

Σήματα και Συστήματα Οδηγός Octave/Matlab

ECE-HMU



Κώστας Μαριάς
Ρούλα Χατζάκη



[CC BY-NC-SA 3.0 GR](https://creativecommons.org/licenses/by-nc-sa/3.0/gr/)

2021

Contents

1. Εισαγωγή σε Octave/Matlab	2
1.1. Παράθυρα εργασίας και βασικές λειτουργίες	2
1.2. Απλές αριθμητικές πράξεις	5
1.3. Ενσωματωμένες συναρτήσεις	7
1.4. Σταθερές και μεταβλητές	8
2. Σειρές/Διανύσματα	9
2.1. Πράξεις με σειρές	10
2.2. Διανύσματα γραμμής – στήλης	12
2.3. Εσωτερικό γινόμενο διανυσμάτων	13
3. Πίνακες	13
3.1. Πράξεις με πίνακες	16
4. Loops-Control statements	19
5. m files-Script-Functions	21
6. Γραφικές Παραστάσεις	22
Bibliography	25

1. Εισαγωγή σε Octave/Matlab

Το MATLAB (matrix laboratory) είναι ένα περιβάλλον για επιστημονικό προγραμματισμό. Αποθηκεύει και κάνει τις πράξεις με βάση την άλγεβρα μητρών. Διαθέτει ισχυρές δυνατότητες στον τομέα των γραφικών καθώς μπορεί να υλοποιήσει από συναρτήσεις πραγματικές, μιγαδικές, πεπλεγμένες συναρτήσεις δύο μεταβλητών κ.α, έως και ιστογράμματα, τομεογράμματα, ραβδοδιαγράμματα, εμβοδογράμματα και άλλα. Το MATLAB είναι ένα εμπορικό πρόγραμμα που υποστηρίζεται από την εταιρεία [Mathworks](https://www.mathworks.com/).

Το Octave είναι ένα περιβάλλον επιστημονικού προγραμματισμού «όμοιο» με το MATLAB. Το Octave είναι ελεύθερο για διανομή, εγκατάσταση και χρήση από όλους (GNU-General Public License).

Οι γλώσσες προγραμματισμού του Matlab και Octave είναι γλώσσες υψηλού προγραμματισμού και έχουν ελάχιστες διαφορές στην σύνταξη. Το MATLAB είναι πιο ευέλικτο, προηγμένο και ισχυρό από το Octave αλλά παραμένει μια δαπανηρή λύση έναντι της δωρεάν.

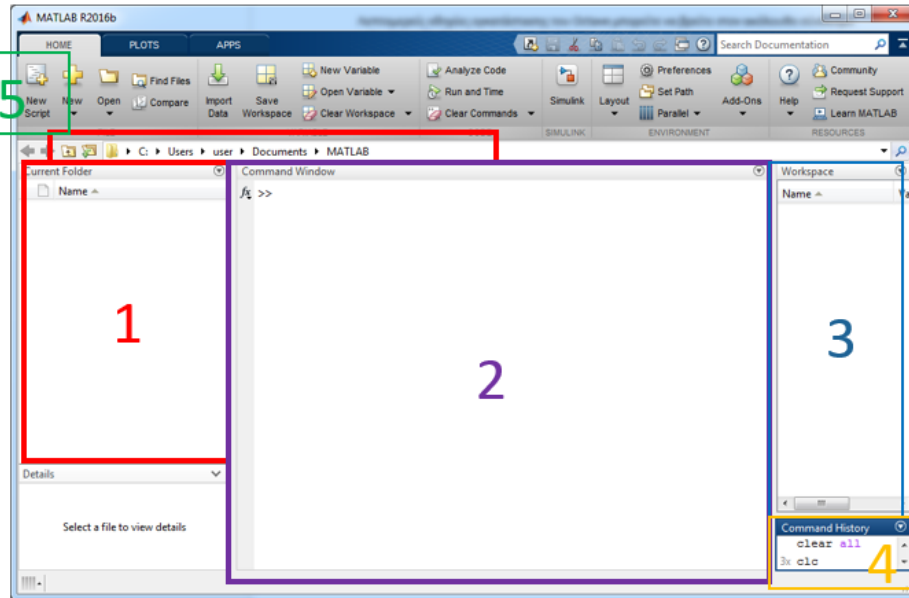
Λεπτομερείς οδηγίες εγκατάστασης του Octave μπορείτε να βρείτε στον ακόλουθο σύνδεσμο wiki : https://wiki.octave.org/Using_Octave . Η τελευταία έκδοση του octave με windows installer είναι η 6.2.0 (released on February 19, 2021) την οποία μπορείτε να κατεβάσετε από τον ακόλουθο σύνδεσμο <https://ftp.gnu.org/gnu/octave/windows/> .

1.1. Παράθυρα εργασίας και βασικές λειτουργίες

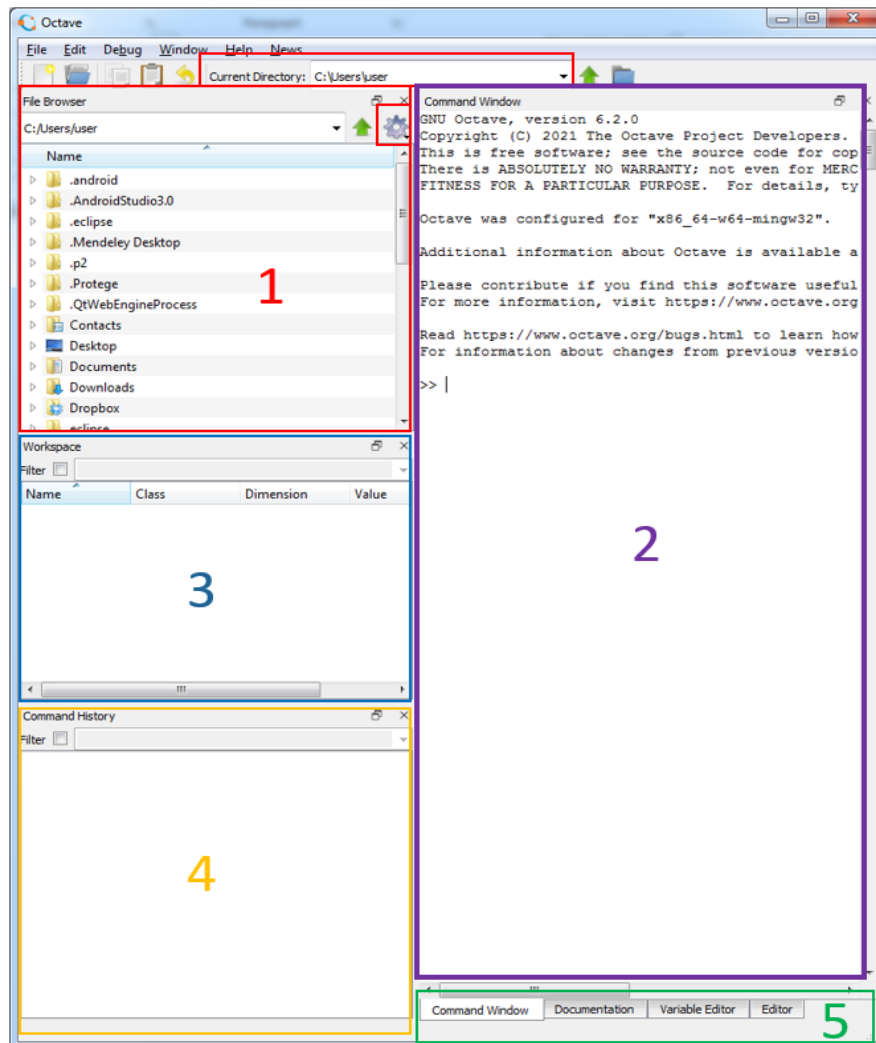
Το κύριο περιβάλλον του Matlab αποτελείται από τέσσερα βασικά παράθυρα εργασίας: τρέχων φάκελος (Current Folder), παράθυρο εντολών (Command Window), χώρου εργασίας (Workspace), και ιστορικού εντολών (Command History), όπως παρουσιάζονται στην Εικόνα 1. Το παράθυρο του επεξεργαστή (Editor) θα ανοίξει όταν κανείς επιλέξει την δημιουργία νέου script (.m file extension) από το κύριο μενού ή με την τροποποίηση της διάταξης από το «Layout tab» στο κύριο μενού.

Αντίστοιχα, ακολουθώντας την ίδια λειτουργικότητα και λίγο διαφορετική ονομασία, η διάταξη των παραθύρων εργασίας του Octave παρουσιάζεται στην Εικόνα 2. Μπορείτε να αλλάξετε τη διάταξη αυτή σύμφωνα με τις ανάγκες σας από το "Window tab" στο κύριο μενού. Πιο αναλυτικά:

- Η επιλεγμένη περιοχή με σήμανση «1» δείχνει τον κατάλογο εργασίας (Working Directory). Μπορείτε να ορίσετε έναν νέο Working Directory εάν πατήσετε το κουμπί γραναζιού. Μπορείτε επίσης να δείτε τη διαδρομή του τρέχοντος Working Directory εάν πληκτρολογήσετε την εντολή «pwd» στο παράθυρο εντολών.
- Η επιλεγμένη περιοχή με σήμανση «2» αφορά το παράθυρο εντολών (Command Window). Το Command Window είναι ο χώρος που πληκτρολογούνται οι εντολές.
- Η επιλεγμένη περιοχή με σήμανση «3» είναι ο χώρος εργασίας (Workspace). Εδώ μπορείτε να δείτε οποιαδήποτε μεταβλητή, μήτρα κ.λπ. που χρησιμοποιείτε.
- Η επιλεγμένη περιοχή με σήμανση «4» είναι το ιστορικό εντολών (Command History). Οι εντολές που έχετε καταχωρήσει θα πρέπει να εμφανίζονται εδώ.
- Η επιλεγμένη περιοχή με σήμανση «5» σας δίνει πρόσβαση στον επεξεργαστή (Editor) για να γράψετε σενάρια ή συναρτήσεις, αλλά και σε μια λεπτομερή τεκμηρίωση του προγράμματος (Documentation) και του ιστορικού εντολών.



Εικόνα 1. Παράθυρα εργασίας Matlab



Εικόνα 2. Παράθυρα εργασίας Octave

Πριν την έναρξη χρήσης των πλατφορμών και ανάλογα με το γνωστικό αντικείμενο εργασίας θα πρέπει κάνεις να δει τα εγκατεστημένα πακέτα (packages) για το Octave και τις εργαλειοθήκες (toolboxes) για το Matlab.

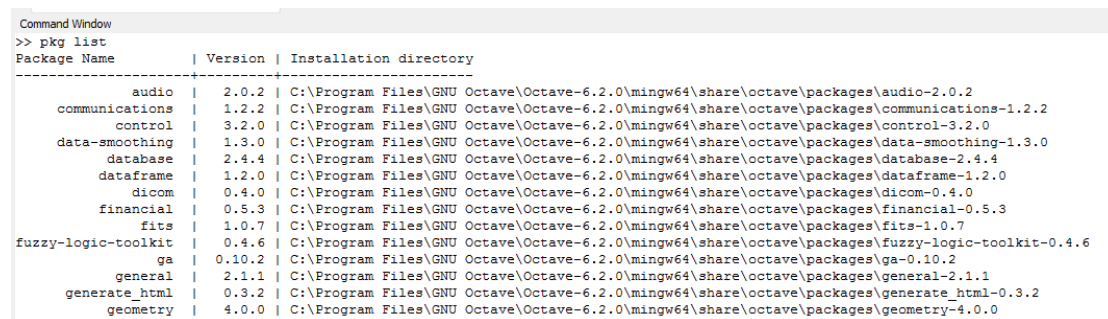
Για να δείτε την λίστα με τα πακέτα που έχουν ήδη εγκατασταθεί στο Octave, πληκτρολογήστε :

```
>> pkg list
```

Στην Εικόνα 3 παρουσιάζονται τα αποτελέσματα της εντολής. Τα πακέτα που έχουν ήδη φορτωθεί επισημαίνονται με *

Το [Octave Forge](#) είναι μια κεντρική τοποθεσία για την ανάπτυξη πακέτων για το GNU Octave, παρόμοια με τις εργαλειοθήκες του Matlab. Για να εγκαταστήσετε ένα πακέτο, χρησιμοποιήστε την εντολή «pkg» πληκτρολογώντας:

```
>> pkg install -forge package_name
>> pkg install package_name
>> pkg load package_name
```



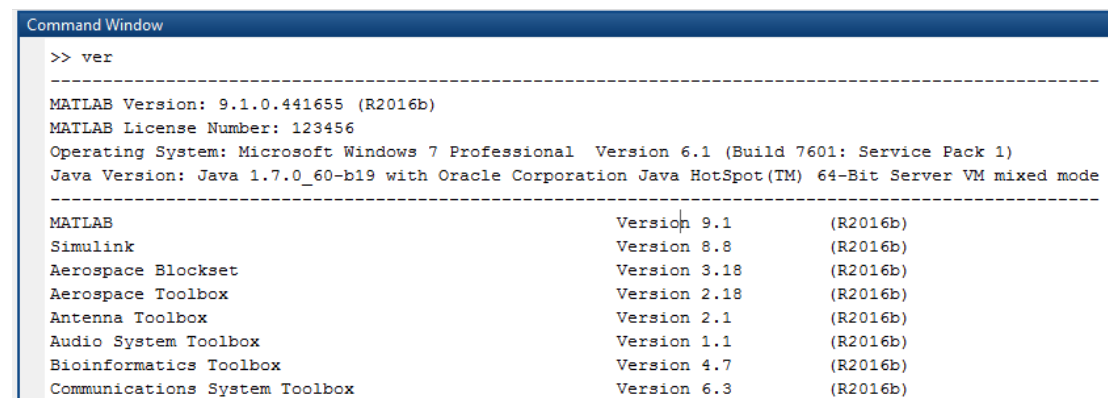
```
Command Window
>> pkg list
Package Name      | Version | Installation directory
-----
audio             | 2.0.2   | C:\Program Files\GNU Octave\Octave-6.2.0\mingw64\share\octave\packages\audio-2.0.2
communications    | 1.2.2   | C:\Program Files\GNU Octave\Octave-6.2.0\mingw64\share\octave\packages\communications-1.2.2
control           | 3.2.0   | C:\Program Files\GNU Octave\Octave-6.2.0\mingw64\share\octave\packages\control-3.2.0
data-smoothing    | 1.3.0   | C:\Program Files\GNU Octave\Octave-6.2.0\mingw64\share\octave\packages\data-smoothing-1.3.0
database          | 2.4.4   | C:\Program Files\GNU Octave\Octave-6.2.0\mingw64\share\octave\packages\database-2.4.4
dataframe         | 1.2.0   | C:\Program Files\GNU Octave\Octave-6.2.0\mingw64\share\octave\packages\dataframe-1.2.0
dicom             | 0.4.0   | C:\Program Files\GNU Octave\Octave-6.2.0\mingw64\share\octave\packages\dicom-0.4.0
financial         | 0.5.3   | C:\Program Files\GNU Octave\Octave-6.2.0\mingw64\share\octave\packages\financial-0.5.3
fits              | 1.0.7   | C:\Program Files\GNU Octave\Octave-6.2.0\mingw64\share\octave\packages\fits-1.0.7
fuzzy-logic-toolkit | 0.4.6   | C:\Program Files\GNU Octave\Octave-6.2.0\mingw64\share\octave\packages\fuzzy-logic-toolkit-0.4.6
ga                | 0.10.2  | C:\Program Files\GNU Octave\Octave-6.2.0\mingw64\share\octave\packages\ga-0.10.2
general           | 2.1.1   | C:\Program Files\GNU Octave\Octave-6.2.0\mingw64\share\octave\packages\general-2.1.1
generate_html     | 0.3.2   | C:\Program Files\GNU Octave\Octave-6.2.0\mingw64\share\octave\packages\generate_html-0.3.2
geometry          | 4.0.0   | C:\Program Files\GNU Octave\Octave-6.2.0\mingw64\share\octave\packages\geometry-4.0.0
```

Εικόνα 3. Octave - Package List

Για να δείτε τις εγκατεστημένες εργαλειοθήκες (toolboxes) στο Matlab αλλά και την εγκατεστημένη έκδοση αυτού (Εικόνα 4), πληκτρολογήστε στο Command Window:

```
>> ver
```

Για την διαχείριση και εγκατάσταση νέων toolboxes στο Matlab, περιηγηθείτε από το κύριο μενού στην επιλογή Add-Ons. Λεπτομερείς οδηγίες μπορείτε να βρείτε [εδώ](#).



```
Command Window
>> ver
-----
MATLAB Version: 9.1.0.441655 (R2016b)
MATLAB License Number: 123456
Operating System: Microsoft Windows 7 Professional Version 6.1 (Build 7601: Service Pack 1)
Java Version: Java 1.7.0_60-b19 with Oracle Corporation Java HotSpot(TM) 64-Bit Server VM mixed mode
-----
MATLAB                Version 9.1      (R2016b)
Simulink              Version 8.8      (R2016b)
Aerospace Blockset   Version 3.18    (R2016b)
Aerospace Toolbox    Version 2.18    (R2016b)
Antenna Toolbox       Version 2.1     (R2016b)
Audio System Toolbox  Version 1.1     (R2016b)
Bioinformatics Toolbox Version 4.7     (R2016b)
Communications System Toolbox Version 6.3     (R2016b)
```

Εικόνα 4. Matlab-Toolboxes

Το MATLAB και το Octave λειτουργούν με εντολές, που είτε δίνονται απ' ευθείας στο Command Window, είτε γράφονται σε κώδικα προγράμματος με τη βοήθεια του Editor. Κάθε μεταβλητή κατέχει θέση στη μνήμη του υπολογιστή, παίρνοντας τιμή που μεταβάλλεται με τη βοήθεια εντολών. Οι μεταβλητές αποθηκεύονται στο Workspace. Για να μπορεί να εκτελεστεί ένα πρόγραμμα, πρέπει να βρίσκεται στο Current Folder/Directory.

Πίνακας 1. Βασικές εντολές διαχείρισης

Εντολή	Ερμηνεία
exit, quit	Έξοδος από το πρόγραμμα
clear	Διαγραφή ενεργών μεταβλητών
clc	Καθαρισμός παραθύρου εργασίας
diary	Αποθήκευση εργασίας σε αρχείο
help	Βοήθεια
who, whload	Κατάλογος ενεργών μεταβλητών εργασίας
save	Αποθήκευση σε αρχείο των μεταβλητών εργασίας
load	Φόρτωση από αρχείο των μεταβλητών εργασίας

***Στον ακόλουθο σύνδεσμο μπορείτε να βρείτε μια σειρά μαθημάτων για το Matlab Mathworks: https://matlabacademy.mathworks.com/?s_tid=acb_tut.

***Αντιστοίχως μια σειρά από βίντεο-μαθήματα για το OCTAVE είναι διαθέσιμα στον ακόλουθο σύνδεσμο wiki https://wiki.octave.org/Video_tutorials.

1.2. Απλές αριθμητικές πράξεις

Το MATLAB και Octave χρησιμοποιούν τους τελεστές +, -, * και / για τις τέσσερις αριθμητικές πράξεις. Οι πράξεις γίνονται από αριστερά προς τα δεξιά, σε ίδια προτεραιότητα. Η προτεραιότητα αλλάζει με χρήση παρενθέσεων. Οι αγκύλες [...] και τα άγκιστρα {...} έχουν διαφορετικές χρήσεις. Οι εντολές καταχωρούνται με (Enter) και τα αποτελέσματα εμφανίζονται αν υπάρχουν εντολές εμφάνισης. Η έλλειψη του «;» στο τέλος της εντολής είναι πιο απλή εντολή εμφάνισης των αποτελεσμάτων. Σε μία γραμμή μπορούν να γραφτούν πολλές εντολές. Οι εντολές χωρίζονται μεταξύ τους με «;» ή «,». Αν σε μια εντολή που εξαγει αποτέλεσμα δεν οριστεί μεταβλητή, αυτό αποθηκεύεται αυτόματα σε μια μεταβλητή με όνομα « ans ». Κάποια απλά παραδείγματα:

```
>> x=1; y=7
Y=7
>> 3 + 5
ans = 8
>> 3 - 5
ans = -2
>> 3 * 5
ans = 15
>> 3/5
ans = 0.6
```

Οι πράξεις μπορούν να συνδεθούν και αλυσιδωτά όπως:

```
>> 3+5+2
ans =10
>> 3*3*3
ans = 27
```

όπου ειδικά για την τελευταία υπάρχει και ο τελεστής της ύψωσης σε δύναμη ^:

```
>> 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 πιο πολύπλοκες εκφράσεις χρησιμοποιούνται παρενθέσεις κατά τον συνήθη τρόπο:

```
>> 2^5 + 4*(33 - 2*(6+2/7))
ans = 113.7143
```

Πίνακας 2.. Αριθμητικοί & Αυξητικοί Τελεστές (Arithmetic & Increment Operators) *Ιδίοι σε Matlab και Octave.

x + y	Πρόσθεση (Addition) των αριθμών ή συμβατών διανυσμάτων x και y
x - y	Αφαίρεση (Subtraction) των αριθμών ή συμβατών διανυσμάτων x και y
x * y	Πολλαπλασιασμός αριθμών ή συμβατών πινάκων (matrix multiplication)
x .* y	Στοιχείο προς στοιχείο πολλαπλασιασμός (element by element multiplication)
x / y	Δεξιά διαίρεση (right division, conceptually equivalent to (inverse (y') * x)')
x ./ y	Στοιχείο προς στοιχείο δεξιά διαίρεση (element by element right division)
x \ y	Αριστερή διαίρεση (left division, conceptually equivalent to inverse (x) * y)
x .\ y	Στοιχείο προς στοιχείο αριστερή διαίρεση (element by element left division)
x ^ y	Τελεστής ύψωσης σε δύναμη (power operator)
x .^ y	Στοιχείο προς στοιχείο τελεστής ύψωσης σε δύναμη (power operator)
- x	Αντίθεση (negation)
+ x	unary plus (a no-op)
x'	complex conjugate transpose
x.'	Μετάθεση (transpose)
++ x (-- x)	Αύξηση (Μείωση) - increment (decrement), return new value
x ++ (x --)	Αύξηση (Μείωση) - increment (decrement), return old value

Πίνακας 3. Συγκριτικοί & Boolean Τελεστές (Comparison and Boolean Operators). Αυτοί οι τελεστές δουλεύουν στοιχείο-προς-στοιχείο, αξιολογώντας πάντοτε και τα δυο ορίσματα.*Ιδίοι σε Matlab και Octave.

x < y	Αληθές εάν x είναι μικρότερο από y
x <= y	Αληθές εάν x είναι μικρότερο από ή ίσο του y
x == y	Αληθές εάν x είναι ίσο του y
x >= y	Αληθές εάν x είναι μεγαλύτερο από ή ίσο του y
x > y	Αληθές εάν x είναι μεγαλύτερο του y
x != y	Αληθές εάν x δεν είναι ίσο του y
x & y	Αληθές εάν και τα δύο x και y είναι αληθή
x y	Αληθές εάν το λιγότερο ένα από τα δύο x ή y είναι αληθή
x && y	Αληθές εάν και τα δύο x και y είναι αληθή
x y	Αληθές εάν το λιγότερο ένα από τα δύο x ή y είναι αληθή

1.3. 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 = 1.4142

>> exp(1)           % εκθετική συνάρτηση
ans = 2.7183

>> log(exp(1))      % φυσικός λογάριθμος
ans = 1

>>log10(10^2)       % δεκαδικός λογάριθμος
ans = 2
```

Τριγωνομετρικές συναρτήσεις

```
>> pi               % η σταθερά π
ans = 3.1416

>> sin(pi/4)        % ημίτονο
ans = 0.7071

>> cos(pi/2)        % συνημίτονο
ans =6.1230e-017    ← πρακτικά το αποτέλεσμα
                    είναι 0

>> tan(pi/4)        % εφαπτομένη
ans = 1.0000

>> asin(0.5)        % τόξο ημιτόνου
ans = 0.5236

>> atan(1)          % τόξο εφαπτομένης
ans = 0.7854
```

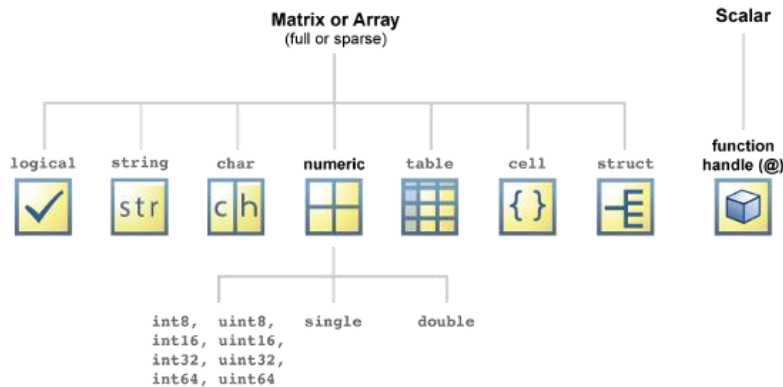
όπου το σύμβολο '%' χρησιμοποιείται για την εισαγωγή σχολίων. Αν η γωνία δίνεται σε μοίρες, τότε την μετατρέπουμε σε ακτίνια πολλαπλασιάζοντας με το $\pi/180$. Παράδειγμα υπολογισμού του $\cos(60^\circ)$:

```
>> cos(60*pi/180)
ans = 0.5000
```

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.

1.4. Σταθερές και μεταβλητές

Οι μεταβλητές είναι θέσεις στη μνήμη με όνομα, τύπο και τιμή. Τα ονόματα των μεταβλητών: μπορούν να ξεκινούν από λατινικό γράμμα (Case sensitive) και μπορεί να περιέχουν γράμματα, αριθμούς ή και κάτω παύλα (underscore, «_»). Δεν επιτρέπονται τα κενά διαστήματα και οι μεσαίες παύλες («-»). Οι διαφορετικοί τύποι δεδομένων/κλάσεις παρουσιάζονται στην Εικόνα 5, για περισσότερες πληροφορίες μεταβείτε [εδώ](#).



Εικόνα 5. Θεμελιώδεις τύποι δεδομένων MATLAB

Πίνακας 4. Συνήθειες τύποι(ή κλάσεις) δεδομένων

Τύπος	Περιγραφή
uint8	Unsigned Integers 8 bit, in the range [0, 255] (1 byte per element)
uint16	Unsigned Integers 16 bit, In the range [0, 65535] (2 bytes per element)
uint32	Unsigned Integers 32 bit, In the range [0, 4294967295] (4 bytes per element)
int8	Signed Integers 8 bit, in the range [-128,127] (1 byte per element)
int16	Signed Integers 16 bit, In the range [-32768,32767] (2 bytes per element).
int32	Signed Integers 32 bit, In the range [-2147483648,2147483647] (4 bytes per element)
single	Μονής ακρίβειας κινητής υποδιαστολής 64 bit, approximate range±1038 (4 bytes per element).
double	Διπλής ακρίβειας κινητής υποδιαστολής 64 bit, approximate range ±10308 (8 bytes per element)
character	ASCII 1 byte (στο MATLAB Unicode 2 bytes)
logical	1 byte, τιμές: true (1) ή false (0)

Είναι εφικτή η μετατροπή μεταξύ κλάσεων, με την γενική σύνταξη να είναι $B = \text{class_name}(A)$, δειτε τις εντολές στον Πίνακας 5. Για παράδειγμα, εάν ο πίνακας A είναι κλάσης `uint8`, μπορούμε να δημιουργήσουμε τον αντίστοιχο πίνακα διπλής ακρίβειας, B με την εντολή $B = \text{double}(A)$. Εάν ο πίνακας C είναι διπλός με τιμές στο εύρος $[0, 255]$, μπορεί να μετατραπεί σε `uint8` με την εντολή $D = \text{uint8}(C)$. Εάν ένας πίνακας έχει τιμές εκτός του εύρους $[0,255]$ και μετατραπεί σε `uint8` όπως προηγουμένως, το MATLAB μετατρέπει όλες τις αρνητικές τιμές σε 0 και όλες τις τιμές πάνω από 255 σε 255. Οι τιμές στο εύρος στρογγυλοποιούνται στον πλησιέστερο ακέραιο. Επομένως, πρέπει πρώτα να κλιμακώσετε τις τιμές ενός διπλού πίνακα, έτσι ώστε όλα τα στοιχεία του να είναι εντός του εύρους $[0, 255]$ πριν την μετατροπή σε `uint8`.

Πίνακας 5. Εντολές για την μετατροπή μεταξύ κλάσεων

Name	Converts Input to:	input
im2uint8	uint8	logical, uint8, uint16, int16, single, and double
im2uint16	uint16	logical, uint8, uint16, int16, single, and double
im2double	double	logical, uint8, uint16, int16, single, and double
im2single	single	logical, uint8, uint16, int16, single, and double
mat2gray	double in the range [0, 1]	logical, uint8, int8, uint16, int16, uint32, int32, single, and double
im2bw	logical	uint8, uint16, int16, single, and double

Οι μεταβλητές μπορεί να περιέχουν είτε απλές τιμές, είτε πίνακες με πολλές τιμές. Στις μεταβλητές ανατίθενται τιμές με τον τελεστή ανάθεσης «=». Μπορούμε να δίνουμε στις σταθερές και μεταβλητές ονόματα της επιλογής μας. Για παράδειγμα, έστω ότι θέλουμε να υπολογίσουμε το ακόλουθο: $\sin(60 \cdot \pi / 180)^2 + \cos(60 \cdot \pi / 180)^2$, 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 = 1
```

Τα σύμβολα theta, a και b αντιπροσωπεύουν σταθερές ή μεταβλητές ανάλογα με το αν 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 = 1.0472
>> sin(ans)
ans = 0.8660
```

2. Σειρές/Διανύσματα

Ως 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

>> decimals = [1.0 1.2 1.4 1.6 1.8 2.0]
decimals =
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 = 9
>> even(1)
ans = 2
```

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

```
>>length(even)
ans = 10
```

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

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

Ενας εναλλακτικός τρόπος εισαγωγής της παραπάνω σειράς είναι ο εξής:

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

>> even(2) = 4
even =
     2     4

>> even(3) = 6
even =
     2     4     6

.....

>> even(10) = 20
even =
     2     4     6     8     10     12     14     16     18     20
```

2.1. Πράξεις με σειρές

Έστω οι σειρές $A = [a_1 \ a_2 \ \dots \ a_n]$ και $B = [b_1 \ b_2 \ \dots \ b_n]$. Η πρόσθεση και η αφαίρεση των δύο σειρών ορίζονται ως εξής:

$$A + B = [a_1 + b_1, \ a_2 + b_2, \ \dots, \ a_n + b_n]$$

$$A - B = [a_1 - b_1, \ a_2 - b_2, \ \dots, \ a_n - b_n]$$

Για παράδειγμα:

```
>> odd +
even ans =
     3     7    11    15    19    23    27    31    35    39

>> even -
odd ans =
     1     1     1     1     1     1     1     1     1     1
```

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 =
```

```

1  3  5  7  9  11  13  15  17  19
>> even =
2:2:20 even =
2  4  6  8  10  12  14  16  18  20

```

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

```

>> natural =
1:10 natural =
1  2  3  4  5  6  7  8  9  10

>> inv_odd = 19:-
2:1 inv_odd =
19 17 15 13 11 9 7 5 3 1

>>decimals = 0:0.1:1
decimals =
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

```

Ορισμός πράξης πολλαπλασιασμού σειρών:

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

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

```

>> odd.
*even
ans =
2 12 30 56 90 132 182 240 306 380

```

Ορισμός διαίρεσης (από αριστερά και από δεξιά) σειρών:

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

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

Παραδείγματα:

```

>> odd./even
ans =

Columns 1 through 7
0.5000 0.7500 0.8333 0.8750 0.9000 0.9167 0.9286

Columns 8 through 10
0.9375 0.9444 0.9500

>> odd.\even ans
=

2.0000 1.3333 1.2000 1.1429 1.1111 1.0909 1.0769

1.0667 1.0588 1.0526

```

Ορισμός ύψωσης σε δύναμη:

$$A.^m = [a1m, a2m, \dots, anm]$$

Παράδειγμα:

```
>> natural.^2
ans =
    1    4    9   16   25   36   49   64   81
  100
```

ΣΗΜΕΙΩΣΗ:

Πολλές από τις ενσωματωμένες συναρτήσεις του 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 =
    0    0.1736    0.3420    0.5000    0.6428    0.7660
  0.8660    0.9397    0.9848    1.0000
```

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

2.2. 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 =
    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
>> N = [1:5]
N =
    1    2    3    4    5
```

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 =
1 2 3 4 5
>> Att = At'
Att =
1
2
3
4
5

```

2.3. Εσωτερικό γινόμενο διανυσμάτων

Εστω ένα διάνυσμα γραμμής $A = [a_1 \ a_2 \ \dots \ a_n]$ και ένα διάνυσμα στήλης $B = [b_1; b_2; \dots; b_n]$ με τον ίδιο αριθμό στοιχείων n . Το εσωτερικό γινόμενο $A * B$ των δύο διανυσμάτων είναι καθαρός αριθμός και δίνεται από την ακόλουθη εξίσωση:

$$A * B = a_1b_1 + a_2b_2 + \dots + a_nb_n$$

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

```

>> A' * B
ans = 106

```

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

```

>> N * N'
ans = 55

```

3. Πίνακες

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 =

```

1	2	3
4	5	6
7	8	9

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. Γενικά, έστω ότι έχουμε:

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}$$

τότε ,

- το **A(n,m)** μας δίνει το **a_{nm}** % A(Την συγκεκριμένη n-γραμμή, Την συγκεκριμένη m - στήλη)
- το **A(:,m)** μας δίνει την **m-στήλη του A** % A(Όλες τις γραμμές, Την συγκεκριμένη m -στήλη)
- το **A(n,:)** μας δίνει την **n-γραμμή του A** % A(Την συγκεκριμένη n-γραμμή, όλες τις στήλες)

Για παράδειγμα,

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

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

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

```
>> size(A)
ans =
    3 3
```

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 =

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

>> size(C)
ans =
    3    7
```

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 =

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

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 =

     1     2
     4     5
```

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 =

     1     2
     4     5
     7     8
    10    11
    13    14

>> C(1,:)
ans =
     1    23

>> C(:,1)
ans =

     1
     4
     7
    10
    13

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

     1     2     3
```



```

4      5      6
7      8      9

```

3.1. Πράξεις με πίνακες

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

```

>> S1 = C(1:3, :)
S1 =
     1     2     3
     4     5     6
     7     8     9

>> S2 = C(3:5, :)
S2 =
     7     8     9
    10    11    12
    13    14    15

>> S=S1+S2
S =
     8    10    12
    14    16    18
    20    22    24

>> D=S1-S2
D =
    -6    -6    -6
    -6    -6    -6
    -6    -6    -6

```

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 $m \times n$ και $n \times k$ αντίστοιχα. Τότε, οι διαστάσεις του νέου πίνακα P που αντιστοιχεί στο γινόμενο των δύο πινάκων θα είναι $m \times k$ και τα στοιχεία του θα δίνονται από την εξίσωση:

$$p_{ij} = \sum_{k=1}^m a_{ik} b_{kj}$$

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

```

>> P=S1*S2
P =
    66    72    78

```

```

156    171    186
246    270    294

```

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

```

>> S1
*S1
ans =
    30    36    42
    66    81    96

    102    126    150

```

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 =
    30    36    42
    66    81    96
    102   126   150

>> S1^3
ans =
    468    576    684
   1062   1305   1548
   1656   2034   2412

```

Όπως 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 =
    0    10    20
   30    40    50
   60    70    80

>> angle =
pi*angle/180 angle =
    0    0.1745    0.3491
   0.5236    0.6981    0.8727
   1.0472    1.2217    1.3963

>> sin(a
ngle) ans
=
    0    0.1736    0.3420
   0.5000    0.6428    0.7660
   0.8660    0.9397    0.9848

```

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

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

```
>> zeros
(2,2) ans
=
     0     0
     0     0
```

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 =
     1     2     3     4
     5     6     7     8
     9    10    11    12
```

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

```
m =
```

```
3
```

```
n =
```

```
4
```

```
>> ones(m, n)
```

```
ans =
```

```
     1     1     1     1
     1     1     1     1
     1     1     1     1
```

ή πιο απλά ακόμη:

```
>> ones(size
```

```
(A)) ans =
```

```
     1     1     1     1
     1     1     1     1
     1     1     1     1
```

Πίνακας 6. Εντολές για τον χειρισμό πινάκων. *Ίδιες σε Matlab και Octave.

rows (a)	Αριθμός γραμμών του a (return number of rows of a)
columns (a)	Αριθμός στηλών του a return number of columns of a
all (a)	Έλεγχος για μη μηδενικά στοιχεία (check if all elements of a nonzero)
any (a)	Έλεγχος για οποιοδήποτε μη μηδενικό (check if any elements of a nonzero)
find (a)	Επιστρέφει τους δείκτες των μη-μηδενικών στοιχείων (return indices of nonzero elements)
sort (a)	Βάζει σε τάξη τα στοιχεία κάθε στήλης του a (order elements in each column of a)
sum (a)	Προσθέτει τα στοιχεία στις στήλες του a (sum elements in columns of a)
prod (a)	Πολλαπλασιασμός των στοιχείων στις στήλες του a (product of elements in columns of a)
min (args)	Βρίσκει ελάχιστες τιμές (find minimum values) του args
max (args)	Βρίσκει μέγιστες τιμές (find maximum values) του args
rem (x, y)	Βρίσκει το υπόλοιπο της διαίρεσης of x/y (find remainder of x/y)
reshape (a, m, n)	Αναδιαμορφώνει a σε mxn πίνακα (reformat a to be m by n)
diag (v, k)	Δημιουργεί διαγώνιους πίνακες (create diagonal matrices)
linspace (b, l, n)	Δημιουργεί διάνυσμα με ίση απόσταση στοιχείων (create vector of linearly-spaced elements)
logspace (b, l, n)	Δημιουργεί διάνυσμα με λογαριθμική-κλίμακα στοιχεία (create vector of log-spaced elements)
eye (n, m)	Δημιουργεί nxm μοναδιαίο πίνακα (create n by m identity matrix)
ones (n, m)	Δημιουργεί nxm πίνακα με τιμές ένα (create n by m matrix of ones)
zeros (n, m)	Δημιουργεί nxm πίνακα με τιμές μηδέν (create n by m matrix of zeros)
rand (n, m)	Δημιουργεί nxm πίνακα με τυχαίες τιμές από ομοιόμορφη κατανομή (create n by m matrix of random values from uniform distribution)
length	Μήκος του διανύσματος
size	Μέγεθος πίνακα
median	Διάμεσος
mean	Μέσος όρος
std	Τυπική απόκλιση

4. Loops-Control statements

- **for loop**

Οι βρόχοι for έχουν την εξής δομή:

```
for index = initial value (: step) : final value
    statements
end
```

Οι λέξεις “for” και “end” χρησιμοποιούνται στην αρχή και στο τέλος του βρόχου, ο μετρητής index παίρνει τις τιμές από initial value μέχρι final value με βήμα step, και οι εντολές (statements) εκτελούνται για όλες τις τιμές του μετρητή index. Αν παραλείψουμε το βήμα, τότε χρησιμοποιείται το 1 σαν βήμα.

Παράδειγμα

```

j = 0;
for i=1:10           %με βήμα 1
    j = j + 1;
end

j                   %για να δούμε την τιμή στο j

k = 0;
for i=0:5:20       %με βήμα 5
    for j=1:3      %διπλό loop, το j με βήμα 1
        k = i*5+j;
    end
end

k
για λεπτομέρειες πληκτρολογήστε:
help for

```

- **while loop**

Οι βρόχοι while είναι της μορφής:

```

while relation
    statements
end

```

Οι λέξεις “while” και “end” χρησιμοποιούνται στην αρχή και στο τέλος του βρόχου. Η ακολουθία εντολών «statements» εκτελούνται εφόσον η συνθήκη relation ικανοποιείται (δηλ. είναι αληθής) και σταματούν όταν αυτή παύει να ισχύει. Για να γράψουμε τη συνθήκη relation χρησιμοποιούμε τους σχεσιακούς και λογικούς τελεστές

```

j = 5;
while j>0           % while (συνθήκη = true) do
    disp(['j = ', num2str(j)]);
    j = j - 1;
end
j

```

για λεπτομέρειες πληκτρολογήστε:
help while

- **if ... then ... else**

Η εντολή if μας επιτρέπει να ελέγξουμε αν μια (ή περισσότερες) συνθήκες ισχύουν και να εκτελέσουμε σε κάθε περίπτωση την επιθυμητή ακολουθία εντολών και πράξεων. Η εντολή έχει την γενική μορφή:

```

if relation
    statement(s)
end

```

ή πιο σύνθετα:

```

if relation_1
    statement(s)
elseif relation_2
    statement(s)
else
    statement(s)
end

```

Οι συνθήκες ελέγχονται με τη χρήση σχεσιακών και λογικών τελεστών. Σημειώνουμε επίσης ότι η εντολή `elseif` γράφεται σαν μια λέξη (δεν πρέπει να υπάρχει κενό μεταξύ του `else` και του `if`).

```

j = 5;
k = -4;
if ((j~=k+1) & (k>=0)) | j==abs(k)
    k = k+j;
elseif j==k+1
    k = k+2*j;
else
    k = 0;
end
k

```

για λεπτομέρειες πληκτρολογήστε:

```

help if
help elseif

```

5. m files-Script-Functions

Τα m-files είναι τα αρχεία που δημιουργούνται για την ανάπτυξη μεγάλων και συχνά πολύπλοκων ακολουθιών εντολών. Διακρίνονται σε `script` και `function` αρχεία. Συγκεκριμένα:

- Αρχεία `script` ή αρχεία εντολών (`script m-files` or `command files`) τα οποία δεν έχουν ορίσματα εισόδου και εξόδου αλλά εκτελούν μια ακολουθία εντολών σε μεταβλητές του χώρου εργασίας.
- Αρχεία συναρτήσεων (`function m-files`) τα οποία περιλαμβάνουν μια γραμμή ορισμού συνάρτησης, δέχονται ορίσματα εισόδου και επιστρέφουν μεταβλητές εξόδου, και των οποίων οι εσωτερικές μεταβλητές είναι τοπικές (εκτός αν δηλωθούν ως ολικές με την εντολή `global`).

Γενικά για την ονομασία τους ακολουθούμε τους κανόνες ονοματολογίας που ισχύουν και για τις μεταβλητές:

- Το όνομα αρχίζει με γράμμα (του αγγλικού αλφαβήτου).
- Το όνομα περιέχει μόνο γράμματα, αριθμούς και υποπαύλες (`underscore`).
- Δεν χρησιμοποιούνται ονόματα που έχουν δεσμευτεί (π.χ. συναρτήσεις βιβλιοθήκης και εργαλειοθηκών).
- Προτιμώνται μικρά ονόματα για πρακτικούς λόγους αν και δεν υπάρχει περιορισμός στο μήκος των ονομάτων.
- τόσο η συνάρτηση όπως και τα αρχεία `script` καλούνται με το όνομά τους από το `command window`.

Τα αρχεία συναρτήσεων (function m-files) περιέχουν μια ολοκληρωμένη ακολουθία εντολών με μεταβλητές εισόδου, `input1`, `input2`, με την οποία υπολογίζονται νέες μεταβλητές εξόδου `output1`, `output2`, Η δομή της συνάρτησης έχει την εξής μορφή:

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

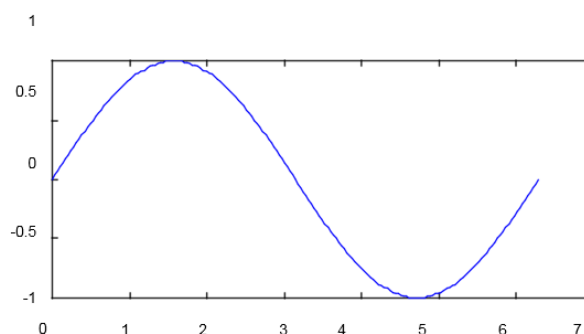
6. Γραφικές Παραστάσεις

Πίνακας 7. Βασικές συναρτήσεις για γραφικές παραστάσεις

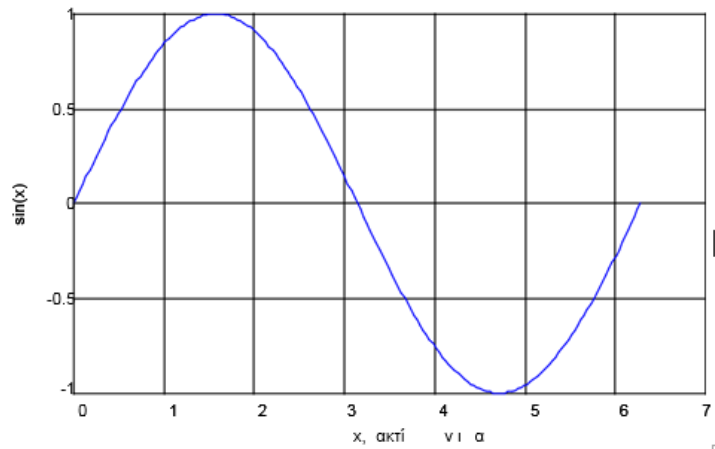
plot ()	Η συνάρτηση plot χρησιμοποιείται για την κατασκευή του γραφήματος μιας επίπεδης καμπύλης τα σημεία της οποίας είναι αποθηκευμένα στα ισομήκη διανύσματα x και y .
subplot()	Η συνάρτηση subplot μας επιτρέπει να βάλουμε πολλά γραφήματα στο ίδιο παράθυρο γραφικών
figure ()	Άνοιγμα (άλλου) παραθύρου γραφικών
title()	Προσθήκη τίτλου
xlabel ()	Προσθήκη ετικέτας στον οριζόντιο άξονα
ylabel()	Προσθήκη ετικέτας στον κατακόρυφο άξονα
axis	Εντολή για τους άξονες <i>equal</i> , <i>square</i> , <i>tight</i> κ.α
legend()	Προσθήκη λεζάντας
text()	Προσθήκη κειμένου στη θέση (x_i, y_i)
grid on	Δημιουργία πλέγματος
grid off	Αφαιρεί το πλέγμα
hold on	Πάγωμα του τρέχοντος παραθύρου γραφικών για το σχεδιασμό και άλλων καμπυλών

Έστω ότι θέλουμε να κάνουμε τη γραφική παράσταση της ημιτονοειδούς συνάρτησης στο διάστημα $[0, 2\pi]$. Η βασική συνάρτηση για δισδιάστατες απεικονίσεις είναι η `plot` (για λεπτομέρειες πληκτρολογήστε `help plot`). Άλλες χρήσιμες συναρτήσεις είναι οι `xlabel`, `ylabel` για την εισαγωγή κειμένου στις γραφικές παραστάσεις.

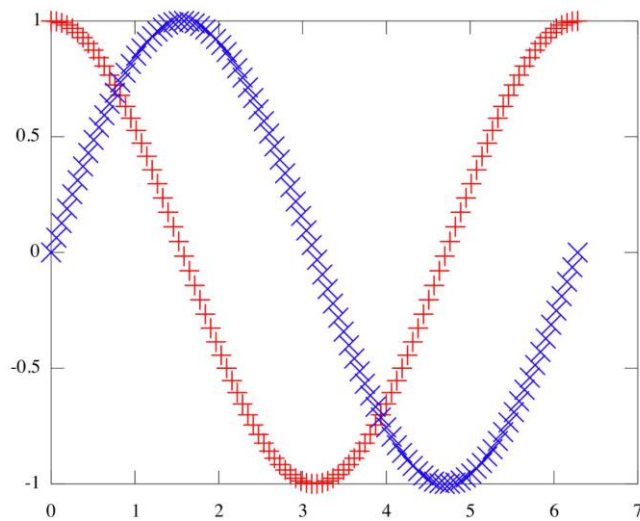
```
>> x = 0: pi/90: 2*pi;
>> y = sin(x);
>> plot(x,y)
```



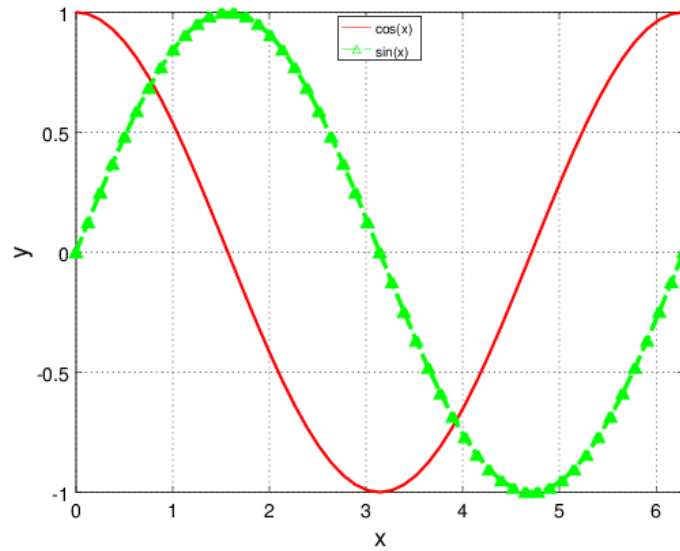
```
>> grid
>> xlabel('x, ακτίνια')
>> ylabel('sin(x)')
```



```
>> x = linspace(0,2*pi,100);
>> plot(x,cos(x),'r+',x,sin(x),'bx');
```



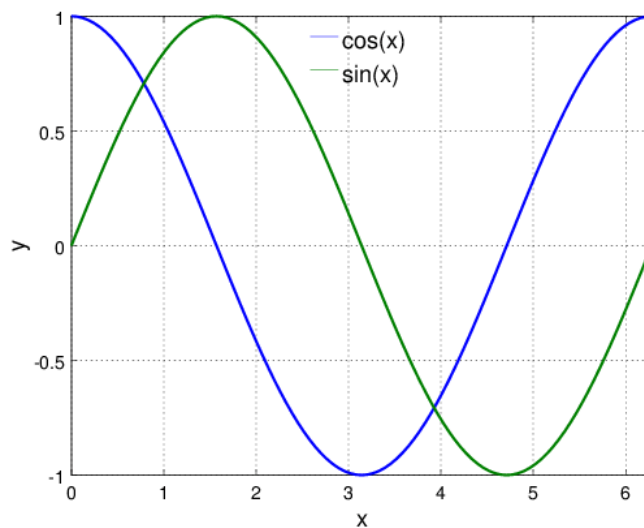
```
>> clear; cla; clf;
>> x = linspace(0, 2*pi, 51)';
>> y1 = cos(x);
>> y2 = sin(x);
>> plot(x, y1, 'r', 'LineWidth', 2, x, y2, 'g--^',
>>      'LineWidth', 2);
>> grid on;
>> xlabel("x", 'FontWeight', 'bold', 'FontSize', 18);
>> ylabel("y", 'FontWeight', 'bold', 'FontSize', 18);
>> axis([0 2*pi -1 1]);
>> legend("cos(x)", "sin(x)", 'Location', 'North');
>> set(gca, 'FontSize', 14);
```

```

>> clear; cla; clf;
>> x = (0:pi/50:2*pi)';
>> y = [cos(x) sin(x)];
>> plot(x, y, 'Linewidth', 2);
>> grid on;
>> xlabel("x", 'FontWeight', 'bold', 'FontSize', 18);
>> ylabel("y", 'FontWeight', 'bold', 'FontSize', 18);
>> axis([0 2*pi -1 1]);
>> h = legend("cos(x)", "sin(x)", 'Location', 'North');
>> legend(h, 'boxoff');
>> set(h, 'FontSize', 18);
>> set(gca, 'FontSize', 14)

```



Αναφορές

- [1] Δ. Α. Π. Λερός, «ΣΗΜΕΙΩΣΕΙΣ ΜΑΘΗΜΑΤΟΣ:ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΠΡΟΣΟΜΟΙΩΣΗ ΜΕ Η/Υ [OCTAVE/MATLAB – DIGITAL SIGNAL PROCESSING – QUEUES]» Τμήμα Μηχανικών: Ειδικότητα (Τ-Η), 2009.
- [2] Ι. Καλατζής, «Συνοπτικός οδηγός MATLAB / OCTAVE,» 2018.
- [3] Μ. Karampidis, Κ. Kostas «Octave/Matlab Tutorial,» 2019.
- [4] Αθανάσιος Σταυρακούδης, Εκπαιδευτικό υλικό,
<http://stavrakoudis.econ.uoi.gr/stavrakoudis/?iid=253> 2019.