



# ΕΙΣΑΓΩΓΗ ΣΤΟ MATLAB

Κολοβού Αθανασία

Ε.Τ.Ε.Π.

<http://users.uoa.gr/~akolonou/>

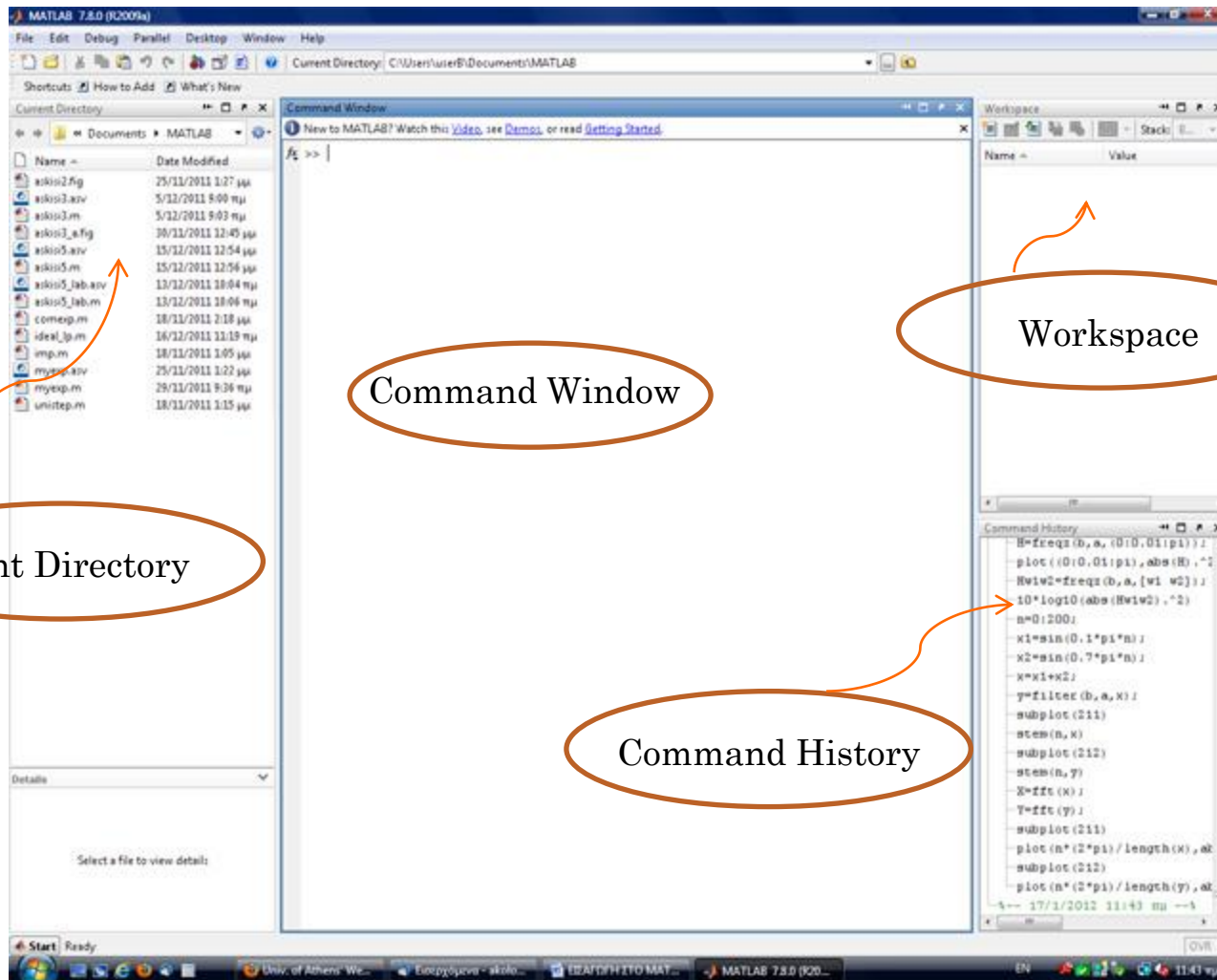
# MATRIX LABORATORY

- Μαθηματικό λογισμικό πακέτο
- Everything is a matrix...
- Εύκολο να ορίσουμε τους πίνακες

```
>> A = [16 3; 5 10]
A =
    16     3
     5    10
```



# MATLAB DESKTOP



# MATLAB DESKTOP

- **Παράθυρο εντολών (Command Window)** για να εισάγουμε, εκτελέσουμε και να δούμε τα αποτελέσματα των εντολών.
  - Χρησιμοποιούμε τα πλήκτρα ↓ ↑ για να εμφανιστεί μία προηγούμενη εντολή
- **Ιστορικό εντολών ( Command History )** μας εμφανίζει τις πιο πρόσφατες εντολές που έχουμε εισάγει στο Command Window.
- **Χώρος εργασίας (Workspace)** μας δείχνει πληροφορίες για τις μεταβλητές που έχουμε δηλώσει.
  - Το Matlab θυμάται παλαιότερες τιμές και μεταβλητές
  - `clear` διαγράφει όλες τις μεταβλητές από το χώρο εργασίας (Workspace)
  - `clc` καθαρίζει το Command Window, δεν διαγράφει μεταβλητές
  - Με διπλό click σε μία μεταβλητή εμφανίζεται στον Variable Editor



# MATLAB DESKTOP

- Variable editor

The image shows two windows from the MATLAB desktop. The left window is the 'Variable Editor - y', which displays a grid for editing the values of a 100x1 double array named 'y'. The grid shows values for rows 1 through 18. The right window is the 'Workspace', which lists variables and their properties.

Name	Value	Min	Max
ans	0.5600	0.5600	0.5600
h	[21,16,19,15,16,13]	13	21
i	12	12	12
roll	100	100	100
x	<1x100 double>	1	6
x1	<100x1 double>	1	6
x2	<100x1 double>	1	6
xbin	[1,2,3,4,5,6]	1	6
y	<100x1 double>	2	12
z	<1x12 double>	0	0.1900



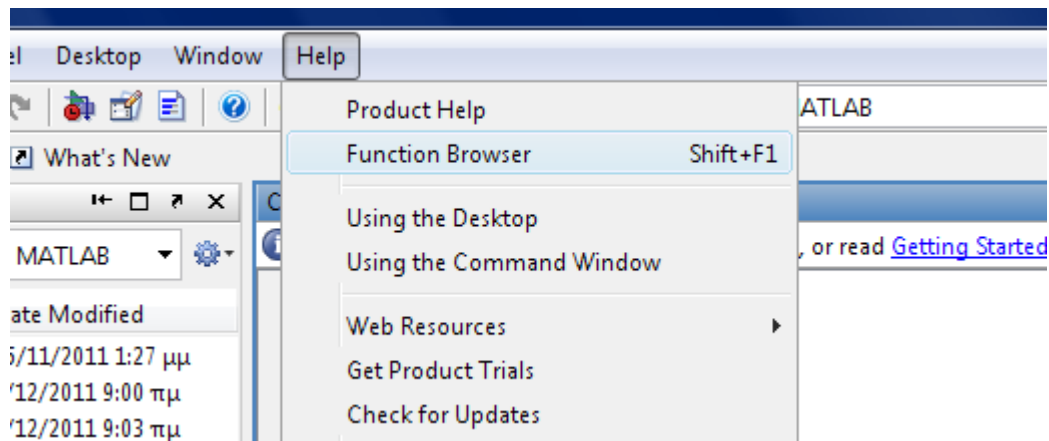
# FILE TYPES

- \*.m , χρησιμοποιείται για να αποθηκεύσουμε τον κώδικά μας ,μια σειρά εντολών που συνθέτουν μια λειτουργία
- \*.mat , binary MATLAB format files
- \*.fig , Matlab figure format
- Μπορούμε να χρησιμοποιήσουμε προκαθορισμένες συναρτήσεις η να γράψουμε δικές μας



# HELP MANUAL

- Από τη γραμμή εντολών, απλά γράφοντας help και το όνομα της συνάρτησης, πχ help cos μας δίνει πληροφορίες για την cosine συνάρτηση.
- Function Browser



# ΔΙΑΧΕΙΡΙΣΗ ΠΙΝΑΚΩΝ

- Ορισμός πίνακα

```
>>B=[3 5;9 7]
```

```
B =
```

```
     3     5  
     9     7
```

- Ορισμός διανύσματος

```
>>x = [1 2 5 1]
```



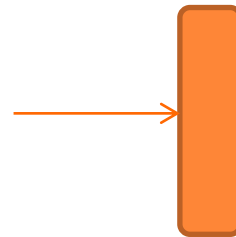
```
x =
```

```
     1     2     5     1
```

- Ανάστροφος  $y=x'$

```
y =
```

```
     1  
     2  
     5  
     1
```





# ΔΙΑΧΕΙΡΙΣΗ ΠΙΝΑΚΩΝ

- Δημιουργία πινάκων από συναρτήσεις

- `zeros(M,N)` ,  $M \times N$  πίνακας με μηδενικά

Αριθμός γραμμών

Αριθμός στηλών

```
>>M = zeros(3,2)
M =
    0    0
    0    0
    0    0
```

- `ones(M,N)`  $M \times N$  πίνακας με 1

```
>>M = ones(3,2)
M =
    1    1
    1    1
    1    1
```

- `rand(M,N)`  $M \times N$  πίνακας από ομοιόμορφα κατανομημένους τυχαίους αριθμούς στο διάστημα (0,1)

```
>>M = rand(3,2)

M =
    0.8147    0.9134
    0.9058    0.6324
    0.1270    0.0975
```



# ΔΙΑΧΕΙΡΙΣΗ ΠΙΝΑΚΩΝ

- Εξαγωγή στοιχείου  $x(i,j)$

```
>>x = [1 2 3; 5 1 4; 3 2 -1]
```

```
x =
```

```
     1     2     3
     5     1     4
     3     2    -1
```

```
>> y=x(2,3)
```

```
y =
```

```
     4
```

- Εξαγωγή ολόκληρης γραμμής

```
>> y=x(3,:)
```

```
y =
```

```
     3     2    -1
```

- Εξαγωγή ολόκληρης στήλης

```
>> y=x(:,2)
```

```
y =
```

```
     2
     1
     2
```



# ΔΙΑΧΕΙΡΙΣΗ ΠΙΝΑΚΩΝ

- Ορισμός πινάκων με βήμα

$$A=[a_{start}:b:a_{end}]$$

- Αν το βήμα  $b$  είναι ίσο με τη μονάδα, τότε αυτό μπορεί να παραλειφθεί

$$A=[a_{start}:a_{end}]$$

- Θα κατασκευάσουμε το  $A = (-1, 0, 1, 2, 3, 4, 5, 6, 7)$  με δύο τρόπους:

```
>>u=[-1:1:7]
```

```
u =  -1      0      1      2      3      4      5      6      7
```

```
>>u=[-1:7]
```

```
u =  -1      0      1      2      3      4      5      6      7
```



# ΔΙΑΧΕΙΡΙΣΗ ΠΙΝΑΚΩΝ

- Η ίδια λογική μπορεί να χρησιμοποιηθεί για την κατασκευή πινάκων

```
A=[1:10;10:-1:1]
```

```
A =
```

```
     1     2     3     4     5     6     7     8     9    10
    10     9     8     7     6     5     4     3     2     1
```

- Μπορούμε να χρησιμοποιήσουμε και την εντολή `linspace`

```
a=linspace(0,12,5) % Ξεκινώντας από το 0 , μέχρι το 12, 5 τιμές
```

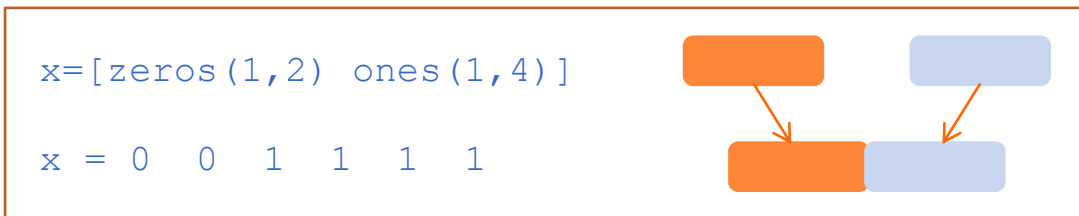
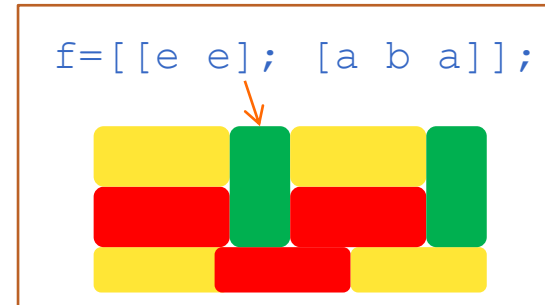
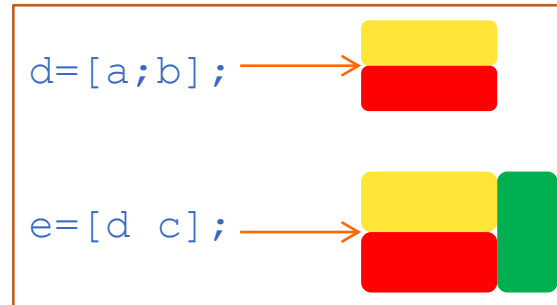
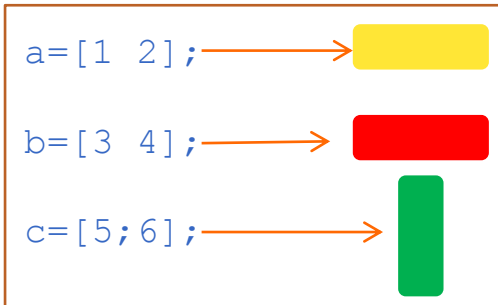
```
a =
```

```
     0     3     6     9    12
```



# ΔΙΑΧΕΙΡΙΣΗ ΠΙΝΑΚΩΝ

- Συνένωση (concatenation) πινάκων με το σύμβολο [ ]



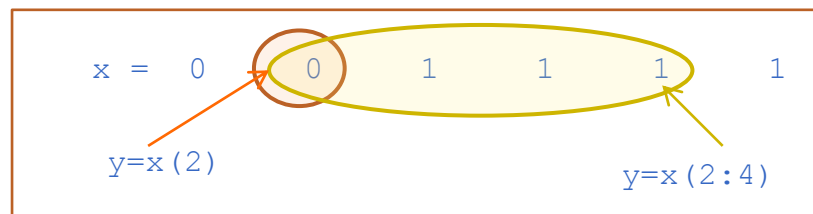
- Επιλογή υπο-πίνακα (subscription) με το σύμβολο ( )

```
y = x(2)
```

```
y = 0
```

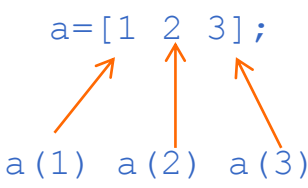
```
y = x(2:4)
```

```
y = 0 1 1 1
```



**Indexing starts with 1 not 0!!**

```
a=[1 2 3];
```



$a(1)$   $a(2)$   $a(3)$



# ΟΙ ΕΝΤΟΛΕΣ WHO ΚΑΙ WHOS

## ○ Εντολή who

```
>> who
```

Your variables are:

```
A B E M
```

## ○ Εντολή whos VariableName

```
>> whos A
```

Name	Size	Bytes	Class	Attributes
A	1x5	40	double	



# ΒΑΣΙΚΕΣ ΠΡΑΞΕΙΣ

- Το MATLAB μπορεί να χρησιμοποιηθεί σαν απλή αριθμομηχανή. Για τις βασικές πράξεις χρησιμοποιούνται τα σύμβολα που φαίνονται στον πιο κάτω πίνακα:

ΣΥΜΒΟΛΟ	ΠΡΑΞΗ
+	Πρόσθεση
-	Αφαίρεση
*	Πολλαπλασιασμός
/	Διαίρεση
^	Ύψωση σε δύναμη

```
>>3/5          ans = 0.6000
>>5^4          ans = 625
>>18/1.25      ans = 14.4000
```



# ΒΑΣΙΚΕΣ ΠΡΑΞΕΙΣ

- Υπάρχει η δυνατότητα μία πράξη όπως είναι ο πολλαπλασιασμός ή η διαίρεση να γίνεται κατά τα στοιχεία του πίνακα ένα προς ένα αρκεί πριν από το σύμβολο της πράξης να υπάρχει η τελεία.

ΣΥΜΒΟΛΟ	ΠΡΑΞΗ
.*	Πολλαπλασιασμός στοιχείο-στοιχείο
./	Διαίρεση στοιχείο-στοιχείο
.^	Ύψωση σε δύναμη στοιχείο-στοιχείο

```
T=[1:7]
T = 1     2     3     4     5     6     7
```

Ύψωνουμε τώρα κάθε στοιχείο του T στο τετράγωνο με την εντολή T.^2 και μετά με την ισοδύναμη εντολή T.\*T

```
a=T.^2
```

```
a = 1     4     9     16    25    36    49
```

```
b=T.*T
```

```
b = 1     4     9     16    25    36    49
```





# ΒΑΣΙΚΕΣ ΠΡΑΞΕΙΣ

- Πράξεις μεταξύ δύο διανυσμάτων

```
a=[1 2 3 4];  
b=[8 6 2 4];
```

```
a.*b
```

```
ans = 8 12 6 16
```

```
a./b
```

```
ans = 0.1250 0.3333 1.5000 1.0000
```

```
a.^b
```

```
ans = 1 64 9 256
```

$$\begin{bmatrix} 1 & 2 & 5 \end{bmatrix} .* \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \text{error}$$
$$\begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix} .* \begin{bmatrix} 4 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \\ 5 \end{bmatrix}$$

- Στην πρόσθεση και την αφαίρεση τα διανύσματα πρέπει να έχουν ίδιο μέγεθος

$$\begin{array}{r} [3 \ 4 \ 5 \ 8] \\ + [2 \ 6 \ 1 \ 5] \\ \hline = [5 \ 10 \ 6 \ 13] \end{array}$$

$$\begin{bmatrix} 3 \\ -1 \\ 5 \\ 9 \end{bmatrix} - \begin{bmatrix} 2 \\ 0 \\ 4 \\ 7 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ 1 \\ 2 \end{bmatrix}$$

- Η πράξη

`c=row+column` θα έδινε λάθος

- Χρησιμοποιούμε το ' (transpose) για να κάνουμε τα μεγέθη συμβατά

`c=row+column'`

`c=row'+column`



# ΟΙ ΕΝΤΟΛΕΣ LENGTH, SIZE

## ○ Η εντολή size

```
a=zeros(2,4)
[m,n]=size(a)
```

```
m =      2
n =      4
```

## ○ Η εντολή length

```
b=zeros(1,5);
length(b)
ans =
      5
```

```
length(a) %max(size(a))
ans =
      4
```



# ΣΥΝΑΡΤΗΣΕΙΣ ΒΙΒΛΙΟΘΗΚΗΣ

- Το Matlab περιέχει πολλές έτοιμες συναρτήσεις βιβλιοθήκης. Μερικές από αυτές περιγράφονται στον παρακάτω πίνακα.

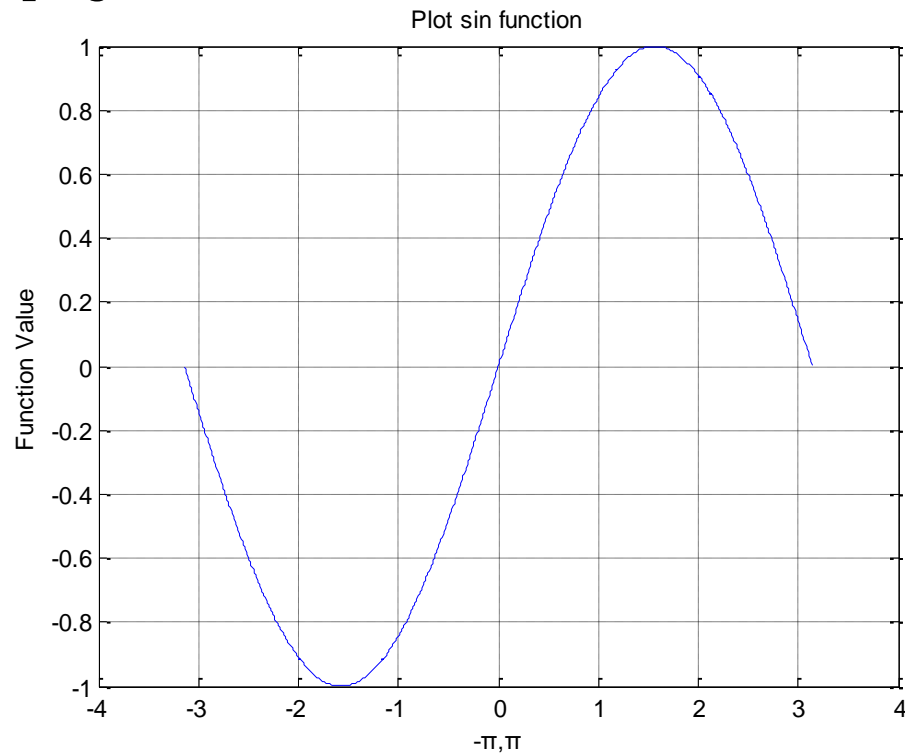
Όνομα συνάρτησης	Είδος-Περιγραφή
<b>sin,cos,tan</b>	Τριγωνομετρικές
<b>exp,log,log2,log10,sqrt</b>	Εκθετικές
<b>abs</b>	Η απόλυτη τιμή
<b>angle</b>	Γωνία
<b>real</b>	Το πραγματικό μέρος μιγαδικού αριθμού
<b>imag</b>	Το φανταστικό μέρος του μιγαδικού αριθμού
<b>fix</b>	Στρογγύλευση προς το μηδέν
<b>floor</b>	Στρογγύλευση προς το μείον άπειρο
<b>ceil</b>	Στρογγύλευση προς το συν άπειρο
<b>round</b>	Στρογγύλευση στον πλησιέστερο ακέραιο
<b>max , min</b>	Το μεγαλύτερο (η το μικρότερο) στοιχείο ενός πίνακα
<b>mean</b>	Μέση τιμή
<b>sort</b>	Ταξινόμηση
<b>sum</b>	Άθροισμα στοιχείων σε ένα πίνακα



# MATLAB GRAPHICS

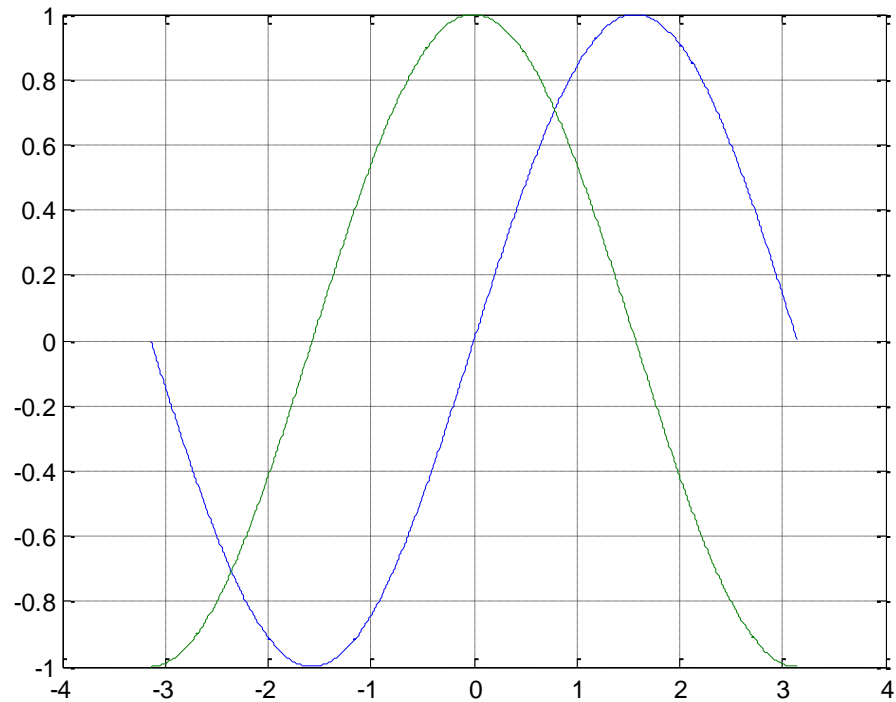
- Η εντολή plot
- Χρησιμοποιούμε τις εντολές `title`, `xlabel`, `ylabel` και `legend` στα γραφήματα μας

```
x = -pi:0.01:pi;  
plot(x,sin(x)), grid on  
title('Plot sin function');  
xlabel('-π,π');
```



# MULTIPLE GRAPHS

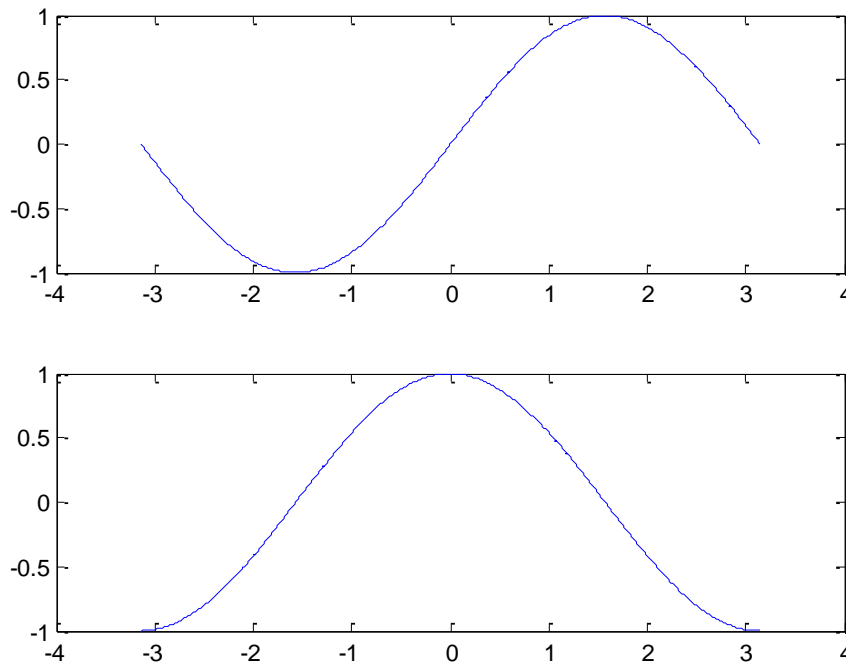
```
x = -pi:0.01:pi;  
Y=sin(x);  
A=sin(x+pi/2);  
plot(x,Y,x,A);grid on
```



# MULTIPLE GRAPHS

- Με την εντολή `subplot(mnp)` χωρίζουμε ένα γράφημα σε  $m \times n$  θέσεις και επιλέγουμε κάθε φορά την  $p$  θέση για να τοποθετήσουμε το τρέχον plot

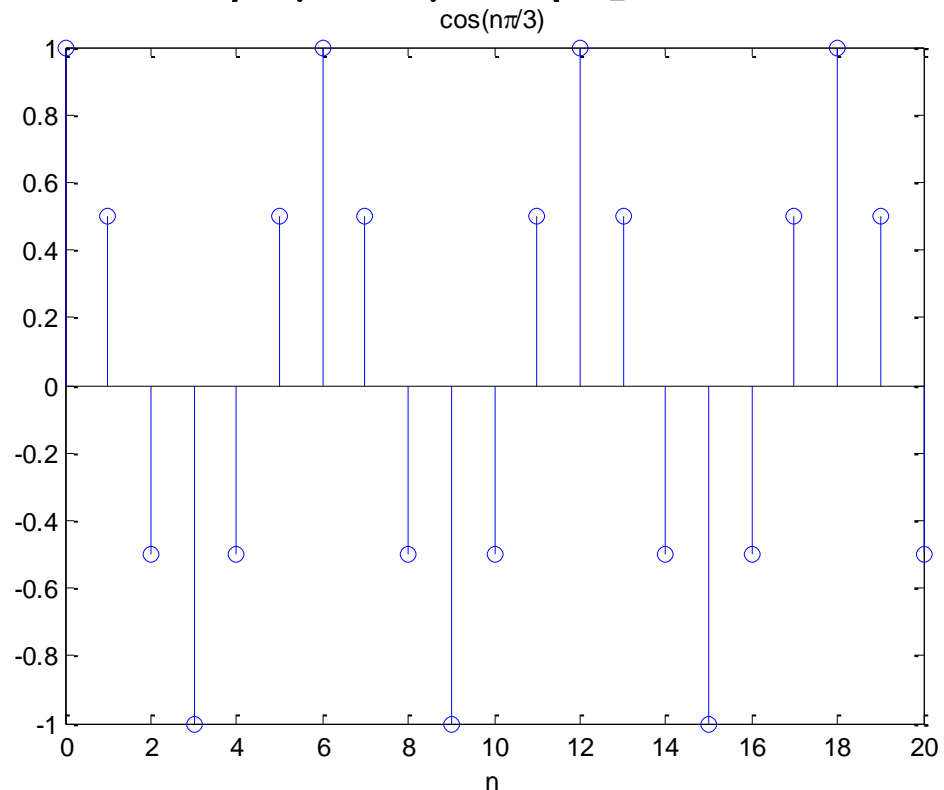
```
x = -pi:0.01:pi;  
Y=sin(x);  
A=sin(x+pi/2);  
subplot(211);plot(x,Y)  
  
subplot(212);plot(x,A)
```



# STEM()

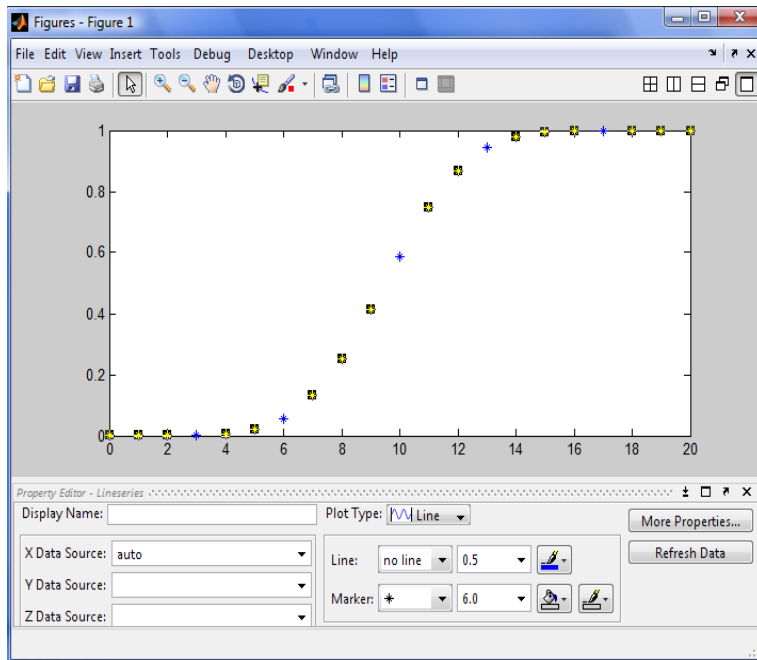
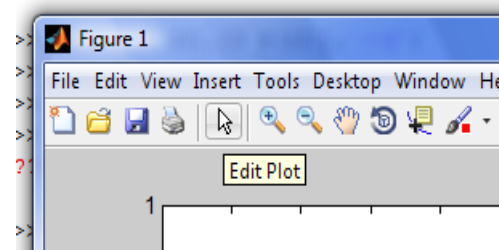
- Η εντολή `stem()` χρησιμοποιείται για να σχεδιάσουμε διακριτά δεδομένα
- Η χρήση της `stem()` είναι παρόμοια με την `plot()`

```
n = 0:20;  
x = cos(pi*n/3);  
stem(n, x)  
title('cos(n*pi/3)')  
xlabel('n')
```



# MATLAB GRAPHICS

- Το Matlab για την αποδοτικότερη επεξεργασία των γραφικών παρέχει το interactive plotting environment ή *plotting tools* διαθέσιμα αν σε ένα γράφημα πατήσουμε το Edit Plot και κάνουμε διπλό click πάνω στη γραφική
- Υπάρχει και η εντολή *plottools*





# M-FILES

- Στο Matlab μπορούμε να εκτελούμε μία σειρά από εντολές οι οποίες βρίσκονται αποθηκευμένες σε αρχεία τύπου \*.m
- Υπάρχουν δύο είδη τέτοιων αρχείων τα *script files* και τα *function files*.
- Ένα *script file* περιλαμβάνει μία σειρά από εντολές matlab. Αν για παράδειγμα το αρχείο λέγεται roll.m , τότε η εντολή roll θα προκαλέσει την εκτέλεση όλων των εντολών του αρχείου.
- Τα *function files* παρέχουν επεκτασιμότητα στο matlab. Μπορούμε να δημιουργήσουμε δικές μας συναρτήσεις οι οποίες θα έχουν το ίδιο status με τις υπόλοιπες built-in συναρτήσεις.



# M-FILES

Πρέπει να ξεκινάει με τη λέξη `function`

Όνομα της συνάρτησης, ίδιο με το όνομα του m file

```
function [a,b,c]=FunName(input1,input2)
```

Μεταβλητές εξόδου (αν είναι περισσότερες από μία πρέπει να είναι μέσα σε [])

Μεταβλητές εισόδου

## ○ Παράδειγμα

```
function a=randint(m,n)

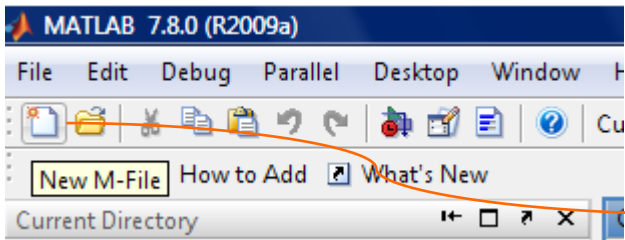
%This function generates a mxn
%matrix of random integers
%between 0 and 9

a=floor(10*rand(m,n));
```

```
a=randint(5,3)

a =

     8     0     1
     9     2     9
     1     5     9
     9     9     4
     6     9     8
```



Εδώ click για να ανοίξουμε τον editor όπου γράφουμε το αρχείο.



# FLOW CONTROL

## ○ Η εντολή *for*



## ○ Η εντολή *if*

Μπορούμε να χρησιμοποιήσουμε

$\geq$ ,  $\leq$ ,  $<$ ,  $>$ ,  $==$ ,  $\sim$

```
if (a <= 2)  
    b = 1;  
elseif (a >=4)  
    b = 2;  
else  
    b = 3;  
end
```



# FLOW CONTROL

- Η εντολή *while*

```
while cond  
    commands  
end
```

- Οι εντολές στο `command block` εκτελούνται όσο η έκφραση `cond` είναι `true`.
- Προσοχή στα `infinite loops!!!`



# ΠΑΡΑΔΕΙΓΜΑ

- Έχω την εντολή `x=sin(linspace(0,10*pi,100));`  
Πόσες τιμές του x είναι θετικές ?

Με χρήση <code>loop</code>	Με χρήση της <code>find</code>
<pre>count=0; for n=1:length(x)     if x(n)&gt;0         count=count+1;     end end  count = 49</pre>	<pre>count=length(find(x&gt;0));  count=49</pre>

- Καλύτερα να αποφεύγουμε τα loops!



# ΠΗΓΕΣ

- <http://www.math.toronto.edu/mpugh/primer.pdf>
- <http://www.mathworks.com/>
- [http://courses.washington.edu/css457/matlab/learning\\_matlab.pdf](http://courses.washington.edu/css457/matlab/learning_matlab.pdf)
- <http://www-h.eng.cam.ac.uk/help/documentation/docsource/matlab1.pdf>
- <http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-094-introduction-to-matlab-january-iap-2010/>
- [http://software-carpentry.org/4\\_0/matlab/](http://software-carpentry.org/4_0/matlab/)

