



## ΕΙΣΑΓΩΓΗ ΣΤΗ ΜΑΤLAB

---

**Γεώργιος Θ. Μπαλάσης**

Διευθυντής Ερευνών

*Ινστιτούτο Αστρονομίας, Αστροφυσικής, Διαστημικών Εφαρμογών και  
Τηλεπισκόπησης*

*Εθνικό Αστεροσκοπείο Αθηνών*

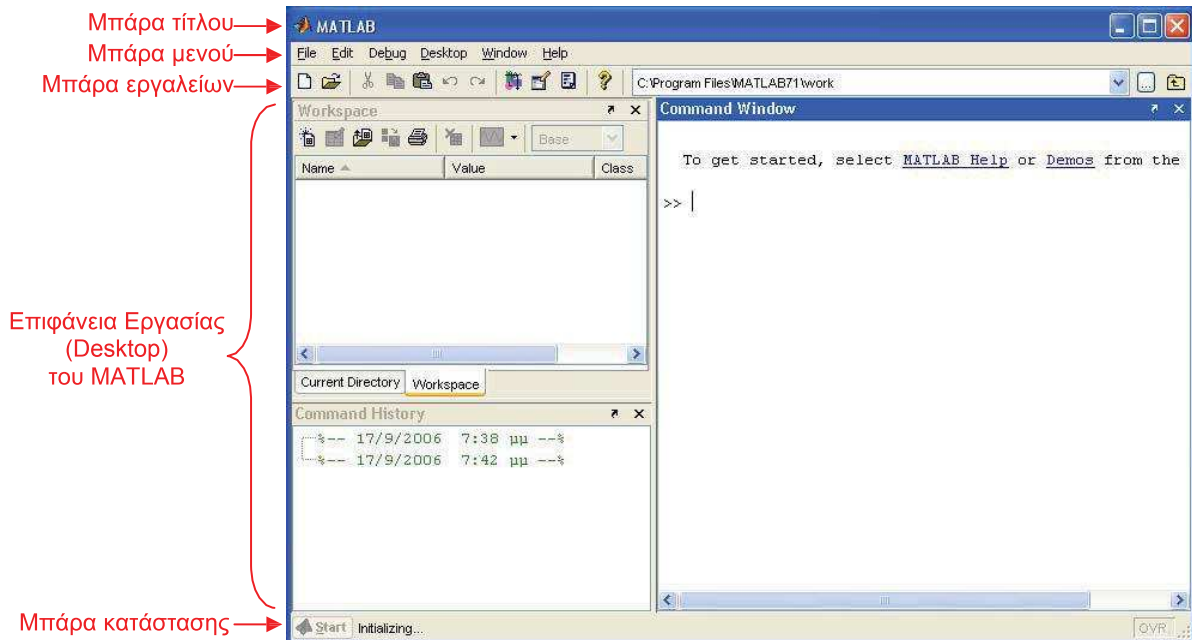
Αθήνα, Μάιος 2020

# 1 Το περιβάλλον εργασίας του MATLAB

Σε αυτήν την ενότητα θα εξοικειωθείτε με το περιβάλλον αλληλεπίδρασης του MATLAB με το χρήστη.

## 1.1 Μια πρώτη ματιά

Μετά την πρώτη εγκατάσταση και εκτέλεση του MATLAB (version 7) στα Windows, εμφανίζεται η οθόνη της παρακάτω εικόνας, που δείχνει το παράθυρο που απεικονίζει το περιβάλλον εργασίας του MATLAB.



Συνοπτικά, στο παράθυρο του MATLAB εμφανίζονται, από πάνω προς τα κάτω:

- f* η μπάρα τίτλου,
- f* η μπάρα του μενού επιλογών,
- f* η γραμμή εργαλείων με τα διάφορα εικονίδια,
- f* η Επιφάνεια Εργασίας (Desktop) του MATLAB, η οποία αποτελείται από διάφορα υπο-παράθυρα και καταλαμβάνει το μεγαλύτερο μέρος του παραθύρου του MATLAB και, τέλος,
- f* η μπάρα κατάστασης.

### Εργασία 1.1α. Εκκίνηση του MATLAB και εντοπισμός των στοιχείων του παραθύρου του.

- Ανοίξτε το MATLAB με διπλό κλικ του ποντικιού στο αντίστοιχο εικονίδιο που βρίσκεται στην επιφάνεια εργασίας των Windows.
- Παρατηρήστε το παράθυρο του MATLAB και εντοπίστε τα στοιχεία του (τη μπάρα τίτλου, το μενού, τη γραμμή εργαλείων και το Desktop του MATLAB)


## 1.2 Περιγραφή των τμημάτων του περιβάλλοντος εργασίας του MATLAB

Ακολουθεί περιγραφή των τμημάτων του παραθύρου του MATLAB:

- f Η μπάρα τίτλου (title bar) του MATLAB: Στο δεξί τμήμα είναι τα εικονίδια για ελαχιστοποίηση, μεγιστοποίηση και κλείσιμο όπως σε όλα τα προγράμματα των Windows:



### Εργασία 1.2α. Χειρισμός του παραθύρου του MATLAB.

- Κλείστε το MATLAB από το κουμπί  πάνω δεξιά . Στη συνέχεια ξανανοίξτε το.
- Ελαχιστοποιήστε το παράθυρο του MATLAB στη γραμμή εργασιών (taskbar) των Windows.
- Μεγιστοποιήστε το παράθυρο του MATLAB.
- Αλλάξτε μέγεθος και μετακινήστε το παράθυρο του MATLAB σε διάφορες θέσεις της οθόνης των Windows.

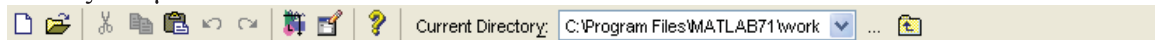
- f Η μπάρα του μενού επιλογών (menu bar): Στα αναπτυσσόμενα προς τα κάτω τμήματα του μενού επιλογών περιέχονται όλες οι εντολές χειρισμού του MATLAB:



### Εργασία 1.2β. Εξοικείωση με το μενού του MATLAB.

- Ανοίξτε με το ποντίκι τις επιλογές του μενού και περιηγηθείτε σ' αυτές ώστε να εξοικειωθείτε, χωρίς να ενεργοποιήσετε καμία εντολή. Παρατηρείστε τις συντομεύσεις πληκτρολογίου που ενεργοποιούν άμεσα τις εντολές.

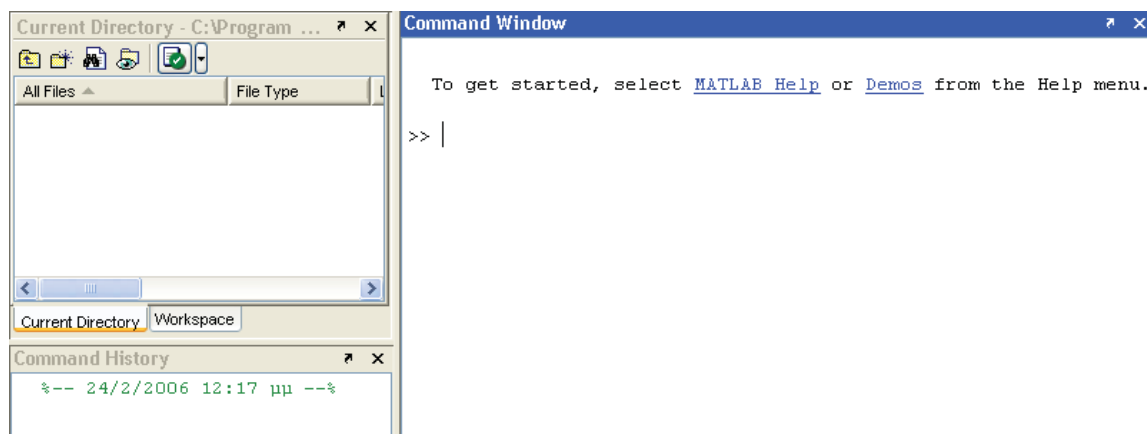
- f Η μπάρα εργαλείων (toolbar): Εδώ υπάρχουν εικονίδια που αντιστοιχούν στις πιο βασικές εντολές του μενού:



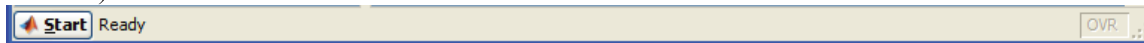
### Εργασία 1.2γ. Εξοικείωση με τη γραμμή εργαλείων του MATLAB.

- Αφήνοντας λίγες στιγμές το δείκτη του ποντικιού πάνω από ένα εικονίδιο, παρατηρείστε το ταμπελάκι βοήθειας που εμφανίζεται που μας πληροφορεί για την εντολή που εκτελείται όταν πιεστεί αυτό το εικονίδιο.
- Για τα επτά (7) πρώτα εικονίδια από αριστερά, εντοπίστε στο μενού τις αντίστοιχες εντολές.

- f Η Επιφάνεια Εργασίας (Desktop) του MATLAB: Το Desktop καταλαμβάνει το μεγαλύτερο μέρος του παραθύρου του MATLAB και αποτελείται από διάφορα παράθυρα εργασίας (πρόκειται ουσιαστικά για υπο-παράθυρα, μέσα στο παράθυρο του MATLAB):



f Η μπάρα κατάστασης (status bar): Εδώ, ανάμεσα σε άλλες πληροφορίες, εμφανίζεται η κατάσταση του MATLAB σε κάθε στιγμή (βλ. και σελίδα **Error! Bookmark not defined.**).

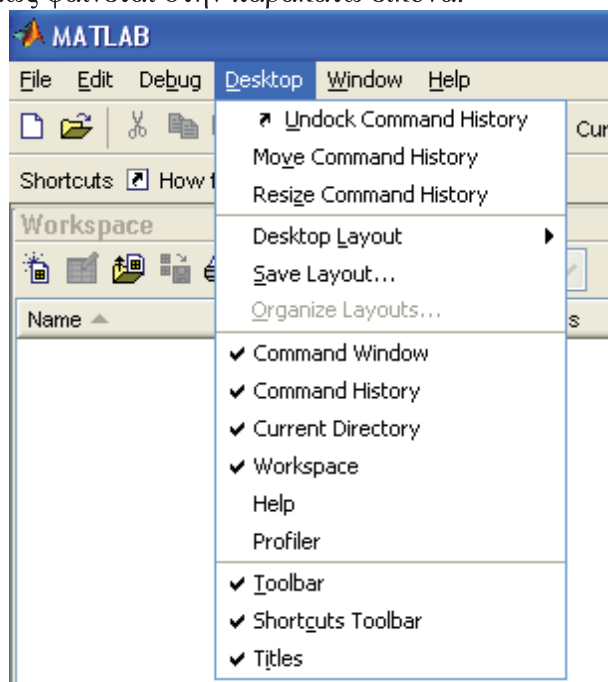


### 1.3 Τα παράθυρα της Επιφάνειας Εργασίας (Desktop) του MATLAB

Στο Desktop του MATLAB μπορούν να βρίσκονται τα εξής παράθυρα:

- Command Window: Παράθυρο όπου μπορούμε να δώσουμε εντολές και να εκτελεστούν άμεσα.
- Command History: Ιστορικό εντολών που δόθηκαν στο Command Window.
- Current Directory: Πρόχειρος φάκελος εργασίας για γρήγορη εκτέλεση και αποθήκευση προγραμμάτων και δεδομένων.
- Workspace: Παράθυρο για απεικόνιση των μεταβλητών που υπάρχουν στη μνήμη.
- Help: Παράθυρο βοήθειας κατά τα πρότυπα των Windows.
- Profiler: Παράθυρο επισκόπησης του χρόνου εκτέλεσης των διαφόρων συναρτήσεων ώστε να προχωρήσουμε σε βελτιστοποίηση του προγράμματός μας.

Μπορούμε να επιλέξουμε την εμφάνιση ή όχι των παραθύρων αυτών από την επιλογή Desktop του μενού, όπως φαίνεται στην παρακάτω εικόνα.



Επίσης, αν και δεν φαίνονται στην παραπάνω εικόνα, μπορεί να είναι ανοιχτά τα εξής παράθυρα:

- Editor: Παράθυρο για τη συγγραφή κώδικα προγραμμάτων του MATLAB.
- Figure: Παράθυρο γραφικών, όπου μπορούν να απεικονιστούν εικόνες και γραφικές παραστάσεις.

#### ***Εργασία 1.3α. Εξοικείωση με τα παράθυρα του Desktop του MATLAB.***

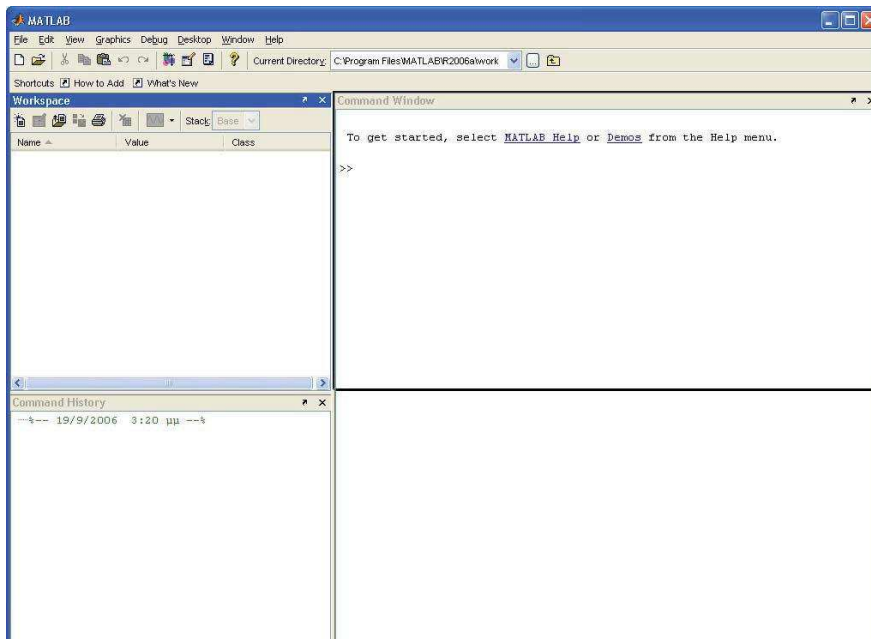
- Παρατηρείστε και εντοπίστε τα παράθυρα του MATLAB που βρίσκονται ανοικτά αυτή τη στιγμή.

- Ανοίξτε και κλείστε ένα-ένα από τα παράθυρα του MATLAB που υπάρχουν στην επιλογή Desktop του μενού, ώστε να εξοικειωθείτε με τη διαδικασία αυτή.

## 1.4 Χειρισμός των παραθύρων του Desktop

Τα παράθυρα του Desktop του MATLAB μπορεί να παρατίθενται οριζόντια, κατακόρυφα ή το ένα πίσω από το άλλο (tabbed windows) Στην τελευταία περίπτωση μπορούμε να επιλέξουμε αυτό που θέλουμε με πίεση του δείκτη του ποντικιού σε ταμπελάκι που βρίσκεται στο κάτω μέρος τους.

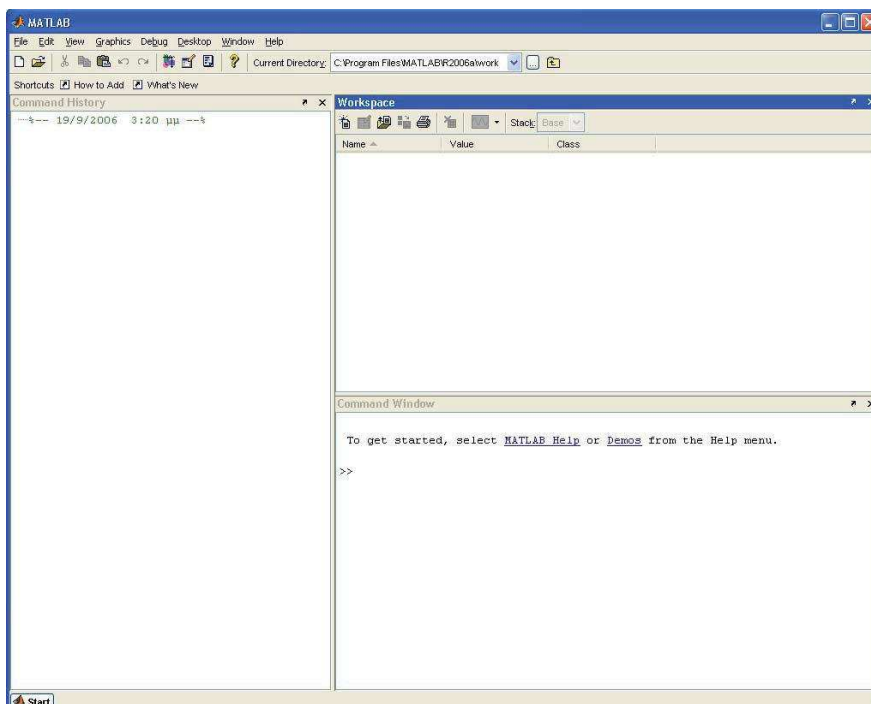
Αλλαγή θέσης και τρόπου παράθεσης παραθύρων: Η μετακίνηση ενός παραθύρου μπορεί να γίνει με το ποντίκι από τη μπάρα τίτλου του (τμήμα στο άνω μέρος του παραθύρου με τον τίτλο του, π.χ. **Command Window**), με πίεση και μετακίνηση με το αριστερό πλήκτρο του ποντικιού σε οποιοδήποτε σημείο (εκτός από το δεξί τμήμα) της μπάρας τίτλου. Κατά τη μετακίνηση ενός παραθύρου εμφανίζεται το περίγραμμά του για έλεγχο της νέας θέσης.



**Βήμα 1:**

Μετακίνηση με πατημένο το αριστερό πλήκτρο του ποντικιού.

Καθώς μετακινούμε το παράθυρο, εμφανίζεται το περίγραμμά του ώστε να δούμε τη νέα του θέση.



**Βήμα 2:**

Απελευθέρωση του ποντικιού.

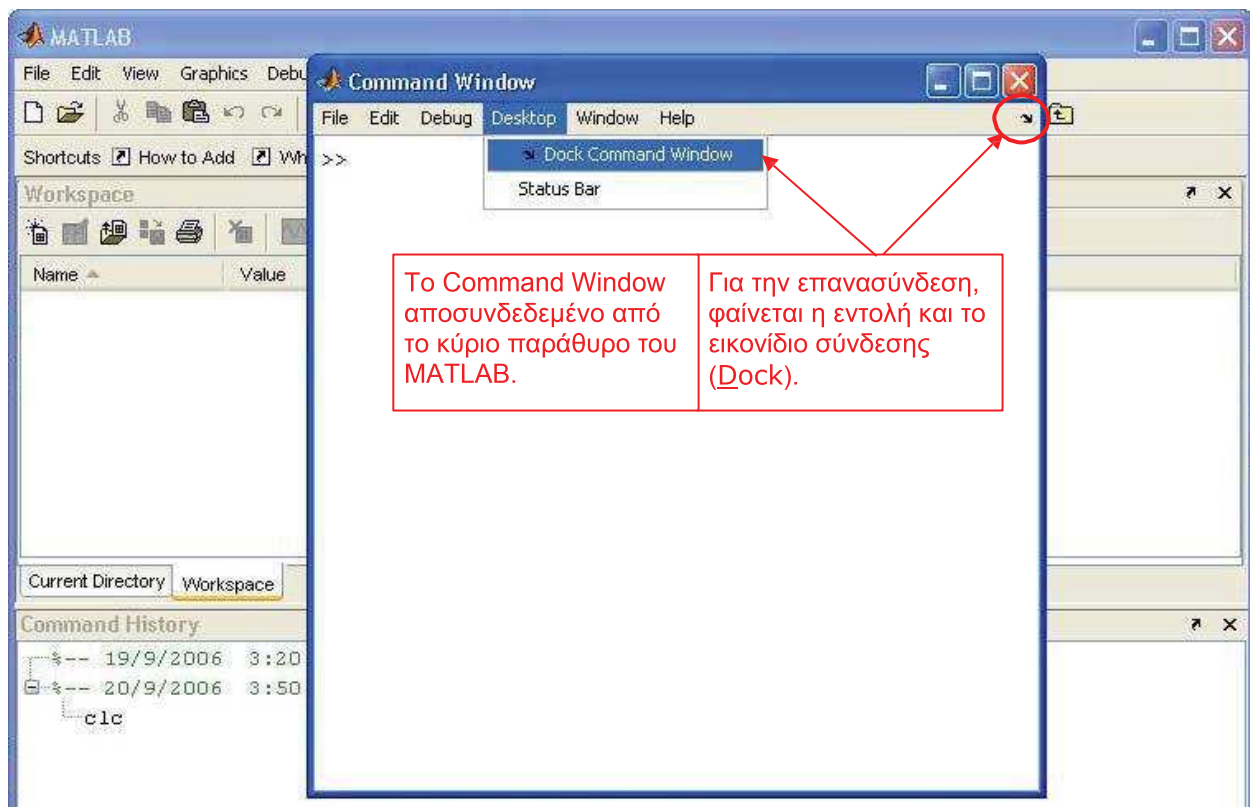
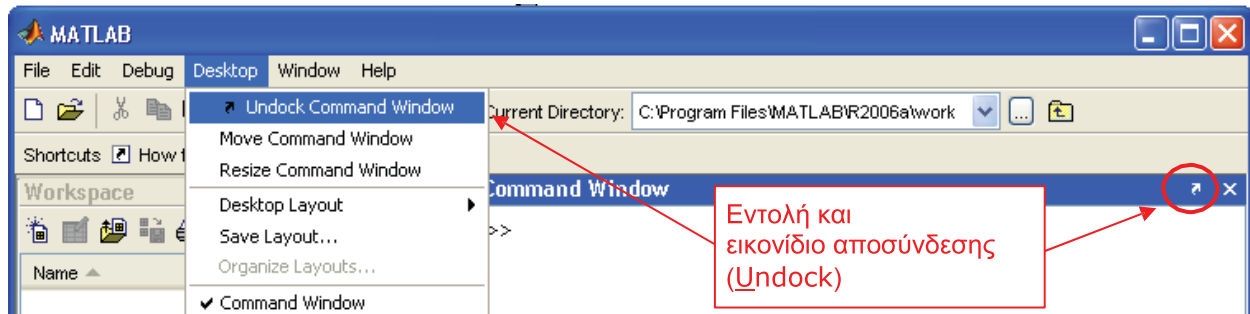
Το παράθυρο τοποθετείται στη νέα θέση.

Τα υπόλοιπα παράθυρα διευθετούνται κατάλληλα αυτομάτως.

### Εργασία 1.4α. Διευθέτηση των παραθύρων του Desktop MATLAB.

- Κλείστε όλα τα παράθρα του Desktop του MATLAB.
- Ανοίξτε μόνο τα παράθρα που φαίνονται στην προηγούμενη εικόνα και διευθετήστε τα όπως ακριβώς φαίνονται εκεί.

Αποσύνδεση (undock) και σύνδεση (dock) στο Desktop: Τα παράθρα εργασίας του MATLAB μπορούν να «αποσυνδεθούν» από το Desktop, ώστε να γίνουν ελεύθερα παράθρα των Windows. Η αποσύνδεση και η επανασύνδεση ενός παραθύρου του MATLAB γίνεται με τις επιλογές Undock ή Dock του μενού Desktop, ή με τα εικονίδια «↗» ή «↖» που βρίσκονται στο δεξί τμήμα της μπάρας τίτλου.



### Εργασία 1.4β. Αποσύνδεση και σύνδεση παραθύρων του MATLAB.

- Αποσυνδέστε και επανασυνδέστε διάφορα παράθρα ώστε να εξοικειωθείτε.
- Παρατηρήστε ότι τα αποσυνδεδεμένα παράθρα συμπεριφέρονται ως διαφορετικά παράθρα των Windows (π.χ. εμφανίζονται στην γραμμή εργαλείων των Windows).

## 1.5 Εργαστηριακές ασκήσεις

Οι παρακάτω εργασίες να εκτελεστούν σε Η/Υ με Windows και MATLAB 7:

### Άσκηση 1.5α

- i) Ανοίξτε το MATLAB.
- ii) Κλείστε όλα τα παράθυρα του Desktop του MATLAB.
- iii) Ανοίξτε τα παράθυρα *Command Window* και *Workspace*.
- iv) Κάντε ενεργό το παράθυρο του *Command Window* και στη συνέχεια του *Workspace*.
- v) Κλείστε το MATLAB.

### Άσκηση 1.5β

- i) Ανοίξτε το MATLAB.
- ii) Ανοίξτε τον *Editor* και ένα νέο παράθυρο για γράψιμο προγράμματος.
- iii) Διευθετήστε με τέτοιο τρόπο το Desktop του MATLAB, έτσι ώστε να είναι ανοιχτά μέσα σ' αυτό μόνο τα παράθυρα του *Editor* και του *Command Window*, το ένα δίπλα στο άλλο.
- iv) Γράψτε μια στοιχειώδη εντολή (π.χ.  $x=5$ ) στο παράθυρο του προγράμματος στον *Editor*, και αποθηκεύστε το πρόγραμμα με όνομα *test.m*. Εκτελέστε το πρόγραμμα. Εντοπίστε το αποτέλεσμα (πού εμφανίστηκε;)
- v) Κλείστε το MATLAB.

### Άσκηση 1.5γ

- i) Ανοίξτε το MATLAB.
- ii) Ανοίξτε στα Windows το φάκελο του *Current Directory*. Δημιουργήστε ένα νέο φάκελο εκεί, με όνομα *MyWork*. Αλλάξτε το *Current Directory* του MATLAB ώστε να είναι ο νέος φάκελος *MyWork*.
- iii) Κλείστε το MATLAB.

### Άσκηση 1.5δ

- i) Ανοίξτε το MATLAB.
- ii) Γράψτε μερικές στοιχειώδεις εντολές σε ένα παράθυρο του *Editor* και αποθηκεύστε το πρόγραμμα με όνομα *test1.m*. Ανοίξτε ένα νέο παράθυρο για συγγραφή προγράμματος. Αντιγράψτε και επικολλήστε (*copy-paste*) το πρόγραμμα *test.m* στο νέο παράθυρο και αποθηκεύστε το με όνομα *test2.m*.
- iii) Κλείστε το MATLAB.
- iv) Ανοίξτε το φάκελο του *Current Directory* του MATLAB από τα Windows και σβήστε τα δύο αρχεία *test1.m* και *test2.m*.

## 2 Μεταβλητές και εντολή ανάθεσης. Αριθμητικές πράξεις και εντολές στο Command Window.

Σε αυτήν την ενότητα θα εξοικειωθείτε με τη λειτουργία του Command Window για εκτέλεση απλών πράξεων και εντολών, και συγκεκριμένα με:

- Τον ορισμό μεταβλητών και την εντολή ανάθεσης.
- Τις αριθμητικές πράξεις στο Command Window και την προτεραιότητά τους.
- Τις ειδικές μεταβλητές `ans` και `pi`.
- Τις εντολές `format`, `clc`.
- Τις εντολές `clear`, `whos`.
- Τις συναρτήσεις `sin/cos/tan`, `exp/log`, `round`, `fix/rem`, `rand`.
- Τη χρήση των βελών του πληκτρολογίου (↑↓) για την εύρεση των προηγούμενων εντολών.

### 2.1 Συνοπτική θεωρία

Command Window (Παράθυρο Εντολών): Το Command Window είναι ένα παράθυρο του Desktop του του MATLAB, στο οποίο μπορούν:

- να οριστούν μεταβλητές
- να γίνουν πράξεις
- να δοθούν εντολές και
- να παρουσιαστούν αποτελέσματα.

Μεταβλητές: Είναι θέσεις στη μνήμη του υπολογιστή, που αντιπροσωπεύονται από ένα όνομα.

- Όνομα μεταβλητής: Αρχίζει από λατινικό γράμμα και περιέχει μόνο λατινικά γράμματα, αριθμούς ή κάτω παύλες (`_`, underscore), π.χ. `x`, `Y3`, `a_b`.
- Τύποι μεταβλητών: Αριθμητικές μεταβλητές (`-4`, `47.03`,...), μεταβλητές χαρακτήρα (2 Bytes, `'a'`, `'!'`, `'7'`,...) και λογικές μεταβλητές (1 Byte, `true`, `false`). Αν δεν οριστεί διαφορετικά, οι αριθμητικές μεταβλητές είναι `double` (κινητής υποδιαστολής 8 Bytes).
- Ειδικές μεταβλητές:
  - `ans`: Μεταβλητή με την τιμή της πιο πρόσφατης πράξης (αν δεν έχει ανατεθεί αλλού)
  - `pi = 3.141592653589793`: Προσέγγιση του  $\pi$  με 15 δεκαδικά ψηφία.

Τελεστές: Είναι σύμβολα που αντιστοιχούν σε πράξεις ή εντολές (π.χ. `=`, `+`, `*`, `>`, ...).

Πράξεις: Η σειρά προτεραιότητας των αριθμητικών πράξεων, μαζί με τους αντίστοιχους αριθμητικούς τελεστές, είναι:

- ^ ύψωση σε δύναμη
- \* / πολλαπλασιασμός και διαίρεση
- + - πρόσθεση και αφαίρεση
- αλλαγή προσήμου

Σε ίδια προτεραιότητα, οι πράξεις γίνονται από αριστερά προς τα δεξιά.

Για αλλαγή προτεραιότητας χρησιμοποιούνται παρενθέσεις.

Εντολές: Εντολές σχετικές με τον τρόπο εμφάνισης στο Command Window είναι:



- **Εντολή ανάθεσης (τελεστής «=»):** Υπολογίζεται η τιμή που βρίσκεται στη δεξιά πλευρά του τελεστή ανάθεσης (=), και ανατίθεται στη μεταβλητή της αριστερής πλευράς.
- **clc:** Καθαρισμός του Command Window.
- **clear:** Καθαρισμός των μεταβλητών από τη μνήμη.
- **whos:** Εμφάνιση των μεταβλητών που υπάρχουν στη μνήμη.
- **format short** ή **long:** Εμφάνιση έως 4 ή 15 δεκαδικών ψηφίων.  
Η εντολή `format short / long` έχει σχέση μόνο με το πλήθος των δεκαδικών ψηφίων που εμφανίζονται στο Command Window. Εσωτερικά οι πράξεις γίνονται πάντα με τη μέγιστη ακρίβεια.

Πολλές εντολές στην ίδια γραμμή: Σε μια γραμμή μπορούν να δοθούν πάνω από μία εντολές, αρκεί να χωρίζονται με ελλ. ερωτηματικό (;) ή κόμμα (,).

Εμφάνιση τιμής μεταβλητής: Γράφουμε το όνομα της μεταβλητής και πατάμε Enter (.-).

Απόκρυψη εμφάνισης μεταβλητής μετά από ανάθεση: Αν στο τέλος μιας εντολής ανάθεσης υπάρχει ελληνικό ερωτηματικό (;), τότε η τιμή της μεταβλητής δεν εμφανίζεται.

Χρήσιμες μαθηματικές συναρτήσεις:

- **sin, cos, tan:** Τριγωνομετρικοί αριθμοί (η γωνία σε ακτίνια).  
Π.χ. το ημίτονο των  $30^\circ$  είναι:  $\sin(30 \cdot \pi / 180)$ ;
- **exp:** Εκθετική συνάρτηση, δηλαδή  $e^x = \exp(x)$ . Π.χ.  $e^{-\sqrt{2}} = \exp(-2^{0.5})$ .
- **log:** Λογαριθμική συνάρτηση με βάση  $e$ . Π.χ.  $\log(\exp(1)) = 1$ .
- **round:** Στρογγυλοποίηση στον πλησιέστερο ακέραιο. Π.χ.  $\text{round}(2.8) = 3$ .
- **fix, rem:** Πηλίκο και υπόλοιπο διαίρεσης. Π.χ.  $\text{fix}(17/5) = 3$ ,  $\text{rem}(17, 5) = 2$ .
- **rand(1):** Παραγωγή ενός τυχαίου αριθμού από 0 έως 1.

## 2.2 Εργασίες κατανόησης θεωρίας

Εργασία 2.2α: Ορισμός μεταβλητών και εντολή ανάθεσης (=). Αριθμητικές πράξεις.

Οι μεταβλητές ορίζονται με χρήση του τελεστή ανάθεσης (=), είτε απ' ευθείας, είτε μέσω πράξεων με ή χωρίς άλλες μεταβλητές.

Αν δεν θέλουμε να εμφανιστεί η τιμή μιας μεταβλητής, βάζουμε ελληνικό ερωτηματικό (;) στο τέλος της εντολής ανάθεσης.

- Πληκτρολογήστε τα παρακάτω στο Command Window και παρατηρήστε τα αποτελέσματα:

Πληκτρολόγηση στο Command Window	Αποτελέσματα
<p>Ανάθεση αριθμού:</p> <pre>&gt;&gt; x = 6 &gt;&gt; y = 20;</pre>	<pre>x =      6</pre>

<p>Ανάθεση χαρακτήρα:  <b>&gt;&gt; c = 'A'</b></p> <p>Ανάθεση σειράς χαρακτήρων (συμβολοσειράς, string):  <b>&gt;&gt; myName = 'John'</b></p>	<p><b>c =</b> <b>A</b></p> <p><b>myName =</b> <b>John</b></p>
<p>Ανάθεση αποτελέσματος πράξεων με αριθμούς:  <b>&gt;&gt; a = 1+1</b></p>	<p><b>a =</b> <b>2</b></p>
<p>Ανάθεση αποτελέσματος πράξεων με τιμές μεταβλητών:  <b>&gt;&gt; z = x - y - 4</b></p>	<p><b>z =</b> <b>-18</b></p>
<p>Ανάθεση αποτελέσματος με αντικατάσταση μεταβλητής:  <b>&gt;&gt; z = x - z</b></p>	<p><b>z =</b> <b>-24</b></p>

**Εργασία 2.2β: Προτεραιότητα αριθμητικών πράξεων.**

1) ^ (δύναμη) 2) \*/ (επί διά) 3) +- (συν πλην) 4) - (αλλαγή προσήμου)

Σε ίδια προτεραιότητα, οι πράξεις γίνονται από αριστερά προς τα δεξιά.

Για αλλαγή προτεραιότητας χρησιμοποιούνται παρενθέσεις.

- Πληκτρολογήστε τα παρακάτω στο Command Window και παρατηρήστε τα αποτελέσματα:

<b>Πληκτρολόγηση στο Command Window</b>	<b>Αποτελέσματα</b>
<b>&gt;&gt; b = 2+3*4</b>	<b>b =</b> <b>14</b>
<b>&gt;&gt; b = (2+3)*4</b>	<b>b =</b> <b>20</b>
<b>&gt;&gt; c = -2^2 + 1, c = (-2)^2 + 1</b> (Δύο εντολές σε μία γραμμή, χωρισμένες με κόμμα)	<b>c =</b> <b>-3</b> <b>c =</b> <b>5</b>
<b>&gt;&gt; d = 16^0.5 + 4/2</b>	<b>d =</b> <b>6</b>
<b>&gt;&gt; d = 16^(1/2) + 4/2</b>	<b>d =</b> <b>6</b>
<b>&gt;&gt; d = 16^1/2 + 4/2</b>	<b>d =</b> <b>10</b>

**Εργασία 2.2γ: Επίλυση προβλημάτων.**

- Πληκτρολογήστε τα παρακάτω στο Command Window και παρατηρήστε τα αποτελέσματα:

Πληκτρολόγηση στο Command Window	Αποτελέσματα
<p>Πρόβλημα: Υπολογισμός των λύσεων της εξίσωσης <math>3x + 2 = 0</math>. Λύση: Λύνουμε πρώτα ως προς <math>x</math>: <math>x = -2/3</math>. Άρα: &gt;&gt; <b>x = -2/3</b></p>	<p><b>x =</b> <b>-0.6667</b></p>
<p>Πρόβλημα: i) Εύρεση λύσεων της εξίσωσης <math>x^2 - 5x + 6 = 0</math>. Λύση:  &gt;&gt; <b>a=1; b=-5; c=6;</b> (Τρεις εντολές σε μία γραμμή) Οι δύο λύσεις <math>\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}</math> είναι: &gt;&gt; <b>x1 = ( -b + (b^2-4*a*c)^0.5 ) / (2*a)</b> &gt;&gt; <b>x2 = ( -b - (b^2-4*a*c)^0.5 ) / (2*a)</b> Σημείωση: Για να αποφύγουμε να ξαναπληκτρολογήσουμε μια ίδια ή σχεδόν ίδια εντολή, πιέζουμε το άνω βέλος (<b>↑</b>) στο πληκτρολόγιο. Έτσι, εμφανίζεται η προηγούμενη εντολή (οπότε, στη συγκεκριμένη περίπτωση, αλλάζουμε μόνο το πρόσημο της τετραγωνικής ρίζας).</p> <p>ii) Εύρεση λύσεων της εξίσωσης <math>x^2 + 3x - 4 = 0</math>. Λύση με διακρίνουσα: &gt;&gt; <b>a=1; b=3; c=-4;</b> &gt;&gt; <b>D = b^2 - 4*a*c;</b> &gt;&gt; <b>x1 = (-b + D^0.5) / (2*a)</b> &gt;&gt; <b>x2 = (-b - D^0.5) / (2*a)</b></p>	<p><b>x1 =</b> <b>3</b> <b>x2 =</b> <b>2</b></p> <p><b>x1 =</b> <b>1</b> <b>x2 =</b> <b>-4</b></p>
<p>Πρόβλημα: Η μετατόπιση <math>h</math> σε συνάρτηση με το χρόνο <math>t</math> κατά την ελεύθερη πτώση ενός σώματος δίνεται από τη σχέση: <math>h = \frac{1}{2}gt^2</math>, όπου <math>g=9.81m/s^2</math> η επιτάχυνση της βαρύτητας. (i) Να βρείτε σε πόσο χρόνο θα φτάσει στο έδαφος ένα σώμα αν το αφήσουμε από ύψος 20m. (ii) Σε πόσο χρόνο θα έφτανε το σώμα στο έδαφος, αν βρισκόμασταν στη Σελήνη όπου η βαρύτητα είναι το 1/6 της γήινης; Λύση: i) Επιλύουμε αρχικά ως προς <math>t</math>: <math>h = \frac{1}{2}gt^2 \Rightarrow t = \sqrt{\frac{2h}{g}}</math> Ορισμός μεταβλητών: &gt;&gt; <b>h = 20; g = 9.81;</b>  Υπολογισμός χρόνου: &gt;&gt; <b>t = (2*h/g)^0.5</b></p>	<p><b>t =</b> <b>2.0193</b></p>

ii) Υπολογισμός της τιμής του  $g$  στη Σελήνη:

```
>> g = g/6;
```

Υπολογισμός του χρόνου πτώσης στη Σελήνη:

```
>> t = (2*h/g)^0.5
```

```
t =  
4.9462
```

Εργασία 2.2δ: Ειδικές μεταβλητές ans και pi.

**ans:** Προσωρινή μεταβλητή με το αποτέλεσμα της πιο πρόσφατης πράξης, αν δεν έχει ανατεθεί σε άλλη μεταβλητή.

**pi = 3.141592653589793:** Προσέγγιση του  $\pi$  με 15 δεκαδικά ψηφία.

- Πληκτρολογήστε τα παρακάτω στο Command Window και παρατηρήστε τα αποτελέσματα:

Command Window	Αποτελέσματα
Προσωρινή μεταβλητή ans: >> 5+3 >> ans + 2	ans = 8 ans = 10
Ειδική μεταβλητή pi:  >> format long; pi (Δύο εντολές σε μία γραμμή, χωρισμένες με ;)  >> format short; pi (Εμφανίζονται 4 δεκαδικά ψηφία λόγω του format short, αλλά στις πράξεις εσωτερικά χρησιμοποιούνται και τα 15.)	ans = 3.141592653589793  ans = 3.1416
Εύρεση περιφέρειας και εμβαδού κύκλου διαμέτρου 4.5m (περιφέρεια=2πR, εμβαδόν=πR <sup>2</sup> , όπου R=ακτίνα κύκλου): >> R = 4.5/2; >> perifereria = 2*pi*R >> embadon = pi*R^2	perifereia = 14.1372 embadon = 15.9043

Εργασία 2.2ε: Εντολή format.

**format short ή long:** Εμφάνιση 4 ή 15 δεκαδικών ψηφίων στο Command Window.

- Πληκτρολογήστε τα παρακάτω στο Command Window και παρατηρήστε τα αποτελέσματα:

Command Window	Αποτελέσματα
>> format long;  >> x = 15/7	x = 2.142857142857143

>> <code>format short;</code>	
>> <code>x = 15/7</code>	<code>x =</code> <code>2.1429</code>

**Εργασία 2.2στ: Εντολή `clc`.**

`clc`: καθαρισμός του *Command Window*.

- Πληκτρολογήστε τα παρακάτω στο *Command Window* και παρατηρήστε τα αποτελέσματα:

Command Window	Αποτελέσματα
>> <code>clc</code>	(Καθαρίζει το <i>Command Window</i> )

**Εργασία 2.2ζ: Εντολή `whos`.** Εμφάνιση μεταβλητών που υπάρχουν στη μνήμη.

- Ορίστε τις μεταβλητές `x`, `y` και `z` με τιμές της επιλογής σας (π.χ. `x=5; y=x^2; z=x+y;`). Στη συνέχεια, πληκτρολογήστε τις παρακάτω εντολές στο *Command Window* και παρατηρήστε τα αποτελέσματα:

Command Window	Αποτελέσματα																
>> <code>whos</code>	<table border="1"> <thead> <tr> <th>Name</th> <th>Size</th> <th>Bytes</th> <th>Class</th> </tr> </thead> <tbody> <tr> <td><code>x</code></td> <td><code>1x1</code></td> <td><code>8</code></td> <td><code>double</code></td> </tr> <tr> <td><code>y</code></td> <td><code>1x1</code></td> <td><code>8</code></td> <td><code>double</code></td> </tr> <tr> <td><code>z</code></td> <td><code>1x1</code></td> <td><code>8</code></td> <td><code>double</code></td> </tr> </tbody> </table>	Name	Size	Bytes	Class	<code>x</code>	<code>1x1</code>	<code>8</code>	<code>double</code>	<code>y</code>	<code>1x1</code>	<code>8</code>	<code>double</code>	<code>z</code>	<code>1x1</code>	<code>8</code>	<code>double</code>
Name	Size	Bytes	Class														
<code>x</code>	<code>1x1</code>	<code>8</code>	<code>double</code>														
<code>y</code>	<code>1x1</code>	<code>8</code>	<code>double</code>														
<code>z</code>	<code>1x1</code>	<code>8</code>	<code>double</code>														

**Εργασία 2.2η: Εντολή `clear`.** Καθαρισμός μεταβλητών από τη μνήμη.

- Αφού εκτελέσετε την εργασία 2.2ζ, πληκτρολογήστε τις παρακάτω εντολές στο *Command Window* και παρατηρήστε τα αποτελέσματα:

Command Window	Αποτελέσματα												
>> <code>clear x</code>  (Καθαρίζει η μεταβλητή <code>x</code> από τη μνήμη. Μπορούμε να το διαπιστώσουμε με <code>whos</code> ;)	(Αποτέλεσμα της εντολής <code>whos</code> : Λείπει η μεταβλητή <code>x</code> .)  <table border="1"> <thead> <tr> <th>Name</th> <th>Size</th> <th>Bytes</th> <th>Class</th> </tr> </thead> <tbody> <tr> <td><code>y</code></td> <td><code>1x1</code></td> <td><code>8</code></td> <td><code>double</code></td> </tr> <tr> <td><code>z</code></td> <td><code>1x1</code></td> <td><code>8</code></td> <td><code>double</code></td> </tr> </tbody> </table>	Name	Size	Bytes	Class	<code>y</code>	<code>1x1</code>	<code>8</code>	<code>double</code>	<code>z</code>	<code>1x1</code>	<code>8</code>	<code>double</code>
Name	Size	Bytes	Class										
<code>y</code>	<code>1x1</code>	<code>8</code>	<code>double</code>										
<code>z</code>	<code>1x1</code>	<code>8</code>	<code>double</code>										
>> <code>clear all</code>  (Καθαρίζει η μεταβλητή <code>x</code> από τη μνήμη. Μπορούμε να το διαπιστώσουμε με <code>whos</code> ;)	(Αποτέλεσμα της εντολής <code>whos</code> : Δεν εμφανίζεται καμμία μεταβλητή.)												
>> <code>whos</code>													

**Εργασία 2.2ι: Τριγωνομετρικές συναρτήσεις `sin`, `cos`, `tan`** (έγερση τριγωνομετρικών αριθμών μιας γωνίας σε ακτίνια). Τα ορίσματα (είσοδος) των συναρτήσεων μπαίνουν σε παρένθεση. Το αποτέλεσμα (έξοδος) παρουσιάζεται στο *Command Window*.

- Πληκτρολογήστε τα παρακάτω στο *Command Window* και παρατηρήστε τα αποτελέσματα:

Command Window	Αποτελέσματα
Εφαπτομένη $\pi/4$ rad: >> <code>tan(pi/4)</code>	ans = 1.0000
Ημίτονο $37.5^\circ$ (ακτίνια = μοίρες $\cdot \pi/180$ ): >> <code>sin(37.5*pi/180)</code>	ans = 0.6088

**Εργασία 2.2ια:** Μαθηματικές συναρτήσεις `exp`, `log`. Η `exp(x)` δίνει το  $e^x$ . Η `log(x)` δίνει το λογάριθμο του  $x$  με βάση το  $e$  (φυσικός ή νεπέρειος λογάριθμος).  
- Πληκτρολογήστε τα παρακάτω στο Command Window και παρατηρήστε τα αποτελέσματα:

Command Window	Αποτελέσματα
Εμφάνιση των αριθμών $e^2$ , $1/e$ και $e$ :  >> <code>exp(2), exp(-1), exp(1)</code> (Τρεις εντολές σε μία γραμμή, χωρισμένες με κόμμα ,)	ans = 7.3891 ans = 0.3679 ans = 2.7183
>> <code>log(2)</code>  >> <code>log(exp(1))</code>	ans = 0.6931 ans = 1

**Εργασία 2.2ιβ:** Συνάρτηση `rand` (παραγωγή τυχαίων αριθμών).  
- Πληκτρολογήστε τα παρακάτω στο Command Window και παρατηρήστε τα αποτελέσματα:

Command Window	Αποτελέσματα
Εμφάνιση ενός τυχαίου αριθμού από 0 έως 1: >> <code>rand(1)</code>	ans = 0.8147
Εμφάνιση ενός τυχαίου αριθμού από 0 έως 100: >> <code>100*rand(1)</code>	ans = 90.5792
Εμφάνιση ενός τυχαίου ακέραιου αριθμού από 0 έως 100: >> <code>round(100*rand(1))</code>	ans = 13
Ρίψη ζαριού: Εμφάνιση ενός τυχαίου ακέραιου αριθμού από 1 έως 6: >> <code>1 + round(5*rand(1))</code>	ans = 6

**Εργασία 2.2γ:** Συνάρτηση `round` (στρογγυλοποίηση στον πλησιέστερο ακέραιο). Το «.5» στρογγυλοποιείται προς τα «πάνω», δηλαδή προς τον πλησιέστερο ακέραιο κατ' απόλυτη τιμή.

- Πληκτρολογήστε τα παρακάτω στο *Command Window* και παρατηρήστε τα αποτελέσματα:

Command Window	Αποτελέσματα
>> <code>round(12.1)</code>	<code>ans =</code> 12
>> <code>round(-1.9)</code>	<code>ans =</code> -2
>> <code>round(3.5)</code>	<code>ans =</code> 4
>> <code>round(-3.5)</code>	<code>ans =</code> -4

**Εργασία 2.2δ:** Συναρτήσεις `fix` και `rem` (πηλίκο και υπόλοιπο διαίρεσης).

Η `fix(x)` στρογγυλοποιεί έναν αριθμό  $x$  προς το 0 (οπότε, η `fix(x/y)` μπορεί να χρησιμοποιηθεί για την εύρεση του πηλίκου της ακέραιας διαίρεσης  $x/y$ ).

Η `rem(x, y)` δίνει το υπόλοιπο της ακέραιας διαίρεσης  $x/y$  (ισοδύναμη εντολή με τη `rem`, για θετικούς αριθμούς, είναι η `mod`).

- Πληκτρολογήστε τα παρακάτω στο *Command Window* και παρατηρήστε τα αποτελέσματα:

Command Window	Αποτελέσματα
>> <code>fix(17/5)</code>	<code>ans =</code> 3
>> <code>rem(17,5)</code>	<code>ans =</code> 2
<p>Παράδειγμα: Πόσες ώρες και πόσα λεπτά είναι τα 115 λεπτά της ώρας;</p>	
>> <code>hours = fix(115/60)</code>	<code>hours =</code> 1
>> <code>minutes = rem(115,60)</code>	<code>minutes =</code> 55

### 2.3 Εργαστηριακές ασκήσεις

Συμπληρώστε τις κατάλληλες εντολές στο *Command Window*, ώστε να λάβετε τα αποτελέσματα που παρουσιάζονται παρακάτω:

**Άσκηση 2.3α:** Βρείτε το εμβαδόν (σε  $cm^2$ ) τραπεζοειδούς πλακέτας με μήκος μεγάλης βάσης  $B=28cm$ , μικρής βάσης  $b=17cm$  και ύψους  $h=5.5cm$  (εμβαδόν τραπεζίου  $E = \frac{B+b}{2} \cdot h$ ):

Command Window (συμπληρώστε)	Αποτελέσματα
Ορισμός αρχικών μεταβλητών: >> B = ..... >> ..... >> .....	B = 28 b = 17
Υπολογισμός εμβαδού >> E = .....	Y = 5.5000 E = 123.7500

**Άσκηση 2.3β:** Η μετατόπιση ( $x$ ) σε συνάρτηση με το χρόνο ( $t$ ) κατά την ευθύγραμμη ομαλά επιταχυνόμενη κίνηση ενός σώματος δίνεται από τη σχέση:  $x = v_0 t + \frac{1}{2} a t^2$ , όπου  $a$  η επιτάχυνση του σώματος και  $v_0$  η αρχική του ταχύτητα. Να βρείτε (i) την απόσταση που θα διανύσει ένα αυτοκίνητο που υποτίθεται ότι κινείται σε ευθύγραμμο δρόμο με σταθερή επιτάχυνση  $3m/s^2$  σε χρόνο  $17s$ , ξεκινώντας με αρχική ταχύτητα  $12m/s$  και (ii) πόση θα έπρεπε να ήταν η επιτάχυνσή του ώστε να είχε διανύσει απόσταση  $1000m$ , στον ίδιο χρόνο και με ίδια αρχική ταχύτητα:

Command Window (συμπληρώστε)	Αποτελέσματα
i) Ορισμός αρχικών μεταβλητών: >> ..... Υπολογισμός μετατόπισης: >> .....	x = 637.5000
ii) Ορισμός νέας τιμής του <b>x</b> : >> ..... Υπολογισμός νέας τιμής της επιτάχυνσης: >> .....	a = 5.5087

**Άσκηση 2.3γ:** Η σχέση βαθμών Φαρενάιτ ( $F$ ) και Κελσίου ( $C$ ) είναι η εξής:  
 $F = \frac{9}{5} C + 32$ .

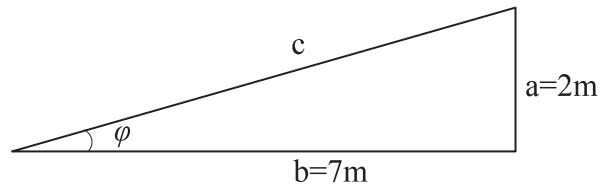
- Πληκτρολογήστε τις κατάλληλες πράξεις στο *Command Window* και συμπληρώστε κατάλληλα τα παρακάτω κενά ώστε (i) να βρείτε σε πόσους βαθμούς Φαρενάιτ αντιστοιχούν οι 50 βαθμοί Κελσίου και (ii) αφού επιλύσετε την εξίσωση ως προς  $C$ , να βρείτε σε πόσους βαθμούς Κελσίου αντιστοιχούν οι 50 βαθμοί Φαρενάιτ:



Command Window (συμπληρώστε)	Αποτελέσματα
i) Εύρεση βαθμών Φαρενάιτ για 50 βαθμούς Κελσίου:  >> .....	ans = 122
ii) Εύρεση βαθμών Κελσίου για 50 βαθμούς Φαρενάιτ:  >> .....	ans = 10

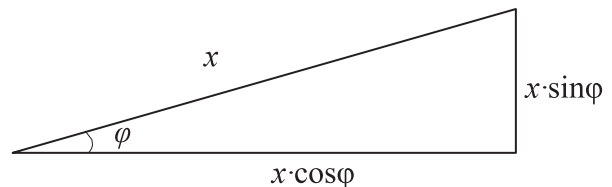
**Άσκηση 2.3δ:** Να υπολογίσετε, για το διπλανό τρίγωνο:

- i) το μήκος της υποτείνουσας  $c$  και
- ii) την εφαπτομένη της γωνίας  $\varphi$ .



Command Window (συμπληρώστε)	Αποτελέσματα
i) Ορισμός αρχικών μεταβλητών: >> ..... Υποτείνουσα: >> .....	c = 7.2801
ii) Εφαπτομένη της γωνίας $\varphi$ :  >> .....	ans = 0.2857

**Άσκηση 2.3ε:** Η υποτείνουσα ενός ορθογωνίου τριγώνου είναι 5m και η μία οξεία γωνία είναι  $\varphi=20^\circ$ . Να υπολογίσετε τα μήκη των δύο κάθετων πλευρών καθώς και το εμβαδόν του τριγώνου:



Command Window (συμπληρώστε)	Αποτελέσματα
Απέναντι πλευρά από τη γωνία $\varphi$ :  >> .....	ans = 1.7101
Προσκειμένη πλευρά στη γωνία $\varphi$ :  >> .....	ans = 4.6985
Εμβαδόν τριγώνου:  >> .....	ans = 4.0174

**Άσκηση 2.3στ:** (i) Καθαρίστε από τη μνήμη όλες τις προηγούμενες μεταβλητές. (ii) Εμφανίστε τον αριθμό  $e$  (τη βάση των νεπερείων λογαρίθμων) με 15 δεκαδικά ψηφία, αποθηκεύοντάς τον σε μια μεταβλητή με όνομα "e". (Να γράψετε όλες τις εντολές για την εκτέλεση των (i) και (ii) σε μία μόνο γραμμή στο Command Window.)

Command Window (συμπληρώστε)	Αποτελέσματα
>> .....	e = 2.718281828459046

**Άσκηση 2.3ζ:** Ένας κινητήρας περιστρέφεται με γωνιακή ταχύτητα 352 μοίρες ανά δευτερόλεπτο. Αν ο κινητήρας εργαστεί για 23 sec, να βρείτε πόσες πλήρεις περιστροφές και πόσες υπόλοιπες μοίρες έχει εκτελέσει:

Command Window (συμπληρώστε)	Αποτελέσματα
Αρχικά βρίσκουμε το σύνολο μοιρών που περιστράφηκε ο κινητήρας:  >> .....	moires = 8096
Αριθμός πλήρων περιστροφών:  >> .....	ans = 22
Υπόλοιπη γωνία σε μοίρες:  >> .....	ans = 176

**Άσκηση 2.3η:** Να γράψετε κατάλληλες εντολές στο Command Window που να παράγουν (i) έναν τυχαίο αριθμό από 0 έως 20, (ii) έναν τυχαίο αριθμό από 1 έως 20, (iii) έναν τυχαίο ακέραιο αριθμό από 1 έως 20.

Command Window (συμπληρώστε)	Αποτελέσματα
i) Τυχαίος αριθμός από 0 έως 20:  >> .....	ans = 16.294
ii) Τυχαίος αριθμός από 1 έως 20:  >> .....	ans = 18.21
iii) Τυχαίος ακέραιος αριθμός από 1 έως 20:  >> .....	ans = 3

## Δημιουργία και εκτέλεση προγραμμάτων.


Εντολές εισόδου από το πληκτρολόγιο και εξόδου στην οθόνη.


Σε αυτήν την ενότητα θα δημιουργήσετε και θα εκτελέσετε τα πρώτα σας απλά προγράμματα. Επίσης, θα δείτε τις εντολές εισόδου δεδομένων από το πληκτρολόγιο και εξόδου αποτελεσμάτων στην οθόνη. Συγκεκριμένα, θα εξοικειωθείτε με:

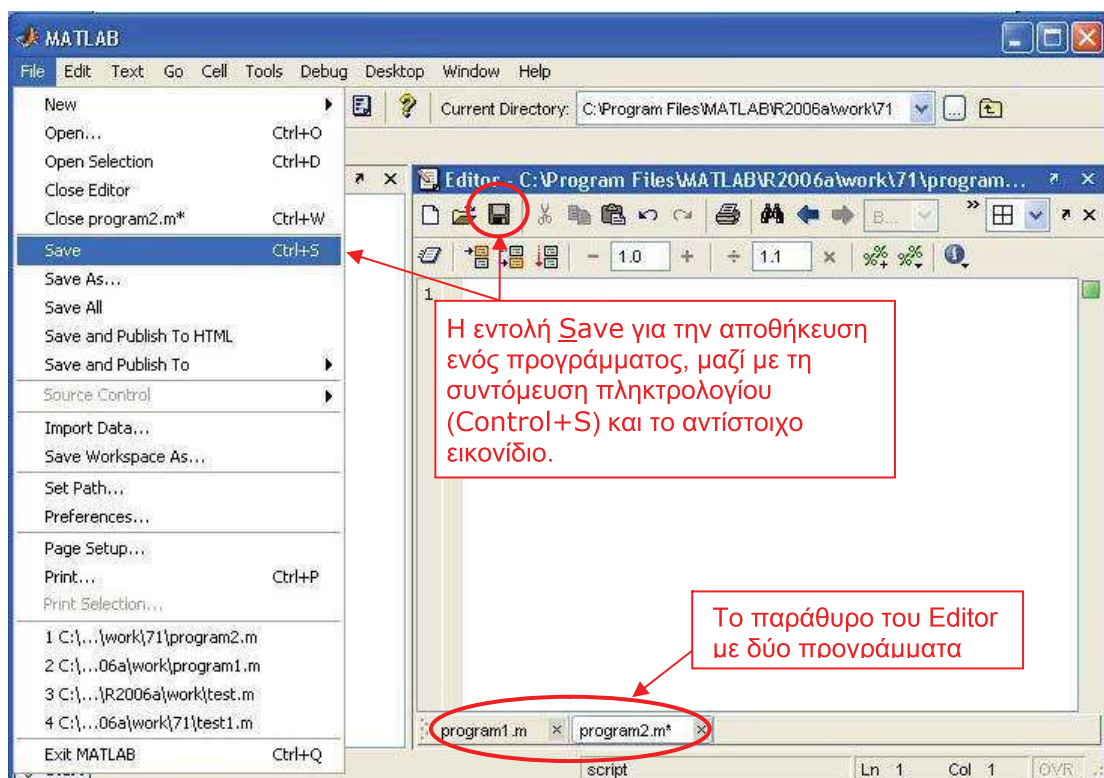
- Τη λειτουργία του Editor του MATLAB για τη δημιουργία, αποθήκευση και εκτέλεση ενός προγράμματος.
- Τις λειτουργίες των Current Directory και Path.
- Τη δημιουργία, την αποθήκευση και την εκτέλεση ενός προγράμματος καθώς και τον τρόπο με τον οποίο γίνεται το γράψιμο και η επεξεργασία του κώδικά του.
- Τις εντολές `input`, `disp` και `fprintf`.

### 1.1 Συνοπτική θεωρία

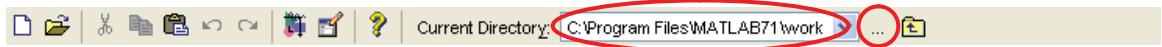
**Editor:** Το MATLAB έχει ενσωματωμένο Editor (Διορθωτή), κατάλληλο για τη συγγραφή του κώδικα των προγραμμάτων. Στον Editor του MATLAB μπορούν να ανοίξουν ένα ή περισσότερα «παράθυρα» για συγγραφή κώδικα. Τα παράθυρα μέσα στον Editor μπορούν να είναι σε όποια θέση θέλουμε, π.χ. σε οριζόντια ή κατακόρυφη παράθεση ή το ένα πίσω από το άλλο. Μπορούμε να επιλέξουμε το πρόγραμμα που θέλουμε να επεξεργαστούμε κάνοντας κλικ στο ταμπλάκι του, που βρίσκεται στο κάτω μέρος του Editor.


**Δημιουργία νέου προγράμματος:** Μενού `File | New ▶ M-File` (ή `Control+N` ή εικονίδιο ). Τότε ανοίγει ένα κενό παράθυρο για το γράψιμο του κώδικα ενός προγράμματος. Τα προγράμματα του MATLAB έχουν επέκταση `.m` και ονομάζονται m-files.

**Αποθήκευση προγράμματος:** Μενού `File | Save` (ή `Control+S` ή εικονίδιο ). Κατά τη συγγραφή ενός προγράμματος πρέπει να το αποθηκεύουμε συχνά, για λόγους ασφαλείας. Το όνομα ενός προγράμματος πρέπει να αρχίζει από λατινικό γράμμα (πεζό ή κεφαλαίο) και μπορεί να περιλαμβάνει λατινικά γράμματα (`a...z`, `A...Z`), αριθμητικά ψηφία (`0...9`) ή κάτω παύλες (`_`, underscore).

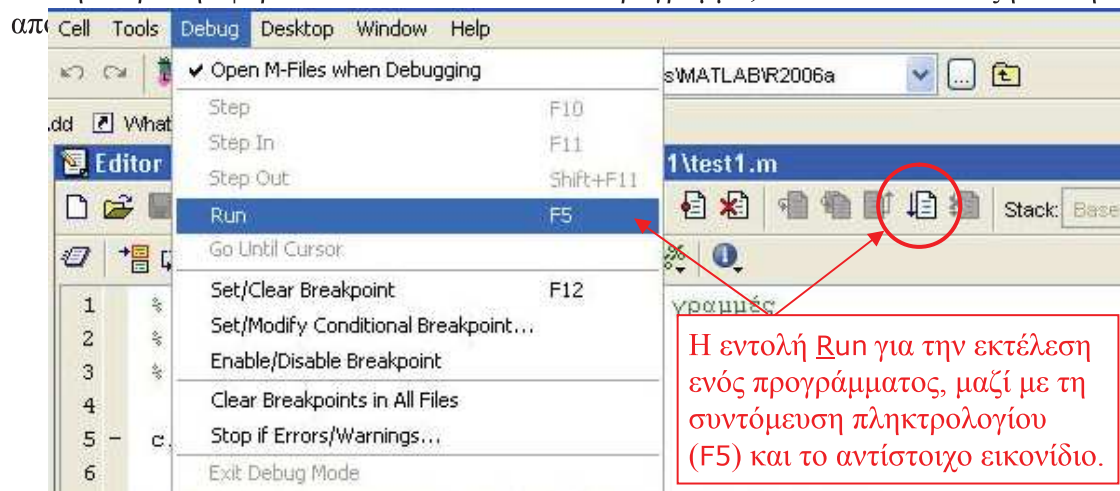


Current Directory: Αν δεν δηλωθεί διαφορετικά, ένα πρόγραμμα αποθηκεύεται σε ένα φάκελο που λέγεται Current Directory (φαίνεται στη γραμμή εργαλείων). Μπορούμε να αλλάξουμε το Current Directory πατώντας το κουμπί με τις τρεις τελείες (...) που βρίσκεται δεξιά και να επιλέξουμε κάποιον άλλο φάκελο:

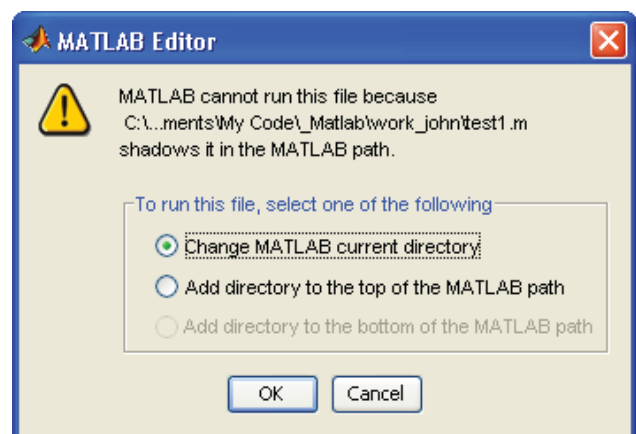


Εκτέλεση προγράμματος: Μενού **Debug** | **Run** (ή F5, ή εικονίδιο ). Για να μπορεί να εκτελεστεί, το πρόγραμμα πρέπει να βρίσκεται στο Current Directory ή στο Path του MATLAB. Το Path είναι ένα σύνολο από φακέλους των Windows (μπορούμε να προσθέσουμε ή να αφαιρέσουμε από το μενού **File** | **Set Path...**), στους οποίους μπορούμε να αποθηκεύσουμε για λόγους οργάνωσης τα προγράμματά μας.

Την πρώτη φορά που εκτελείται ένα πρόγραμμα, το MATLAB ζητά την



Κάθε χρήστης μπορεί για λόγους οργάνωσης να αποθηκεύει τα προγράμματά του σε διαφορετικούς φακέλους. Όταν ανοίξουμε ένα πρόγραμμα, για να μπορεί να τρέξει πρέπει να αλλάξουμε το Current Directory ώστε να γίνει ίδιο με το φάκελο στον οποίο βρίσκεται το πρόγραμμα. Αν δεν συμβαίνει αυτό, τότε όταν προσπαθήσουμε να τρέξουμε το πρόγραμμα το MATLAB μας «ρωτάει» με κατάλληλο παράθυρο διαλόγου που ανοίγει εκείνη τη στιγμή (βλέπε διπλανή εικόνα) αν θέλουμε να αλλάξουμε το Current Directory ώστε να γίνει ίδιο με του φακέλου στον οποίο βρίσκεται το προγράμμα μας, οπότε αρκεί να πιάσουμε με το ποντίκι το

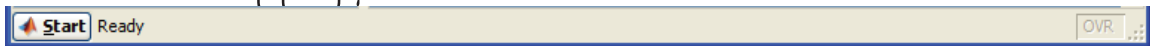


Το πλήθος των παραθύρων εργασίας του MATLAB δυσκολεύει τον αρχάριο χρήστη, έτσι αρχικά συστήνεται η εμφάνιση μόνο του Command Window (όπου δίνουμε εντολές και παρουσιάζονται αποτελέσματα) και του Editor (όπου γράφουμε τα προγράμματά μας). Η μπάρα κατάστασης (status bar): Στη μπάρα κατάστασης εμφανίζεται η κατάσταση του MATLAB σε κάθε στιγμή. Η κατάσταση αυτή μπορεί να είναι:

- Initializing...: Κατάσταση αρχικοποίησης του MATLAB κατά την έναρξή του.



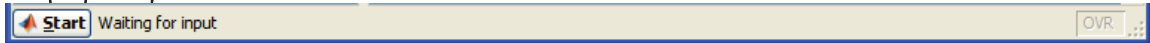
- Ready: Κατάσταση ετοιμότητας. Το MATLAB δεν εκτελεί τίποτε και ο χρήστης έχει τον έλεγχο. Μπορούμε να εκτελέσουμε οποιοδήποτε πρόγραμμα, εντολή ή ενέργεια.



- Busy: Κατάσταση εργασίας: Το MATLAB εκτελεί κάποιο πρόγραμμα.



- Waiting for input: Κατάσταση αναμονής για είσοδο δεδομένων από το πληκτρολόγιο.





Στις δύο τελευταίες καταστάσεις (Busy / Waiting for input) ο χρήστης μπορεί να ανακτήσει τον έλεγχο, διακόπτοντας ό,τι εκτελείται εκείνη τη στιγμή, πατώντας το συνδυασμό πλήκτρων Control+Break, αφού πρώτα ενεργοποιήσει το Command Window.

### Συνοπτικός πίνακας ενεργειών δημιουργίας, αποθήκευσης και συγγραφής προγράμματος

Ακολουθούν οι ενέργειες που σχετίζονται με τη δημιουργία ενός προγράμματος και την επεξεργασία του κώδικά του, καθώς και το πώς ενεργοποιούνται: (α) ή με εντολές του μενού, (β) με συντομεύσεις πληκτρολογίου, ή (γ) πατώντας σε ένα εικονίδιο στη γραμμή εργαλείων:

Ενέργεια	Μενού	Πληκτρολόγιο	Εικονίδιο
----------	-------	--------------	-----------






#### **Δημιουργία και αποθήκευση προγράμματος**

Νέο πρόγραμμα	File   <u>N</u> ew ▶ M-File	Control+N	
Αποθήκευση προγράμματος	File   <u>S</u> ave	Control+S	

## Εκτέλεση προγράμματος

Εκτέλεση προγράμματος	Debug   Run	F5	
-----------------------	-------------	----	---

## Γράψιμο και επεξεργασία κώδικα προγράμματος

Αποκοπή	Edit   Cut	Control+X	
Αντιγραφή	Edit   Copy	Control+C	
Επικόλληση	Edit   Paste	Control+V	
Αναίρεση ενέργειας	Edit   Undo	Control+Z	
Ακύρωση αναίρεσης	Edit   Redo	Control+Y	
Επιλογή όλων	Edit   Select All	Control+A	
Αυτόματες εσοχές	Text   Smart Indent	Control+I	

### Εργασία 1.1α. Δημιουργία και αποθήκευση προγραμμάτων στον Editor.

- Κλείστε όλα τα παράθυρα, και αφήστε ανοικτά μόνο το Command Window και τον Editor, το ένα δίπλα στο άλλο και συνδεδεμένα μέσα στο Desktop του MATLAB.
- Δημιουργήστε 2 νέα παράθυρα για γράψιμο κώδικα προγράμματος. Γράψτε κάτι (οτιδήποτε) μέσα σ' αυτά και αποθηκεύστε τα με ένα όνομα της επιλογής σας. Μεταβείτε από το ένα παράθυρο προγράμματος στο άλλο.
- Ανοίξτε το Current Τα περιεχόμενα του Current Directory μπορούμε να τα δούμε με δύο τρόπους: (i) Ανοίγοντας το παράθυρο επισκόπησης του Current Directory μέσα από το MATLAB (Desktop | Current Directory). (ii) Ανοίγοντας τον αντίστοιχο φάκελο από τα Windows.

### Είσοδος δεδομένων από το πληκτρολόγιο:

#### input

```
μεταβλητή αριθμού = input('μήνυμα');  
μεταβλητή κειμένου = input('μήνυμα', 's');
```

### Έξοδος αποτελεσμάτων στην οθόνη (στο Command Window):

#### **disp και fprintf**

Η disp εμφανίζει κείμενο ή την τιμή μιας μεταβλητής.

Η fprintf εμφανίζει κείμενο μαζί με τιμές μεταβλητών.

```
disp(μεταβλητή);
```

```
disp('κείμενο');
```

```
fprintf('κείμενο με ειδικούς χαρακτήρες', μεταβλητές);
```

Ειδικοί χαρακτήρες για την fprintf:

%d = απεικόνιση σε μορφή ακεραίου αριθμού

%f = απεικόνιση σε μορφή αριθμού με δεκαδικά ψηφία

%X.Yf = απεικόνιση σε μορφή αριθμού με X συνολικές θέσεις και Y δεκαδικά ψηφία

%c = απεικόνιση σε μορφή χαρακτήρα


%s = απεικόνιση σε μορφή σειράς χαρακτήρων (συμβολοσειράς, string)

\n = αλλαγή γραμμής

\t = στηλοθέτης (tab).

## 1.2 Εργασίες κατανόησης θεωρίας

Εργασία 1.2α: Δημιουργία, αποθήκευση και εκτέλεση ενός προγράμματος. Εντολή disp. Θα δημιουργήσουμε ένα απλό πρόγραμμα που θα εμφανίζει ένα μήνυμα στο Command Window με την εντολή disp, η οποία χρησιμοποιείται για απεικόνιση κειμένου ή τιμών μεταβλητών:

- Δημιουργήστε το παράθυρο ενός νέου προγράμματος στον Editor (μενού File | New ► M-File, ή πληκτρολόγιο Control+N, ή εικονίδιο .
- Γράψτε στο κενό παράθυρο την εντολή disp('Hello!');
- Αποθηκεύστε το πρόγραμμα με όνομα program32a.m.
- Εκτελέστε το νέο πρόγραμμα και παρατηρήστε το αποτέλεσμα στο Command Window.
- Παρατηρήστε την ύπαρξη του νέου προγράμματος στο Current Directory.

Πληκτρολόγηση στον Editor program32a.m	Αποτελέσματα
<pre>disp('Hello!');</pre>	<pre>Hello! &gt;&gt;</pre>

Εργασία 1.2β: Χρήση μεταβλητών μέσα σε πρόγραμμα.

- Δημιουργήστε ένα νέο πρόγραμμα και αποθηκεύστε το με όνομα program32b.m.
- Γράψτε στο πρόγραμμα τα ακόλουθα, για τον υπολογισμό των δευτερολέπτων μιας ημέρας, και εκτελέστε το:

program32b.m	Αποτελέσματα
<pre>SecondsPerHour = 60*60; HoursPerDay = 24;  SecondsPerDay = SecondsPerHour*HoursPerDay;  disp('Number of seconds in one day:'); disp(SecondsPerDay);</pre>	<pre>Number of seconds in one day: 86400  &gt;&gt;</pre>

Εργασία 1.2γ: Εντολή fprintf. Η fprintf χρησιμοποιείται για απεικόνιση κειμένου μαζί με τιμές μεταβλητών. Ο τρόπος εμφάνισης των μεταβλητών καθορίζεται από ειδικούς χαρακτήρες, οι οποίοι μπαίνουν μέσα στο κείμενο, στη θέση και με τη σειρά όπου θέλουμε να απεικονιστούν οι μεταβλητές. Οι πιο συνηθισμένοι ειδικοί χαρακτήρες είναι:

%d=απεικόνιση ακεραίου, %f=απεικόνιση δεκαδικών ψηφίων, %s=απεικόνιση συμβολοσειράς, \n=αλλαγή γραμμής.

- Δημιουργήστε ένα νέο πρόγραμμα και αποθηκεύστε το με όνομα program32c.m:

program32c.m
<pre>clc;  name = 'John'; weight = 70; height = 1.75;  fprintf('Name = %s.\nWeight = %d. Height = %.2f.', name, weight, height);</pre>

## Αποτελέσματα

Name = John.

Weight = 70. Height = 1.75.>>

Παρατήρηση: Στο πρόγραμμα χρησιμοποιήσαμε στην πρώτη γραμμή την εντολή `clc`, ώστε να καθαρίσει το *Command Window* και να βλέπουμε καθαρά τα αποτελέσματα.

Εργασία 1.2δ: Εντολή `input`. Η `input` χρησιμοποιείται για είσοδος δεδομένων από το πληκτρολόγιο. Για εισαγωγή συμβολοσειράς (*string*) χρησιμοποιείται η παράμετρος `'s'`:

- Δημιουργήστε ένα νέο πρόγραμμα και αποθηκεύστε το με όνομα `program32d.m`. Γράψτε στο πρόγραμμα τα ακόλουθα, εκτελέστε το και δώστε από το πληκτρολόγιο το όνομά σας, το βάρος και το ύψος σας. Το πρόγραμμα θα υπολογίσει το Δείκτη Σωματικού Βάρους σας:

### program32d.m

```
clc;

name = input('Name? ', 's');
weight = input('Weight? ');
height = input('Height? ');

bmi = weight/height^2;

fprintf('\nThanks %s.\n', name);
fprintf('Your BMI is %.1f.\n', bmi);
```

### Αποτελέσματα (μπορεί να διαφέρουν, διότι εξαρτώνται από τα στοιχεία που θα εισάγετε)

Name? George

Weight? 78

Height? 1.82

Thanks George.

Your BMI is 23.5.

>>

Σημείωση: Ο Δείκτης Σωματικού Βάρους (*Body Mass Index, BMI*) ενός ατόμου ισούται με το πηλίκο του βάρους (σε *kg*) προς το τετράγωνο του ύψους (σε *m*):  $BMI = \text{Βάρος}(\text{kg}) / \text{Ύψος}(\text{m})^2$ .

Ο *BMI* θεωρείται ένας απλός αλλά αρκετά αξιόπιστος δείκτης για την εκτίμηση της παχυσαρκίας ενήλικων. Οι κατηγορίες αναλόγως του *BMI* είναι (<http://www.nhlbisupport.com/bmi/>):

Λιπόβαρος:  $BMI < 18.5$

Κανονικό βάρος:  $18.5 \leq BMI < 25$

Υπέρβαρος:  $25 \leq BMI < 30$

Παχύσαρκος:  $BMI \geq 30$



Παρατηρήσεις για την οργάνωση ενός προγράμματος:

1) Συνήθως στην αρχή του προγράμματος βάζουμε και την εντολή `clear all`, **ώστε να καθарίσουν όλες οι τυχόν προηγούμενες μεταβλητές από τη μνήμη.**

2) Για να είναι πιο κατανοητό ένα πρόγραμμα, πρέπει να το εμπλουτίζουμε με σχόλια, που επεξηγούν συνοπτικά τη λειτουργία του. Οτιδήποτε βρίσκεται σε μια γραμμή μετά από το χαρακτήρα «`%`» θεωρείται από το **MATLAB** ως σχόλιο και αγνοείται κατά την εκτέλεση του προγράμματος.

3) Στο τελευταίο πρόγραμμα μπορούμε να διακρίνουμε και τα βασικά τμήματα που αποτελούν ένα πρόγραμμα: «Είσοδος δεδομένων», «Επεξεργασία» και «Εξόδος αποτελεσμάτων».

Σύμφωνα με τις προηγούμενες παρατηρήσεις, το τελευταίο πρόγραμμα μπορεί να γραφτεί ως εξής:

**program32d\_1.m**

```
clc;
clear all;

% Eισodos dedomenwn
name = input('Name? ','s');
weight = input('Weight? ');
height = input('Height? ');

% Ypologismos BMI
bmi = weight/height^2;

% Eksodos apotelesmatwn
fprintf('\nThanks %s.\n', name);
fprintf('Your BMI is %.1f.\n', bmi);
```

# ΠΙΝΑΚΕΣ

# Μιγαδικοί αριθμοί

- $\text{abs}(1+i)$

ans =

1.4142

- $(1+i)'$

ans =

1.0000 - 1.0000i

# Πίνακες

- Για να εισάγετε τον πίνακα του Dürer, απλά δώστε στο Παράθυρο Εντολών

```
A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]
```

- Το MATLAB δίνει:

```
A =
```

```
16   3   2  13
 5  10  11   8
 9   6   7  12
 4  15  14   1
```

# Πίνακες

- `sum(A)`

`ans =`

34 34 34 34

- `A'` (ανάστροφος πίνακας)

`ans =`

16	5	9	4
3	10	6	15
2	11	7	14
13	8	12	1

!!! ΠΡΟΣΟΧΗ ΕΑΝ Ο ΠΙΝΑΚΑΣ ΕΧΕΙ ΜΙΓΑΔΙΚΟΥΣ ΑΡΙΘΜΟΥΣ !!!  
!!! ΤΟΤΕ Ο ΑΝΑΣΤΡΟΦΟΣ ΔΙΝΕΤΑΙ ΑΠΟ ΤΟΝ `A'` !!!

# Πίνακες

- `sum(A')`

`ans =`

34

34

34

34

- `diag(A)`

`ans =`

16

10

7

1

- `sum(diag(A))`

`ans =`

34

# Πίνακες

- Για να προσθέσετε 2 πίνακες θα πρέπει να έχουν τις ίδιες διαστάσεις.

- $A + A'$

ans =

32	8	11	17
8	20	17	23
11	17	14	26
17	23	26	2

- Για να πολλαπλασιάσετε 2 πίνακες θα πρέπει, εάν δεν είναι τετραγωνικοί, ο δεύτερος να έχει τόσες γραμμές όσες στήλες ο πρώτος.

# Πίνακες

- $A^T * A$

ans =

378	212	206	360
212	370	368	206
206	368	370	212
360	206	212	378

- $A .* A$  (διαφορετικό από  $A * A = A^2$ !)

ans =

256	9	4	169
25	100	121	64
81	36	49	144
16	225	196	1



# Πίνακες

- $\det(A)$  (ορίζουσα)

```
ans =  
    0
```

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

- $\text{inv}(B)$  (Αντίστροφος)

```
ans =  
 -2.0000    1.0000  
  1.5000  -0.5000
```

# Πίνακες

- $A(1,4) + A(2,4) + A(3,4) + A(4,4)$

ans =

34

- $X = A;$
- $X(4,5) = 17$

X =

16	3	2	13	0
5	10	11	8	0
9	6	7	12	0
4	15	14	1	17

# Πίνακες

- `Z = zeros(2,4)`

ans =

```
0 0 0 0
0 0 0 0
```

- `F = 5*ones(3,3)`

ans =

```
5 5 5
5 5 5
5 5 5
```

# Πίνακες

- 1:10

1 2 3 4 5 6 7 8 9 10

- 100:-7:50

100 93 86 79 72 65 58 51

- 0:pi/4:pi

0 0.7854 1.5708 2.3562 3.1416

- $\text{sum}(A(1:4,4))=\text{sum}(A(:,\text{end}))$

# Πίνακες

- `vertcat([1 2; 3 4], [5 6; 7 8])`

ans =

1	2
3	4
5	6
7	8

- `horzcat([1 2; 3 4], [5 6; 7 8])`

ans =

1	2	5	6
3	4	7	8

# Πίνακες

- `A = [ 3 7 5; 6 8 3; 0 4 2];`

- `sort(A)`

- `ans =`

```
0 4 2
3 7 3
6 8 5
```

- `sort(A,'descend')`

- `ans =`

```
6 8 5
3 7 3
0 4 2
```

## 1 Πίνακες

Σε αυτήν την ενότητα θα εξοικειωθείτε με την έννοια των πινάκων στον προγραμματισμό (χωρίς τον ιδιαίτερο τρόπο χειρισμού των πινάκων στο MATLAB), και συγκεκριμένα θα δείτε:

- πώς ορίζεται ένας πίνακας, μιας ή δύο διαστάσεων
- πώς χειριζόμαστε και πώς αναφερόμαστε σε μεμονωμένα στοιχεία ενός πίνακα
- πώς σαρώνουμε έναν ολόκληρο πίνακα για την εύρεση της μέσης τιμής των στοιχείων του
- πώς μπορούμε να χειριστούμε συγκεκριμένες γραμμές ή στήλες ενός πίνακα

### 1.1 Συνοπτική θεωρία

Πίνακας (matrix) ονομάζεται μια ομάδα μεταβλητών ίδιου τύπου με ένα κοινό όνομα. Οι επιμέρους απλές μεταβλητές του πίνακα λέγονται στοιχεία του πίνακα και το πλήθος τους λέγεται μέγεθος του πίνακα. Κάθε στοιχείο χαρακτηρίζεται από το όνομα του πίνακα και από τους δείκτες του (θετικοί ακέραιοι που μπαίνουν σε παρένθεση και δηλώνουν τη θέση του στοιχείου).

Μονοδιάστατοι πίνακες: Ένας πίνακας μιας διάστασης (λέγεται και *διάνυσμα*, *vector*) μπορεί να είναι πίνακας-γραμμή ή πίνακας-στήλη, ανάλογα με το πώς διατάσσονται τα στοιχεία του. Το μέγεθος ενός μονοδιάστατου πίνακα λέγεται και μήκος του πίνακα.

Δισδιάστατοι πίνακες: Τα στοιχεία ενός δισδιάστατου πίνακα διατάσσονται σε γραμμές και στήλες. Ένας πίνακας με N γραμμές και M στήλες ονομάζεται πίνακας διαστάσεων  $N \times M$  (η πρώτη διάσταση είναι πάντα οι γραμμές και η δεύτερη οι στήλες). Το πλήθος των στοιχείων ενός δισδιάστατου πίνακα ισούται με το γινόμενο  $N \cdot M$ .

Συναρτήσεις πινάκων:

`length`: Επιστρέφει τη μεγαλύτερη από τις διαστάσεις ενός πίνακα. Στους μονοδιάστατους πίνακες δίνει απ' ευθείας το πλήθος των στοιχείων τους.

`size`: Επιστρέφει τις διαστάσεις ενός πίνακα. Χρησιμοποιείται συχνά για την εύρεση της τελικής τιμής των επαναληπτικών βρόχων (`for`) κατά τη σάρωση ενός πίνακα, και για την εύρεση του πλήθους των στοιχείων ενός πίνακα (ισούται με το γινόμενο των διαστάσεών του).

## 1.2 Εργασίες κατανόησης θεωρίας

**Εργασία 1.2α: Ορισμός μονοδιάστατου πίνακα.** Τα στοιχεία ενός πίνακα περιλαμβάνονται σε αγκύλες [...]. Τα στοιχεία ενός πίνακα-γραμμής χωρίζονται με κενό διάστημα ή κόμμα (,). Τα στοιχεία ενός πίνακα-στήλης χωρίζονται με ελληνικό ερωτηματικό (;) ή με αλλαγή γραμμής (↵Enter).

- Πληκτρολογήστε τα παρακάτω στο Command Window και παρατηρήστε τα αποτελέσματα:

Πληκτρολόγηση στο Command Window	Αποτελέσματα
>> A = [8 7 5 9 10]	A = 8 7 5 9 10
>> A = [8, 7, 5, 9, 10]	A = 8 7 5 9 10
>> B = ['Η' 'e' 'l' 'l' 'o' '!']	B = Hello!
>> C = [-1;0;4]	C = -1 0 4
>> D = [12 -3 7]	D = 12 -3 7

**Εργασία 1.2β: Αναφορά στα στοιχεία μονοδιάστατου πίνακα.** Η αναφορά σε ένα στοιχείο ενός μονοδιάστατου πίνακα γίνεται με το όνομα του πίνακα και το δείκτη του μέσα σε παρένθεση.

- Πληκτρολογήστε τα παρακάτω στο Command Window και παρατηρήστε τα αποτελέσματα:

Πληκτρολόγηση στο Command Window	Αποτελέσματα
>> A = [8 7 5 9 10];	ans = 7
>> A(2)	x = 89
>> x = A(3) + 12*A(2)	A = 8 7 -1 9 10
>> A(3) = -1	



**Εργασία 1.2γ: Σάρωση μονοδιάστατου πίνακα.** Η προσπέλαση όλων των στοιχείων ενός πίνακα λέγεται *σάρωση*. Η σάρωση ενός μονοδιάστατου πίνακα μπορεί να γίνει με ένα επαναληπτικό βρόχο `for`. Στο παρακάτω πρόγραμμα σαρώνεται ένας μονοδιάστατος πίνακας για την εύρεση της μέσης τιμής των στοιχείων του.

- Πληκτρολογήστε το παρακάτω πρόγραμμα στον *Editor* και αποθηκεύστε το με όνομα `program82a.m`. Το πρόγραμμα υπολογίζει τη μέση τιμή ενός μονοδιάστατου πίνακα. Κατανοήστε τη δομή και τη λειτουργία του και στη συνέχεια εκτελέστε το (`Debug|Run` ή `F5`).

Πληκτρολόγηση στον Editor <code>program82a.m</code>	Αποτελέσματα
<pre>% Find mean value of a vector clc clear all  % Matrix definition A = [8 7 -1 9 6];  % Calculate mean value s = 0; for i=1:5     s = s + A(i); end s = s/5;  % Display mean value fprintf('Mean value of A is %.1f\n', s);</pre>	<pre>Mean value of A is 5.8 &gt;&gt;</pre>

**Εργασία 1.2δ: Συνάρτηση `length`.** Σε μονοδιάστατο πίνακα, η `length` δίνει το πλήθος των στοιχείων του. Χρησιμοποιείται συχνά ως τελική τιμή στους βρόχους σάρωσης.

- Πληκτρολογήστε τα παρακάτω στο *Command Window* και παρατηρήστε τα αποτελέσματα:

Command Window	Αποτελέσματα
<pre>&gt;&gt; A = [8 7 -1 9 6]</pre>	<pre>A =      8     7    -1     9     6</pre>
<pre>&gt;&gt; n = length(A)</pre>	<pre>n =      5</pre>

- Ανοίξτε το `program82a.m` και αποθηκεύστε το ως `program82b.m`, αφού το τροποποιήσετε όπως παρακάτω ώστε να χρησιμοποιείται η `length` για την εύρεση του πλήθους των στοιχείων του πίνακα. Με τον τρόπο αυτό το πρόγραμμα γίνεται πιο γενικό, π.χ. αν αλλάζουμε τον πίνακα `A` προσθέτοντας ή αφαιρώντας στοιχεία, δεν χρειάζεται καμία άλλη αλλαγή για να λειτουργήσει σωστά το πρόγραμμα.

<code>program82b.m</code>	Αποτελέσματα
<pre>% Find mean value of a vector clc clear all</pre>	<pre>Mean value of A is 5.8 &gt;&gt;</pre>

```

% Matrix definition
A = [8 7 -1 9 6];

% Calculate mean value
s = 0;
for i=1:length(A)
    s = s + A(i);
end
s = s/length(A);

% Display mean value
fprintf('Mean value of A is %.1f\n', s);

```

**Εργασία 1.2ε: Δημιουργία νέου πίνακα με σταδιακή αύξηση των στοιχείων του.**

Στο MATLAB είναι δυνατή η δυναμική αύξηση των στοιχείων ενός πίνακα (σταδιακή αύξηση του μεγέθους του πίνακα).

- Πληκτρολογήστε το παρακάτω πρόγραμμα (program82c.m). Παρατηρήστε στα Αποτελέσματα τη σταδιακή αύξηση των στοιχείων του πίνακα κατά την επαναληπτική διαδικασία.

program82c.m	Αποτελέσματα
<pre> % Dynamic creation of a vector clc clear all  % Ορισμός πινάκων A = [ 8 7 -1 9 6]; B = [-5 0 1 3 -4];  % Άθροισμα πινάκων στοιχείο-προς-στοιχείο % και σταδιακή δημιουργία του C for i=1:length(A)     C(i) = A(i) + B(i) end </pre>	<pre> C =      3  C =      3     7  C =      3     7     0  C =      3     7     0    12  C =      3     7     0    12     2 </pre>

**Εργασία 1.2στ: Ορισμός διδιάστατου πίνακα και αναφορά στα στοιχεία του.** Τα στοιχεία μιας γραμμής χωρίζονται με κενό διάστημα ή κόμμα (,) και οι γραμμές χωρίζονται με ελληνικό ερωτηματικό (;) ή με αλλαγή γραμμής (↵Enter). Η αναφορά σε ένα στοιχείο ενός διδιάστατου πίνακα γίνεται με τους δείκτες του σε παρένθεση χωρισμένους με κόμμα (πρώτα ο δείκτης της γραμμής και μετά της στήλης).

- Πληκτρολογήστε τα παρακάτω στο Command Window και παρατηρήστε τα αποτελέσματα:

Command Window	Αποτελέσματα
<pre> &gt;&gt; A = [8 7 -1 9 6         7 6 4 0 6]  &gt;&gt; B = [1, 2, 3; 4, 5, 6]  &gt;&gt; A(2,3) </pre>	<pre> A =      8     7    -1     9     6      7     6     4     0     6  B =      1     2     3      4     5     6 </pre>

<pre>&gt;&gt; x = A(2,3) + B(1,2)^2</pre>	<pre>ans =      4  x =      8</pre>
---	-------------------------------------

**Εργασία 1.2ζ: Συναρτήσεις length και size σε δισδιάστατους πίνακες.** Η συνάρτηση size δίνει το πλήθος των στοιχείων κάθε διάστασης του πίνακα. Η συνάρτηση length στους δισδιάστατους πίνακες δίνει τη μεγαλύτερη διάστασή τους.

- Πληκτρολογήστε τα παρακάτω στο Command Window και παρατηρήστε τα αποτελέσματα:

Command Window	Αποτελέσματα
<pre>&gt;&gt; A = [8 7 -1 9 6; 7 6 4 0 6]</pre>	<pre>A =      8     7    -1     9     6      7     6     4     0     6</pre>
<pre>&gt;&gt; size(A)</pre>	<pre>ans =      2     5</pre>
<pre>&gt;&gt; size(A,1)</pre>	<pre>ans =      2</pre>
<pre>&gt;&gt; size(A,2)</pre>	<pre>ans =      5</pre>
<pre>&gt;&gt; n = size(A,1)*size(A,2)</pre>	<pre>n =     10</pre>
<pre>&gt;&gt; length(A)</pre>	<pre>ans =      5  &gt;&gt;</pre>

**Εργασία 1.2ζ: Σάρωση δισδιάστατου πίνακα.** Η σάρωση ενός δισδιάστατου πίνακα μπορεί να γίνει με διπλό επαναληπτικό βρόχο for.

- Πληκτρολογήστε το παρακάτω πρόγραμμα (program82d.m) στον Editor, στο οποίο σαρώνεται ένας δισδιάστατος πίνακας για τον υπολογισμό της μέσης τιμής των στοιχείων του:

program82d.m	Αποτελέσματα
<pre>% Find mean value of a matrix clc clear all  % Matrix definition A = [8 7 -1 9      7 6 4 0]</pre>	<pre>A =      8     7    -1     9      7     6     4     0      2     0    -3     5  Mean: 3.67  &gt;&gt;</pre>

```

2 0 -3 5]

% Find mean values
s = 0;
for i=1:size(A,1)
    for j=1:size(A,2)
        s = s + A(i,j);
    end
end
end
s = s/(size(A,1)*size(A,2));

% Display mean value
fprintf('Mean: %.2f\n', s);

```

**Εργασία 1.2η: Προσπέλαση συγκεκριμένων γραμμών ή στηλών διδιάστατου πίνακα.** Αν πρέπει να προσπελαστούν τα στοιχεία μιας συγκεκριμένης γραμμής ή στήλης ενός πίνακα, τότε κρατάμε σταθερό το δείκτη της γραμμής ή της στήλης:

- Πληκτρολογήστε το παρακάτω πρόγραμμα (program82e.m) στον Editor, στο οποίο υπολογίζεται η μέση τιμή της 3ης γραμμής ενός διδιάστατου πίνακα:

program82e.m	Αποτελέσματα
<pre> % Find mean value of a row clc clear all  % Matrix definition A = [8 7 -1 9       7 6 4 0       2 0 -3 5]  % Find mean value of 3rd row s = 0; for j=1:size(A,2)     s = s + A(3,j); end s = s/size(A,2);  % Display mean value of 3rd row fprintf('Mean value of 3rd row: %.2f\n',s); </pre>	<pre> A =      8     7    -1     9      7     6     4     0      2     0    -3     5  Mean value of 3rd row: 1.00 &gt;&gt; </pre>

## Γραφικές παραστάσεις (1ο μέρος)

Σε αυτήν την ενότητα θα εξοικειωθείτε με τον τρόπο απεικόνισης γραφικών παραστάσεων στο MATLAB, και συγκεκριμένα με τις εντολές:

- plot
- close
- hold
- grid
- xlabel, ylabel
- legend
- title

Σημείωση: Σε αυτήν την ενότητα η εντολή `plot` θα χρησιμοποιηθεί με στοιχειώδη τρόπο, δηλαδή με χρήση απλών μεταβλητών, χωρίς πίνακες.

### 1.1 Συνοπτική θεωρία

Για την απεικόνιση γραφικών παραστάσεων και εικόνων το MATLAB διαθέτει ειδικά παράθυρα γραφικών που λέγονται *figures*.

plot: Γραφική παράσταση δύο διαστάσεων.

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

```
plot(x, y, 'χρώμα & είδος σημείου'), π.χ.
```

```
plot(x, y, 'b+') , plot(x, y, 'ro')
```

όπου  $x, y$  = συντεταγμένες του σημείου που θα εκτυπωθεί

**Σύμβολα χρώματος:** r g b y c(cyan) m(magenta)  
k(black) w(white)

**Σύμβολα είδους σημείου:** + o \* . x s(square)  
d(diamond) p(pentagone) h(hexagone) ^  
v > <

close = Κλείσιμο των παραθύρων γραφικών.

close = Κλείνει το τρέχον figure

close(n) = Κλείνει το figure με αριθμό n

`close all` = Κλείνει όλα τα ανοιχτά *figures*

`figure` = Δημιουργία ή ενεργοποίηση παραθύρου γραφικών.

`figure` = Δημιουργία νέου *figure*

`figure(n)` = Δημιουργία νέου *figure* με αριθμό *n* ή ενεργοποίηση του *figure* με αριθμό *n*

`hold`: Επιτρέπει ή αποτρέπει την εκτύπωση περισσότερων γραφημάτων στο ίδιο *figure*.

`hold on` = Η γραφική παράσταση παραμένει, και απεικονίζεται και η επόμενη στο ίδιο *figure*

`hold off` = Η τρέχουσα γραφική παράσταση σβήνει, και η επόμενη απεικονίζεται μόνη της

`grid`: Πλέγμα αξόνων.

`grid on` = Δημιουργία πλέγματος αξόνων

`grid off` = Σβήσιμο πλέγματος αξόνων

`xlabel`, `ylabel`: Τίτλοι αξόνων *x* και *y*.

`xlabel('κείμενο')`

`xlabel('κείμενο')`

`title`: Τίτλος γραφήματος.

`title('κείμενο')`

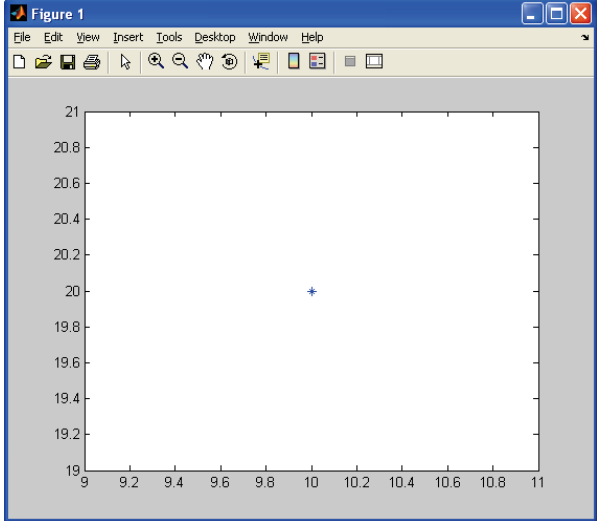
`legend`: Υπόμνημα.

`legend('κείμενο 1', 'κείμενο 2', 'κείμενο 3', ...)`

## 1.2 Εργασίες κατανόησης θεωρίας

**Εργασία 1.2α:** Εντολή plot: Η εντολή `plot` απεικονίζει ένα σημείο σε ένα παράθυρο γραφικών (*figure*). Αν δεν υπάρχει ανοιχτό *figure*, η `plot` δημιουργεί ένα.

- Εκτελέστε την παρακάτω εντολή στο *Command Window* και παρατηρήστε το αποτέλεσμα:

Command Window	Αποτελέσματα
<pre>plot(10,20, 'b*');</pre>	

**Εργασία 1.2β:** Εντολή close: Η εντολή `close` κλείνει το τρέχον *figure*. Η `close all` κλείνει όλα τα ανοιχτά *figures*.

- Εκτελέστε την παρακάτω εντολή στο *Command Window* και παρατηρήστε το αποτέλεσμα:

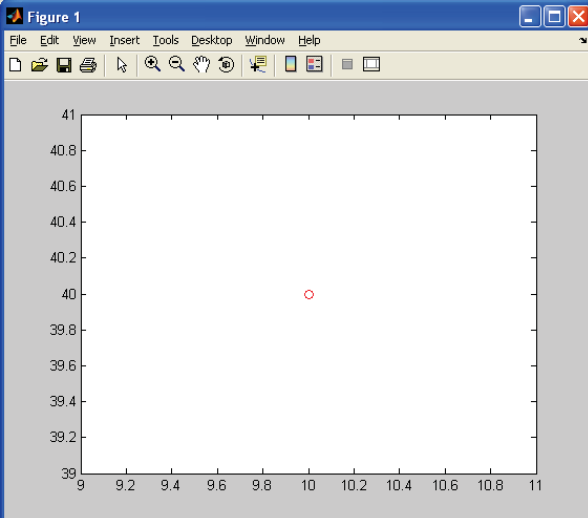
Command Window	Αποτελέσματα
<pre>close;</pre>	Παρατηρούμε ότι κλείνει το <i>figure</i> που είχε ανοίξει με την προηγούμενη εργασία (1.2α).

*Σημείωση:* Όταν εργαζόμαστε με *figures*, συνήθως βάζουμε την εντολή `close all` στην αρχή του προγράμματος (μαζί με τις `clc` και `clear`) ώστε να κλείσουν όλα τα προηγούμενα *figures*.

**Εργασία 1.2γ: Εντολή hold:** Όταν δοθεί μια νέα εντολή plot, τότε η προηγούμενη γραφική παράσταση σβήνεται και απεικονίζεται μόνο η νέα. Όμως, αν χρησιμοποιηθεί η εντολή hold on, τότε η νέα γραφικά παράσταση τυπώνεται μαζί με την παλιά.

- Εκτελέστε τις παρακάτω εντολές στο Command Window και παρατηρήστε τα αποτελέσματα:

- (i) Χωρίς την εντολή hold on (ή με την εντολή hold off) η επόμενη plot σβήνει το αποτέλεσμα της προηγούμενης:

Command Window	Αποτελέσματα
<pre>close all;  plot(10,20,'b*'); plot(10,30,'gs'); plot(10,40,'ro');</pre>	

Σημείωση: Σε κάθε μία από τις προηγούμενες plot, παρατηρήστε ότι το αποτέλεσμα της προηγούμενης έχει σβήσει.

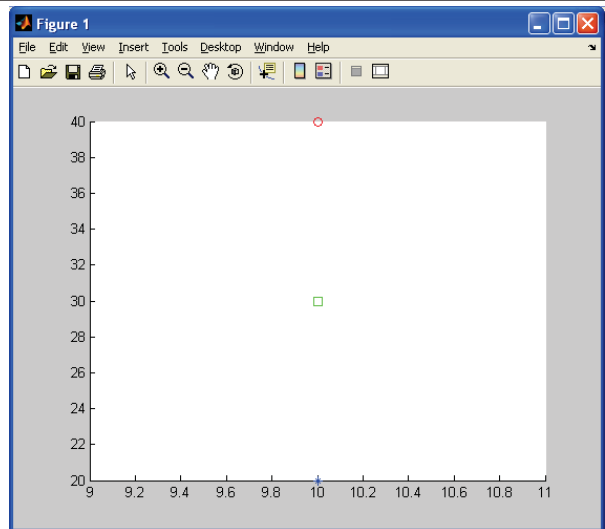


- (ii) Με την εντολή `hold on`, η επόμενη `plot` απεικονίζεται μαζί με την προηγούμενη:

### Command Window

```
close all;  
  
hold on;  
plot(10,20,'b*');  
plot(10,30,'gs');  
plot(10,40,'ro');
```

### Αποτελέσματα



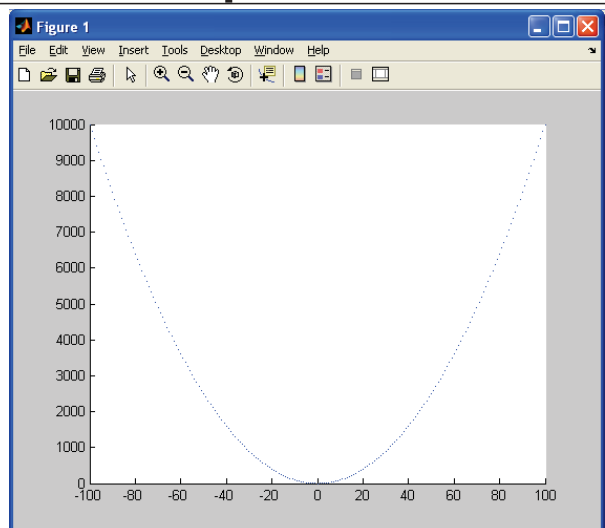
Εργασία 1.2δ: Γραφική παράσταση συνάρτησης: Η εντολή `plot` μπορεί να χρησιμοποιηθεί μέσα σε ένα βρόχο `for`, ώστε να απεικονιστεί η γραφική παράσταση μιας μαθηματικής συνάρτησης.

- Γράψτε και εκτελέστε το παρακάτω πρόγραμμα (`program_1a.m`), που απεικονίζει τη γραφική παράσταση μιας παραβολής.

### Program\_1a.m

```
clc;  
clear all;  
close all;  
  
hold on;  
for x=-100:100  
    y = x^2;  
    plot(x,y);  
end
```

### Αποτελέσματα



- Γράψτε και εκτελέστε το παρακάτω πρόγραμμα (program1\_b.m), που απεικονίζει τη γραφική παράσταση της συνάρτησης του ημιτόνου.

Θα χρησιμοποιηθούν και οι παρακάτω εντολές:

- grid: Δημιουργία πλέγματος στο διάγραμμα.
- xlabel, ylabel: Δημιουργία τίτλων στους άξονες.
- title: Δημιουργία τίτλου στο διάγραμμα.

### program1\_b.m

```
clc;
clear all;
close all;

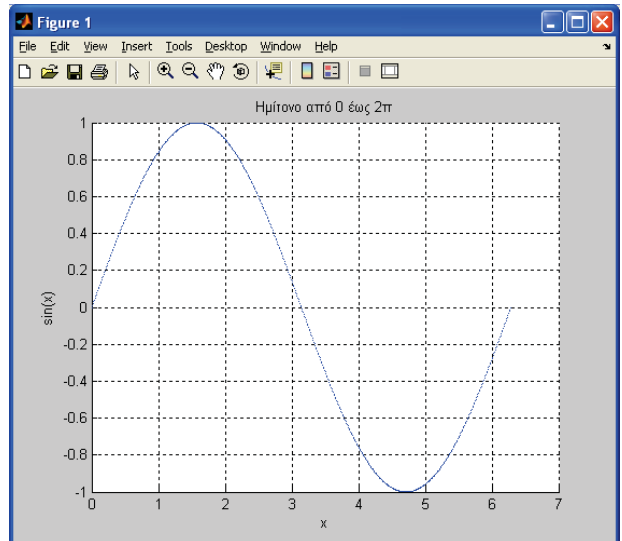
hold on;
% Γραφική παράσταση
ημιτόνου
% με πεδίο ορισμού από
0 έως 2π
for x=0:0.01:2*pi
    y = sin(x);
    plot(x,y);
end

% Δημιουργία πλέγματος
grid on;

% Τίτλοι αξόνων
xlabel('x');
ylabel('sin(x)');

% Τίτλος γραφήματος
title('Ημίτονο από 0
έως 2π');
```

### Αποτελέσματα



- Γράψτε και εκτελέστε το παρακάτω πρόγραμμα (program1\_c.m), που απεικονίζει τη γραφική παράσταση αθροίσματος δύο τριγωνομετρικών συναρτήσεων.

Θα χρησιμοποιηθεί και οι παρακάτω εντολή:

- legend: Δημιουργία υπομνήματος.

### program1\_c.m

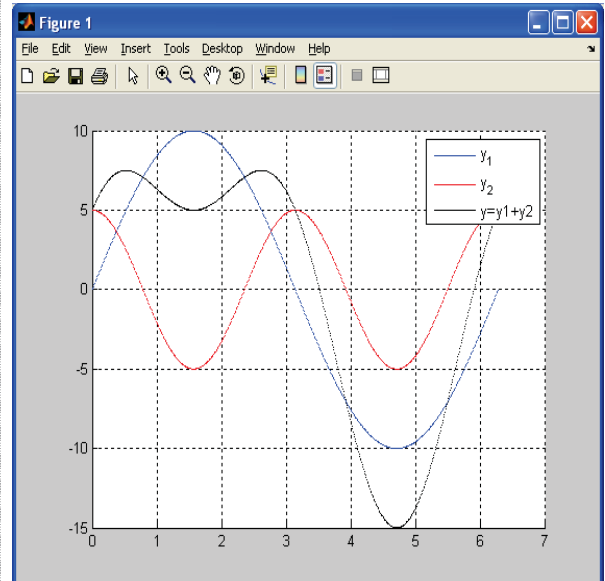
```
clc;
clear all;
close all;

hold on;
for x=0:0.01:2*pi
    y1 = 10*sin(x);
    y2 = 5*cos(2*x);
    y = y1 + y2;
    plot(x,y1,'b');
    plot(x,y2,'r');
    plot(x,y,'k');
end

% Δημιουργία πλέγματος
grid on;

% Υπόμνημα
legend('y_1', 'y_2', 'y=y1+y2');
```

### Αποτελέσματα



### Εργασία 1.2ε: Ένα πρόβλημα φυσικής:

Ένα σώμα εκτελεί οριζόντια βολή από ύψος 100m με αρχική ταχύτητα οριζόντιας κατεύθυνσης και μέτρου 5m/sec.

α) Να γίνει πρόγραμμα σε MATLAB όπου να απεικονίζεται γραφικά η θέση του σώματος (δηλαδή οι συντεταγμένες του) για χρόνο από 0 έως 5s.

β) Μπορείτε, από τη γραφική παράσταση που προκύπτει, να βρείτε κατά προσέγγιση σε πόση οριζόντια απόσταση από το σημείο που εκτοξεύσαμε το σώμα θα φτάσει στο έδαφος (βεληνεκές);

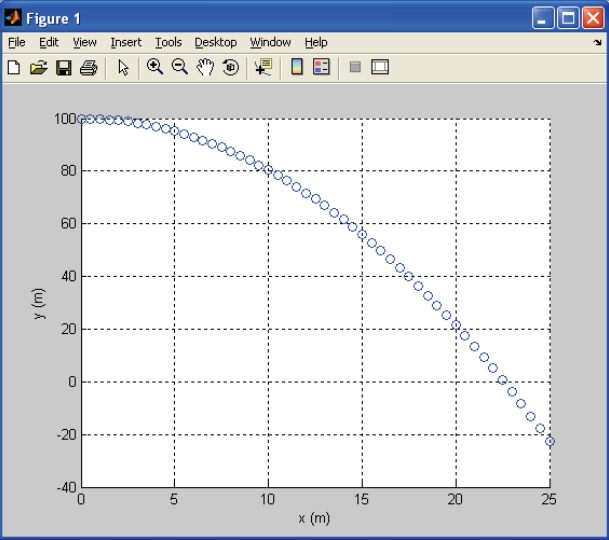
Δίνεται ότι οι συντεταγμένες ενός σώματος που εκτελεί οριζόντια βολή από ύψος  $H$  σε συνάρτηση με το χρόνο ( $t$ ) είναι (αν θεωρήσουμε αμελητέα την αντίσταση του αέρα, και αν θέσουμε την αρχή των αξόνων στην επιφάνεια του εδάφους.):

$$x = v_0 \cdot t$$
$$y = H - \frac{1}{2} g t^2$$

όπου  $v_0$  η αρχική ταχύτητα του σώματος και  $g=9.81\text{m/s}^2$  η επιτάχυνση της βαρύτητας.

Λύση:

α) Γράψτε και εκτελέστε το παρακάτω πρόγραμμα (program1\_d.m):

program1_d.m	Αποτελέσματα
<pre>clc;clear;close all;  H = 100; u0 = 5; g = 9.81;  hold on; for t=0:0.1:5     x = u0*t;     y = H - (1/2)*g*t^2;     plot(x,y,'bo'); end grid on; xlabel('x (m)'); ylabel('y (m)');</pre>	

β) Όταν το σώμα φτάσει στο έδαφος, τότε  $y=0$ . Από τη γραφική παράσταση βλέπουμε ότι η συντεταγμένη  $x$  του σώματος για  $y=0$  είναι κατά προσέγγιση ίση με 22 ή 23. Κατά συνέπεια, μπορούμε να πούμε ότι το βεληνεκές της βολής είναι περίπου 22.5m.

**Σημείωση 1:** Για να επιβεβαιώσετε αυτό το προσεγγιστικό αποτέλεσμα, μπορείτε να δώσετε στο Command Window την εντολή

`plot(22.5, 0, 'k*');` ώστε να απεικονιστεί ένα μαύρο αστεράκι στη θέση με  $x=22.5$  και  $y=0$ .

**Σημείωση 2:** Αναλυτική εύρεση του χρόνου πτώσης: Λύνουμε την εξίσωση  $y = H - \frac{1}{2}gt^2$  ως προς  $t$  αφού θέσουμε  $y=0$ , ώστε να βρούμε το χρόνο που κάνει το σώμα να φθάσει στο έδαφος:  
 $0 = H - \frac{1}{2}gt^2 \Rightarrow t = \sqrt{\frac{2H}{g}}$ .

Δίνουμε στο *Command Window* την εντολή  $t = (2 * H / g) ^ 0.5$  και βρίσκουμε  $t = 4.5152$ .

Από την εξίσωση  $x = v_0 * t$  (και την εντολή  $x = u_0 * t$  στο *Command Window*) βρίσκουμε  $x = 22.5762$ , δηλαδή η προσέγγισή μας ήταν πολύ καλή.

## Γραφικές παραστάσεις (2ο μέρος)

Σε αυτήν την ενότητα θα εξοικειωθείτε με τον τρόπο απεικόνισης γραφικών παραστάσεων στο MATLAB χρησιμοποιώντας την εντολή `plot` με πίνακες.

Επίσης, θα δείτε επιπλέον εντολές σχετικές με γραφικές παραστάσεις, και συγκεκριμένα τις:

- `figure`
- `axis`
- `subplot`
- `text`

### 1.1 Συνοπτική θεωρία

Στην προηγούμενη ενότητα είδαμε ότι στην εντολή `plot(x, y)` οι μεταβλητές  $x$  και  $y$  ήταν απλές μεταβλητές (μία τιμή του  $x$  και μία τιμή του  $y$ ), οπότε απεικονίζεται ένα μόνο σημείο. Στην περίπτωση αυτή, αν θέλουμε να απεικονίσουμε πολλά σημεία, όπως σε μία γραφική παράσταση, η `plot` πρέπει να βρίσκεται μέσα σε ένα επαναληπτικό βρόχο (π.χ. `for`).

Όμως, η σύνταξη της εντολής `plot` επιτρέπει τα  $x$  και  $y$  να είναι μονοδιάστατοι πίνακες, που περιέχουν τις αντίστοιχες συντεταγμένες όλων των σημείων της γραφικής παράστασης. Στην περίπτωση αυτή, η `plot` δεν είναι ανάγκη να βρίσκεται μέσα σε επαναληπτικό βρόχο, διότι απεικονίζονται όλα τα σημεία ταυτόχρονα, επιταχύνοντας έτσι την εκτύπωση. Επίσης, στην περίπτωση αυτή η `plot` συνδέει τα σημεία με μια γραμμή, κάνοντας πιο ρεαλιστικό το αποτέλεσμα της γραφικής παράστασης.

plot: Γραφική παράσταση δύο διαστάσεων.

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

```
plot(x, y, 'χρώμα και είδος σημείων και γραμμής')
```

$x, y$  = μονοδιάστατοι πίνακες με τις συντεταγμένες των σημείων που θα απεικονιστούν

Σύμβολα χρώματος: `r g b y c m k w`

Σύμβολα είδους σημείων: `+ o * . x s d p h ^ v >`  
<

Σύμβολα είδους γραμμής: - : -- -.

figure: Η εντολή `figure` δημιουργεί ένα νέο `figure` ή κάνει ενεργό ένα υπάρχον.

`figure` = Δημιουργία ενός νέου παραθύρου γραφικών.

`figure(n)` = Γίνεται ενεργό το παράθυρο γραφικών με αριθμό `n` («ενεργό» `figure` σημαίνει ότι οι εντολές απεικόνισης θα εφαρμοστούν σε αυτό).

axis = Χειροκίνητος καθορισμός ορίων αξόνων.

`axis([xmin xmax ymin ymax])` =

Όρια άξονα  $x$ : από `xmin` έως `xmax`

Όρια άξονα  $y$ : από `ymin` έως `ymax`

subplot: Η εντολή `subplot` επιτρέπει την απεικόνιση πολλών διαγραμμάτων στο ίδιο `figure` αλλά σε διαφορετικούς άξονες.

`subplot(m, n, p)`: Δημιουργία  $m \times n$  θέσεων απεικόνισης μέσα σε ένα `figure` (με `m` γραμμές και `n` στήλες) και απεικόνιση στη θέση `p` (μετρώντας από αριστερά προς τα δεξιά και στη συνέχεια από πάνω προς τα κάτω).

text: Απεικόνιση μηνύματος μέσα σε ένα γράφημα.

`text(x, y, 'κείμενο')` = Απεικόνιση του 'κειμένου' στις συντεταγμένες `x, y` του γραφήματος.

## 1.2 Εργασίες κατανόησης θεωρίας

Εργασία 1.2α: Εντολή plot (με τις συντεταγμένες των σημείων σε μονοδιάστατους πίνακες):

- Δημιουργήστε και εκτελέστε το παρακάτω πρόγραμμα (program2\_a.m):

### program2\_a.m

```
clc;
clear all;
close all;

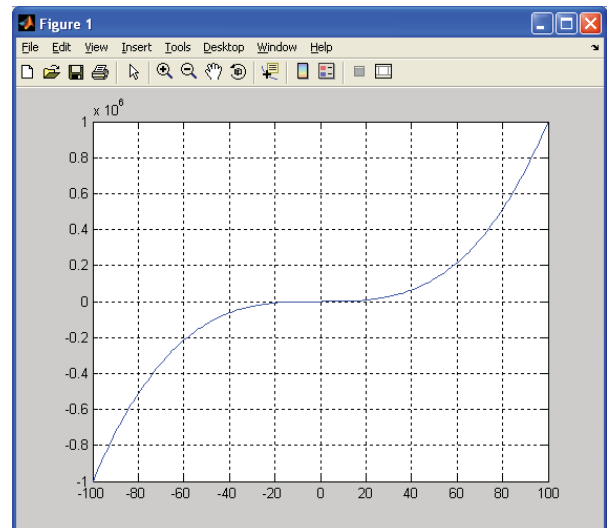
% Δημιουργία πίνακα με
τιμές του x
x = [-100:100];

% Δημιουργία πίνακα με
τιμές του y
for i=1:length(x)
    y(i) = x(i)^3;
end

% Γραφική παράσταση του
y ως προς x
plot(x,y);

grid on;
```

### Αποτελέσματα



Εργασία 1.2β: Εντολές axis και text:

- `axis([xmin xmax ymin ymax])` = Χειροκίνητος καθορισμός ορίων αξόνων.
- `text(x,y,'κείμενο')` = Απεικόνιση μηνύματος στις συντεταγμένες x, y.



- Δημιουργήστε και εκτελέστε το παρακάτω πρόγραμμα (program2\_b.m):

### program2\_b.m

```
clc;
clear all;
close all;

% Πεδίο ορισμού x από -2π
έως 2π
x = [-2*pi:0.01:2*pi];

for i=1:length(x)
    y(i) = sin(x(i));
end

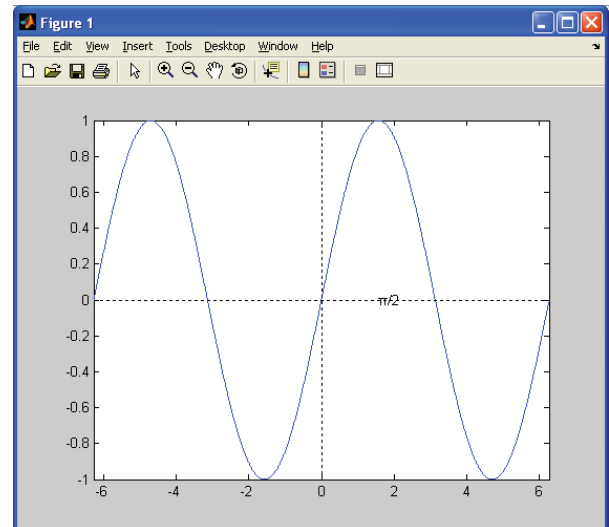
% Γραφική παράσταση
ημιτόνου
plot(x,y);

% Όρια αξόνων:
% xx' από -2π έως 2π
% yy' από -1 έως +1
axis([-2*pi 2*pi -1 1]);

% Δημιουργία αξόνων με
διακεκομμένες γραμμές
hold on;
plot([-2*pi 2*pi],[0
0],'k:');
plot([0 0],[-1 1],'k:');

% Κείμενο στο γράφημα
text(pi/2,0,'π/2');
```

### Αποτελέσματα



**Εργασία 1.2γ: Εντολή subplot:** Η εντολή subplot επιτρέπει την απεικόνιση πολλών διαγραμμάτων στο ίδιο figure αλλά σε διαφορετικούς άξονες.

- subplot(m,n,p): Δημιουργία m×n θέσεων απεικόνισης μέσα σε ένα figure και απεικόνιση στη θέση p (μετρώντας από αριστερά προς τα δεξιά και στη συνέχεια από πάνω προς τα κάτω).
- Δημιουργήστε και εκτελέστε το παρακάτω πρόγραμμα (program1\_c.m):

### program1\_c.m

```
clc;
clear all;
close all;

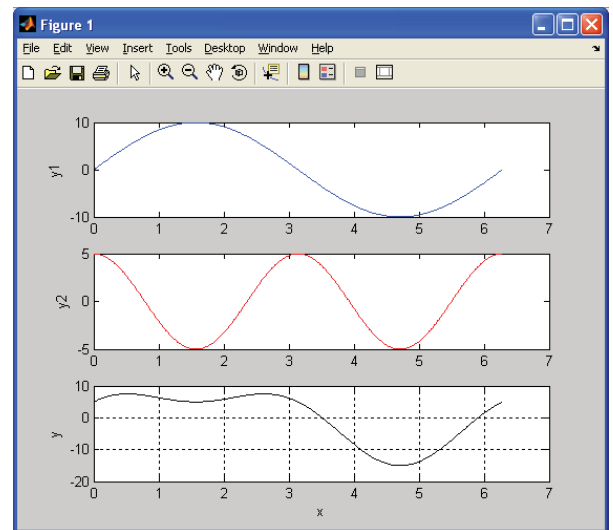
% Πεδίο ορισμού x από 0
% έως 2π
x = [0:0.01:2*pi];

% Δημιουργία πινάκων με
% τις τιμές
% των y1, y2 και του
y=y1+y2
for i=1:length(x)
    y1(i) = 10*sin(x(i));
    y2(i) = 5*cos(2*x(i));
    y(i) = y1(i) + y2(i);
end

% Δημιουργία 1ου υπο-
% γραφήματος
subplot(3,1,1);
% Απεικόνιση του y1 στο
% υπο-γράφημα
plot(x,y1,'b');
ylabel('y1');

% Δημιουργία 2ου υπο-
% γραφήματος
subplot(3,1,2);
```

### Αποτελέσματα



```
% Απεικόνιση του y2 στο
υπο-γράφημα
plot(x,y2,'r');
ylabel('y2');

% Δημιουργία 3ου υπο-
γραφήματος
subplot(3,1,3);
% Απεικόνιση του y στο
υπο-γράφημα
plot(x,y,'k');
xlabel('x');
ylabel('y');
grid on;
```

Σε αυτήν την ενότητα θα εξοικειωθείτε με:

- τις λογικές μεταβλητές
- τις συγκριτικές πράξεις, τις λογικές πράξεις και τους τελεστές τους.
- την εντολή διακλάδωσης if.

### Συνοπτική θεωρία

Λογικές μεταβλητές: Μεταβλητές όπου αποθηκεύονται οι λογικές τιμές true ή false («αλήθεια» ή «ψεύδος»), που αντιστοιχούν στις τιμές 1 ή 0 αντίστοιχα, με μέγεθος 1 byte.

Συγκριτικές πράξεις: Σύγκριση δύο αριθμητικών μεταβλητών με έξοδο μια λογική τιμή (1/0):

Συγκριτική πράξη	Τελεστής	Αποτέλεσμα	Παράδειγμα	Αποτέλεσμα
Σύγκριση ισότητας	==	true ή false	5==9	0
Σύγκριση ανισότητας	!=	true ή false	5!=9	1
Σύγκριση για μεγαλύτερο	>	true ή false	5>9	0
Σύγκριση για μικρότερο	<	true ή false	5<9	1
Σύγκριση για μεγαλύτερο ή ίσο	>=	true ή false	5>=9	0
Σύγκριση για μικρότερο ή ίσο	<=	true ή false	5<=9	1

Λογικές πράξεις: «Πράξεις» μεταξύ λογικών μεταβλητών με έξοδο μια λογική τιμή (1/0). Οι λογικές πράξεις μπορούν να συνδέσουν απλές συγκριτικές πράξεις, με αποτέλεσμα να προκύπτουν σύνθετες συγκριτικές πράξεις:

Λογική πράξη	Τελεστής	Αποτέλεσμα	Παράδειγμα	Αποτέλεσμα
AND	&	true (1) αν και οι δύο είσοδοι είναι true (1)	5==9 & 5<9	0
OR		true (1) αν τουλάχιστον μια είσοδος είναι true (1)	5==9   5<9	1
NOT	~	true (1) αν η είσοδος είναι false (0) και αντιστρόφως	~(5<9)	0

Διακλάδωση if: Διακλάδωση της ροής του προγράμματος μέσω συνθήκης.

```
if <συνθήκη>
  <εντολές>
end
```

Εναλλακτική διακλάδωση: if - else

```
if <συνθήκη>
  <εντολές 1>
else
  <εντολές 2>
end
```

Πολλαπλή διακλάδωση: if - elseif - else

```
if <συνθήκη 1>
  <εντολές 1>
elseif <συνθήκη 2>
  <εντολές 2>
elseif <συνθήκη 3>
  <εντολές 3>
...
else
  <εντολές>
end
```

Προσοχή: Αν αληθεύουν περισσότερες από μία συνθήκες, εκτελούνται μόνο οι εντολές της πρώτης από αυτές.

## Εργασίες κατανόησης θεωρίας

Λογικές μεταβλητές, λογικές πράξεις, συγκριτικές πράξεις. Θα δημιουργήσουμε μερικές λογικές μεταβλητές (i) απ' ευθείας και (ii) μέσω λογικών και συγκριτικών πράξεων:

Command Window	Αποτελέσματα																												
i) Δημιουργία λογικών μεταβλητών (απ' ευθείας):  >> a = true  >> b = false	a = 1 b = 0																												
ii) Δημιουργία λογικών μεταβλητών μέσω συγκριτικών και λογικών πράξεων:  >> c = 5>9  >> ~a  >> 5~=9  >> d = a & b  >> c   ans	c = 0 ans = 0 ans = 1 d = 0 ans = 1																												
iii) Επισκόπηση των λογικών μεταβλητών που δημιουργήθηκαν:  >> x = 1  >> whos	x = 1  <table border="1"> <thead> <tr> <th>Name</th> <th>Size</th> <th>Bytes</th> <th>Class</th> </tr> </thead> <tbody> <tr> <td>a</td> <td>1x1</td> <td>1</td> <td>logical</td> </tr> <tr> <td>ans</td> <td>1x1</td> <td>1</td> <td>logical</td> </tr> <tr> <td>b</td> <td>1x1</td> <td>1</td> <td>logical</td> </tr> <tr> <td>c</td> <td>1x1</td> <td>1</td> <td>logical</td> </tr> <tr> <td>d</td> <td>1x1</td> <td>1</td> <td>logical</td> </tr> <tr> <td>x</td> <td>1x1</td> <td>8</td> <td>double</td> </tr> </tbody> </table>	Name	Size	Bytes	Class	a	1x1	1	logical	ans	1x1	1	logical	b	1x1	1	logical	c	1x1	1	logical	d	1x1	1	logical	x	1x1	8	double
Name	Size	Bytes	Class																										
a	1x1	1	logical																										
ans	1x1	1	logical																										
b	1x1	1	logical																										
c	1x1	1	logical																										
d	1x1	1	logical																										
x	1x1	8	double																										

Σημείωση: Παρατηρείστε το μέγεθος των λογικών μεταβλητών στη μνήμη (1 Byte), σε αντίθεση με τη μεταβλητή x, που είναι αριθμητική μεταβλητή τύπου double και καταλαμβάνει 8 Bytes (η x δημιουργήθηκε μόνο για να δείτε τη διαφορά της με την a που, φαινομενικά, έχει ίδια τιμή).

Σύνθετες συγκριτικές πράξεις. Έστω x ο βαθμός ενός σπουδαστή. Δημιουργήστε  
 i) μια λογική μεταβλητή (a) που να είναι αληθής αν ο σπουδαστής πάρει βαθμό «λίαν καλώς» (δηλαδή αν ο βαθμός x είναι από 6.50 έως 8.49), και  
 ii) μία λογική μεταβλητή (b) που να είναι αληθής αν ο σπουδαστής δεν πάρει «λίαν καλώς» (δηλαδή αν ο βαθμός x βρίσκεται έξω από το διάστημα 6.50 έως 8.49).

Command Window	Αποτελέσματα
>> x = 7; >> a = (x>=6.50 & x<=8.49)	a = 1
>> x = 9; >> b = (x<6.50   x>8.49)	b = 1

Σημείωση: Οι παρενθέσεις δεν είναι υποχρεωτικές και μπήκαν μόνο για καλύτερη κατανόηση.

Εντολή if. Να γραφτεί πρόγραμμα (program a.m ) στο οποίο να επιλέγεται η πρωτοβάθμια εξίσωση. Στο πρόγραμμα να γίνεται έλεγχος για την περίπτωση του να είναι ο συντελεστής του αγνώστου ίσος με το μηδέν, οπότε και να απεικονίζεται κατάλληλο μήνυμα:

program a.m	Αποτελέσματα (εξαρτώνται από τα στοιχεία που θα εισάγετε)
<pre> clc; clear all;  disp('Epilysi tis ax+b=0');  a = input('a = '); b = input('b = ');  if a~=0     x = -b/a;     fprintf('x = %f\n', x); end  if a==0     disp('Den yparxei lysi. '); end         </pre>	<p>1<sup>ο</sup> παράδειγμα εκτέλεσης του προγράμματος:</p> <pre> Epilysi tis ax+b=0 a = 4 b = 6 x = -1.500000 &gt;&gt;         </pre> <p>2<sup>ο</sup> παράδειγμα εκτέλεσης του προγράμματος:</p> <pre> Epilysi tis ax+b=0 a = 0 b = 6 Den yparxei lysi. &gt;&gt;         </pre>

Εντολή if-else. Να γραφτεί πρόγραμμα (program b.m ) στο οποίο να επιλέγεται η δευτεροβάθμια εξίσωση. Στο πρόγραμμα να γίνεται έλεγχος για την περίπτωση που προκύπτει αρνητική διακρίνουσα, οπότε και να απεικονίζεται κατάλληλο μήνυμα:

program b.m	Αποτελέσματα
<pre> clc; clear all;  disp('Epilysi tis ax^2+bx+c=0');  a = input('a = '); b = input('b = '); c = input('c = ');  D = b^2 - 4*a*c;  if D&gt;=0     x1 = (-b + D^0.5)/(2*a);     x2 = (-b - D^0.5)/(2*a);     fprintf('x1 = %f\nx2 = %f\n', x1, x2); else     disp('Arnitiki diakrinousa'); end         </pre>	<p>1<sup>ο</sup> παράδειγμα εκτέλεσης του προγράμματος:</p> <pre> Epilysi tis ax^2+bx+c=0 a = 1 b = 2 c = 3 Arnitiki diakrinousa &gt;&gt;         </pre> <p>2<sup>ο</sup> παράδειγμα εκτέλεσης του προγράμματος:</p> <pre> Epilysi tis ax^2+bx+c=0 a = 1 b = -5 c = 6 x1 = 3.000000 x2 = 2.000000 &gt;&gt;         </pre>

Εντολή if-elseif-else. Να γραφτεί πρόγραμμα (programc.m) στο οποίο να υπολογίζεται ο Δείκτης Σωματικού Βάρους (BMI) ενός ασθενούς και να απεικονίζεται μήνυμα ανάλογο με την κατηγορία του

programc.m	Αποτελέσματα (εξαρτώνται από τα στοιχεία που θα εισάγετε)
<pre> clc; clear all;  disp('BMI calculation');  w = input('Weight (kg) ? '); h = input('Height (m) ? '); bmi = w / h^2;  fprintf('BMI = %.1f (' ,bmi);  if bmi&lt;18.5     fprintf('Underweight'); elseif bmi&gt;=18.5 &amp; bmi&lt;25     fprintf('Normal weight'); elseif bmi&gt;=25 &amp; bmi&lt;30     fprintf('Overweight'); else     fprintf('Obesity'); end  fprintf('\n');</pre>	<pre> BMI calculation Weight (kg) ? 68 Height (m) ? 1.72 BMI = 23.0 (Normal weight) &gt;&gt;</pre>

Σημείωση: Ο Δείκτης Σωματικού Βάρους (Body Mass Index, BMI) ενός ατόμου ισούται με το πηλίκο του βάρους (σε kg) προς το τετράγωνο του ύψους (σε m):

$$BMI = \text{Βάρος}(kg) / \text{Υψος}(m)^2$$

Ο BMI θεωρείται ένας απλός αλλά αρκετά αξιόπιστος δείκτης για την εκτίμηση της παχυσαρκίας ενήλικων. Οι κατηγορίες αναλόγως του BMI είναι (<http://www.nhlbisupport.com/bmi/>):

- Λιπόβαρος:  $BMI < 18.5$
- Κανονικό βάρος:  $18.5 \leq BMI < 25$
- Υπέρβαρος:  $25 \leq BMI < 30$
- Παχύσαρκος:  $BMI \geq 30$

## Εντολή επανάληψης while

### Σε αυτήν την ενότητα:

- θα εξοικειωθείτε με την εντολή επανάληψης while και
- θα αναπτυχθούν τρεις βασικές εφαρμογές της:
  - Έλεγχος εγκυρότητας εισαγωγής από το πληκτρολόγιο
  - Συνεχής εισαγωγή από το πληκτρολόγιο
  - Συνεχής λειτουργία μενού επιλογών

### Συνοπτική θεωρία

Εντολή while: Επανάληψη ομάδας εντολών υπό συνθήκη.

```
while <συνθήκη>  
    <εντολές>  
end
```

Αρχικά ελέγχεται η συνθήκη και αν είναι αληθής εκτελούνται οι εντολές. Στη συνέχεια επανελέγχεται η συνθήκη, και όσο παραμένει αληθής οι εντολές εκτελούνται επαναληπτικά.

Διαφορά for από while: Η for και η while είναι εντολές επανάληψης. Η for χρησιμοποιείται μόνο όταν είναι γνωστός ο αριθμός των επαναλήψεων, ενώ στη while ο βρόχος επαναλαμβάνεται συνεχώς για όσο ικανοποιείται μια συνθήκη.

Μετατροπή for σε while: Η while είναι γενικότερη από τη for, οπότε ένας επαναληπτικός βρόχος for μπορεί να μετατραπεί σε while, όμως το αντίστροφο δεν είναι πάντα δυνατόν (βλέπε m files: Synoptiki\_theoria\_1.m, Synoptiki\_theoria\_2.m).

### Εργασίες κατανόησης θεωρίας

Εργασία α: Εντολή while: Ένα απλό παράδειγμα.

- Γράψτε και εκτελέστε το παρακάτω πρόγραμμα (**program\_a.m**):

<b>program_a.m</b>	<b>Αποτελέσματα</b>
clc; clear all;	5
n = 5;	4
while n>0	3
disp(n);	2
n = n - 1; pause(0.5)	1
end	Boom!
disp ('    Boom!');	

Εργασία β: Εντολή while. Ένα μαθηματικό παράδειγμα.

- Το άπειρο άθροισμα  $S = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} \dots$  τείνει στο 1. Να κατασκευάσετε πρόγραμμα (**program\_b.m**) που να βρίσκει το παραπάνω άθροισμα, μέχρι το αποτέλεσμα να υπερβεί το 0.999. Να απεικονιστεί το αποτέλεσμα καθώς και ο αριθμός των επαναλήψεων που απαιτήθηκαν:



## program\_b.m

```
clc;
clear all;

s = 0; p = 1;

while s<=0.999
    p = 2*p;
    s = s + 1/p, pause(0.5)
end
```

## Αποτελέσματα

```
s =
    0.5000

s =
    0.7500

s =
    0.8750

s =
    0.9375

s =
    0.9688

s =
    0.9844

s =
    0.9922

s =
    0.9961

s =
    0.9980

s =
    0.9990
```

***Εργασία γ:*** Εφαρμογή while. ***Εγκυρότητα εισαγωγής από το πληκτρολόγιο:*** Η while μπορεί να χρησιμοποιηθεί για ***έλεγχο εγκυρότητας*** των τιμών που δίνουμε από το πληκτρολόγιο:

- Γράψτε και εκτελέστε το παρακάτω πρόγραμμα (**program\_c.m**):

<b>program_c.m</b>	<b>Αποτελέσματα</b>
<pre> clc; clear all;  N = 10; % Εισαγωγή 10 τιμών s = 0;  for i=1:N     x = input('Βαθμός? ');      while (x&lt;0    x&gt;10)         disp('Σφάλμα: Βαθμός 0 έως 10');         x = input('Βαθμός? ');     end      s = s + x; end fprintf('Μέσος όρος βαθμών: %.1f\n',s/N); </pre>	<pre> Βαθμός? 1 Βαθμός? 2 Βαθμός? 3 Βαθμός? 4 Βαθμός? 5 Βαθμός? 12 !!! Σφάλμα: Βαθμός 0 έως 10 Βαθμός? 6 Βαθμός? 7 Βαθμός? 8 Βαθμός? 9 Βαθμός? 10 Μέσος όρος βαθμών: 5.5 </pre>

**Εργασία δ:** Εφαρμογή while. Συνεχής εισαγωγή από το πληκτρολόγιο: Αν θέλουμε να δώσουμε πολλές τιμές από το πληκτρολόγιο, μπορούμε να χρησιμοποιήσουμε for μόνο αν ξέρουμε το ακριβές πλήθος τους. Σε αντίθετη περίπτωση, χρησιμοποιούμε while.

- Γράψτε και εκτελέστε το παρακάτω πρόγραμμα (**program\_d.m**):

<b>program_d.m</b>
<pre> clc; clear all;  disp('Μετατροπή θερμοκρασίας (έξοδος προγράμματος με -999)'); disp(' '); % ή με fprintf('Μετατροπή θερμοκρασίας (έξοδος με -999)\n\n')  C = input('θερμοκρασία σε Celcius? '); while C~-999     F = (9/5)*C + 32;     fprintf('%.1f Celcius = %.1f Fahrenheit\n\n',C,F);     C = input('θερμοκρασία σε Celcius? '); end  disp('Bye bye'); </pre>
<b>Αποτελέσματα</b>
<pre> Μετατροπή θερμοκρασίας (έξοδος προγράμματος με -999)  θερμοκρασία σε Celcius? 100 100.0 Celcius = 212.0 Fahrenheit  θερμοκρασία σε Celcius? 200 200.0 Celcius = 392.0 Fahrenheit  θερμοκρασία σε Celcius? 300 300.0 Celcius = 572.0 Fahrenheit  θερμοκρασία σε Celcius? -999 Bye bye </pre>

**Εργασία ε:** Εφαρμογή while. Συνεχής λειτουργία μενού επιλογών: Η while μπορεί να χρησιμοποιηθεί για τη συνεχή λειτουργία ενός μενού, μέχρι να δοθεί κατάλληλη επιλογή εξόδου.

- Για να κατανοήσετε τη λειτουργία της εντολής while για τη συνεχή λειτουργία ενός μενού, γράψτε και εκτελέστε το παρακάτω πρόγραμμα (**program\_e.m**):

### **program\_e.m**

```
clc;
clear all;

disp('***CALCULATOR***');

selection = 'A'; % Αρχικοποίηση με μια τυχαία τιμή εκτός του 'E' ώστε να γίνει
                % είσοδος στον επαναληπτικό βρόχο while για 1η φορά

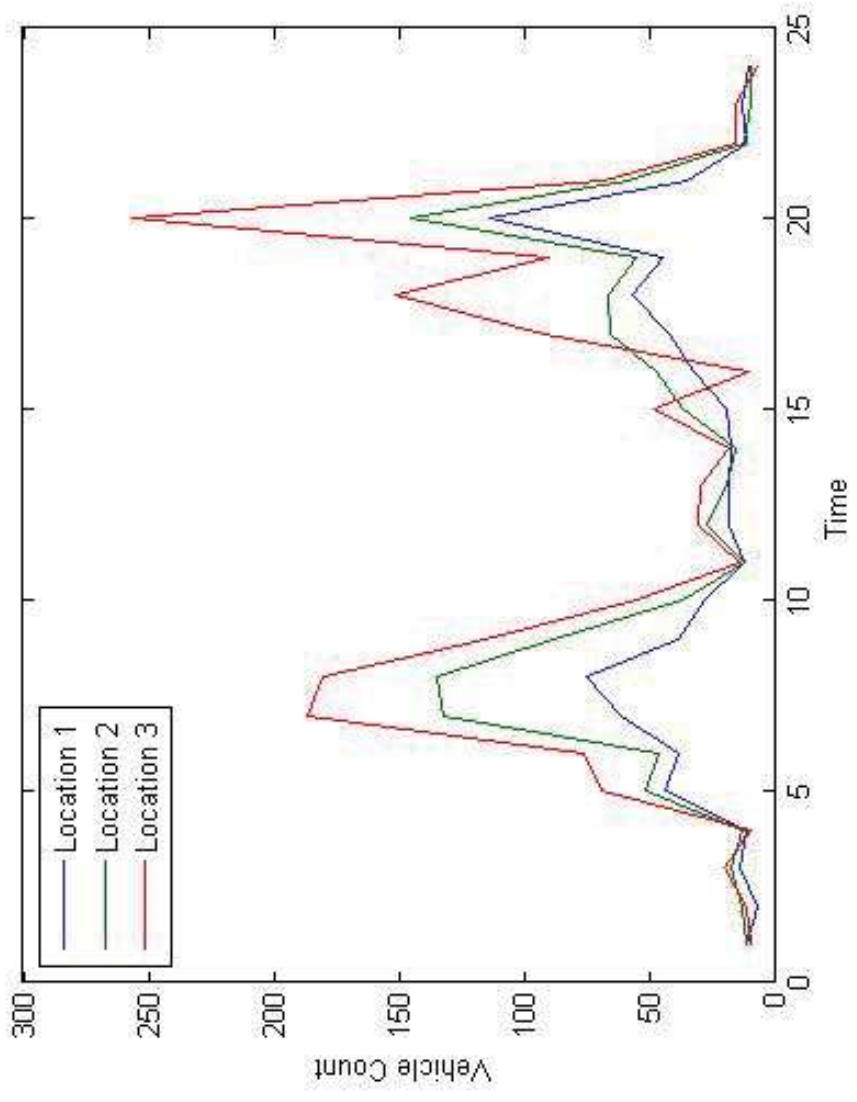
while selection~='E'
    % Κατασκευή του μενού
    clc;
    fprintf('\n');
    disp('***MENU***');
    disp('I. Input new numbers');
    disp('A. Addition (a+b)');
    disp('S. Subtraction (a-b)');
    disp('M. Multiplication (a*b)');
    disp('D. Division (a/b)');
    disp('P. Power (a^b)');
    disp('E. Exit');
    fprintf('\n');

    % Εισαγωγή από το χρήστη ενός χαρακτήρα
    selection = input('Press a capital letter (then press enter): ', 's');

    % Επιλογή και εκτέλεση αντίστοιχης ενέργειας
    if selection=='I'
        disp('Give me two numbers:');
        a = input('a = ');
        b = input('b = ');
    elseif selection=='A'
        fprintf('a + b = %f\nPress any key to continue...',a+b); pause;
    elseif selection=='S'
        fprintf('a - b = %f\nPress any key to continue...',a-b); pause;
    elseif selection=='M'
        fprintf('a*b = %f\nPress any key to continue...',a*b); pause;
    elseif selection=='D'
        if b~=0
            fprintf('a/b = %f\nPress any key to continue...',a/b); pause;
        else
            disp('Impossible division! - Press any key to continue...'); pause;
        end
    elseif selection=='P'
        if a<0 && b<1
            disp('Impossible power! - Press any key to continue...'); pause;
        else
            fprintf('a^b = %f\nPress any key to continue...',a^b); pause;
        end
    elseif selection=='E'
        disp('Bye bye...');
    else
        disp('Error in selection! - Press any key to continue...'); pause;
    end
end
```

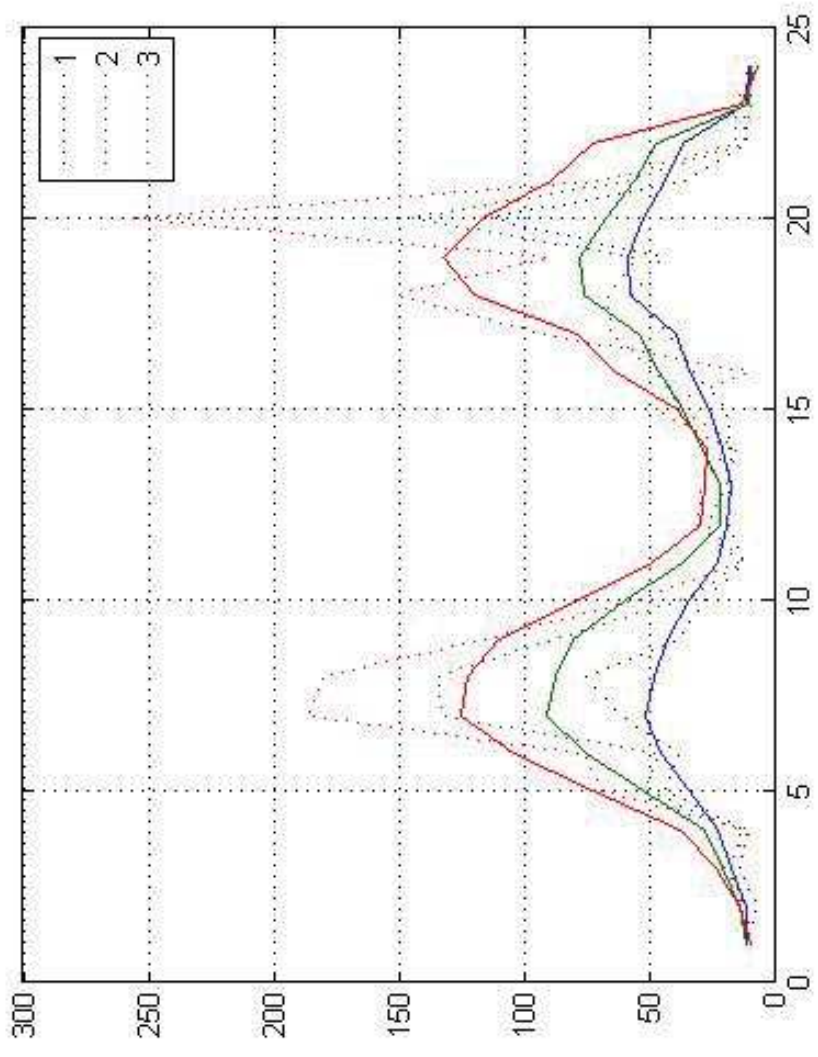
# Ανάλυση-επεξεργασία δεδομένων

- `load count.dat`
- `[n,p] = size(count)`
- `t = 1:n;`
- `plot(t,count),`
- `legend('Location 1','Location 2','Location 3',2)`
- `xlabel('Time'), ylabel('Vehicle Count')`



# Εξομάλυνση δεδομένων

- load count.dat
- n=length(count);
- 
- C1=count(:,1);
- C2=count(:,2);
- C3=count(:,3);
- C1s=smooth(count(:,1));
- C2s=smooth(count(:,2));
- C3s=smooth(count(:,3));
- 
- plot(1:n,C1,'.',1:n,C2,'.',1:n,C3,'. '), grid on
- h=legend('1','2','3');
- hold on;
- plot(1:n,C1s,1:n,C2s,1:n,C3s), grid on



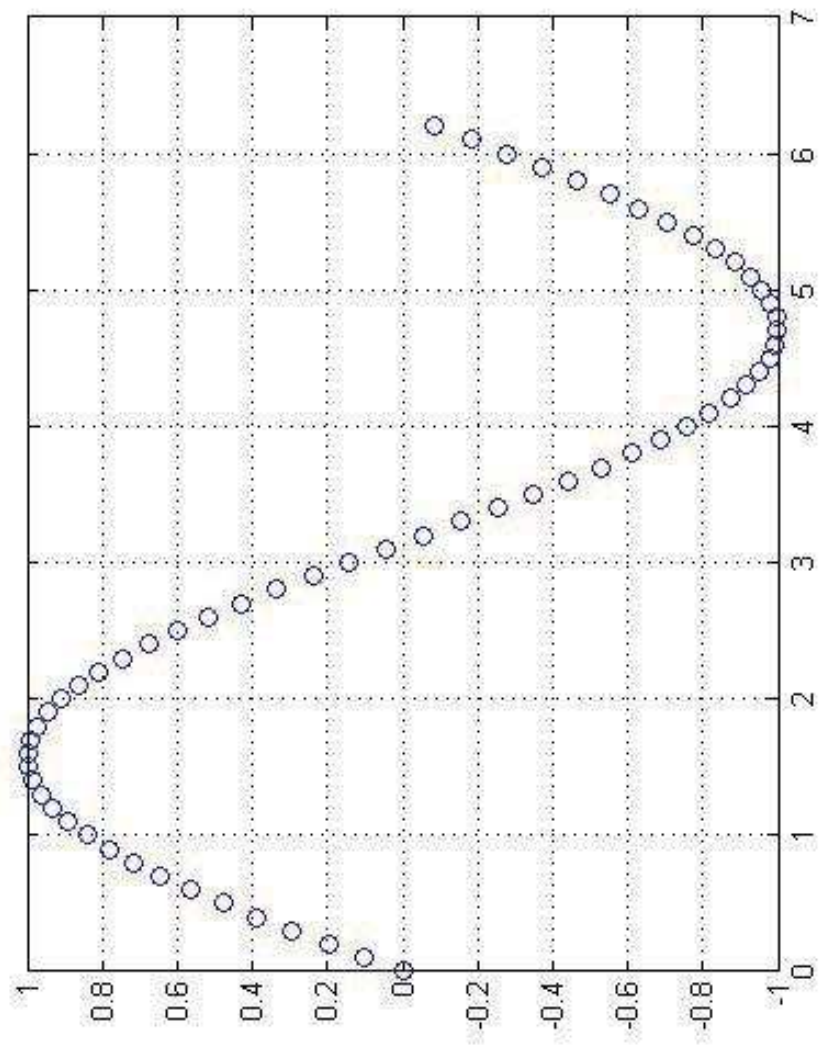
# Δεδομένα που περιέχουν και μη-μετρήσεις (π.χ. λόγω βλάβης του μετρητικού συστήματος)

- Θεωρήστε ένα 3x3 πίνακα δεδομένων που όμως το στοιχείο (2,2) δεν υπάρχει, όπως ο παρακάτω:
- `a = magic(3); a(2,2) = NaN`
- `a =`
- `8 1 6`
- `3 NaN 7`
- `4 9 2`
- Υπολογίστε το άθροισμα του πίνακα αυτού:
- `sum(a)`
- `ans =`
- `15 NaN 15`
- Πως αφαιρούμε ένα στοιχείο NaN από ένα διάνυσμα:
- `x(isnan(x)) = [];`



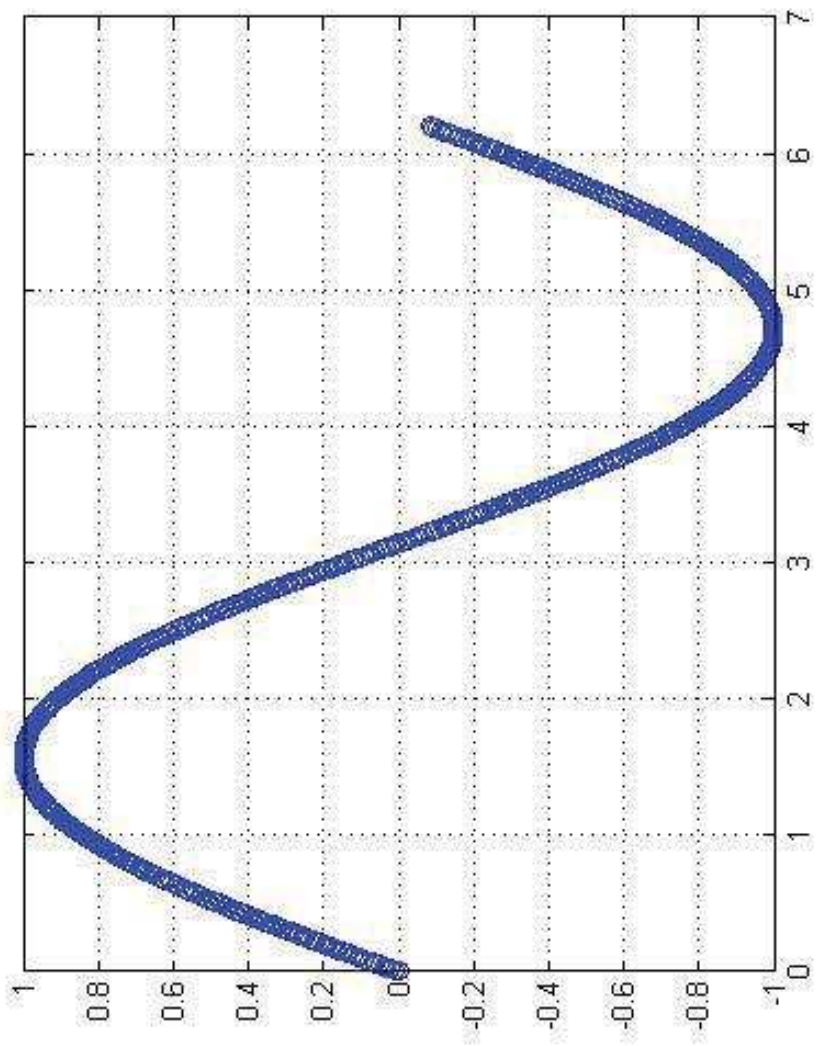
# Παρεμβολή δεδομένων

- `x=0:0.1:2*pi;`
- `y=sin(x);`
- `plot(x,y,'o'), grid on`



# Παρεμβολή δεδομένων

- `xi=0:0.01:2*pi;`
- `yi=interp1(x,y,xi);`
- `plot(xi,yi,'o'), grid on`



# Ανάλυση-επεξεργασία δεδομένων

- load count.dat

Μέγιστη τιμή

- $mx = \max(\text{count})$

Μέση τιμή

- $\mu = \text{mean}(\text{count})$

Τυπική απόκλιση

- $\sigma = \text{std}(\text{count})$

- $mx =$
- 114       145       257
- $\mu =$
- 32.0000   46.5417   65.5833
- $\sigma =$
- 25.3703   41.4057   68.0281

# Εύρεση της θέσης στη χρονοσειρά της μέγιστης και ελάχιστης τιμής

- `[mx,indx] = max(count)`

- `mx =`

- `114    145    257`

- `indx =`

- `20    20    20`

- `min(count(:))`

- `ans =`

- `7`

# Εύρεση και αφαίρεση αποκλίσεων από τα δεδομένα

- Στη στατιστική μία αποκλίνουσα τιμή είναι μία παρατήρηση (μέτρηση) που απέχει (αριθμητικά) πολύ από τα υπόλοιπα δεδομένα και μπορεί για παράδειγμα να οφείλεται σε μία βλάβη του μετρητικού μας συστήματος.
- Έστω ότι έχουμε έναν πίνακα  $A$  ο οποίος περιέχει τις μετρήσεις από ένα μετρητικό σύστημα που η λειτουργία του εδράζεται σε μία εργαστηριακή ή βιομηχανική διάταξη ενός αισθητήρα (λ.χ. κίνησης, θερμοκρασίας κ.λπ.). Η μαθηματική συνθήκη που μας δίνει τις πιθανές αποκλίνουσες τιμές που προέκυψαν από το εν λόγω πείραμα ή γενικότερα από τη λειτουργία του αισθητήρα είναι:

- $\text{abs}(A - \text{mean}(A)) > 3 * \text{std}(A)$

# Εύρεση και αφαίρεση αποκλίσεων από τα δεδομένα

- Ωστόσο θα πρέπει να είμαστε ιδιαίτερα προσεκτικοί όταν εφαρμόζουμε την προηγούμενη συνθήκη για την εύρεση πιθανών τιμών στις μετρήσεις μας που αποκλίνουν σημαντικά. Στο παράδειγμα που ακολουθεί λόγω χάρη θα πρέπει να λάβουμε υπόψη μας τις διαστάσεις του πίνακα μετρήσεων που προέκυψε από τη λειτουργία του αισθητήρα.
- **load count.dat**
- **size(count)**
- **MeanVal=mean(count);**
- **count-MeanVal ???**
- Η τελευταία πράξη ωστόσο δεν μπορεί να πραγματοποιηθεί γιατί οι δύο πίνακες (**count**, **MeanVal**) έχουν διαφορετικές διαστάσεις!!!



# Εύρεση και αφαίρεση αποκλίσεων από τα δεδομένα

- Θα πρέπει συνεπώς να φροντίσουμε οι δύο προηγούμενοι πίνακες (**count**, **MeanVal**) να έχουν τις ίδιες διαστάσεις ως εξής:
- **MeanMat=ones(24,1)\*mean(count);**
- **x = count - MeanMat;**
- Στη συνέχεια: **y=std(count);**
- **abs(x) > 3\*y ???**
- Ούτε η τελευταία πράξη όμως δεν μπορεί να πραγματοποιηθεί γιατί και αυτοί οι δύο πίνακες (**x**, **y**) έχουν διαφορετικές διαστάσεις!!!

# Εύρεση και αφαίρεση αποκλίσεων από τα δεδομένα

- Συνεπώς φροντίζουμε εκ νέου και αυτοί οι δύο πίνακες (**x**, **y**) να έχουν τις ίδιες διαστάσεις:
- **outliers = abs(x) > 3\*ones(24,1)\*y;**
- Αναλυτικά η συνθήκη αποκλίσεων για το παράδειγμα αυτό είναι:
- **outliers =  
abs(count - ones(24,1)\*mean(count)) > 3\*ones(24,1)\*std(count);**
- Πλήθος outliers: **nout = sum(outliers);**
- Αφαίρεση outliers από τις μετρήσεις: **count(any(outliers,2),:) = [ ];**
- **size(count)**

Φασματική ανάλυση

**Fast Fourier Transform**

# Ανάλυση Fourier □

- ▣ Η **ανάλυση Fourier** είναι ένα πεδίο των εφαρμοσμένων μαθηματικών το οποίο προέκυψε από την προσπάθεια αναπαράστασης μίας συνάρτησης ως αθροίσματος απλούστερων περιοδικών τριγωνομετρικών συναρτήσεων.
- ▣ Επομένως κεντρική ιδέα στην ανάλυση Fourier είναι η προσπάθεια για κατανόηση των ιδιοτήτων μίας συνάρτησης (η οποία μπορεί να αναπαριστά π.χ. ένα σήμα) μέσω διάσπασής της σε γνωστά, στοιχειώδη μέρη (*αποσύνθεση*). Η ανάστροφη διαδικασία, η κατασκευή μίας συνάρτησης από γνωστές, βασικές συναρτήσεις, ονομάζεται *σύνθεση*. Με τον όρο *ανάλυση Fourier* αναφερόμαστε και στις δύο διεργασίες.
- ▣ Η μέθοδος αυτή εφαρμόστηκε για πρώτη φορά από τον Ζοζέφ Φουριέ στην προσπάθειά του να ερευνήσει τη διάδοση της θερμότητας.

# Εφαρμογές στην επεξεργασία σήματος

- ▣ Όταν επεξεργαζόμαστε σήματα, όπως ήχο, ραδιοκύματα, κύματα φωτός, σεισμικά κύματα, ακόμα και εικόνες, η ανάλυση Fourier μπορεί να απομονώσει μεμονωμένους συντελεστές από μια σύνθετη κυματομορφή, συγκεντρώνοντάς τους για ευκολότερη ανίχνευση και/ή αφαίρεση. Μία μεγάλη οικογένεια τεχνικών επεξεργασίας σήματος αποτελείται από μετασχηματισμό Fourier ενός σήματος, χειρισμό μετασχηματισμένων με Fourier δεδομένων με απλό τρόπο και αντιστροφή του μετασχηματισμού.
- ▣ Μερικά παραδείγματα είναι τα παρακάτω:
- ▣ Τηλεφωνική κλήση: το τονικό σήμα για κάθε πλήκτρο τηλεφώνου, όταν πιέζεται, είναι το καθένα ένα σύνολο από δύο ξεχωριστούς τόνους (συχνότητες). Η ανάλυση Fourier μπορεί να χρησιμοποιηθεί για να διαχωριστεί (ή *αναλυθεί*) το σήμα του τηλεφώνου, για να αποκαλύψει τους δύο τόνους από τους οποίους αποτελείται και συνεπώς ποιο κουμπί πατήθηκε.
- ▣ Περιφραξη θορύβου από ηχογραφήσεις για να αφαιρεθεί ο «ήσυχος» θόρυβος του παρασκηνίου με την εξάλειψη των συντελεστών Fourier που δεν υπερβαίνουν ένα καθορισμένο εύρος,
- ▣ Εξίσωση των ηχητικών ηχογραφήσεων με μία σειρά από ζωνοπερατά φίλτρα,
- ▣ Ψηφιακή ραδιοφωνική λήψη χωρίς υπερετερώδυνο κύκλωμα, όπως σε ένα σύγχρονο κινητό τηλέφωνο,
- ▣ Δημιουργία του ηχητικού φασματογραφήματος που χρησιμοποιείται για την ανάλυση ήχων.

# FFT – Παράδειγμα

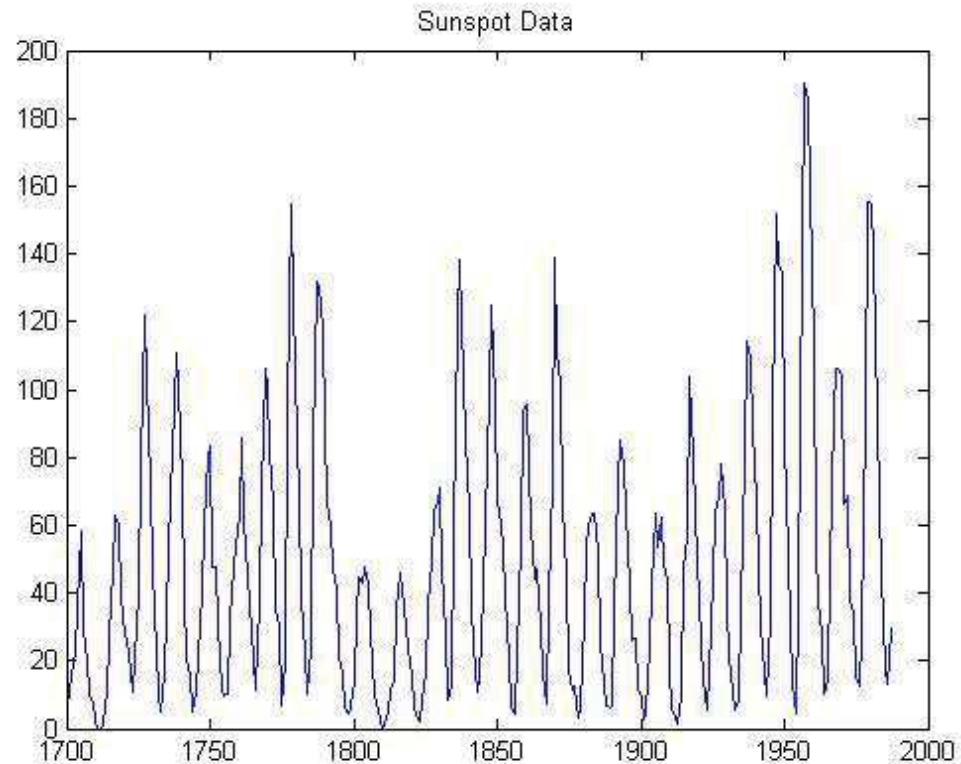
- ▣ Οι επιστήμονες ιχνηλατούν τους **ηλιακούς κύκλους** μετρώντας τις **ηλιακές κηλίδες** – ψυχρές περιοχές, στο μέγεθος πλανητών, πάνω στον Ήλιο όπου ισχυροί μαγνητικοί βρόγχοι διαγράφονται μέσω της ορατής επιφάνειας του αστεριού.
- ▣ Οι **ηλιακές κηλίδες** είναι παροδικά φαινόμενα που εμφανίζονται στην επιφάνεια του Ηλίου, τη λεγόμενη φωτόσφαιρα, της οποίας και θεωρούνται οι περισσότερο εντυπωσιακοί και ενδιαφέροντες σχηματισμοί της. Είναι ορατές ως σκοτεινές μικρές ή μεγαλύτερες κυκλικές επιφάνειες - κηλίδες, σε σχέση με τις γειτονικές περιοχές της φωτόσφαιρας, που περιβάλλονται από λιγότερο σκοτεινές στεφάνες ινώδους υφής.
- ▣ Ο αριθμός των κηλίδων στην ηλιακή επιφάνεια αυξάνεται γρήγορα και μετά μειώνεται με βραδύτερο ρυθμό κάθε περίπου 11 χρόνια. Αυτή η περιοδικότητα, την οποία ακολουθεί η γενικότερη ηλιακή δραστηριότητα, αποκαλείται «**ενδεκαετής ηλιακός κύκλος**» ή «**ενδεκαετής κύκλος της ηλιακής δραστηριότητας**».

# FFT – Παράδειγμα



# FFT – Γραφική παράσταση I

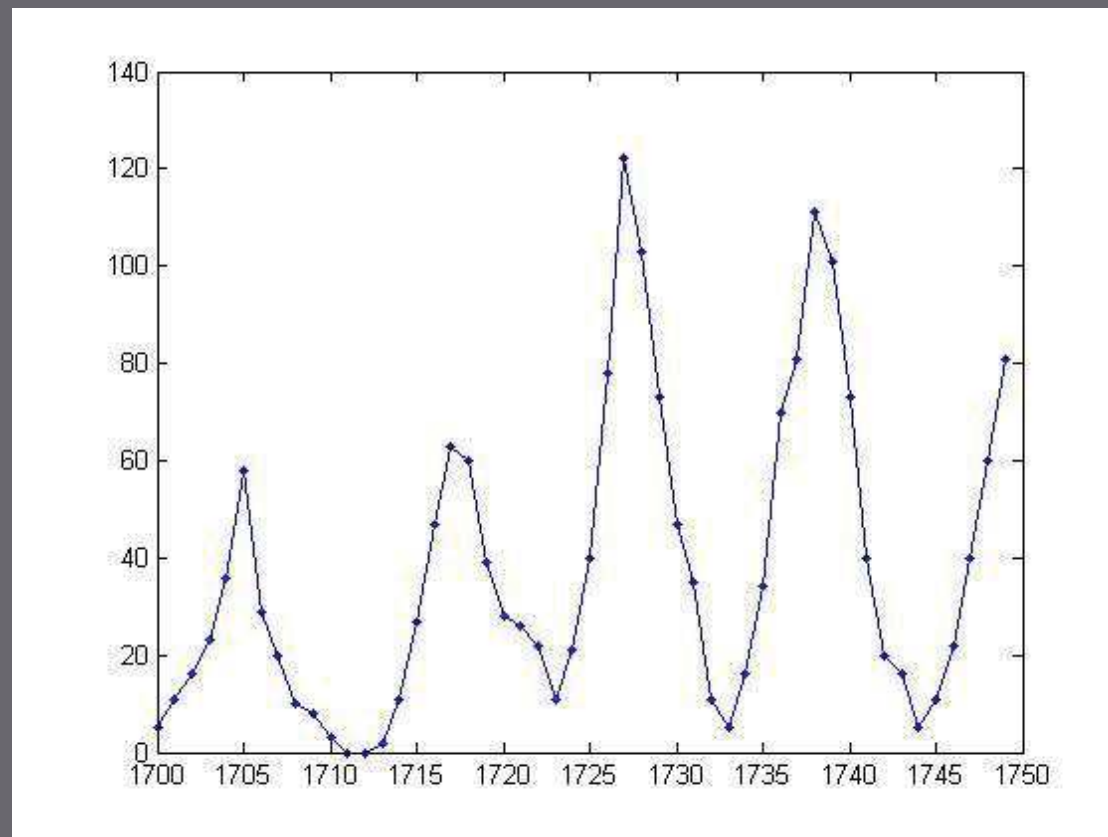
```
load sunspot.dat  
year=sunspot(:,1);  
relNums=sunspot(:,2);  
plot(year,relNums)  
title('Sunspot Data')
```





# FFT – Γραφική παράσταση II

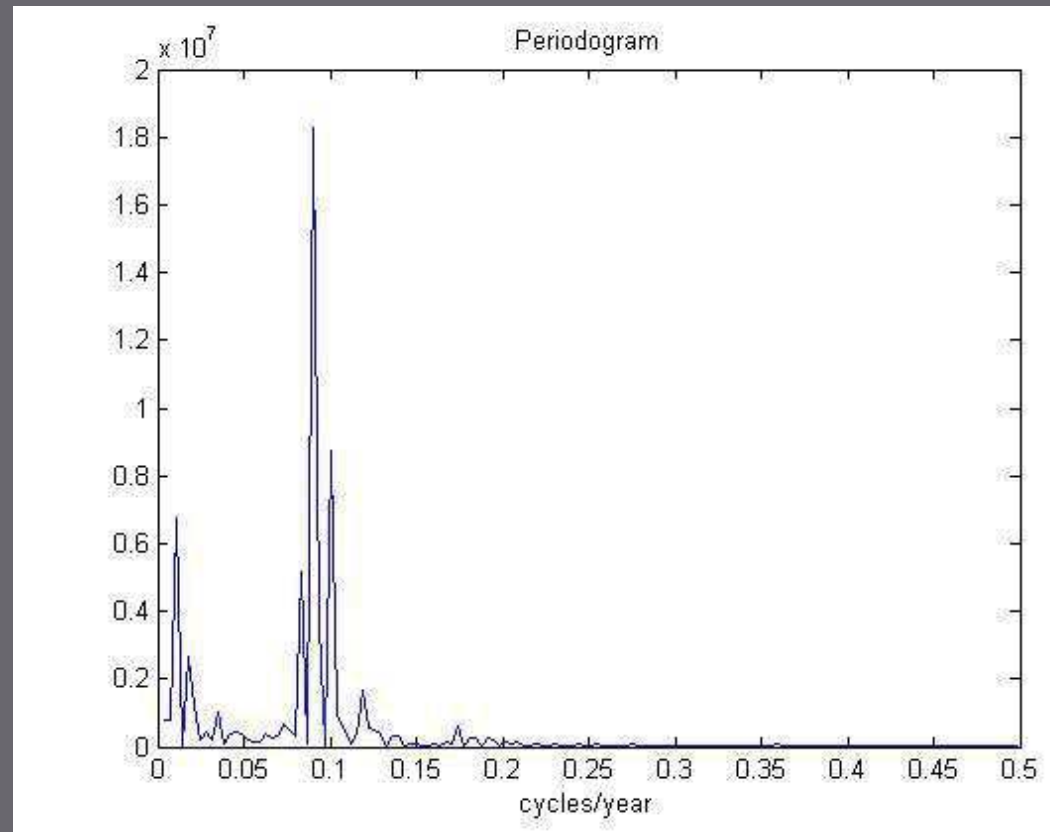
```
plot(year(1:50),relNums(1:50),'b.-');
```





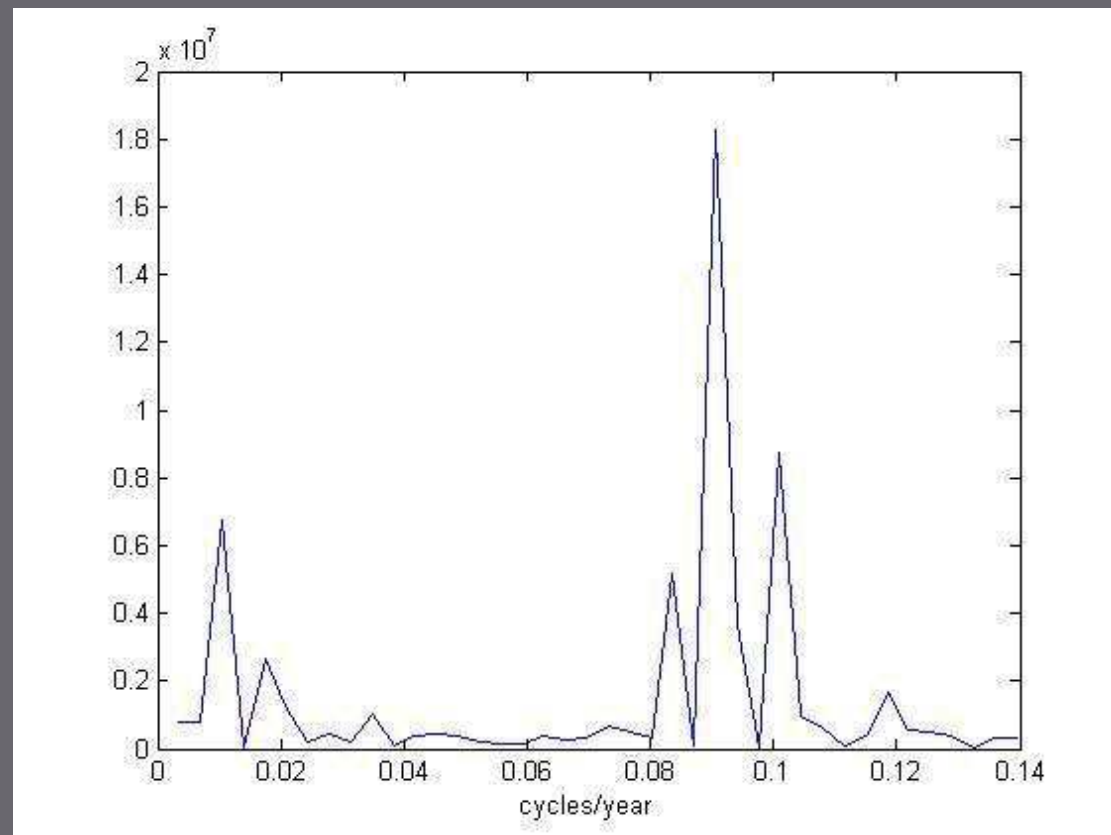
# Περιοδόγραμμα I

```
Y = fft(relNums);  
Y(1)=[ ];  
  
n=length(Y);  
power =  
    abs(Y(1:floor(n/2))).^2;  
nyquist = 1/2;  
freq = (1:n/2)/(n/2)*nyquist;  
plot(freq,power)  
xlabel('cycles/year')  
title('Periodogram')
```



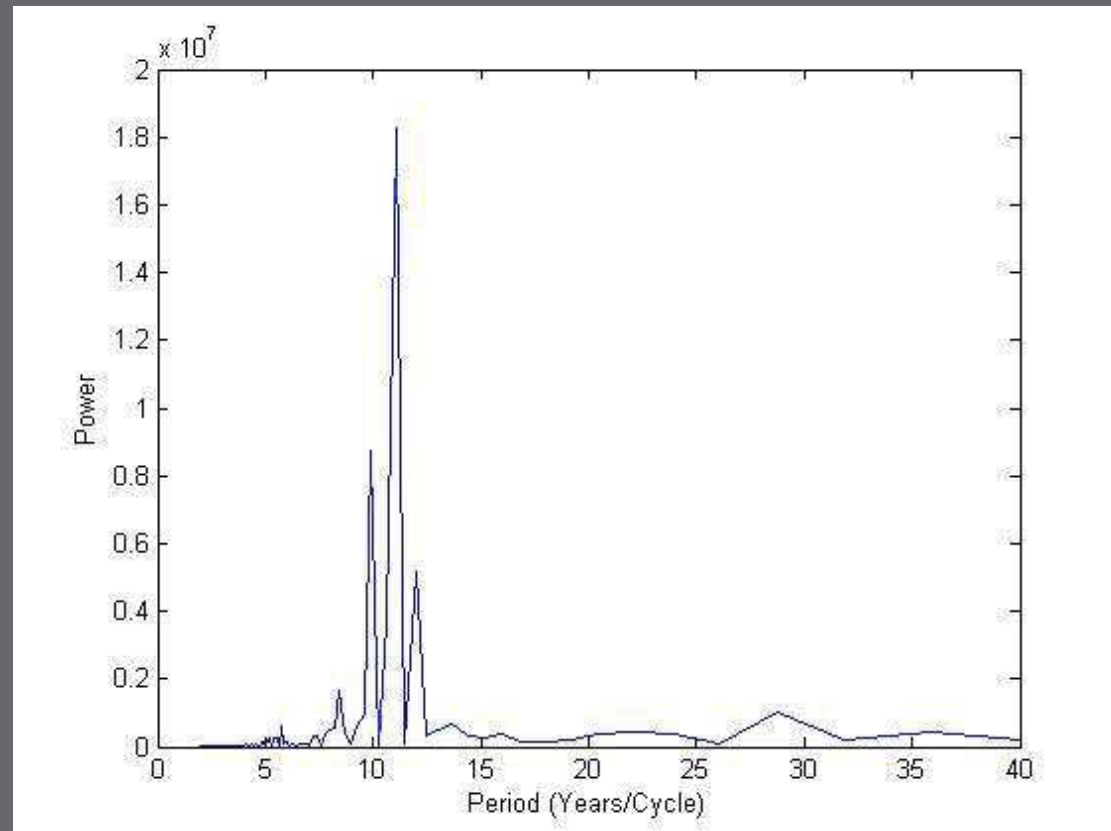
# Περιοδόγραμμα II

```
plot(freq(1:40),power(1:40))  
xlabel('cycles/year')
```



# Περιοδόγραμμα I

```
period=1./freq;  
plot(period,power);  
axis([0 40 0 2e+7]);  
ylabel('Power');  
xlabel('Period  
(Years/Cycle)');
```



# Εύρεση περιόδου ηλιακού κύκλου

```
hold on;  
index=find(power==  
max (power));  
mainPeriodStr=  
num2str (period(index));  
plot(period(index),  
power (index),'r.',  
      'MarkerSize',25);  
text(period(index)+2,  
power (index),  
['Period = ',mainPeriodStr]);  
hold off;
```

