Σχολή Θετικών Επιστημών Τμήμα Μαθηματικών Εθνικόν και Καποδιστριακόν Πανεπιστήμιο Αθηνών

453. Γραφικά με Η/Υ

Matlab και Γραφικά

2020

1 Εισαγωγή

Στη συγκεκριμένη διάλεξη θα υπενθυμίσουμε μερικά βασικά στοιχεία από το υπολογιστικό πακέτο Matlab τα οποία θα χρειαστούμε για να υλοποιήσουμε την προαιρετική εργασία που ανακοινώθηκε, αλλά και για να δούμε στην πράξη όσα έχουμε πει για τους μετασχηματισμούς.

1.1 Matlab Desktop

Το παράθυρο με το οποίο έρχεται σε πρώτη επαφή ο χρήστης όταν εκκινεί το πρόγραμμα, ονομάζεται Matlab desktop. Μέρος του είναι και το command window, το βασικό μέρος αλληλεπίδρασης του με το software. Επιλέγοντας από το menu την επιλογή view και στην συνέχεια την επιλογή Desktop Layout μπορούμε να παραμετροποιήσουμε το παράθυρο ανάλογα με τις ανάγκες μας.

📣 MATLAB R20176	– a ×
HOWE PLOTS APPS	🛃 🔄 🕹 🕼 📾 🐨 😋 🕐 Search Documentation 🛛 🔎 Log In
Image: Construction Image: Construction	O Consulty Oreconstrainty Oreconstrain
 	م -
Current Folder 💿 Command Window	Workspace (*
Image: Section 1 JA Image: Section 2 JA Image: Section 2 Section 2 Im	Norm - Notor Communit Makey 3 03-Max-20 11145
Details v Solver a file to view details	
The Bast	
म २ म 🖌 🛅 💁 🖹 🔤 🚾 🐙 🎻 🧈	🍉 🐱 🖓 🕸 (PHK 🛛 1143 (SMA) - 20

Σχήμα 1: Matlab desktop

Μπορεί το command window να είναι το μέρος που εκτελούμε τους υπολογισμούς μας με το πακέτο, αλλά βασικό εργαλείο για εμάς αποτελεί ο ενσωματωμένος editor του Matlab,στον οποίο γράφουμε, χρησιμοποιώντας την γλώσσα προγραμματισμού, τις μεθόδους που θα υλοποιήσουμε. Εκεί θα δημιουργούμε τον βασικό τύπο αρχείων που υποστηρίζει το πρόγραμμα. Τα λεγόμενα m-files

Τα m-files χωρίζονται σε 2 μεγάλες κατηγορίες:

- 1. T α script files
- 2. Ta function files



Σχήμα 2: Ο editor του Matlab

Τα πρώτα μπορούμε απλοϊκά να πούμε ότι είναι όπως ένα σενάριο σε ένα θέατρο, όπου ο διερμηνέας του Matlab (interpreter) διαβάζει μια μια τις εντολές που περιέχει και τις εκτελεί. Ένα script file αλληλοεπιδρά με το περιβάλλον του πακέτου και δημιουργεί ή μεταβάλει μεταβλητές. Ενώ τα δεύτερα αποτελούν μαύρα κουτιά και ο μόνος τρόπος που έχουν για να επικοινωνήσουν με το υπόλοιπο σύστημα είναι μέσω των μεταβλητών εισόδου και εξόδου που έχουν ορισθεί.

1.2 Octave

Mia Open Source εναλλακτική του Matlab είναι το Octave το οποίο χρησιμοποιεί την ίδια σύνταξη για να προγραμματίσεις και μπορείτε να το βρείτε στον ακόλουθο σύνδεσμο : https://www.gnu.org/software/octave/

2 Μεταβλητές

To Matlab παρέχει ένα σύνολο πρωτογενών δομών δεδομένων, το πιό βασικό είναι οι floating point πίνακες. Οι σχεδιαστές του πακέτου, βελτιστοποίησαν την χρήση τους, κάνοντας τη διαχείριση τους από εμάς ευκολότερη σε σχέση με άλλες γλώσσες προγραμματισμού. Και ο βασικός τρόπος για να δουλέψουμε με αυτούς είναι με την βοήθεια μεταβλητών.

2.1 Επιτρεπτά ονόματα

Για τις μεταβλητές πρέπει να αχολουθούμε τους αχόλουθους απλούς χανόνες

- 1. Χρησιμοποιούμε πάντα λατινικούς χαρακτήρες και αριθμούς
- 2. Από ειδικούς χαρακτήρες επιτρέπεται μόνο η χρήση της κάτω παύλας _
- 3. Πάντα αρχίζουμε με γράμμα και δεν αφήνουμε κενά ανάμεσα στις λέξεις
- 4. Υπάρχει διάκριση ανάμεσα στα κεφαλαία και στα πεζά
- 5. Δεν επιτρέπονται σαν όνομα μεταβλητών οι λέξεις που επιστρέφει η εντολή **iskeyword**

Έτσι για παράδειγμα, μπορούμε να χρησιμοποιήσουμε τις λέξεις :

divider, c1, mesos_oros, CAR

Αλλά όχι τις:

1a, mesos oros, if, μεταβλητή

2.2 Ανάθεση τιμής

Για να ορίσουμε μια τιμή σε κάποια μεταβλητή, είτε δίνοντάς της απευθείας κάποια, είτε εκχωρώντας το αποτέλεσμα μιας πράξης, αρκεί να γράψουμε:

μεταβλητή = τιμή

Παραδείγματα:

a = 1;

Το αποτέλεσμα αυτής της πράξης μας είναι η δημιουργία της μεταβλητής a η οποία έχει την τιμή 1. Η μεταβλητή a είναι ένας 1 × 1 πίναχας. Με τον ίδιο τρόπο μπορούμε να δημιουργήσουμε μεταβλητές που θα έχουν ως τιμές :

b = 4.3; c = a+b; d = sqrt(5); v = [1 2 3]; A = [1 2;3 4]; B = ones(4); C = zeros(2,3); s = 'Yes';

Το ελληνικό ερωτηματικό (;) στο τέλος κάθε γραμμής δεν έχει σχέση με την ανάθεση τιμών σε μεταβλητή, αλλά οδηγεί το Matlab να μην εμφανίσει το αποτέλεσμα της διαδικασίας που κάνει στην οθόνη μας.

2.3 Προκαθορισμένες Μεταβλητές

To Matlab διαθέτει ένα σύνολο ειδικών μεταβλητών που έχουν μια προκαθορισμένη αρχική τιμή. Όλες αυτές οι μεταβλητές αν τους εκχωρήσουμε μια νέα τιμή θα κρατήσουν την καινούρια.

Αν θέλουμε να επανέλθει η προκαθορισμένη τιμή, είτε θα κάνουμε χρήση της εντολής **clear** είτε θα κλείσουμε το Matlab. Μερικές από αυτές είναι οι ακόλουθες:

ans : Προεπιλεγμένο όνομα που χρησιμοποιείται για αποτελέσματα

pi : Το *π*

inf : Αντιπροσωπεύει το ∞

NaN : Αντιπροσωπεύει μη αριθμό πχ το $\frac{0}{0}$

ί ή j : Αντιπροσωπεύει την φανταστική μονάδα

3 Πράξεις

Έχουμε την δυνατότητα, με το Matlab να κάνουμε βασικές μαθηματικές πράξεις όπως θα κάναμε με μια αριθμομηχανή, είτε χρησιμοποιώντας μόνο αριθμούς είτε και μεταβλητές που τις έχουμε αρχικοποιήσει με μια τιμή, τέτοια ώστε να έχει νόημα η πράξη.

3.1 Βασικές Πράξεις

Οι γνωστές μας πράξεις γίνονται με την χρήση των τελεστών +, -, *, /, ^, ενώ η σειρά υπολογισμού τους καθορίζεται από τους συνήθεις κανόνες προτεραιότητας.

Εκτός από τις παραδοσιακές πράξεις το Matlab υποστηρίζει σχεσιακές και λογικές πράξεις. Ως είσοδοι σε όλες τις σχεσιακές και λογικές παραστάσεις, οι μη μηδενικές τιμές θεωρούνται αληθείς και οι μηδενικές ψευδείς. Τα αποτελέσματα τέτοιων πράξεων παίρνουν τιμές 1 και 0.

3.2 Σχεσιακοί Τελεστές

Matlab	Μαθηματικά
>	>
<	<
>=	\geq
<=	\leq
==	=
$\sim =$	\neq

Προσοχή: Όταν συγκρίνουμε 2 τιμές χρησιμοποιούμε το == . Το = είναι ΜΟΝΟ για ανάθεση σε μια μεταβλητή

3.3 Λογικοί Τελεστές

\sim	Λογική Άρνηση
&&	Λογική Σύζευξη
	Λογική Διάζευξη

4 Εντολές Εισόδου & Εξόδου

4.1 Εντολή Εισόδου

Πολλές φορές κατά τη διάρκεια εκτέλεσης ενός προγράμματος, καλείται ο χρήστης να εισάγει δεδομένα. Αυτό είναι εφικτό με την χρήση της εντολής input

```
a = input('Give a number');
```

Όταν εκτελεστεί η συγκεκριμένη εντολή βλέπουμε στο command window το μήνυμα που έχουμε μέσα στην παρένθεση και περιμένει το πρόγραμμα να εισάγει ο χρήστης δεδομένα και να πατήσει enter, μετά η τιμή θα ανατεθεί στην μεταβλητή.

Αν και υπάρχει η δυνατότητα να εισαχθούν δεδομένα σε ένα πρόγραμμα του Matlab και με άλλους τρόπους. Όπως για παράδειγμα μέσω εισαγωγής δεδομένων από αρχεία, κάτι τέτοιο ξεφεύγει από τα όρια του μαθήματος.

4.2 Εντολές Εξόδου

Για την έξοδο δεδομένων στην οθόνη μας έχουμε πιο πολλές επιλογές. Αρχικά ας μιλήσουμε για την χρήση του ελληνικού ερωτηματικού (;) στο τέλος των πράξεων και των εντολών. Αν δεν βάλουμε ; στο τέλος των εντολών αυτό έχει σαν αποτέλεσμα την εκτύπωση των δεδομένων μας στην οθόνη. Ακόμα και όταν κάνουμε απλά μια ανάθεση μεταβλητών. Για αυτό καλό είναι να βάζουμε πάντα ερωτηματικό και αν θέλουμε να εμφανιστεί κάτι στην οθόνη να χρησιμοποιήσουμε κάποια από τις 2 εντολές που ακολουθούν.

► disp

```
disp('Message')
disp(variable)
```

Στην πρώτη περίπτωση εμφανίζεται το μήνυμα που βρίσκεται μέσα στην παρένθεση της disp. Στην δεύτερη εμφανίζεται η τιμή που έχει η μεταβλητή. Η disp δέχεται μόνο ένα όρισμα ανά κλήση.

5 Πίναχες

Ο βασικός τύπος δεδομένων στο Matlab είναι οι double πίνακες. Ένα βαθμωτό αναπαρίσταται σαν 1×1 πίνακας. Σαν διάνυσμα θα αναφέρουμε τους πίνακες $1\times n$ ή $n\times1$. Τα στοιχεία ενός πίνακα μπορούν να είναι :

- Αριθμοί
- Χαρακτήρες
- Συμβολοσειρές
- Άλλοι πίνακες και δομές

Όμως τα στοιχεία του κάθε πίνακα πρέπει να είναι του ίδιου τύπου. Υπάρχει βέβαια μια ειδική κατηγορία πινάκων, που δεν είναι απαραίτητη μια τέτοια ομοιομορφία, αλλα δε θα ασχοληθούμε με τέτοιες δομές.

Οι πίναχες του Matlab έχουν την δυνατότητα να προσαρμόζουν τις διαστάσεις τους ανάλογα. Έτσι αν προσθέσουμε ένα έξτρα στοιχείο, έχει την δυνατότητα να αλλάζει διαστάσεις έτσι ώστε να συμπεριλαμβάνει το νέο στοιχείο. Αυτό μας επιτρέπει σε σχέση με άλλες γλώσσες προγραμματισμού που μεριχές απαιτούν προχαθορισμένες διαστάσεις όταν αρχιχοποιούμε τους πίναχες.

5.1 Αρχικοποίηση Πινάκων

Λόγω της σημαντικότητας που έχει για το πακέτο η έννοια του πίνακα, περιμένουμε ότι η εισαγωγή δεδομένων σε μορφή πίνακα, θα γίνεται με εύχρηστο τρόπο. Πράγματι το Matlab έχει διάφορές μεθόδους με τις οποίες μπορούμε να αρχικοποιήσουμε έναν πίνακα. Στα πλαίσια των γραφικών θα περιοριστούμε στην απλή εισαγωγή.

5.2 Εισαγωγή Πίνακα

Έστω ότι θέλουμε να εισάγουμε τον πίναχα $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ αρχεί να γράψουμε στο command line την παραχάτω έχφραση:

$$A = [1 2; 3 4]$$

είτε εναλλακτικά

A = [1,2;3,4]

Όπως είναι εύχολο να αντιληφθούμε, το (;) χρησιμοποιείται όχι όπως έχουμε συνηθίσει ως τώρα, αλλά υποδηλώνει την αλλαγή γραμμής σε έναν πίναχα, ενώ τα στοιχεία της ίδιας γραμμής μπορώ να τα διαχωρίζω είτε αφήνοντας χενό αναμεσά τους είτε με την χρήση του (,).

Έτσι αν θέλουμε για παράδει
γμα να εισάγουμε το διάνυσμα $v = \begin{bmatrix} 4 & 1 & 7 \end{bmatrix}$ αρχεί να πληχτρολογίσου
με:

 $v = [4 \ 1 \ 7]$

Ένω αν θέλουμε το διάνυσμα $u = \begin{bmatrix} 5\\3\\8 \end{bmatrix}$ τότε θα πρέπει να γράψουμε:

u =[5; 3; 8]

5.3 Ανάκτηση στοιχείων πίνακα

Αν θέλουμε να δούμε την τιμή που έχει ένα συγκεκριμένο στοιχείο ενός διανύσματος ή ενός πίνακα αρκεί να γράψουμε το όνομα της μεταβλητής μας και μέσα σε παρένθεση την θέση του στοιχείου που μας ενδιαφέρει. Στο Matlab σε αντίθεση με τις περισσότερες γλώσσες προγραμματισμού που ξεκινάνε την αρίθμηση των θέσεων από την θέση 0. Η αρίθμηση γίνεται όπως γνωρίζουμε από τα μαθηματικά και ξεκινάει από την θέση 1.

Παράδειγμα 1: Αν έχουμε το διάνυσμα $v = \begin{bmatrix} 5 & 9 & 2 & 7 \end{bmatrix}$ και θέλουμε να φτιάξουμε να υπολογίσουμε το άθροισμα των στοιχείων του, θα χρειαστεί να τα πάρουμε ένα, ένα και να τα προσθέσουμε. Αυτό θα γινόταν με το παρακάτω πρόγραμμα:

Παράδειγμα 2: Αν έχουμε εισάγει τον παρακάτω πίνακα στο Matlab:

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

Και θέλουμε να εμφανίσουμε το στοιχείο που βρίσκεται στην θέση (2,3), αρκεί να γράψουμε την ακόλουθη εντολή στο command line.

disp(A(2,3))

6 Σχεδίαση στο Matlab

6.1 Καθορισμός του σημείου παρατήρησης

Πρώτα καλό είναι να αναφέρουμε και λίγα πράγματα για το πώς εμφανίζονται τα σχήματα στο γραφικό περιβάλλον της Matlab. Για να γίνουμε πιο ακριβείς από ποιο σημείο παρατηρούμε το κάθε σχήμα που σχεδιάζουμε. Το σημείο που βρίσκεται το υποθετικό μας μάτι και το σημείο στο οποίο εστιάζει γίνονται κατανοητά από την εικόνα.



Σχήμα 3: Σημείο παρατήρησης

Τα βέλη ορίζουν τις θετικές κατευθύνσεις. Το σημείο παρατήρησης (viewpoint) καθορίζεται πλήρως από το Azimuth και το Elevation. Αν θέλουμε να δούμε που βρίσκετε αρκεί να γράψουμε στο command line [az,el] = view και θα μας επιστρέψει τη θέση του. Ενώ το σημείο στο οποίο εστιάζει είναι πάντα το μέσο του παραθύρου στο οποίο βρίσκεται το γράφημα. Αν θέλουμε να αλλάξουμε τις τιμές που έχει όταν ξεκινάει το σχεδιασμό το πρόγραμμά μας αρκεί να γράψουμε view(az, el) ή view([azel]) ή view([x, y, z]) αν θέλουμε να δώσουμε το σημείο παρατήρησης σε καρτεσιανές συντεταγμένες. Αν θέλουμε να επαναφέρουμε τις αρχικές τιμές αρκεί να δώσουμε σαν εντολή

- Για δύο διαστάσεις view(2), az = 0, el = 90.
- Για τρεις διαστάσεις view(3), az = -37.5, el = 30.

Πάντα οι τιμές θα είναι σε μοίρες.

Εχτός από το viewpoint, σημαντικό ρόλο στο πως βλέπουμε μια εικόνα παίζει και το παράθυρο παρατήρησης. Αυτό καθορίζεται αυτόματα από το πρόγραμμα και έχει σαν όρια τις ακραίες τιμές του αντικειμένου ή των δεδομένων που εμείς οπτικοποιούμε. Βέβαια μπορούμε να αλλάξουμε τις διαστάσεις του. Αρκεί να χρησιμοποιήσουμε την εντολή axis. Για περισσότερες λεπτομέρειες για το ποιοι είναι όλοι οι δυνατοί τρόποι χρησιμοποίησης της, μπορούμε να αν πληκτρολογήσουμε στο command line help axis.

6.2 plot xai plot3

Ο ευχολότερος τρόπος σχεδιασμού δισδιάστατων χαμπύλων στο Matlab, είναι χρησιμοποιώντας την εντολή plot. Η plot σχεδιάζει σημεία στο επιπέδου. Αυτό όμως που την κάνει πολύτιμο εργαλείο, είναι η δυνατότητα να ενώνει δοσμένα διαδοχικά σημεία.

Παράδειγμα 3: Σχεδιάστε το τρίγωνο με κορυφές τα σημεία A = (0,0), B = (5,1), C = (3,5), θα περίμενε κανείς ότι αν γράψουμε στο Command Window την ακόλουθη εντολή θα έβγαινε το σχήμα.

plot([0 5 3],[0 1 5])



Σχήμα 4: Αποτέλεσμα της plot([0 5 3],[0 1 5])

Παρατηρούμε ότι δεν έχει κλείσει το σχήμα μας. Αυτό συμβαίνει γιατί η plot καταλαβαίνει ότι θα πρέπει να ενώσει το 1ο με το 2ο, το 2ο με το 3ο με τη σειρά που τα διαβάζει. Για να ενώσει και το 3ο με το 1ο, θα πρέπει να εισάγουμε τις συντεταγμένες του 1ου σημείου ξανά στο τέλος.

plot([0 5 3 0],[0 1 5 0])



Σχήμα 5: Αποτέλεσμα της plot([0 5 3 0],[0 1 5 0])

Αντίστοιχα αν θέλουμε να σχεδιάσουμε ένα σχήμα σε 3 διαστάσεις μπορούμε να χρησιμοποιήσουμε την εντολή plot3 όπου την δουλεύουμε ακριβώς σαν την plot απλά εισάγοντας και ένα 3ο διάνυσμα που αντιστοιχεί στην τρίτη συντεταγμένη των σημείων.

Αν και για τρισδιάστατα αντικείμενα μπορούμε να χρησιμοποιήσουμε άλλες εντολές όπως

- \bullet surf
- patch
- mesh

6.3 hold on

Αν θέλουμε στο ίδιο παράθυρο να εμφανίσουμε 2 διαφορετικά σχέδια, που θα προέρχονται από διαφορετικές κλήσεις εντολών όπως της plot πρέπει να χρησιμοποιήσουμε την εντολή **hold on**. Γιατί αλλιώς κάθε φορά που καλούμε μια εντολή σχεδίασης, σβήνει ότι υπάρχει στο figure.

7 Παράδειγμα μετασχηματισμού

Έστω το τρίγωνο στον \mathbb{R}^3 με κορυφές A = (0, 0, 1), B = (5, 1, 1), C = (3, 5, 1).Εφαρμόστε στροφή ως προς τον άξονα x κατά γωνία $\frac{\pi}{3}$ και στη συνέχεια μεταφορά κατά διάνυσμα $\vec{v} = (2, 3, 1)$. Και σχεδιάστε το μετασχηματισμένο σχήμα που προκύπτει.

Ορίζουμε τον πίναχα με τα σημεία του σχήματος, με την βοήθεια ομογενών συντεταγμένων. Κάθε σημείο του τριγώνου θα αποτελεί μια στήλη του πίναχα, και η τελευταία στήλη θα είναι ίδια με την πρώτη, έτσι ώστε εύχολα να μπορούμε να εισάγουμε κάθε γραμμή απευθείας στην plot και να έχουμε ένα χλειστό σχήμα.

Προσοχή : Αυτό το κάνουμε MONO εάν το σχήμα μας είναι κλειστό. Σε περίπτωση μιας τεθλασμένης γραμμής δεν χρειάζεται να προσθέσουμε έξτρα στήλη.

 $T = \begin{bmatrix} 0 & 5 & 3 & 0; 0 & 1 & 5 & 0; 1 & 1 & 1 & 1; 1 & 1 & 1 \end{bmatrix}$

Στην συνέχεια ορίζουμε τους 2 πίναχες μετασχηματισμών με βάση τους γνωστούς μας τύπους.

```
R_x = [1 0 0 0;
0 cos(pi/3) -sin(pi/3) 0;
0 sin(pi/3) cos(pi/3) 0;
0 0 0 1]
T_v = [1 0 0 2;
0 1 0 3;
0 0 1 1;
0 0 0 1]
```

Αν εφαρμόσουμε ότι γνωρίζουμε από την θεωρία. Το νέο μας σχήμα προκύπτει από τον ακόλουθο πολλαπλασιασμό πινάκων.

$New_T = T_v * R_x * T$

Για να σχεδιάσουμε και το αρχικό και το μετασχηματισμένο σχήμα στο ίδιο γράφημα αρκεί να γράψουμε στο Command Window τις 3 γραμμές κώδικα που ακολουθούν. Με την πρώτη σχεδιάζουμε το αρχικό σχήμα, με την δεύτερη εξασφαλίζουμε ότι αυτό δεν θα σβηστεί όταν με την εντολή της τρίτης γραμμής του ζητήσουμε να σχεδιάσει το καινούριο σχήμα.

```
Mε μπλέ χρώμα θα δούμε το αρχικό μας σχήμα και με κόκκινο το τελικό.
plot3(T(1,:),T(2,:),T(3,:))
hold on
plot3(New_T(1,:),New_T(2,:),New_T(3,:))
```



Σχήμα 6: Το πριν και το μετά του μετασχηματισμού