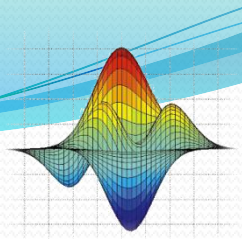


# Ψηφιακή Επεξεργασία Σημάτων

Εισαγωγή στο Matlab  
Μέρος Β΄

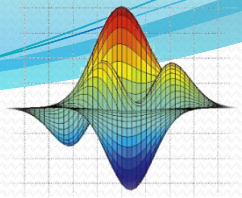
Κυριακίδης Ιωάννης

2011



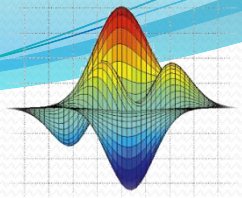
# Εισαγωγή στα M-Files

- Τα m-files της Matlab είναι τα αντίστοιχα των συναρτήσεων (functions) και των υπορουτινών (subroutines) που συναντάμε σε άλλες γλώσσες προγραμματισμού.
- Τα m-files που δημιουργεί ο χρήστης συμπληρώνουν τις συναρτήσεις των βιβλιοθηκών της Matlab που είναι επίσης m-files.
- Με τα m-files μπορούμε να δημιουργήσουμε δικές μας ακολουθίες εντολών είτε ως συναρτήσεις είτε ως script εντολών.
  - **Τα αρχεία script ή αρχεία εντολών** (script m-files or command files) δεν έχουν ορίσματα εισόδου και εξόδου αλλά εκτελούν μια ακολουθία εντολών σε μεταβλητές του χώρου εργασίας.
  - **Τα αρχεία συναρτήσεων** (function m-files) περιλαμβάνουν μια γραμμή ορισμού συνάρτησης, δέχονται ορίσματα εισόδου και επιστρέφουν μεταβλητές εξόδου. Οι εσωτερικές μεταβλητές των συναρτήσεων είναι τοπικές, εκτός αν δηλωθούν ως καθολικές με την εντολή global.



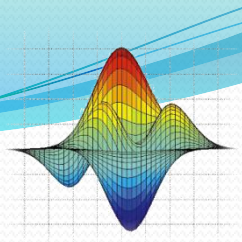
# Εισαγωγή στα M-Files

- Το Matlab διαθέτει τον δικό του editor για την δημιουργία και διαχείριση των m-files.
  - Όμως *m-files* μπορούμε να δημιουργήσουμε και με οποιονδήποτε editor όπως το *notepad* ή το *wordpad*.
- Για άνοιγμα του editor του Matlab χρησιμοποιήστε την εντολή `>> edit`
- Για να φορτώσετε ένα υπάρχον αρχείο: File → Open
- Θα πρέπει να βρίσκονται στον φάκελο εργασίας (*working directory*) ή στον φάκελο (*directory*) της Matlab ή στις διαδρομές αναζήτησης (*research paths*) της Matlab.
- Μπορούν να καλούν άλλα m-files ή ακόμα τον ίδιο τον εαυτό τους (αναδρομικά m-files).




# Ονόματα αρχείων

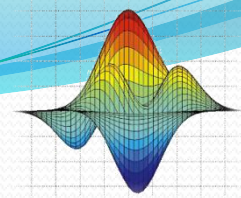
- Τα ονόματα των m-files μπορούν να περιέχουν μόνο:
  - Λατινικούς χαρακτήρες
  - Αριθμούς
  - Underscore (\_)
- Δεν επιτρέπεται να αρχίζουν με αριθμό.
- Δεν χρησιμοποιούνται ονόματα που έχουν δεσμευτεί από τη Matlab (π.χ. συναρτήσεις βιβλιοθήκης και εργαλειοθηκών) .
- Ορθά ονόματα αρχείων: askisi1.m, askisi1\_12345.m
- Λανθασμένα ονόματα αρχείων : 12315ask.m, ask-1.m, 1234.m



# Αρχεία εντολών (script)

- Τα αρχεία τύπου script περιέχουν μια ακολουθία εντολών της Matlab η οποία εκτελείται αν γράψουμε το όνομα του αρχείου (χωρίς την επέκταση .m), π.χ.:  

```
>> myscript1
```
- Επίσης η εκτέλεση της εντολής μπορεί να γίνει πατώντας το κουμπί “” από το toolbar.
- Τα script files είναι χρήσιμα για την εισαγωγή δεδομένων (π.χ. μεγάλων πινάκων) και για την επανάληψη μεγάλων ακολουθιών εντολών για διαφορετικά δεδομένα.
- Για την δημιουργία ενός νέου αρχείου εντολών:  
*File* → *New* → *Blank M-File*



# Παράδειγμα αρχείου εντολών

```
% myscript1.m
```

```
clear
```

```
clc
```

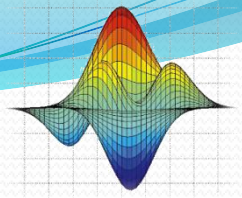
```
A=[1:5;6:10;11:15]
```

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

```
C=2*A
```

```
Command Window
```

```
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  
  
fx >>
```



# Αρχεία συναρτήσεων (functions)

- Τα αρχεία συναρτήσεων (function m-files) περιέχουν μια ολοκληρωμένη ακολουθία εντολών της Matlab για την επίτευξη κάποιου στόχου.

Λαμβάνουν μεταβλητές εισόδου

input1, input2, ....

και υπολογίζουν τις μεταβλητές εξόδου

output1, output2, .....

- Η δομή τους είναι η εξής:

```
function [output1, output2, ....] = filename (input1, input2, ....)
```

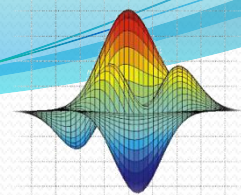
```
%Πληροφορίες για την συνάρτηση (επιστρέφονται από την εντολή help)
```

```
κώδικας υπολογισμού των εξόδων
```

```
output1 = .....
```

```
output2 = .....
```

```
end
```



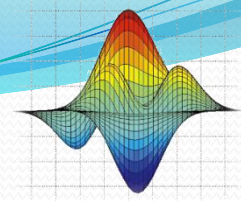
# Δημιουργία νέου αρχείου συνάρτησης

- Για νέο αρχείο συναρτήσεων: *File* → *New* → *Function M-File*

The screenshot shows the MATLAB Editor window titled "Editor - Untitled3\*". The menu bar includes File, Edit, Text, Go, Cell, Tools, Debug, Desktop, Window, and Help. The toolbar contains various icons for file operations, editing, and execution. Below the toolbar is a numeric keypad with fields for "1.0" and "1.1", and buttons for "+", "x", "%%", and "i". The main editor area contains the following code:

```
1 function [ output_args ] = Untitled3( input_args )
2 %UNTITLED3 Summary of this function goes here
3 % Detailed explanation goes here
4
5
6 end
7
8
```

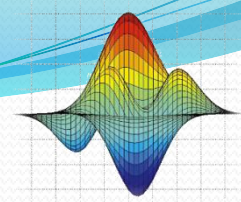
The status bar at the bottom indicates "Untitled3", "Ln 6 Col 4", and "OVR".



# Παράδειγμα αρχείου συνάρτησης

- Για να φτιάξουμε μια συνάρτηση με όνομα mySum η οποία θα λαμβάνει ως είσοδο 2 στοιχεία και επιστρέφει το άθροισμα τους θα πρέπει να γραφεί ο εξής κώδικας:

```
1 function [c] = mySum(a,b)
2 %mySum Prosθetei dyo arithmous tous a kai b
3 % Epistrefei to athroisma dyo arithmwv
4
5     c=a+b;
6
7 end
```



# Χρήση αρχείου συνάρτησης

- Παρακάτω βλέπουμε πως μπορούμε να χρησιμοποιήσουμε την συνάρτηση `mySum` που μόλις δημιουργήσαμε.

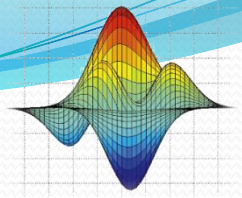
```
Command Window
>> mySum(5,3)

ans =

     8

>> help mySum
mySum Προσθετει δυο αριθμους tous a kai b
      Επιστrefei to athroisma δυο αριθμwn

fx >>
```



# Δομή της εντολής If...else....

- Η εντολή if μας επιτρέπει να ελέγξουμε αν μια (ή περισσότερες) συνθήκες ισχύουν και να εκτελέσουμε σε κάθε περίπτωση την επιθυμητή ακολουθία εντολών και πράξεων.

```
if <συνθήκη 1>
```

```
    <εντολές>
```

```
else if <συνθήκη 2>
```

```
    <εντολές>
```

```
else
```

```
    <εντολές>
```

```
end
```

```
if A>B
```

```
    C=A-B
```

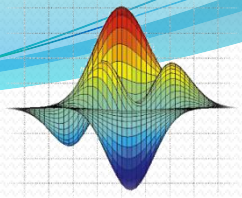
```
else if B>A
```

```
    C=B-A
```

```
else
```

```
    C=0
```

```
end
```



# Δομή της εντολής switch-case

Η εντολή switch-case μας δίνει τη δυνατότητα να επιλέξουμε για εκτέλεση μια ομάδα εντολών από άλλες πιθανές ομάδες. Η γενική της δομή έχει ως εξής:

```
switch <μεταβλητή>
```

```
    case <τιμή1>
```

```
        <εντολές>
```

```
    case <τιμή2>
```

```
        <εντολές>
```

```
    otherwise
```

```
        <εντολές>
```

```
end
```

```
switch vathmos
```

```
    case {0,1,2,3}
```

```
        disp('Den perases')
```

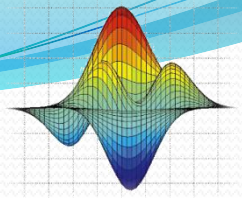
```
    case {4}
```

```
        disp('Katoxyrws')
```

```
    otherwise
```

```
        disp('Perases')
```

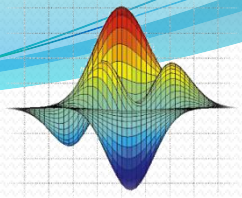
```
end
```



# Δομή της εντολής for (βρόχος)

- Οι λέξεις “for” και “end” χρησιμοποιούνται στην αρχή και στο τέλος του βρόχου, ο μετρητής index παίρνει τις τιμές από την initial value μέχρι την final value με βήμα step, και οι <εντολές> εκτελούνται για όλες τις τιμές του μετρητή index.
- Αν παραλείψουμε το βήμα step, τότε η Matlab χρησιμοποιεί ως βήμα την τιμή 1.
- Αξίζει να αναφέρουμε ότι το βήμα μπορεί να είναι αρνητικός αριθμός.

```
for index = initial value : step : final value  
    <εντολές>  
end
```



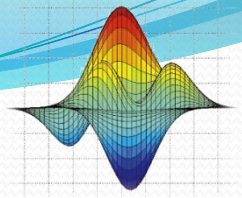
# Παράδειγμα εντολής for

- Για να υψώσουμε τα στοιχεία του διανύσματος  $x=[1\ 2\ 3\ 4\ 5]$ , μπορούμε να χρησιμοποιήσουμε τον βρόχο for.

```
for i=1:5  
    x(i) = i^2;  
end
```

- Ο παραπάνω τρόπος δεν είναι ο καταλληλότερος για να υψώσουμε τα στοιχεία ενός διανύσματος σε μια δύναμη. Ο ενδεικτικός τρόπος στη Matlab είναι ο εξής:

```
x=(1:5).^2
```

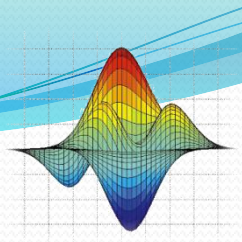


# Δομή της εντολής while (βρόχος)

- Οι λέξεις “**while**” και “**end**” χρησιμοποιούνται στην αρχή και στο τέλος του βρόχου. Η ακολουθία <εντολών> εκτελείται όσο η <συνθήκη> είναι αληθής.

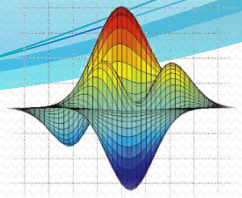
```
while <συνθήκη>  
    <εντολές>  
end
```

```
n=0;  
while n<=5  
    n = n+1;  
end
```



# Οι εντολές `break` και `continue`

- Οι βρόχοι `for` και `while` μπορούν να διακοπούν με την εντολή **`break`** η οποία μεταφέρει τον έλεγχο στην πρώτη εντολή μετά το τελικό `end` του βρόχου στον οποίο χρησιμοποιείται.
- Έτσι αν έχουμε πολλαπλούς βρόχους ο έλεγχος περνά στον επόμενο (εξωτερικό βρόχο). Η εντολή `break` ορίζεται μόνο μέσα σε βρόχους `for` και `while`.
- Η εντολή **`continue`** μεταβιβάζει τον έλεγχο στην επόμενη επανάληψη ενός βρόχου `for` ή `while` χωρίς να εκτελεστούν οι επόμενες εντολές του βρόχου.



# Προγραμματισμός συνάρτησης

- Να ορίσετε μια συνάρτηση με όνομα "paragontiko" η οποία θα υπολογίζει το παραγοντικό ( $x!$ ) ενός αριθμού (π.χ.  $5! = 1*2*3*4*5=120$ )

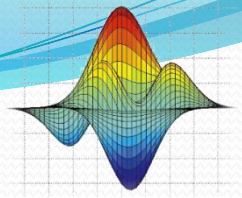
```
function y=paragontiko(x)
    y=1;
    for i=1:x
        y=y*i;
    end
end
```

```
Command Window
>> paragontiko(5)

ans =

    120

fx >> |
```



# Προγραμματισμός συνάρτησης

- Να ορίσετε μια συνάρτηση με όνομα `mysearch` όπου θα παίρνει σαν όρισμα ένα πίνακα και ένα αριθμό και θα επιστρέφει 0 αν δεν υπάρχει ο αριθμός στον πίνακα και 1 εάν υπάρχει.

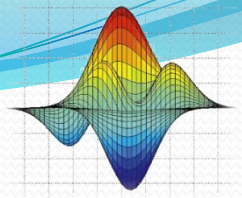
```
function y=mysearch(pin,ar)
    y=0;
    for i=1:length(pin)
        if pin(i)==ar
            y=1;
            break;
        end
    end
end
```

```
Command Window
>> mysearch([1 2 3 4],3)

ans =

     1

fx >>
```



# Προγραμματισμός συνάρτησης

- Τροποποιήστε την προηγούμενη συνάρτηση “mysearch” ώστε να επιστρέφει την πρώτη θέση όπου βρήκε τον αριθμό αναζήτησης.

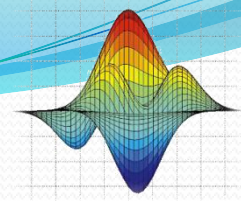
```
function y=mysearch(pin,ar)
    y=0;
    for i=1:length(pin)
        if pin(i)==ar
            y=i;
            break;
        end
    end
end
```

```
Command Window
>> mysearch([4 5 6 7],5)

ans =

     2

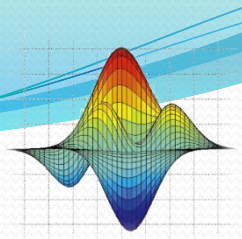
fx >> |
```



# Μαθηματικές συναρτήσεις

- Μερικές βασικές μαθηματικές συναρτήσεις και η σημασία τους φαίνονται στον παρακάτω πίνακα. Για να δείτε όλες τις μαθηματικές συναρτήσεις τρέξτε την εντολή `>> help elfun`

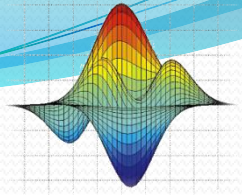
Συνάρτηση	Λειτουργία
sin	Ημίτονο
cos	Συνημίτονο
tan	Εφαπτομένη
sec	Τέμνουσα
cot	Συνεφαπτομένη
exp	Εκθετική συνάρτηση $e^{-x}$
log	Νεπέριος (φυσικός) λογάριθμος
log10	Δεκαδικός λογάριθμος
sqrt	Τετραγωνική ρίζα
abs	Απόλυτη τιμή
rem	Υπόλοιπο διαίρεσης



# Δημιουργία γραφικών παραστάσεων

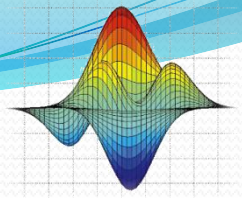
- Για να απεικονίσουμε γραφικά τα αποτελέσματα μιας δισδιάστατης συνάρτησης χρησιμοποιούμε συναρτήσεις της βιβλιοθήκης `graph2d`.
- Η βασική εντολή της βιβλιοθήκης που θα χρησιμοποιήσουμε είναι η εντολή `plot`.
- **`plot(x,y,option)`** εμφανίζει την γραφική παράσταση του  $y$  ως προς το  $x$ , το τρίτο όρισμα της εντολής χρησιμοποιείται για να καθορίσουμε το χρώμα και το σύμβολο σχεδίασης της γραφικής παράστασης.

# Συναρτήσεις για γραφικά



- Οι κυριότερες συναρτήσεις για γραφικά είναι συγκεντρωμένες στον ακόλουθο πίνακα:

Εντολή	Περιγραφή	Παράδειγμα
plot	Δημιουργεί το γράφημα του y συναρτήσει του x	plot(x,y)
title	Προσθήκη τίτλου	title('Titlos')
xlabel	Προσθήκη ετικέτας στον οριζόντιο άξονα	xlabel('Χρονos, t')
ylabel	Προσθήκη ετικέτας στον κατακόρυφο άξονα	ylabel('Taxuthta, cm/s')
legend	Προσθήκη λεζάντας	legend('First', 'Second')
text	Προσθήκη κειμένου στη θέση (xi, yi)	text(xi, yi, 'string')
grid	Δημιουργία πλέγματος	grid ή grid on ή grid off
figure	Άνοιγμα (άλλου) παραθύρου γραφικών	figure(2)
hold	Πάγωμα του τρέχον παραθύρου γραφικών για το σχεδιασμό και άλλων καμπυλών	hold on ή hold off
axis	Κλείδωμα/ξεκλείδωμα αξόνων Ίσες μονάδες αξόνων Διαγραφή αξόνων Όρια αξόνων.	axis axis equal axis off axis([xmin, xmax,ymin,ymax])



# Παράδειγμα γραφικών παραστάσεων

- Έστω ότι θέλουμε να δημιουργήσουμε την γραφική παράσταση την συνάρτησης  $f(x)=2x+1$ , για  $x=0,1,2$ .

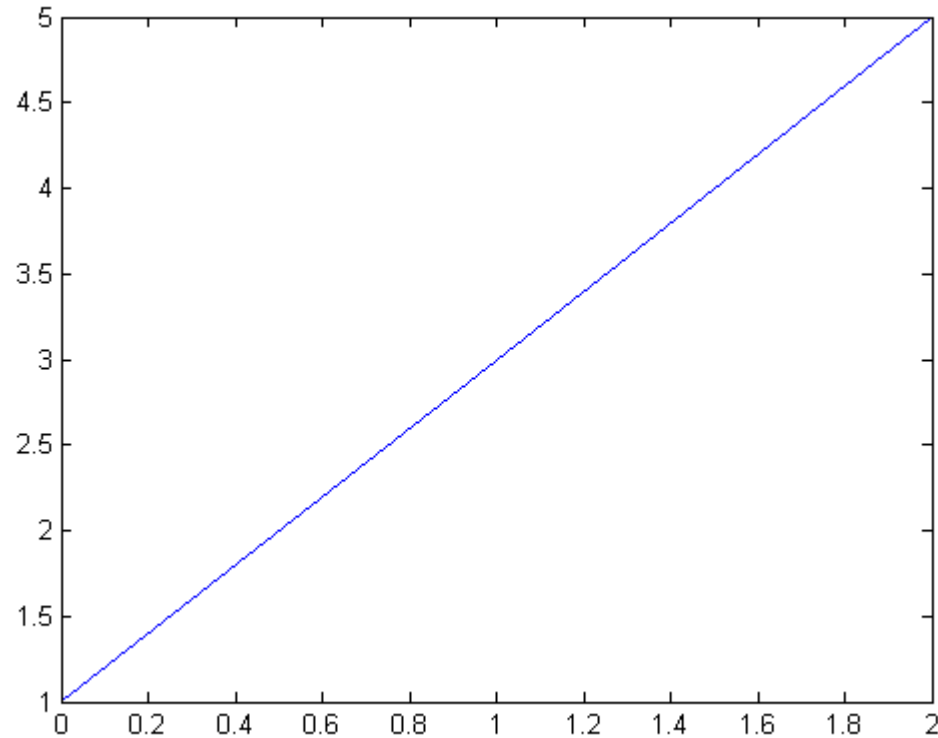
%Ο κώδικας:

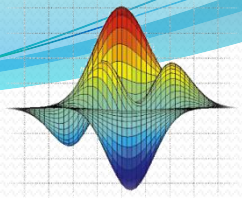
```
x=[0 1 2];
```

```
y=[1 3 5];
```

```
figure(1)
```

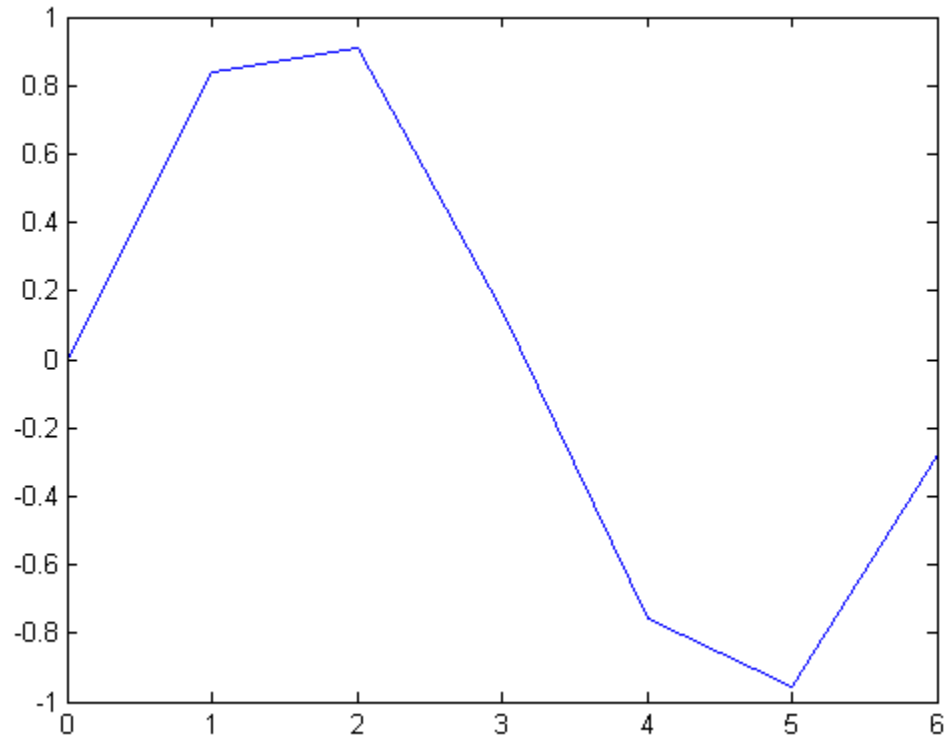
```
plot(x,y);
```



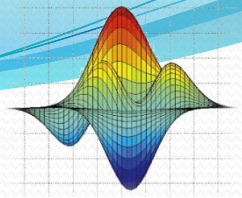


# Παράδειγμα γραφικών παραστάσεων

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

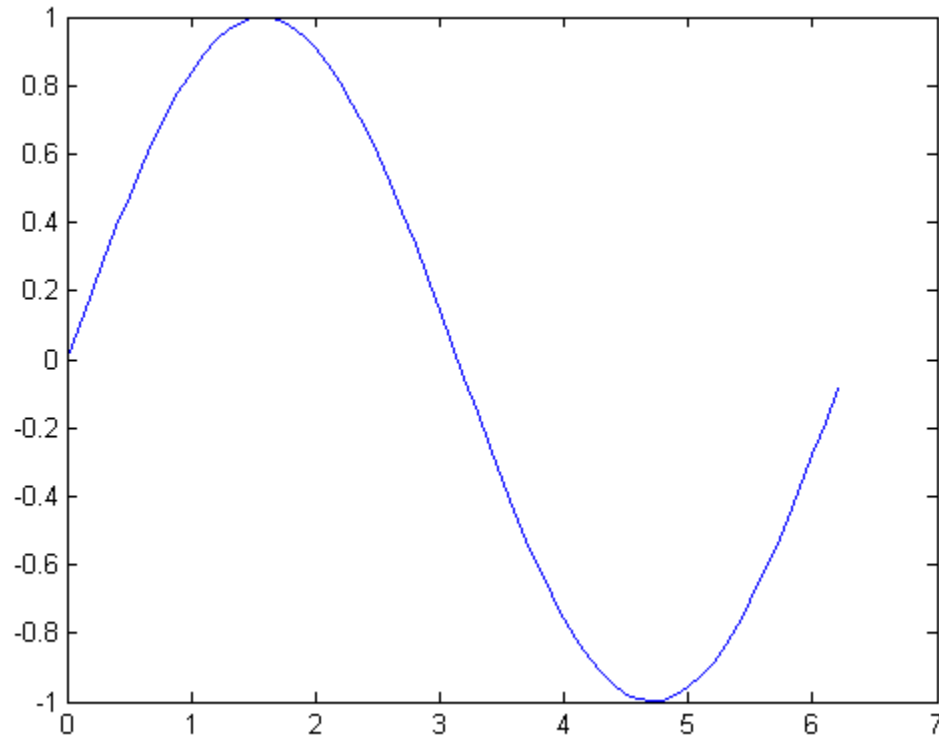


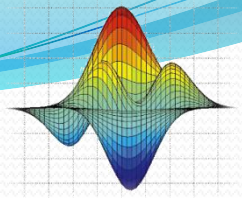
- Το γράφημα δεν εμφανίζεται ομαλό γιατί δεν έχουμε αρκετά δείγματα στον άξονα x.



# Παράδειγμα γραφικών παραστάσεων

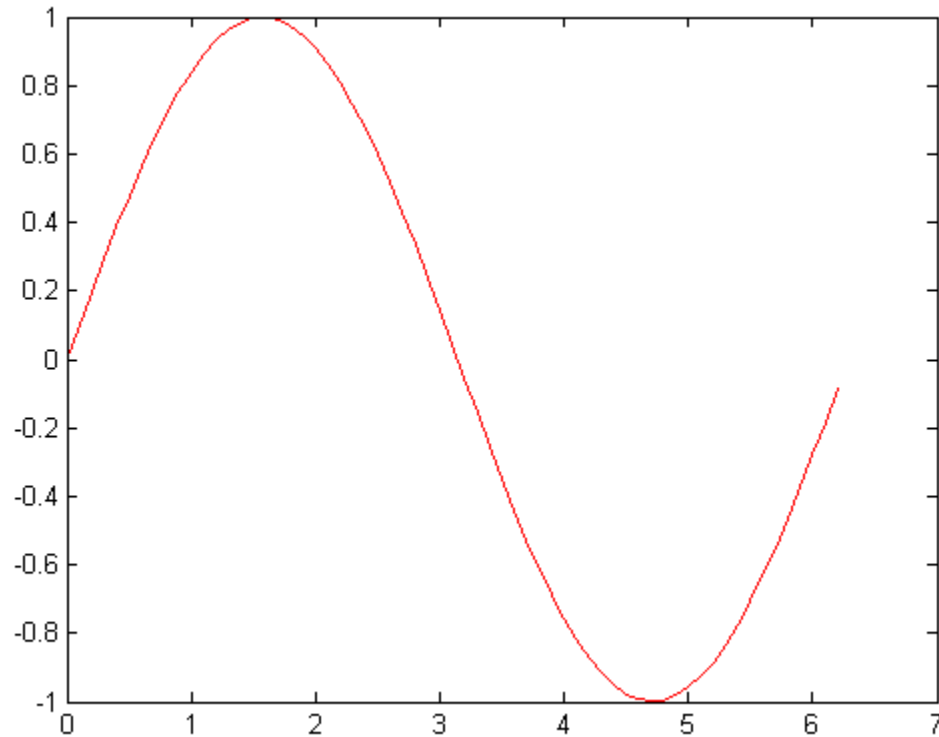
```
x=[0 : 0.1 : 2*pi];  
y=sin(x);  
figure(1)  
plot(x,y)
```

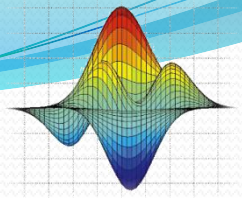




# Παράδειγμα γραφικών παραστάσεων

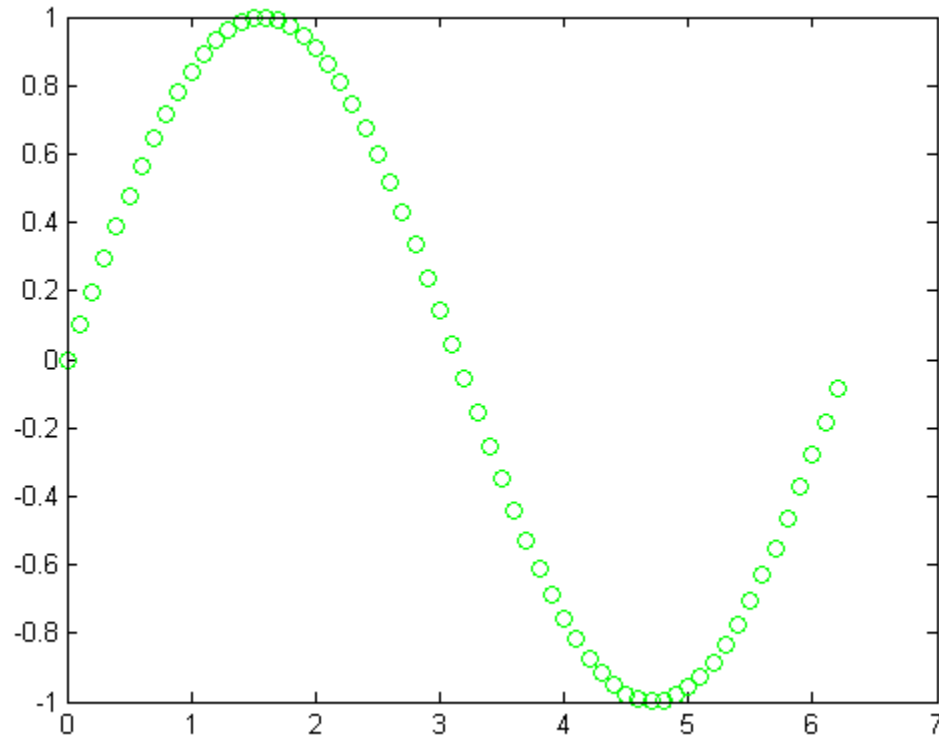
```
x=[0 : 0.1 : 2*pi];  
y=sin(x);  
figure(1)  
plot(x,y,'r')
```

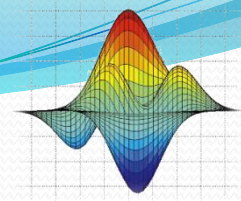




# Παράδειγμα γραφικών παραστάσεων

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





# Παράδειγμα γραφικών παραστάσεων

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

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

```
figure(1)
```

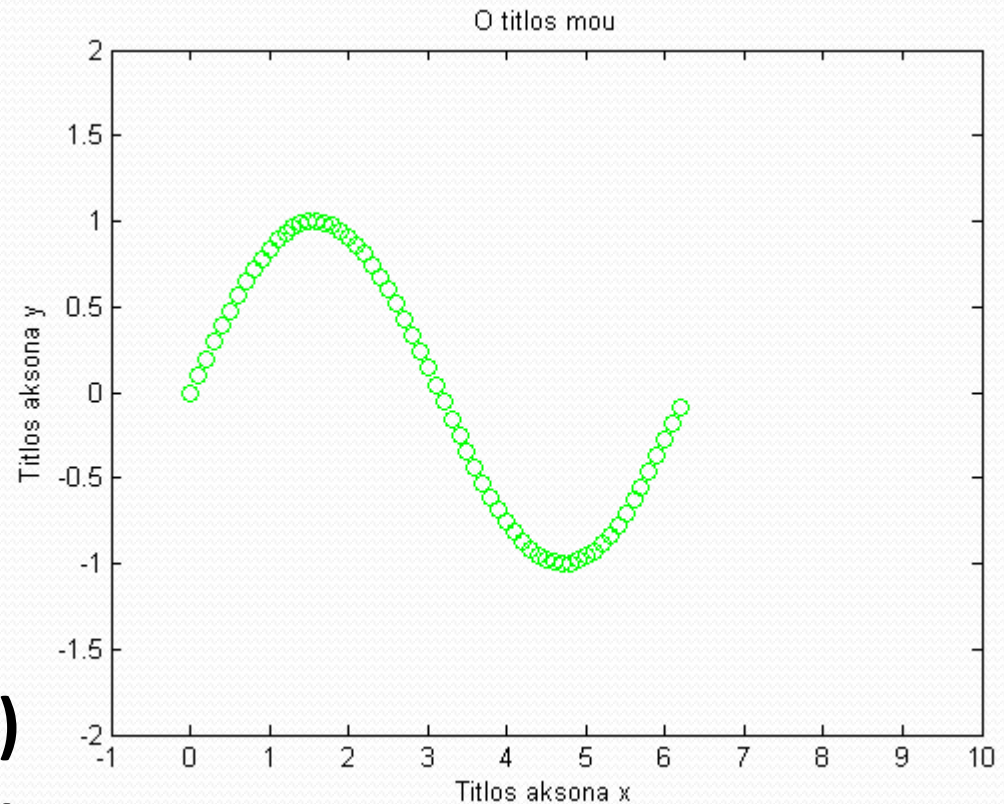
```
plot(x,y,'go')
```

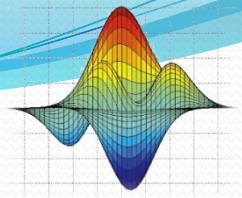
```
axis([-1 10 -2 2])
```

```
title('O titlos mou')
```

```
xlabel('Titlos aksona x')
```

```
ylabel('Titlos aksona y')
```





# Παράδειγμα γραφικών παραστάσεων

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

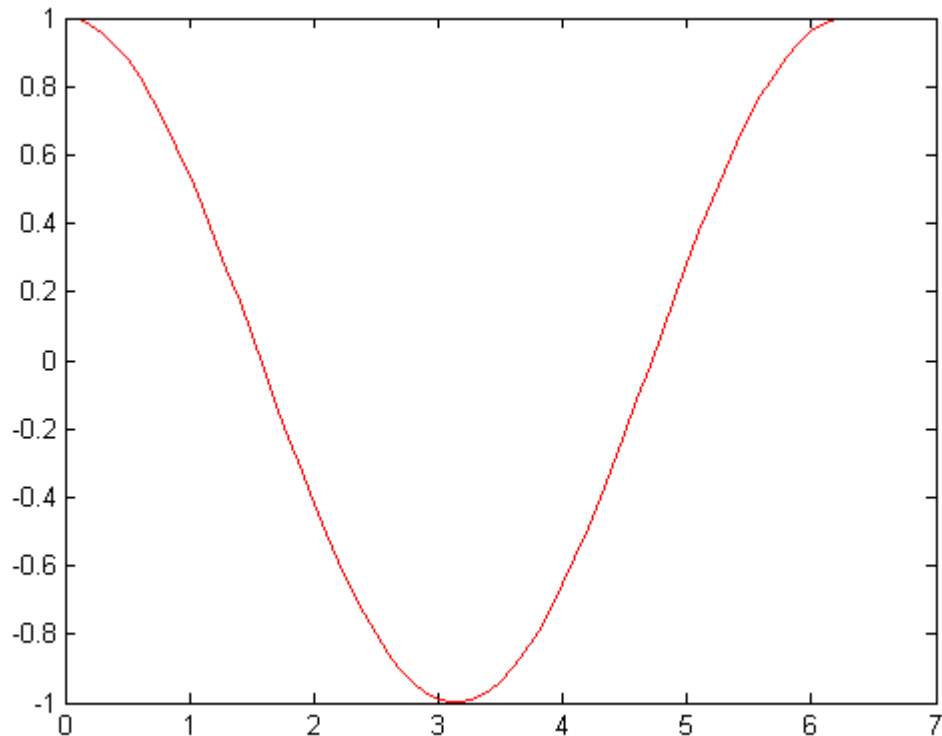
```
y=sin(x);
```

```
figure(1)
```

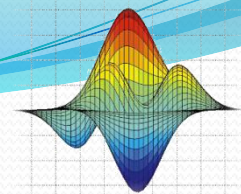
```
plot(x,y,'g')
```

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

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



- Παρατηρούμε ότι ενώ τρέχουμε δύο φορές την συνάρτηση plot εμφανίζεται μόνο ένα figure!



# Παράδειγμα γραφικών παραστάσεων

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

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

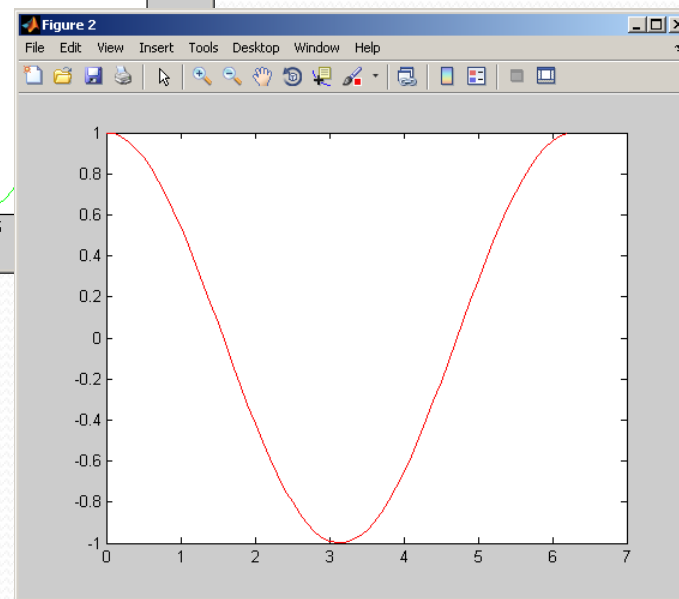
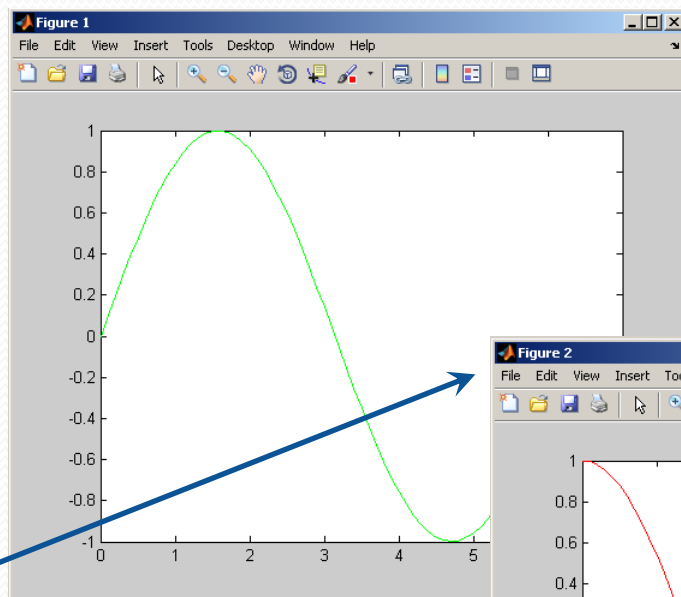
```
figure(1)
```

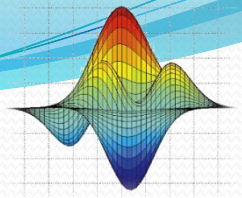
```
plot(x,y,'g')
```

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

```
figure(2)
```

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





# Παράδειγμα γραφικών παραστάσεων

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

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

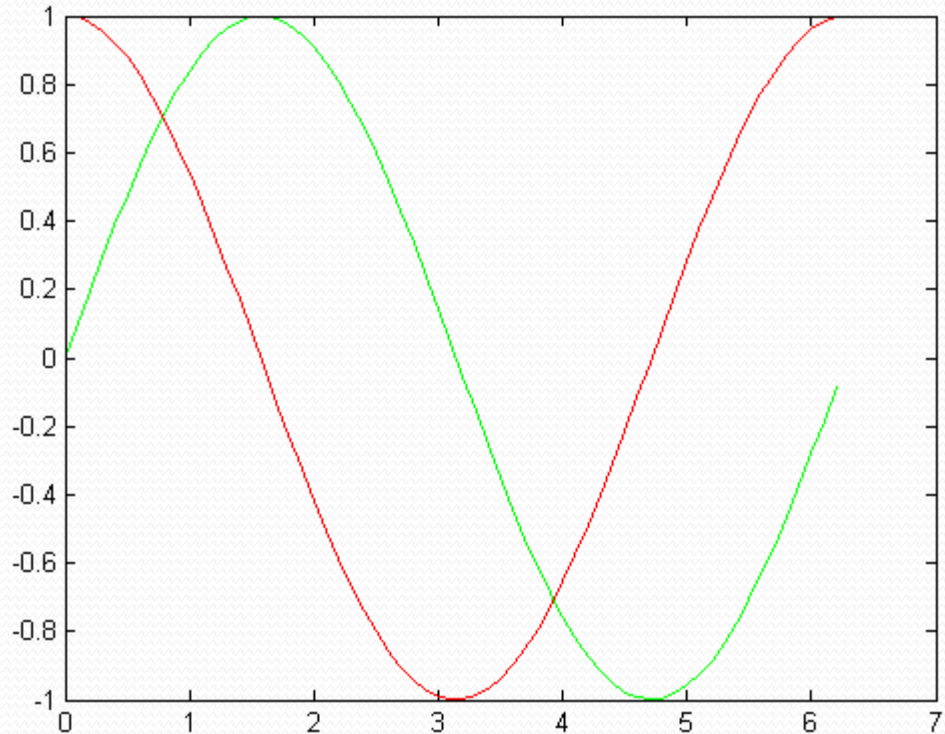
```
figure(1)
```

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

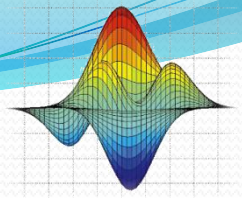
```
y2=cos(x);
```

```
hold on
```

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

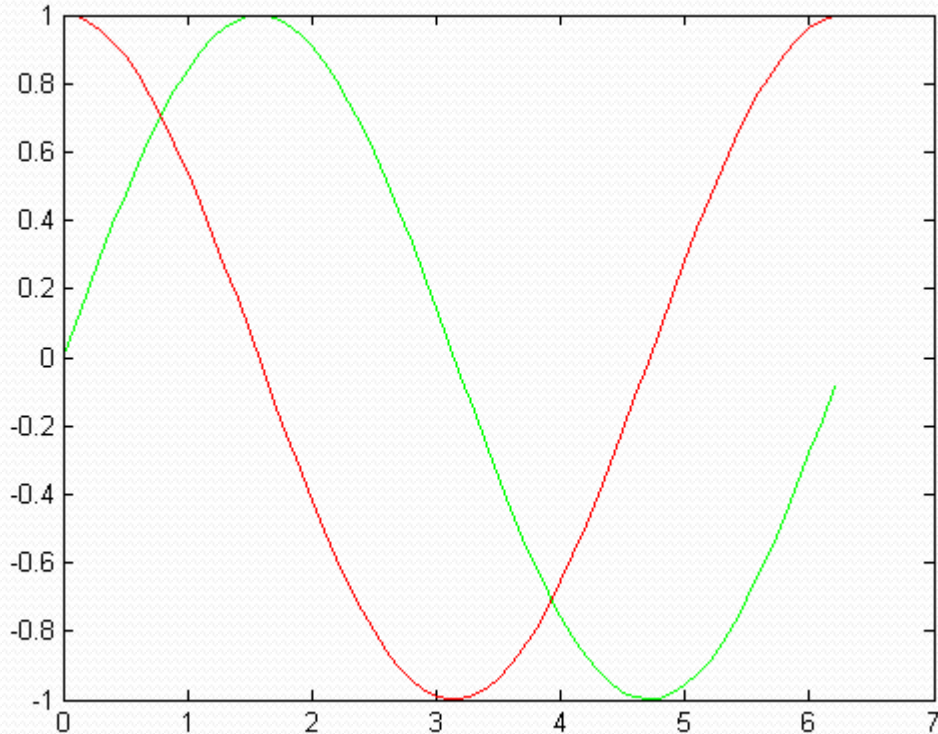


- Με την εντολή “hold on”, λέμε στο figure 1 να “περιμένει” για να σχεδιάσει και την επόμενη καμπύλη.

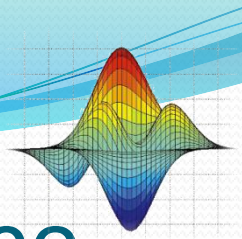


## Παράδειγμα γραφικών παραστάσεων

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



- Αντί να χρησιμοποιήσουμε την εντολή “hold on” μπορούμε να δημιουργήσουμε το ίδιο γράφημα χρησιμοποιώντας την εντολή plot, όπως παρουσιάζεται παραπάνω.



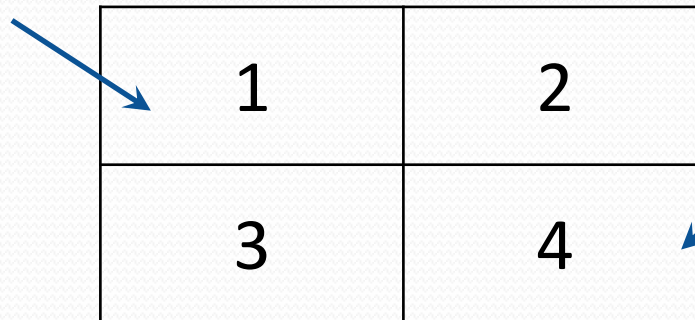
# Πολλαπλά γραφήματα στο ίδιο παράθυρο

- Η συνάρτηση **subplot** μας επιτρέπει να βάλουμε πολλά γραφήματα στο ίδιο παράθυρο γραφικών.

**subplot(πλήθος γραμμών, πλήθος στηλών, θέση)**

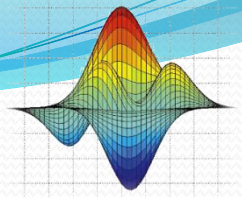
- Η αρίθμηση (θέση) των γραφημάτων γίνεται κατά γραμμές, δηλαδή:

**subplot(2,2,1)**



**subplot(2,2,4)**

# Παράδειγμα γραφικών παραστάσεων



```
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)')
```

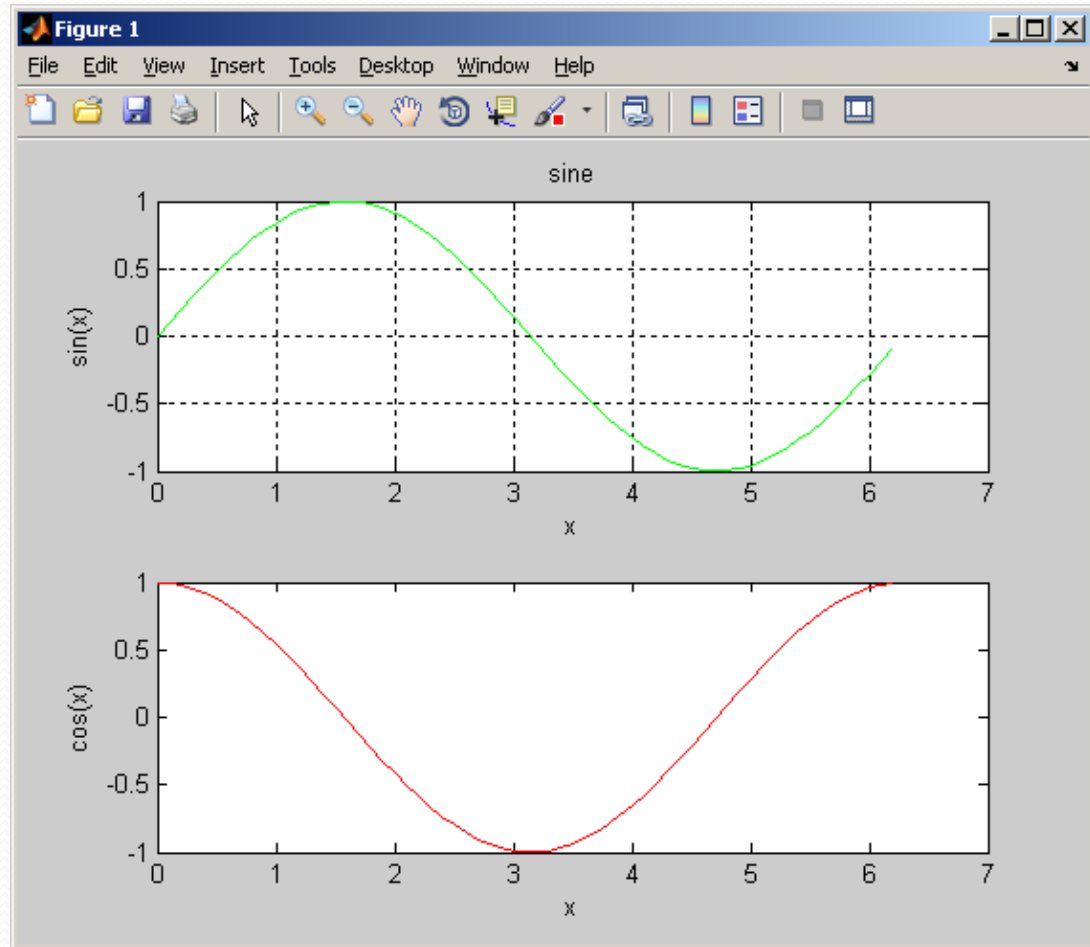
```
y2=cos(x);
```

```
subplot(2,1,2)
```

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

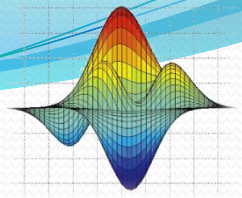
```
xlabel('x')
```

```
ylabel('cos(x)')
```





Απορίες - Ερωτήσεις ;



# Ασκήσεις για το σπίτι



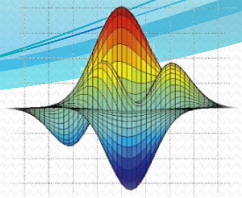
Οι ασκήσεις είναι ατομικές !!!

Αποστέλλετε όλα τα αρχεία m-file σε ένα συμπιεσμένο

αρχείο με όνομα: lab02\_OMX\_YYYY

(όπου X ο αριθμός ομάδας εργαστηρίου και YYYY το ΑΜ σας)

Στο email: [kyriakidis@teicrete.gr](mailto:kyriakidis@teicrete.gr)



# Άσκηση 1 για το σπίτι

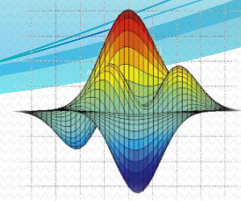
- Τροποποιήστε την συνάρτηση “mysearch” (η οποία επιστρέφει την πρώτη θέση που βρήκε τον αριθμό αναζήτησης), ώστε να λειτουργεί για δισδιάστατους πίνακες. Εάν δεν υπάρχει θα επιστρέφει [0,0].

```
Command Window
```

```
>> x=[1 2 3;4 5 6;7 8 9]
x =
     1     2     3
     4     5     6
     7     8     9

>> mysearch(x,6)
ans =
     2     3

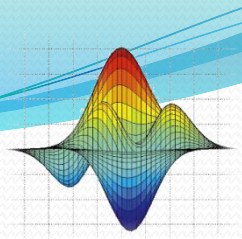
fx >> |
```



## Άσκηση 2 για το σπίτι

- Τροποποιήστε την άσκηση 1 ώστε να μπορεί να βρίσκει τις όλες τις θέσεις που υπάρχει ο αριθμός αναζήτησης.

```
Command Window
>> x=[1 2 3;3 4 6;7 3 2]
x =
     1     2     3
     3     4     6
     7     3     2
>> mysearch(x,3)
ans =
     1     3
     2     1
     3     2
fx >> |
```



## Άσκηση 3 για το σπίτι

Σε ένα script m-file δημιουργήστε τη γραφική παράσταση (με πλέγμα) του συνημίτονου στο διάστημα από  $[0$  έως  $2\pi]$  με βήμα  $\pi/90$ . Δημιουργήστε μια νέα γραφική παράσταση (με πλέγμα) του ημίτονου στο διάστημα  $[-2\pi, 2\pi]$  διατηρώντας τον ίδιο ρυθμό δειγματοληψίας.

1. Εμφανίστε τις 2 γραφικές παραστάσεις σε ξεχωριστά figures.
2. Εμφανίστε και τις 2 γραφικές παραστάσεις στο ίδιο figure με κοινούς άξονες στο διάστημα  $[-\pi, \pi]$ . Χρησιμοποιήστε την εντολή subplot. Τα figures να εμφανίζονται στην ίδια γραμμή. Τέλος προσθέστε ετικέτες στους άξονες.