



# Ψηφιακή Επεξεργασία Σήματος

---

Εργαστήριο 2  
Matlab

Αλέξανδρος Μανουσάκης

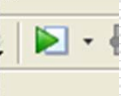
# Format output

---

|                                |                                    |
|--------------------------------|------------------------------------|
| > <b><i>format short</i></b>   | fixed point, 5 digits              |
| > <b><i>format long</i></b>    | fixed point, 15 digits             |
| > <b><i>format short e</i></b> | scientific notation, 5 digits      |
| > <b><i>format long e</i></b>  | scientific notation, 15 digits     |
| > <b><i>format short g</i></b> | fixed or floating-point, 5 digits  |
| > <b><i>format long g</i></b>  | fixed or floating-point, 15 digits |
| > <b><i>format hex</i></b>     | hexadecimal format                 |
| > <b><i>format '+'</i></b>     | +, -, and blank                    |
| > <b><i>format bank</i></b>    | dollars and cents                  |

# Δημιουργία αρχείων .m

---

- Οι ακολουθίες εντολών του MATLAB μπορούν να γραφούν σε αρχεία των οποίων οι ονομασίες θα έχουν κατάληξη **m**, και θα ονομάζονται κατ' αναλογία **αρχεία-m**.
  - Ένα αρχείο-**m** δημιουργείτε από το μενού *File* □ *New* □ *M-file*
  - Η εκτέλεση του αρχείου-**m** μπορεί να γίνει είτε με τη χρήση του εικονιδίου  "**save and run**" που βρίσκεται πάνω στο toolbar του editor του Matlab, είτε πληκτρολογώντας το όνομα ενός τέτοιου αρχείου, χωρίς το **.m** στο command window.
-

# Δημιουργία αρχείων .m

```
%DHMIOYRGIA ARXEIOY m
```

```
clear
```

```
clc
```

```
A=[1:5;6:10;11:15]
```

```
B=A.^2
```

```
C=2*A
```

Command Window

 New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#).

A =

|    |    |    |    |    |
|----|----|----|----|----|
| 1  | 2  | 3  | 4  | 5  |
| 6  | 7  | 8  | 9  | 10 |
| 11 | 12 | 13 | 14 | 15 |

B =

|     |     |     |     |     |
|-----|-----|-----|-----|-----|
| 1   | 4   | 9   | 16  | 25  |
| 36  | 49  | 64  | 81  | 100 |
| 121 | 144 | 169 | 196 | 225 |

C =

|    |    |    |    |    |
|----|----|----|----|----|
| 2  | 4  | 6  | 8  | 10 |
| 12 | 14 | 16 | 18 | 20 |
| 22 | 24 | 26 | 28 | 30 |

>>

# Το filename δεν πρέπει...

---

- Να περιέχει οτιδήποτε εκτός από λατινικούς χαρακτήρες, αριθμούς και \_ (underscore)
- Να αρχίζει με αριθμό

Ορθά filename: ask1.m, ask\_1.m

Λανθασμένα filename: 1ask.m, ask-1.m,  
5.m, a!.m

---

# Δημιουργία γραφικών παραστάσεων

---

Για να απεικονίσουμε γραφικά τα αποτελέσματα μιας δισδιάστατης συνάρτησης χρησιμοποιούμε συναρτήσεις της βιβλιοθήκης `graph2d`.

- > Η πιο βασική εντολή της βιβλιοθήκης είναι η ***plot***.
  - > ***plot(x,y,option)*** εμφανίζεται η γραφική παράσταση του  $y$  ως προς το  $x$ , το τρίτο όρισμα της εντολής χρησιμοποιείται για να καθορίσουμε το χρώμα και το σύμβολο σχεδίασης της γραφικής παράστασης
-

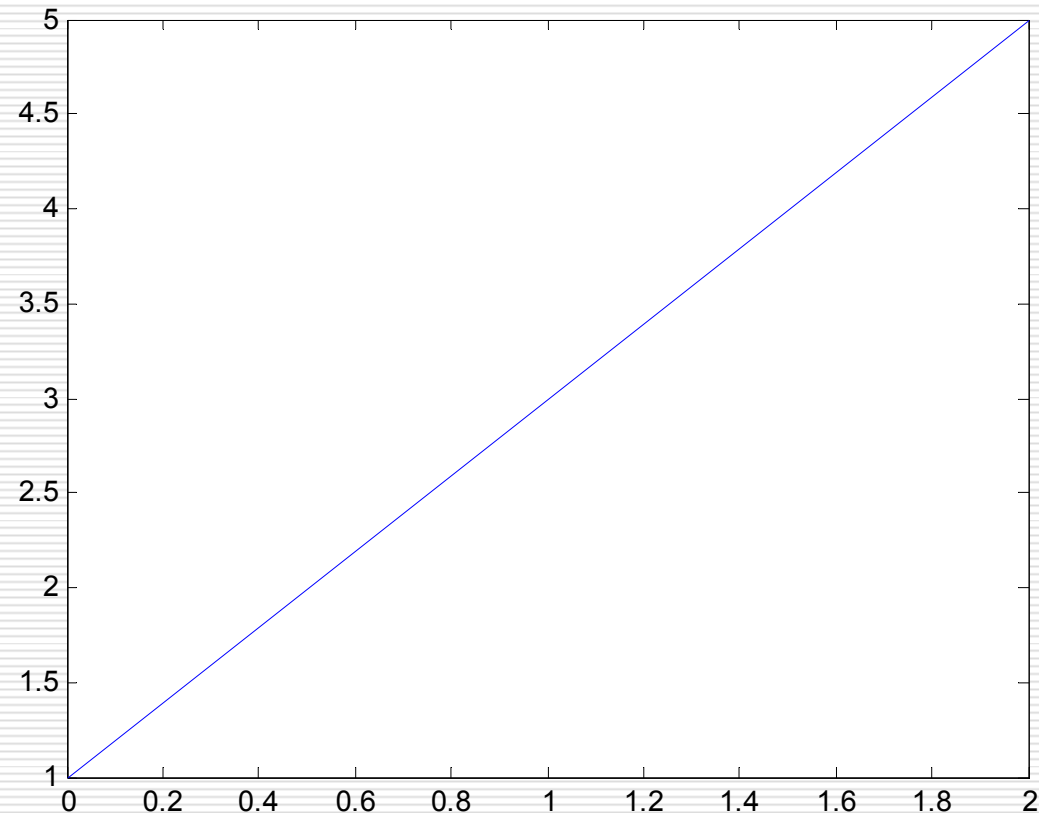
# Εξίσωση 1<sup>ου</sup> βαθμού

---

$$f(x) = 2x + 1$$

|          |   |   |   |
|----------|---|---|---|
| <b>x</b> | 0 | 1 | 2 |
| <b>y</b> | 1 | 3 | 5 |

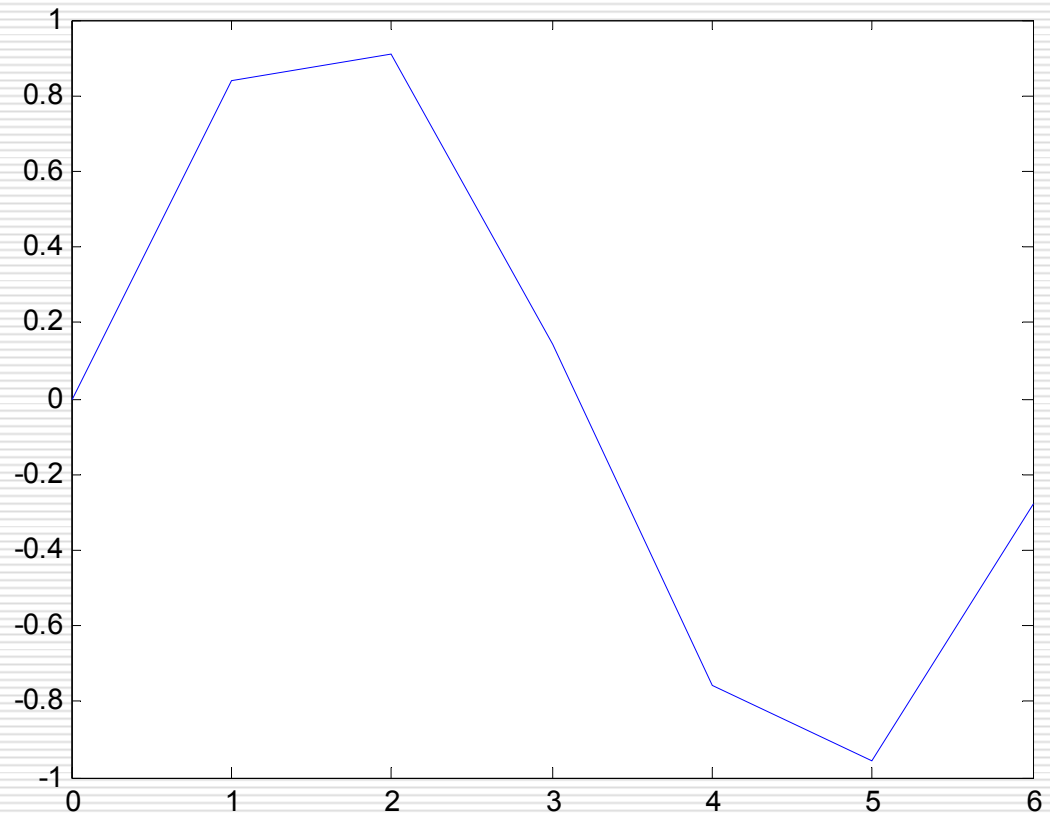
```
x=[0 1 2];  
y=[1 3 5];  
figure(1)  
plot(x,y)
```



# Γραφικές παραστάσεις

---

```
x=[0:1:2*pi];  
y=sin(x);  
figure(1)  
plot(x,y)
```

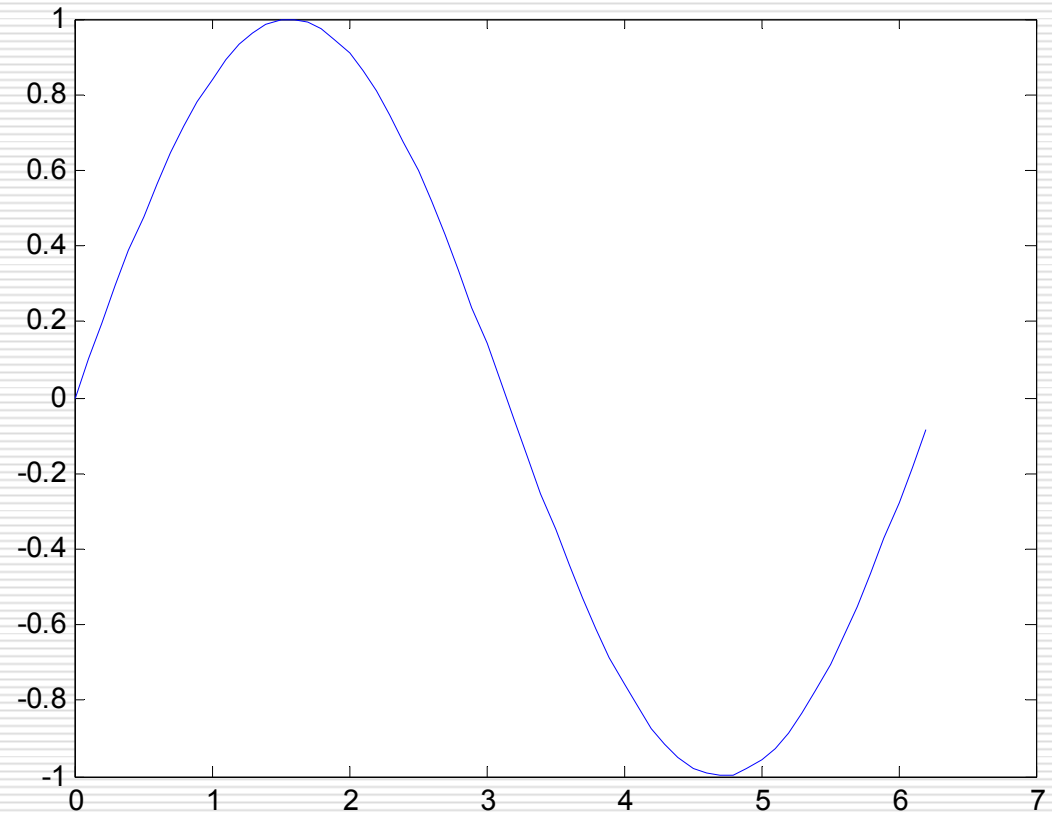




# Γραφικές παραστάσεις

---

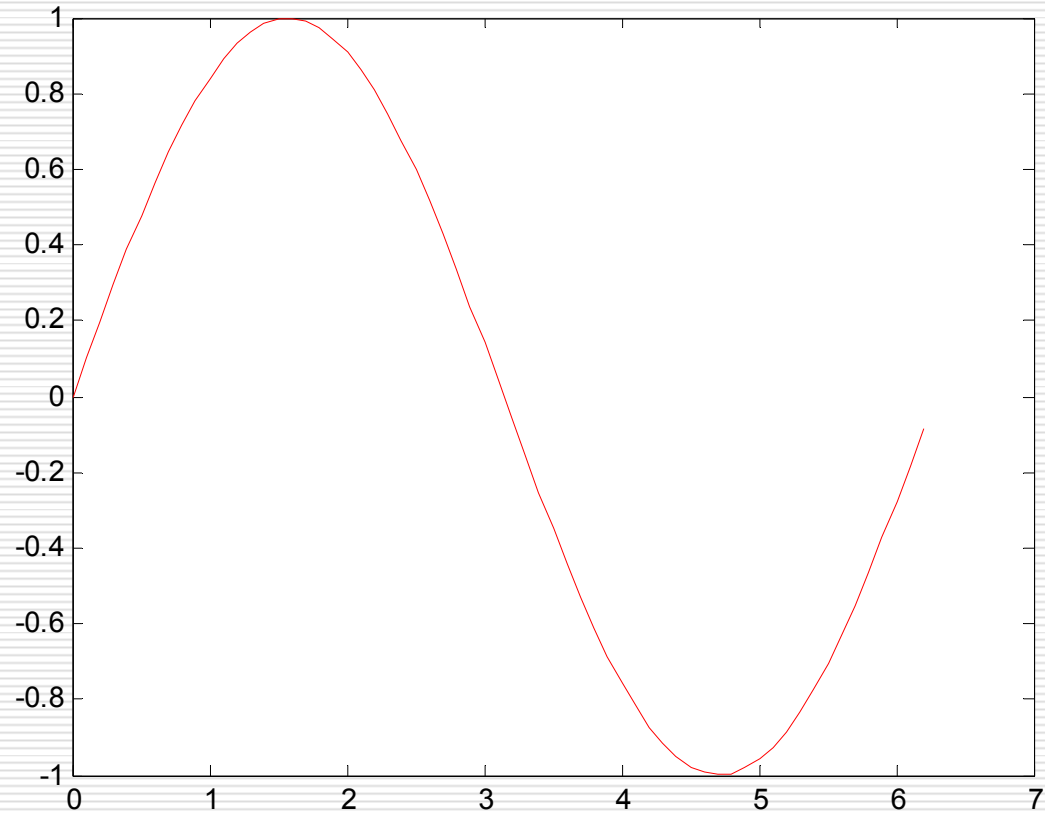
```
x=[0:0.1:2*pi];  
y=sin(x);  
figure(1)  
plot(x,y)
```



# Γραφικές παραστάσεις

---

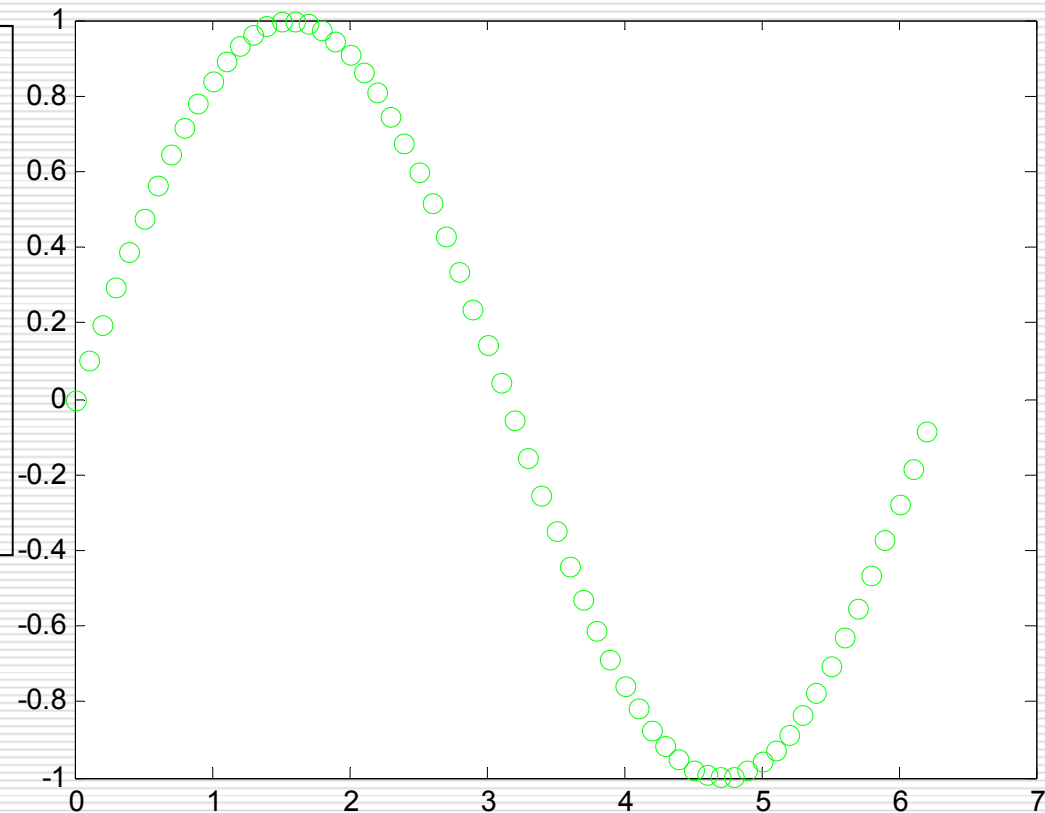
```
x=[0:0.1:2*pi];  
y=sin(x);  
figure(1)  
plot(x,y,'r')
```



# Γραφικές παραστάσεις

---

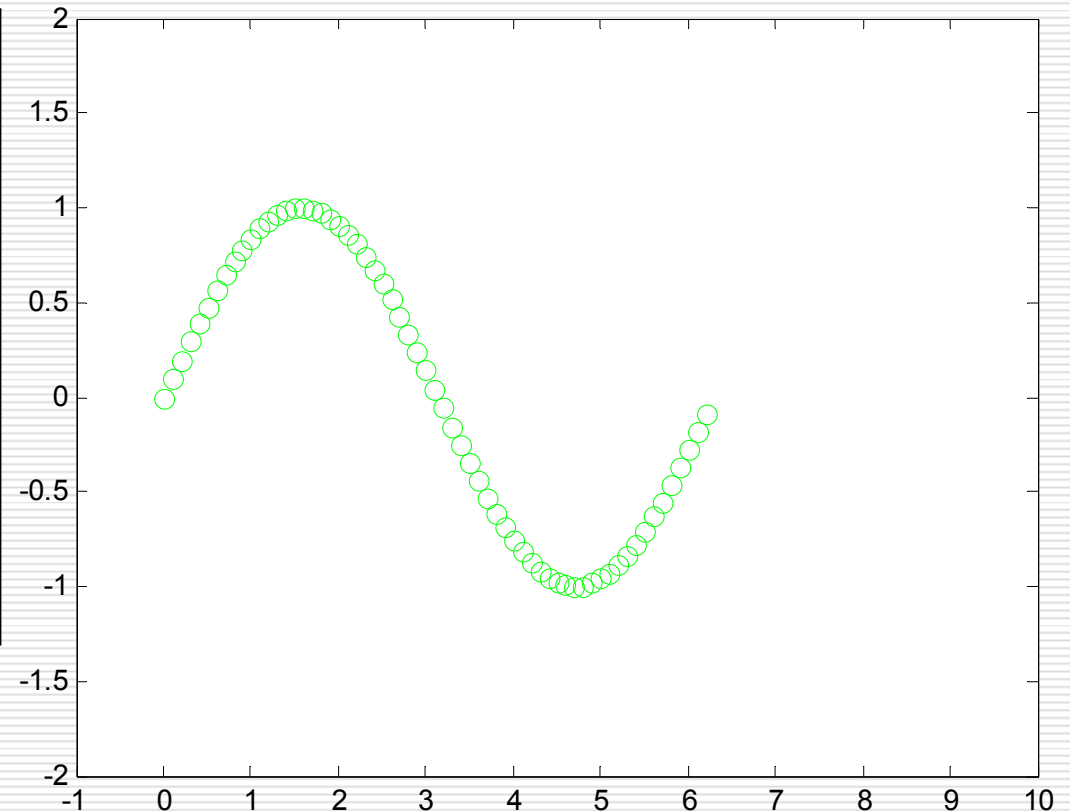
```
x=[0:0.1:2*pi];  
y=sin(x);  
figure(1)  
plot(x,y,'go')
```



# Αλλαγή αξόνων (axis)

---

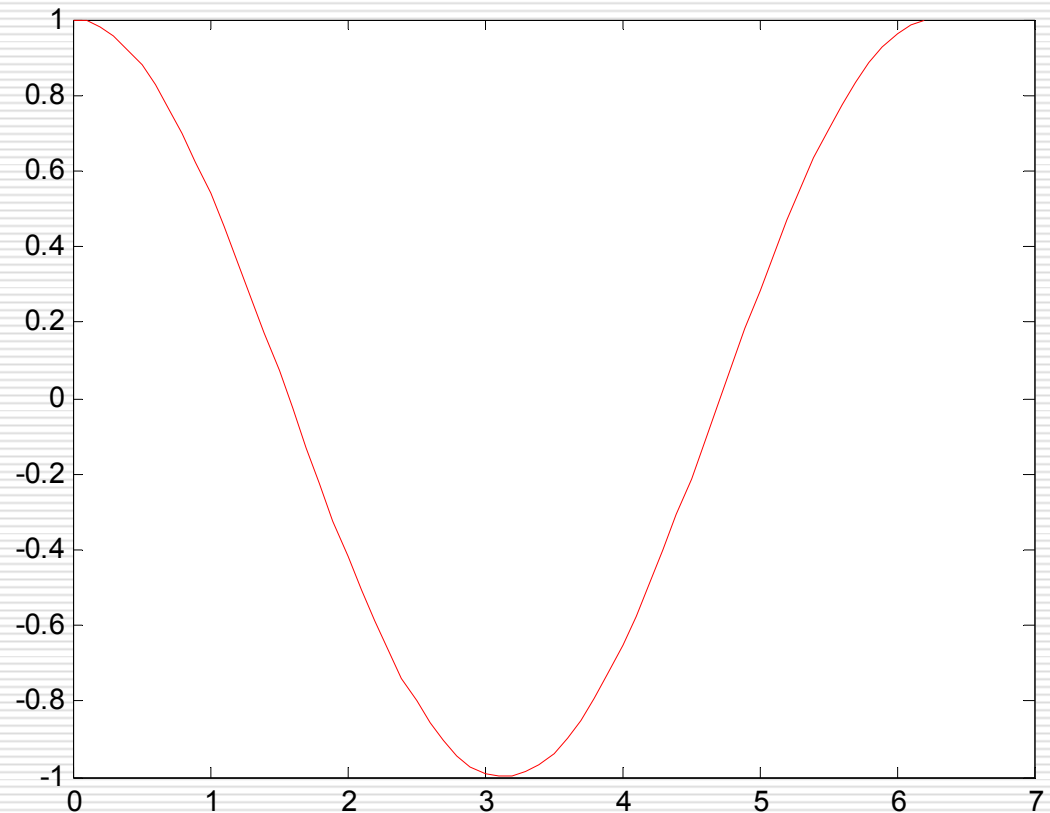
```
x=[0:0.1:2*pi];  
y=sin(x);  
figure(1)  
plot(x,y,'go')  
axis([-1 10 -2 2])
```



# Γραφικές παραστάσεις

---

```
x=[0:0.1:2*pi];  
y=sin(x);  
figure(1)  
plot(x,y,'g')  
y2=cos(x);  
plot(x,y2,'r')
```



# Γραφικές παραστάσεις

```
x=[0:0.1:2*pi];
```

```
y=sin(x);
```

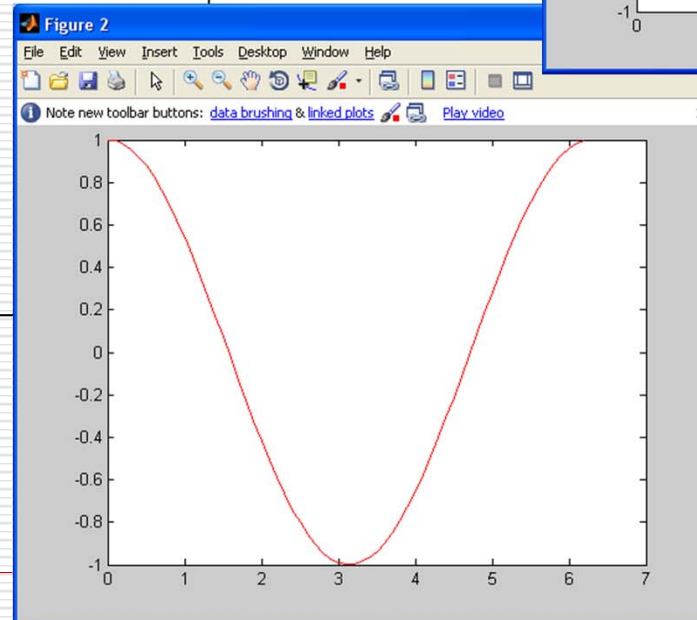
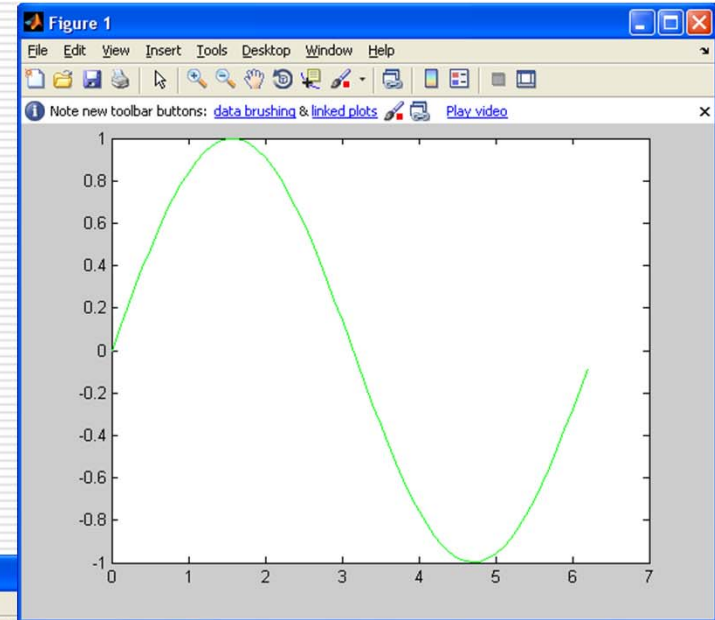
```
figure(1)
```

```
plot(x,y,'g')
```

```
y2=cos(x);
```

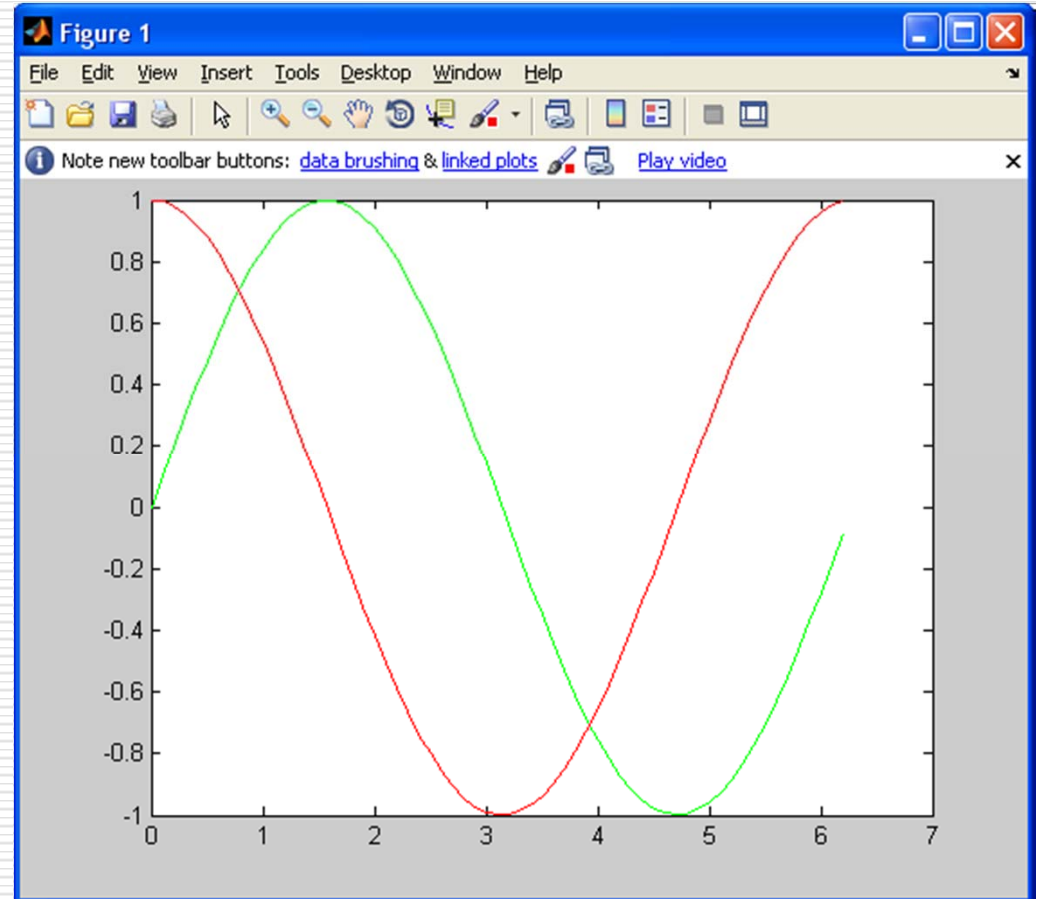
```
figure(2)
```

```
plot(x,y2,'r')
```



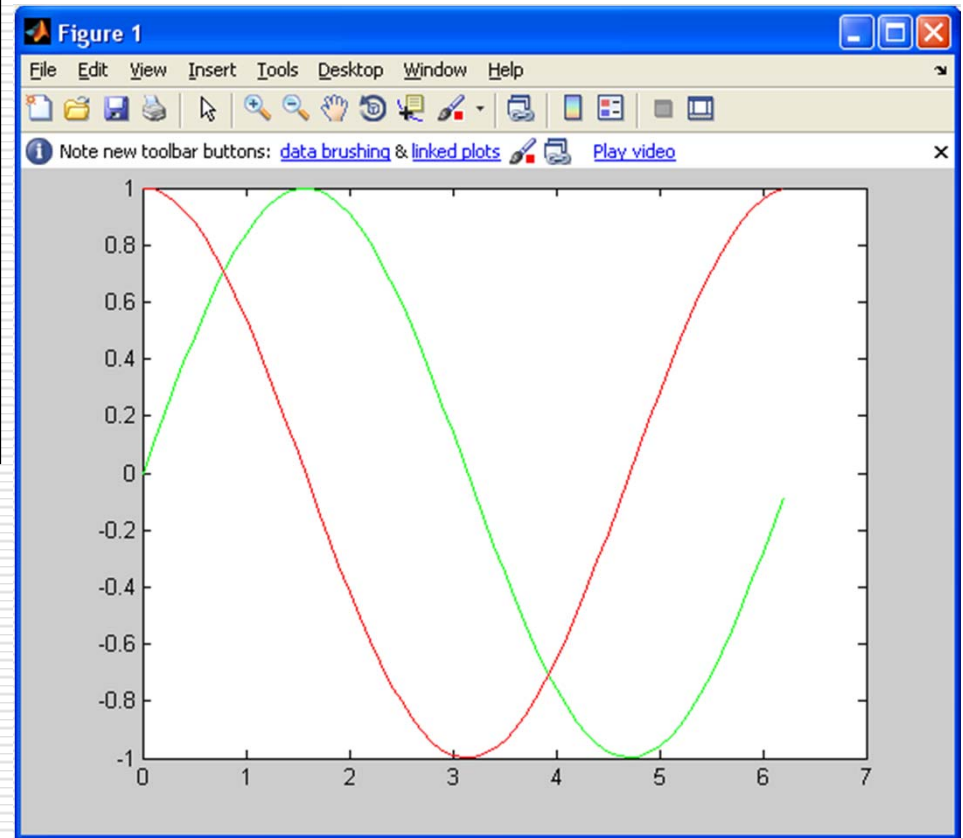
# Γραφικές παραστάσεις

```
x=[0:0.1:2*pi];  
y=sin(x);  
figure(1)  
plot(x,y,'g')  
y2=cos(x);  
hold on  
plot(x,y2,'r')
```



# Γραφικές παραστάσεις

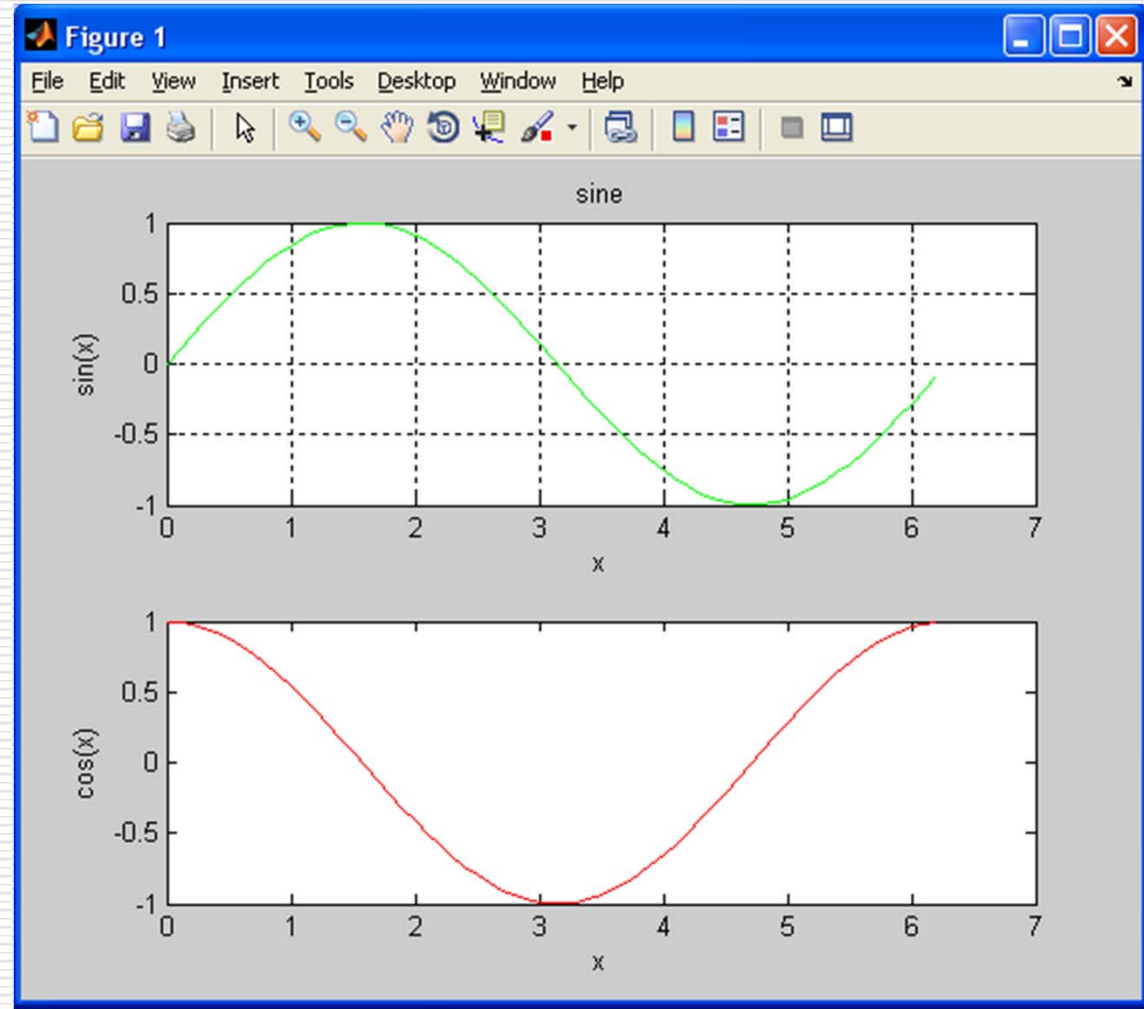
```
x=[0:0.1:2*pi];  
y=sin(x);  
y2=cos(x);  
figure(1)  
plot(x,y,'g',x,y2,'r')
```





# Γραφικές παραστάσεις

```
x=[0:0.1:2*pi];  
y=sin(x);  
figure(1)  
subplot(2,1,1)  
plot(x,y,'g')  
grid  
title('sine')  
xlabel('x')  
ylabel('sin(x)')  
y2=cos(x);  
subplot(2,1,2)  
plot(x,y2,'r')  
xlabel('x')  
ylabel('cos(x)')
```



# Λογικές πράξεις

---

*If* <εκφραση>

.....

*elseif*

.....

*else*

.....

*end*

## **Λογικοί Τελεστές**

`==, ~=, <, >, <=, >= &, |, ~`

Χρησιμοποιήστε την εντολή `help +` για να δείτε μια λίστα με όλους του τελεστές

---

# Βρόχοι

---

*for*

*...*

*end*

```
for i=1:5  
    [i i^2]  
end
```

*While*

*...*

*end*

```
i=1;  
while i<=5  
    [i i^2]  
    i=i+1;  
end
```

# Προγραμματισμός συνάρτησης

---

Στο MATLAB μπορούμε να προγραμματίσουμε τις συναρτήσεις που εμείς θέλουμε βάζοντας σαν πρώτη λέξη του προγράμματος το `function`. Αυτά τα αρχεία ονομάζονται αρχεία συναρτήσεων είναι και αυτά αρχεία **.m** και λαμβάνουν εξωτερικά ορίσματα τα οποία περιέχονται σε παρενθέσεις αμέσως μετά το όνομα της συνάρτησης.

---

# Προγραμματισμός συνάρτησης

---

- Φτιάξτε μια συνάρτηση η οποία θα υπολογίζει το παραγοντικό ( $x!$ ) ενός αριθμού (πχ  $5! = 1 * 2 * 3 * 4 * 5 = 120$ )

```
% paragontiko.m  
function y=paragontiko(x)  
y=1;  
for i=1:x  
    y=y*i;  
end
```

```
>> paragontiko(5)  
  
ans =  
  
    120
```

# Άσκηση 1

---

Να ορίσετε μια *function* που θα παίρνει σαν όρισμα ένα πίνακα και ένα αριθμό και θα επιστρέφει 0 αν δεν υπάρχει ο αριθμός στον πίνακα και ένα στην περίπτωση που υπάρχει.

---

# Άσκηση 1

---

```
%mysearch.m
function y=mysearch(pin,ar)
y=0;
for i=1:length(pin)
    if pin(i)==ar
        y=1;
        break
    end
end
end
```

```
>> mysearch([1 2 3 4],3)
ans =
    1
```

# Άσκηση 1

---

```
function y=mysearch(pin,ar)
y=strcat('Ο αριθμός,',num2str(ar),' δεν
        υπάρχει στον πίνακα,[',num2str(pin),']');
for i=1:length(pin)
    if pin(i)==ar
        y=strcat('Ο αριθμός,',num2str(ar),'
                υπάρχει στον πίνακα,[',num2str(pin),']');
        break
    end
end
end
```

---



# Άσκηση 2

---

- Να γίνουν οι απαραίτητες μετατροπές στην προηγούμενη συνάρτηση ώστε να λειτουργεί για δισδιάστατους πίνακες
-