

Μέθοδοι Επίλυσης με Η/Υ

4^ο Μάθημα: Δομές επανάληψης & Γραφήματα

Ν.Δ. Λαγαρός, Α. Στάμος, Χ. Φραγκουδάκης

Το σημερινό μάθημα περιέχει:

1. δομές επανάληψης:
 - η εντολή for
 - η εντολή while
2. εισαγωγή στα γραφήματα:
 - plot
 - figure
 - hold on

δομές επανάληψης – for loops

Σύνταξη:

```
for x = array  
    (... εντολές...)  
end
```

παράδειγμα:

```
for i = 1:10
```

```
    disp(i)                % η τιμή του i αλλάζει σε κάθε επανάληψη
```

```
    k = 50*i+32;           % η τιμή του k αλλάζει σε κάθε επανάληψη  
                           (δεν αποθηκεύεται όμως)
```

```
    A(i) = sin(i*pi/10);   % η θέση i του μητρώου A παίρνει τιμές
```

```
end
```

k %η τελευταία τιμή του $k=50*10+32=82$ **ποιά η διαφορά με την disp(k)?**

δομές επανάληψης

με την εντολή **for** μπορούμε να δώσουμε τιμές σε ένα μητρώο:

```
>> for n = 1:5
>>     for m = 5:-1:1
>>         A(n,m) = n^2+m^2;
>>     end
>> end
```

ή να πολλαπλασιάσουμε μητρώα, διανύσματα κλπ:

```
>> a = [0, -1, 4; 9, -14, 25; -34, 49, 64];
>> v = [4; -2; 10];

>> b=zeros(3,1); % αρχικά ορίζουμε και μηδενίζουμε το b
>> for i = 1:3
>>     for j = 1:3
>>         b(i)=b(i)+a(i,j)*v(j);
>>     end
>> end
```

Ελέγξτε το αποτέλεσμα του πολλαπλασιασμού με την πράξη: $a*v$

δομές επανάληψης – while loop

Είδαμε την εντολή **for** που συντάσσεται:

```
for x = array  
    (... εντολές...)  
end
```

Εναλλακτικά μπορούμε να χρησιμοποιήσουμε την εντολή **while**

```
while συνθήκη  
    (... εντολές...)  
end
```

```
A = zeros(10,1);  
while n <= 10    % συνθήκη ενεργή όσο το n είναι μικρότερο ή ίσο από 10.  
  
    n = n + 2;    % η τιμή του n μεταβάλλεται σε κάθε επανάληψη  
    disp(n)  
    A(n) = 50*n+32; % δίνουμε τιμές στα A(2), A(4), A(6), A(8) και A(10)  
  
end
```

δομές επανάληψης – while loop

Παράδειγμα while-loop:

ο παρακάτω βρόχος ονομάζεται ατέρμων, γιατί ??

```
i=0;  
while 1<2  
  
    i=i+1  
    disp('DE STAMATW, DE STAMATW ... ')  
  
end
```

πατήστε ταυτόχρονα Ctrl + C για να διακόψετε το πρόγραμμα

Διαστάσεις διανυσμάτων και μητρώων

οι εντολές **length** και **size**

έστω μητρώο A διαστάσεων **3x4**

size(A) δίνει τις διαστάσεις ενός μητρώου (π.χ. 3, 4)

```
>> ss=size(a)  
ss =  
     3     4
```

size(A,2) δίνει το μήκος της 2^{ης} διάστασης του μητρώου A (π.χ 4)

```
>> ss=size(a,2)  
ss =  
     4
```

length(A) δίνει τη μεγαλύτερη διάσταση (π.χ 4)

```
>> ss=length(a)  
ss =  
     4
```

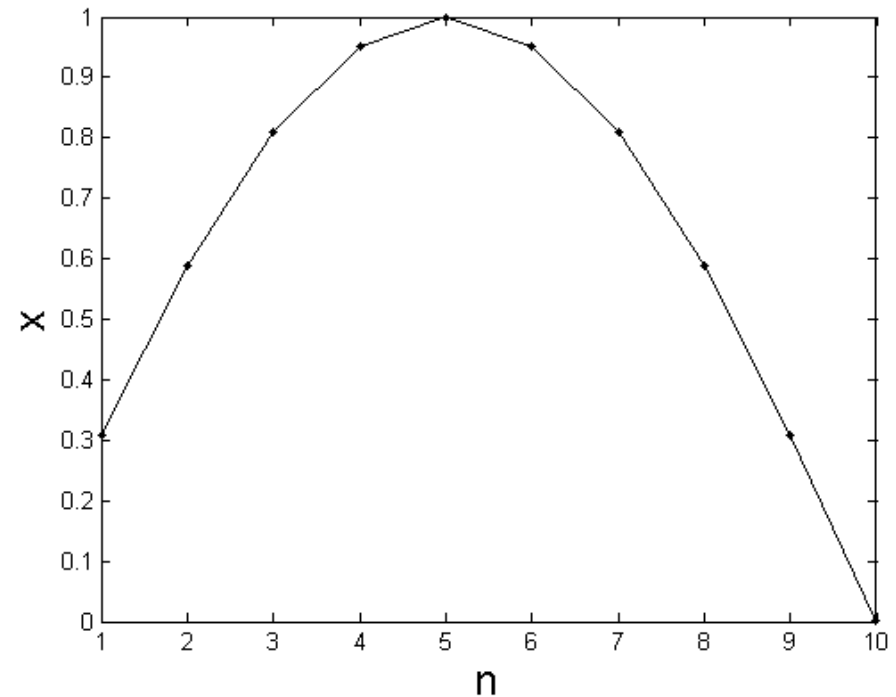
Γραφήματα στο matlab

Έστω η συνάρτηση $y = \sin(\pi x/10)$,
θέλουμε το γράφημα της στο
διάστημα 0,10.

```
n = 1:10;  
x= sin(n*pi/10) ;  
plot(n,x,'k.-')
```

Τα **n** και **x** είναι δύο διανύσματα
(με την ίδια διάσταση!)

η εντολή 'k.-' ορίζει το είδος της
γραμμής (k = μαύρο χρώμα,
. = ευθεία με κουκίδα σε κάθε
σημείο)



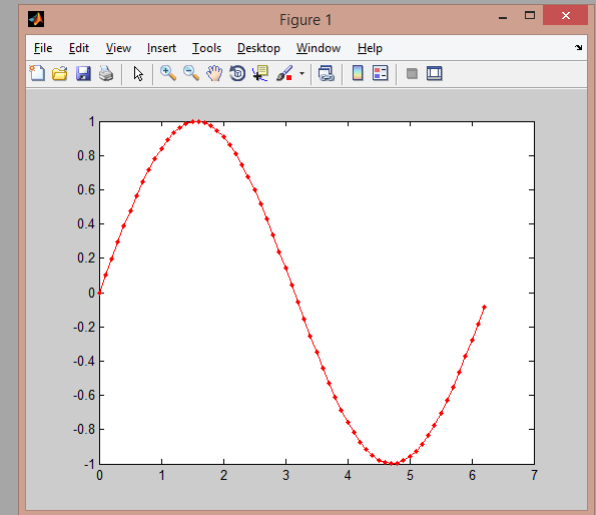
Συνάρτηση Plot

Απλή χρήση της συνάρτησης **plot**:

```
>>x = 0:0.1:2*pi
```

```
>>y = sin(x)
```

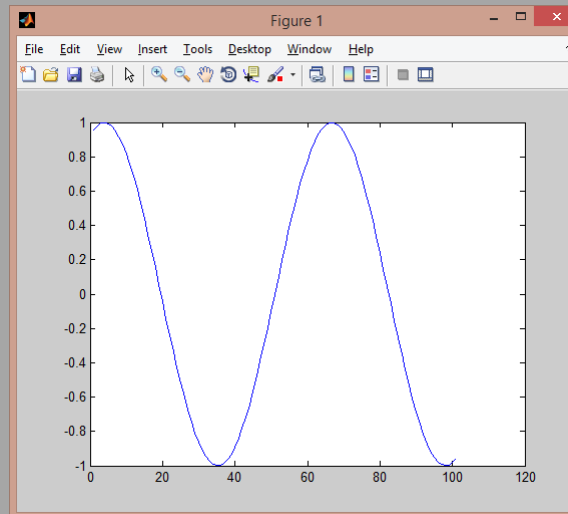
```
>>plot(x, y, 'r.-')
```



```
>>x=-5:1:5;
```

```
>>plot(sin(x))
```

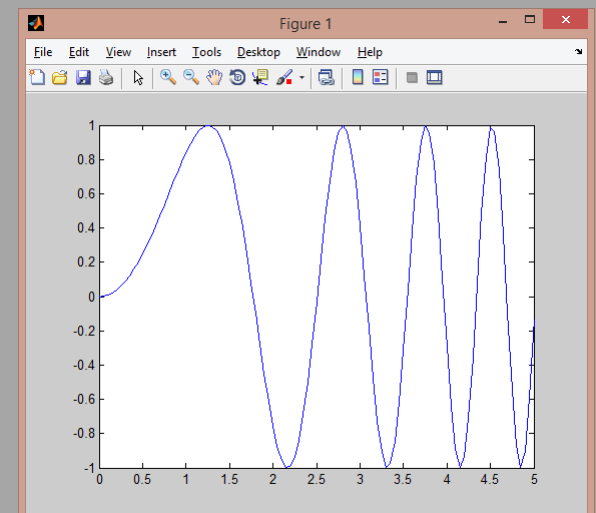
```
>>plot(x,sin(x), 'g')
```



```
>>x=0:0.05:5;
```

```
>>y=sin(x.^2);
```

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



hold on

Εάν θέλουμε στο ίδιο γράφημα να υπάρχουν 2 εντολές plot, χρησιμοποιούμε την εντολή **hold** (εάν την ξεχάσουμε θα τυπωθεί η μόνο η τελευταία εντολή plot)

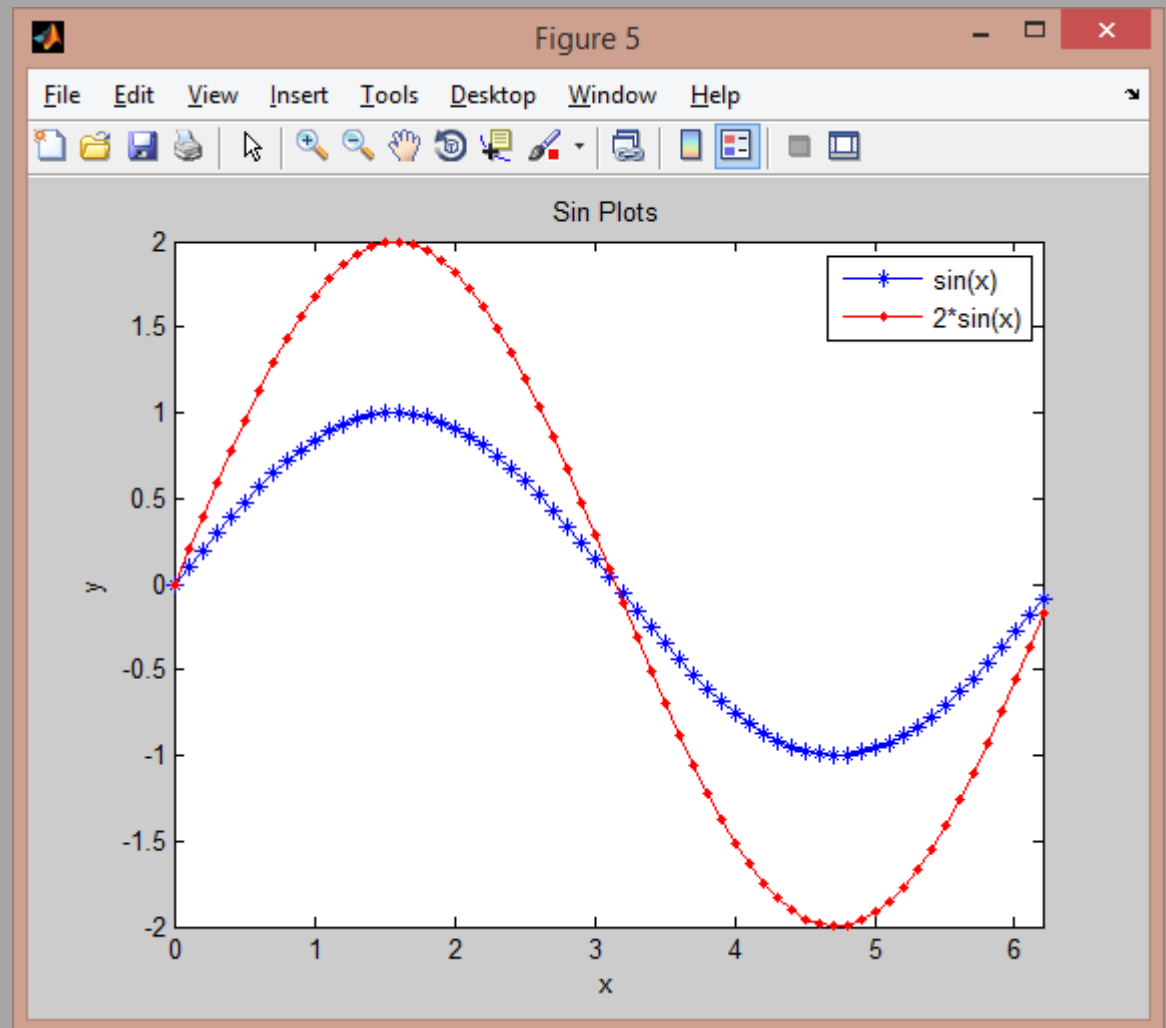
```
x = 0:0.1:2*pi;
```

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

```
hold on
```

```
plot(x, y, 'b*-')
```

```
plot(x, 2*y, 'r.-')
```



hold on

Εάν θέλουμε στο ίδιο script να υπάρχουν 2 ή περισσότερα διαφορετικά γραφήματα, χρησιμοποιούμε την εντολή **figure**

```
>> x = 0:0.1:2*pi;
```

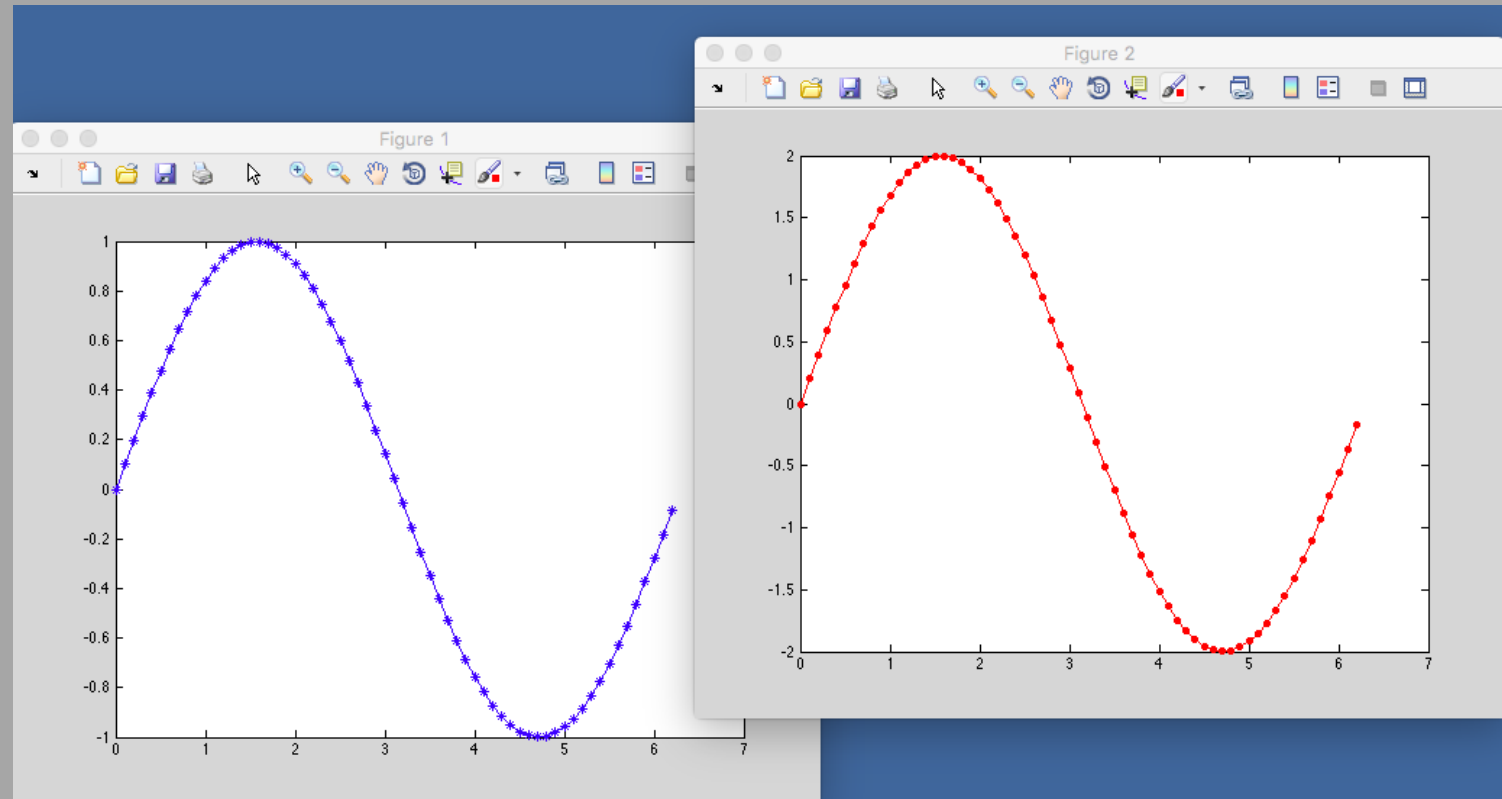
```
>> y = sin(x);
```

```
>> figure(1)
```

```
>> plot(x, y, 'b*-')
```

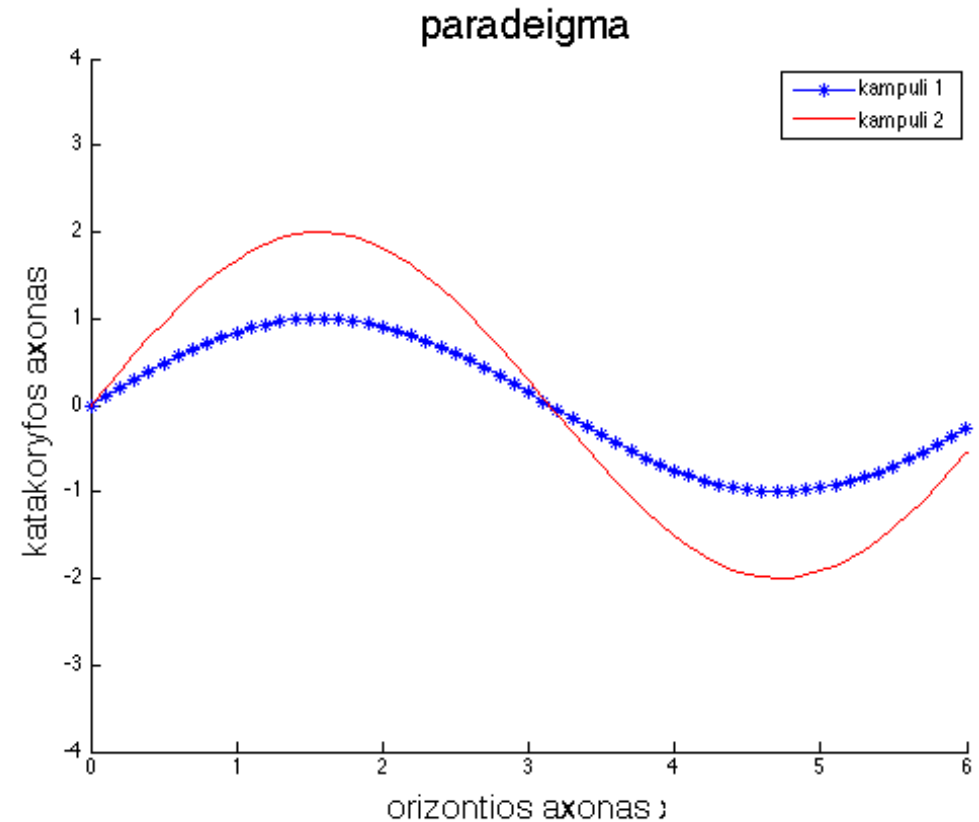
```
>> figure(2)
```

```
>> plot(x, 2*y, 'r.-')
```



τίτλος και άξονες γραφήματων

```
figure(1)
hold on
x = 0:0.1:2*pi;
y = sin(x);
plot(x, y, 'b*-')
plot(x, 2*y, 'r.-')
```



```
title('paradeigma', 'FontSize', 20) % τίλος γραφήματος &
                                     ορισμός μεγέθους γραμματοσειράς
xlabel('orizontios axonas x', 'FontSize', 16) % όνομα αξονα x
ylabel('katakoryfos axonas y', 'FontSize', 16) % όνομα αξονα y
legend('kampuli 1', 'kampuli 2') % υπόμνημα
axis([0 6 -4 4]) % όρια αξόνων [κάτω x, άνω x, κάτω y, άνω y]
```

διαφορετικά είδη γραμμών

Μπορούμε να έχουμε διαφορετικά είδη γραμμών και σημαδιών

Επιλογές
χρωμάτων:

Specifier	Color
y	yellow
m	magenta
c	cyan
r	red
g	green
b	blue
w	white
k	black

Επιλογές
σημαδιών:

Specifier	Marker
o	Circle
+	Plus sign
*	Asterisk
.	Point
x	Cross
s	Square
d	Diamond
^	Upward-pointing triangle
v	Downward-pointing triangle
>	Right-pointing triangle
<	Left-pointing triangle
p	Pentagram
h	Hexagram

```
plot(x, y, 'm*-')
```

```
plot(x, 2*y, 'b.-')
```

```
plot(x, 5*y, 'r+-', 'MarkerSize',10) % μέγεθος των σημαδιών, εδώ του «+»
```

διαφορετικά είδη γραμμών

Επιλογές γραμμών:

Specifier	Line Style
-	Solid line (default)
--	Dashed line
:	Dotted line
-.	Dash-dot line

```
plot(x, y, 'm-') % συνεχής γραμμή
```

```
plot(x, 2*y, 'b--') % διακεκομμένη
```

```
plot(x, 2*y, 'r:') % με κουκίδες
```

```
plot(x, 2*y, 'g-.') % εναλλαγή κουκίδας με παύλα
```

```
plot(x, y, 'm-', 'LineWidth',3) % καθορισμός του πάχους της γραμμής
```

Η εντολή subplot

Σύνταξη:

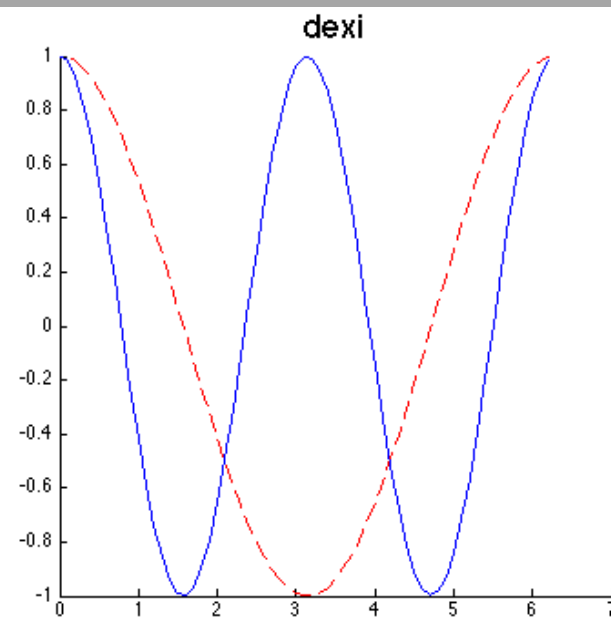
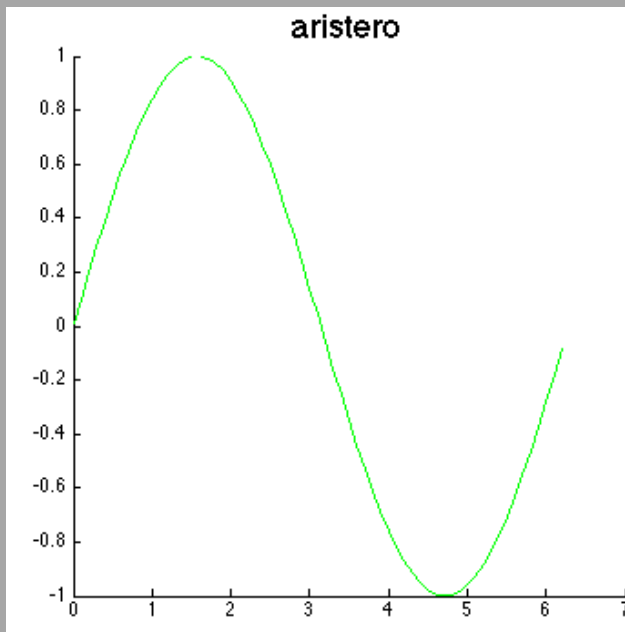
subplot(m,n,p)

χωρίζει το παράθυρο του γραφήματος σε έναν πίνακα $m \times n$ θέσεων και τοποθετεί το κάθε αντικείμενο στη θέση p

```
figure(3)
```

```
x = 0:0.1:2*pi;  
subplot(1,2,1)  
hold on  
title('aristero', 'FontSize', 18)  
plot(x, sin(x), 'g')
```

```
subplot(1,2,2)  
title('dexi', 'FontSize', 18)  
hold on  
plot(x, cos(x), 'r--')  
plot(x, cos(2*x), 'b-')
```



Η εντολή subplot

δοκιμάστε επίσης τις εντολές:

- box on
- grid on

Ερωτήσεις...