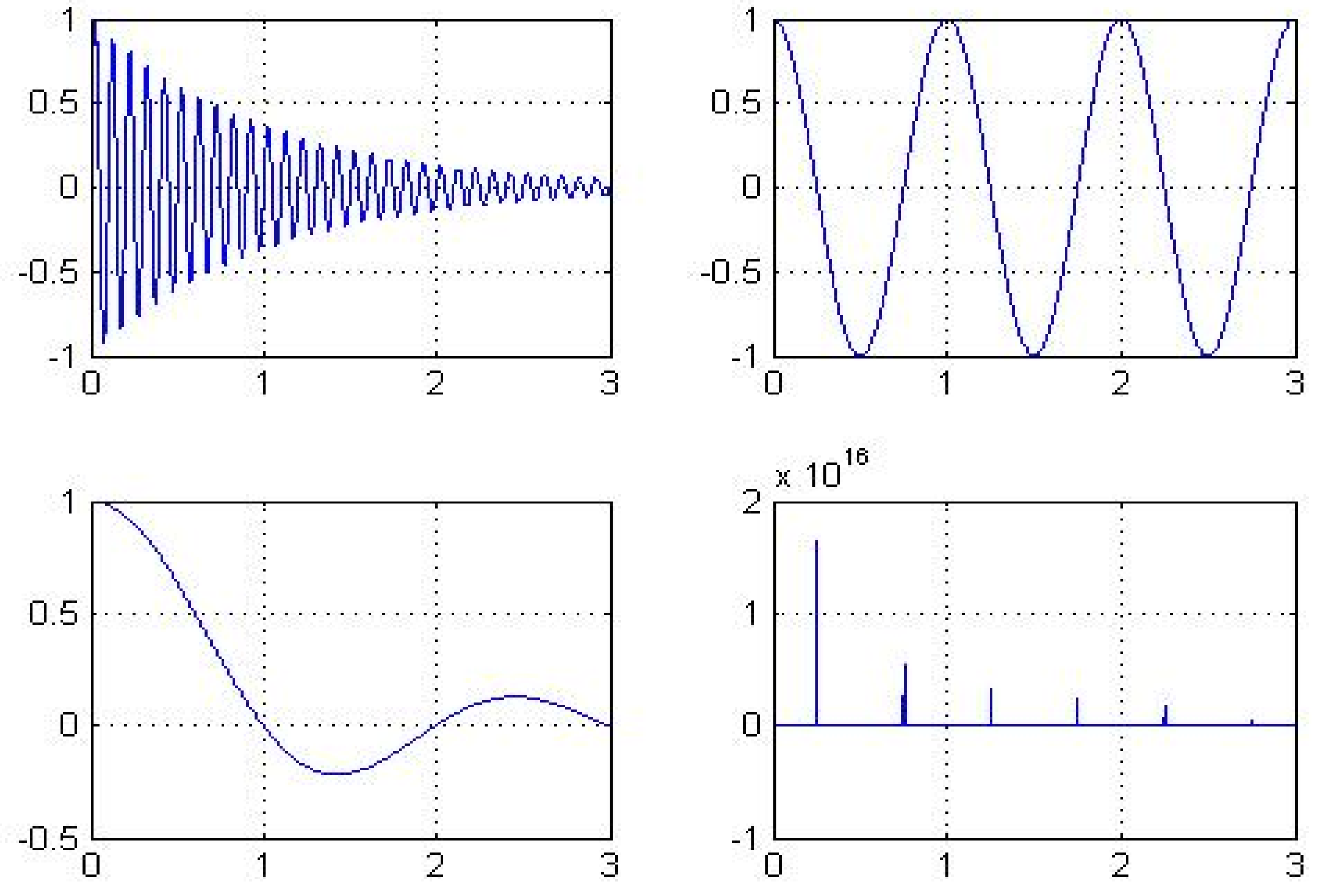
Plotting 2 curves in 1 graph window



Plotting 4 curves in 1 graph window

```
t=0:1e-4:3;
f= exp(-t).*sin(2*pi*10*t);
g= cos(2*pi*t);
y= tan(2*pi*t);
w= sinc(t);
subplot(2,2,1),plot(t, f),grid
subplot(2,2,2),plot(t, g),grid
subplot(2,2,3),plot(t, w),grid
subplot(2,2,4),plot(t, y),grid
```

Plotting 4 curves in 1 graph



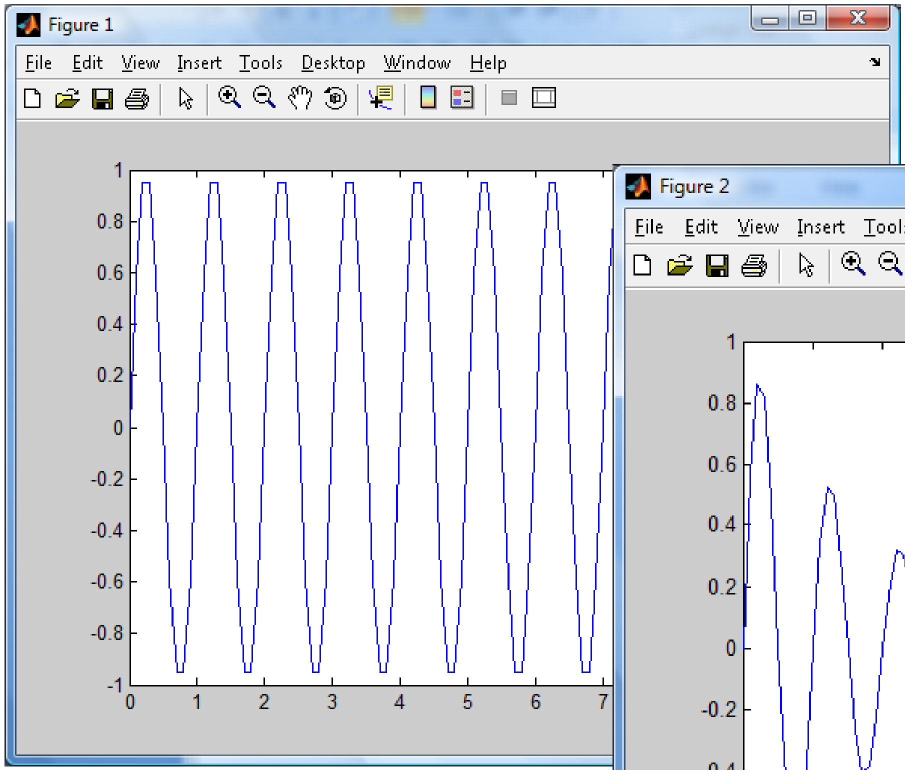
Plotting curves in different graph windows

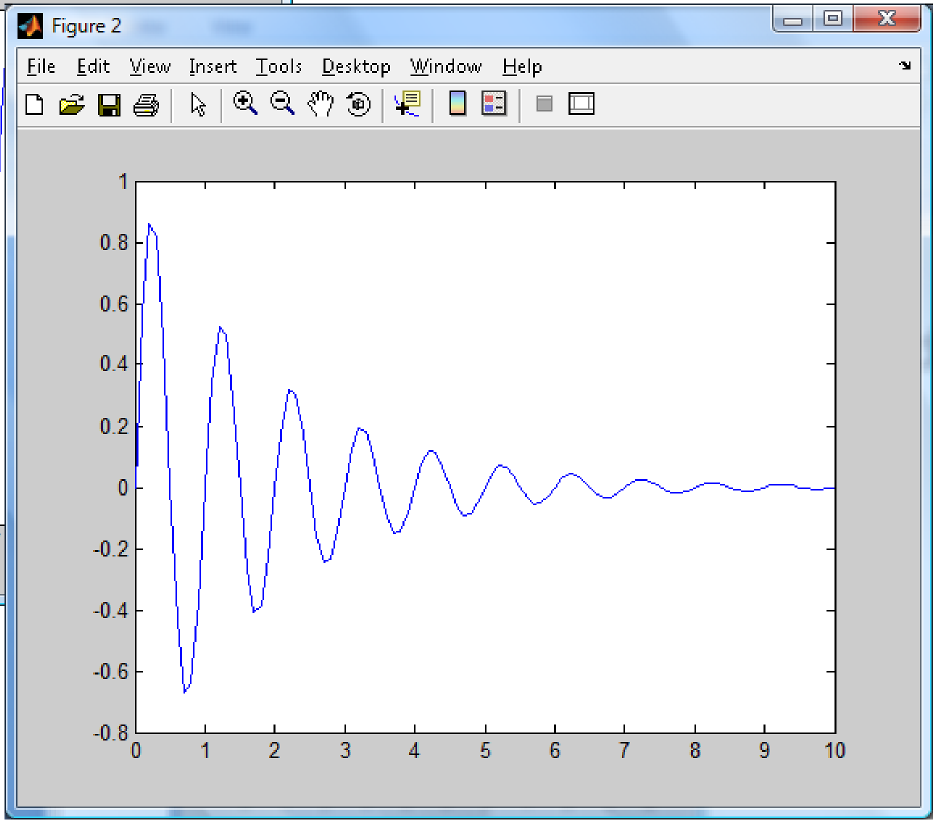
- To open a new window and to plot a curve, we use the command: figure
- Example:

```
t=0:0.1:10;
y1=sin(2*pi*t);
y2=exp(-0.5*t)*sin(2*pi*t);
plot(t,y1)
figure,
plot(t,y2)
```

- In this way, the first graph stays in the first window, and a new window opens and displays the second graph.

Plotting curves in different windows





Multidimensional Functions

- Consider the function f(x, y):

$$f(x,y) = xe^{-x^2-y^2}$$

- f(x,y) is defined over the range:
  - $-2<x<2$
  - $-3<y<3$
- To define the domain of definition of this multidimensional function, we use the command:
  - ndgrid

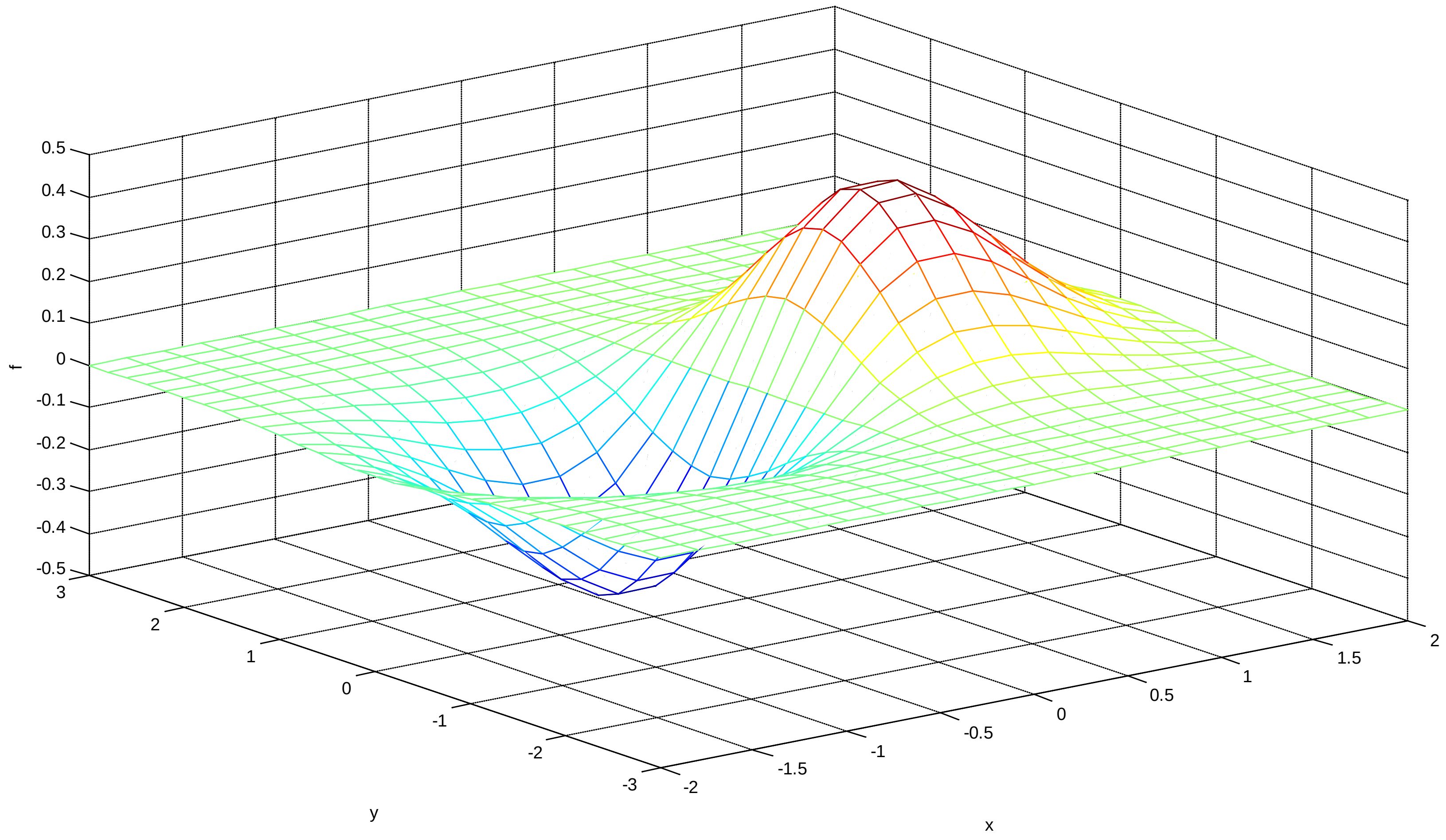
Multidimensional Functions

- Plotting F(x,y) is done by:

```
x=-2:0.2:2;
y=-3:0.2:3;
[X,Y]=ndgrid(x,y);
f=X.*exp(-X.^2-Y.^2);
mesh(X,Y,f)
```

- Use the help command to further understand **ndgrid** and **mesh** commands.
- It plots in a 3-Dimensional Space the points whose coordinates are in matrices X,Y and f.

Multidimensional Functions



Looking Forward

- This is the last lecture concerning the basics of MATLAB.
- You are expected to know and understand these three lectures very well.

- The upcoming three lectures will cover:
  - Linear Systems.
  - Control System Design
  - Simulink and Filters

Control Systems Design and Analysis Under MATALB

Using Simulink to simulate and observe the behavior of controlled systems
+
Simulating Butterworth filters