


## ΤΟ ΔΙΑΔΡΑΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ ΕΡΓΑΣΙΑΣ ΤΟΥ MATLAB

### Έναρξη - Τερματισμός

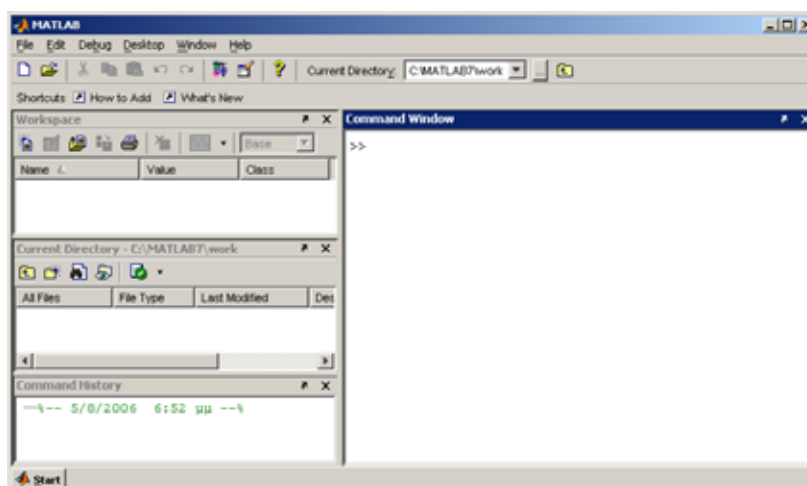
Η έναρξη της λειτουργίας του MatLab εξαρτάται από το λειτουργικό σύστημα.

Στα συστήματα UNIX πληκτρολογούμε στη προτροπή του συστήματος (%): **matlab**

Στα συστήματα WINDOWS η έναρξη γίνεται:

- από το εικονίδιο MatLab:  στην επιφάνεια εργασίας,
- από το μενού: **Έναρξη > Όλα τα Προγράμματα > MATLAB**, ή,
- πληκτρολογώντας στη προτροπή του DOS (>): **matlab**.

Το κύριο παράθυρο της εφαρμογής περιέχει πολλά επί μέρους παράθυρα. Για παράδειγμα, στην παρακάτω εικόνα βλέπουμε τέσσερα από αυτά: Το **Command Window** (παράθυρο εντολών - CW) είναι το βασικό παράθυρο όπου δίνουμε όλες τις εντολές και παίρνουμε τα αποτελέσματα, το Command History περιέχει το ιστορικό των εντολών που δώσαμε, το Workspace (περιοχή εργασίας) περιέχει τα ονόματα και τις τιμές όλων των μεταβλητών που έχουμε χρησιμοποιήσει και το Current Directory παρουσιάζει τα περιεχόμενα του φακέλου μέσα στον οποίο αναπτύσσουμε την εφαρμογή μας. Υπάρχουν και άλλα βοηθητικά παράθυρα, π.χ., για τις γραφικές παραστάσεις, για την εμφάνιση των πινάκων, ο editor των προγραμμάτων, κ.ά., τα οποία θα παρουσιάσουμε παρακάτω.



Όταν το MatLab ξεκινά έχει σαν αρχικό φάσκειλο εργασίας τον **c:\matlab\work**, και, εφόσον υπάρχει εκεί, εκτελεί το αρχείο **startup.m**. Αν δεν υπάρχει θα ελέγξει και στο φάκελο **c:\matlab\toolbox\local**. Με αυτό τον τρόπο το MatLab μας δίνει τη δυνατότητα να ρυθμίσουμε την εκκίνηση του, τοποθετώντας στο αρχείο **startup.m** τις εντολές που επιθυμούμε να εκτελούνται πρώτες.

Ο **τερματισμός** της εφαρμογής MatLab γίνεται είτε από το μενού: **File > Exit MATLAB**, ή, δίνοντας στο Command Window την εντολή **exit** ή **quit**:

```
Command Window
>> exit
ή
>> quit
```

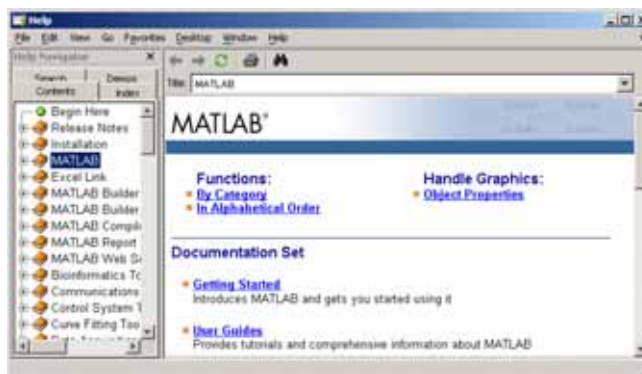
Η εντολή **exit** τερματίζει αμέσως την εφαρμογή και ισοδυναμεί με το πλήκτρο X στην επάνω δεξιά γωνία του παράθυρου της εφαρμογής.

Αντίθετα η **quit** είναι πιο σταδιακή καθώς εκτελεί πρώτα ένα αρχείο με εντολές τερματισμού, το **finish.m**, στο οποίο μπορούμε να τοποθετήσουμε εντολές οι οποίες αποθηκεύουν την περιοχή εργασίας μας με όλες τις τρέχουσες τιμές των μεταβλητών, κλείνουν τα ανοιχτά αρχεία, ολοκληρώνουν τα γραφικά, και γενικά εξασφαλίζουν την ομαλή συνέχεια της εργασίας μας την επόμενη φορά που θα χρησιμοποιήσουμε το MatLab.

Ένας συχνός συνδυασμός των δύο αρχείων είναι, το μεν **finish.m** να έχει την εντολή **save** που αποθηκεύει όλες τις τρέχουσες μεταβλητές από τη μνήμη στο αρχείο **matlab.mat**, το δε **startup.m** να έχει την εντολή **load** η οποία φορτώνει το αρχείο **matlab.mat** στη μνήμη.

## Βοήθεια & Πληροφορίες

Για οποιαδήποτε **βοήθεια** ή διευκρίνιση το MatLab διαθέτει όλες τις απαραίτητες οδηγίες παραδείγματα και πληροφορίες κάτω από το μενού: **Help > MatLab Help (F1)**



Επίσης, στο Command Window η εντολή **help** μας δίνει σαν απάντηση μια λίστα με όλες τις διαθέσιμες εντολές της εφαρμογής.:

```
Command Window
>> help
HELP topics

matlab\general      - General purpose commands.
matlab\ops          - Operators and special characters.
matlab\lang         - Programming language constructs.
matlab\elmat       - Elementary matrices and matrix
manipulation.
matlab\elfun        - Elementary math functions.
matlab\specfun      - Specialized math functions.
matlab\matfun       - Matrix functions - numerical linear
algebra.
... ..
```

Για να ζητήσουμε βοήθεια για μια συγκεκριμένη εντολή γράφουμε αμέσως μετά την εντολή help το όνομά της εντολής:

```
Command Window
>> help exit
EXIT Exit from MATLAB.
EXIT terminates MATLAB.

See also quit.
>>
```

Το MatLab διαθέτει Επιπλέον διάφορα παραδείγματα για κάθε κατηγορία προβλήματος, προσβάσιμα από το μενού: **Help > Demos**, ή, στο Command Window με την εντολή **demo**:

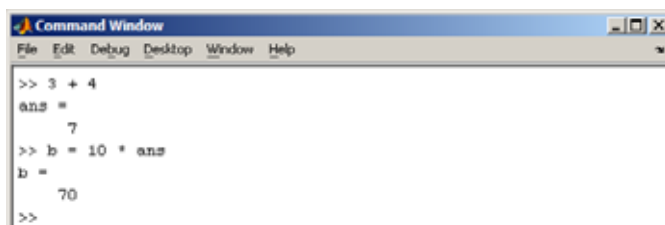
```
Command Window
>> demo
```



## Τα Παράθυρα του Περιβάλλοντος Εργασίας

### Το Παράθυρο Εντολών (Command Window)

Το Command Window (ή **CW** για συντομία) είναι το κύριο μέσο επικοινωνίας του χρήστη με την εφαρμογή MatLab. Εδώ δίνονται οι εντολές και επιστρέφονται όσα αποτελέσματα δεν απαιτούν ειδικό παράθυρο γραφικών.



```
Command Window
File Edit Debug Desktop Window Help
>> 3 + 4
ans =
     7
>> b = 10 * ans
b =
    70
>>
```

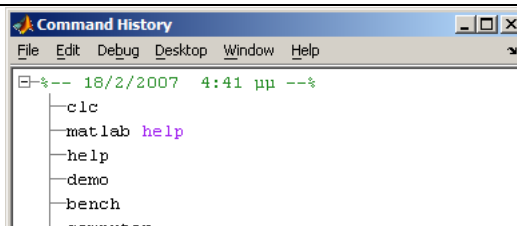
Το Command Window (CW) χρησιμοποιεί το σύμβολο (`>>`) για να δείξει στον χρήστη ότι περιμένει εντολές (προτροπή - prompt). Για να είναι πιο ευδιάκριτες οι εντολές που θα δίνει ο χρήστης (`↵`) και τα αποτελέσματα που θα παίρνει, στο βιβλίο αυτό το CW παρουσιάζεται πάντοτε με την παρακάτω μορφή:



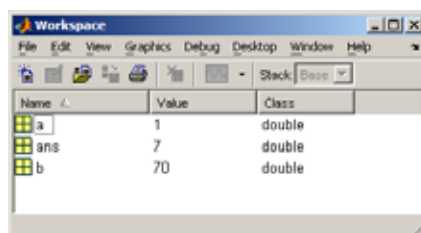
```
Command Window
>> 3 + 4 ↵
ans =
     7
>> b = 10 * ans ↵
b =
    70
>>
>> a = sin(pi/2) ↵
a =
     1
>>
```

### Το Ιστορικό των Εντολών (Command History)

Όλες οι εντολές που δίνουμε στο CW αποθηκεύονται σε μια λίστα με τη σειρά που δόθηκαν. Από το παράθυρο του CW έχουμε πρόσβαση σε όλες τις προηγούμενες εντολές με τα βέλη του πληκτρολογίου (άνω - κάτω). Η λίστα με το ιστορικό των εντολών περιέχεται και στο παράθυρο Command History όπου μπορούμε να επιλέξουμε μια εντολή ή ένα σύνολο εντολών για εκτέλεση ή για αποθήκευση σε αρχείο, κλπ.



### Η Περιοχή Εργασίας (Workspace)



Το παράθυρο Workspace περιέχει όλες τις μεταβλητές και τα μητρώα που χρησιμοποιούμε στην εργασία μας και μας πληροφορεί για τις ιδιότητές τους όπως για παράδειγμα την τιμή, την ακρίβεια, τις διαστάσεις, κλπ. Επίσης, διαθέτει μενού και εντολές για τη διαγραφή, την εμφάνιση, τη σχεδίαση, ή και την εκτύπωσή τους. Τα περιεχόμενα του Workspace εμφανίζονται και στο CW με τις εντολές **who** & **whos**:

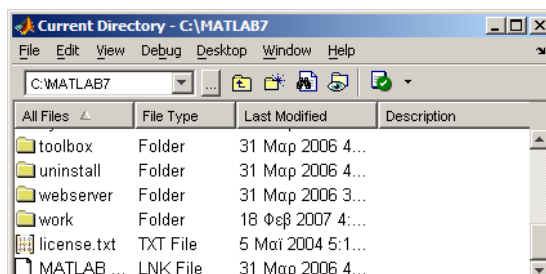
```
Command Window
>> who
Your variables are:
a    ans  b

>> whos
  Name      Size      Bytes  Class

  a         1x1         8  double array
  ans       1x1         8  double array
  b         1x1         8  double array

Grand total is 3 elements using 24 bytes
>>
```

### Το Παράθυρο Φακέλων και Αρχείων (*Current Directory*)

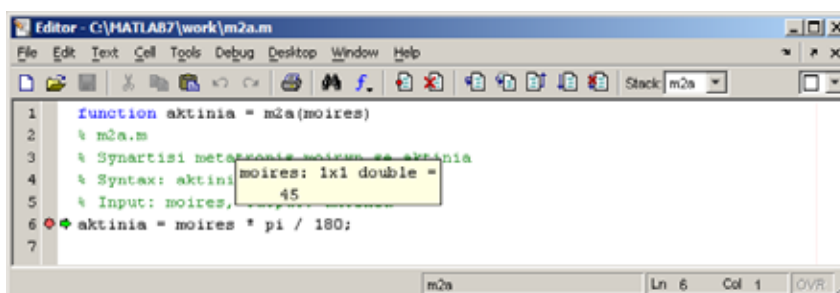


Το παράθυρο *Current Directory* δείχνει τα περιεχόμενα του φακέλου στον οποίο εργαζόμαστε και στον οποίο θα αποθηκευτούν τα αρχεία μας, τα δεδομένα, τα αποτελέσματα και ότι άλλο αποτελεί το τρέχον έργο μας. Τα περιεχόμενα του *Current Directory* εμφανίζονται και στο *CW* με τις εντολές **dir** & **ls**.

Command Window

```
>> dir
.          help          rtw          webservice
..         ja           simulink     work
MATLAB 7.0.lnk  java          stateflow
bin        jhelp         sys
demos     license.txt   toolbox
extern    notebook     uninstall
>>
```

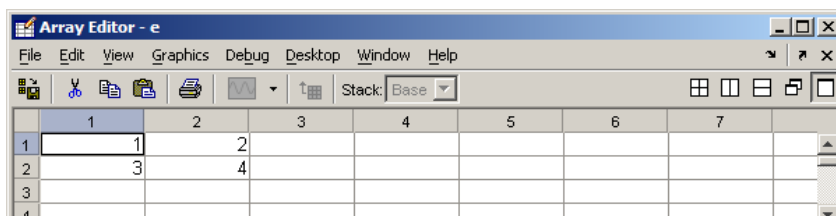
### Ο Επεξεργαστής Κειμένου & Προγραμμάτων (*Editor/Debugger*)



Αν και όλοι οι Η/Υ διαθέτουν κάποιον επεξεργαστή κειμένου, το *MatLab* έρχεται με τον δικό του *Editor* ο οποίος παρέχει στον χρήστη πολλές επιπλέον ευκολίες για την ανάπτυξη των

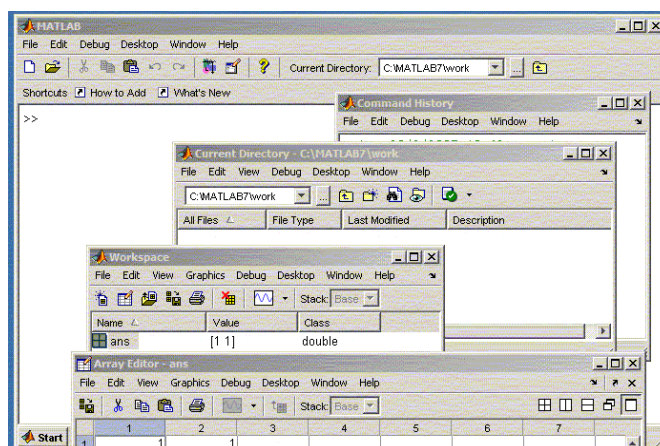
προγραμμάτων του (.m αρχεία) και λειτουργεί και σαν debugger. Έτσι, μπορούμε να παρακολουθήσουμε την ροή εκτέλεσης του προγράμματος και να πληροφορηθούμε τις τιμές που παίρνουν οι μεταβλητές.

### Ο Επεξεργαστής Μητρώων (Array Editor)



Το MatLab κάνει εκτεταμένη χρήση των μητρώων γι' αυτό και διαθέτει ειδικό παράθυρο για την εμφάνιση και την επεξεργασία τους. Ο επεξεργαστής μητρώων ενεργοποιείται από τη περιοχή εργασίας αυτόματα μόλις επιλέξουμε το μητρώο που μας ενδιαφέρει. Επιπλέον, ο Array Editor μπορεί να παρουσιάσει ένα μητρώο με 2-D ή 3-D γραφικά.

Όλα τα παράθυρα του MatLab μπορούν να αποσπαστούν από το κεντρικό παράθυρο της εφαρμογής και να τοποθετηθούν οπουδήποτε στην επιφάνεια εργασίας.



Στη συνέχεια του βιβλίου, και στο μεγαλύτερο μέρος του, θα χρησιμοποιούμε κυρίως το Command Window (CW), στο οποίο δίνουμε τις εντολές και παίρνουμε τα περισσότερα αποτελέσματα, και τον **Editor**, στον οποίο συντάσσουμε τα προγράμματα μας. Για το λόγο αυτό θα παρουσιάσουμε τη λειτουργία τους με περισσότερη λεπτομέρεια στο τμήμα που ακολουθεί.

## Βασική Χρήση και Ρυθμίσεις του Command Window

Η πραγματοποίηση απλών υπολογισμών στο MatLab είναι πολύ εύκολη. Το CW λειτουργεί σαν μεταφραστής (Interpreter) που μετατρέπει κάθε μας εντολή σε γλώσσα μηχανής, την εκτελεί, και αμέσως κάτω από την εντολή γράφει το αποτέλεσμα.

### Οι Βασικές Πράξεις (+), (-), (\*), (/) & (^)

Οι πράξεις με αριθμούς γράφονται όπως σε οποιαδήποτε άλλη γλώσσα προγραμματισμού και εκτελούνται άμεσα. Το αποτέλεσμα (ans =) δίνεται αμέσως από κάτω από την εντολή.

Command Window:

```
>> 2+3*5^2
ans =
    77
>>
```

Η σειρά εκτέλεσης των πράξεων είναι η ίδια σε όλες τις γλώσσες προγραμματισμού, δηλαδή, πρώτα γίνεται η ύψωση σε δύναμη (^), μετά οι πολλαπλασιασμοί (\*) και οι διαιρέσεις (/), και τέλος, οι προσθέσεις (+) και οι αφαιρέσεις (-). Η σειρά μπορεί να αλλάξει με τη χρήση παρενθέσεων οι οποίες προηγούνται αρχίζοντας πρώτα από τη πιο εσωτερική.

Για παράδειγμα η παράσταση:  $1.33 \cdot \left(1 + \frac{43 - 6^2}{11.2 \cdot (2 + 3/4)}\right) \cdot 10^3$ , δίνεται ως εξής:

Command Window:

```
>> 1.33*(1+(43-6^2)/(11.2*(2+3/4)))*10^3
ans =
 1.6323e+003
>>
```

### Τα Ειδικά Σύμβολα (...), (;) & (:)

Αν μια εντολή δεν χωράει σε μια γραμμή, μπορεί να συνεχίσει στις επόμενες εφόσον στο τέλος κάθε μιας τοποθετηθούν αποσιωπητικά (...).

Command Window:

```
>> 1.33*(1+(43-6^2)/ ...
(11.2*(2+3/4)))*10^3
ans =
 1.6323e+003
>>
```

Κάθε γραμμή μπορεί να περιέχει περισσότερες από μία εντολές, εφόσον αυτές διαχωρίζονται μεταξύ τους με κόμμα (,).

Command Window:

```
>> 2+3, 4^2
ans =
     5
ans =
    16
>>
```

Σε κάθε εντολή που δίνουμε στο CW, το MatLab απαντά με το αποτέλεσμα της. Αυτό δεν είναι πάντοτε επιθυμητό, ειδικά για ενδιάμεσους υπολογισμούς. Η απάντηση του MatLab (έξοδος) μπορεί να απενεργοποιηθεί με τη τοποθέτηση ενός ερωτηματικού (semicolon) (;) στο τέλος κάθε εντολής. Στη περίπτωση αυτή, παρόλο που δεν βλέπουμε το αποτέλεσμα, αποθηκεύεται στην ειδική μεταβλητή **ans** και μπορεί να χρησιμοποιηθεί στη συνέχεια των υπολογισμών.

Command Window:

```
>> 1.33*(1+(43-6^2)/(11.2*(2+3/4)))*10^3;
>>
... ..
>> ans
ans =
    1.6323e+003
```

### Οι Ειδικές Μεταβλητές

Το MatLab έχει ορισμένες ειδικές μεταβλητές όπως είναι οι: **pi** ( $\pi$ ), **eps**, **ans**, **i**, **j**, κλπ.

Η μεταβλητή **pi** περιέχει τον αριθμό  $\pi = 3.14159\dots$ . Μπορούμε να δούμε τη τιμή του αριθμού **pi** στο CW ή να τον χρησιμοποιήσουμε σε τριγωνομετρικές πράξεις:

Command Window:

```
>> pi
ans =
    3.1416
>> sin(pi/2)
ans =
     1
>> help pi
PI      3.1415926535897...
PI = 4*atan(1) = imag(log(-1)) = 3.1415926535897...
Reference page in Help browser
doc pi
```

Στο MatLab όλοι οι υπολογισμοί γίνονται με διπλή ακρίβεια, δηλαδή περίπου δεκαέξι ψηφίων για κάθε αριθμό, σε 32-bit μηχανές. Η μικρότερη διαφορά που διακρίνει δυο αριθμούς εξαρτάται από το σφάλμα στρογγυλοποίησης της συγκεκριμένης υπολογιστικής μηχανής (computer) και λέγεται **eps**.

Command Window:

```
>> computer
ans =
PCWIN
>> eps
ans =
2.2204e-016
>>
```

Όταν οι εντολές του CW εκτελούν πράξεις χωρίς να αναθέτουν (=) το αποτέλεσμα σε μια μεταβλητή, τότε το αποτέλεσμα τους αποθηκεύεται πάντα στη μεταβλητή **ans**.

Command Window:

```
>> x = 1+2 , 2+3
x =
3
ans =
5
>>
```

Το MatLab διαθέτει μιγαδικούς αριθμούς και συγκεκριμένα, εφόσον δεν έχουν οριστεί διαφορετικά, οι μεταβλητές **i** και **j** αντιστοιχούν στη τετραγωνική ρίζα του -1 (**sqrt(-1)**).

Command Window:

```
>> i
ans =
0 + 1.0000i
>> j
ans =
0 + 1.0000i
>> i^2
ans =
-1
>> help i
I Imaginary unit.
As the basic imaginary unit SQRT(-1), i and j are used to enter
complex numbers. For example, the expressions 3+2i, 3+2*i, 3+2j,
3+2*j and 3+2*sqrt(-1) all have the same value.

Since both i and j are functions, they can be overridden and used
as a variable. This permits you to use i or j as an index in FOR
loops, etc.

See also j.
>>
```

## Μεταβλητές

Στους απλούς υπολογιστές χειρός (calculators) τα αποτελέσματα εμφανίζονται στην οθόνη και όταν πραγματοποιήσουμε την επόμενη πράξη τα προηγούμενα διαγράφονται για να εμφανιστεί το νέο αποτέλεσμα.

Αυτό δεν συμβαίνει στις γλώσσες προγραμματισμού και στο MatLab. Εδώ τα αποτελέσματα των πράξεων ή των εντολών αποθηκεύονται στη μνήμη της περιοχής εργασίας μας, το κάθε ένα με άλλο όνομα, ώστε να μπορούν να ανακληθούν αργότερα. Τα ονόματα αυτά, που αναφέρονται στις θέσεις μνήμης και περιέχουν κάποια τιμή, λέγονται μεταβλητές.

Τα ονόματα των μεταβλητών αποτελούνται από Λατινικά γράμματα (**a-z, A-Z**), αριθμούς (**0-9**) και το σύμβολο (**\_**). Το μέγιστο μήκος των ονομάτων είναι 63 χαρακτήρες και δεν πρέπει να συμπίπτουν με δεσμευμένες λέξεις ή εντολές του MatLab.

Στο MatLab τα κεφαλαία γράμματα διακρίνονται από τα αντίστοιχα πεζά (case sensitive). Δηλαδή, το "y" και το "Y" είναι διαφορετικές μεταβλητές και έχουν διαφορετικές τιμές.

Η ανάθεση τιμών στις μεταβλητές γίνεται με το σύμβολο ίσον (=). Αριστερά του τοποθετείται το όνομα της μεταβλητής και δεξιά του η αλγεβρική παράσταση. Μόνο ένα όνομα μεταβλητής επιτρέπεται σε κάθε ανάθεση. Π.χ.:

Command Window:

```
>> imax = 10;
>> xronos = 0.0;
>> embado_trigwnou = 0.5 * basi * ypsos;
>> A3 = 3.0;
```

Οι μεταβλητές του MatLab δεν χρειάζονται ειδικές δηλώσεις, όπως στις γλώσσες προγραμματισμού, αλλά δημιουργούνται αυτόματα στη περιοχή εργασίας (**Workspace**) μόλις τους γίνει η πρώτη ανάθεση τιμής.

Για να γράψουμε μια πιο σύνθετη παράσταση μπορούμε να χρησιμοποιήσουμε και περισσότερες από μια ενδιάμεσες μεταβλητές. Όπως για παράδειγμα, η παρακάτω συνάρτηση μπορεί να γραφτεί:

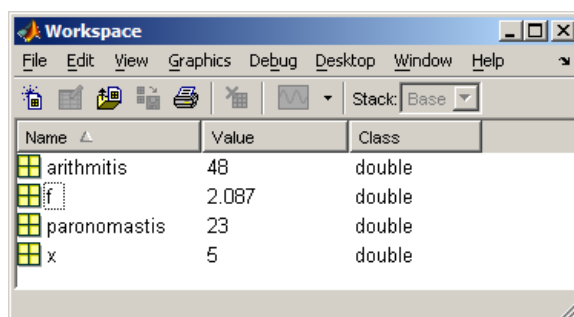
$$f = \frac{x^3 - 3x^2 + x - 7}{x^2 + 0.5x - 4.5}$$

Command Window:

```
>> x = 5;
>> arithmitis = x^3-3*x^2+x-7;
>> paronomastis = x^2+0.5*x-4.5;
>> f = arithmitis / paronomastis
f =
    2.0870
>>
```

### Η Περιοχή Εργασίας (Workspace)

Όλες οι μεταβλητές που χρησιμοποιούμε βρίσκονται στη περιοχή εργασίας (Workspace). Τα περιεχόμενά της εμφανίζονται στο παράθυρο Workspace. Για παράδειγμα το **Workspace** που δημιουργήθηκε που τις προηγούμενες εντολές θα περιέχει 4 μεταβλητές:



Η διαγραφή μιας μεταβλητής γίνεται με την εντολή **clear** και το όνομα της μεταβλητής. Η εντολή **clear** σιέτη διαγράφει όλες τις μεταβλητές από το Workspace.

Command Window:

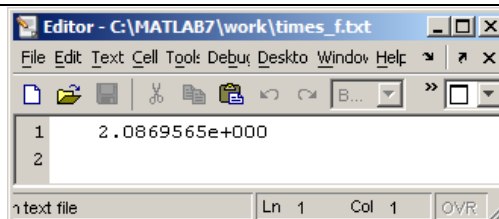
```
>> clear x
```

Οι τιμές των μεταβλητών μπορούν να εξαχθούν και να αποθηκευτούν σε αρχεία κειμένου (text) με την εντολή **save**. Για να αποθηκεύσουμε τη τιμή της *f* στο αρχείο "times\_f.txt" δίνουμε:

Command Window:



```
>> save times_f.txt f -ascii
```

και το αρχείο που δημιουργείται θα περιέχει:



Η εντολή **save** σκέτη αποθηκεύει όλο το workspace με τις μεταβλητές σε ένα αρχείο με το όνομα `matlab.mat`. Το αρχείο αυτό μπορούμε να το επαναφέρουμε αργότερα με την εντολή **load**, και να συνεχίσουμε την εργασία μας.

Command Window:



```
>> save   
  
Saving to: matlab.mat  
  
>> load   
  
Loading from: matlab.mat  
  
>>
```

### *Ρυθμίσεις*

Όταν το CW έχει πάρα πολλά στοιχεία (εντολές, αποτελέσματα, κλπ.) μπορούμε να το καθαρίσουμε με την εντολή **clc**. Αν τα στοιχεία αυτά είναι χρήσιμα μπορούμε να κρατάμε παράλληλα ένα αντίγραφο του CW σε αρχείο κειμένου ε την εντολή **diary** (on/off).

Οι γραμμές των εντολών στο CW έχουν μεταξύ τους ένα κενό ώστε να είναι πιο ευδιάκριτες αλλά αυτό επιμηδύνει τις απαντήσεις και καταλαμβάνουν πιο πολύ χώρο στο παράθυρο. Η εντολή **format compact** αφαιρεί το ενδιάμεσο κενό ενώ η εντολή **format loose** το επαναφέρει.

Command Window:

```
>> format loose   
>> pi  
  
ans =  
  
    3.1416  
  
>> format compact   
>> pi  
ans =  
    3.1416  
>>
```

Τα αριθμητικά αποτελέσματα του *MatLab* είναι διπλής ακρίβειας. Αυτό σημαίνει ότι κάθε αριθμός μπορεί να έχει και 16 σημαντικά ψηφία αλλά δεν είναι δυνατόν να εκτυπώνονται πάντοτε όλα τα ψηφία στο CW. Με την εντολή **format short** ή **format long** ρυθμίζουμε το πλήθος των εμφανιζόμενων ψηφίων που επιθυμούμε.

Command Window:

```
>> format long, a = sqrt(2)
a =
    1.41421356237310
>> format short, a = sqrt(2)
a =
    1.4142
```

Αν οι αριθμοί πρέπει να εμφανίζονται σε μορφή εκθετικού (π.χ. 1.0E+2) τότε χρησιμοποιούμε αντίστοιχα τις εντολές **format short e** ή **format long e**.

Τέλος, οποιαδήποτε εντολή ή ενέργεια που έχουμε δώσει μπορεί να ακυρωθεί ή να διακοπεί μέσα από το CW, αν δώσουμε με το πληκτρολόγιο την εντολή **^C** ή **ctrl-C**.

## ΕΠΙΛΥΣΗ ΑΠΛΩΝ ΠΡΟΒΛΗΜΑΤΩΝ ΣΤΟ MATLAB ΜΕ ΑΚΟΛΟΥΘΙΕΣ ΕΝΤΟΛΩΝ

Στο MatLab μπορούμε να λύσουμε ένα πρόβλημα χωρίς να κατασκευάσουμε απαραίτητα κάποιο πρόγραμμα. Απλά εκτελούμε τα βήματα της λύσης, όπως θα κάναμε και με το χέρι, με τη μορφή εντολών στο CW.

Το βασικό κέρδος από την απλή αυτή χρήση του MatLab είναι ότι, (α) όλα τα αποτελέσματα αποθηκεύονται στο Workspace και είναι διαθέσιμα κάθε στιγμή για να ξαναχρησιμοποιηθούν, (β) όλες οι εντολές αποθηκεύονται στο History και είναι διαθέσιμες κάθε στιγμή για να ξαναδοθούν ή να διορθωθούν, και, (γ) μπορούμε να δοκιμάσουμε εύκολα και γρήγορα πολλούς τρόπους λύσης ώστε να καταλήξουμε στο βέλτιστο τρόπο για το πρόβλημα μας.

Ακόμη και όταν αναπτύσσουμε προγράμματα στο MatLab είναι καλό, πρώτα να δοκιμάζουμε τις πιο σύνθετες εντολές τους στο CW, ώστε να είμαστε βέβαιοι για τη σωστή υλοποίησή και τα αποτελέσματά τους, και, μετά να τις ενσωματώνουμε στο πρόγραμμα.

### Απλοί υπολογισμοί και εντολές ορισμού ή ανάθεσης

#### Παράδειγμα 1: Μετατροπή Μοιρών σε Ακτίνια

Για να μετατραπουν οι μοίρες σε ακτίνια εφαρμόζεται η σχέση:

$$\frac{aktinia}{\pi} = \frac{moires}{180} \Leftrightarrow aktinia = \frac{moires \cdot \pi}{180}$$

Έστω ότι πρέπει να μετατραπουν οι 45° σε ακτίνια. Δίνουμε τη τιμή 45 σε μια μεταβλητή (*moires*) και στη συνέχεια δίνουμε τον τύπο υπολογισμού:  $aktinia = moires * \pi / 180$ . Το αποτέλεσμα θα είναι 0.7854, που ισούται με το  $\pi/4$ .

Command Window:

```
>> moires = 45
moires =
    45
>> aktinia = moires * pi / 180
aktinia =
    0.7854
>> pi/4
ans =
    0.7854
```

**Παράδειγμα 2: Μετατροπή θερμοκρασίας °C σε °F**

Ο τύπος μετατροπής από βαθμούς Κελσίου σε Φαρενάιτ είναι:  $F = 32 + C * 9/5$ .  
Αν  $C = 36.6$  τότε το  $F$  θα είναι:

Command Window:

```
>> C = 36.6; F = 32+C*9/5
F =
    97.8800
```

**Παράδειγμα 3: Απόσταση μεταξύ 2 σημείων**

Τα δύο σημεία  $S_1$  &  $S_2$  έχουν συντεταγμένες στο επίπεδο X-Y:  $S_1(x_1, y_1)$  &  $S_2(x_2, y_2)$ . Η απόσταση μεταξύ τους είναι (Πυθαγόρειο θεώρημα):

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Έστω ότι μας δίνονται τα δύο σημεία  $S_1(1,6)$  &  $S_2(4,8)$ , ο υπολογισμός της απόστασης στο MatLab θα έχει ως εξής:

Command Window:

```
>> x1 = 1; y1 = 4; x2 = 3; y2 = 8;
>> d = sqrt( (x2-x1)^2 + (y2-y1)^2 )
d =
    4.4721
```

**Παράδειγμα 4: Μετατροπή Καρτεσιανών συντεταγμένων σε Σφαιρικές**

Οι τύποι μετατροπής των Καρτεσιανών συντεταγμένων σε σφαιρικές είναι:

$$R = \sqrt{X^2 + Y^2}, \quad F = \text{τοξεφ}\left(\frac{Y}{X}\right)$$

Το  $\text{τοξεφ}()$  στο MatLab είναι η συνάρτηση  $\text{atan}()$ . Αν  $X = 4$  &  $Y = 4$  η ακτίνα  $R$  και η γωνία  $F$  θα είναι:

Command Window:

```
>> X = 4; Y = 4;
>> R = sqrt(X^2+Y^2), F = atan(Y/X)
R =
    5.6569
F =
    0.7854
```

## Εσωτερικές Συναρτήσεις

Στα παραδείγματα 3 και 4 χρησιμοποιήσαμε στους υπολογισμούς και συναρτήσεις του MatLab. Συγκεκριμένα είδαμε την τετραγωνική ρίζα **sqrt()** και το τόξο εφαπτομένης **atan()**. Το MatLab διαθέτει εκατοντάδες τέτοιες συναρτήσεις τις οποίες θα δούμε αναλυτικότερα παρακάτω. Μερικές από τις πλέον χρησιμοποιούμενες συναρτήσεις περιγράφονται στον επόμενο πίνακα. Ο πλήρης κατάλογος των συναρτήσεων έχει τοποθετηθεί στα Παραρτήματα του βιβλίου.

Όνομα Συναρτησης	Ερμηνεία	Παράδειγμα	Εντολή	Αποτέλεσμα
sin	ημίτονο	ημ(π)	sin(pi)	0.0
cos	συνημίτονο	συν(π)	cos(pi)	1.0
tan	εφαπτομένη	εφ(π/4)	tan(pi/4)	1.0
asin	τόξο ημιτόνου	τοξημ(1)	asin(1.0)	1.5708 (pi/2)
atan	τόξο εφαπτομένης	τοξεφ(1)	atan(1.0)	0.7854 (pi/4)
exp	εκθετικό	e <sup>1</sup>	exp(1.0)	2.7183
log	φυσικός λογάριθμος	ln(e)	log(2.7183)	1.0
log10	δεκαδικός λογάριθμος	log <sub>10</sub> (100)	log10(100.0)	2.0
sqrt	Τετραγωνική ρίζα	√16	sqrt(16.0)	4.0

## Οριακοί υπολογισμοί και σφάλματα

Στις περισσότερες μηχανές τα όρια των αριθμών κυμαίνονται από 10-308 έως 10+308 τα οποία είναι αρκετά για τη πλειοψηφία των υπολογισμών. Παρόλα αυτά μπορεί να προκύψει ένα αποτέλεσμα έξω από αυτή τη περιοχή. Όταν ένας αριθμός ξεπεράσει τα όρια αυτά, έχουμε υπερχείλιση στον εκθέτη (exponent overflow & exponent underflow) και το αποτέλεσμα δεν είναι σωστό. Π.χ.:

Command Window:

```
>> x = 2.0e+200 * 3.0e+200
x =
    Inf
>> y = 2.0e-200 * 3.0e-200
y =
    0
```

Η υπερχείλιση προς τα πάνω (overflow) αναφέρεται σαν **Inf** και η υπερχείλιση προς τα κάτω (underflow) στρογγυλοποιείται στο **0**. Άλλες περιπτώσεις που συναντάμε συχνά είναι η διαίρεση με το 0 και ο μη-αριθμός. Το αποτέλεσμα της διαίρεση με το 0 είναι μια άπειρη ποσότητα ( $\infty$ ), που αναφέρεται σαν **Inf**, ενώ το αποτέλεσμα άλλων πράξεων όπως 0/0 είναι εντελώς απροσδιόριστο και αναφέρεται σαν **NaN** (Not-a-Number).

Command Window:

```
>> z = 7.0 / 0.0
Warning: Divide by zero.
z =
    Inf
>> r = 0.0 / 0.0
Warning: Divide by zero.
r =
    NaN
```

## Πίνακες και Διανύσματα (Μητρώα)

Τα μητρώα είναι διατάξεις αριθμών σε γραμμές και στήλες στους οποίους εφαρμόζουμε ομοειδείς πράξεις ή μετατροπές με σκοπό π.χ. τη μετατροπή συντεταγμένων, τη λύση συστημάτων εξισώσεων, την επίλυση προβλημάτων γραμμικής άλγεβρας, τη βελτιστοποίηση μεγεθών, κλπ. Π.χ.:

$$\begin{matrix} [1] & [1 \ 2] & \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} & \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \\ (1 \times 1) & (1 \times 2) & (3 \times 1) & (2 \times 3) \end{matrix}$$

Το MatLab κατασκευάστηκε για να διευκολύνει το χειρισμό των μητρώων, κάτι που δεν προσφέρει καμία γλώσσα προγραμματισμού. Το MatLab στην ουσία λειτουργεί με ένα μόνο είδος μεταβλητής, το μητρώο, και κάθε μεταβλητή του MatLab αναφέρεται σε κάποιο μητρώο. Όσον αφορά τις διαστάσεις τους, διακρίνουμε 4 κατηγορίες μητρώων:

1. Τη **βαθμωτή** ποσότητα, δηλαδή έναν απλό αριθμό, που δεν είναι άλλο παρά ένα μητρώο διαστάσεων (1 x 1).
2. Τα **διανύσματα** που είναι μητρώα με μόνο μια γραμμή (1 x M) ή μια στήλη (N x 1).
3. Τα **τετραγωνικά** μητρώα (N x M) ή **πίνακες** τα οποία είναι και τα πιο συνηθισμένα.
4. Τα **τριών και άνω διαστάσεων** μητρώα (N x M x L x ...), τα οποία δεν είναι εύκολο να παρασταθούν στο επίπεδο (χαρτί ή οθόνη), αλλά χρησιμοποιούνται ευρύτατα από όλους τους επιστήμονες, τους μηχανικούς, ή, τους ερευνητές, και το MatLab τα χειρίζεται με την ίδια ευκολία όπως τα υπόλοιπα.

Για να **ορίσουμε ένα πίνακα** στο MatLab χρησιμοποιούμε τις αγκύλες [ ]. Τα στοιχεία του πίνακα που βρίσκονται στην ίδια γραμμή τα χωρίζουμε με κενά, και τις γραμμές τις χωρίζουμε μεταξύ τους με το ερωτηματικό (;). Π.χ.:

Command Window:

```
>> A = [ 1 2 3 ; 4 5 6 ]
A =
     1     2     3
     4     5     6
>> B = [ 1 2 3
4 5 6
7 8 9]
B =
     1     2     3
     4     5     6
     7     8     9
>> C = [1 2 3 4]
C =
     1     2     3     4
>> D = [1; 2; 3; 4]
D =
     1
     2
     3
     4
```

Για να αναφερθούμε σε **ένα συγκεκριμένο στοιχείο** ενός πίνακα γράφουμε το όνομα του πίνακα και αμέσως μετά σε παρένθεση ( ) γράφουμε τους δείκτες που ορίζουν τη θέση του στοιχείου που θέλουμε. Οι δείκτες είναι τόσοι όσες και οι διαστάσεις του πίνακα. Π.χ.

Command Window:

```
>> A(2,2), B(3,2)
ans =
     5
ans =
     8
```

Μπορούμε επίσης να επιλέξουμε και **μια περιοχή του πίνακα**, αν στη θέση κάθε δείκτη βάλουμε την αρχική του τιμή, μια άνω κάτω τελεία (:) και τη τελική του τιμή. Π.χ.:

Command Window:

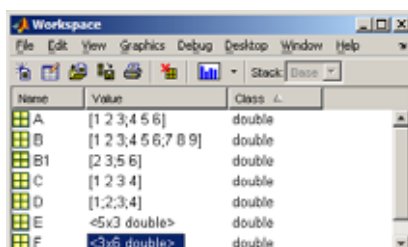
```
>> B1 = B(1:2,2:3)
B1 =
     2     3
     5     6
```

Τέλος μπορούμε να κατασκευάσουμε μεγάλους πίνακες από άλλους μικρότερους, ή από μείγμα πινάκων και στοιχείων. Π.χ.:

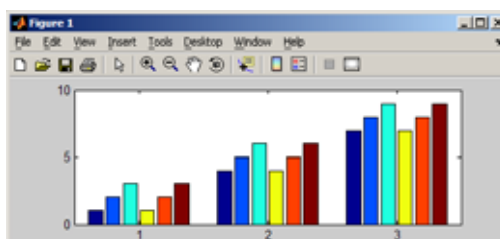
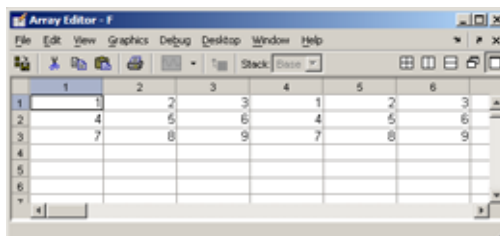
Command Window:

```
>> E = [ B; 10 11 12; 13 14 15]
E =
     1     2     3
     4     5     6
     7     8     9
    10    11    12
    13    14    15
>> F = [B B]
F =
     1     2     3     1     2     3
     4     5     6     4     5     6
     7     8     9     7     8     9
```

Οι πίνακες που δημιουργήσαμε με τις προηγούμενες εντολές βρίσκονται όλοι στη περιοχή εργασίας μας και εμφανίζονται στο παράθυρο Workspace:



Από το παράθυρο Workspace μπορούμε να επιλέξουμε να επεξεργαστούμε ένα πίνακα με τον Array Editor ή να τον παρουσιάσουμε σαν διάγραμμα, ιστόγραμμα, κλπ. Π.χ. για τον πίνακα F θα έχουμε αντίστοιχα:



### Συναρτήσεις για Δημιουργία Πινάκων

Το MatLab διαθέτει επιπλέον εντολές για την δημιουργία πινάκων ειδικής μορφής. Η **zeros()** δημιουργεί πίνακες με όλα τα στοιχεία 0, η **ones()** δημιουργεί πίνακες με όλα τα στοιχεία 1, και, η **eye()** δημιουργεί το μοναδιαίο πίνακα. Π.χ.:

```
Command Window:
>> Q = zeros(3,3)
Q =
    0    0    0
    0    0    0
    0    0    0
>> P = ones(3,3)
P =
    1    1    1
    1    1    1
    1    1    1
>> I = eye(3)
I =
    1    0    0
    0    1    0
    0    0    1
```

Άλλες συναρτήσεις χρήσιμες για τη δημιουργία πινάκων είναι οι:

Όνομα Συνάρτησης	Εξήγηση
rand	γενήτρια τυχαίου πίνακα
magic	μαγικά τετράγωνα
hilb	πίνακας Hilbert
triu	άνω τριγωνικός πίνακας
tril	κάτω τριγωνικός πίνακας
diag	ορισμός ή εμφάνιση της διαγωνίου

Εκτός από τις παραπάνω μεθόδους μπορούμε να δημιουργήσουμε διανύσματα με επαναληπτική μέθοδο, δίνοντας την αρχική τιμή, το βήμα και τη τελική τιμή που θα έχουν τα στοιχεία τους:

```
Command Window:
>> X = 1:5
X =
    1    2    3    4    5
>> Y = -8:2:8
Y =
   -8   -6   -4   -2    0    2    4    6    8
```

## Πράξεις με πίνακες

Οι βασικές πράξεις με πίνακες είναι οι:

- \* Πρόσθεση (+),
- \* Αφαίρεση (-),
- \* Πολλαπλασιασμός (\*),
- \* Ύψωση σε Δύναμη (^)
- \* Διαίρεση (/) & Αριστερή Διαίρεση (\)
- \* Ανάστροφος (')

Command Window:

```
>> (2*P-I)/2
ans =
    0.5000    1.0000    1.0000
    1.0000    0.5000    1.0000
    1.0000    1.0000    0.5000

>> A'
ans =
     1     4
     2     5
     3     6

>> A*A'
ans =
    14    32
    32    77
```

Η πρόσθεση και η αφαίρεση πινάκων γίνονται στοιχείο-στοιχείο ενώ οι άλλες πράξεις (\*, ^, /, \) γίνονται σαν πράξεις μητρώων. Το MatLab μας δίνει τη δυνατότητα να αναγκάσουμε και τις άλλες πράξεις να γίνουν στοιχείο-στοιχείο αν βάλουμε μια τελεία πριν το σύμβολο της πράξης, δηλαδή (\*, .^, ./, .\). Π.χ.:

Command Window:

```
>> [1 2 ; 3 4]^2
ans =
     7    10
    15    22

>> [1 2 ; 3 4].^2
ans =
     1     4
     9    16
```

Οι πράξεις στοιχείο-στοιχείο είναι πολύ χρήσιμες και στη γραφική απεικόνιση των αποτελεσμάτων μας. Εδώ ας σημειώσουμε ότι, όταν κάνουμε πράξεις πινάκων με βαθμωτά μεγέθη αυτές γίνονται πάντοτε στοιχείο-στοιχείο.

Το *MatLab* διαθέτει πολλές επιπλέον εντολές (συναρτήσεις) για ειδικές, ή πολύπλοκες, ή πολύ συχνές πράξεις πινάκων, όπως είναι η **inv()** που υπολογίζει τον αντίστροφο ενός πίνακα, η **max()** που επιστρέφει το μέγιστο στοιχείο του πίνακα, ή, η **eig()** που υπολογίζει τις ιδιοτιμές και τα ιδιοδιανύσματα ενός πίνακα. Λόγω του μεγάλου πλήθους των συναρτήσεων του *MatLab*, ο πλήρης κατάλογος με όλες τις συναρτήσεις έχει τοποθετηθεί στα Παραρτήματα του βιβλίου.

## Γραφικές Παραστάσεις

Τα αποτελέσματα των εργασιών σχεδόν πάντοτε περιέχουν και διαγράμματα ή γραφικές παραστάσεις. Οι γραφικές παραστάσεις είναι επίσης ο καλύτερος τρόπος για να παρουσιάσει κανείς μεγάλο αριθμό αποτελεσμάτων και για να εξάγει γενικά συμπεράσματα που δεν είναι προφανή από τους αριθμούς και μόνο.

Το *MatLab* διαθέτει ένα μεγάλο αριθμό ευέλικτων εντολών για τη δημιουργία γραφικών που καλύπτουν όλες τις ανάγκες του χρήστη, από τα απλά και γρήγορα γραφικά για το εργαστηριακό φύλλο έργου, μέχρι τα γραφικά με τις ειδικές προδιαγραφές που δημοσιεύονται σε διεθνή επιστημονικά περιοδικά.

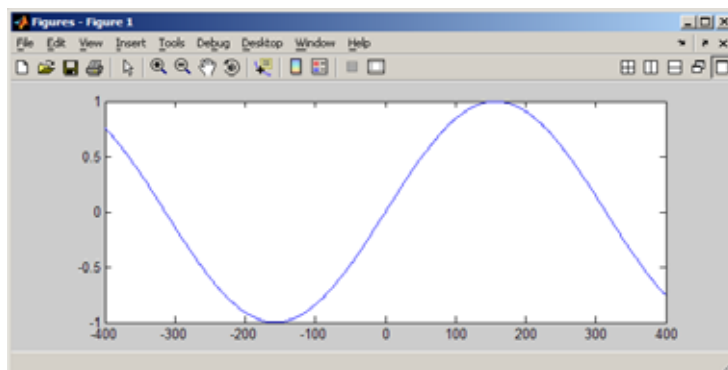
Οι πιο απλές εντολές για να δημιουργηθεί ένα γραφικό περιέχονται στον πίνακα που ακολουθεί:

Εντολή	Ερμηνεία
<code>plot(x,y)</code>	σχεδιάζει το $y$ vs $x$
<code>plot(x,y1,x,y2,x,y3)</code>	σχεδιάζει τα $y1, y2$ & $y3$ vs $x$ στο ίδιο γραφικό
<code>xlabel('X label')</code>	βάζει ετικέτα στον άξονα των $X$
<code>ylabel('Y label')</code>	βάζει ετικέτα στον άξονα των $Y$
<code>title('Title')</code>	βάζει τίτλο επάνω από το γραφικό
<code>grid</code>	προσθέτει ένα πλέγμα κάθετων & οριζόντιων γραμμών
<code>gtext('Text')</code>	προσθέτει όποιο κείμενο θέλουμε στο σημείο που θα δείξουμε με το ποντίκι.

Τα δεδομένα που θέλουμε να σχεδιάσουμε θα έχουν προκύψει από τις προηγούμενες εντολές ή από κάποιο αρχείο. Εδώ θα τα δημιουργήσουμε με δυο εντολές. Πρώτα δημιουργούμε το διάνυσμα  $X$  το οποίο αρχίζει από το  $-400$  και με βήμα  $1$  φτάνει στο  $400$  και περιέχει  $801$  στοιχεία. Στη συνέχεια δημιουργούμε το  $Y$  δηλαδή τη συνάρτηση που θέλουμε να σχεδιάσουμε και η οποία είναι:  $Y = \eta\mu(X/100)$ . Εφόσον το  $X$  είναι διάνυσμα, και το  $Y$  θα είναι διάνυσμα με το ίδιο μήκος. Τέλος δίνουμε την εντολή **plot(X,Y)**, και εμφανίζεται η γραφική παράσταση.

Command Window:

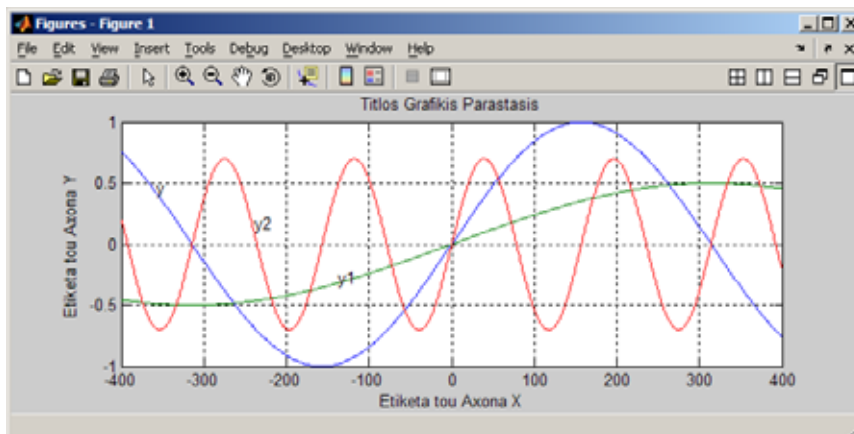
```
>> x = -400:400 ; y = sin(x/100);  
>> plot(x,y)
```



Στη συνέχεια θα δημιουργήσουμε μερικές ακόμα συναρτήσεις, θα τις σχεδιάσουμε όλες μαζί και θα προσθέσουμε πλέγμα, ετικέτες και τίτλους με τις εντολές του προηγούμενου πίνακα.

Command Window:

```
>> y1 = 0.5*sin(x/200) ; y2 = 0.7*sin(x/25);  
>> plot(x,y,x,y1,x,y2)  
>> grid  
>> xlabel('Etiketa tou Axona X')  
>> ylabel('Etiketa tou Axona Y')  
>> title('Titlos Grafikis Parastasis')  
>> gtext('y'), gtext('y1'), gtext('y2')
```

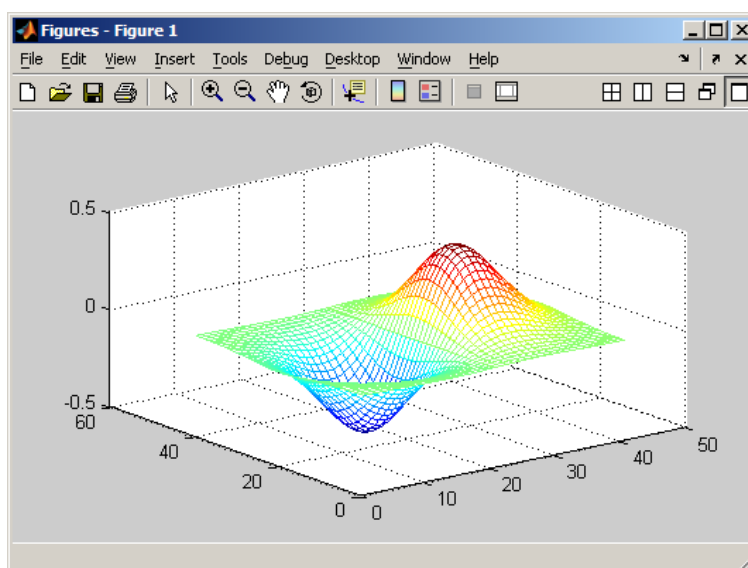


Για τις τρισδιάστατες (3-D) γραφικές παραστάσεις, το MatLab διαθέτει ειδικές εντολές όπως είναι η `mesh()`, `surf()`, κ.ά., που σχεδιάζουν τα δεδομένα στους άξονες X-Y-Z.

Για παράδειγμα για να σχεδιάσουμε την συνάρτηση:  $z = x \cdot \exp(-x^2 - y^2)$  στο διάστημα  $-2 < x < 2$  και  $-2 < y < 2$  δίνουμε τις εντολές:

Command Window:

```
>> [x,y] = meshgrid( -2:.1:2, -2:.1:2);  
>> z = x .* exp( -x.^2 - y.^2);  
>> mesh(z)
```



## Μετατροπή της εργασίας μας σε πρόγραμμα (M-file)

Όλα τα παραπάνω παραδείγματα που δείξαμε αλλά και η συντριπτική πλειοψηφία των εργασιών στο MatLab απαιτούν περισσότερες από μια εντολές για να ολοκληρωθούν. Επιπλέον, κάθε φορά που αλλάζει κάτι στα δεδομένα, θα πρέπει να ξαναγράφονται και να εκτελούνται όλες οι εντολές με τη σειρά. Ας μη ξεχνάμε όμως ότι, σκοπός της χρήσης του υπολογιστή δεν είναι μόνο να κάνει γρήγορα μια πράξη αλλά και να μπορεί να την επαναλάβει αργότερα χωρίς να χρειάζεται να του ξαναδοθούν όλες οι οδηγίες μια-μια.

Για να γίνει λοιπόν σωστή και αποδοτική χρήση της εφαρμογής του MatLab σε ένα Η/Υ θα πρέπει οι εργασίες μας, αφού περάσει το δοκιμαστικό στάδιο, να πάρει τη μορφή

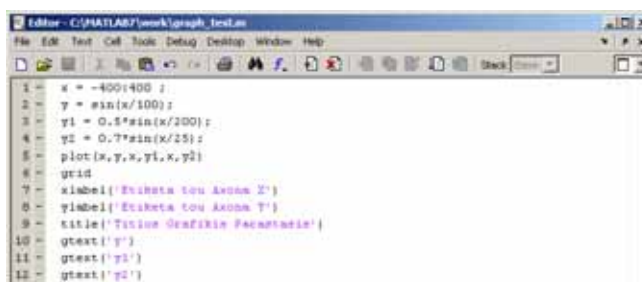
προγράμματος ώστε να αποθηκευτεί στον Η/Υ και να είναι διαθέσιμη σε κάθε επόμενη χρήση ή βελτίωση.

Η διαδικασία μετατροπής μιας ακολουθίας εντολών του CW σε πρόγραμμα είναι απλούστατη. Αντιγράφουμε όλες τις εντολές με την ίδια σειρά σε ένα αρχείο κειμένου και το αποθηκεύουμε με ένα όνομα και με επέκταση (.m). Από τη στιγμή αυτή έχουμε δημιουργήσει ένα πρόγραμμα, μια νέα εντολή ή συνάρτηση του MatLab, και όταν θα γράφουμε το όνομα του προγράμματος μας στο CW θα εκτελούνται με τη σειρά όλες οι εντολές που περιέχει.

### **Παράδειγμα 5: M-file με Γραφικές παραστάσεις**

Για παράδειγμα, οι γραφικές παραστάσεις της προηγούμενης παραγράφου απαιτούσαν πάνω από 10 εντολές για να πραγματοποιηθούν. Ας δούμε πως όλες αυτές τις εντολές τοποθετούνται σε ένα πρόγραμμα, και τι επιπλέον δυνατότητες μας παρέχονται όταν τις εκτελούμε σαν πρόγραμμα του MatLab.

Ανοίγουμε τον **Editor** από το μενού **File > New > M-file** και αντιγράφουμε από το CW όλες τις εντολές που δώσαμε στη προηγούμενη παράγραφο. Το αποθηκεύουμε με το όνομα: `graph_test.m`, δίνουμε στο CW την εντολή: `>> graph_test` και παρατηρούμε τα γραφικά να δημιουργούνται.



```
1 - x = -400:400 ;
2 - y = sin(x/100);
3 - y1 = 0.5*sin(x/200);
4 - y2 = 0.7*sin(x/25);
5 - plot(x,y,x,y1,x,y2)
6 - grid
7 - xlabel('Etiketa tou Axona X')
8 - ylabel('Etiketa tou Axona Y')
9 - title('Titlos Grafikis Parastasis')
10 - gtext('y')
11 - gtext('y1')
12 - gtext('y2')
```

Για να διακρίνεται το παράθυρο του Editor με τα M-files από το παράθυρο εντολών (CW), στο βιβλίο αυτό θα παριστάνεται πάντα με την παρακάτω μορφή:

M-file: `graph_test.m`

---

```
x = -400:400 ;
y = sin(x/100);
y1 = 0.5*sin(x/200);
y2 = 0.7*sin(x/25);
plot(x,y,x,y1,x,y2)
grid
xlabel('Etiketa tou Axona X')
ylabel('Etiketa tou Axona Y')
title('Titlos Grafikis Parastasis')
gtext('y')
gtext('y1')
gtext('y2')
```

---

Όταν μεταφέρουμε την εργασία μας σε ένα M-file μας δίνεται η δυνατότητα να προσθέσουμε και βοηθητικές εντολές οι οποίες εξυπηρετούν στην καλύτερη ανταλλαγή πληροφοριών με τον χρήστη του τελικού προγράμματος. Μπορούμε να προσθέσουμε εντολές εισόδου όπως η **input** που θα ζητάνε από τον χρήστη δεδομένα, εντολές εξόδου όπως η **disp** που θα εμφανίζουν στο χρήστη μηνύματα ή αποτελέσματα, εντολές μορφοποίησης των αποτελεσμάτων όπως οι **fprintf** & **sprintf**, ακόμη και γραμμές με επεξηγηματικά σχόλια οι οποίες αρχίζουν με το σύμβολο (%), δεν λαμβάνονται υπ' όψιν από το MatLab, και απευθύνονται στον χρήστη που δημιουργεί το M-file.

### *Παράδειγμα 6: Πρόγραμμα μετατροπής Μοιρών σε Ακτίνια*

Στο Παράδειγμα 1 είδαμε με ποια εντολή στο CW μετατρέπουμε τις μοίρες σε ακτίνια. Η μετατροπή αυτή είναι χρήσιμη γιατί όλοι μιλάμε με μοίρες αλλά οι τριγωνομετρικές συναρτήσεις στους Η/Υ λειτουργούν κυρίως με ακτίνια (rad). Εδώ, θα δημιουργήσουμε ένα πρόγραμμα MatLab (M-file) που θα κάνει τη μετατροπή αυτή. Γράφουμε τις εντολές του Παραδείγματος 1 στον Editor και αποθηκεύουμε το M-file με το όνομα m2a.m

```
M-file: m2a.m
moires = 45
aktinia = moires * pi / 180
```

Εκτελώντας την νέα μας εντολή (M-file) m2a στο CW, θα έχουμε:

Command Window:

```
>> m2a
moires =
    45
aktinia =
    0.7854
```

Το «πρόγραμμά» μας δουλεύει μια χαρά αλλά είναι πρακτικά άχρηστο γιατί υπολογίζει μόνο μια γωνία, αυτή των 45°. Για να το κάνουμε σωστό πρόγραμμα, θα του προσθέσουμε μια εντολή που θα ζητά από τον χρήστη (input) τις μοίρες της γωνίας και άλλη μια που θα δίνει την απάντηση (disp). Θα προσθέσουμε επίσης δυο μηνύματα ('...') και επεξηγηματικά σχόλια (%) για κάθε εντολή.

```
M-file: m2a.m
% Entoli Eisodou, me erwtisi gia tis moires
moires = input('DWSE TH GWNIA SE MOIRES: ');

% Entoli Ypologismou twn aktiniwn
aktinia = moires * pi / 180;

% Entoles Exodou, me to minyma kai to apotelesma se aktinia
disp('H GWNIA SE AKTINIA EINAI');
disp(aktinia);
```

Εκτελώντας τώρα το πρόγραμμα στο CW, θα πάρουμε ένα σαφώς καλύτερο αποτέλεσμα:

Command Window:

```
>> m2a
DWSE TH GWNIA SE MOIRES: 33
H GWNIA SE AKTINIA EINAI
0.5760
>> m2a
DWSE TH GWNIA SE MOIRES: 180
H GWNIA SE AKTINIA EINAI
3.1416
```

Για να ολοκληρώσουμε το πρόγραμμα πρέπει να προσθέσουμε στην αρχή 2-3 γραμμές σχολίων που θα το περιγράφουν. Η περιγραφή θα πρέπει να περιλαμβάνει τουλάχιστον το όνομα του, τη λειτουργία του και τις ενέργειες του χρήστη. Μια πιθανή τελική μορφή του προγράμματός μας είναι η παρακάτω:

```
M-file: m2a.m
% m2a.m
% Programma metatropis Moirwn se Aktinia:
% O xristis dinei ti gwnia se moires,
% kai to programma tin ypologizei se aktinia.

% Entoli Eisodou, me erwtisi gia tis moires
moires = input('DWSE TH GWNIA SE MOIRES: ');

% Entoli Ypologismou twn aktiniwn
aktinia = moires * pi / 180;

% Entoles Exodou, me to minyma kai to apotelesma se aktinia
disp('H GWNIA SE AKTINIA EINAI');
disp(aktinia);
```

Το MatLab έχει ένα πολύ χρήσιμο χαρακτηριστικό, όταν δώσουμε την εντολή `help` και το όνομα ενός προγράμματος, χρησιμοποιεί τις πρώτες γραμμές σχολίων του προγράμματος και τις παρουσιάζει στην οθόνη σαν βοήθεια προς τον χρήστη. Έτσι, ότι γράψουμε στα πρώτα σχόλια αποτελεί και τις οδηγίες, τη βοήθεια ή το manual του προγράμματός μας. Πράγματι αν δώσουμε στο CW την εντολή θα έχουμε:

Command Window:

```
>> help m2a
m2a.m
Programma metatropis moirwn se aktinia:
O xristis dinei ti gwnia se moires, kai
to programma tin ypologizei se aktinia.
>>
```

### Παράδειγμα 7: Συνάρτηση μετατροπής Μοιρών σε Ακτίνια

Το πρόγραμμα που φτιάξαμε στο προηγούμενο παράδειγμα μπορεί να εκτελεστεί μόνο από τον χρήστη. Όμως οι μετατροπές των γωνιών είναι συχνά χρήσιμες και μέσα σε άλλα προγράμματα. Για να μπορεί ένα πρόγραμμα να κληθεί από ένα άλλο πρόγραμμα πρέπει να το κάνουμε συνάρτηση (function). Η συνάρτηση είναι ένα υποπρόγραμμα και διαφέρει από τα προγράμματα μόνο στον τρόπο με τον οποίο παίρνει τα δεδομένα και επιστρέφει το αποτέλεσμα.

Οι συναρτήσεις δέχονται τα δεδομένα από την παρένθεση που ακολουθεί το όνομά τους, και επιστρέφουν την απάντηση στη θέση του ονόματός τους, σαν να ήταν μεταβλητές.

Για να μετατρέψουμε το `m2a.m` σε συνάρτηση πρέπει να κάνουμε τις παρακάτω αλλαγές στην αρχή και στο τέλος του προγράμματος. Η πρώτη εντολή θα πρέπει να είναι η εντολή `function`, και θα περιέχει τη μεταβλητή που επιστρέφει η συνάρτηση, το `=`, το όνομά της, και, τη παρένθεση με τη μεταβλητή εισόδου (δεδομένα) της συνάρτησης. Στη συνέχεια διαγράφουμε όλες τις άλλες εντολές εισόδου και εξόδου (`input`, `disp`, κλπ.).

```
M-file: m2a.m


---


function aktinia = m2a(moires)
% m2a.m
% Synartisi metatropis moirwn se aktinia
% Syntax: aktinia = m2a(moires)
% Input: moires, Output: aktinia
aktinia = moires * pi / 180;


---


```

Μπορούμε τώρα να εκτελέσουμε τη συνάρτηση μέσα σε οποιοδήποτε άλλο πρόγραμμα. Για να θυμηθούμε τη σύνταξή της δίνουμε `help m2a` και εμφανίζονται οι γραμμές με τα σχόλια που γράψαμε. Αν τώρα μια μεταβλητή π.χ., `akt180` πρέπει να περιέχει τα ακτίνια των 180 μοιρών θα γράψουμε τη εντολή: `akt180 = m2a(180)`.

Command Window:

```
>> help m2a
m2a.m
Synartisi metatropis moirwn se aktinia
Syntax: aktinia = m2a(moires)
Input: moires, Output: aktinia

>> akt180 = m2a(180)
akt180 =
    3.1416

>>
```

Όπως βλέπουμε, τα M-files του MatLab εκτός από τις απλές εντολές ανάθεσης, μπορεί να περιέχουν και άλλα είδη εντολών όπως: εντολές που πραγματοποιούν επαναλήψεις (`for`),

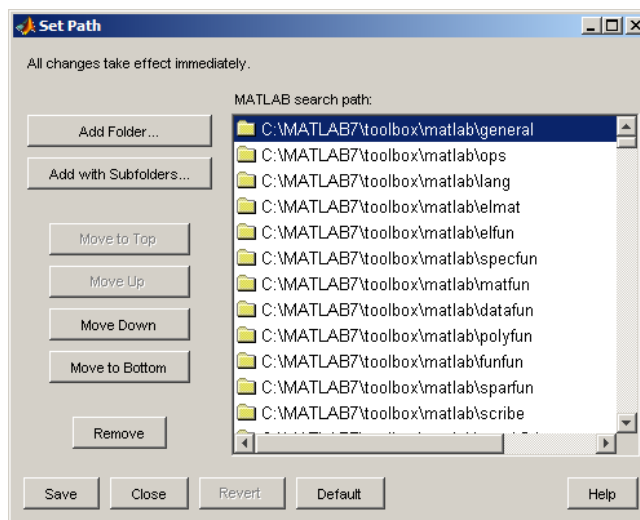
εντολές σύγκρισης ή ελέγχου (if, case), εντολές μορφοποίησης της εξόδου (fprintf), εντολές χειρισμού αρχείων και I/O, και άλλες που θα τις δούμε πιο αναλυτικά στα επόμενα κεφάλαια.

Το *MatLab* λοιπόν, εκτός από αλληλεπιδραστικό περιβάλλον είναι ταυτόχρονα και προγραμματιστικό περιβάλλον. Επιπλέον, κάθε πρόγραμμα (M-file) που φτιάχνουμε είναι ταυτόχρονα και μια νέα εντολή ή συνάρτηση του *MatLab*. Αυτή η ιδιότητα δίνει στο πακέτο τη δυνατότητα να προσαρμόζεται στις ανάγκες μας και να επεκτείνεται και να ενημερώνεται απεριόριστα. Σήμερα έχουμε στη διάθεσή μας εκατοντάδες τέτοιες προσθήκες οι οποίες αναπτύχθηκαν από επιστήμονες που χρησιμοποιούσαν το *MatLab*. Τα συμπληρωματικά αυτά πακέτα με αρχεία (M-files) ονομάζονται *Toolbox* και προσθέτουν πληθώρα εντολών και συναρτήσεων στις ήδη υπάρχουσες, διευκολύνοντας ακόμη περισσότερο τους υπόλοιπους χρήστες.

## Άλλες Εντολές ή Ρυθμίσεις

### *Path*

Για να λειτουργήσει ομαλά το περιβάλλον του *MatLab*, θα πρέπει να έχει πρόσβαση σε όλους τους φακέλους που περιέχουν τις εντολές και τα M-file. Αυτό γίνεται με τη ρύθμιση του **search path** του *MatLab*. Το **search path** πρέπει να περιέχει όλες τις διαδρομές των φακέλων που περιέχουν αρχεία του *MatLab* (M-file), τα πρόσθετα *Toolbox*, ή τον φάκελο με τις εργασίες μας. Το **search path** του *MatLab* ρυθμίζεται μέσα από το μενού: **File > Set Path...** προσθέτοντας και αφαιρώντας φακέλους στο αντίστοιχο παράθυρο που εμφανίζεται. Η σειρά εμφάνισης στο **search path** είναι σημαντική όταν υπάρχουν δυο M-file με το ίδιο όνομα σε διαφορετικούς φακέλους καθώς θα εκτελείται πάντα αυτό που προηγείται.



## Ημερομηνία

Το *MatLab* διαθέτει επίσης εντολές σχετικές με την ημερομηνία και ώρα του συστήματός μας. Η εντολή **date** εμφανίζει την τρέχουσα ημερομηνία του Η/Υ και είναι ιδιαίτερα πρακτική για τη χρονική σήμανση δεδομένων, αποτελεσμάτων και αρχείων καταγραφής (diary). Πιο ολοκληρωμένη είναι η εντολή **clock** η οποία δίνει την ημερομηνία και την ώρα του συστήματος με τρόπο ώστε να είναι επεξεργάσιμη από άλλες εντολές ή προγράμματα. Η έξοδος της **clock** είναι ένα διάνυσμα 6 στοιχείων με τη συγκεκριμένη σειρά: [ έτος μήνας ημ/νία ώρα λεπτά δεύτερα ]. Η έξοδος της **clock** είναι σε εκθετική μορφή αλλά μπορεί να γίνει ακέραια με τη βοήθεια της **fix()**.

Command Window:

```
>> date
ans =
23-Feb-2006

>> clock
ans =
1.0e+003 *
2.0060    0.0020    0.0230    0.0100    0.0160    0.0423

>> fix(clock)
ans =
2006         2         23         10         16
45
```

## Ταχύτητα εκτέλεσης

Η ταχύτητα εκτέλεσης των εντολών είναι επίσης σημαντική για τα περισσότερα επιστημονικά προβλήματα. Αλλιώς σχεδιάζεται μια προσομοίωση που τρέχει μέσα σε 10 λεπτά και αλλιώς μια που χρειάζεται 2 μέρες για να ολοκληρωθεί. Για να έχουμε μια εκτίμηση της απόδοσης του Η/Υ που χρησιμοποιούμε, το *MatLab* διαθέτει ένα δικό του τεστ επιδόσεων. Η σύγκριση των επιδόσεων του Η/Υ μας με άλλα συστήματα αναφοράς, γίνεται με την εντολή **bench**.

Για παράδειγμα για έναν Η/Υ με επεξεργαστή Intel Pentium 4, στα 1.5 GHz η εντολή δίνει τα παρακάτω συγκριτικά αποτελέσματα.

Command Window

```
>> bench
```

