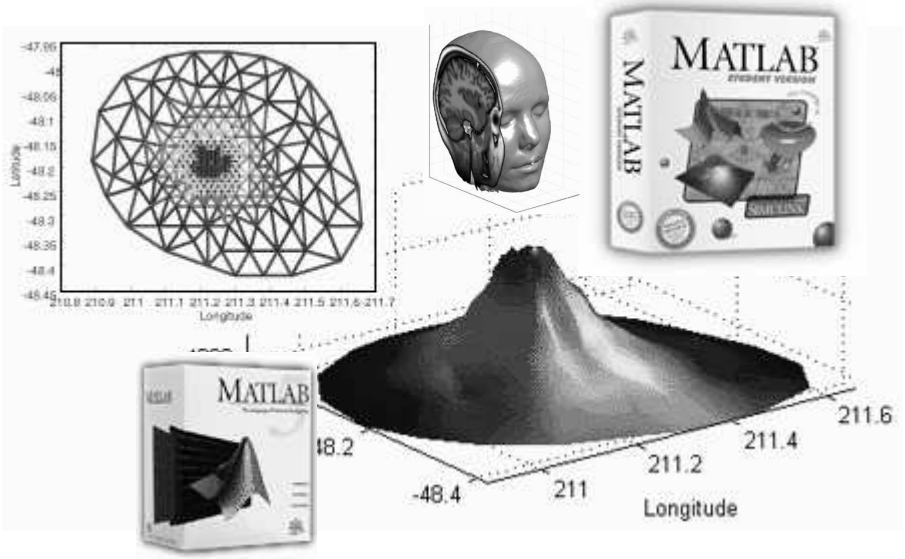
ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΜΑΘΗΜΑΤΙΚΩΝ

Ξεκινώντας με το

MATLAB

Revised by Clinton Wolfe
Original by David Hart



Προσαρμογή στα Ελληνικά από τους

Μ. Βάβαλη και Τ. Κατελανή

ΠΕΡΙΕΧΟΜΕΝΑ

1. Τί είναι το Matlab? 3
1.1. Πώς να χρησιμοποιήσετε αυτό το έγγραφο 3
1.2. Πού να βρείτε το Matlab 4
1.3. Πώς να ξεκινήσετε και να βγείτε από το Matlab 4
2. Συντακτικό 4
2.1. Πώς δουλεύει το Matlab 4
2.2. Σύμβολα και στίξη 5
2.3. Πώς να Ελέγχετε την Έξοδο 6
3. Περισσότερα Πάνω στους Πίνακες 7
3.1. Περισσότεροι τρόποι να κατασκευάζετε πίνακες 7
3.1.1. Ενσωματωμένοι τρόποι κατασκευής 7
3.1.2. Αλληλουχία 7
3.1.3. Προγραμματισμός 8
3.2. Περισσότερες λειτουργίες πινάκων 8
3.2.1. Βαθμωτά μεγέθη 8
3.2.2. Διανύσματα 9
3.2.3. Πίνακες 9
3.3. Λύνοντας Εξισώσεις 10
3.4. Σώζοντας και φορτώνοντας πίνακες 10
4. Γραφήματα 11
4.1. Έννοιες Γραφημάτων 11
4.2. Βασικά 2-διάστατα Γραφήματα 12
4.3. Βασικά 3-διάστατα Γραφήματα 13
4.4. Προχωρημένη Σχεδίαση 14
5. Προγραμματισμός 14
5.1. Πώς να προγραμματίζετε 14
5.2. Συναρτήσεις 14
5.3. Μαζικός Προγραμματισμός 15
5.4. Ιδέες προγραμματισμού 15
6. Τα επόμενα βήματα 16

1. Τί είναι το Matlab

Το MATLAB είναι ένα πρόγραμμα υπολογιστών για ανθρώπους που χρησιμοποιούν αριθμητικούς υπολογισμούς, ειδικά στη γραμμική άλγεβρα (πίνακες). Ξεκίνησε ως ένα πρόγραμμα "Εργαστηρίου Πινάκων" ("MATrixLABoratory") που είχε σκοπό να παρέχει αλληλεπιδρώσα προσπέλαση στις βιβλιοθήκες Linpack και Eispack. Από τότε έχει αναπτυχθεί αρκετά, για να γίνει ένα ισχυρότατο εργαλείο στην οπτικοποίηση, στον προγραμματισμό, στην έρευνα, στην επιστήμη των μηχανικών, και στις επικοινωνίες.

Στο δυναμικό του Matlab συμπεριλαμβάνονται μοντέρνοι αλγόριθμοι, δυνατότητες χειρισμού τεράστιων ποσοτήτων δεδομένων, και ισχυρά προγραμματιστικά εργαλεία. Το Matlab δεν είναι σχεδιασμένο για συμβολικούς υπολογισμούς, αλλά αντισταθμίζει αυτή την αδυναμία του επιτρέποντας στο χρήστη να συνδέεται άμεσα με το Maple. Η επιφάνεια αλληλεπίδρασης βασίζεται κυρίως σε κείμενο, γεγονός που μπορεί να συγχύσει μερικούς χρήστες.

Το Matlab έρχεται ως πακέτο του βασικού προγράμματος, με πολλές "εργαλειοθήκες", που πωλούνται ξεχωριστά. Θα καλύψουμε μόνο το βασικό πακέτο. Η τρέχουσα έκδοση (Ιούνιος 1999) είναι το Matlab 5.3 έκδοση 11.

1.1. Πώς να χρησιμοποιήσετε αυτό το έγγραφο

Αυτό το έγγραφο προορίζεται να χρησιμοποιείται όταν κάθεστε σε έναν υπολογιστή που τρέχει είτε NT είτε το σύστημα των X windows. Υποτίθεται ότι θα δώσετε τις εντολές που εμφανίζονται, και στη συνέχεια θα περιμένετε το αποτέλεσμα.

Ο αναγνώστης αυτού του εγγράφου θα πρέπει να έχει τουλάχιστον μία μικρή εξοικείωση με τη γραμμική άλγεβρα και να χρησιμοποιεί άνετα τους υπολογιστές. Προκειμένου να γίνουμε ευρύτερα κατανοητοί, δεν θα καλύψουμε μηχανικά θέματα (π.χ. επεξεργασία σημάτων, φασματική ανάλυση), ωστόσο όμως θα επισημάνουμε τις σημαντικές διαφορές ανάμεσα στα διάφορα πακέτα προς αυτή την κατεύθυνση.

Αν χρησιμοποιείτε το Matlab σε ένα τερματικό που βασίζεται σε κείμενο, ίσως να θέλετε να ακολουθήσετε το εγχειρίδιο εκμάθησης Χρησιμοποιώντας Λογισμικό Μαθηματικών σε UNIX όταν θα τελειώσετε με αυτό.

Καθ' όλο αυτό το έγγραφο, θα χρησιμοποιούμε τις παρακάτω συμβάσεις.

Παράδειγμα	**Επεξήγηση**
File Open	Διαλέξτε τον κατάλογο αρχείων, και επιλέξτε Open
`A = 5`	Είσοδος που πρέπει να πληκτρολογηθεί στην προτροπή του Matlab
`A = 5`	Έξοδος του Matlab.
	Χρήσιμη συμβουλή.

1.2. Πού να βρείτε το Matlab

Το MATLAB διατίθεται για πολλά διαφορετικά συστήματα υπολογιστών στο Μαθηματικό Τμήμα του Πανεπιστημίου Κρήτης.

- Για όλα τα Τεχνολογικά Κέντρα Φοιτητών με Windows
- Όλες οι ομάδες UNIXκαι LINUX

Μία έκδοση για μαθητές διατίθεται από τα τοπικά βιβλιοπωλεία για τα προσωπικά σας συστήματα Windows, Macintosh, και Linux.

1.3. Πώς να ξεκινήσετε και να βγείτε από το Matlab

Για να ξεκινήσετε το Matlab:

- Σε windows NT, επιλέξτε Start Menu ->Programs->Statistics-Math->Matlab.
- Σε συστήματα UNIX, πληκτρολογείστε `matlab` για να το τρέξετε με αλληλεπίδραση. Παρατηρείστε ότι τα σχήματα θα εμφανισθούν σε ένα Xwindow, αν έχετε θέσει τη μεταβλητή DISPLAY. Μπορείτε ακόμα να πληκτρολογήσετε `matlab -display hostname:0`.

Σημειώστε ότι το Matlab συνήθως χρειάζεται 10-15 δευτερόλεπτα για να φορτώσει. Τελικά, θα πρέπει να δείτε:

```
To get started, type one of these: helpwin, helpdesk, or demo.
For product information, type tour or visit www.mathworks.com.
>>
```

Η γραμμή >> είναι η προτροπή του Matlab.
Για να βγείτε από το Matlab, πληκτρολογείστε (στην προτροπή του matlab) `quit` ή επιλέξτε File->Exit MATLAB.

2. Συντακτικό

2.1. Πώς δουλεύει το Matlab

Το Matlab δουλεύει εκτελώντας τις μαθηματικές εντολές που εισάγετε στο παράθυρο εντολών. Η προκαθορισμένη επιλογή είναι, κάθε έξοδος να τυπώνεται απευθείας στο παράθυρο.

Ακόμα, σας επιτρέπεται να εκχωρείτε ένα όνομα σε μία έκφραση για δική σας ευκολία. Να θυμάστε ότι το όνομα που εκχωρείτε είναι μόνο ένα όνομα, και δεν αναπαριστά καμία μαθηματική μεταβλητή (όπως θα έκανε στο Maple, για παράδειγμα). Κάθε όνομα πρέπει να έχει μία τιμή κάθε στιγμή. Αν προσπαθήσετε να διαβάσετε την τιμή ενός μη προσδιορισμένου ονόματος, θα πάρετε μήνυμα λάθους.

Σχεδόν τα πάντα στο Matlab είναι πίνακες, είτε μοιάζουνε με τέτοιους είτε όχι. Αυτό μπορεί να σας πάρει λίγο μέχρι να το συνηθίσετε. Θα εισάγουμε τις λειτουργίες με το στυλ πίνακα μαζί με τα αριθμητικά τους ισοδύναμα, ώστε να μπορείτε να καταλάβετε τα καλούπια που παρουσιάζονται στο συντακτικό.

2.2. Σύμβολα και στίξη

Το Matlab σχεδιάστηκε για να χρησιμοποιεί σαφώς προκαθορισμένη σημειογραφία. Δοκιμάστε τα παρακάτω παραδείγματα στον υπολογιστή σας.

Είσοδος	**Έξοδος**	**Σχόλια**

`2 + 3`
`7-5`
`34*212`
`1234/5786`
`2^5`
	`ans = 5`
	`ans = 2`
	`ans = 7208`
	`ans = 0.2173`
	`ans = 32`
		Η αριθμητική δουλεύει όπως αναμενόταν. Σημειώστε ότι στο αποτέλεσμα δίνεται το όνομα "ans" κάθε φορά.

`a = sqrt(2)`	`a = 1.4142`	Μπορείτε να επιλέξετε τα ονόματα που θέλετε για τα διάφορα αντικείμενα.

`b = a, pi, 2 + 3i`	`b = 1.4142`
	`ans = 3.1416`
	`ans = 2.0000 + 3.0000i`
		Μπορείτε να χρησιμοποιήσετε κόμματα για να βάλετε περισσότερες από μία εντολές σε μία γραμμή. Τα Pi, i, και j είναι σημεία επαφής.

`c = sin(pi)` `eps`	`c = 1.2246e-016` `ans = 2.2204e-016`	Το "eps" είναι το τρέχον όριο ακρίβειας. Οτιδήποτε μικρότερο από "eps" είναι πιθανότατα μηδέν. Σημειώστε ότι το Matlab καταλαβαίνει (και περιμένει από εσάς να καταλαβαίνετε!) την επιστημονική σημειογραφία.

`d = [1 2 3 4 5 6 7 8 9]`
`e = [1:9]`
`f = 1:9`
	`d = 1 2 3 4 5 6 7 8 9`
	`e = 1 2 3 4 5 6 7 8 9`
	`f = 1 2 3 4 5 6 7 8 9`
		Τα "d", "e", και "f" είναι όλα διανύσματα. Είναι ισοδύναμα. Παρατηρείστε τη χρήση του τελεστή ":" - μετράει (με μονάδες) από ένα νούμερο στο επόμενο.

`g = 0:2:10`
`f(3)`
`f(2:7)`
`f(:)`
	`g = 0 2 4 6 8 10`
	`ans = 3`
	`ans = 2 3 4 5 6 7`
	`1 2 3 4 5 6 7 8 9`
		Περισσότερες χρήσεις του συμβόλου ":". Παρατηρείστε ότι μπορείτε να το χρησιμοποιείτε για να πάρετε ένα κομμάτι ενός διανύσματος (ή πίνακα, ή κύβου, κλπ), ή για να το πάρετε ολόκληρο.

`h = [1 2 3];`
`h'`(nothing)
	`ans = 1`
	`2`
	`3`
		Ένα ερωτηματικό ";" θα αποτρέψει την εμφάνιση της εξόδου. Ένα απλό σύμβολο " ' " υπολογίζει την

		αλληλομετάθεση ενός πίνακα, ή σε αυτή την περίπτωση, αλλάζει θέση μεταξύ διανυσμάτων-γραμμών και διανυσμάτων-στηλών.

`h * h'`
`h .* h`
`h + h`
	`ans = 14`
	`ans = 1 4 9`
	`ans = 2 6 8`
		Μαθηματικές πράξεις σε διανύσματα. Το * είναι πολλαπλασιασμός πινάκων, επομένως οι διαστάσεις πρέπει να ταιριάζουν.
		Το " .* " είναι πολλαπλασιασμός εισόδου με είσοδο.

`g = [ 1 2 3; 4 5 6; 7 8 9]`	`g = 1 2 3`
	`4 5 6`
	`7 8 9`
		Εισάγοντας έναν πίνακα.

`g(2,3)`
`g(3,:)`
`g(2,3) = 4`
	`ans = 6`
	`ans = 7 8 9`
	`g = 1 2 3`
	`4 5 4`
	`7 8 9`
		Προσπελαύνοντας τα στοιχεία ενός πίνακα.
		Παρατηρείστε τη χρήση του συμβόλου ":" για να προσπελάσετε μία ολόκληρη γραμμή.

`g^2`
 	
`g .^ 2`
	`ans = 30 36 42`
	`66 81 96`
	`102 126 150`
	`ans = 1 4 9`
	`16 25 36`
	`49 64 81`
		Το πρώτο πολλαπλασιάζει τον πίνακα με τον εαυτό του. Το δεύτερο υψώνει στο τετράγωνο κάθε είσοδο του πίνακα.

2.3. Πώς να Ελέγχετε την Έξοδο

Πριν προχωρήσουμε σε πιο εξειδικευμένα πράγματα στους πίνακες, θα ήταν φρόνιμο να αναφέρουμε θέματα μορφοποίησης.

Η εντολή που πρέπει να ξέρετε είναι `format` .

- Για να ελέγχετε τα κενά μεταξύ των γραμμών, χρησιμοποιείστε το `format compact` .
- Για να δείτε και τα 15 ψηφία που χρησιμοποιήθηκαν στον υπολογισμό, χρησιμοποιείστε το `format long` .
- Για να δείτε μόνο 5 ψηφία, χρησιμοποιείστε το `format short` .
- Για να αποτρέψετε εντελώς την έξοδο, χρησιμοποιείστε ένα ερωτηματικό στο τέλος της εντολής.

Για να δείτε κι άλλες επιλογές, πληκτρολογείστε `help format` .

Σημειώστε ότι το Matlab *πάντα* χρησιμοποιεί "διπλή" ακρίβεια (περίπου 15 ψηφία) στους υπολογισμούς του. Αυτές οι εντολές ρυθμίζουν μερικώς τον τρόπο που εμφανίζεται η έξοδος.

3. Περισσότερα Πάνω στους Πίνακες

3.1. Περισσότεροι τρόποι να κατασκευάζετε πίνακες

3.1.1. Ενσωματωμένοι τρόποι κατασκευής

Υπάρχουν πολλοί ενσωματωμένοι τρόποι κατασκευής πινάκων. Ακολουθούν μερικοί:

Είσοδος	Έξοδος	Σχόλια

`rand(2)`
 	
`rand(2,3)`
	`ans = 0.9501 0.6068`
	`0.2311 0.4860`
	`ans = 0.8913 0.4565 0.8214`
	`0.7621 0.0185 0.4447`
		Δημιουργεί έναν πίνακα με στοιχεία εισόδου τυχαία κατανεμημένα ανάμεσα στα 0 και 1

`zeros(2)`
 	
`ones(2)`
	`ans = 0 0`
	`0 0`
	`ans = 1 1`
	`1 1`
		Δημιουργεί έναν 2x2 πίνακα με όλα τα στοιχεία εισόδου μηδενικά (ή ίσα με τη μονάδα).

`eye(2)`	`ans = 1 0` `0 1`	Ταυτοτικός πίνακας I.

`hilb(3)`	`ans = 1.0000 0.5000 0.3333`
	`0.5000 0.3333 0.2500`
	`0.3333 0.2500 0.2000`	

Για να πάρετε περισσότερες πληροφορίες σχετικά με το θέμα, δείτε τη σελίδα βοήθειας για ειδικούς πίνακες- πληκτρολογείστε `help elmat` .

3.1.2. Αλληλουχία

Μπορούν να σχηματιστούν νέοι πίνακες από τους παλιούς. Υποθέστε ότι έχουμε

```
a = [1 2; 3 4]
a = 1 2
    3 4
```

Τότε μπορούμε να φτιάξουμε έναν καινούργιο πίνακα με οποιονδήποτε από τους παρακάτω τρόπους.

Είσοδος	Έξοδος
`[a, a, a]`	`ans = 1 2 1 2 1 2` `3 4 3 4 3 4`

`[a; a; a]`	`ans = 1 2`
	`3 4`
	`1 2`
	`3 4`
	`1 2`
	`3 4`

`[a, zeros(2); zeros(2), a']`	`ans = 1 2 0 0`
	`3 4 0 0`
	`0 0 1 3`
	`0 0 2 4`

3.1.3. Προγραμματισμός

Οι πίνακες μπορούν επίσης να κατασκευαστούν μέσω προγραμματισμού. Ακολουθεί ένα παράδειγμα που χρησιμοποιεί δύο βρόγχους "for".

```
for i=1:10,
   for j=1:10,
      t(i,j) = i/j;
   end
end
```

Παρατηρείστε ότι δεν υπάρχει έξοδος, αφού η μόνη εντολή που θα μπορούσε να παράγει έξοδο (`t(i,j) = i/j;`) τελειώνει με ερωτηματικό. Χωρίς το ερωτηματικό, το Matlab θα τύπωνε τον πίνακα t 100 φορές!

Θα καλύψουμε το κομμάτι του προγραμματισμού με περισσότερες λεπτομέρειες αργότερα.

3.2. Περισσότερες λειτουργίες πινάκων

Όπως είδαμε και ενωρίτερα, τα +, -, *, και / ορίζονται διαισθητικά για τους πίνακες. Όταν υπάρχει αμφιβολία για το αν μία λειτουργία θα εκτελεί αριθμητικές πράξεις με τους πίνακες (σε αντίθεση με τις αριθμητικές πράξεις μεταξύ των στοιχείων εισόδου ένα-προς-ενα), να θυμάστε ότι το ".*" (τελεία-αστεράκι) θα πολλαπλασιάζει τα στοιχεία εισόδου ένα προς ένα, και το "*" (αστεράκι) θα εκτελεί πολλαπλασιασμούς πινάκων.

3.2.1. Βαθμωτά μεγέθη

Ένα βαθμωτό μέγεθος είναι απλώς ένας αριθμός. Το Matlab τους αποθηκεύει εσωτερικά ως πίνακες, αλλά τους μεταχειρίζεται σαν να ήταν αριθμοί.

Όλες οι λειτουργίες που εμπλέκουν ένα βαθμωτό μέγεθος και έναν πίνακα επηρεάζουν ένα-προς-ενα τα στοιχεία εισόδου του πίνακα, με μία εξαίρεση: τον τελεστή της δύναμης ("^"). Με το a όπως ορίστηκε παραπάνω, δοκιμάστε τα παρακάτω:

Είσοδος	**Έξοδος**	**Σχόλια**
`b=2`	`b=2`	Ορίζει το b να είναι ένα βαθμωτό μέγεθος.
`a + b`	`ans = 3 4` `5 6`	Η πρόσθεση δουλεύει στοιχείο-προς-στοιχείο.
`a * b`	`ans = 2 4` `6 8`	Το ίδιο και ο πολλαπλασιασμός.

`a ^ b`	`ans = 7 10` `15 22`	Αυτή είναι ύψωση *πίνακα* σε δύναμη - a*a
`a .^ b`	`ans = 1 4` `9 16`	Ύψωση σε δύναμη στοιχείου-προς-στοιχείο.

3.2.2. Διανύσματα

Ένα διάνυσμα είναι απλά ένας πίνακας με μία μόνο γραμμή (ή μία στήλη). Το Matlab κάνε μία διάκριση ανάμεσα στα διανύσματα-γραμμή και στα διανύσματα-στήλη, και θα βγάλει ανάλογο μήνυμα αν δεν πάρει αυτό που περίμενε.

Σε όλες τις αριθμητικές λειτουργίες, το Matlab μεταχειρίζεται ένα διάνυσμα ως πίνακα, οπότε γνωρίζουμε ήδη πώς να πολλαπλασιάζουμε ένα διάνυσμα με ένα βαθμωτό μέγεθος. Το Matlab επίσης σας δίνει τις παρακάτω λειτουργίες:

Είσοδος	**Έξοδος**	**Σχόλια**
`v = [1 2 3]` `u = [3 2 1]`	`v = [1 2 3]` `u = [3 2 1]`	Ορίζει ένα ζευγάρι διανυσμάτων
`v * u`	`Error`	Οι διαστάσεις δε συμφωνούν.
`v * u'`	`ans = 10`	Δουλεύει αν πάρουμε την αλληλομετάθεση.
`dot(v,u)`	`ans = 10`	Το εσωτερικό γινόμενο είναι το ίδιο πράγμα.
`cross(v,u)`	`ans = -4 8 -4`	Το εξωτερικό γινόμενο δουλεύει μόνο για 3-διάστατα διανύσματα.

3.2.3. Πίνακες

Το Matlab έχει όλες τις συνηθισμένες λειτουργίες ενσωματωμένες, καθώς και τις περισσότερες από τις πιο ασήμαντες.

Είσοδος	**Έξοδος**	**Σχόλια**

`k = 16 2 3`
`5 11 10`
`9 7 6`
	`k = 16 2 3`
	`5 11 10`
	`9 7 6`
		Ορίζει έναν πίνακα.

`rref(k)`	`ans = 1 0 0`
	`0 1 0`
	`0 0 1`
		Τύπος ελαχιστοποίησης γραμμών στο επίπεδο.

`rank(k)`	`ans = 3`	Η τάξη.
`det(k)`	`ans = -136`	Η ορίζουσα.

`inv(k)`	`ans = 0.0294 -0.0662 0.0956`
	`-0.4412 -0.5074 1.0662`
	`0.4706 0.6912 -1.2206`
		Ο αντίστροφος του πίνακα.

`[vec,val] = eig(k)`	`vec = -0.4712 -0.4975 -0.0621`
	`-0.6884 0.8282 -0.6379`
	`-0.5514 0.2581 0.7676`
	`val = 22.4319 0 0`
	`0 11.1136 0`
	`0 0 -0.5455`
		Τα ιδιοδιανύσματα και οι ιδιοτιμές του πίνακα.
		Οι στήλες του "vec" είναι τα ιδιοδιανύσματα,
		και τα διαγώνια στοιχεία του "val" είναι οι ιδιοτιμές

Υπάρχουν πολλές ακόμα συναρτήσεις όπως αυτές. Πληκτρολογείστε `help matfun` για να τις δείτε όλες.

3.3. Λύνοντας Εξισώσεις

Μία από τις κύριες χρήσεις των πινάκων είναι η αναπαράσταση συστημάτων γραμμικών εξισώσεων. Αν ο `a` είναι ένας πίνακας που περιέχει τους συντελεστές ενός συστήματος γραμμικών εξισώσεων, το `x` είναι ένα διάνυσμα-στήλη που περιέχει τους "αγνώστους", και το `b` είναι το διάνυσμα στήλη της "δεξιάς πλευράς", οι σταθεροί όροι, τότε η εξίσωση πινάκων $a\ x = b$ αναπαριστά το σύστημα των εξισώσεων.

Αν ξέρετε λίγη γραμμική άλγεβρα, θα μπορούσατε να χρησιμοποιήσετε τις εντολές του Matlab που παρουσιάστηκαν πιο πάνω για να επαυξήσετε τον πίνακα και στη συνέχεια να βρείτε τον τύπο ελαχιστοποίησης γραμμών στο επίπεδο. Παρόλα αυτά, το Matlab παρέχει έναν πιο απλό μηχανισμό για την επίλυση γραμμικών εξισώσεων:

```
x = a \ b
```

Αυτό μπορεί να διαβαστεί ως "το x ισούται με το a-επί τον αντίστροφο του b."
Δοκιμάστε τον με τους πίνακες

```
a = [1 2 3; 4 5 6; 7 8 10]; b = [1 1 1]';
```

Θα πρέπει να πάρετε

```
x = -1 1 0
```

Για να επαληθεύσετε αυτό τον ισχυρισμό, δοκιμάστε το παρακάτω:

```
a*x, a*x - b, eps
```

Τα αποτελέσματα είναι:

```
ans = 1 1 1
ans = 1.0e-015 *
    -0.1110
    -0.6661
    -0.2220
ans = 2.2204e-016
```

Παρατηρείστε ότι το `a*x - b` βρίσκεται πολύ κοντά στο `eps` - που σημαίνει ότι είναι όσο πιο κοντά στο μηδέν γίνεται.

Αν δεν υπάρχει λύση, παρέχεται μία λύση "ελαχίστων τετραγώνων" (το a*x - b είναι το μικρότερο δυνατό). Εισάγετε

```
a(3,3) = 9; b = [1 1 0]';
```

(που κάνει τον πίνακα ιδιάζον και αλλάζει το b) και εισάγετε ξανά τις παραπάνω εντολές, χρησιμοποιώντας το επάνω βελάκι για να τις επαναφέρετε. Παρατηρείστε ότι η λύση είναι λίγο ανακριβής.

3.4. Σώζοντας και φορτώνοντας πίνακες

Σε αντίθεση με το Maple και το Mathematica, το Matlab δεν έχει την έννοια του φύλλου εργασίας ("worksheet"). Όταν βγείτε από τοMatlab, δεν θα προτραπείτε να σώσετε την εργασία

σας.

Αν θέλετε ένα αντίγραφο της δουλειάς σας, μπορείτε να γυρίσετε σε "logging" πληκτρολογώντας

`diary 'c:\scratch\session.txt'` σε μηχανήματα NT
`diary '~/session.txt'` σε UNIX.

Παρατηρείστε ότι η έξοδος θα σωθεί παράλληλα με την είσοδο, οπότε το αρχείο δε μπορεί να χρησιμοποιηθεί άμεσα ως έγγραφο.

Μπορεί απλά να θέλετε να σώσετε έναν ή περισσότερους πίνακες. Η εντολή diary είναι ένας πολύ άχαρος τρόπος για να το κάνετε αυτό.
Για να σώσετε την τιμή της μεταβλητής "x" σε ένα αρχείο απλού κειμένου που ονομάζεται "x.value" χρησιμοποιείστε την
`save x.value x -ascii`
Για να σώσετε όλες τις μεταβλητές σε ένα αρχείο που λέγεται "mysession.mat" σε μορφή επαναφόρτωσης, χρησιμοποιείστε την
`save mysession`
Για να επανακτήσετε αυτό την προσπάθειά σας, χρησιμοποιείστε την
`load mysession`
Στη συνέχεια, για να δείτε τα σωσμένα αρχεία, σε συστήματα UNIX πληκτρολογείστε τις εντολές:

```
% more session.txt
% more x.value
```

Σε σύστημα Windows NT, ανοίξτε το κατάλληλο αρχείο με το Notepad.

4. Γραφήματα

Το Matlab έχει εξαιρετικές δυνατότητες γραφικών (πρέπει να χρησιμοποιείτε κάποιο τερματικό που υποστηρίζει γραφικά για να χρησιμοποιείτε τις δυνατότητες). Παρόλα αυτά, χρησιμοποιώντας γραφικά στο Matlab είναι θεμελιωδώς διαφορετικό από το να χρησιμοποιείτε γραφικά στο Maple ή στο Mathematica.

4.1. Έννοιες Γραφημάτων

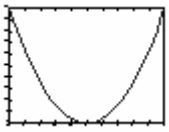
Πριν να δούμε τις δυνατότητες καταγραφής σχεδίου του Matlab, σκεφτείτε τί *είναι* στην πραγματικότητα ένα γράφημα. Ένα γράφημα είναι ένα σύνολο σημείων, στις 2, 3 ή ακόμα και 4 διαστάσεις, τα οποία μπορεί να είναι ή και να μην είναι συνδεδεμένα με γραμμές ή πολύγωνα. Τα περισσότερα μαθηματικά πακέτα λογισμικού το κρύβουν αυτό από το χρήστη δειγματοληπτώντας από μία συνεχή συνάρτηση για να παράγει τα σημεία.

Το Matlab είναι σχεδιασμένο για να δουλεύει με πίνακες, και όχι με συναρτήσεις. Οι πίνακες είναι ένας αξιόπιστος τρόπος να αποθηκεύετε ένα σύνολο από νούμερα - που είναι ακριβώς αυτό που χρειάζεται όταν χρησιμοποιείτε γραφήματα. Άρα όλες οι εντολές για γραφήματα στο Matlab δέχονται *πίνακες* ως ορίσματά τους, σε αντίθεση με μία συνάρτηση. Αν έχετε συνηθίσει τα διαγράμματα σε μορφή συναρτήσεων, το Matlab μπορεί να σας πάρει κάποιο διάστημα να το συνηθίσετε. Από την άλλη μεριά, ο τρόπος που η προσέγγιση του Matlab κάνει πολύ εύκολο το να δει κανείς τα δεδομένα και να δημιουργήσει γραφήματα που βασίζονται σε λίστες σημείων.

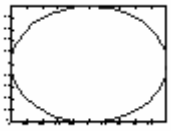
Ένα άλλο εξαιρετικό χαρακτηριστικό του συστήματος γραφημάτων του Matlab είναι ο τρόπος με τον οποίο εμφανίζει τη γραφική έξοδο. Στο Matlab, υπάρχει (συνήθως) μόνο ένα παράθυρο για το διάγραμμα. Οι διαδοχικές εντολές για τη δημιουργία του διαγράμματος θα προστεθούν στο παλιό διάγραμμα, εκτός και αν ζητήσετε να δημιουργηθεί ένα καινούργιο. Αυτό επιτρέπει να φτιαχτεί ένα διάγραμμα, και στη συνέχεια να το ρυθμίσετε ώστε να ταιριάζει στις ανάγκες σας.

4.2. Βασικά 2-διάστατα Γραφήματα

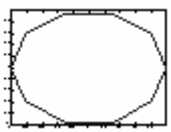
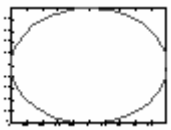
Τώρα που καταλαβαίνετε μερικά παραπάνω πράγματα για τα γραφήματα, δοκιμάστε αυτά τα παραδείγματα. Βεβαιωθείτε ότι τα ακολουθείτε πιστά, αλλιώς μπορεί να μην πάρετε τα ίδια αποτελέσματα.

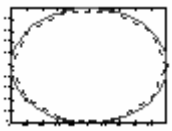
Είσοδος	**Έξοδος**	**Σχόλια**
`x = -10:0.5:10;` `y = x .^ 2;`	(καμία)	Δημιουργεί έναν 1x41 πίνακα που μετράει από το -10 στο ως 10, και έναν 1x41 πίνακα που προκύπτει υψώνοντας στο τετράγωνο τα στοιχεία του πρώτου πίνακα.
`plot(x,y)`		Σχεδιάζει τα σημεία. Κάθε ζευγάρι είναι σχεδιασμένο, ώστε το <x(1), y(1)> να είναι ένα σημείο, το <x(2), y(2)> να είναι ένα σημείο, κλπ.

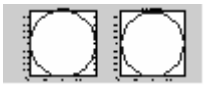
`t = 0:0.1:2*pi;`
`x = cos(t);`
`y = sin(t);`
	(καμία)	Παράγει κάποιους καινούργιους 1x63 πίνακες.

`plot(x,y)`		Σχεδιάζει τα σημεία. Αυτό λέγεται παραμετρική σχεδίαση. Παρατηρείστε ότι αντικαθιστά το προηγούμενο διάγραμμα.

`t = 0:pi/5:2*pi;`
`u = cos(t);`
`v = sin(t);`
	(καμία)	Παράγει κάποιους καινούργιους 1x11 πίνακες. Δείχνει πώς να ελέγχετε την "ανάλυση" του διαγράμματος.

`figure plot(u,v)`		Δημιουργεί ένα νέο παράθυρο για το διάγραμμα, και σχεδιάζει τα σημεία. Παρατηρείστε πόσο ανομοιόμορφος είναι ο πίνακας, αφού χρησιμοποιήσαμε μόνο 11 δειγματικά σημεία.
`plot(x,y, 'ro-')`		Σχεδιάζει και πάλι την επίμαχη έκδοση με κόκκινο, με κύκλους στα στοιχεία δεδομένων, που συνδέονται με ευθείες.

`plot(x,y, 'r-', u,v,'b*:')`		Στο ίδιο διάγραμμα, σχεδιάζει την επίμαχη έκδοση με κόκκινο, και σχεδιάζει τη χαμηλής προτεραιότητας έκδοση με μπλε αστεράκια στα στοιχεία δεδομένων και με διακεκομμένες γραμμές.

`figure`
`subplot(1,2,1)`
`plot(x,y)`
`title('Fine Mesh')`
`subplot(1,2,2)`
`plot(u,v)`
`title('Coarse Mesh')`
		Δημιουργεί ένα καινούργιο σχήμα.
		Το χωρίζει σε μία γραμμή, δύο στήλες, και δίνει βάση στο πρώτο τετράγωνο.
		Σχεδιάζει.
		Δίνει στο τρέχον σχέδιο έναν τίτλο.
		Δίνει βάση στο δεύτερο τετράγωνο.
		Σχεδιάζει.
		Του δίνει έναν τίτλο.

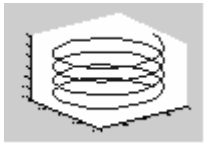
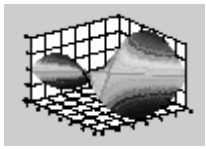
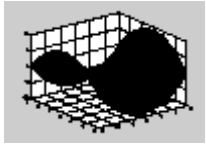
Δείτε τα παρακάτω αρχεία βοήθειας για περισσότερες επιλογές και ιδέες: `help plot`, `help comet`, `help semilogy` και `help fill` .

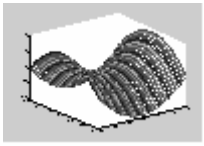
Το Matlab παρέχει πολύ ισχυρά χαρακτηριστικά στο παράθυρο σχεδίασης. Χρησιμοποιείστε τη μπάρα εργαλείων στην κορυφή για να προσθέσετε βέλη, γραμμές, και κείμενο με σχόλια στο διάγραμμά σας.

4.3. Βασικά 3-διάστατα Γραφήματα

Είσοδος	**Έξοδος**	**Σχόλια**

`t = -4*pi:pi/16:4*pi;`
`x = cos(t);`
`y = sin(t);`
`z = t;`
	(καμία)	Παράγει τα δεδομένα...

`plot3(x,y,z)`		... και ζωγραφίζει ένα ελικοειδές..
`[x, y] = meshgrid(-3:0.1:3,-3:0.1:3);` `z = x.^2 - y.^2;`	(καμία)	Παράγει δεδομένα για μία επιφάνεια διαγράμματος.
`mesh(x,y,z)`		Ζωγραφίζει την επιφάνεια χρησιμοποιώντας ένα πλέγμα.
`surf(x,y,z)`		Ζωγραφίζει μία επιφάνεια ως μία "μπαλωμένη" επιφάνεια.

`plot3(x,y,z)`		Παρατηρείστε ότι εξακολουθεί να τη σχεδιάζει, αλλά ως ένα σύνολο από τοξοειδείς καμάρες.

4.4. Προχωρημένη Σχεδίαση

Φυσικά, το Matlab μπορεί να κάνει πολύ περισσότερα από αυτές τις απλές ασκήσεις. Αν θέλετε να μάθετε περισσότερα, δοκιμάστε τα παρακάτω αρχεία βοήθειας:
`help slice`, `help movie`, `help getframe`, `help graph2d`, `help graph3d`, `help graphics`
.

5. Προγραμματισμός

5.1. Πώς να προγραμματίζετε

Μπορεί κανείς να δημιουργήσει ένα πρόγραμμα Matlab χρησιμοποιώντας τον editor της αρεσκείας του, όπως και μπορεί να το αποθηκεύσει σε ένα αρχείο για να το χρησιμοποιήσει αργότερα. Τέτοια αρχεία συνήθως ονομάζονται "m-αρχεία" (Matlab αρχεία) και πρέπει το όνομα τους να τελειώνει με ".m" (π.χ. `foo.m`). Η χρήση m-αρχείων αποτελεί καλή πρακτική προγραμματισμού στην Matlab, ειδικά εάν πρόκειται να χρησιμοποιήσετε ένα σύνολο εντολών πολλές φορές, και σίγουρα θα αυξήσει την παραγωγικότητα σας.

Διαλέξτε File -> New ->M-file για να ενεργοποιήσετε τον ενσωματωμένο editor/debugger της Matlab, ή χρησιμοποιήστε τον editor της αρεσκείας σας. Ακολούθως δημιουργείστε το παρακάτω αρχείο και αποθηκεύστε το με το όνομα sketch.m:

```
[x y] = meshgrid(-3:.1:3, -3:.1:3);
z = x.^2 - y.^2;
mesh(x,y,z);
```

Στη συνέχεια, στο Matlab, βεβαιωθείτε ότι το directory όπου ανήκε το m-αρχείο βρίσκεται στο path. Ελέγξετε το πληκτρολογώντας `pathtool` και επαληθεύοντας ότι το directory σας υπάρχει. (Όσοι δεν έχετε γραφικά μπορείτε να χρησιμοποιήσετε την εντολή `addpath` *`directory`*.)

Τώρα δώστε την εντολή
`sketch`
Το αποτέλεσμα είναι αυτό που θα παίρνατε εάν είχατε εισαγάγει τις τρεις γραμμές του αρχείου, μετά το σύμβολο προτροπής.

Επίσης μπορείτε να εισάγετε δεδομένα με τον εξής τρόπο: αν ένα αρχείο με το όνομα mymatrix.m και βρίσκεται στο τρέχον directory όπου δουλεύετε περιέχει τις γραμμές

```
A = [2 3 4; 5 6 7; 8 9 0];
```

τότε η εντολή mymatrix διαβάζει αυτό το αρχείο και παράγει τον A. Παρόλα αυτά, για μεγάλους πίνακες, είναι πιο ασφαλές να χρησιμοποιείτε τις εντολές `save` και load του Matlab.

5.2. Συναρτήσεις

Οι συναρτήσεις είναι όπως και κάθε άλλο m-αρχείο, αλλά δέχονται ορίσματα, και μεταγλωττίζονται την πρώτη φορά που χρησιμοποιούνται σε ένα δοσμένο %session% (για

λόγους ταχύτητας).

Χρησιμοποιείστε τον αγαπημένο σας editor για να δημιουργήσετε ένα αρχείο με το όνομα sqroot.m, που να περιέχει τις παρακάτω γραμμές.

```
function sqroot(x)
% Compute square root by Newton's method

% Initial guess
xstart = 1;

for i = 1:100
        xnew = ( xstart + x/xstart)/2;
        disp(xnew);
        if abs(xnew – xstart)/xnew < eps, break, end;
        xstart = xnew;
end
```

Στο Matlab δώστε τις εντολές

```
format long
sqroot(19)
```

Θα πρέπει να δείτε την έξοδο της συνάρτησής σας.

Δύο προειδοποιήσεις:

- Μία συνάρτηση έχει πρόσβαση στις μεταβλητές που βρίσκονται στο "workspace" από το οποίο κλήθηκε, αλλά οι μεταβλητές που δημιουργούνται μέσα στη συνάρτηση (οι xstart και xnew, στο προηγούμενο παράδειγμα)είναι τοπικές, που σημαίνει ότι δεν είναι προσβάσιμες από το workspace από όπου καλείται η συνάρτηση. Για περισσότερες πληροφορίες, δείτε το κεφάλαιο "Προγραμματισμός Μ-Αρχείων" στο εγχειρίδιο, *Χρησιμοποιώντας το Matlab*.

- Παρατήρηση: αν διορθώνετε μία συνάρτηση κατά τη διάρκεια μίας προσπάθειας, χρησιμοποιείστε την `clear function_name` για να σβήσετε τη μεταγλωττισμένη έκδοση, ώστε να διαβαστεί η καινούργια.

5.3. Μαζικός Προγραμματισμός

Το MATLAB μπορεί να τρέξει σε "batch mode", με έναν παρόμοιο τρόπο. Αν ένα αρχείο με το όνομα "test.m" περιέχει τις εντολές που θέλετε να τρέξετε (που δεν υποστηρίζουν γραφικά), σε σύστημα UNIX πληκτρολογείστε:

```
% matlab < test.m > homework.out
```

Αυτό διαβάζεται ως "Τρέξε το MATLAB, με είσοδο από το test.in, και έξοδο στο test.out." Το αρχείο που δίνουμε ως είσοδο δε χρειάζεται να ονομάζεται κάτι-τελεία-m", αλλά **πρέπει** να τελειώνει με το `quit` .

5.4. Ιδέες προγραμματισμού

Τα "m-αρχεία" που υπάρχουν ήδη στοMATLAB παρέχουν πολλά παραδείγματα! Για να βρείτε τα σημεία όπου βρίσκονται χρησιμοποιείστε το `path`. Αυτό θα σας οδηγήσει και σε κάποια demos.

Μπορείτε ακόμα να δοκιμάσετε να πληκτρολογήσετε `demo` για να πάρετε μία ιδέα για το εύρος

των εργασιών που μπορούν να επιτευχθούν με το Matlab.

Δοκιμάστε το `help function` για μία άσκηση προγραμματιστών.

6. Τα επόμενα βήματα

Η επίσημη σελίδα της Matlab στο μαθηματικό τμήμα:
http://itia.math.uch.gr/~mav/matlab/start/index.html

στο διαδίκτυο θα σας φανεί σίγουρα ιδιαίτερα χρήσιμη.

Εάν χρησιμοποιείται UNIX, η σελίδα Using Math Software under UNIX :

http://www.indiana.edu/~statmath/math/all/mathunix/index.html

του Πανεπιστημίου της Ινδιάνας θα μπορεί να σας βοηθήσει. Είναι στην Αγγλική γλώσσα.

Τα επίσημα εγχειρίδια χρήσης που συνοδεύουν το λογισμικό της Matlab είναι γενικά πολύ καλά, και προσφέρουν ασυγκρίτως πολύ περισσότερες πληροφορίες από το παρόν βοήθημα. Το Μαθηματικό Τμήμα έχει στην κατοχή μία ευρεία σειρά τέτοιων εγχειριδίων τα οποία βρίσκονται στο Εργαστήριο των Εφαρμοσμένων και Υπολογιστικών Μαθηματικών στην Αίθουσα Γ104.