Αλληλεπίδραση με το Matlab

Περιγραφή της διαδικασίας πως εργαζόμαστε με το Matlab, και της προετοιμασίας και παρουσίασης των αποτελεσμάτων μιας εργασίας με το Matlab.

Ειδικότερα θα συζητήσουμε μερικά στοιχεία του Matlab interface και την χρησιμότητα των

- script M-files
- Function M-files
- Diary files

Το Matlab Desktop

- **Command window :** ο χώρος που δίνονται οι εντολές και οδηγίες οι οποίες οδηγούν το Matlab να υπολογίζει, να δίνει γραφικές παραστάσεις και γενικά όλα τα υπέροχα μαγικά που θα μάθουμε στο μάθημα.
- **Command History** (περιέχει την τρέχουσα ιστορία των εντολών τις οποίες δίνουμε στο command window)

Είναι χρήσιμο για δυο λόγους

1. Μας παρέχει την δυνατότητα μιας γρήγορης επισκόπησης των εντολών που δώσαμε προηγουμένως
2. Μας σώζει χρόνο πληκτρολόγησης

- **Workspace**
- **Current folder**
- **Start :** Είναι συντομεύσεις που μας επιτρέπουν να έχουμε πρόσβαση σε διάφορες δυνατότητες του Matlab (toolboxes, Matlab web tools, Help browser, editor.....)

- **Workspace**

Περιέχει έναν κατάλογο από τις μεταβλητές και το μέγεθός τους (αλλά όχι τις τιμές τους). Κάνοντας διπλό click πάνω στην μεταβλητή ανοίγει ένα νεό παράθυρο που ονμάζεται *Array Editor* στο οποίο μπορούμε να επέμβουμε στις τιμές του διανύσματος των τιμών.

Η εντολή **openvar** ακολουθούμενη με την μεταβλητή ανοίγει πάλι τον Array Editor με τις τιμές της μεταβλητής.

Αν θέλουμε να τελείωσουμε μια εργασία στο Matlab και δεν θέλουμε να υπολογίσουμε πάλι όλα όσα κάναμε, σώζουμε το τρέχον Workspace με **save.**

save myfile σώζει τις τιμές όλων των ενεργών μεταβλητών σε ένα αρχείο που το ονομάζει **myfile.mat**

save myfile X Y σώζει μόνο τις τιμές των μεταβλητών X και Y.

- **load myfile** φορτώνει όλες τις τιμές των μεταβλητών που υπάρχουν στο αρχείο **myfile.mat**

- **Working directory**

Νέα αρχεία που δημιουργούμε μέσα από το Matlab σώζονται στο τρέχον *working directory.* Μπορούμε να αλλάξουμε το dictory αυτό από το αρχικό ή ακόμα και να έχουμε διαφορετικά working directories για διαφορετικά projects.

```
>> cd C:\ProjectA
```

Εναλλακτικά μπορούμε να βάλουμε το directory ProjectA στο path των directories που ψάχνει το Matlab για να βρει αρχεία.

```
>> addpath C:\ProjectA
```

- Το working directory και το path είναι τα μοναδικά directories που ψάχνει το Matlab να βρει αρχεία. **ΠΡΟΣΟΧΗ:** με την ονομασία των αρχείων που σώζετε γενικός κανόνας: δώστε διαφορετικά ονόματα για διαφορετικά αρχεία και projects.

- Εντολές σε γραφικό περιβάλλον που αλλάζουν το path:

editpath ή **pathtool**

- Χρήσιμες εντολές μέσα στο Matlab είναι οι: **pwd** (μας δείχνει το working directory), **ls, dir** (κατάλογος αρχείων στο working directory).

- **Command window**

Έχουμε δει πως δίνουμε εντολές στο command window. Για παράδειγμα ας υποθέσουμε ότι θέλουμε να υπολογίσουμε τις τιμές

sin(0.1)/0.1, sin(0.01)/0.01, and sin(0.001)/0.001

με 15 ψηφία. Το πρόβλημα μπορεί να λυθεί απευθείας στο command window

- >> x = [0.1, 0.01, 0.001];
- >> y = sin(x)./x

y =

0.9983 1.0000 1.0000

Αλλά το αποτέλεσμα δεν είναι αυτό που θέλαμε (ακρίβεια 15 δεκ. Ψηφίων). Για απλά προβλήματα μπορούμε να χρησιμοποιήσουμε το command history και να ξαναδώσουμε εντολές.

- Για πιο σύνθετα προβλήματα όμως ενδείκνυται να χρησιμοποιήσουμε τα λεγόμενα **M-files**

M-Files

- **Script M-files**

Είναι απλά αρχεία κειμένου που περιέχουν εντολές Matlab (δημιουργούνται με απλούς κειμενογράφους όπως το notepad/Windows ή emacs, vi/UNIX).

Υπάρχει η δυνατότητα όμως να χρησιμοποιήσουμε τον editor του Matlab με την εντολή **edit**

Γράψτε τις εντολές

```
format long

x = [0.1, 0.01, 0.001];

y = sin(x)./x
```

σε ένα αρχείο και σώστε το με το όνομα task1.m

Στο Matlab πατήστε

```
>>task1
```

Μπορούμε εύκολα να τροποποιήσουμε το αρχείο μας. Οι εντολές δεν εμφανίζονται στο output του Matlab. Για να γίνει αυτό χρησιμοποιούμε την εντολή **echo on** και στο τέλος του αρχείου **echo off**

Σχόλια: Είναι πολύ χρήσιμο να βάζουμε σχόλια σε ένα αρχείο εντολών Matlab για να θυμόμαστε τι ακριβώς υπολογίσουμε. Αυτό γίνεται αρχίζοντας μια γραμμή στο αρχείο με το σύμβολο **%**

```
echo on

% Χρησιμοποιούμε ακρίβεια 15 ψηφίων

format long

x = [0.1, 0.01, 0.001];

y = sin(x)./x

% Αυτές οι τιμές δείχνουν το γεγονός ότι το όριο του

% sin(x)/x καθώς το x τείνει στο 0 είναι 1.

echo off
```

- **Δομή των script M-Files**

Για να είναι ένα script M-file χρήσιμο να χρησιμοποιηθεί και πάλι θα πρέπει να είναι αυτοδύναμο δίχως να επηρεάζεται από άλλες μεταβλητές που ίσως να χρησιμοποιήσαμε προηγουμένως και δίχως ξεχασμένα γραφικά. Αυτά επιτυγχάνονται με τις εντολές **clear all** και **close all**

```
% Σβήσε όλες τις παλιές μεταβλητές

clear all

% Σβήσε όλα τα παλιά γραφικά

close all

% Δείξε τις εντολές στο παράθυρο command window

echo on

% Χρησιμοποιούμε ακρίβεια 15 ψηφίων

format long

% Όρισε το διάνυσμα με τις τιμές των ανεξάρτητων μεταβλητών

x = [0.1, 0.01, 0.001];

y = sin(x)./x

% Αυτές οι τιμές δείχνουν το γεγονός ότι το όριο του  sin(x)/x καθώς το x τείνει

% στο 0 είναι 1.

echo off
```

Function M-files

Πολλές φορές χρειάζεται να επαναλάβουμε μια διαδικασία για διαφορετικές τιμές μιας μεταβλητής.Τότε μπορούμε να ορίσουμε όπως έχουμε δει μια δικιά μας συνάρτηση με την εντολή **inline.** Αλλά, μερικές φορές είναι προτιμότερο να δημιουργήσουμε μια συνάρτηση χρησιμοποιώντας ένα M-file.

Ας υποθέσουμε ότι θέλουμε να υπολογίσουμε τις τιμές της έκφρασης sin(x)/x με x=10^-b για διάφορες τιμές τις παραμέτρου b. Επιπλέον, ας υποθέσουμε ότι θέλουμε να βρούμε την μικρότερη τιμή του b για την οποία το sin(x^-b)/x^-b και το 1 συμφωνούν σε 15 ψηφία.

Ένα function M-file sinelimit.m που λύνει το πρόβλημα είναι το εξής:

```
function y = sinelimit(c)

% SINELIMIT υπολογίζει το sin(x)/x για x = 10ˆ(-b),

% όπου b = 1, ..., c.

format long

b = 1:c;

x = 10.ˆ(-b);

y = (sin(x)./x)’;
```

Ως M-file ένα function M-file θα πρέπει να βρίσκεται σε ένα Matlab working directory.

Η πρώτη γραμμή του αρχείου περιέχει μια πρόταση **function** που ταυτοποιεί το αρχείο ως ένα function M-file.Επίσης προσδιορίζει το όναμα της συνάρτησης και προσδιορίζει τα ορίσματα και τις τιμές. Στο παράδειγμα την συνάρτηση την ονμάσαμε sinelimit. Το όνομα του **αρχείου** και της **συνάρτησης** πρέπει να είναι τα **ίδια.**

Η συνάρτηση sinelimit δέχεται το c και δίνει το y σαν απάντηση. Οι υπόλοιπες γραμμές ορίζουν την συνάρτηση. b είναι ένα διάνυσμα γραμμή με ακεραίους από το 1 μέχρι το c. Το διάνυσμα y περιέχει τα αποτελέσματα του υπολογισμού sin(x)/x με x=10^-b. Ο τόνος κάνει το y ένα διάνυσμα στήλη. Έχουμε και **;** για να μην εμφανίζονται στην οθόνη τα b, x και y.

Γενικά θα πρέπει να βάζουμε **;** στα function M-files.

- Αν δεν είχαμε **;** στο τέλος το Matlab θα τύπωνε το αποτέλεσμα δυο φορές!!!! μια για το y και μια στην ans.

Loops

Ένα **loop** προσδιορίζει ότι μια εντολή ή μια ομάδα ενολών θα πρέπει να επαναληφθεί μερικές φορές. Αυτό επιτυγχάνεται με την εντολή **for**

Ένα απλό παράδειγμα που υπολογίζει και τυπώνει το

$10! = 10 \cdot 9 \cdot 8 \cdot \cdot \cdot 2 \cdot 1$

```
f = 1;
for n = 2:10
f = f*n;
end
f
```

Παρουσίαση αποτελεσμάτων στο Matlab

Αρχεία diary

Με την εντολή diary μπορούμε να έχουμε τα αποτελέσματα σε ένα αρχείο ή ακόμα όλο το script M-file να τυπώνεται σε ένα απλό txt αρχείο έτσι ώστε τα αποτελέσματά μας να είναι διαθέσιμα για κάποια παρουσίαση.

```
>>delete task2.txt

>>diary task2.txt

>>echo on

>>Matlab commands

>>echo off

>>diary off
```

Παρουσίαση γραφικών

Με την εντολή **print** μπορούμε να σώσουμε ένα γραφικό σε αρχείο eps το οποίο μπορούμε να το ενσωματώσουμε σε κάποιο άλλο αρχείο (π.χ. Word) στο οποίο παρουσιάζουμε τα αποτελέσματά μας. Για παράδειγμα σε ένα script M-file με όνομα print1.m γράφουμε:

```
xx = 2*pi*(0:0.02:1);

plot(xx, sin(xx))

% Put a title on the figure.

title('Figure A: Sine Curve')

pause

% Store the graph in the file figureA.eps.

print -deps figureA
```

Σαν ένα άλλο παράδειγμα ας υποθέσουμε ότι θέλουμε να πάρουμε σε ένα μόνο γράφημα τις γραφικές παραστάσεις των συναρτήσεων sin(x), sin(2 x), sin(3 x). Το script M-file δίνει μια λύση:

```
echo on

% Όρισε τις τιμές της μεταβλητής x.

x = 2*pi*(0:0.01:1);

% Σβήσε όλα τα παλιά γραφικά, και ετοιμάσου για νέα.

close all; axes; hold on

% Τρέξε το loop για τρια γραφήματα ημιτόνων.

for c = 1:3

plot(x, sin(c*x))

echo off

end

echo on

hold off

% Δώσε ένα τίτλο στην εικόνα.

title('Διάφορες καμπύλες ημιτόνων')

pause
```

Μια γενική διαδικασία

- Δημιουργήστε ένα **script M-file** στο working directory που να περιέχει τις εντολές. Χρησιμοποιήστε **echo on** στην αρχή του αρχείου για να βλέπετε τις εντολές και τι αποτέλεσμα δίνουν όταν τρέχει το M-file.
- Δουλέψτε μεταξύ του M-file στον editor και του script που τρέχει μέχρι να είστε ικανοποιημένοι από το αποτέλεσμα. Να σώζετε το M-file σε κάθε αλλαγή που εκτελείτε.
- Προσθέστε σχόλια στο M-file που να εξηγούν τους ενδιάμεσους υπολογισμούς και να ερμηνεύετε τα αποτελέσματα.
- Αν επιθυμείτε προσθέστε τις προτάσεις **delete** and **diary** στα M-files
- Αν παράγετε γραφικά προσθέστε την πρόταση **print** η οποία σώζει τα γραφικά σε αρχεία. Χρησιμοποιήστε **pause** statements όταν πρέπει.
- Αν βάζετε το αρχείο diary σε επεξεργαστή κειμένου τότε μπορείτε να βάλετε το γραφικό αμέσως μετά τις εντολές που το δημιούργησαν. Μπορείτε να αλλάξετε τα fonts του κειμένου έτσι που να είναι πιο εύκολα στην διαφοροποίηση οι εντολές τα σχόλια και το αποτέλεσμα

Μορφοποίηση των M-files

- Χρησιμοποιήστε **clear all** and **close all** στην αρχή του M-file.
- Χρησιμοποιήστε **echo on** πολύ γρήγορα στο M-file για να βλέπετε την δράση και την αντίδραση των εντολών που έχετε δώσει στο Matlab.
- Αν δημιουργήτε γραφικά χρησιμοποιήστε τις εντολές **hold on** και **hold off** προσεκτικά. Γενικά, θα πρέπει κάθε **hold off** να ακολουθείται από **pause.** Αλλιώς το επόμενο γραφικό δεν θα είναι ορατό.
- Μην βάζετε γυμνές εντολές **print** στα M-files. Να σώζετε σε αρχεία.
- Η εντολή **keyboard** είναι μια εναλλακτική της εντολής **pause**. Αν υπάρχει η εντολή **keyboard** σε ένα M-file, τότε όταν το MATLAB φτάσει σε αυτή, η εκτέλεση του M-file διακόπτεται και εμφανίζεται το prompt με το γράμμα K πριν από αυτό. Τότε μπορείτε να εκτελέσετε οποιαδήποτε εντολή του Matlab. Αυτό είναι χρήσιμο αν θέλουμε να εξετάσουμε ή να δώσουμε άλλες τιμές σε μεταβλητές στην καθώς τρέχουμε το script M-file. Για να επιστρέψουμε στην εκτέλεση του script M-file πληκτρολογούμε **return** και μετά το πλήκτρο ENTER
- Ορισμένες φορές είναι προτιμητέα η εντολή **input**. Για παράδειγμα το:
- var = input(’Input var here: ’) στο script οταν το MATLAB φτάσει στο σημείο εκείνο θα κάνει παύση και θα τυπώσει “Input var here:” και θα περιμένει να δώσουμε τιμή στην var
- Με CTRL+C διακόπτεται το τρέξιμο σε ένα script M-file.