





# Επεξεργασία Εικόνας και Βίντεο

1° Φυλλάδιο Εργαστηρίου

Εισαγωγή – στην ΜΑΤLAB

Περιεχόμενα

- 1. Το περιβάλλον της MATLAB
- 2. Σύνταξη Εντολών
- 3. Ασκήσεις

Νικόλαος Γιαννακέας





# ΕΙΣΑΓΩΓΗ

Το Matlab είναι ένα πρόγραμμα υπολογιστών για ανθρώπους που χρησιμοποιούν αριθμητικούς υπολογισμούς, ειδικά στη γραμμική άλγεβρα (πίνακες). Ονομάστηκε έτσι από τα αρχικά του "Εργαστηρίου Πινάκων" (MATrixLABoratory). Από τότε έχει εξελιχθεί σ' ένα ισχυρότατο εργαλείο προγραμματισμού, στην έρευνα, στην επιστήμη των μηχανικών, στην επεξεργασία σήματος και εικόνας, στις επικοινωνίες και σε πολλές άλλες επιστημονικές περιοχές. Στο δυναμικό του Matlab συμπεριλαμβάνονται μοντέρνοι αλγόριθμοι, δυνατότητες χειρισμού τεράστιων ποσοτήτων δεδομένων, και ισχυρά προγραμματιστικά εργαλεία.

## 1. Το περιβάλλον του Matlab

To Matlab παρέχει ένα πλήρες παραθυρικό περιβάλλον. Συνοψίζοντας τα ποιο σημαντικά σημεία το παραθυρικό περιβάλλον από προεπιλογή περιέχει:

## A. Current Directory

Στην βασική εργαλειοθήκη υπάρχει το πεδίο της τρέχουσας διεύθυνσης (current directory). Σε αυτό το μονοπάτι του υπολογιστή θα πρέπει βρίσκονται όλα τα αρχεία τα οποία μπορεί να καλέσει ο κώδικας που θα δημιουργήσουμε και όλες οι συναρτήσεις που έχουμε δημιουργήσει. Επίσης σε αυτή την διεύθυνση θα δημιουργηθεί οποιοδήποτε αρχείο εξαχθεί και αποθηκευτεί από τον κώδικα που εκτελούμε, εάν βεβαίως δεν καθοριστεί ρητά άλλη διαδρομή.

## Β. Παράθυρα

Κατά την εκκίνηση του προγράμματος

- 1. Στη δεξιά πλευρά της οθόνης υπάρχει το **Command Window** (παράθυρο εντολών) στο οποίο πληκτρολογούμε τις εντολές. Στο σημείο αυτό εμφανίζονται και τα μηνύματα λάθος εφόσον υπάρχουν στον κώδικά που δημιουργήσαμε.
- 2. Το **Workspace** είναι ο χώρος όπου εμφανίζονται οι μεταβλητές που χρησιμοποιούμε στο πρόγραμμα.
- 3. Στην κάτω αριστερή πλευρά της οθόνης υπάρχει το Command History (ιστορικό εντολών), όπου φαίνονται οι εντολές που έχουμε ήδη δώσει στο MATLAB.
- 4. Στη θέση του Command History και με τη βοήθεια πάλι της αντίστοιχης ετικέτας που βρίσκεται από κάτω μπορούμε να εμφανίσουμε τα περιεχόμενα του φακέλου των Windows στον οποίο βρισκόμαστε αυτή τη στιγμή (Current Directory).

Υπάρχει δυνατότητα να αλλάξει ο τρόπος που εμφανίζεται η επιφάνεια εργασίας, ανοίγοντας, κλείνοντας, μετακινώντας και αλλάζοντας το μέγεθος των εργαλείων που περιέχει. Μπορείτε να επιλέξετε ορισμένα χαρακτηριστικά για τα εργαλεία που χρησιμοποιείται, με την επιλογή Preferences από το μενού File. Παραδείγματος χάριν, μπορείτε να καθορίσετε τη γραμματοσειρά του κειμένου που γράφετε στο Command Window (παράθυρο εντολών).



### Γ. Βοήθεια

Το Matlab διαθέτει ένα ισχυρό εργαλείο βοήθειας προς το χρήστη, το οποίο παρέχει όλη την απαραίτητη πληροφορία για την σωστή χρήση του. Το περιβάλλον βοήθειας μπορεί κανείς να το προσπελάσει με 2 τρόπους:

- 1. Επιλέγοντας Help -> Product Help,
- 2. Πληκτρολογώντας στο command window helpbrowser







## 2. Σύνταξη Εντολών

## 2.1 Τελεστές

Η πιο απλή χρήση του Matlab συνίσταται στον υπολογισμό μαθηματικών πράξεων μεταξύ απλών αριθμών. Αυτό γίνεται στο **Command Window (γραμμή εντολών)** με τη χρήση των εξής μαθηματικών τελεστών:

Πρόσθεση	+
Αφαίρεση	-
Πολλαπλασιασμός	*
Διαίρεση	/
Διαίρεση προς τα αριστερά	\ (Μια σύνθετη πράξη που χρησιμοποιείται στους πίνακες και στη γραμμική
άλγεβρα του MATLAB )	
Ύψωση σε δύναμη	$\wedge$

Για να υπολογίσουμε κάποια πράξη, απλά την πληκτρολογούμε στο **Command Window** και στη συνέχεια πατάμε **Enter**. Παρατηρούμε ότι σε κάθε περίπτωση ο υπολογιστής μας δείχνει το αποτέλεσμα της πράξης μας αφού πατήσουμε το **Enter**, αποθηκεύοντάς το παράλληλα σε μια μεταβλητή που ονομάζεται **ans**. Μπορούμε φυσικά να δώσουμε στην ίδια γραμμή μια σειρά πράξεων όπως π.χ: 3^2/6+5. Σε αυτήν την περίπτωση το πρόγραμμα θα εκτελέσει τις πράξεις από αριστερά προς τα δεξιά και δίνοντας προτεραιότητα κατά σειρά στην ύψωση σε δύναμη (^), στους πολλαπλασιασμούς και διαιρέσεις (\*,/) και τέλος στις προσθαφαιρέσεις (+,-). Για να αλλάξουμε τη σειρά των πράξεων χρησιμοποιώντας παρενθέσεις.

**ΠΡΟΣΟΧΗ:** Το πρόγραμμα αποθηκεύει στη μεταβλητή **ans** μόνο το αποτέλεσμα της τελευταίας πράξης. Αν εμείς θέλουμε να αποθηκεύσουμε τα αποτελέσματα των πράξεων μας για μελλοντική χρήση θα πρέπει να χρησιμοποιούμε δικές μας μεταβλητές.

## 2.2 Μεταβλητές

Για να αποθηκεύσουμε το αποτέλεσμα μιας πράξης σε μια μεταβλητή, θα πρέπει να πληκτρολογήσουμε το όνομα της μεταβλητής, το σύμβολο (=) και στην συνέχεια την πράξη μας. Το όνομα μιας μεταβλητής μπορεί να περιέχει μόνο λατινικούς χαρακτήρες, πρέπει να αρχίζει οπωσδήποτε από γράμμα, και μπορεί να αποτελείται από οποιονδήποτε συνδυασμό γραμμάτων, αριθμών και της «κάτω παύλας» (\_). Το συνολικό όνομα μιας μεταβλητής μπορεί να έχει μέχρι 31 χαρακτήρες. Τις μεταβλητές μας μπορούμε να τις δούμε στο **Workspace** ή εναλλακτικά στο **Command Window** πληκτρολογώντας την εντολή **who**.

🖊 Workspace 📃 🗖					
ile <u>E</u> dit ⊻iew V	Ve <u>b</u> <u>W</u> indo	w <u>H</u> elp			
🖙 🛃 📓 🖶 Stade Base 💌					
Name	Size	Bytes	Class		
a	1x10	80	double array		
Щ с	1×1	16	double array (complex)		
e e	1×1	4	cell array		
🗮 g	1x10	80	double array (global)		
i i	1x10	10	int8 array		
<b>===</b> 1	1x10	80	double array (logical)		
abo m	1x6	12	char array		
👰 n	1×1	822	inline object		
q 🔀	1x10	164	sparse array		
🛨 s	1x1	406	struct array		
H u	1x10	40	uint32 array		





Αφού αποθηκεύσουμε κάποια μεταβλητή μας, μπορούμε στη συνέχεια να δούμε την τιμή της κάνοντας ένα διπλό κλικ με το ποντίκι στην αντίστοιχη καταχώρηση στο Workspace, ή εναλλακτικά πληκτρολογώντας το όνομά της και πατώντας Enter στο Command Window. Για να διαγράψουμε μια μεταβλητή μας, μπορούμε να κάνουμε δεξί κλικ με το ποντίκι στην αντίστοιχη καταχώρηση στο Workspace και να επιλέξουμε την εντολή delete selection ή εναλλακτικά να πληκτρολογήσουμε στο Command Window την εντολή clear όνομα μεταβλητής, χρησιμοποιώντας το όνομα της μεταβλητής που θέλουμε να διαγράψουμε. Αντίστοιχα, για να διαγράψουμε όλες τις μεταβλητές μας μπορούμε να χρησιμοποιήσουμε την εντολή clear.

### 2.3 Πίνακες

Η διαχείριση πινάκων είναι από τα βασικά σημεία στα οποία το MATLAB υπερτερεί σε σχέση με άλλες ερευνητικές πλατφόρμες. Η δήλωση των πινάκων εφόσον είναι γνωστά τα στοιχεία τους γίνεται με τον παρακάτω τρόπο:

```
A = [1 1 3 4; %tyxaios pinakas 4x4
    5 6 7 8;
    1 2 3 4;
    5 6 7 8];
B = [0 1 0 1; %tyxaios pinakas 4x4
    1 0 1 2;
    2 1 0 1;
    0 1 0 1];
```

Ένα στοιχείο ενός πίνακα μπορεί κάποιος να το προσπελάσει όπως φαίνεται παρακάτω για το στοιχείο του πίνακα Α που βρίσκεται στην δεύτερη γραμμή και πρώτη στήλη του:

A21 = A(2,1);

Η προσπέλαση όλων στοιχείων ενός πίνακα γίνεται με ένα διπλό φωλιασμένο βρόχο for όπως παρακάτω:

```
for i=1:size(A,1)
    for j=1:size(A,2)
        A(i,j);
    end
end
```

Κάνοντας χρήση του σημείου στίξης «:» μπορούμε είτε να δηλώσουμε την προσπέλαση όλων των στοιχείων ενός πίνακα σε κάποια διάσταση του

A\_prwth\_sthlh = A(:,1);





είτε την προσπέλαση ολόκληρου υποπίνακα κάποιου πίνακα:

Amisos = A(2:3, 2:3);

Πίνακες μεγαλύτερης διάστασης ορίζονται ως εξής:

A\_3D(:,:,1) = A; A\_3D(:,:,2) = B;

#### 2.4 Διανύσματα/Μονοδιάστατα σήματα

Δεδομένου ότι τα διανύσματα δεν είναι τίποτα άλλο παρά πίνακες μίας διάστασης, κατ' αντιστοιχία η δήλωση γνωστών διανυσμάτων γίνεται ως εξής:

C = [1 2 3 4]; %tyxaio dianysma 1x4 D = [5;6;7;8]; %tyxaio dianysma 4x1

Η δημιουργία ενός μονοδιάστατου σήματος το οποίο παράγεται από κάποια συγκεκριμένη συνάρτηση πραγματοποιείται με διακριτές τιμές στον οριζόντιο άξονα τις οποίες δημιουργούμε εύκολα σαν να ήταν οι θέσεις τιμών δειγματοληψίας. Για παράδειγμα ένα σήμα/διάνυσμα ημιτόνου μπορεί να δημιουργηθεί ως εξής:

```
t=[-1000:1:1000];
sint = sin(t);
figure, plot(t,sint)
```

(με την εντολή plot δημιουργούμε την γραφική παράσταση των τιμών του σήματος)

Τέλος υπάρχουν εντολές για την δημιουργία «τυποποιημένων» πινάκων ή διανυσμάτων όπως ο μηδενικός πίνακας, ο πίνακας με όλα του τα στοιχεία «1» ή ο πίνακας με τυχαίες τιμές παραγόμενες από γεννήτρια τυχαίων τιμών.

K1=zeros(1,7) K2=ones(5,5) K3=randn(7,7,7)





### 2.5 Δημιουργία Συναρτήσεων

Στο πλαίσιο του δομημένου προγραμματισμού είναι εφικτή η δημιουργία συναρτήσεων τις οποίες μπορούμε να καλούμε όποτε το επιθυμούμε αρκεί το αρχείο της (.m file) να βρίσκεται στην τρέχουσα διεύθυνση που βρισκόμαστε (Current directory). Για την δημιουργία μια συνάρτηση: 1) Ανοίγουμε ένα editor (Desktop -> tick στο "editor") 2) Στην πρώτη γραμμή του editor γράφουμε την εξής σύνταξη:

function [metabliti\_episttrofis1, metabliti\_episttrofis2, ... ] = onoma\_synartisis(orisma1, orisma2, ..., orismaN)

3) Γράφουμε τον κώδικα που θέλουμε να εκτελεί η συνάρτηση. Προσοχή! Όλα τα ορίσματα θεωρούνται γνωστά και όλες οι «metabliti\_episttrofis» πρέπει να υπολογίζονται ώστε να επιστραφούν από την συνάρτηση
4) Καλούμε την συνάρτηση είτε από την γραμμή εντολών είτε από κάποια άλλη συνάρτηση.

[metabliti\_episttrofis1, metabliti\_episttrofis2, ...] = onoma\_synartisis(orisma1, orisma2, ..., orismaN)





# ΠΕΙΡΑΜΑΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ

## Άσκηση 1

Δημιουργήστε μια συνάρτηση η οποία να δέχεται έναν οποιοδήποτε πίνακα Α και να επιστρέφει το είδωλο του πίνακα ως προς τον οριζόντιο άξονα. Εφαρμόστε την συνάρτηση που υλοποιήσατε στον παρακάτω πίνακα στην εικόνα του cameraman.

Υπόδειζη: Για την εισαγωγή της εικόνας χρησιμοποιήστε την εντολή: I = imread('cameraman.tif') Για την προβολή της εικόνας: figure, imshow(uint8(I))

## Άσκηση 2

Δημιουργήστε μια συνάρτηση η οποία να δέχεται έναν περιττό αριθμό m και να επιστρέφει έναν 3-διάστατο πίνακα mxmxm τυχαίων τιμών και την τιμή του σημείου του που βρίσκεται στο κέντρο του. Εφαρμόστε αμυντικό προγραμματισμό για την περίπτωση που κάποιος χρήστης εισάγει άρτια τιμή για το m. Στην περίπτωση όπου το m είναι άρτιο η συνάρτηση να επιστρέφει ένα μήνυμα που να προτρέπει τον χρήστη να εισάγει περιττό αριθμό, και ένα μηδενικό πίνακα και μια μηδενική τιμή.





# ΛΥΣΕΙΣ

Άσκηση 1

```
function mirrored = mirroring(I)
```

```
mirrored = zeros(size(I,1),size(I,2));
count = 1;
for i = size(I,2):-1:1
    mirrored(:,count) = I(:,i);
    count = count + 1;
end
```

## Άσκηση 2

```
function [matrix, center_value] = rand_3d(m)

if mod(m,2)==0
    disp('Please insert an odd value');
    matrix = 0;
    center_value = 0;
    return
else
    matrix = rand(m,m,m);
    center_value = matrix(m/2+0.5,m/2+0.5,m/2+0.5);
end
```