

Επεξεργασία Εικόνας & Βίντεο

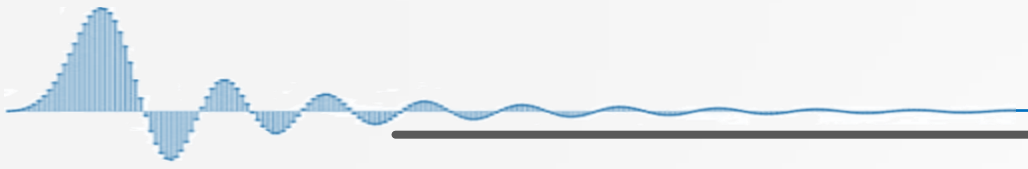


ο1. Εισαγωγή στην Matlab

Εισηγητής: Νικόλαος Γιαννακέας

Επίκουρος Καθηγητής, Σημάτων & Συστημάτων



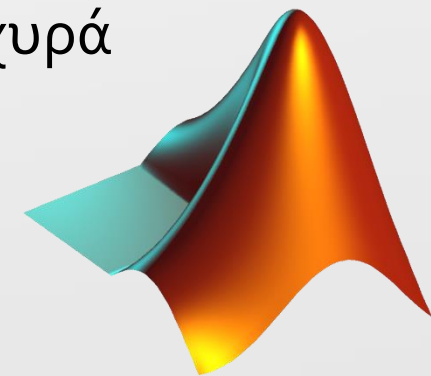


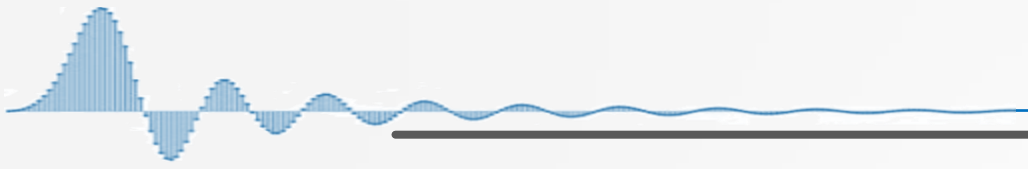
Matlab

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.



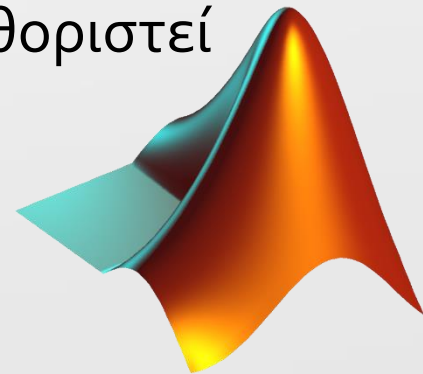


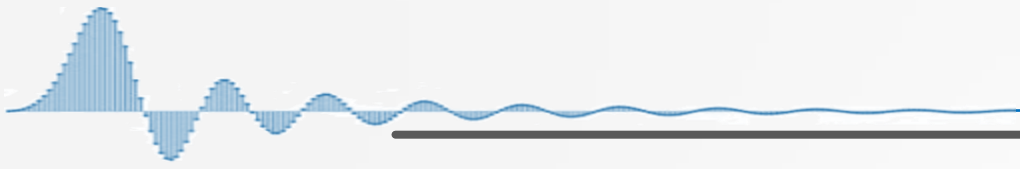
Το περιβάλλον του Matlab

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

A. Current Directory - Τρέχων κατάλογος

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.



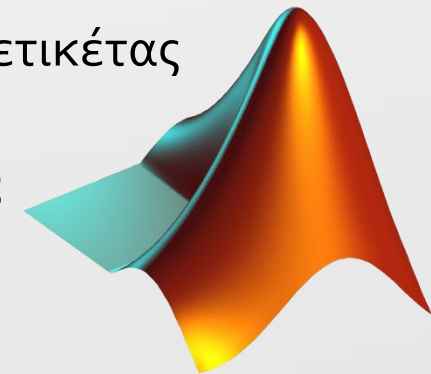


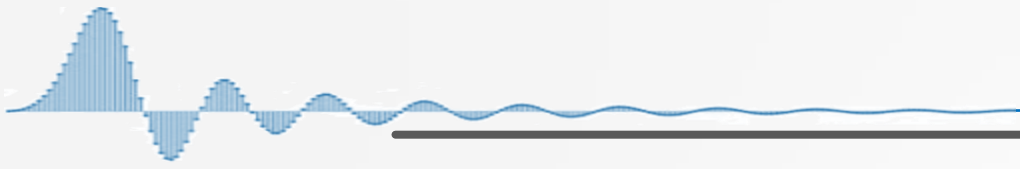
Το περιβάλλον του Matlab

Β. Παράθυρα

Κατά την εκκίνηση του προγράμματος

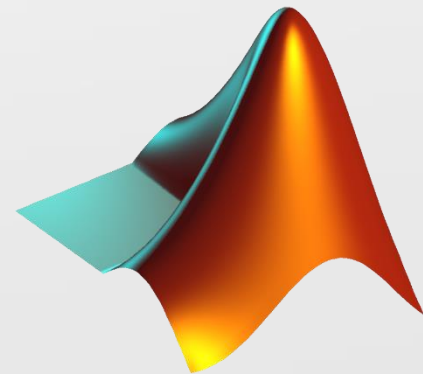
1. Στη δεξιά πλευρά της οθόνης υπάρχει το **Command Window** (παράθυρο εντολών) στο οποίο πληκτρολογούμε τις εντολές. Στο σημείο αυτό εμφανίζονται και τα μηνύματα λάθος εφόσον υπάρχουν στον κώδικά που δημιουργήσαμε.
2. Το **Workspace** είναι ο χώρος όπου εμφανίζονται οι μεταβλητές που χρησιμοποιούμε 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**).

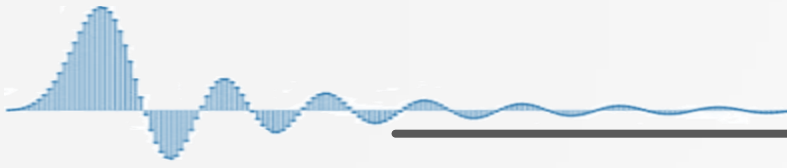




Το περιβάλλον του Matlab

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).





Το περιβάλλον του Matlab

The screenshot shows the MATLAB R2019b environment with several key components highlighted by blue arrows and pink text labels:

- Current Folder – Τρέχων κατάλογος:** Points to the file browser on the left side of the interface.
- Command Window – Παράθυρο Εντολών:** Points to the central area where MATLAB commands are entered and executed.
- Workspace – Χώρος Μεταβλητών:** Points to the top-right pane showing the current workspace variables.
- Command History – Ιστορικό Εντολών:** Points to the bottom-right pane showing a list of previously executed commands.

Name	Value

Command History

```
%-- 7/3/2022 1:14 μμ --%
```

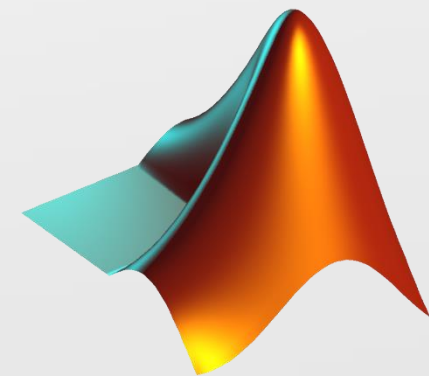


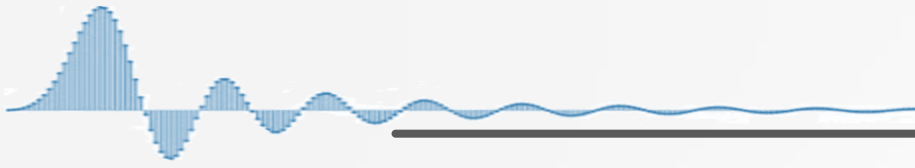
Το περιβάλλον του Matlab

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:

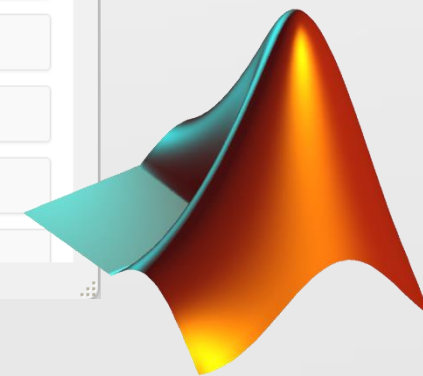
1. Επιλέγοντας **Help -> Product Help**,
2. Πληκτρολογώντας στο command window *helpbrowser*

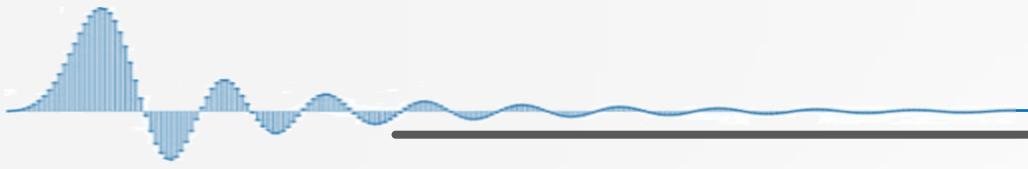




Το περιβάλλον του Matlab

The screenshot shows the MATLAB Documentation website. At the top, there is a navigation bar with tabs for 'All', 'Examples', 'Functions', 'Blocks', and 'Apps'. A search bar labeled 'Search Documentation' is on the right. On the left, a 'CONTENTS' sidebar lists various categories such as MATLAB, Simulink, and various toolboxes. The main content area is titled 'Release Notes' and features four large blue buttons: 'Explore MATLAB', 'Explore Simulink', 'Explore Polyspace', and 'View Installation Help'. Below this, an 'Applications' section lists several application areas with expandable arrows, including 'Math, Statistics, and Optimization', 'Data Science and Deep Learning', 'Signal Processing and Wireless Communications', 'Control Systems', 'Image Processing and Computer Vision', 'Event-Based Modeling', 'Physical Modeling', 'Robotics and Autonomous Systems', 'Real-Time Simulation and Testing', and 'Code Generation'. An 'Edit Preferences' link is visible in the top right of the main content area.



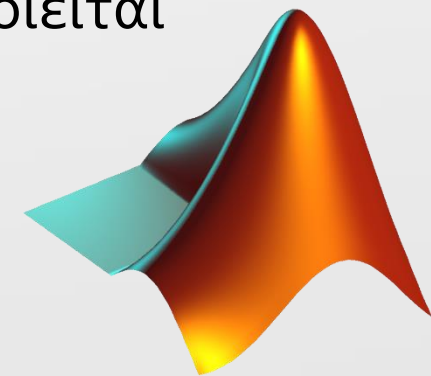


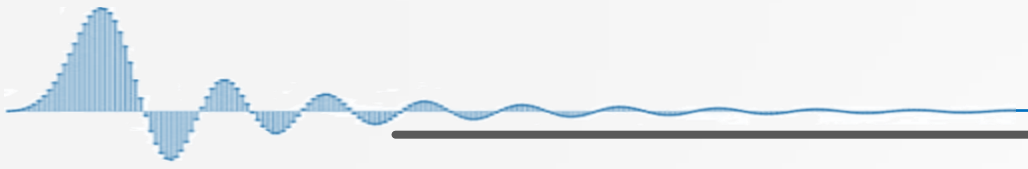
Σύνταξη Εντολών

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:

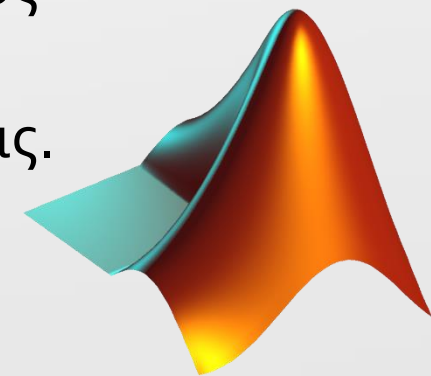
Πρόσθεση	+
Αφαίρεση	-
Πολλαπλασιασμός	*
Διαίρεση	/
Διαίρεση προς τα αριστερά	\ (Μια σύνθετη πράξη που χρησιμοποιείται στους πίνακες και στη γραμμική άλγεβρα του MATLAB)
Ύψωση σε δύναμη	^





Σύνταξη Εντολών

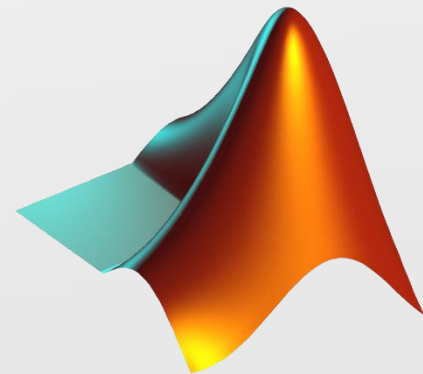
Για να υπολογίσουμε κάποια πράξη, απλά την πληκτρολογούμε στο **Command Window** και στη συνέχεια πατάμε **Enter**. Παρατηρούμε ότι σε κάθε περίπτωση ο υπολογιστής μας δείχνει το αποτέλεσμα της πράξης μας αφού πατήσουμε το Enter, αποθηκεύοντάς το παράλληλα σε μια μεταβλητή που ονομάζεται **ans**. Μπορούμε 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.

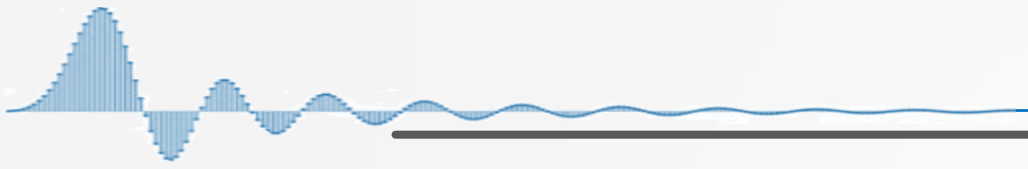




Σύνταξη Εντολών

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.



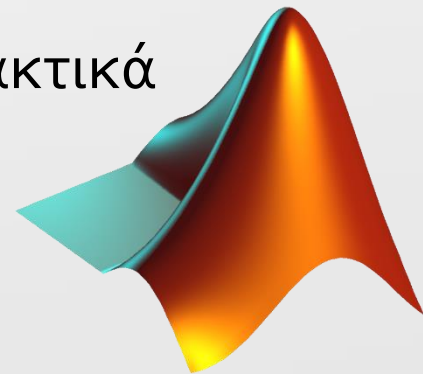


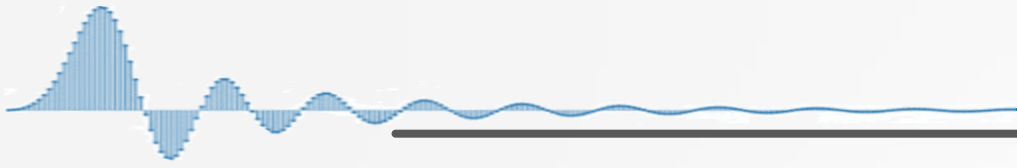
Σύνταξη Εντολών

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.

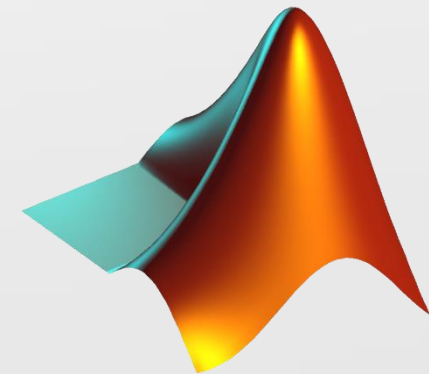
Τις μεταβλητές μας μπορούμε να τις δούμε στο Workspace ή εναλλακτικά στο Command Window πληκτρολογώντας την εντολή who.

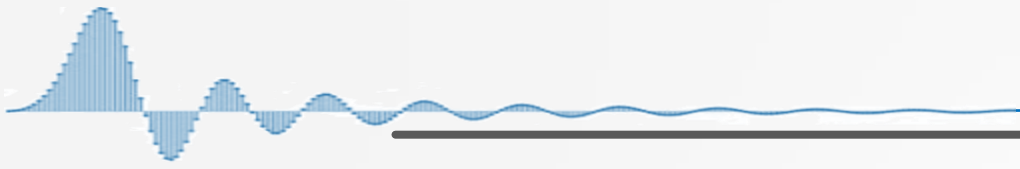




Σύνταξη Εντολών

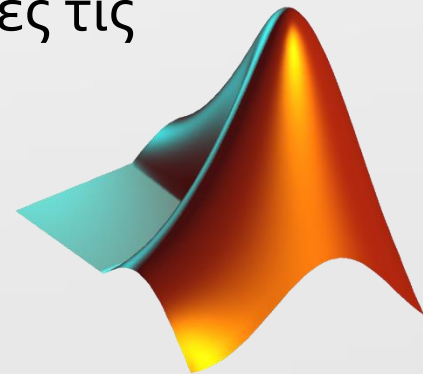
Name ▲	Value
a	30
ans	6.5000
b	50

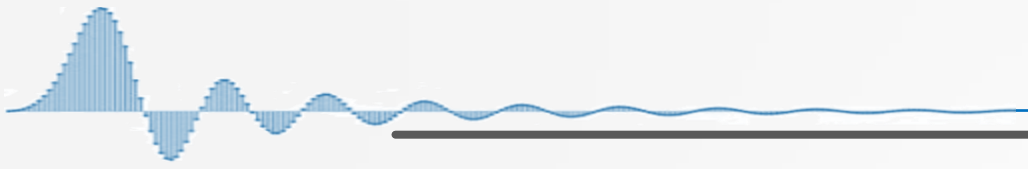




Σύνταξη Εντολών

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.



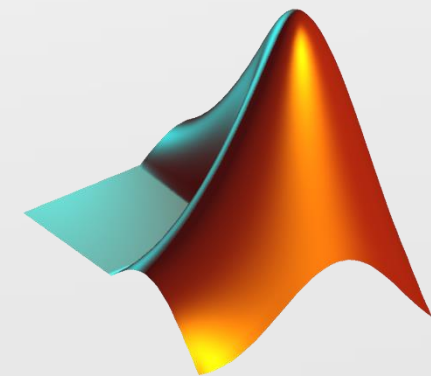


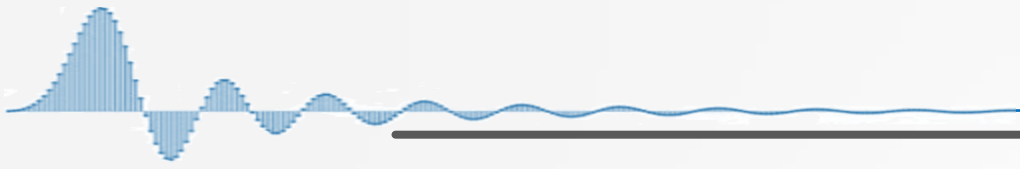
Σύνταξη Εντολών

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];
```





Σύνταξη Εντολών

Πίνακες

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

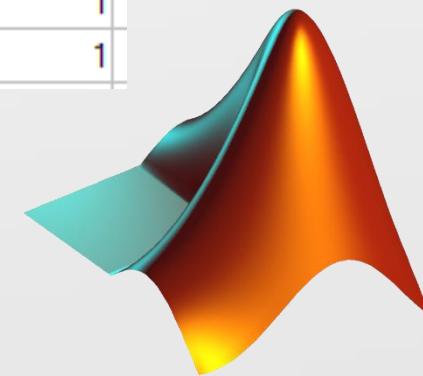


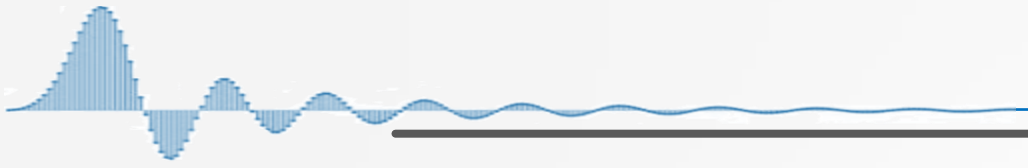
	1	2	3	4
1	1	1	3	4
2	5	6	7	8
3	1	2	3	4
4	5	6	7	8

```
B = [0 1 0 1; %tyxaios pinakas 4x4  
1 0 1 2;  
2 1 0 1;  
0 1 0 1];
```



	1	2	3	4
1	0	1	0	1
2	1	0	1	2
3	2	1	0	1
4	0	1	0	1





Σύνταξη Εντολών

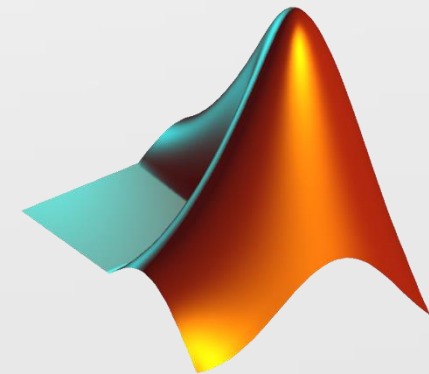
Ένα στοιχείο ενός πίνακα μπορεί κάποιος να το προσπελάσει όπως φαίνεται παρακάτω για το στοιχείο του πίνακα A που βρίσκεται στην δεύτερη γραμμή και πρώτη στήλη του:

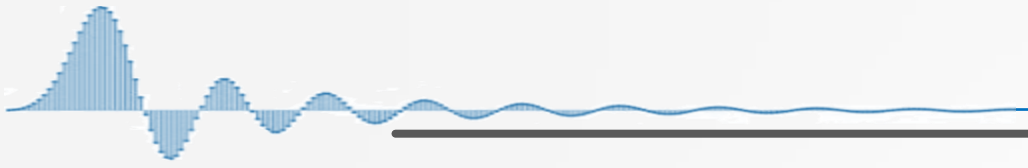
$$A_{21} = A(2,1);$$

$$A = \begin{bmatrix} 1 & 1 & 3 & 4; \\ 5 & 6 & 7 & 8; \\ 1 & 2 & 3 & 4; \end{bmatrix};$$



$$A_{21} = 5;$$





Σύνταξη Εντολών

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

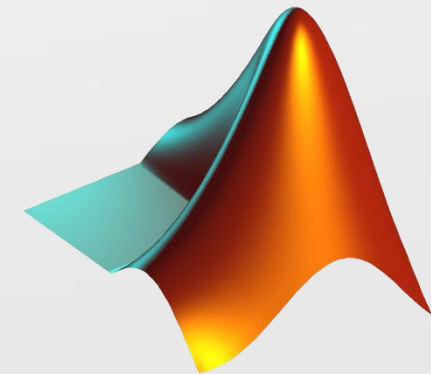
```
for i=1:size(A,1)
  for j=1:size(A,2)
    A(i,j);
  end
end
```

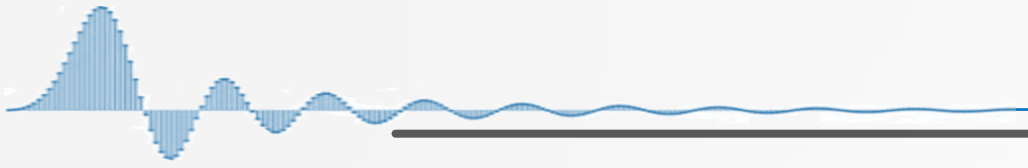


Πλήθος γραμμών



Πλήθος στηλών





Σύνταξη Εντολών

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

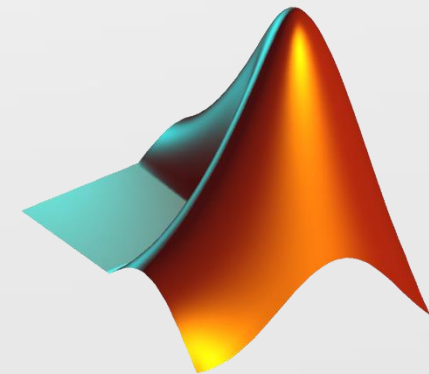
```
A_prwth_sthlh = A(:,1);
```

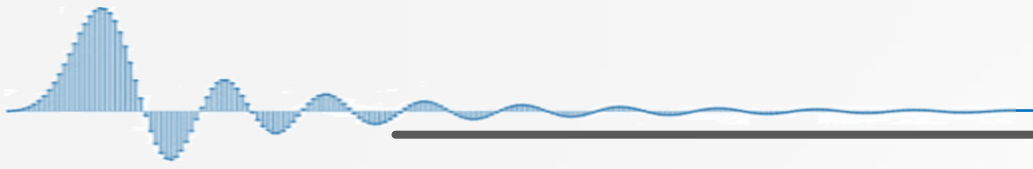
```
A = [1 1 3 4;  
      5 6 7 8;  
      1 2 3 4];
```



A_prwth_sthlh =

1
5
1





Σύνταξη Εντολών

Tip:
A(γραμμή,στήλη);

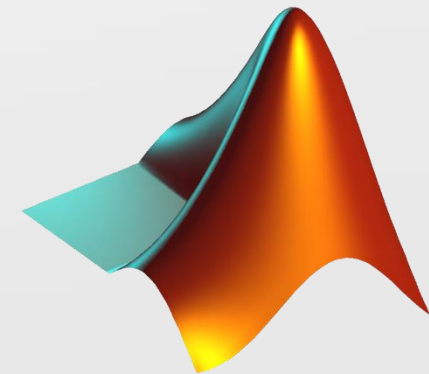
Είτε την προσπέλαση ολόκληρου υποπίνακα κάποιου πίνακα:

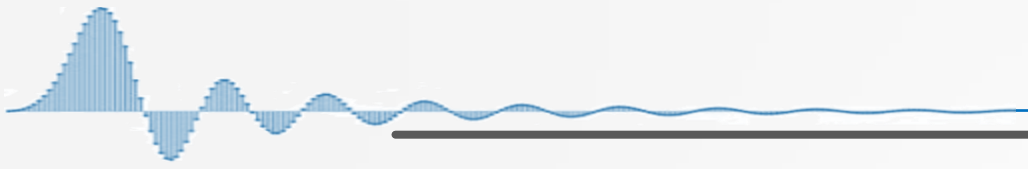
A[2:3,2:3];

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

A[2:3,2:3] =

6 7
2 3



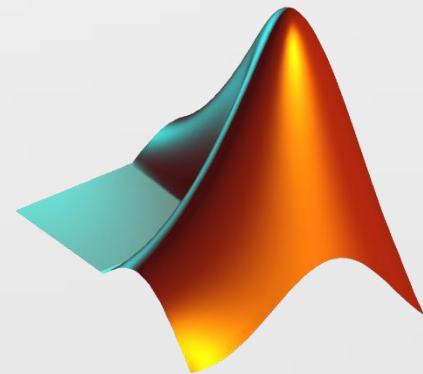


Σύνταξη Εντολών

Πίνακες μεγαλύτερης διάστασης ορίζονται ως εξής:

$$A_3D(:, :, 1) = A;$$

$$A_3D(:, :, 2) = B;$$





Σύνταξη Εντολών

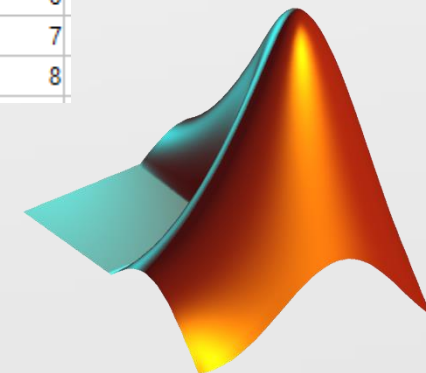
Διανύσματα/Μονοδιάστατα σήματα

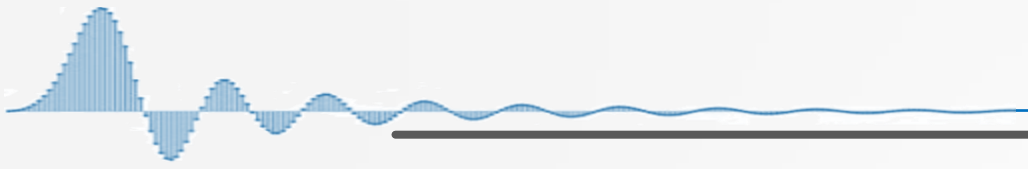
Δεδομένου ότι τα διανύσματα δεν είναι τίποτα άλλο παρά πίνακες μίας διάστασης, κατ' αντιστοιχία η δήλωση γνωστών διανυσμάτων γίνεται ως εξής:

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

	1	2	3	4
1	1	2	3	4

	1
1	5
2	6
3	7
4	8



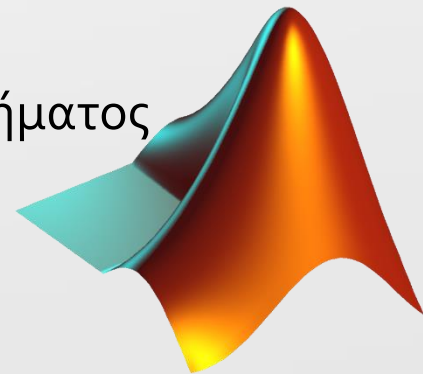


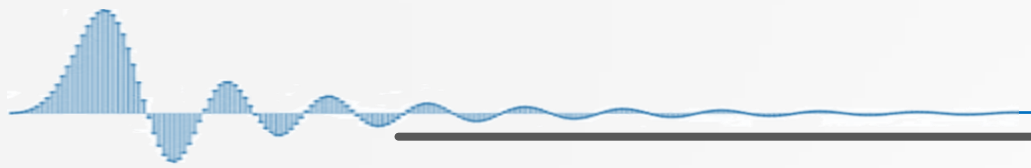
Σύνταξη Εντολών

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
- ✓ Με την εντολή **figure** δημιουργούμε ένα παράθυρο σχήματος





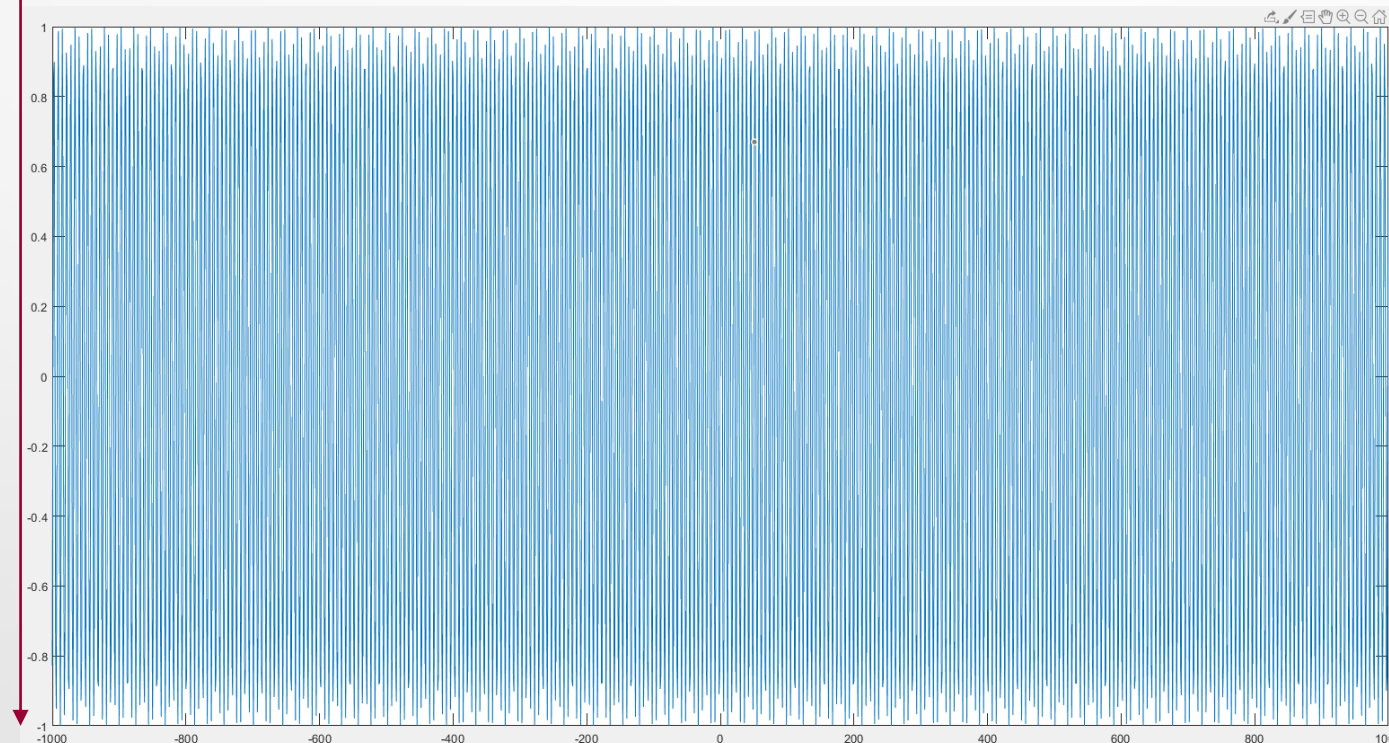
```
t=[-1000:1:1000];  
sint = sin(t);  
figure, plot(t,sint)
```

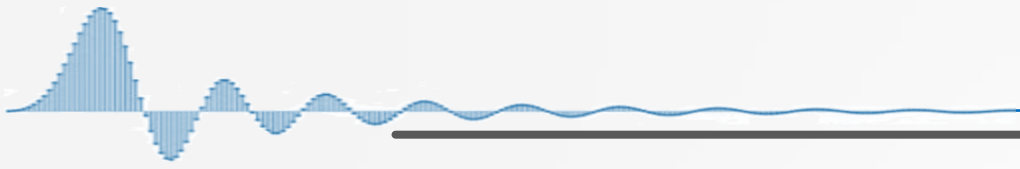
→ Από το -1000 μέχρι το 1000 με βήμα 1

→ sin → ημίτονο

→ figure → παράθυρο

→ plot → γραφική παράσταση





Σύνταξη Εντολών

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.

K2 =

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

K1=zeros(1,7)

K2=ones(5,5)

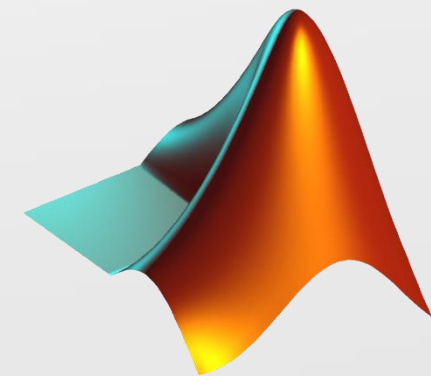
K3=randn(7,7,7)

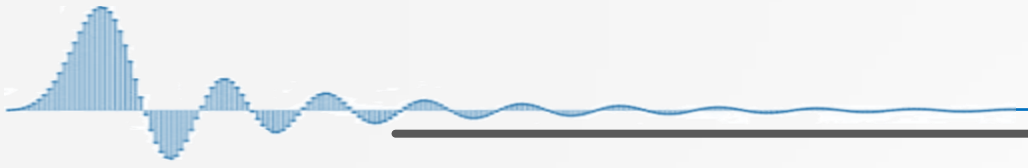
K1 =

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

K3(:, :, 1) =

0.5377	0.3426	0.7147	-1.2075	0.2939	1.4384	0.3192
1.8339	3.5784	-0.2050	0.7172	-0.7873	0.3252	0.3129
-2.2588	2.7694	-0.1241	1.6302	0.8884	-0.7549	-0.8649
0.8622	-1.3499	1.4897	0.4889	-1.1471	1.3703	-0.0301
0.3188	3.0349	1.4090	1.0347	-1.0689	-1.7115	-0.1649
-1.3077	0.7254	1.4172	0.7269	-0.8095	-0.1022	0.6277
-0.4336	-0.0631	0.6715	-0.3034	-2.9443	-0.2414	1.0933

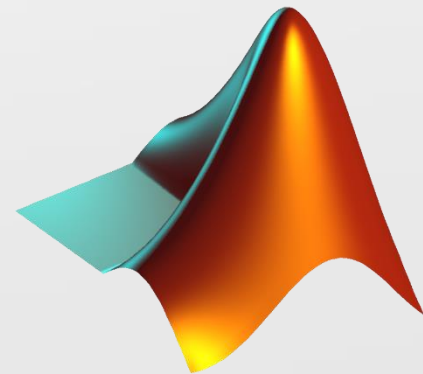


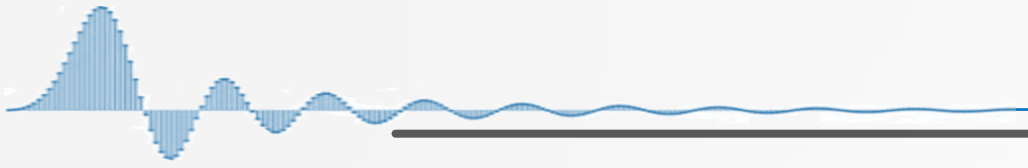


Σύνταξη Εντολών

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 συνάρτησης:

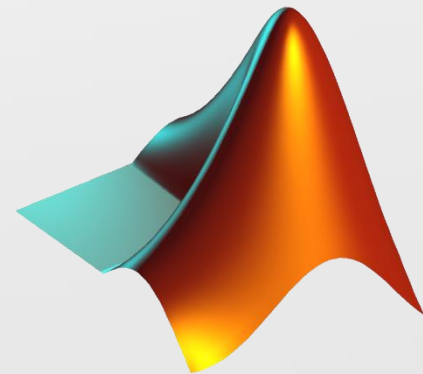


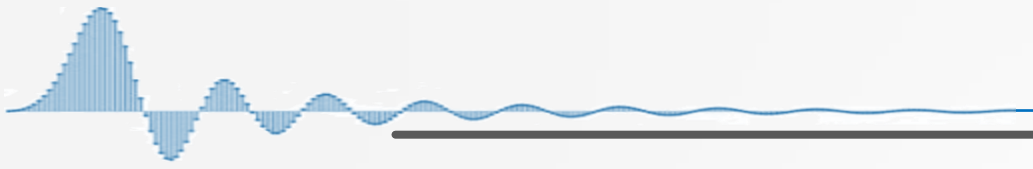


Σύνταξη Εντολών

- 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)
```





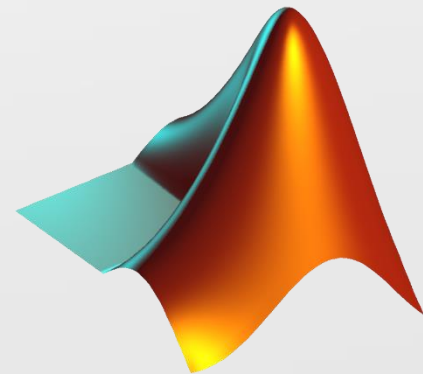
Σύνταξη Εντολών

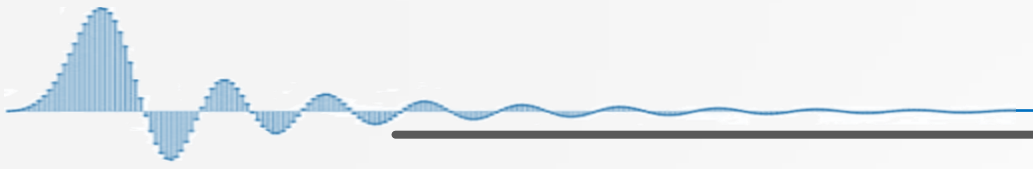
3) Γράφουμε τον κώδικα που θέλουμε να εκτελεί η συνάρτηση.

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)
```





Συναρτήσεις

Tips:

- ✓ Το όνομα της συνάρτησης πρέπει να είναι ίδιο με το όνομα του αρχείου.
- ✓ Για να μπορέσουμε να καλέσουμε την συνάρτηση θα πρέπει να είμαστε στον ίδιο φάκελο.
- ✓ Η υλοποίηση της συνάρτησης βρίσκεται μεταξύ της δεσμευμένης εντολής **function** και **end**

Υλοποίηση Συνάρτησης:

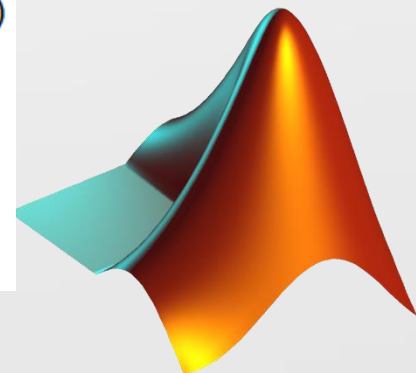
```
square.m  x  +
1  function y = square(x)
2  —      y = x*x;
3  —      end
```

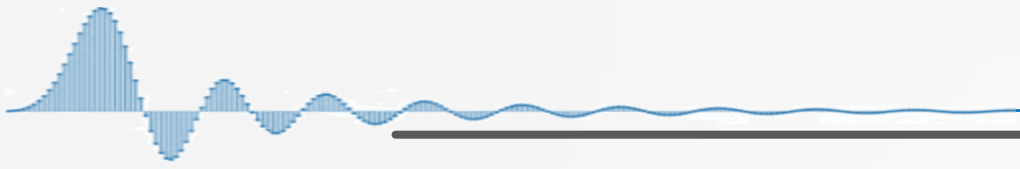
Κλήση Συνάρτησης:

```
>> z = 2;
>> res = square(z)

res =

4
```





Συναρτήσεις

Συνάρτηση με περισσότερες επιστρεφόμενες τιμές:

Υλοποίηση Συνάρτησης:

```
1 function [s,c] = square_c(x)
2     s = x*x;
3     c = x^3;
4 end
```

Κλήση Συνάρτησης:

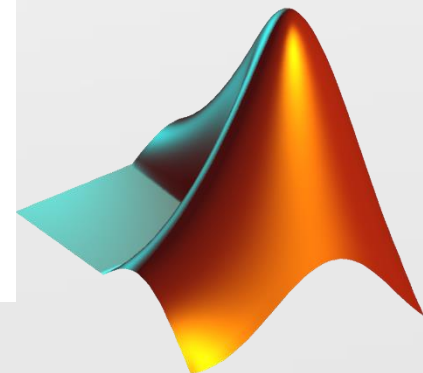
```
>> x = 8;
>> [a,b] = square_c(x)

a =

    64

b =

   512
```



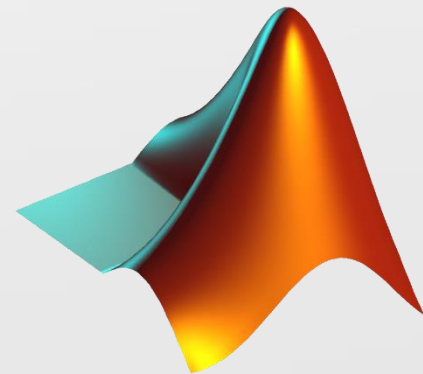


ΠΕΙΡΑΜΑΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ

Άσκηση 1

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

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





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

ΛΥΣΗ

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

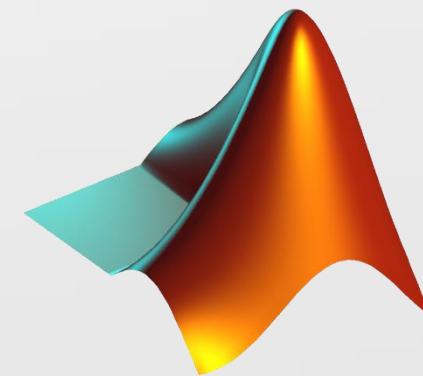
```
    mirrored = zeros(size(I,1),size(I,2));
```

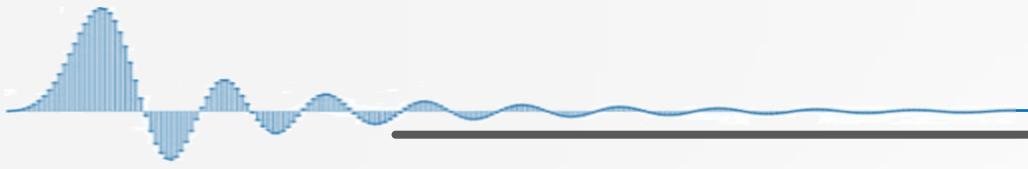
```
end
```

- ✓ Δημιουργούμε μια συνάρτηση με όνομα `mirroring` με όρισμα ένα πίνακα `I`
- ✓ Σε μια μεταβλητή `mirrored` εκχωρούμε ένα μηδενικό πίνακα με μέγεθος το μέγεθος του πίνακα που έχουμε σαν όρισμα

```
mirrored =
```

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





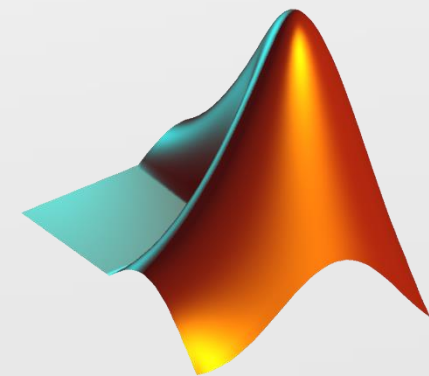
ΛΥΣΗ

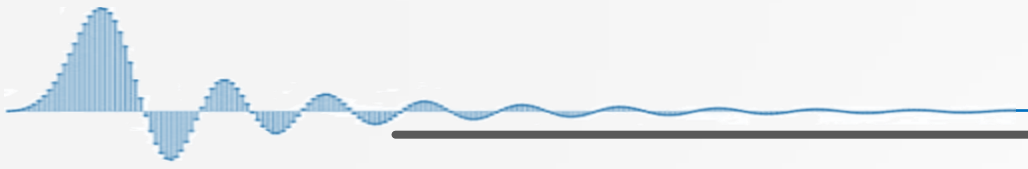
```
function mirrored = mirroring(I)
```

```
    mirrored = zeros(size(I,1),size(I,2));  
    count = 1;
```

```
end
```

- ✓ Αρχικοποιούμε μια μεταβλητή `count` (μετρητή) ίση με 1





ΛΥΣΗ

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

```
    mirrored = zeros(size(I,1),size(I,2));
```

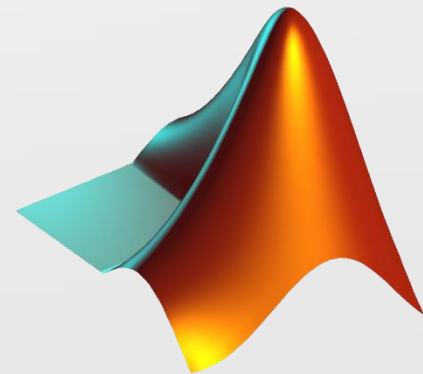
```
    count = 1;
```

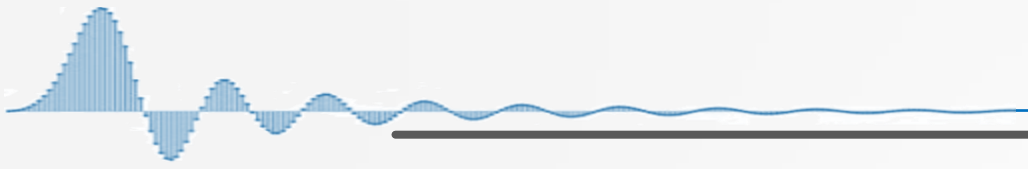
```
    for i = size(I,2):-1:1
```

```
        end
```

```
end
```

- ✓ Υλοποιούμε μια for στην οποία:
 - ✓ Αρχικοποιούμε το i με το size(I,2)
 - ✓ Αφαιρόντας -1 κάθε φορά εκτελούμε το κώδικα στην for
 - ✓ Μέχρι το i = 1



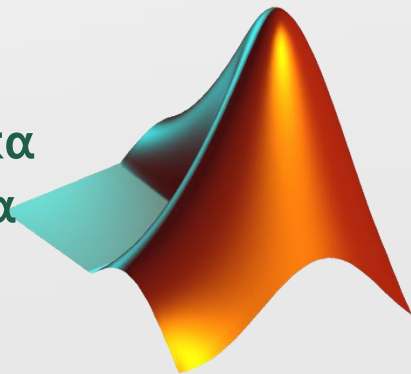


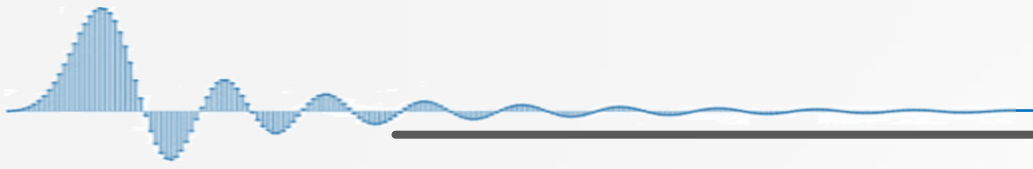
ΛΥΣΗ

```
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
end
```

- ✓ **Λειτουργία της for:**
 - ✓ Στο μηδενικό πίνακα `mirrored` για κάθε στήλη του σύμφωνα με το `count` παίρνουμε την στήλη `i` του πίνακα μας και την βάζουμε στην αντίστοιχη στήλη του πίνακα `mirrored`.
 - ✓ Αυξάνουμε το μετρητή `count` κατά 1





ΛΥΣΗ

Υλοποίηση Συνάρτησης:

```
function mirrored = mirroring(l)

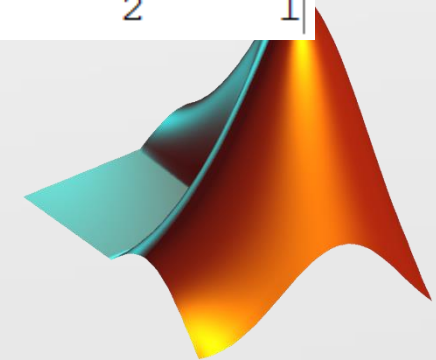
mirrored = zeros(size(l,1),size(l,2));
count = 1;
for i = size(l,2):-1:1
    mirrored(:,count) = l(:,i);
    count = count + 1;
end
end
```

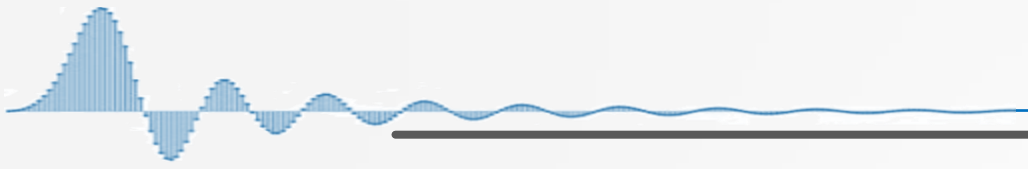
Κλήση Συνάρτησης:

```
>> A = [1 1 3 4;
        5 6 7 8;
        1 2 3 4];
>> mirroring(A)

ans =

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





Εφαρμόστε την συνάρτηση που υλοποιήσατε στην εικόνα του cameraman.

ΛΥΣΗ

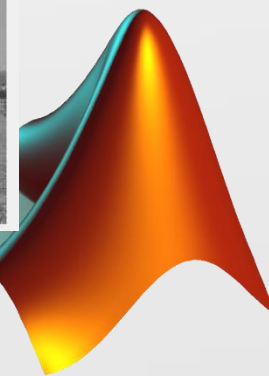
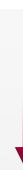
Υλοποίηση Συνάρτησης:

```
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
end
```

Κλήση Συνάρτησης:

```
>> I = imread('cameraman.tif');
>> m = mirroring(I);
>> figure, imshow(uint8(m))
```

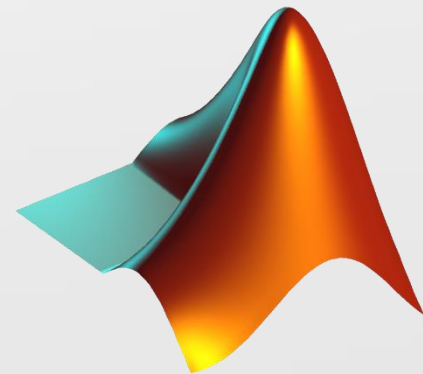


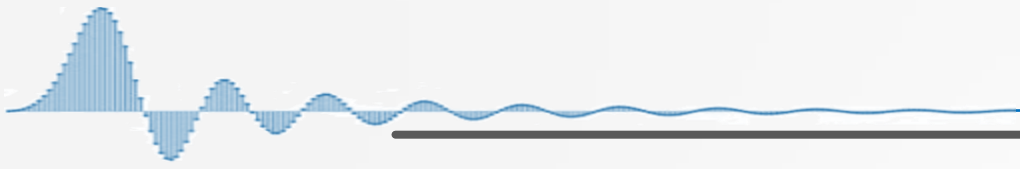


ΠΕΙΡΑΜΑΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ

Άσκηση 2

Δημιουργήστε μια συνάρτηση η οποία να δέχεται έναν περιττό αριθμό m και να επιστρέφει έναν 3-διάστατο πίνακα $m \times m \times m$ 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.

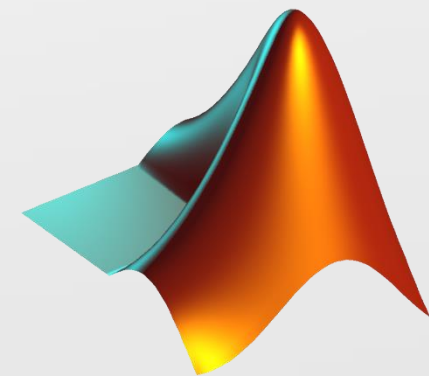


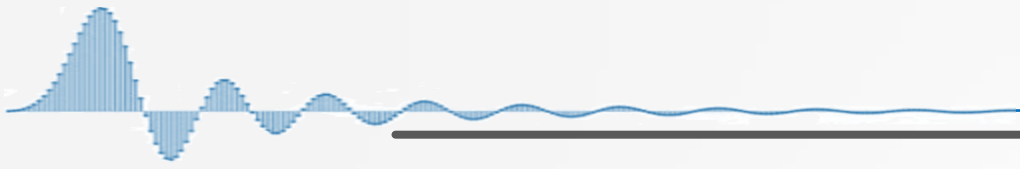


ΛΥΣΗ

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

- ✓ Δημιουργία Συνάρτησης:
 - ✓ Η συνάρτηση μας θα επιστρέφει ένα πίνακα `matrix` και την τιμή της μεταβλητής `center_value`
 - ✓ Η συνάρτηση μας θα δέχεται ένα πίνακα `m`





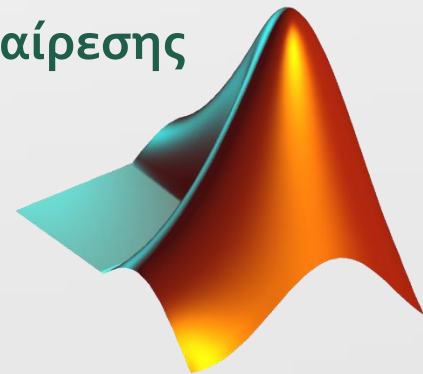
ΛΥΣΗ

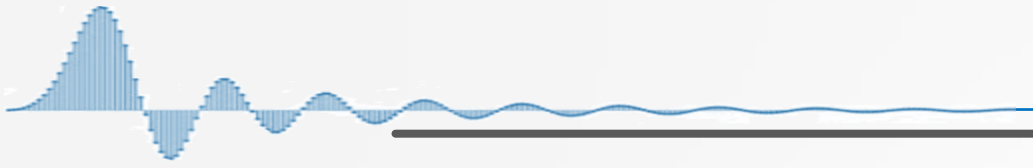
```
function [matrix, center_value] = rand_3d(m)

    if mod(m,2)==0

end
```

- ✓ Υλοποίηση Συνάρτησης:
 - ✓ Δημιουργούμε μια συνθήκη για την περίπτωση που ο αριθμός είναι άρτιος
 - ✓ Η εντολή mod επιστρέφει το υπόλοιπο της διαίρεσης



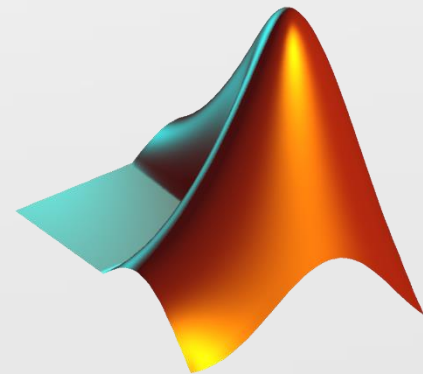


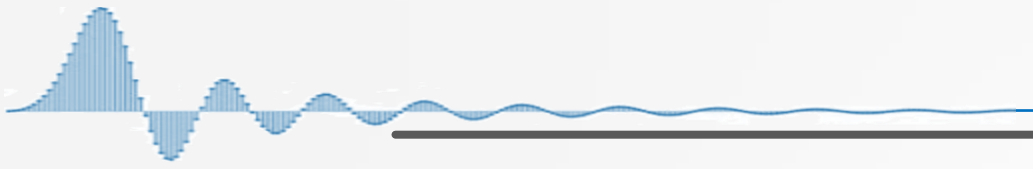
ΛΥΣΗ

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

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

- ✓ Υλοποίηση Συνθήκης:
 - ✓ disp → εμφάνιση μηνύματος





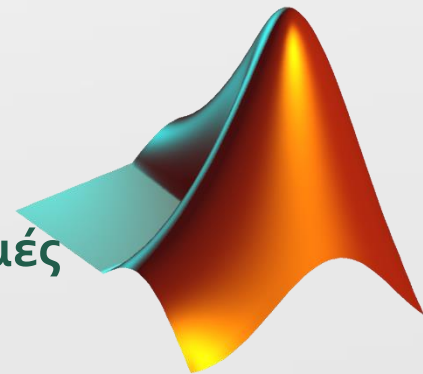
ΛΥΣΗ

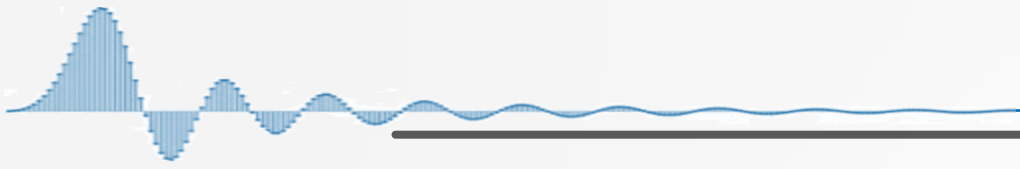
```
function [matrix, center_value] = rand_3d(m)

    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
```

- ✓ Υλοποίηση else:
 - ✓ rand → δημιουργούμε ένα τρισδιάστατο πίνακα με τυχαίες τιμές
 - ✓ Στο center_value εκχωρούμε το μεσαίο στοιχείο





Υλοποίηση Συνάρτησης:

```
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
```

ΛΥΣΗ

Κλήση Συνάρτησης:

```
>> m = 3;
>> [x,y]=rand_3d(m)

x(:,:,1) =

    0.1190    0.3404    0.7513
    0.4984    0.5853    0.2551
    0.9597    0.2238    0.5060

x(:,:,2) =

    0.6991    0.5472    0.2575
    0.8909    0.1386    0.8407
    0.9593    0.1493    0.2543

x(:,:,3) =

    0.8143    0.3500    0.6160
    0.2435    0.1966    0.4733
    0.9293    0.2511    0.3517

y =

    0.1386
```

