ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 1

ΕΙΣΑΓΩΓΗ ΣΤΟ MATLAB

Το εργαστήριο **Σημάτων και Συστημάτων** διεξάγεται στο περιβάλλον του προγράμματος MATLAB με χρήση τόσο του βασικού κορμού του πακέτου που παρέχει πληθώρα έτοιμων ενσωματωμένων συναρτήσεων και πλούσια γραφικά όσο και μερικών εργαλειοθηκών (**toolboxes**) του. Σε σύντομο χρονικό διάστημα, ο χρήστης μπορεί να χρησιμοποιεί τις έτοιμες υπορουτίνες για την επίλυση προβλημάτων από διάφορες εφαρμογές.

Οι βασικοί λόγοι που έχουν καταστήσει το MATLAB ένα από τα πιό δημοφιλή επιστημονικά πακέτα λογισμικού είναι οι εξής:

α) το περιβάλλον του είναι φιλικό προς τον χρήστη
β) παρέχει άμεσες δυνατότητες γραφικής απεικόνισης
γ) έχει πληθώρα ενσωματωμένων συναρτήσεων
δ) παρέχει τη δυνατότητα προσθήκης συναρτήσεων γραμμένων από τον χρήστη
ε) ο προγραμματισμός στο MATLAB είναι απλός
στ) περιλαμβάνει πληθώρα εργαλείων από διάφορες επιστημονικές περιοχές

1. ΒΑΣΙΚΕΣ ΛΕΙΤΟΥΡΓΙΕΣ ΤΟΥ MATLAB

Για να μπούμε στο MATLAB κάνουμε **διπλό κλίκ** στο εικονίδιο του MATLAB και για να βγούμε πληκτρολογούμε **`quit`**. Το προτρεπτικό σήμα (prompt) του MATLAB είναι το **`>>`**.

Απλές αριθμητικές πράξεις

Το MATLAB χρησιμοποιεί τους τελεστές **`+`, `-`, `*`** και **`/`** για τις τέσσερις αριθμητικές πράξεις. Παραδείγματος χάριν:

Παλαμίδης Αλέξανδρος
Μαράς Ανδρέας

```
>>3/5
ans =
   0.6
>>3 + 5 + 2
ans =
   10
>>3^3
ans =
   27
```

Το MATLAB, εκτός από τον τελεστή της διαίρεσης από τα αριστερά `/`, διαθέτει και τελεστή διαίρεσης από τα δεξιά:

```
>>2/4
ans =
   0.5
>>2\4
ans =
   2
```

Για πιό πολύπλοκες εκφράσεις χρησιμοποιούνται παρενθέσεις κατά τον συνήθη τρόπο:

```
>>2^5 + 4*(33 - 2*(6+2/7))
ans =
   ................................
```

Ενσωματωμένες συναρτήσεις

Το MATLAB μας παρέχει ένα πλήθος ενσωματωμένων συναρτήσεων όπως τετραγωνική ρίζα, εκθετικές και λογαριθμικές συναρτήσεις, τριγωνομετρικές και αντίστροφες τριγωνομετρικές συναρτήσεις κ.ά.:

```
>> sqrt(2)      % τετραγωνική ρίζα
ans =
      ...................
>> exp(1)       % εκθετική συνάρτηση
ans =
      ....................
>> log(exp(1)) % φυσικός λογάριθμος
ans =
      ....................
>> log10(10^2) % δεκαδικός λογάριθμος
```

```
ans =
        ………………
 % Τριγωνομετρικές συναρτήσεις

>> pi           % η σταθερά π
ans =
        ……………….
>> sin(pi/4)        % ημίτονο
ans =
        ……………….
>> cos(pi/2)        % συνημίτονο
ans =
        …………………… ← πρακτικά το αποτέλεσμα είναι 0

>> tan(pi/4)        % εφαπτομένη
ans =
        …………………
>> asin(0.5)        % τόξο ημιτόνου
ans =
        …………………
>> atan(1)      % τόξο εφαπτομένης
ans =
        ……………………
```

όπου το σύμβολο '%' χρησιμοποιείται για την εισαγωγή σχολίων. Αν η γωνία δίνεται σε μοίρες, τότε την μετατρέπουμε σε ακτίνια πολλαπλασιάζοντας με το π/180.

Παράδειγμα υπολογισμού του **cos(60o)**:

```
>> cos(60*pi/180)
ans =
        ………………
```

Αν και μερικές φορές το αποτέλεσμα παρουσιάζει σφάλμα λόγω των αριθμητικών προσεγγίσεων των ψηφιακών Η/Υ, όπως στο παραπάνω παράδειγμα υπολογισμού του cos(pi/2) που έπρεπε να δώσει μηδέν, δεν πρέπει να γενικεύουμε και να εκλαμβάνουμε όλους του μικρούς αριθμούς ως μηδέν!

Σταθερές και μεταβλητές

Το MATLAB μας επιτρέπει να δίνουμε στις σταθερές και μεταβλητές ονόματα της επιλογής μας. Για παράδειγμα, έστω ότι θέλουμε να υπολογίσουμε το ακόλουθο:

```
sin(60*pi/180)^2 + cos(60*pi/180)^2
```

Με χρήση σταθερών και μεταβλητών, ο υπολογισμός μπορεί να γίνει ως εξής:

```
>> theta = 60*pi/180;
>> a = sin(theta);
>> b = cos(theta);
>> a^2 + b^2
ans =
        ...............
```

Τα σύμβολα **theta, a** και **b** αντιπροσωπεύουν σταθερές ή μεταβλητές ανάλογα με το άν επιτρέπεται να αλλάζουν στη συνέχεια ή όχι. Το σύμβολο **ans** είναι μεταβλητή και μπορεί επίσης να χρησιμοποιηθεί σε περαιτέρω υπολογισμούς όπως στο παράδειγμα:

```
>> 60*pi/180
ans =
        ..................
>> sin(ans)
ans =
        .....................
```

Μορφή (format)

Το MATLAB παρέχει τη δυνατότητα εμφάνισης των αριθμών με διαφορετικό πλήθος ψηφίων ανάλογα με την ακρίβεια που επιθυμούμε. Φυσικά, η εσωτερική αναπαράσταση των αριθμών είναι ανεξάρτητη από τη μορφή εμφάνισης. Η προεπιλογή (default) της μορφής στο MATLAB όσον αφορά τα σημαντικά δεκαδικά ψηφία δίνεται από την εντολή **'format'** ή **'format short'** που εμφανίζει μέχρι τέσσερα δεκαδικά ψηφία ενώ η προεπιλογή ως προς την από-σταση των γραμμών δίνεται με την εντολή `'format loose'`. Για μεγαλύτερη ακρίβεια μπορούμε να χρησιμοποιήσουμε τις εντολές `'format long'` και `'format long e'` ενώ για την απαλοιφή των κενών γραμμών, που συνιστάται, χρησιμοποιούμε την εντολή `'format compact'`.

Παραδείγματα:

>> format compact
>> pi
ans =
……………
>> format long
>> pi
ans =
……………………
>> format long e
>> pi
ans =
……………………………….
>> x = 2;
>> y = 3;
>> z = x^2 + y^2 + x*y + x + y
z =
……………………………
>>format short

όπου στο τελευταίο παράδειγμα βλέπουμε ότι το ελληνικό ερωτηματικό στο τέλος μιας γραμμής αποτρέπει την εμφάνιση του αποτελέσματος (οι τιμές των μεταβλητών `x` και `y`). Η τιμή της μεταβλητής `z` εμφανίζεται επειδή δεν υπάρχει το σύμβολο `';'` στο τέλος της γραμμής. Περισσότερες πληροφορίες για την μορφή μπορείτε να βρείτε πληκτρολογώντας `'help format'`.

ΣΕΙΡΕΣ - ΔΙΑΝΥΣΜΑΤΑ

Στο MATLAB η σειρά ορίζεται ως μία διατεταγμένη συλλογή αριθμών που περικλείεται από αγκύλες [...] με τα στοιχεία να διαχωρίζονται είτε από κενά είτε από κόμματα.

```
>> odd = [1 3 5 7 9 11 13 15 17 19]
odd =
   1    3    5    7    9    11    13    15    17
19
>> even = [2,4,6,8,10,12,14,16,18,20]
even =
   2    4    6    8   10    12    14    16    18
20
>> dekadikoi = [1.0 1.2 1.4 1.6 1.8 2.0]
dekadikoi =
   1.0000   1.2000   1.4000   1.6000   1.8000
2.0000
```

Τα στοιχεία της σειράς προσδιορίζονται με δείκτες θέσης (φυσικοί αριθμοί) αρχίζοντας από το 1:

```
>> odd(5)
ans =
      ...............
>> even(1)
ans =
      ...............
```

Το πλήθος των στοιχείων μιας σειράς υπολογίζεται από την συνάρτηση **length** του MATLAB:

```
>> length(even)
ans =
      ..................
```

Η εντολή **clear** μηδενίζει (σβήνει από τη μνήμη) τη σειρά:

```
>> clear even
>> even
??? Undefined function or variable 'even'
```

Ένας εναλλακτικός τρόπος εισαγωγής της παραπάνω σειράς, αν και πιο επίπονος, είναι ο εξής:

```
>> even(1) = 2
even =

...............
>> even(2) = 4
```

```
even =
```

....

```
>> even(3) = 6
even =
```

....

```
>> even(10) = 20
even =
```

....

Πράξεις με σειρές

Εστω οι σειρές $\boldsymbol{A} = [\boldsymbol{a}_1\ \boldsymbol{a}_2\ \ldots\ \boldsymbol{a}_n]$ και $\boldsymbol{B} = [\boldsymbol{b}_1\ \boldsymbol{b}_2\ \ldots\ \boldsymbol{b}_n]$. Η πρόσθεση και η αφαίρεση των δύο σειρών ορίζονται ως εξής:

$$\boldsymbol{A} + \boldsymbol{B} = [\boldsymbol{a}_1 + \boldsymbol{b}_1, \boldsymbol{a}_2 + \boldsymbol{b}_2, \ldots, \boldsymbol{a}_n + \boldsymbol{b}_n]$$

$$\boldsymbol{A} - \boldsymbol{B} = [\boldsymbol{a}_1 - \boldsymbol{b}_1, \boldsymbol{a}_2 - \boldsymbol{b}_2, \ldots, \boldsymbol{a}_n - \boldsymbol{b}_n]$$

Για παράδειγμα:

```
>> odd + even
ans =
```

....

```
>> even - odd
ans =
```

....

Στην περίπτωση που τα στοιχεία της σειράς βρίσκονται σε ίσες αποστάσεις, τότε δεν χρειάζεται η αναλυτική εισαγωγή της σειράς αλλά μόνο το πρώτο στοιχείο, το βήμα και το τελευταίο στοιχείο με διαχωριστικό σύμβολο το `':'`. Για παράδειγμα, οι παραπάνω σειρές odd και even μπορούν να ορισθούν και ως εξής:

```
>> odd = 1:2:19
odd =
```

……………………………………………………………………………………………………

…………

```
>> even = 2:2:20
even =
```

……………………………………………………………………………………………………

…………

Οταν το βήμα είναι 1 τότε μπορεί να παραληφθεί ενώ επιτρέπονται επίσης αρνητικά και κλασματικά βήματα:

```
>> natural = 1:10
natural =
```

………………………………………………………………………………………….

```
>> inv_odd = 19:-2:1
inv_odd =
```

………………………………………………………………………………………….

```
>>dekadika = 0:0.1:1
dekadika =
```

……………………………………………………………………………………………………

…………….

Ορισμός πράξης πολλαπλασιαμού σειρών:

$A\ .*\ B = [a_1b_1, a_2b_2, \ldots, a_nb_n]$

όπου το σύμβολο `'.*'` σημαίνει πολλαπλασιασμός στοιχείου προς στοιχείο. Για παράδειγμα:

```
>> odd.*even
ans =
```

……………………………………………………………………….

Ορισμός διαίρεσης (από αριστερά και από δεξιά) σειρών:

$A\ ./\ B = [a_1/b_1, a_2/b_2, \ldots, a_n/b_n]$

$A\ .\backslash\ B = [a_1\backslash b_1, a_2\backslash b_2, \ldots, a_n\backslash b_n] \Leftrightarrow B\ ./\ A$

Παραδείγματα:

```
>> odd./even
ans =
  Columns 1 through 7
```

……………………………………………………………………………………………………
………………………..

```
  Columns 8 through 10
```

………………………………………………………..

```
>> odd.\even
ans =
  Columns 1 through 7
```

……………………………………………………………………………………………………
………………………….

```
  Columns 8 through 10
```

…………………………………………………………..

Ορισμός ύψωσης σε δύναμη:

$A\ .\hat{}\ m = [a_1^m, a_2^m, \ldots, a_n^m]$

Παράδειγμα:

```
>> natural.^2
ans =
```

……………………………………………………………………………………

ΣΗΜΕΙΩΣΗ:

Πολλές από τις ενσωματωμένες συναρτήσεις του MATLAB μπορούν να εφαρμοσθούν σε σειρές αν απλώς, στη θέση του ορίσματος, χρησιμοποιηθεί το όνομα της σειράς:

```
>> angle = 0:10:90;
>> angle = pi*angle/180;
>> sin(angle)
ans =
   Columns 1 through 7
```

....
....

```
   Columns 8 through 10
```

....

Στο παραπάνω παράδειγμα, ο απλός πολλαπλασιασμός ή διαίρεση σειράς με αριθμό οδηγεί στην αντίστοιχη πράξη του αριθμού με κάθε στοιχείο της σειράς.

Διανύσματα γραμμής – στήλης

Οι σειρές της προηγούμενης ενότητας μπορούν να θεωρηθούν και ως διανύσματα γραμμής (οριζόντια) με στοιχεία τα αντίστοιχα στοιχεία της σειράς. Αν και η δήλωση διανυσμάτων γραμμής μπορεί να είναι η ίδια με τη δήλωση των σειρών, είναι καλό να περιλαμβάνουμε τα στοιχεία του διανύσματος μέσα σε αγκύλες **[ ]** όπως:

```
>> odd = [1:2:19]
odd =
```

....

```
>> even = [2,4,6,8,10,12,14,16,18,20]
even =
```

....

```
>> N = [1:5]
N =
```

....

Η δήλωση ενός διανύσματος στήλης είναι ίδια ως προς τη μορφή με αυτήν ενός διανύσματος γραμμής εκτός από το διαχωριστικό σύμβολο που τώρα είναι είτε το `';'` είτε η αλλαγή γραμμής:

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

```
A =
     1
     2
     3
     4
     5
>> B = [2
3
5
7
11]
B =
     2
     3
     5
     7
    11
```

Η μετατροπή ενός διανύσματος στήλης σε γραμμής και το αντίστροφο μπορεί να γίνει με την χρήση του αναστρόφου διανύσματος που συμβολίζεται με την απόστροφο:

```
>> At = A′
At =
     ................................................................................
>> Att = At′
Att =
```

Εσωτερικό γινόμενο διανυσμάτων

Εστω ένα διάνυσμα γραμμής $\boldsymbol{A} = [\boldsymbol{a}_1\ \boldsymbol{a}_2\ \ldots\ \boldsymbol{a}_n]$ και ένα διάνυσμα στήλης $\boldsymbol{B} = [\boldsymbol{b}_1;\ \boldsymbol{b}_2;\ \ldots;\ \boldsymbol{b}_n]$ με τον ίδιο αριθμό στοιχείων n. Το εσωτερικό γινόμενο $\boldsymbol{A} * \boldsymbol{B}$ των δύο διανυσμάτων είναι καθαρός αριθμός και δίνεται από την ακόλουθη εξίσωση:

$\boldsymbol{A} * \boldsymbol{B} = \boldsymbol{a}_1\boldsymbol{b}_1 + \boldsymbol{a}_2\boldsymbol{b}_2 + \ldots + \boldsymbol{a}_n\boldsymbol{b}_n$

Για τα διανύσματα της προηγούμενης ενότητας έχουμε (αφού αναστρέψουμε το διάνυσμα στήλης Α):

```
>> A′*B
ans =
       ..................
```

ενώ το άθροισμα των τετραγώνων των πρώτων 5 φυσικών αριθμών θα είναι:

```
>> N*N'
ans =
          ...............
```

ΠΙΝΑΚΕΣ

Οι πίνακες στο MATLAB περικλείονται σε αγκύλες [ ] και εισάγονται με απλό τρόπο. Με χρήση των διαχωριστικών ***κενό* ή *κόμμα*** για τα στοιχεία γραμμής και του **';'** για την αλλαγή γραμμής μπορούμε να ορίσουμε έναν πίνακα ως εξής:

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

Μπορούμε επίσης να γράψουμε κάθε γραμμή ξεχωριστά όπως:

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

Η πρόσβαση στα στοιχεία του πίνακα γίνεται με χρήση δύο δεικτών μέσα σε παρένθεση με τον πρώτο να προσδιορίζει τη γραμμή και τον δεύτερο τη στήλη:

```
>> A(1,3)
ans =
          ..................
>> A(3,2)
ans =
          ..................
```

Οι διαστάσεις ενός πίνακα δίνονται με τη συνάρτηση **size**:

```
>> size(A)
ans =
            ..........................
```

Δύο πίνακες A και B με τον ίδιο αριθμό γραμμών μπορούν να παρατεθούν ο ένας δίπλα στον άλλο και να δημιουργήσουν έναν νέο πίνακα με τον ίδιο αριθμό γραμμών και πλήθος στηλών όσο και το άθροισμά τους στους αρχικούς πίνακες. Η λειτουργία αυτή ονομάζεται παράθεση πινάκων και συμβολίζεται με [A B]:

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

```
>> size(C)
ans =
            .......................
```

Η δημιουργία ενός νέου πίνακα από δύο πίνακες A και B που έχουν ίδιο πλήθος στηλών είναι επίσης δυνατή με χρήση της λειτουργίας [A; B] όπως στο παράδειγμα:

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

C =
```

Για να εξαγάγουμε έναν υποπίνακα από την γραμμή x1 έως τη γραμμή x2 και από τη στήλη y1 έως τη στήλη y2 μέσα από κάποιον πίνακα A, χρησιμοποιούμε την μορφή **A(x1:x2; y1:y2)**. Για παράδειγμα, αν θέλουμε να εξαγάγουμε τον υποπίνακα που αποτελείται από τις δύο πρώτες γραμμές και στήλες του C θα έχουμε:

```
>> C(1:2,1:2)
ans =
```

Αν θέλουμε να εξαγάγουμε όλες τις γραμμές ή όλες τις στήλες, τότε δεν χρειάζεται να το δηλώσουμε αναλυτικά αλλά χρησιμοποιούμε μόνο το σύμβολο `':'`

```
>> C(:,1:2)
ans =
```

```
>> C(1,:)
ans =
```

....

```
>> C(:,1)
ans =
```

```
>> C(1:3,:)
ans =
```

Στοιχειώδεις πράξεις με πίνακες

Το άθροισμα δύο πινάκων **A** και **B** με τις ίδιες διαστάσεις mxn και με στοιχεία a_{ij} και b_{ij} αντίστοιχα, είναι ένας νέος πίνακας S με διαστάσεις *m*x*n* και στοιχεία s_{ij} που δίνονται από την εξίσωση $\boldsymbol{s}_{\mathbf{ij}} = \boldsymbol{a}_{\mathbf{ij}} + \boldsymbol{b}_{\mathbf{ij}}$. Αντίστοιχα, η διαφορά των πινάκων οδηγεί σε νέο πίνακα με στοιχεία $\boldsymbol{s}_{\mathbf{ij}} = \boldsymbol{a}_{\mathbf{ij}} - \boldsymbol{b}_{\mathbf{ij}}$ **.**

```
>> S1 = C(1:3,:)
S1 =
```

```
>> S2 = C(3:5,:)
S2 =
```

```
>> S = S1 + S2
S =
```

```
>> D = S1 - S2
D =
```

Για να πολλαπλασιάσουμε δύο πίνακες **A** και **B** πρέπει το πλήθος των στηλών του πρώτου να είναι ίδιο με το πλήθος των γραμμών του δεύτερου. Εστω, για παράδειγμα, ότι οι διαστάσεις των **A** και **B** είναι *m*x*p* και *p*x*n* αντίστοιχα. Τότε, οι διαστάσεις του νέου πίνακα **P** που αντιστοιχεί στο γινόμενο των δύο πινάκων θα είναι *m*x*n* και τα στοιχεία του θα δίνονται από την εξίσωση:

$$p_{ij} = \sum a_{ik} b_{kj}$$

Συνεχίζοντας το προηγούμενο παράδειγμα, το γινόμενο των **S1** και **S2** θα είναι:

```
>> P = S1 * S2
P =
```

Ένας τετραγωνικός πίνακας (όπως οι **S1** και **S2**) μπορεί να πολλαπλασιαστεί με τον εαυτό του:

```
>> S1 * S1
ans =
```

Ισοδύναμα, για τετραγωνικούς πίνακες, μπορούμε να χρησιμοποιήσουμε το σύμβολο ύψωσης σε δύναμη (**A**^2 = **A*A**, **A**^3 = **A*A*A**, κ.λπ.):

```
>> S1^2
ans =
```

Οπως και στην περίπτωση των μονοδιάστατων σειρών, έτσι και στην περίπτωση των δισδιάστατων σειρών (δηλαδή, των πινάκων), μία ενσωματωμένη συνάρτηση επιδρά σε κάθε στοιχείο του πίνακα ξεχωριστά:

```
>> angle = [0:10:20;30:10:50;60:10:80]
angle =
```

```
>> angle = pi*angle/180
angle =
```

```
>> sin(angle)
ans =
```

Επίσης, ένας σύντομος τρόπος ορισμού ενός πίνακα με όλο μηδενικά ή μονάδες είναι με χρήση των λειτουργιών **ones(m,n)** και **zeros(m,n)**:

```
>> ones(2,3)
ans =
```

```
>> zeros(2,2)
ans =
```

Τέλος, αν θέλουμε να δημιουργήσουμε έναν πίνακα από μονάδες που να έχει τις ίδιες διαστάσεις με κάποιον πίνακα A, τότε μπορούμε να χρησιμοποιήσουμε τις διαστάσεις που επιστρέφονται από την συνάρτηση **size**:

```
>> A = [1 2 3 4; 5 6 7 8; 9 10 11 12]
A =
```

```
>> [m, n] = size(A)
m =
     .........
n =
     .........
>> ones(m, n)
ans =
```

ή πιό απλά ακόμη:

```
>> ones(size(A))
ans =
```

Δημιουργία αρχείων M

Οι ακολουθίες εντολών του MATLAB μπορούν να γραφούν σε αρχεία των οποίων οι ονομασίες θα έχουν κατάληξη m, και θα ονομάζονται κατ' αναλογία **αρχεία-M**.
Η εκτέλεση του αρχείου-M μπορεί να γίνει είτε με τη χρήση του εικονιδίου "save and run" που βρίσκεται πάνω στο toolbar του editor του matlab, είτε πληκτρολογώντας το όνομα ενός τέτοιου αρχείου, χωρίς το m, προκαλούμε την εκτέλεση όλων των εντολών.
Για **παράδειγμα** ένα αρχείο-M δημιουργείται από το μενού **File → New → M file**
Πληκτρολογώντας τα παρακάτω θα δημιουργήσουμε ένα πρόγραμμα που θα κάνει την γραφική παράσταση μιας εξίσωσης πρώτου βαθμού:

```
%example1
%ΓΡΑΦΙΚΗ ΠΑΡΑΣΤΑΣΗ ΣΥΝΑΡΤΗΣΗΣ ΠΡΩΤΟΥ ΒΑΘΜΟΥ
%ΔΙΝΟΥΜΕ ΤΙΜΕΣ ΣΤΟΥΣ ΠΑΡΑΓΟΝΤΕΣ ΤΗΣ ΕΞΙΣΩΣΗΣ
a=1;
b=2;
%ΔΙΝΟΥΜΕ ΤΙΣ ΤΙΜΕΣ ΤΟΥ Χ
x=-10:1:10;
%ΓΡΑΦΟΥΜΕ ΤΗΝ ΠΡΩΤΟΒΑΘΜΙΑ ΕΞΙΣΩΣΗ
y=a*x+b;
%ΣΧΕΔΙΑΖΟΥΜΕ ΤΗ ΓΡΑΦΙΚΗ ΠΑΡΑΣΤΑΣΗ
grid;
plot(x,y);
```

Αποθηκεύεστε το αρχείο με την εντολή **File→ Save as** στον κατάλογο **work** με το όνομα **example1**.
Για να τρέξετε το αρχείο που μόλις δημιουργήσατε αρκεί να πληκτρολογήσετε
>> **example1**
Στο command window του Matlab.

Στο MATLAB μπορούμε να προγραμματίσουμε τις συναρτήσεις που εμείς θέλουμε βάζοντας σαν πρώτη λέξη του προγράμματος το function. Αυτά τα αρχεία ονομάζονται **αρχεία συναρτήσεων** και λαμβάνουν εξωτερικά ορίσματα τα οποία περιέχονται σε παρενθέσεις αμέσως μετά το όνομα της συνάρτησης.
Για **παράδειγμα** το προηγούμενο πρόγραμμα που έκανε τη γραφική παράσταση μιας εξίσωσης πρώτου βαθιού μπορεί να τροποποιηθεί έτσι ώστε να δίνει το αποτέλεσμα της εξίσωσης αν του δώσουμε τιμές για τα a, b και x

function y=firstorder(a,b,x)
%firstorder(A,B,X)
%ΣΥΝΑΡΤΗΣΗ ΠΡΩΤΟΥ ΒΑΘΜΟΥ Y=AX+B
%ΔΩΣΤΕ ΤΟ Α ΤΟ Β ΚΑΙ ΤΟ Χ ΚΑΙ ΘΑ ΠΑΡΕΤΕ ΤΟ ΑΠΟΤΕΛΕΣΜΑ
%ΓΡΑΦΟΥΜΕ ΤΗΝ ΠΡΩΤΟΒΑΘΜΙΑ ΕΞΙΣΩΣΗ
y=a*x+b;

Αποθηκεύστε το πρόγραμμα με το όνομα firstorder με την ίδια προηγούμενη
διαδικασία στον ίδιο κατάλογο.
Τώρα πληκτρολογώντας το όνομα του αρχείου μαζί με τα δεδομένα που θέλουμε να
του δώσουμε στο περιβάλλον του MATLAB θα πάρουμε:
>>firstorder(5,7,8)
Ans=

……………….

<u>ΜΙΓΑΔΙΚΟΙ ΑΡΙΘΜΟΙ</u>

Στα **παραδείγματα** που ακολουθούν βλέπουμε την ευκολία με την οποία χειρίζεται το MATLAB τους μιγαδικούς. Παρατηρήστε ότι εκτός από τις απλές πράξεις, χρησιμοποιούμε στις εκφράσεις και κάποιες συναρτήσεις όπως οι **`sqrt`** (τετραγωνική ρίζα) και **`sin`** (ημίτονο). Η συνάρτηση **`abs`** υπολογίζει το μέτρο ενός μιγαδικού αριθμού ή απλά την απόλυτη τιμή ενός πραγματικού. Η **`angle`** υπολογίζει το όρισμα (γωνία) του μιγαδικού σε ακτίνια, ενώ τέλος οι **`real`** και **`imag`** επιστρέφουν το πραγματικό και φανταστικό μέρος αντίστοιχα ενός μιγαδικού.

<u>Παραδείγματα:</u>

```
» z=1-2i % i is the imaginary unit
z =
   1.0000- 2.0000i
» w=1-2j % j is also the imaginary unit
w =
   1.0000- 2.0000i
» z1=3*(2-sqrt(-1)*3)
z1 =
   ...................................
» z2=sqrt(-2)
z2 =
          ..........................
» z3=6+sin(.5)*i
z3 =
      .............................
» z4=(z3+w)/z1
z4 =
        ................................
» i^2
ans =
      .............................
» magnitude_z4=abs(z4)
magnitude_z4 =
                ..........................
» angle_z4=angle(z4)
angle_z4 =
          ....................
» deg_z4=angle_z4*180/pi % convert in degrees
deg_z4 =
         ....................
```

```
» real_z4=real(z4)
real_z4 =
            .............................
» imag_z4=imag(z4)
imag_z4 =
     ………………
```

Γραφικές παράστασεις

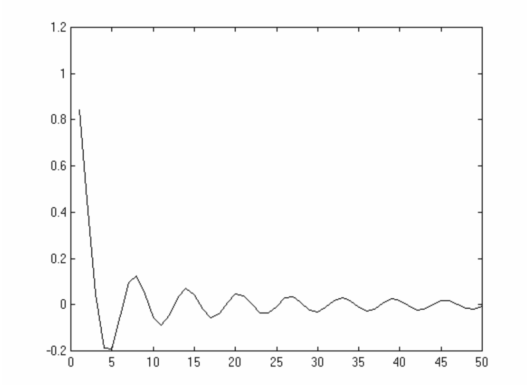
Εστω ότι θέλουμε να κάνουμε τη γραφική παράσταση της ημιτονοειδούς συνάρτησης στο διάστημα [0, 2π]. Η βασική συνάρτηση του MATLAB για δισδιάστατες απεικονίσεις είναι η **`plot`** (για λεπτομέρειες πληκτρολογήστε **`help plot`**). Αλλες χρήσιμες συναρτήσεις είναι η **`grid`** που σχεδιάζει τον κάνναβο και οι **`xlabel, ylabel`** για την εισαγωγή κειμένου στις γραφικές παραστάσεις.

```
>> x = 0: pi/90: 2*pi;
>> y = sin(x);
>> plot(x,y)
```

```
>> grid
>> xlabel('x, ακτίνια')
>> ylabel('sin(x)')
```

Στην πιο απλή περίπτωση η **plot(x)** κάνει γραφική παράσταση των στοιχείων του <u>διανύσματος</u> x ως προς τον αύξοντα αριθμό τους.

```
»n=1:50;
»an=sin(n)./n;
»plot(an)
```



Αν χρησιμοποιήσουμε την μορφή **plot(x,y)**, θα πάρουμε γραφική παράσταση των στοιχείων του y ως προς τα αντίστοιχα του x :

```
»x=linspace(-2*pi,2*pi,1000);
»y=sin(x);
»plot(x,y)
```

Υπάρχει η δυνατότητα να σχεδιαστούν μαζί πολλές γραφικές παραστάσεις με διαφορετικά χρώματα, αν ορίσουμε στην `plot` πολλά ζεύγη διανυσμάτων

```
»x=linspace(0,2,1000);
»y1=exp(x);
»y2=2.5.^x;
»y3=2.^x;
»plot(x,y1,x,y2,x,y3)
```

Στα προηγούμενα παραδείγματα, αφήσαμε το MATLAB να σχεδιάσει τις γραφικές παραστάσεις με προκαθορισμένα χρώματα και συγκεκριμένο τύπο γραμμής (συνεχής γραμμή). Υπάρχει η δυνατότητα να ορίσουμε τα δικά μας χρώματα, τα σύμβολα για τα σημεία και τον τύπο της γραμμής που τα ενώνει. Αυτό επιτυγχάνεται με την εισαγωγή ενός ακόμη ορίσματος στην **`plot,`** μετά από το κάθε ζεύγος διανυσμάτων που θα σχεδιαστεί. Το νέο όρισμα είναι μία σειρά από ειδικούς χαρακτήρες μέσα σε αποστρόφους. Τους διαθέσιμους ειδικούς χαρακτήρες τους βλέπουμε στον παρακάτω πίνακα . Μπορούμε να χρησιμοποιήσουμε έναν ή περισσότερους χαρακτήρες σε οποιαδήποτε σειρά.

Σύμβολο	Χρώμα	Σύμβολο	Τύπος σημείων	Σύμβολο	Τύπος γραμμής
b	μπλε	.	.	-	συνεχής
g	πράσινο	o	o	:	διάστικτη
r	κόκκινο	x	x	-.	συνδυασμός -.
c	κυανό	+	+	--	διακεκομμένη
m	πορφυρό	*	*		
y	κίτρινο	s	τετράγωνο		
k	μαύρο	d	ρόμβος		
w	άσπρο	<, >	τρίγωνο		
		p	πεντάγωνο		
		h	εξάγωνο		

Οι παρακάτω εντολές θα δημιουργήσουν δύο καμπύλες σε μία γραφική παράσταση. Η πρώτη θα έχει μπλε χρώμα, τα σημεία θα είναι πεντάγωνα και η γραμμή διάστικτη, ενώ η δεύτερη θα σχεδιαστεί με κόκκινο χρώμα, τα σημεία θα είναι κύκλοι και η γραμμή διακεκομμένη .

```
»x=linspace(-1,1,15);
»y1=asin(x);
»y2=acos(x);
»plot(x,y1,'b:p',x,y2,'r--o')
```

Θα αναφέρουμε τώρα εντολές οι οποίες ρυθμίζουν την εμφάνιση της γραφικής παράστασης.

Η εντολή **`grid on`** προσθέτει ένα πλέγμα γραμμών για καλύτερη ανάγνωση των συντεταγμένων, ενώ η **`grid off`** αφαιρεί το πλέγμα. Η **`grid`** απλά λειτουργεί σαν διακόπτης ανάμεσα στις δύο καταστάσεις .

```
»x=linspace(-3,3,150);
»y=exp(-x.^2);
»plot(x,y)
»grid on
```

Κανονικά, μία γραφική παράσταση πλαισιώνεται από ένα ορθογώνιο με τις υποδιαιρέσεις των αξόνων. Η εντολή **`box off`** εμφανίζει την γραφική παράσταση χωρίς το ορθογώνιο, ενώ η **`box on`**, το επαναφέρει. Η εντολή **`box`** λειτουργεί σαν διακόπτης .

```
»x=linspace(-10,10,150);
»y=1./(1+x.^2);
»plot(x,y)
»box off
```

Μπορούμε να προσθέσουμε ετικέτες (*labels*) στον οριζόντιο και κατακόρυφο άξονα μίας γραφικής παράστασης με τις εντολές **`xlabel`** και **ylabel**. Μπορούμε ακόμη να βάλουμε τίτλο σε όλη την γραφική παράσταση με την εντολή **title**

```
»x=-2*pi:.01:2*pi;
»y=atanh(sin(x))-sin(x);
»plot(x,y)
»title('PLot of a function')
»xlabel('Variable x')
»ylabel('Variable y')
```

Υπάρχει τρόπος να προσθέσουμε κείμενο μέσα στο γράφημα με την εντολή **text**. Η γενική μορφή είναι **text(x,y,'κείμενο')** όπου (x, y) είναι οι συντεταγμένες του σημείου όπου θα γραφεί το κείμενο. Με τις παρακάτω εντολές θα δημιουργήσουμε μία γραφική παράσταση με τις καμπύλες sin(x) και cos(x) και θα προσθέσουμε ένα κείμενο στη θέση (2.5,0.7).

```
»x=linspace(0,2*pi,150);
»y=sin(x);
»z=cos(x);
»plot(x,y,x,z,':')
»text(2.5,.7,'sin(x)')
```

Ένας πιο απλός τρόπος να εισάγουμε κείμενο σε γράφημα, είναι να χρησιμοποιήσουμε την εντολή **gtext('κείμενο')**. Αυτή μας οδηγεί στο παράθυρο του γραφήματος όπου εμφανίζεται ένας σταυρός ο οποίος μετακινείται με το mouse. Μόλις εντοπίσουμε το κατάλληλο σημείο, εισάγουμε το κείμενο με click.

```
»gtext('cos(x)')
```

Αντί να παρεμβάλουμε κείμενο μέσα στο γράφημα για να ξεχωρίζουμε τις γραμμές, μπορούμε να εισάγουμε μία συνολική επιγραφή με την εντολή **legend**. Αυτόματα εισάγεται ένα ορθογώνιο το οποίο περιέχει τις επιγραφές που θα ορίσουμε για κάθε καμπύλη του γραφήματος. Το ορθογώνιο αυτό μπορούμε να το μετακινήσουμε με το mouse (click and drag) σε όποιο σημείο του γραφήματος επιθυμούμε.

```
»legend('sin(x)','cos(x)')
```

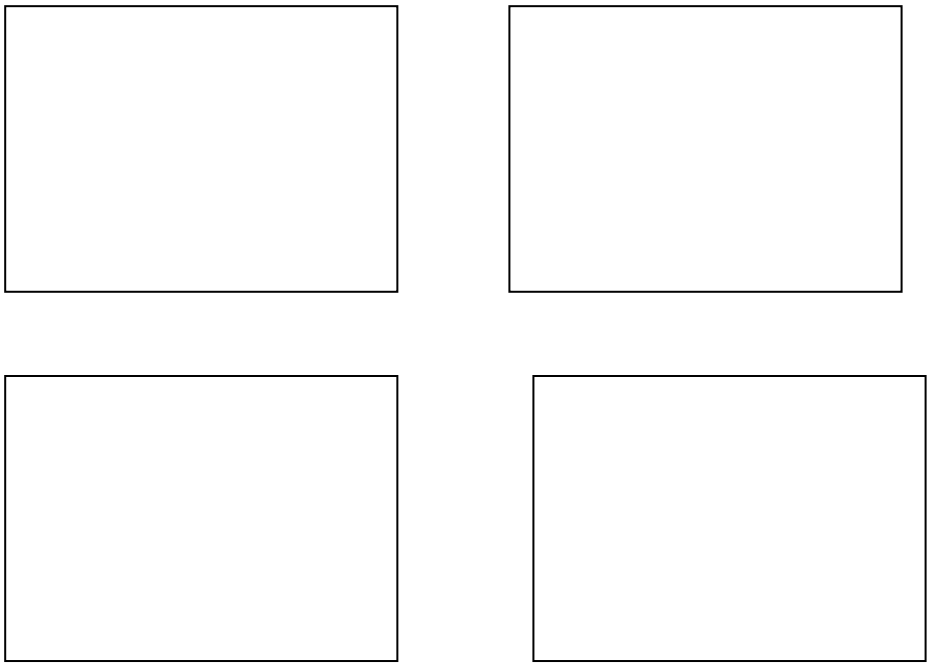
Η επιγραφή διαγράφεται με την εντολή **legend off**.

Ένα πολύ σημαντικό χαρακτηριστικό των γραφημάτων του MATLAB είναι η δυνατότητα μορφοποίησης των αξόνων. Αυτή επιτυγχάνεται με την εντολή **axis** η οποία έχει πολλές ειδικές περιπτώσεις (ζητήστε να εμφανιστούν με την εντολή `help axis`). Οι πιο απλές είναι οι **axis off** και **axis on** που αντίστοιχα εξαφανίζουν και εμφανίζουν τους άξονες.

Αν θέλουμε να έχουμε γραφήματα σε πολλά παράθυρα, μπορούμε να ανοίξουμε νέα παράθυρα γραφικών από το **File** του κυρίως menu επιλέγοντας **New Figure**. Ιδιαίτερα χρήσιμη είναι και η δυνατότητα να σχεδιάσουμε πολλά διαφορετικά γραφήματα με διαφορετικούς άξονες στο ίδιο παράθυρο. Η εντολή **subplot(m,n,p)** διαιρεί το ενεργό παράθυρο γραφικών σε $m \times n$ θέσεις ενώ ταυτόχρονα καθιστά ενεργή την p-θέση. Οι παρακάτω εντολές έχουν αποτέλεσμα τη δημιουργία των 4 γραφικών που βλέπουμε στο Σχήμα 16. Προσέξτε την χρήση της εντολής **axis([xmin,xmax,ymin,ymax])** για τον καθορισμό των ορίων των αξόνων. Η διαίρεση του παράθυρου αναιρείται με την εντολή **subplot(1,1,1)**.

```
>>x=linspace(-2*pi,2*pi,1000);
>>y1=sin(x);
>>y2=x;
>>z=y1+y2;
>>w=y1.*y2;
>>subplot(2,2,1)
>>plot(x,y1),axis([-2*pi,2*pi,-1,1]),title('sin(x)')
>>subplot(2,2,2)
>>plot(x,y2),axis([-2*pi,2*pi,-2*pi,2*pi]),title('x')
```

```
>>subplot(2,2,3)
>>plot(x,z),axis([-2*pi,2*pi, 2*pi,2*pi]),title ('sin(x)+x' )
>>subplot(2,2,4)
 >>plot(x,w),axis([-2*pi,2*pi,-2*pi,2*pi]),title('x*sin(x)')
```



<u>Συμβολικές εκφράσεις</u>

Το MATLAB διαθέτει μία συλλογή από συναρτήσεις κάτω από το όνομα ***Symbolic Math Toolbox*** που χρησιμεύουν σαν εργαλεία για την εκτέλεση συμβολικών πράξεων όπως η επίλυση αλγεβρικών και διαφορικών εξισώσεων, η παραγώγιση και η ολοκλήρωση συναρτήσεων, η εύρεση οριζουσών και χαρακτηριστικών ριζών κ.α..

Σε κάθε συμβολική έκφραση είναι σημαντικό να δηλώνουμε αρχικά τις μεταβλητές που παίρνουν μέρος στις εκφράσεις σαν

συμβολικές μεταβλητές (*symbolic variables*). Αυτό γίνεται με τις συναρτήσεις **sym** (για μία μόνο μεταβλητή) και **syms** (για πολλές μαζί μεταβλητές). Για παράδειγμα, με τις εντολές που ακολουθούν, δηλώνουμε μία μεταβλητή x σαν συμβολική και στη συνέχεια την χρησιμοποιούμε για να υπολογίσουμε την παράγωγο της συνάρτησης cosx.

```
»x=sym('x')
x =
        .........
»diff(cos(x))
ans =
          ..................
```

Με τις παρακάτω εντολές, δηλώνουμε 4 συμβολικές μεταβλητές μαζί, κατασκευάζουμε ένα συμβολικό πίνακα με αυτές και κατόπιν υπολογίζουμε την ορίζουσά του.

```
»syms('a','b','c','d')
»M=[a,b;c,d]
M =


»det(M)
  ans =
          ...........................
```

(α) Αριθμητές και παρονομαστές ρητών παραστάσεων

Όταν έχουμε μια σύνθετη έκφραση με κλάσματα που θέλουμε να την μετατρέψουμε σε ρητή παράσταση, χρησιμοποιούμε την εντολή **numden**. Αυτή μας επιστρέφει ξεχωριστά τον αριθμητή και τον παρονομαστή της ρητής παράστασης. Για παράδειγμα, αν θέλουμε να μετατρέψουμε την παράσταση

$$\frac{1}{x+1}-\frac{2}{x-2}+\frac{2}{x+5}$$

σε ρητή, δίνουμε τις εντολές:

```
»syms x
»g=1/(x+1)-2/(x-2)+2/(x+5)
g =

    ...............................................
»[n d]=numden(g)
n =

    ................................
d =


     ................................
```

Προκύπτει λοιπόν ότι η ζητούμενη παράσταση είναι

$$\frac{x^2-11x-24}{(x+1)(x-2)(x+5)}.$$

(β) Συμβολικά αθροίσματα – σειρές

Η εντολή **symsum** χρησιμοποιείται για να υπολογίζει συμβολικά αθροίσματα εκφράσεων. Η πιο βολική μορφή της είναι η

symsum(*έκφραση, μεταβλητή-δείκτης, κάτω όριο, πάνω όριο*).

Για παράδειγμα, θα υπολογίσουμε τα παρακάτω αθροίσματα:

$$\sum_{k=0}^{n} w^k = \frac{1-w^{n+1}}{1-w}, \qquad \sum_{k=1}^{n} k = \frac{n(n+1)}{2},$$

$$\sum_{k=0}^{\infty} w^k = \frac{1}{1-w}, \qquad \sum_{k=0}^{\infty} \frac{x^k}{k!} = e^x$$

```
»syms x w k n
»f1=w^k;f2=k;f3=x^k/sym('k!');
```

```
»symsum(f1,k,0,n)
ans =

     ......................................................
»symsum(f2,k,1,n)
ans =

     ..................................................... .
»symsum(f1,k,0,inf)
ans =

     ..............................
»symsum(f3,k,0,inf)
ans =

      ...................
```

Προσέξτε ότι τα αποτελέσματα είναι ισοδύναμα με αυτά που περιμένουμε αλλά έχουν άλλη μορφή.(Π.χ. $\frac{1}{2}(n+1)^2 - \frac{1}{2}n - \frac{1}{2} = \frac{n(n+1)}{2}$). Προσέξτε επίσης τον τρόπο με τον οποίο δηλώνουμε το παραγοντικό (!) αφού το MATLAB δεν μπορεί να το αναγνωρίσει διαφορετικά. Για να δηλώσουμε το άπειρο, χρησιμοποιούμε τη μεταβλητή του MATLAB **`inf`**.

(γ) Παραγώγιση

Η παραγώγιση συναρτήσεων γίνεται με την εντολή **`diff`**. Οι διαφορετικές της μορφές είναι:

- **`diff`**(*έκφραση*): Παραγωγίζει την έκφραση ως προς μία από τις συμβολικές μεταβλητές της. Είναι προτιμότερο να χρησιμοποιείται όταν η έκφραση περιλαμβάνει μία μόνο μεταβλητή.
- **`diff`**(*έκφραση, n*): Παραγωγίζει την έκφραση n φορές.

- **`diff`**(*έκφραση, μεταβλητή, n*): Παραγωγίζει την έκφραση n φορές ως προς την μεταβλητή που θα καθορίσουμε.

Για παράδειγμα, θα υπολογίσουμε τις μερικές παραγώγους πρώτης και δεύτερης τάξης της συνάρτησης 2 μεταβλητών

$$f(x,y) = e^{-2x}\sin y\,.$$

```
»syms x y
»f=exp(-2*x)*sin(y)
f =

    ........................................
»df_dx=diff(f,x)
df_dx =

         ..............................................
»df_dy=diff(f,y)
df_dy =

         ..............................................
```

Σημειώστε ότι η συνάρτηση `diff` χρησιμοποιείται στο βασικό μέρος του MATLAB για να υπολογίζει τις διαφορές των διαδοχικών στοιχείων ενός διανύσματος ή των στηλών ενός πίνακα. Το MATLAB αποφασίζει ποια εκδοχή της εντολής θα χρησιμοποιήσει αφού είναι σε θέση να ξεχωρίζει τις συμβολικές από τις αριθμητικές εκφράσεις.

<u>(δ) Ολοκλήρωση</u>

Η εντολή με την οποία μπορούμε να υπολογίζουμε ολοκληρώματα, είναι η **`int`**. Έχει και αυτή διάφορες μορφές αλλά οι πιο πλήρεις είναι οι εξής:

`int`(*έκφραση, μεταβλητή*): Υπολογίζει το αόριστο ολοκλήρωμα ως προς τη συμβολική μεταβλητή που θα ορίσουμε. Υπάρχει περίπτωση είτε το ολοκλήρωμα να είναι αδύνατο να δοθεί σε κλειστή μορφή, είτε η συνάρτηση να είναι τόσο πολύπλοκη που το MATLAB να μη μπορεί να υπολογίσει την αντιπαράγωγο.

`int`(*έκφραση, μεταβλητή, κάτω όριο, πάνω όριο*): Υπολογίζει το ορισμένο ολοκλήρωμα σε κάποιο διάστημα ολοκλήρωσης.

Για **παράδειγμα**, θα υπολογίσουμε τα ολοκληρώματα:

$$\int \frac{3x+7}{(x^2+2x+5)^2}dx = \frac{x-2}{2(x^2+2x+5)} + \frac{1}{4}\tan^{-1}\left(\frac{x+1}{2}\right)$$

```
»syms x
»g1=(3*x+7)/(x^2+2*x+5)^2
g1 =
```

....

```
»int(g1,x)
ans =
```

....

(ε) Επίλυση εξισώσεων

Η συνάρτηση **`solve`** χρησιμοποιείται για την εύρεση των ριζών μιας συμβολικής έκφρασης. Μπορούμε να καθορίσουμε μεταβλητή ως προς την οποία θα γίνει η επίλυση (συνιστάται).

```
»syms a b c x
»solve(a*x^2+b*x+c,x)
ans =
[ 1/2/a*(-b+(b^2-4*a*c)^(1/2))]
[ 1/2/a*(-b-(b^2-4*a*c)^(1/2))]
»pretty(ans)
```