

ΣΗΜΕΙΩΣΕΙΣ  
ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ  
ΗΛΕΚΤΡΟΝΙΚΩΝ  
ΥΠΟΛΟΓΙΣΤΩΝ ΜΕ MATLAB

Δρ Αθανάσιος Φράγκου

Βόλος, 2017

## ΕΙΣΑΓΩΓΗ

Οι σημειώσεις αυτές έχουν γραφτεί στο πλαίσιο των παραδόσεων του μαθήματος του 1ου εξαμήνου «Προγραμματισμός Η/Υ» που αποτελούν μέρος αυτού. Αρχικά παρουσιάζονται οι βασικές έννοιες της MATLAB με τις πρώτες βασικές λειτουργίες προκειμένου να μπορεί ο φοιτητής να έχει μια αρχική πρόσβαση στην εξειδικευμένη αυτή γλώσσα προγραμματισμού μαθηματικών. Το δεύτερο μέρος εισάγει τον φοιτητή σε βασικές έννοιες προγραμματισμού όπως οι εντολές επαναλήψεων και οι λογικές εκφράσεις. Για περεταίρω μελέτη μπορεί ο φοιτητής να απευθυνθεί στις αναφορές που δίνονται στο τέλος των σημειώσεων αυτών.

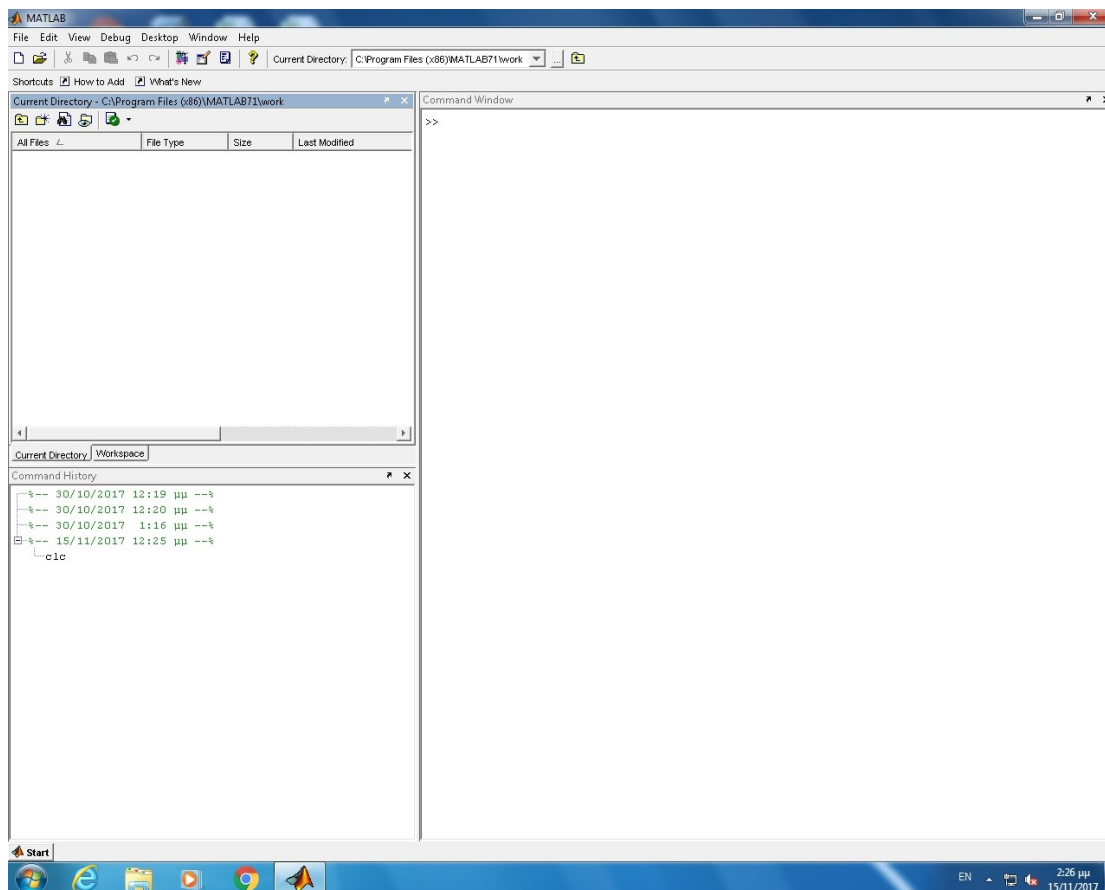
## Περιεχόμενα

Μέρος 1° .....	4
Βασικές Έννοιες .....	4
1. Αρχική Οθόνη .....	5
2. Βασικές πράξεις.....	6
2.1 Ο επιστημονικός συμβολισμός των αριθμών .....	7
3. Μεταβλητές .....	8
4. Πίνακες.....	11
4.1 Γινόμενο – διαίρεση - δυνάμεις διανυσμάτων .....	13
4.2 Πίνακες μορφής $n \times m$ και $n \times n$ .....	17
5. Πολυώνυμα.....	23
6. Βασικές συναρτήσεις.....	23
7. Εξαγωγή Δεδομένων - Γραφήματα .....	28
7.1 Εκτύπωση δεδομένων στην οθόνη .....	28
7.2 Γραφήματα .....	32
Μέρος 2ο .....	40
Εισαγωγή στον Προγραμματισμό .....	40
1.Αρχεία με κατάληξη .m (m-files) .....	41
2.Αρχεία συνάρτησης (function) .....	43
3.Βασικές εντολές προγραμματισμού .....	46
3.1 βρόχοι for (for loops).....	46
3.2 βρόχοι while (while loops) .....	50
3.3 εντολή if (if statement) .....	53
Βιβλιογραφία.....	56

Μέρος 1<sup>ο</sup>  
Βασικές Έννοιες

## 1. Αρχική Οθόνη

Αφού εγκαταστήσουμε το MATLAB στον υπολογιστή ξεκινούμε το πρόγραμμα και θα εμφανίζεται στην οθόνη μας το αρχικό παράθυρο της MATLAB



Αποτελείται από τα εξής μέρη (υποπαράθυρα)

1. Το υποπαράθυρο του Current Directory
2. Το υποπαράθυρο Command Window
3. Το υποπαράθυρο workspace

### ***Current Directory***

Ο χώρος (τρέχων φάκελος) του σκληρού δίσκου στον οποίο βρισκόμαστε τη στιγμή που ανοίγουμε το MATLAB και μπορούμε να αποθηκεύσουμε τα αρχεία μας. Αυτό

μπορεί εύκολα να αλλάξει προκειμένου να αποθηκεύσουμε σε άλλο φάκελο την εργασία μας.

### ***Command Window***

Στο χώρο αυτό γράφουμε τις εντολές για την υλοποίηση των πράξεων καθώς και άλλων λειτουργιών στο MATLAB

### ***Workspace***

Αποτελεί το χώρο αποθήκευσης και διαχείρισης των μεταβλητών

## **2. Βασικές πράξεις**

Στο MATLAB μπορούμε να κάνουμε τις βασικές πράξεις όπως σε μια αριθμομηχανή με τα εξής σύμβολα:

Πρόσθεση:                    +

Αφαίρεση:                   -

Πολλαπλασιασμός:        \*

Διαίρεση:                    /

Δύναμη:                      ^

### **Παράδειγμα**

```
>> 5+3
```

```
ans =
```

```
8
```

```
>> 12*2
```

```
ans =
```

```
24
```

```
>> 27/3
```

```
ans =
```

```
9
```

```
>> 98/3
```

```
ans =
    32.6667

>> 5^2

ans =
    25
```

## 2.1 Ο επιστημονικός συμβολισμός των αριθμών

Το MATLAB εντυπώνει κάποιον αριθμό σε επιστημονική μορφή χρησιμοποιώντας το γράμμα e μετά τον αριθμό χωρίς κενό και ύστερα από αυτό τον εκθέτη τον οποίο έχει η δύναμη του 10.

### Παράδειγμα

Θα υπολογίσουμε το άθροισμα των  $2.2345 \cdot 10^3$  και  $5.3245 \cdot 10^4$

```
> 2.2345*10^3

ans =
    2.2345e+03

5.3245*10^4

ans =
    5.3245e+04

>> 2.2345*10^3 + 5.3245*10^4

ans =
    5.5479e+04
```

Οι πράξεις μεταξύ των αριθμών εκτελούνται όπως και ορίζεται αλγεβρικά, δηλαδή αρχικά απαλείφονται οι παρενθέσεις, έπειτα οι δυνάμεις στη συνέχεια οι πολλαπλασιασμοί και διαιρέσεις και τέλος οι προσθέσεις και αφαιρέσεις από αριστερά προς τα δεξιά.

### Παράδειγμα

Να υπολογιστεί η παράσταση

$$\frac{(2 + 3)^3 + (-1)^2}{4 \cdot 5^4}$$

```
> ((2+3)^3+(-1)^2)/4*5^4
```

```
ans =
```

```
1.9688e+04
```

### 3. Μεταβλητές

Πολλές φορές κατά την σύνταξη εντολών στη MATLAB είναι αναγκαίο να καταχωρούνται συγκεκριμένες τιμές αριθμών σε μια μεταβλητή. Η αριθμητική τιμή που εκχωρείται στη μεταβλητή καταλαμβάνει θέση στη μνήμη του υπολογιστή και σε δεδομένη στιγμή που καλείται το όνομα αυτής ο υπολογιστής επαναφέρει την αριθμητική αυτή τιμή.

Το όνομα της μεταβλητής πρέπει να ξεκινά από γράμμα (αγγλικά) και να αποτελείται από γράμματα αριθμούς και κάτω παύλες (πχ met\_10). Έπειτα από κάθε μεταβλητή ακολουθεί το σύμβολο της ισότητας (=) και ολοκληρώνεται η δήλωσή της με την εκχώρηση της τιμής. Προσοχή δε μπορούν να χρησιμοποιηθούν ονόματα που έχουν δεσμευθεί από τη MATLAB όπως συναρτήσεις κλπ. Αποδεκτές μεταβλητές είναι για παράδειγμα οι met, met25s, xy01, XY01, s3445, ενώ μη αποδεκτές είναι για παράδειγμα οι m-et, 25met, %xy, @nikos.

#### Παράδειγμα

```
>> X=25
```

```
X =
```

```
25
```

```
>> Y=15
```

```
Y =
```

```
15
```

```
>> Z=X+Y
```



```
Z =  
40
```

Οι αριθμητικές μεταβλητές εκχωρούνται σε μια θέση μνήμης η οποία φαίνεται στο **Command Window** και το αποτέλεσμα των πράξεων μεταξύ δύο ή περισσότερων μεταβλητών. Επιπλέον τα ανωτέρω εκχωρούνται στο υποπαράθυρο **Workspace**.

Εάν δεν επιθυμούμε να επαναλαμβάνεται η μεταβλητή στην οθόνη αλλά να την κρατά στη μνήμη για οπουδήποτε μελλοντική χρήση, τότε βάζουμε ερωτηματικό μετά την εισαγωγή της εντολής. Για το ανωτέρω παράδειγμα γράφουμε:

```
>> X=25;  
>> Y=15;  
>> Z=X+Y  
Z =  
40
```

Δεσμευμένες μεταβλητές από το MATLAB που δεν μπορούμε να εκχωρήσουμε ως μεταβλητές είναι για παράδειγμα οι τριγωνομετρικές συναρτήσεις και κάποιοι δεσμευμένοι αριθμοί όπως ο  $\pi=3.14$

```
>> pi  
  
ans =  
  
3.1416
```

Εάν επιθυμούμε την εμφάνιση περισσότερων δεκαδικών ψηφίων του αριθμού μπορούμε με την εντολή `format` να αυξήσουμε ή να μειώσουμε τον αριθμό των δεκαδικών ψηφίων

```
>> format long  
>> pi  
  
ans =  
  
3.141592653589793
```

```
>> format short
>> pi
```

```
ans =
```

```
3.1416
```

Εκτός από πραγματικούς αριθμούς μπορούν να καταχωρηθούν και μεταβλητές μιγαδικών αριθμών.

### Παράδειγμα

Έστω οι αριθμοί  $a = 4 + 5i$  και  $b = 2.1 - 3.5i$ . Να υπολογιστεί το άθροισμα, το γινόμενο και το πηλίκο μεταξύ του  $a$  και  $b$ .

```
>> a=4+5i
```

```
a =
```

```
4.0000 + 5.0000i
```

```
>> b=2.1-3.5i
```

```
b =
```

```
2.1000 - 3.5000i
```

```
>> a+b
```

```
ans =
```

```
6.1000 + 1.5000i
```

```
>> a*b
```

```
ans =
```

```
25.9000 - 3.5000i
```

```
>> a/b
```

```
ans =
```

```
-0.5462 + 1.4706i
```

## 4. Πίνακες

Στη MATLAB μπορούν να εισαχθούν δύο μορφές πινάκων:

A. Διανύσματα (Πίνακας μορφής  $1 \times n$ )

B. Πίνακες μορφής  $n \times m$  και  $n \times n$

Έστω ότι θέλουμε να προσθέσουμε τα διανύσματα  $a = (1,2,3)$ ,  $b = (2,4,1)$

Αρχικά εισάγουμε τα διανύσματα ως μεταβλητές στο MATLAB και έπειτα εκτελούμε την πράξη.

Εισάγουμε το διάνυσμα μέσα σε αγκύλες και τα στοιχεία αυτού χωρισμένα με κενά ή κόμμα (διάνυσμα γραμμή) ή χωρισμένα με ελληνικό ερωτηματικό (;) εάν επιθυμούμε να εισάγουμε διάνυσμα στήλη

```
>> x=[1 4 2 6]
```

```
x =
```

```
1      4      2      6
```

```
>> y=[3 5 2 1]
```

```
y =
```

```
3      5      2      1
```

ή τα στοιχεία του διανύσματος χωρισμένα με ερωτηματικό

```
>> x=[1;4;2;6]
```

```
x =
```

```
1
```

```
4
```

```
2
```

```
6
```

```
>> y=[3;5;2;1]
```

```
y =
```

```
3
```

```
5
2
1
```

Μπορούμε εύκολα να προσθέσουμε ή να αφαιρέσουμε τα δύο αυτά διανύσματα γράφοντας

```
>> sum=x+y
```

```
sum =
```

```
4
9
4
7
```

```
>> diaf=x-y
```

```
diaf =
```

```
-2
-1
0
5
```

Μπορούμε να εισάγουμε διάνυσμα n-στοιχείων ισαπέχοντα με την εντολή **linspace(a,b,n)**

### Παράδειγμα

Να εισάγετε διάνυσμα t με στοιχεία από το μηδέν έως το δύο [0,2] με 11 ισαπέχοντα στοιχεία μεταξύ τους και να υπολογιστεί η τιμή της παράστασης  $y = 5\eta\mu(2\pi t)$

```
>> t=linspace(0,2,11)
```

```
t =
```

```
0      0.2000    0.4000    0.6000    0.8000    1.0000    1.2000
1.4000    1.6000    1.8000    2.0000
```

```
>> y=5*sin(2*pi*t)
```

```
y =
```

```

0      4.7553    2.9389   -2.9389   -4.7553   -0.0000    4.7553
2.9389   -2.9389   -4.7553   -0.0000

```

#### 4.1 Γινόμενο – διαίρεση - δυνάμεις διανυσμάτων

Το γινόμενο διανυσμάτων μπορεί να είναι είτε το κλασικό εσωτερικό γινόμενο

$$\vec{a} \cdot \vec{b} = \sum_{i=1}^n a_i b_i$$

Είτε το γινόμενο  $\vec{a} \cdot \vec{b} = [a_1 b_1, a_2 b_2, \dots, a_n b_n]$ , δηλαδή γινόμενο στοιχείο με στοιχείο.

Για τα διανύσματα  $a = (1,2,3), b = (2,4,1)$  υπολογίζεται το εσωτερικό γινόμενο με (\*) ενώ το γινόμενο στοιχείο προς στοιχείο με (.\*). όπως παρακάτω:

Εσωτερικό γινόμενο

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

```
a =
```

```

1      2      3

```

```
>> b=[2;4;1]
```

```
b =
```

```

2

```

```

4

```

```

1

```

```
>> es_gin=a*b
```

```
es_gin =
```

```

13

```

Προσοχή στην εισαγωγή του διανύσματος. Πρέπει το διάνυσμα  $b$  να είναι διάνυσμα στήλη (από τον ορισμό του εσωτερικού γινομένου). Εάν όχι το MATLAB μας ειδοποιεί με μήνυμα λάθους όπως παρακάτω:

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

```
a =
```

```
1      2      3
```

```
>> b=[2 4 1]
```

```
b =
```

```
2      4      1
```

```
>> es_gin=a*b
```

```
Error using *
```

```
Inner matrix dimensions must agree.
```

### Γινόμενο στοιχείο προς στοιχείο

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

```
a =
```

```
1      2      3
```

```
>> b=[2 4 1]
```

```
b =
```

```
2      4      1
```

```
>> Had_gin=a.*b
```

```
Had_gin =
```

```
2      8      3
```

Η *διαίρεση* μεταξύ διανυσμάτων δεν ορίζεται ωστόσο μπορούμε να διαιρέσουμε τα στοιχεία δύο διανυσμάτων *στοιχείο προς στοιχείο* όπως και στον πολλαπλασιασμό.

Για την πράξη αυτή χρησιμοποιούμε το σύμβολο *(./)*.

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

```
a =
```

```
1     2     3
```

```
>> b=[2 4 1]
```

```
b =
```

```
2     4     1
```

```
>> diair=a./b
```

```
diair =
```

```
0.5000    0.5000    3.0000
```

Εάν κατά τη διαίρεση μεταξύ των αριθμών παρουσιαστεί μηδενικός παρονομαστής τότε το MATLAB δίνει ως εξαγόμενο το σύμβολο **Inf**

```
>> a=[2 4 6]
```

```
a =
```

```
2     4     6
```

```
>> b=[2 2 0]
```

```
b =
```

```
2     2     0
```

```
>> a./b
```

```
ans =
```

```
1      2      Inf
```

Παρατήρηση: Εάν δεν θέσουμε όνομα μεταβλητής για το εξαγόμενο τότε θέτει το MATLAB την μεταβλητή `ans`

Εάν επιχειρηθεί διαίρεση μεταξύ δύο μηδενικών τότε το MATLAB δίνει ως εξαγόμενο το σύμβολο **NaN** (Not a Number)

```
>> a=[2 4 0]
```

```
a =
```

```
2      4      0
```

```
>> b=[2 2 0]
```

```
b =
```

```
2      2      0
```

```
>> a./b
```

```
ans =
```

```
1      2      NaN
```

Για να υπολογίσουμε τη δύναμη των στοιχείων ενός διανύσματος χρησιμοποιούμε τον τελεστή `(.^)`. Για το διάνυσμα  $\vec{x}(2, 4, 3, 7)$  υπολογίζουμε την  $\overrightarrow{x^2}$

```
>> x=[2 4 3 7]
```

```
x =
```

```
2      4      3      7
```



```
>> dyn=x.^2
```

```
dyn =
```

```
4      16      9      49
```

## 4.2 Πίνακες μορφής $n \times m$ και $n \times n$

Έστω ότι θέλουμε να υπολογίσουμε την  $C = 2A - 3B$  αλγεβρική παράσταση που αποτελείται από τους πίνακες  $A$  και  $B$  και αριθμούς όπου

$$A = \begin{bmatrix} 1 & 3 & 2 \\ 4 & 2 & 3 \\ 5 & 6 & 1 \end{bmatrix}, B = \begin{bmatrix} 4 & 2 & 3 \\ 1 & 3 & 6 \\ 7 & 2 & 9 \end{bmatrix}$$

Η εισαγωγή των πινάκων μπορεί να γίνει είτε γραμμή - γραμμή είτε σε μια σειρά με διαχωριζόμενες τις γραμμές με ελληνικό ερωτηματικό (;)

```
>> A=[1 3 2;4 2 3;5 6 1]
```

```
A =
```

```
1      3      2
4      2      3
5      6      1
```

```
>> B=[4 2 3
```

```
1 3 6
```

```
7 2 9]
```

```
B =
```

```
4      2      3
1      3      6
7      2      9
```

Η εισαγωγή της πράξης και το αποτέλεσμα φαίνεται παρακάτω

```
C=2*A-3*B
```

```
C =
```

```
-10     0     -5  
     5    -5    -12  
    -11     6    -25
```

Οι πράξεις του πολλαπλασιασμού αριθμού με πίνακα και της αφαίρεσης δύο πινάκων έγινε με τον κλασσικό αλγεβρικό τρόπο.

Αντίστοιχα με τα διανύσματα το MATLAB μπορεί να υπολογίσει το αλγεβρικό γινόμενο πινάκων κάνοντας χρήση του συμβόλου (\*) και το γινόμενο πινάκων στοιχείο προς στοιχείο κάνοντας χρήση του συμβόλου (.\*)).

### Παράδειγμα

Να υπολογιστεί το:

i.  $C1 = A \cdot B,$

ii.  $C2 = A \times B$

```
>> A=[1 3 2;4 2 3;5 6 1]
```

```
A =
```

```
1     3     2  
4     2     3  
5     6     1
```

```
>> B=[4 2 3;1 3 6;7 2 9]
```

```
B =
```

```
4     2     3  
1     3     6  
7     2     9
```

I.

```
>> C1=A*B
```

```
C1 =
```

21	15	39
39	20	51
33	30	60

II.

```
>> C2=A.*B
```

C2 =

4	6	6
4	6	18
35	12	9

Για τον υπολογισμό του ανάστροφου του πίνακα  $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$  ο οποίος είναι

$$A_{αναστρ} = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

γράφουμε

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

A =

1	2	3
4	5	6
7	8	9

```
>> A'
```

ans =

1	4	7
2	5	8
3	6	9

Η εντολή **size** μας δίνει τις διαστάσεις ενός πίνακα  $B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 1 & 6 & 2 \\ 7 & 3 & 1 \end{bmatrix}$

```
>> B=[1 2 3;4 5 6;1 6 2;7 3 1]
```

```
B =
```

1	2	3
4	5	6
1	6	2
7	3	1

```
>> size(B)
```

```
ans =
```

4	3
---	---

Όπως βλέπουμε οι διαστάσεις είναι 4x3.

Επιπλέον μπορούν να εισαχθούν και ειδικές κατηγορίες πινάκων όπως:

1. Πίνακας με μηδενικά στοιχεία με την εντολή **zeros**

```
>> C=zeros(4,5)
```

```
C =
```

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Εισάγεται πίνακας τεσσάρων γραμμών και πέντε στηλών με στοιχεία μηδενικά.

2. Πίνακας με στοιχεία μονάδες με την εντολή **ones**

```
>> D=ones(3,5)
```

```
D =
```

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

Εισάγεται πίνακας τριών γραμμών και πέντε στηλών με στοιχεία μονάδα.

3. Μοναδιαίος Πίνακας με την εντολή **eye**

```
>> I=eye(5)
```

I =

1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

Εισάγεται μοναδιαίος πίνακας 5x5.

#### 4. Διαγώνιος Πίνακας από δεδομένο διάνυσμα με την εντολή **diag**

```
>> u=[4 3 7 2]
```

u =

4	3	7	2
---	---	---	---

```
>> U=diag(u)
```

U =

4	0	0	0
0	3	0	0
0	0	7	0
0	0	0	2

Από το διάνυσμα  $u = (4,3,7,2)$  δημιουργήθηκε ο διαγώνιος πίνακας U

Για την εξαγωγή στοιχείων από κάποιο πίνακα που δημιουργήσαμε πληκτρολογούμε το όνομα του πίνακα και τη συντεταγμένη του στοιχείου που επιθυμούμε.

```
>> B=[1 2 3;4 5 6;1 6 2;7 3 1]
```

B =

1	2	3
4	5	6
1	6	2
7	3	1

```
>> B(4,1)
```

```
ans =
```

```
7
```

Για τον πίνακα  $B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 1 & 6 & 2 \\ 7 & 3 & 1 \end{bmatrix}$  έγινε εξαγωγή του στοιχείου της 4<sup>ης</sup> γραμμής - 1<sup>ης</sup> στήλης,

δηλαδή το επτά (7).

Για εξάγουμε ολόκληρη στήλη πληκτρολογούμε

```
>> B(:,2)
```

```
ans =
```

```
2
```

```
5
```

```
6
```

```
3
```

Για εξάγουμε δύο στήλες ή περιοχές πληκτρολογούμε

```
> B(:,2:3)
```

```
ans =
```

```
2     3
```

```
5     6
```

```
6     2
```

```
3     1
```

## 5. Πολυώνυμα

Τα πολυώνυμα παριστάνονται στη MATLAB όπως και τα διανύσματα, δηλαδή εισάγουμε τους συντελεστές του πολυωνύμου με τρόπο όμοιο ενός διανύσματος.

Προκειμένου λοιπόν να εισάγουμε το  $P(x) = x^4 + 2x^3 + 5x^2 - 1$  θεωρούμε τους συντελεστές ως διάνυσμα και γράφουμε

```
>> P=[1 2 5 0 -1]
```

P =

```
1      2      5      0     -1
```

Παρατηρούμε ότι δεν υπάρχει ο πρωτοβάθμιος όρος οπότε θέσαμε μηδέν στην αντίστοιχη θέση.

## 6. Βασικές συναρτήσεις

Ενδέχεται μία παράσταση που εκχωρούμε σε μια μεταβλητή να περιέχει εκτός από αριθμούς και άλλες βασικές συναρτήσεις.

Κάποιες συναρτήσεις βιβλιοθήκης που μπορεί να αναγνωρίσει το MATLAB μεταξύ άλλων πολλών είναι:

1. η συνάρτηση **sqrt(x)** (δηλ. η τετραγωνική ρίζα αριθμού x)
2. η εκθετική συνάρτηση **exp(x)** (δηλ. η  $e^x$ )
3. η λογαριθμική συνάρτηση **log(x)** (δηλ. η  $\ln x$ )
4. η λογαριθμική συνάρτηση **log10(x)** (δηλ. η  $\log x$ )
5. η απόλυτη τιμή ενός αριθμού **abs(x)**
6. πραγματικό μέρος ενός μιγαδικού αριθμού **real(z)**
7. φανταστικό μέρος ενός μιγαδικού αριθμού **imag(z)**
8. Οι τριγωνομετρικές συναρτήσεις με το όρισμα σε ακτίνια **sin(x)**, **cos(x)**, **tan(x)**
9. Οι αντίστροφες τριγωνομετρικές συναρτήσεις με το όρισμα σε ακτίνια **asin(x)**, **acos(x)**, **atan(x)**
10. Οι υπερβολικές συναρτήσεις **sinh(x)**, **cosh(x)**, **tanh(x)**
11. Οι αντίστροφες υπερβολικές συναρτήσεις **asinh(x)**, **acosh(x)**, **atanh(x)**

### Παραδείγματα

Εισάγουμε τριγωνομετρικές σε ακτίνια και το MATLAB επιστέφει μοίρες

```
>> A=cos(pi)
```

```
A =
```

```
-1
```

```
>> A=cos(2*pi)
```

```
A =
```

```
1
```

Μπορούμε να εισάγουμε αντίστροφες τριγωνομετρικές συναρτήσεις με όρισμα μεταξύ των τιμών [-1,1] και το MATLAB να επιστρέψει μοίρες.

```
>> b=acosd(1)
```

```
b =
```

```
0
```

```
>> b=acosd(-1)
```

```
b =
```

```
180
```

```
>> Y = asind(1)
```

```
Y =
```

```
90
```

```
>> Y = asind(-1)
```

```
Y =
```

```
-90
```



Μέσω της εντολής βοήθειας (help) του MATLAB μπορείτε να δείτε περισσότερα παραδείγματα για τη λειτουργία των συναρτήσεων και εν γένει των εντολών του MATLAB πληκτρολογώντας την εντολή help

```
>> help exp
```

```
exp - Exponential
```

This MATLAB function returns the exponential `ex` for each element in array `X`.

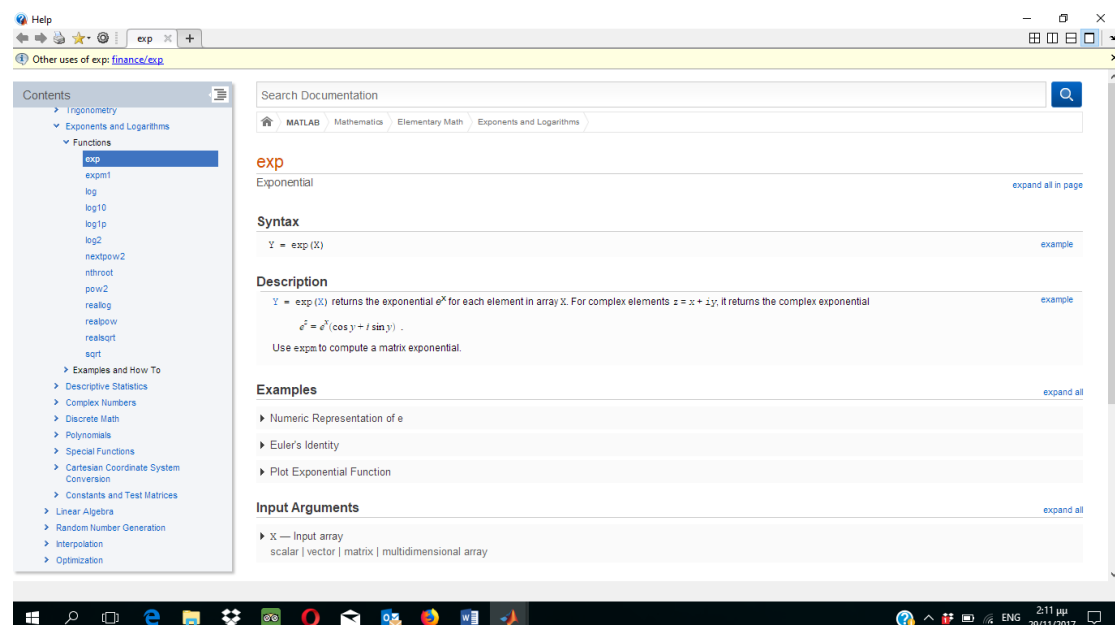
```
Y = exp(X)
```

Reference page for `exp`

See also `expint`, `expm`, `expml`, `log`, `log10`, `mpower`, `power`

Other uses of `exp`

`finance/exp`



### Άσκηση 1

Να εισαχθεί ο παρακάτω πίνακας A στο MATLAB και να γίνουν οι υπολογισμοί.

$$\begin{bmatrix} 1.2500 & 0.9500 & 0.8000 \\ 1.3200 & 0.9900 & 1.0000 \\ 1.1000 & 0.8800 & 0.7500 \\ 1.2200 & 0.6600 & 0.4500 \end{bmatrix}$$

1. Να βρεθεί ο ανάστροφός του  $A'$
2. Να γίνει η πράξη  $B = 0.5 * A * 3 * (A' - 0.12)$
3. Να γίνει η πράξη  $B = 0.5 * A + 3 * (A' - 0.12)$ . Τι παρατηρείτε;
4. Από τον πίνακα B να εξαχθούν σε πίνακα C στήλες 2, 3, 4
5. Από τον πίνακα C να επιλεγεί το στοιχείο  $d$  της 2<sup>ης</sup> γραμμής – 1<sup>ης</sup> στήλης και να γίνει η πράξη  $apot = (d - 4) - 0.488$ .
6. Να υπολογιστεί  $sinim = \frac{\sigma_{\nu}(apot) + \eta_{\mu}(apot)}{\sinh^2(apot)}$

### Λύση

1. Εισαγωγή πίνακα και εύρεση ανάστροφου

```
>> A=[1.25 0.95 0.8; 1.32 0.99 1; 1.1 0.88 0.75; 1.22 0.66 0.45];  
>> A=[1.25 0.95 0.8; 1.32 0.99 1; 1.1 0.88 0.75; 1.22 0.66 0.45]
```

A =

```
1.2500    0.9500    0.8000  
1.3200    0.9900    1.0000  
1.1000    0.8800    0.7500  
1.2200    0.6600    0.4500
```

```
>> D=A'
```

D =

```
1.2500    1.3200    1.1000    1.2200  
0.9500    0.9900    0.8800    0.6600  
0.8000    1.0000    0.7500    0.4500
```

## 2. εισαγωγή και υπολογισμός της πράξης

```
>> B=(0.5*A)*3*(D-0.12)
```

B =

4.1175	4.5458	3.6765	3.2280
4.4899	4.9880	4.0140	3.4749
3.7251	4.1184	3.3290	2.8991
3.3486	3.6513	2.9711	2.7704

## 3. εισαγωγή και υπολογισμός της πράξης –εύρεση σφάλματος

```
>> B=0.5*A+3*(D-0.12)
```

```
Error using +  
Matrix dimensions must agree.
```

## 4. Εξαγωγή στηλών

```
>> C=B(:,2:4)
```

C =

4.5458	3.6765	3.2280
4.9880	4.0140	3.4749
4.1184	3.3290	2.8991
3.6513	2.9711	2.7704

## 5. Επιλογή στοιχείου και πράξη

```
>> d=C(2,1)
```

```
d =
```

```
4.9880
```

```
>> apot=(d-4)-0.488
```

```
apot =
```

```
0.5000
```

## 6. Υπολογισμός sinim

```
sinim=(cos(apot)+sin(apot))/(sinh(apot))^2
```

```
sinim =
```

```
4.9985
```

## 7. Εξαγωγή Δεδομένων - Γραφήματα

### 7.1 Εκτύπωση δεδομένων στην οθόνη

Ο πιο απλός τρόπος για να τυπώσουμε τα αποτελέσματα της MATLAB στο παράθυρο εντολών είναι να χρησιμοποιήσουμε την εντολή **disp**

Η εντολή disp παραλείπει το όνομα της μεταβλητής και το '=' και εκτυπώνει την τιμή της συνάρτησης σύμφωνα με την τρέχουσα μορφή (format)

Προκειμένου να εξάγουμε και να οπτικοποιήσουμε τα δεδομένα που δημιουργούνται από μια σειρά εντολών στη MATLAB μπορούμε να γράψουμε απλώς το όνομα της μεταβλητής ή να μη χρησιμοποιήσουμε ελληνικό ερωτηματικό στο τέλος της εντολής που την υπολογίζει. Παρακάτω εμφανίζεται ο αριθμός  $\pi$ . Επιπλέον χρησιμοποιώντας την εντολή `format` εμφανίζεται ο αριθμός  $\pi$  με δύο διαφορετικές μορφές.

```
>> disp(pi)
3.1416
>> format long
>> disp(pi)
3.141592653589793
>> format short
>> disp(pi)
3.1416
```

### Η εντολή **fprintf**

Η εντολή **fprintf** χρησιμοποιείται αντί της `disp` στην όταν θέλουμε να εκτυπώσουμε δεδομένα σε πιο πολύπλοκη **μορφή** (format) σε αρχείο ή στην οθόνη.

Συντάσσεται ως εξής

`fprintf(fid, format, A, B, ....)`

Το `fid` δηλώνει την ταυτότητα του αρχείου που θα εκτυπωθούν τα δεδομένα.

Η απλή μορφή (format) που εκτυπώνεται μια μεταβλητή που μπορεί να περιέχεται στην εντολή `fprintf` είναι όπως στο παρακάτω παράδειγμα:

`%12.5f`

`%` (έναρξη προδιαγραφής)\_12.5(μήκος πεδίου.ακρίβεια)\_f (πραγματικός αριθμός)

Με την `fprintf` μπορούμε να εκτυπώσουμε δεδομένα και τα οποία να εμφανίζονται μέσα σε συγκεκριμένη φράση ως εξής:

```
>> a=9;
>> b=sqrt(a);
>> fprintf('The square root is %6.3f \n', b)
The square root is 3.000
```

Με αυτή τη σύνταξη της εντολής τυπώνει την τετραγωνική ρίζα αριθμού και αλλάζει γραμμή

Με την fprintf μπορούμε να σε διάφορες μορφές να τυπώσουμε τα στοιχεία ενός πίνακα A.

### Παράδειγμα

Να εκτυπωθούν σε μορφή πίνακα οι τιμές της  $f(x) = x^2$  για  $x = 0, 0.5, 1, 1.5, \dots$

```
>> x=0:0.5:4;
>> Y=[x; x.^2];
>> fprintf('%5.1f %8.3f \n',Y);
 0.0    0.000
 0.5    0.250
 1.0    1.000
 1.5    2.250
 2.0    4.000
 2.5    6.250
 3.0    9.000
 3.5   12.250
 4.0   16.000
```

### **Η εντολή input**

Η εντολή input έχει τη γενική μορφή

**X=input('ekfrasi') ή X=input('ekfrasi', 's')**

όπου **ekfrasi** ένα αλφαριθμητικό. Με την εντολή αυτή εμφανίζεται στην οθόνη η “ekfrasi” και το σύστημα αναμένει από τον χρήστη να εισαγάγει την τιμή της μεταβλητής X που μπορεί να είναι αριθμός ή αλφαριθμητικό ή διάνυσμα ή πίνακας ή ακόμα το αποτέλεσμα μιας ολόκληρης παράστασης σε γλώσσα MATLAB.

### Παράδειγμα

```
>> X=input('dose timh: ')
Dose timh: 5.3
X =
```

### 5.3

```
>> Eponymo=input('dose eponymo: ','s')
```

```
Dose eponymo: papadopoulos
```

```
Eponymo =  
Papadopoulos
```

```
>> disp(Eponymo)  
Papadopoulos
```

Η input μας δίνει την επιλογή να αλλάξουμε και γραμμή χρησιμοποιώντας το σύμβολο **\n**. Αν θέλουμε να γράψουμε μια πρόταση σε 2 γραμμές τότε η εντολή μας θα πρέπει να είναι της μορφής:

**X=input('line 1 \n line 2 ')**

#### Παράδειγμα 1.5.10

```
>> X=input('Enter \n Full name: ', 's')  
Enter  
Full name: George Papadopoulos  
X =  
George Papadopoulos
```

#### Άσκηση 2

Να υπολογιστεί το εμβαδόν επιφανείας και ο όγκος ενός στερεού ορθογωνίου παραλληλογράμμου και τα αποτελέσματα να εκτυπωθούν στην οθόνη συνοδευόμενα από κατάλληλη επεξήγηση. Για την εισαγωγή των δεδομένων να εκτυπώνεται η κατάλληλη φράση.

#### Λύση

```
>> Lx=input('Μήκος ορθογωνίου Lx ');  
>> Ly=input('Πλάτος ορθογωνίου Ly ');  
>> Lz=input('Ύψος ορθογωνίου Lz ');  
>> E=2*(Lx*Ly+Ly*Lz+Lz*Lx)  
>> V=Lx*Ly*Lz;  
>> fprintf('διαστάσεις στερεού ορθ. παραλληλογράμμου %6.2f %6.2f %6.2f \n', Lx,Ly,Lz);  
>> fprintf('το εμβαδόν του στερεού ορθ. παραλληλογράμμου είναι %6.2f και ο όγκος %6.2f \n', E,V);
```

### Άσκηση

Θα εκτυπωθούν σε στήλες οι τιμές του ημιτόνου για γωνίες από 0 έως 360 μοίρες με βήμα 30 μοίρες. Η 1η στήλη να περιέχει τις γωνίες και η 2<sup>η</sup> στήλη τις τιμές σε ακτίνια

### Λύση

```
>> x1=0:30:360;  
>> x2=x1*pi/180;  
>> y=sin(x2);  
A=[x1;y];  
fprintf('%6.2f %12.8f \n', A)
```

```
0.00    0.00000000  
30.00    0.50000000  
60.00    0.86602540  
90.00    1.00000000  
120.00    0.86602540  
150.00    0.50000000  
180.00    0.00000000  
210.00   -0.50000000  
240.00   -0.86602540  
270.00   -1.00000000  
300.00   -0.86602540  
330.00   -0.50000000  
360.00   -0.00000000
```

## 7.2 Γραφήματα

Για την παρουσίαση των δεδομένων μας με γραφήματα η MATLAB έχει πολλές δυνατότητες. Η εντολή που μας δίνει τη δυνατότητα να παρουσιάσουμε τα εξαγόμενα δεδομένα μορφή γραφήματος είναι η **plot**. Χρησιμοποιείται για την κατασκευή γραφήματος καμπύλης σε άξονες x-y.

### Παράδειγμα

Για τα διανύσματα

```
>> X=[1 2 3 4 5 6 7 8 9 10]
```

```
X =
```



```

1      2      3      4      5      6      7      8      9      10

>> Y=[0.3 0.4 0.5 0.7 0.75 0.8 0.85 0.88 0.9 1.1]

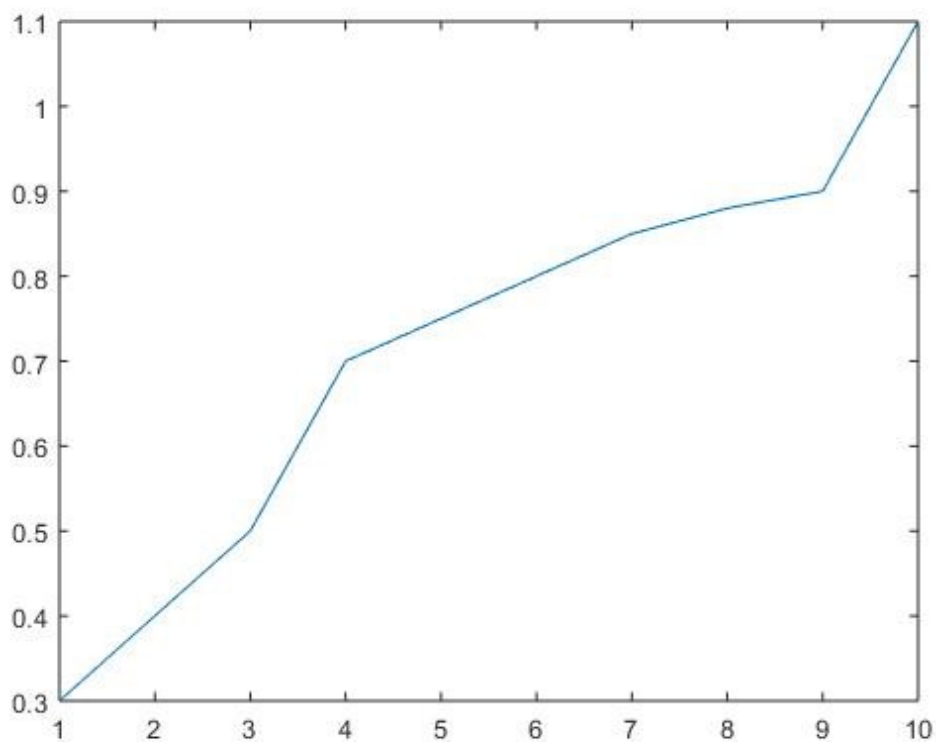
Y =

    0.3000    0.4000    0.5000    0.7000    0.7500    0.8000    0.8500
    0.8800    0.9000    1.1000

>> plot(X,Y)
X=[1 2 3 4 5 6 7 8 9 10] και Y=[0.3 0.4 0.5 0.7 0.75 0.8 0.85 0.88 0.9
0.1]

```

Εμφανίζεται το γράφημα



Εικόνα 1

Μπορούμε να κάνουμε και διαγράμματα συναρτήσεων όπως για παράδειγμα η συνάρτηση ημιτονο ( $\sin x$ ) στο διάστημα  $[-\pi, \pi]$  με βήμα 0.1

Κατασκευάζουμε το διάστημα  $[-\pi, \pi]$  και το χωρίζουμε σε υποδιαστήματα μήκους 0.01 και αποθηκεύουμε τα σημεία  $x_i$  στο διάνυσμα  $x$

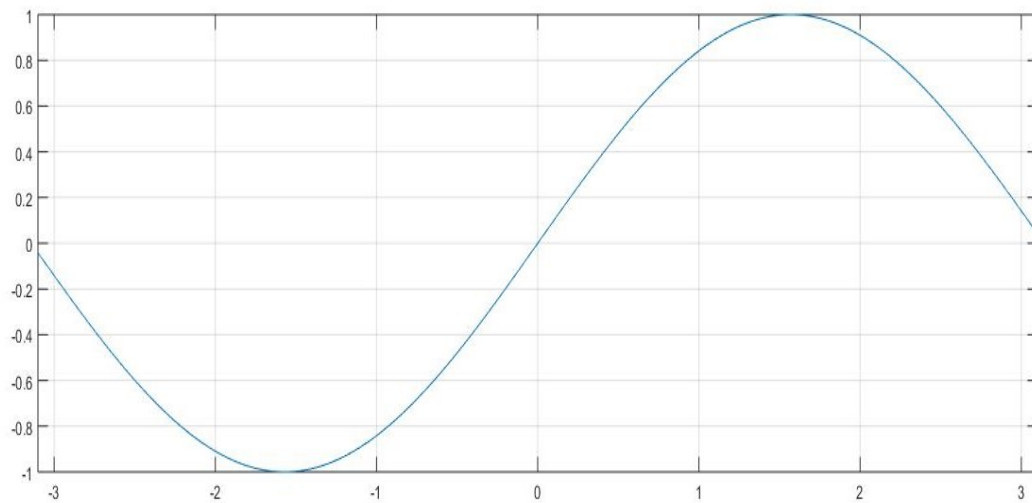
```
x=-pi:0.01:pi;
```

ορίζουμε το ημίτονο

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

Κατασκευάζουμε το γράφημα της  $y = \sin(x)$  στο διάστημα  $[-\pi, \pi]$

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



Εικ.2

Όπως αναφέραμε και προηγούμενα μπορούμε αν χρησιμοποιήσουμε την εντολή **linspace** για την εισαγωγή του αριθμού των σημείων που επιθυμούμε για το γράφημά μας

Στην περίπτωσή μας θα γράφαμε εάν θέλαμε το γράφημα της συνάρτησης ημίτονο να έχει πλήθος σημείων  $x_i$  εκατό

```
x=linspace(0,2*pi,100);
```

Συνοψίζοντας για να σχεδιάσουμε τη συνάρτηση  $\sin x$  θα γράφαμε

```
>> x=linspace(0,2*pi,100);
```

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

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

Ετικέτες και τίτλοι στα γραφήματα

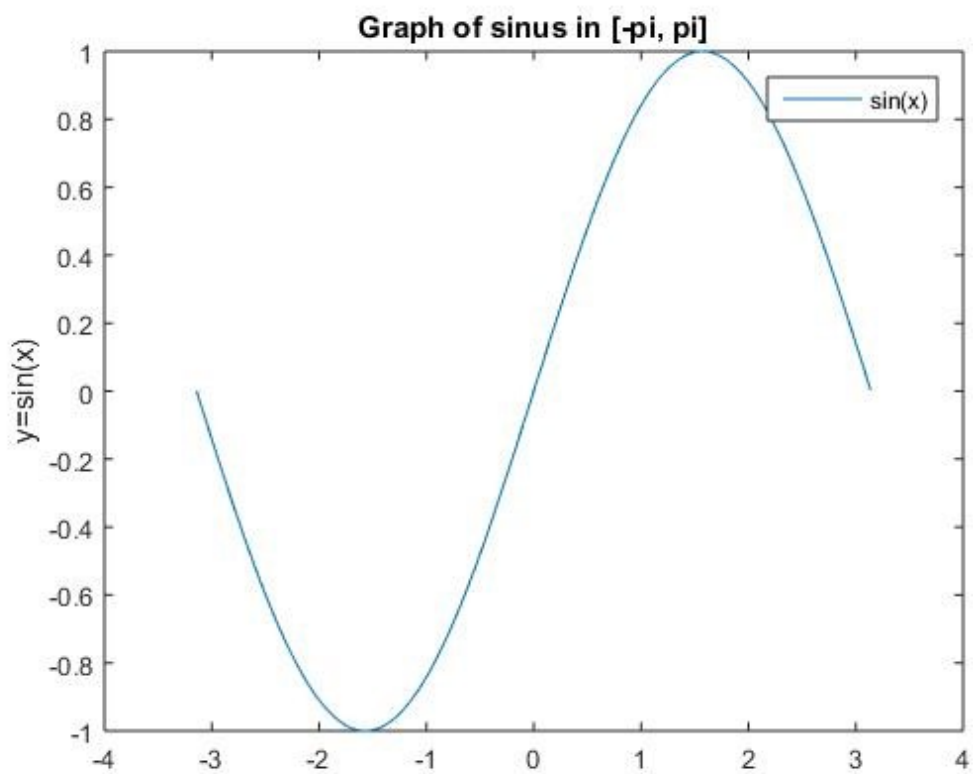
Μπορούμε στα γραφήματα να τοποθετήσουμε ετικέτες στους άξονες τίτλο και λεζάντα με τις παρακάτω εντολές

```

>> x=-pi:0.01:pi;
>> y=sin(x);
>> plot(x,y);
>> ylabel('sin(x)')
>> xlabel('time')
>> title('Graph of sinus in [-pi, pi]')
>> legend('sin(x)');

```

Το γράφημα που παίρνουμε είναι το εξής:



Εικόνα 3

Επιπλέον μπορούμε να εμπλουτίσουμε το γράφημα με χρώματα, σύμβολα και ιδιαίτερου τύπου γραμμές εκχωρώντας τα στοιχεία εντός της εντολής plot ως εξής:

```

>> plot(x,y, ' [color][style][ltype]' )

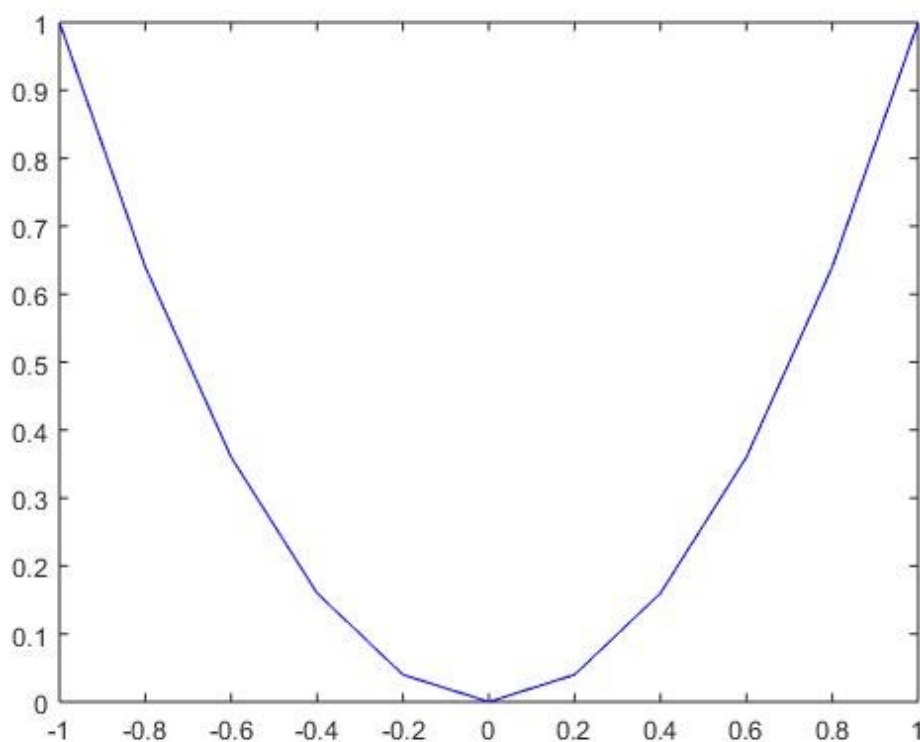
```

Όπου [color] ο τύπος χρώματος, [style] σύμβολο της τιμής και [ltype] ο τύπος της γραμμής. Στους παρακάτω πίνακες φαίνονται οι κυριότεροι από αυτούς τους τύπους.

[color]	Χρώμα	[style]	Σύμβολο	[ltype]	Τύπος γραμμής
b	μπλε	.	τελεία	-	συνεχής
g	πράσιν ο	o	κύκλος	:	λεπτή διακεκομμένη
r	κόκκινο	x	χι	--	αδρή διακεκομμένη
c	κυανό	+	συν	-. .	διακεκομμένη- τελείες
m	μωβ	*	αστερίσκος		
y	κίτρινο	s	τετράγωνο		
k	μαύρο	d	ρόμβος		
w	άσπρο				

Με τις παρακάτω εντολές σχεδιάζεται η μεταβλητή  $y = x^2$  στο διάστημα  $[-1, 1]$  με βήμα 0.2.

```
>> x = -1:0.2:1;
>> y=x.^2;
>> plot(x,y, 'b-')
```



Εικόνα 5

### Η εντολή **fplot**

Εντολή που μας επιτρέπει να σχεδιάσουμε συνάρτηση με υποχρεωτική εισαγωγή του πεδίου σχεδίασης του γραφήματος. Η συνάρτηση μπορεί να είναι μία από τις ήδη δομημένες συναρτήσεις του MATLAB είτε μια συνάρτηση που μπορεί να εισάγει ο ίδιος ο προγραμματιστής. Ο τρόπος εισαγωγής της εντολής είναι

**fplot(f, [xmin, xmax])**

Στα παρακάτω εισάγονται α) η γραφική παράσταση της συνάρτησης ημιτόνου στο διάστημα  $[-2\pi, 2\pi]$

```
>> fplot('sin(x)', [-2*pi, 2*pi])
```

Β) η γραφική παράσταση της συνάρτησης  $f(x) = x^2$  στο διάστημα  $[-2, 2]$

```
>> f = @(x) x.^2;  
>> fplot(f, [-2, 2])
```

Το σύμβολο **@ (handler)** μας επιτρέπει να εισάγουμε συνάρτηση μίας ή περισσότερων μεταβλητών και να την εκχωρήσει το MATLAB σε μια μεταβλητή.

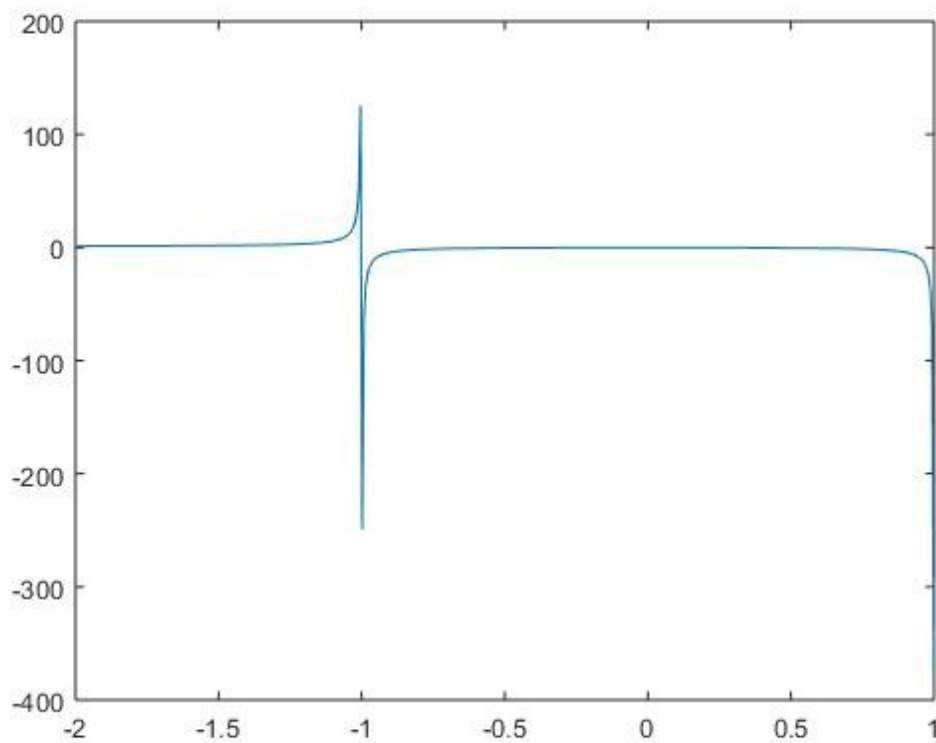
Για παράδειγμα η συνάρτηση  $f(x) = \frac{x^2}{x^2-1}$  μπορεί να εισαχθεί απευθείας με **@** ως εξής:

```
>> f = @(x) x.^2./(x.^2-1);
```

- Το σύμβολο **@** δηλώνει ότι η  $f$  είναι συνάρτηση.
- Μετά το σύμβολο **@** ακολουθεί η μεταβλητή εισόδου της συνάρτησης ( $x$ ).
- Ορισμός της συνάρτησης με κενό μετά τη μεταβλητή εισόδου.

Το γράφημα αυτής στο διάστημα  $[-2, 1]$  είναι:

```
>> fplot(f, [-2, 1])
```



Εικόνα 6.

### Η εντολή **comet**

Με την εντολή **comet** το MATLAB μας παρέχει τη δυνατότητα να δούμε ζητούμενο γράφημα εν κινήσει με τρόπο ώστε να διαγράφεται η τροχιά της ζητούμενης μορφής της καμπύλης

```
x=-pi:0.01:pi;
y=sin(x);
>> comet(x,y)
```

Το μειονέκτημα της εντολής είναι ότι δεν εκτυπώνεται το Γράφημα ούτε αποθηκεύεται.

### Πολλαπλά γραφήματα με **subplot**

Υπάρχουν περιπτώσεις που επιθυμούμε να τοποθετηθούν πολλαπλά γραφήματα και να φαίνονται στην οθόνη το ένα δίπλα στο άλλο σαν σε μορφή πίνακα. Το αποτέλεσμα αυτό μπορεί να μας το δώσει η εντολή **subplot**. Στο παρακάτω παράδειγμα τοποθετούνται τέσσερις γραφικές παραστάσεις σε ένα παράθυρο.

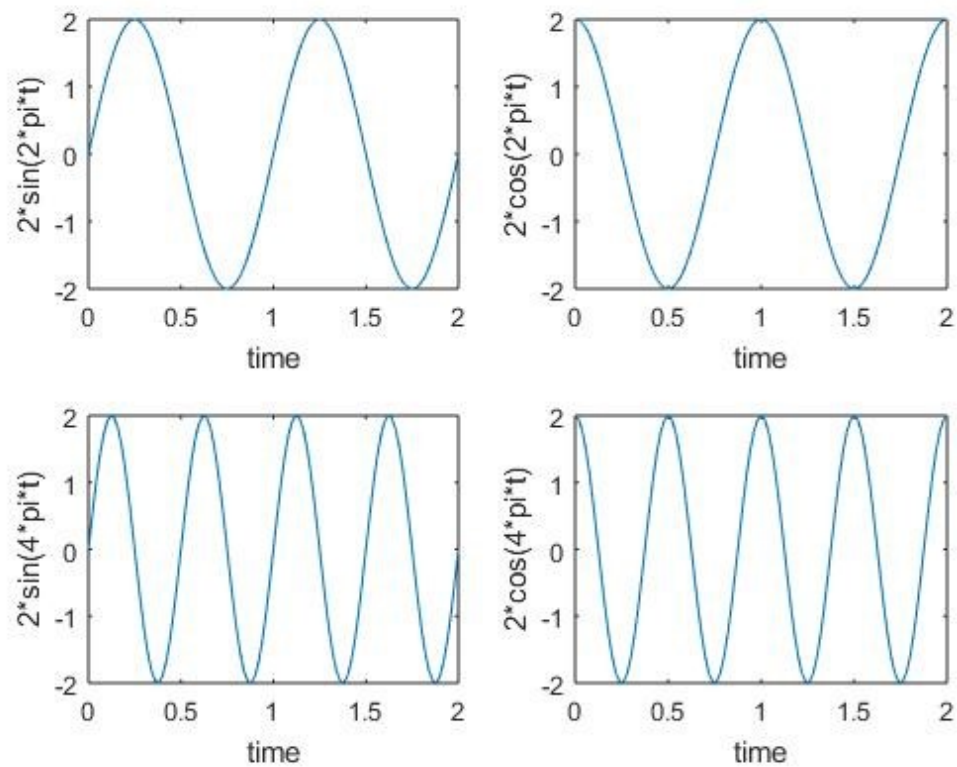
```
>> t=linspace(0,2,101);
>> y1=2*sin(2*pi*t);
>> y2=2*cos(2*pi*t);
>> y3=2*sin(4*pi*t);
```

```

>> y4=2*cos(4*pi*t);
>> subplot(2,2,1)
>> plot(t,y1)
>> xlabel('time'),ylabel('2*sin(2*pi*t)')
>> subplot(2,2,2)
>> plot(t,y2)
>> xlabel('time'),ylabel('2*cos(2*pi*t)')
>> subplot(2,2,3)
>> plot(t,y3)
>> xlabel('time'),ylabel('2*sin(4*pi*t)')
>> subplot(2,2,4)
>> plot(t,y4)
>> xlabel('time'),ylabel('2*cos(4*pi*t)')

```

Το αποτέλεσμα των εντολών δίνεται από τα παρακάτω γραφήματα



## Μέρος 2ο

### Εισαγωγή στον Προγραμματισμό



## 1.Αρχεία με κατάληξη .m (m-files)

Συνήθως όταν γράφουμε απλές πράξεις ή κάνουμε απλούς υπολογισμούς γράφουμε τις εντολές στο χώρο του υποπαραθύρου Command Line. Όταν όμως πρέπει να γραφτούν πολυπλοκότερες εντολές με είναι καλό να γράφονται σε αρχεία τύπου editor. Τα αρχεία αυτά είναι της μορφής `ονομα_αρχείου.m`, «τρέχουν» σε περιβάλλον MATLAB και ονομάζονται m-files.

Υπάρχουν δύο κατηγορίες m-files:

- α. Τα αρχεία εντολών (command files) τα οποία εκτελούν μια ακολουθία εντολών που μπορεί να οδηγούν σε συγκεκριμένο αποτέλεσμα.
- β. Τα αρχεία συναρτήσεων (function m-files) τα οποία αποτελούνται από τη γραμμή ορισμού μιας συγκεκριμένης συνάρτησης, δέχονται δεδομένα (τιμές) εισόδου και επιστρέφουν το αποτέλεσμα της προγραμματισμένης συνάρτησης.

Για τη δημιουργία ενός m-file

1. Πληκτρολογούμε στο χώρο Command Line την εντολή `edit` ή
2. Στη γραμμή εργαλείων του MATLAB ακολουθούμε τη διαδρομή

**File → New → M-file.**

3. Εντός του χώρου πληκτρολογούμε τη σειρά εντολών.
4. Αποθηκεύουμε το m-file.

Προσοχή! Για να χρησιμοποιήσουμε ένα m-file πρέπει να αποθηκεύεται στο φάκελο εργασίας μας (working directory).

### Παράδειγμα

Στο παρακάτω παράδειγμα δίνεται η λύση ενός συστήματος 3x3 με ορίζουσες. Οι εντολές είναι καταχωρημένες σε αρχείο `sys3x3.m`

```
>> A = [1 3 0;2 2 4;0 1 2];  
b = [2;1;2];  
disp('Ο pivakas A divetai apo'),A  
disp('H orizousa tou pivaka A eivai')  
detA=det(A)
```

```

disp('To diavusma b divetai apo'),b
disp('H lush tou susthmatos Ax=b eivai')
x = A\b

```

Εάν πληκτρολογήσουμε sys3x3.m στο Command lime τότε παρουσιάζεται η λύση του προβλήματος

Ο pivakas A divetai apo

A =

```

1      3      0
2      2      4
0      1      2

```

Η orizousa tou pivaka A eivai

detA =

-12

Το diavusma b divetai apo

b =

```

2
1
2

```

Η lush tou susthmatos Ax=b eivai

x =

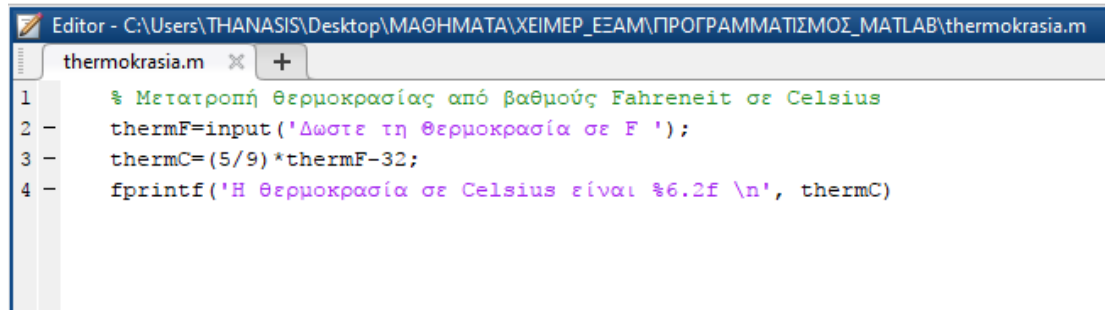
```

-1.5000
1.1667
0.4167

```

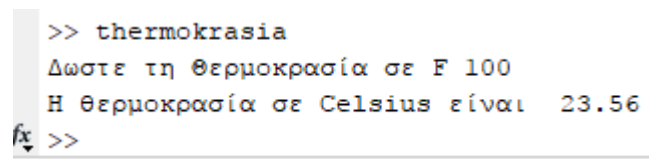
### Παράδειγμα

Να γραφτεί πρόγραμμα που να μετατρέπει τη θερμοκρασία από βαθμούς Fahrenheit σε βαθμούς Celsius



```
Editor - C:\Users\THANASIS\Desktop\ΜΑΘΗΜΑΤΑ\ΧΕΙΜΕΡ_ΕΞΑΜ\ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ_MATLAB\thermokrasia.m
thermokrasia.m
1 % Μετατροπή θερμοκρασίας από βαθμούς Fahrenheit σε Celsius
2 - thermF=input('Δώστε τη θερμοκρασία σε F ');
3 - thermC=(5/9)*thermF-32;
4 - fprintf('Η θερμοκρασία σε Celsius είναι %6.2f \n', thermC)
```

Εικ 8. Αρχείο εντολών μετατροπής θερμοκρασίας



```
>> thermokrasia
Δώστε τη θερμοκρασία σε F 100
Η θερμοκρασία σε Celsius είναι 23.56
fx >>
```

Εικ 9. εκτελώντας το πρόγραμμα thermokrasia.m

## 2.Αρχεία συνάρτησης (function)

Ο ορισμός αρχείων συνάρτησης στο MATLAB περιέχει μεταβλητές εισόδου (input) και μέσα από τη διαδικασία εντολών που εμείς θέτουμε παράγει αποτελέσματα εξόδου (output). Για το λόγο αυτό πριν ξεκινήσουμε να σχεδιάσουμε ένα αρχείο συνάρτησης πρέπει να γνωρίζουμε τον ακριβή αριθμό μεταβλητών εισόδου και των μεταβλητών - αποτελεσμάτων εξόδου.

Η Δομή μιας συνάρτησης είναι η εξής:

*A. Επικεφαλίδα*

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

Το onomasinartisis είναι το όνομα του αρχείου συνάρτησης με το οποίο υποχρεωτικά αποθηκεύεται το m-file που θα δημιουργηθεί με την ονομασία onomasinartisis.m. Οι μεταβλητές εισόδου (input) περικλείονται σε παρενθέσεις ενώ οι μεταβλητές - αποτελέσματα εξόδου (output) περικλείονται σε αγκύλες.

### *B. Σχόλια*

Τα σχόλια περιγράφουν το περιεχόμενο και τη λειτουργία της συνάρτησης. Πριν από κάθε σχόλιο τίθεται το σύμβολο % και το χρώμα της γραμματοσειράς γίνεται πράσινο. Τα σχόλια δεν επηρεάζουν την ομαλή εξέλιξη της ροής του προγράμματος σε όποια θέση κι αν βρίσκονται μέσα σε αυτό.

### *Γ. Εντολές*

Οι εντολές αποτελούν το κύριο μέρος της συνάρτησης η σειρά των οποίων δίνει τη λογική ροή της λειτουργίας της συνάρτησης κάνοντας χρήση των δεδομένων εισόδου που εμείς θέσαμε αρχικά.

### Παράδειγμα

Να γραφτεί συνάρτηση που να υπολογίζει το άθροισμα, το γινόμενο και το μέσο όρο πέντε αριθμών.

Το πρόγραμμα θα έχει την εξής δομή:

```
function [sum, gin, mo] = sumgin (x1, x2, x3, x4, x5)
% function [sum, gin, mo] = sumgin (x1, x2, x3, x4, x5)
%
% Paradeigma function m-file
% Υπολογίζει το athroisma, to ginomeno kai to meso oro pente arithmwn
%
% Onoma function : sumgin
% Onoma m-file : sumgin.m
% Metablhtes eisodou: x1, x2, x3, x4, x5
% Metablhtes exodou : sum (athroisma tw'n x1, x2, x3, x4, x5)
% prod (ginomeno tw'n x1, x2, x3, x4, x5)
% mo (mesos oros x1, x2, x3, x4, x5)
sum=x1+x2+x3+x4+x5;
gin=x1*x2*x3*x4*x5;
mo=(x1+x2+x3+x4+x5)/5;
% Telos tou sumgin.m
```

Για να καλέσουμε τη συνάρτηση πρέπει να βάλουμε τους πέντε αριθμούς δεξιά με το όνομα της συνάρτησης και αριστερά της ισότητας τις μεταβλητές που θα καταχωρηθεί το αποτέλεσμα

```
>> [s,g,m] = sumgin(1, -3, 4, 1,-7)
```

```
s =
```

```
-4
```

```
g =
```

-18

m =

-0.8000

Στο επόμενο παράδειγμα δημιουργείται ένα **αρχείο συνάρτησης** με το οποίο υπολογίζεται το εμβαδόν παράπλευρης επιφάνειας και ο όγκος στερεού ορθογωνίου παραλληλογράμμου .

```
function [emvadon,ogkos]=parallilof(Lx,Ly,Lz)
% πρόγραμμα υπολογισμού εμβαδου και όγκου
% στερεού ορθογωνίου παραλληλογράμμου
% Lx=μήκος
% Ly=πλάτος
% Lz=ύψος
% emvadon=εμβαδόν
% ogkos= όγκος
emvadon=2*(Lx*Ly+Ly*Lz+Lz*Lx);
ogkos=Lx*Ly*Lz;
end
```

για να υπολογίσουμε το εμβαδόν και τον όγκο χρησιμοποιώντας τη συνάρτηση αρκεί να την καλέσουμε με τα ορίσματά της και να ορίσουμε τις τιμές.

```
[E,V]=parallilof(2,3,4)
```

E =

52

V =

24

Προσοχή: το όνομα μιας νέας συνάρτησης δεν πρέπει να συμπίπτει με όνομα συνάρτησης βιβλιοθήκης του MATLAB, όπως οι *sin*, *cos*, *tan*, *max*, *exp* κπλ . Γενικά

για την ονομασία μιας συνάρτησης ακολουθούμε τους κανόνες ονοματολογίας που ισχύουν και για τις μεταβλητές.

### Μαθηματικές συναρτήσεις

Ένας χρήσιμος τύπος συνάρτησης είναι οι μαθηματικές συναρτήσεις. Μπορούμε εύκολα να ορίσουμε μια συνάρτηση και να βρίσκουμε τις τιμές της για συγκεκριμένα x. Στο παρακάτω παράδειγμα ορίζεται η συνάρτηση

$$f_1 = \frac{3x^4}{x^2 - 1}$$

```
function [F] = f1(x)
F = 3*x^4/(x^2-1);
% Τελος του f1.m
```

Την οποία καλούμε για παράδειγμα x=2

```
>> f1(2)
```

```
ans =
```

```
16
```

## 3.Βασικές εντολές προγραμματισμού

Προκειμένου να δημιουργήσουμε ένα ολοκληρωμένο πρόγραμμα με πρόσθετες δυνατότητες στο MATLAB είναι απαραίτητο να κάνουμε χρήση των δομών ελέγχου ροής (flow control structures). Οι δομές αυτές αποτελούνται από τους βρόχους for (for loops) και while (while loops). Εκτός από αυτούς τους βασικούς βρόχους κύριο ρόλο στο προγραμματισμό παίζει και η εντολή if.

### 3.1 βρόχοι for (for loops)

Ένας βρόχος for δομείται ως εξής :

```
for i = arxikitimi: vima : telikitimi
entoli_1
entoli_2
```

```
.
```

```

.
.
entoli_n
end

```

Το γράμμα  $i$  είναι ο μετρητής και παίρνει την αρχική τιμή που του έχουμε δηλώσει στην `arxikitimi` (αρχή του βρόγχου) μέχρι και την `telikitimi` (τέλος του βρόγχου) με κάποιο προκαθορισμένο βήμα (μεταβλητή `vima`). Οι τιμές αυτές διαχωρίζονται με άνω και κάτω τελεία (`:`) η οποία υποδηλώνει το διάστημα των τιμών από...έως.

Αν παραλείψουμε το βήμα, τότε η MATLAB χρησιμοποιεί σαν βήμα το 1. Μπορεί το βήμα να είναι αρνητικός αριθμός.

### Παράδειγμα

Για τον υπολογισμό της παράστασης

$$(i + 1) \cdot 2$$

Γράφουμε

```

>> for j=1:4
(j+1).*2
end

```

```
ans =
```

```
4
```

```
ans =
```

```
6
```

```
ans =
```

```
8
```

```
ans =
```

```
10
```

Με το βρόχο `for` μπορούμε να κάνουμε και πράξεις εντός των στοιχείων διανυσμάτων, εάν θέλουμε για παράδειγμα να υψώσουμε στο τετράγωνο τα στοιχεία του διανύσματος  $\alpha=(1,2,3,4,5,6,7,8)$  γράφουμε

```

>> for k=1:8;
y(k) = k^2;
end

```

### «Εγκλωβισμένη» for

Μπορεί ένας βρόγχος for να περιέχεται σε ένα άλλο βρόχο for οπότε μιλάμε για «εγκλωβισμένους» βρόχους οι οποίοι είναι χρήσιμοι όταν πρέπει να εισάγουμε τιμές σε ένα πίνακα διαστάσεων NxM

Παρακάτω εισάγεται πίνακας 5x4 (Πέντε γραμμές X Τέσσερις στήλες).

```
>> for i=1:5
for j=1:4
A(i,j)=i*j
end
end
```

Το αποτέλεσμα της σειράς εντολών είναι

A =

1	2	3	4
2	4	6	8
3	6	9	12
4	8	12	16
5	10	15	20

Μια πολύ σημαντική χρήση του βρόγχου for είναι για τον υπολογισμό αθροισμάτων της μορφής

$$\sum_{i=1}^n a_n$$

Επίσης ο υπολογισμός γινομένων της μορφής

$$\prod_{i=1}^n a_n$$

Στην περίπτωση του αθροίσματος μηδενίζουμε το άθροισμα και έπειτα κάνουμε άθροιση των υπόλοιπων όρων ως εξής:



```

athr=0;
for i=1:n
    athr=athr+a(i);
end

```

Στην περίπτωση του γινομένου θέτουμε αρχικά αυτό με ένα και κάνουμε το εξής:

```

gin=1;
for i=1:n
    gin=gin*a(i);
end

```

Έστω ότι θέλουμε να υπολογίσουμε το άθροισμα

$$\sum_{i=1}^3 (i+2)^i$$

Αυτό γράφεται ως εξής:

```

>> athr=0;
>> for i=1:3
    athr=athr+(i+2)^i
end

```

με αποτέλεσμα για κάθε επανάληψη

```

athr =
     3

```

```

athr =
    19

```

```

athr =
   144

```

Εάν το αναλύσουμε βήμα – βήμα θα έχουμε

- Στην αρχή μηδενίζουμε το άθροισμα.
- Στη συνέχεια έχουμε τις επαναλήψεις:

1. Για  $i = 1$

$$athr = 0 + (1 + 2)^1 = 3$$

2. Για  $i = 2$

$$athr = 3 + (2 + 2)^2 = 19$$

3. Για  $i = 3$

$$athr = 19 + (3 + 2)^3 = 144$$

### Παράδειγμα

Να υψωθούν τα στοιχεία του πίνακα A στον κύβο.

```
>> A=[ 2 3 -2 ; 1 4 0 ; -2 1 8]
```

```
A =
```

```
     2     3    -2
     1     4     0
    -2     1     8
```

```
>> for i=1:3
    for j=1:3
        A2(i,j) = A(i,j)^3;
    end
end
```

## 3.2 βρόχοι while (while loops)

Πριν παρουσιάσουμε τους βρόχους while θα κάνουμε μια αναφορά σε ελέγχους που μπορούμε να κάνουμε με το MATLAB σχετικούς με τη σύγκριση τιμών ή μεταβλητών μεταξύ τους. Ένας τέτοιος έλεγχος ονομάζεται *λογικός έλεγχος* και παίρνει την τιμή ένα (1) εάν είναι αληθής ή την τιμή μηδέν (0) είναι ψευδής. Για την δημιουργία λογικών ελέγχων χρησιμοποιούμε τους λογικούς τελεστές:

<    μικρότερο από  
>    μεγαλύτερο από  
<=   μικρότερο ή ίσο  
>=   μεγαλύτερο ή ίσο  
==   ίσο

Έτσι μπορούν να συγκριθούν αριθμοί

2==5

Και να επιστρέψει ως αποτέλεσμα 1 ή 0 ανάλογα με το αν είναι αληθής η πρόταση ή όχι.

Άλλη σύγκριση που μπορεί να γίνει είναι αυτή μεταξύ στοιχείων πινάκων ή διανυσμάτων

```
>> A=[1 3 2;4 1 3;2 2 4]
```

```
A =
```

1	3	2
4	1	3
2	2	4

```
>> B=[4 3 3;1 1 2;5 2 4]
```

```
B =
```

4	3	3
1	1	2
5	2	4

```
>> A==B
```

```
ans =
```

0	1	0
0	1	0
0	1	1

```
>> A==4
```

```
ans =
```

0	0	0
1	0	0
0	0	1

Ακόμη μια σύγκριση που μπορεί να γίνει είναι ο συνδυασμός λογικών ελέγχων με το λογικό «και» (&), ή το λογικό «ή» (|)

```
>> A<3 & A>1
```

```
ans =
```

0	0	1
0	0	0
1	1	0

```
>> A<3 | A==4
```

```
ans =
```

1	0	1
1	1	0
1	1	1

Οι βρόχοι while κάνουν χρήση των παραπάνω λογικών ελέγχων και ιδίως του τελεστή «μικρότερο από». Αυτό διότι με τους βρόχους while εκτελείται μια διαδικασία που περιγράφεται με εντολές εντός του βρόγχου, έως ότου ισχύσει μια λογική συνθήκη. Η διαδικασία είναι επαναληπτική.

Η σύνταξη της εντολής είναι η εξής:

**While** συνθήκη

Εντολές

**End**

Έστω ότι θέλουμε να υπολογίσουμε το άθροισμα

$$\sum_{i=1}^n (i+1)^i < 1000$$

Βλέπουμε ότι ικανοποιεί μια συνθήκη για την οποία γράφουμε

```
>> S=0;
i=1;
while S+(i+1)^i<1000;
    S=S+(1+i)^i;
    i=i+1;
end
```

```
>> [i,S]
```

```
ans =
```

```
5    700
```

Η εντολή αφού μηδενίσει το αρχικό άθροισμα υπολογίζει τόσες επαναλήψεις μέχρι να φτάσει στην επανάληψη που το αποτέλεσμα είναι άνω του 1000. Τότε ως τελικό αποτέλεσμα δίνει τον αριθμό των επαναλήψεων και το αποτέλεσμα του αθροίσματος.

Αναλυτικότερα γίνονται τα εξής:

*1<sup>η</sup> Επανάληψη*

$$s = 0$$

$$i = 1$$

$$s = 0 + (1 + 1)^1 = 2$$

$$i = 1 + 1 = 2$$

*2<sup>η</sup> Επανάληψη*

$$s = 2$$

$$i = 2$$

$$s = 2 + (1 + 2)^2 = 2 + 9 = 11$$

$$i = 2 + 1 = 3$$

*3<sup>η</sup> Επανάληψη*

$$s = 11$$

$$i = 3$$

$$s = 11 + (1 + 3)^3 = 11 + 64 = 75$$

$$i = 3 + 1 = 4$$

*4<sup>η</sup> Επανάληψη*

$$s = 75$$

$$i = 4$$

$$s = 75 + (1 + 4)^4 = 75 + 625 = 700$$

$$i = 4 + 1 = 5$$

*5<sup>η</sup> Επανάληψη*

$$\begin{aligned}
 s &= 700 \\
 i &= 5 \\
 s &= 700 + (1 + 5)^5 = 700 + 7776 = 8476
 \end{aligned}$$

Το πρόγραμμα σταματά εδώ, διότι βρίσκει αριθμό μεγαλύτερο του 1000.

### 3.3 εντολή if (if statement)

Εάν στη ροή ενός προγράμματος παρουσιάζονται συνθήκες οι οποίες πρέπει να ελέγξουμε εάν ισχύουν χρησιμοποιούμε την εντολή if

Η εντολή συντάσσεται απλά ως εξής:

```

If σχέση
εντολές
else
εντολές
end

```

Εάν παρουσιάζονται περισσότεροι έλεγχοι τότε η εντολή συντάσσεται ως εξής

```

If σχέση_1
εντολές
elseif σχέση_2
εντολές
else
εντολές
end

```

#### Παράδειγμα

Υπολογισμός τετραγωνικής ρίζας αριθμού εάν αυτός είναι θετικός

```

>> x=input('dose arithmo x ')
>> if x>=0
y=sqrt(x)
end

```

με αποτέλεσμα

y =

3

Με εναλλακτική εντολή else

```
>> x=input('dose arithmo x ');  
if x>=0  
y=sqrt(x)  
else disp(' o arithmos einai arnhtikos')  
end
```

το παραπάνω μπορεί να γραφτεί σε αρχείο m-file με το όνομα riza.m

Ασκήσεις

1. Να γραφτεί πρόγραμμα που να υπολογίζει σε πόσα χρόνια αρχικό κεφάλαιο ποσού 10000 Ευρώ θα γίνει 20000 ευρώ με επιτόκιο 5%.
2. Να γραφτεί πρόγραμμα που να ταξινομεί έξι αριθμούς κατ' αύξουσα σειρά (ταξινόμηση φυσαλίδας).

## **Βιβλιογραφία**

Σαρρής Ι., Καρακασίδης Θ, (2014) Αριθμητικές Μέθοδοι και Εφαρμογές για Μηχανικούς, Εκδόσεις Τζιόλα

Γεωργίου Γ., Ξενοφώντος Χ., (2007) Εισαγωγή στη MATLAB.