

ΕΙΣΑΓΩΓΗ ΣΤΟ ΜΑΤΛΑΒ[®] ΚΑΙ ΤΗΝ OCTAVE

ΒΑΣΙΣΜΕΝΗ ΣΤΟ ΕΓΧΕΙΡΙΔΙΟ
«*GETTING STARTED WITH MATLAB*»
THE MATHWORKS INC.

ΠΡΟΣΑΡΜΟΓΗ ΚΑΙ ΑΠΟΔΟΣΗ

ΑΝΔΡΕΑΣ ΤΖΑΝΗΣ
ΛΕΚΤΟΡΑΣ ΓΕΩΦΥΣΙΚΗΣ

ΑΘΗΝΑ 2007

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΕΧΟΜΕΝΑ.....	1
1 ΕΙΣΑΓΩΓΗ.....	3
ΤΙ ΕΙΝΑΙ ΤΟ MATLAB® ΚΑΙ Η OCTAVE;	3
ΓΕΝΙΚΗ ΟΡΓΑΝΩΣΗ.....	4
2 ΑΝΥΣΜΑΤΑ ΚΑΙ ΠΙΝΑΚΕΣ.....	5
ΤΟ ΜΑΓΙΚΟ ΤΕΤΡΑΓΩΝΟ	5
Εισαγωγή (εγγραφή) πινάκων	5
Άθροισμα (<i>sum</i>), ανάστροφος (<i>transpose</i>) και διαγώνιοι (<i>diag</i>).....	6
Δείκτες (<i>subscripts</i>).....	7
Ο τελεστής «άνω και κάτω τελεία» (<i>colon</i>).....	7
Η συνάρτηση <i>magic</i>	8
ΕΚΦΡΑΣΕΙΣ (EXPRESSIONS).....	9
Μεταβλητές.....	9
Αριθμοί.....	9
Τελεστές.....	9
Συναρτήσεις.....	10
Παραδείγματα εκφράσεων	10
ΕΡΓΑΖΟΜΕΝΟΙ ΜΕ ΠΙΝΑΚΕΣ.....	11
Κατασκευή πινάκων.....	11
Η συνάρτηση <i>load</i>	11
M-Files.....	12
Σύνδεση (<i>concatenation</i>)	12
Διαγραφή γραμμών και στηλών.....	12
Γραμμική άλγεβρα	13
Εργαζόμενοι με πίνακες (αριθμητικές διατάξεις - <i>arrays</i>).....	14
Δεδομένα πολλών μεταβλητών.....	15
Πράξεις με πίνακες και μονόμετρα μεγέθη.....	16
Λογικοί δείκτες	16
Η συνάρτηση <i>find</i> (εύρεση)	17
ΕΜΦΑΝΙΣΗ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΣΤΟ ΠΑΡΑΘΥΡΟ ΕΝΤΟΛΩΝ.....	18
Η οδηγία <i>format</i> (διαμόρφωση).....	18
Αναστολή παρουσίασης αποτελεσμάτων	18
Εισαγωγή μακρών εκφράσεων.....	18
Command Line Editing	18
3 ΓΡΑΦΙΚΑ	20
ΒΑΣΙΚΕΣ ΓΡΑΦΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ	20
Κατασκευή γραφήματος	20
Πολλαπλά δεδομένα σε ένα γράφημα.....	21
Γραμμές και Χρώματα	21
Φανταστικά και Μιγαδικά δεδομένα.....	22
Υπέρθεση παραστάσεων σε υπάρχον γράφημα	23
ΠΑΡΑΘΥΡΑ ΓΡΑΦΙΚΩΝ (ΕΙΚΟΝΩΝ)	23
Καθαρισμός παραθύρου για νέο γράφημα.....	23
Πολλαπλά γραφήματα σε μία εικόνα (παραθύρο).....	24
Έλεγχος των αξόνων.....	24
Υπομνήματα και τίτλοι	25
ΤΡΙΔΙΑΣΤΑΤΑ ΓΡΑΦΗΜΑΤΑ.....	26
Γραφική παράσταση συναρτήσεων δύο μεταβλητών	26
Εικόνες (<i>images</i>)	27
Εκτύπωση γραφικών.....	27

4 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ	28
ΈΛΕΓΧΟΣ ΡΟΗΣ	28
<i>if, else, και elseif</i>	28
<i>switch και case</i>	29
<i>for</i>	29
<i>while</i>	30
<i>continue</i>	30
<i>break</i>	31
<i>try - catch</i>	31
<i>return</i>	31
ΠΡΟΓΡΑΜΜΑΤΑ	32
<i>Scripts</i>	32
<i>Συναρτήσεις (functions)</i>	33
<i>Τύποι Συναρτήσεων</i>	33
<i>Κοινές (Global) Μεταβλητές</i>	34
<i>Εισαγωγή αλφαριθμητικών (string) ορισμάτων σε συναρτήσεις</i>	35
<i>Η συνάρτηση eval</i>	36
<i>Χειριστήρια (handles) συναρτήσεων</i>	36
<i>Ανυσματοποίηση (Vectorization)</i>	36
<i>Προκαθορισμός διαστάσεων (Preallocation)</i>	36
ΑΛΛΕΣ ΜΟΡΦΕΣ ΚΑΙ ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ	37
<i>Πολυδιάστατοι πίνακες</i>	37
<i>Κυψελοπίνακες (Cell Arrays)</i>	38
<i>Χαρακτήρες και κείμενο</i>	40
<i>Δομές (Structures)</i>	41
5. ΑΣΚΗΣΕΙΣ	43

1 Εισαγωγή

Τι είναι το MATLAB[®] και η OCTAVE;

Είναι ολοκληρωμένα συστήματα ανάλυσης δεδομένων και ανάπτυξης επιστημονικών και τεχνικών εφαρμογών. Προσφέρουν μία εύχρηστη γλώσσα προγραμματισμού, υπολογιστικές δυνατότητες υψηλών επιδόσεων και προχωρημένη γραφική απεικόνιση δεδομένων και αποτελεσμάτων, μέσα σε ένα φιλικό διαλογικό περιβάλλον όπου τα προβλήματα και οι λύσεις διατυπώνονται με οικείο μαθηματικό τρόπο. Μερικές τυπικές χρήσεις τους είναι:

- Μαθηματικά και υπολογισμοί.
- Ανάπτυξη αλγορίθμων
- Πρόσκτηση δεδομένων
- Εξομοίωση και Προσομοίωση
- Ανάλυση και απεικόνιση δεδομένων
- Επιστημονικά και τεχνικά γραφικά
- Ανάπτυξη εφαρμογών.

Αμφότερα τα συστήματα παρουσιάζονται ως διαλογικά περιβάλλοντα, χειριζόμενα (και διαχειριζόμενα) *ανύσματα* και *πίνακες* χωρίς να απαιτούν προτεραιά γνώση ή ρύθμιση των διαστάσεών τους. Η εξαιρετικά χρήσιμη αυτή δυνατότητα επιτρέπει την ταχεία αλγοριθμοποίηση και επίλυση σχεδόν των περισσότερων επιστημονικών και τεχνικών υπολογιστικών προβλημάτων, ιδιαίτερα δε όσων φυσικά διατυπώνονται συναρτήσεϊ διανυσμάτων και πινάκων.

Το όνομα MATLAB προέρχεται από σύντμηση των λέξεων *matrix laboratory* (*εργαστήριο πινάκων*) και έχει πλέον καταστεί βιομηχανικό πρότυπο στον τομέα των υπολογιστικών συστημάτων και μαθηματικών. Το MATLAB βασικά αναπτύχθηκε για να προσφέρει εύκολη πρόσβαση στο λογισμικό μαθηματικής και αριθμητικής ανάλυσης που αναπτύχθηκε στα πλαίσια των μεγάλων προγραμμάτων LINPACK (γραμμική άλγεβρα/ γραμμικά συστήματα) και EISPACK (ιδιοτιμές/ ιδιοδιανύσματα/ φασματικά αναπτύγματα πινάκων). Σήμερα, η υπολογιστική μηχανή του MATLAB βασίζεται στην μετεξέλιξη αυτού του λογισμικού, ήτοι στις βιβλιοθήκες LAPACK και BLAS οι οποίες περιλαμβάνουν την πλέον προηγμένη τεχνολογία χειρισμού και ανάλυσης πινάκων. Το MATLAB εξελίχθηκε δια της συνεχούς προσφοράς καινοτομιών και βελτιώσεων εκ μέρους πάρα πολλών χρηστών και προγραμματιστών. Σε πολλά ακαδημαϊκά περιβάλλοντα των ΗΠΑ και της Ευρώπης αποτελεί βασικό εργαλείο έρευνας και διδασκαλίας εισαγωγικών και προχωρημένων μαθημάτων επιστήμης και τεχνολογίας. Στην βιομηχανία αποτελεί ένα από τα αρτιότερα εργαλεία έρευνας, ανάλυσης και ανάπτυξης εφαρμογών.

Η OCTAVE αναπτύχθηκε με σκοπό να αποτελέσει το «MATLAB του φτωχού» και προσφέρεται δωρεάν στα πλαίσια του ακαδημαϊκού πνεύματος του οργανισμού GNU¹ που προωθεί το απεριόριστα ελεύθερο λογισμικό ανοικτού κώδικα. Αν και η υπολογιστική της μηχανή και επιδόσεις είναι η ίδια με του MATLAB (οι βιβλιοθήκες LAPACK και BLAS διανέμονται δωρεάν από την κυβέρνηση των ΗΠΑ), τα δύο συστήματα δεν συγκρίνονται όσον αφορά το επίπεδο του περιβάλλοντος εργασίας και των προσφερομένων ευκολιών, γραφικών δυνατοτήτων, προγραμματιστικών εργαλείων, προγραμματιστικών συντομεύσεων και τεχνικής υποστήριξης, με ασυζήτητη την υπεροχή του MATLAB. Παρ' όλα αυτά, με λίγη ευφυΐα, ιδίως εάν χρησιμοποιεί LINUX, ο «φτωχός» μπορεί να κάνει την δουλειά του και με το παραπάνω...

Αμφότερα τα συστήματα διαθέτουν ένα σύνολο πρόσθετων, εξειδικευμένων προγραμματιστικών λύσεων που ονομάζονται *εργαλειοθήκες* (*toolboxes*). Αυτές είναι εκτενείς (κατά περίπτωση) συλλογές συναρτήσεων (m-files) που επεκτείνουν τις δυνατότητες των συστημάτων με στόχο την επίλυση προ-

¹ Το ακρωνύμιο GNU προέρχεται από την ατέρμονα αναδρομική φράση GNU is Not Unix – είναι δηλαδή μαθηματικό λογοπαίγνιο κατά το πνεύμα του Douglas Hofstadter (βλ. Douglas Hofstadter, *Goedel, Escher, Bach* Penguin, 1979).

βλημάτων συγκεκριμένου τύπου. Για παράδειγμα, εργαλειοθήκες διατίθενται για προβλήματα επεξεργασίας σήματος, ελέγχου συστημάτων, νευρωνικών δικτύων, προσομοίωσης δεδομένων κ.λ.π.

Γενική Οργάνωση

Σε γενικές γραμμές τα συστήματα απαρτίζονται από πέντε κύριες ενότητες:

1. Περιβάλλον εργασίας και ανάπτυξης εφαρμογών: Πρόκειται για το σύνολο των εργαλείων που διευκολύνουν στην απρόσκοπτη χρήση των υπολογιστικών μηχανών. Στο MATLAB πολλά από αυτά τα εργαλεία είναι γραφικά (*Graphical User Interfaces- GUIs*) και περιλαμβάνουν το περιβάλλον εργασίας (*desktop*), το Παράθυρο Εντολών (*Command Window*), αρχείο εντολών, ένα συντάκτη (*editor*) και ελεγκτή (*debugger*) προγραμμάτων και «περιηγητές» (*browsers*) βοήθειας, του χώρου εργασίας και των αρχαιοκαταλόγου εργασίας. Στην OCTAVE οι δυνατότητες αυτές είναι μάλλον φτωχές. Υπάρχει μόνο το παράθυρο εντολών και περιορισμένες δυνατότητες βοήθειας.

2. Βιβλιοθήκες Μαθηματικών Συναρτήσεων και Προγραμμάτων: Πρόκειται για εκτεταμένη συλλογή υπολογιστικών αλγορίθμων, από στοιχειώδεις συναρτήσεις (π.χ. άθροισμα, ημίτονο, συνιμίτονο, μιγαδική αριθμητική κ.ά.), μέχρι εξαιρετικά περίπλοκες λειτουργίες, όπως γενικευμένη αντιστροφή πινάκων, ιδιοτιμές/ ιδιοδιανύσματα, ειδικές συναρτήσεις (π.χ. Bessel, Neumann) και μετασχηματισμούς Fourier.

3. Γλώσσα προγραμματισμού: Είναι σχεδόν πανομοιότυπη και για τα δύο συστήματα αλλά επειδή πρώτα αναπτύχθηκε ως μέρος του MATLAB θα την αποκαλούμε «γλώσσα MATLAB». Πρόκειται για γλώσσα υψηλού επιπέδου,² με πλήρη σειρά εντολών ελέγχου ροής, συναρτήσεις, δομές, εντολές εισόδου/ εξόδου και στοιχεία αντικειμενικού προγραμματισμού. Είναι εύχρηστη τόσο για «μικροπρογραμματισμό», δηλαδή ταχεία κατασκευή όχι και τόσο έξυπνα δομημένων προγραμμάτων ειδικού σκοπού, όσο και για «μακροπρογραμματισμό», δηλαδή συστηματική ανάπτυξη μεγάλων και πολύπλοκων εφαρμογών.

4. Γραφικά: Το MATLAB προσφέρει εξαιρετικές δυνατότητες γραφικής αναπαράστασης ανυσμάτων και πινάκων. Περιλαμβάνει GUI και συναρτήσεις για διδιάστατη και τριδιάστατη αναπαράσταση, επεξεργασία ειδώλων/ εικόνας, κατασκευή κινούμενης εικόνας (video) και παρουσίαση αποτελεσμάτων. Η OCTAVE βασίζεται κυρίως στο πακέτο GNUPLOT (το οποίο στην έκδοση για MS Windows έχει ενσωματώσει), και το οποίο αν και υπό όρους καλύπτει τις κοινότερες ανάγκες, υπολείπεται κατά πολύ της αρτιότητας και δυνατοτήτων του MATLAB.

5. Εξωτερικές διεπαφές/API: Πρόκειται για βιβλιοθήκη που επιτρέπει στον χρήστη την συγγραφή και χρήση προγραμμάτων C και FORTRAN που αλληλεπιδρούν με το MATLAB ή την OCTAVE. Περιλαμβάνει διευκολύνσεις για την κλήση συναρτήσεων και διαδικασιών των υπολογιστικών μηχανών από τα προγράμματα του χρήστη, παρέχοντας έτσι σχεδόν απεριόριστες δυνατότητες.

² Στην ορολογία της επιστήμης υπολογιστών, ως «υψηλού επιπέδου» νοείται μία γλώσσα κατανοητή από ανθρώπινα όντα. Οι «γλώσσες μηχανής», δηλαδή ο δυαδικός κώδικας οδηγούν τον οποίο κατανοούν οι επεξεργαστές είναι γλώσσα «χαμηλού επιπέδου». Ένα πρόγραμμα υψηλού επιπέδου μεταφράζεται σε γλώσσα χαμηλού επιπέδου με μεταγλωττιστές (compilers) ή διερμηνείς (interpreters) προκειμένου να δημιουργηθεί το εκτελέσιμο, (γνωστό από την Microsoft ως .EXE).

2 Ανύσματα και Πίνακες

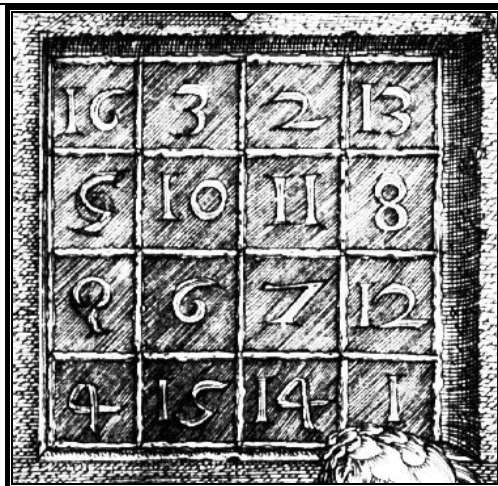
Το Μαγικό Τετράγωνο

Ένας Πίνακας (Μήτρα) αποτελεί ορθογώνια διάταξη αριθμών. Οι κοινοί μονόμετροι αριθμοί (*scalars*) μπορούν να λογισθούν ως πίνακες διαστάσεων 1×1 . Οι πίνακες μίας μόνο διάστασης ($1 \times N$ ή $N \times 1$) ονομάζονται ανύσματα. Τα ανύσματα In MATLAB, a matrix is a rectangular array of numbers. Για τους λόγους αυτούς (και για καλύτερη προσαρμογή) βοηθά να σκεπτόμαστε τα πάντα ως πίνακες.

Ένα πολύ καλό παράδειγμα πίνακα που θα χρησιμοποιηθεί στην παρούσα εισαγωγή είναι το «μαγικό τετράγωνο», ένα από τα οποία εμφανίζεται στο χαρακτηριστικό έργο Μελαγχολία I του Γερμανού καλλιτέχνη και ερασιτέχνη μαθηματικού Albrecht Dürer (βλ. Εικόνα 1). Η εικόνα αυτή είναι πλήρης μαθηματικών συμβολισμών και προσεκτική παρατήρηση θα αποκαλύψει ένα τετράγωνο πίνακα στην άνω δεξιά γωνία. Ο πίνακας αυτός είναι ένα «μαγικό τετράγωνο» και στην εποχή του Dürer, πολλοί πίστευαν ότι όντως είχε μαγικές ιδιότητες. Και τέτοιες μεν μπορεί να μην έχει, διαθέτει όμως ορισμένα συναρπαστικά χαρακτηριστικά άξια περαιτέρω διερεύνησης.



Εικόνα 1. Το χαρακτηριστικό Μελαγχολία I του Albrecht Duerer



Εικόνα 2. Μαγικό τετράγωνο. Λεπτομέρεια από την Μελαγχολία I του Duerer

Εισαγωγή (εγγραφή) πινάκων

Οι πράξεις με πίνακες είναι σχεδιασμένες να γράφονται και φαίνονται με τον φυσικότερο δυνατό τρόπο, όπως δηλαδή και στο χαρτί. Για αρχή, εκκινήσετε το υπολογιστικό σύστημα (MATLAB ή OCTAVE) και επαναλάβετε το παρεχόμενα παραδείγματα. Η εισαγωγή πινάκων στα συστήματα μπορεί να γίνει με αρκετούς διαφορετικούς τρόπους:

- ✓ Με απ' ευθείας εισαγωγή μίας σειράς αριθμών
- ✓ Με ανάγνωση (φόρτωμα) πινάκων από κατάλληλα εξωτερικά αρχεία δεδομένων.
- ✓ Με παραγωγή πινάκων χρησιμοποιώντας τις κατάλληλες ενσωματωμένες συναρτήσεις.
- ✓ Με κατασκευή πινάκων χρησιμοποιώντας ίδιες συναρτήσεις και προγράμματα.

Κατ' αρχάς εισάγουμε τα στοιχεία του πίνακα του Dürer υπό τη μορφή σειράς αριθμών, ακολουθώντας τις ακόλουθες συμβάσεις:

- Τα στοιχεία εκάστης σειράς (γραμμής) του πίνακα χωρίζονται με κενά ή κόμματα.
- Το ερωτηματικό ";" σημειώνει το τέλος κάθε γραμμής.
- Ολόκληρη την σειρά των στοιχείων κλείνεται σε αγκύλες, [].

Για να εισαχθεί ο πίνακας του Dürer, στο Παράθυρο Εντολών γράψτε απλά

```
A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]
```

Η σύστημα αποκρίνεται εμφανίζοντας τον πίνακα που μόλις εισήχθη:

```
A =
    16     3     2    13
     5    10    11     8
     9     6     7    12
     4    15    14     1
```

Ο πίνακας A είναι αυτός του χαρακτηριστικού. Από την στιγμή που εισήχθη, ο πίνακας διατηρείται ως σύμβολο και περιεχόμενο στον χώρο εργασίας, μέχρις ότου σβηστεί από τον χρήστη. Αναφερόμαστε σ' αυτόν με το σύμβολο – όνομά του, A. Ας διερευνήσουμε λοιπόν τις ιδιότητές του – γιατί λέγεται μαγικός;

Άθροισμα (sum), ανάστροφος (transpose) και διαγώνιοι (diag)

Όσοι έχουν ασχοληθεί, γνωρίζουν ότι οι ειδικές ιδιότητες του μαγικού τετραγώνου έχουν να κάνουν με διαφορετικούς τρόπους άθροισης των στοιχείων του. Άθροιση κατά μήκος οιασδήποτε γραμμής ή στήλης ή των κυρίων διαγώνιων δίδει πάντοτε το αυτό αποτέλεσμα, πράγμα που μπορούμε να πιστοποιήσουμε εύκολα. Ας προσπαθήσουμε την πράξη:

```
sum(A)
```

Η απόκριση είναι

```
ans =
    34    34    34    34
```

Όταν δεν προσδιορίζεται ακριβώς το όνομα μίας μεταβλητή εξόδου, το σύστημα χρησιμοποιεί την εσωτερική μεταβλητή ans (συντόμευση της αγγλικής λέξης *answer* - απάντηση), για να αποθηκεύσει τα αποτελέσματα. Στην συγκεκριμένη περίπτωση υπολογίσατε ένα οριζόντιο άνυσμα που περιέχει τα άθροισμα των στηλών του A. Κάθε στήλη έχει το ίδιο «μαγικό» άθροισμα 34.

Τι γίνεται όμως με τις γραμμές; Η συνάρτηση sum είναι προγραμματισμένη να χειρίζεται τις στήλες ενός πίνακα. Για τον λόγο αυτό, εάν κανείς θελήσει να υπολογίσει τα άθροισμα των γραμμών, πρέπει να αναστρέψει τον πίνακα και μετά να αναστρέψει το αποτέλεσμα. Η πράξη της αναστροφής δηλώνεται συμβολίζεται με μία απλή απόστροφο και στρέφει ένα πίνακα γύρω από την κύρια διαγώνιό του, μετατρέποντας μία στήλη (κατακόρυφο άνυσμα) σε μία γραμμή (οριζόντιο άνυσμα). Έτσι,

```
A'
```

παράγει

```
ans =
    16     5     9     4
     3    10     6    15
     2    11     7    14
    13     8    12     1
```

και

```
sum(A')'
```

παράγει ένα κατακόρυφο άνυσμα που περιέχει το άθροισμα των γραμμών του πίνακα:

```
ans =
    34
    34
    34
    34
```

Το άθροισμα των στοιχείων της κυρίας διαγωνίου λαμβάνεται με επάλληλη εφαρμογή των συναρτήσεων *sum* και *diag*:

```
diag(A)
```

παράγει

```
ans =  
    16  
    10  
     7  
     1
```

και

```
sum(diag(A))
```

δίδει

```
ans =  
    34
```

Η *αντι-διαγώνιος* του πίνακα δεν έχει μεγάλη μαθηματική σημασία και δεν υπάρχει ειδική συνάρτηση που να την υπολογίζει. Υπάρχει όμως μία συνάρτηση που ανακτά ένα πίνακα κατοπτρικά ως προς την πρώτη στήλη του, η *fliplr*. Τότε, η πράξη

```
sum(diag(fliplr(A)))
```

δίδει το άθροισμα της αντι-διαγωνίου του πίνακα.

```
ans =  
    34
```

Πιστοποιήσαμε λοιπόν ότι ο πίνακας στο χαρακτηριστικό του Dürer όντως είναι ένα μαγικό τετράγωνο, και στην πορεία είδαμε στοιχεία από τον τρόπο χρήσης των συστημάτων μας. Θα εξακολουθήσουμε να χρησιμοποιούμε τον πίνακα αυτό για περαιτέρω εξερεύνηση και εξοικείωση με τις υπολογιστικές μας μηχανές.

Δείκτες (subscripts)

Το στοιχείο της γραμμής *i* και στήλης *j* του *A* δηλώνεται ως *A(i,j)*. Για παράδειγμα, *A(4,2)* στο μαγικό μας τετράγωνο είναι ο αριθμός που αντιστοιχεί στην τέταρτη γραμμή και δεύτερη στήλη και ισούται προς 15. Έτσι, για να υπολογίσουμε το άθροισμα των στοιχείων της τέταρτης στήλης του *A*, γράφουμε:

```
A(1,4) + A(2,4) + A(3,4) + A(4,4) ↵
```

Αυτό παράγει

```
ans =  
    34
```

Είναι επίσης δυνατό να αναφερθούμε στα στοιχεία ενός πίνακα με ένα μόνο δείκτη, *A(k)*. Αυτός είναι ο συνήθης τρόπος ανάκλησης στοιχείων από οριζόντια ή κατακόρυφα ανύσματα, αλλά μπορεί να εφαρμοσθεί και σε διδιάστατους πίνακες, στην οποία περίπτωση εξισώνονται με ένα μακρύ κατακόρυφο διάνυσμα σχηματιζόμενο από τις στήλες τους. Έτσι, στο μαγικό μας τετράγωνο *A(8)* είναι ένας εναλλακτικός τρόπος ανάκλησης της τιμής 15 που έχει καταχωρηθεί στο *A(4,2)*. Αν επιχειρηθεί η ανάκληση της τιμής στοιχείου εκτός των διαστάσεων του πίνακα, προκύπτει σφάλμα:

```
t = A(4, 5)  
Index exceeds matrix dimensions.
```

Από την άλλη μεριά, εάν εγγραφεί η τιμή ενός στοιχείου εκτός των διαστάσεων του πίνακα, το μέγεθός του αυξάνεται αυτόματα για να καταχωρήσει το νέο δεδομένο. Για παράδειγμα,

```
X = A; X(4,5) = 17
```

```
X =  
    16     3     2    13     0  
     5    10    11     8     0  
     9     6     7    12     0  
     4    15    14     1    17
```

Ο τελεστής «άνω και κάτω τελεία» (colon)

Πρόκειται για έναν από τους σπουδαιότερους τελεστές και χρησιμοποιείται με αρκετούς διαφορετικούς τρόπους. Η έκφραση

```
1:10
```


είναι ένα οριζόντιο άνυσμα (σειρά) που περιέχει τους ακέραιους αριθμούς από 1 έως 10:

```
ans =  
    1     2     3     4     5     6     7     8     9    10
```

Τα μη μοναδιαία βήματα πρέπει να δηλώνονται ακριβώς. Για παράδειγμα,

```
100:-7:50
```

```
ans =  
    100     93     86     79     72     65     58     51
```

και

```
0:pi/4:pi
```

```
ans =  
    0     0.7854     1.5708     2.3562     3.1416
```

Εκφράσεις με δείκτες που περιλαμβάνουν τον τελεστή colon αναφέρονται σε *μερίσματα* ενός πίνακα:

```
A(1:k,j)
```

Είναι τα πρώτα k στοιχεία της j-στης στήλης του A. Έτσι, η

```
sum(A(1:4,4))
```

υπολογίζει το άθροισμα των στοιχείων της τέταρτης στήλης. Αλλά υπάρχει ένας καλύτερος τρόπος. Η «άνω και κάτω» τελεία αφ' εαυτής ανακαλεί αναφέρεται *όλα* τα στοιχεία μίας σειράς ή στήλης ενός πίνακα, και η λέξη-κλειδί *end* ανακαλεί την *τελευταία* σειρά ή στήλη. Έτσι, η

```
sum(A(:,end))
```

υπολογίζει το άθροισμα των στοιχείων της τελευταίας στήλης του A:

```
ans =  
    34
```

Γιατί είναι το μαγικό άθροισμα ενός τετραγώνου 4x4 ίσο με 34; Εάν οι ακέραιοι αριθμοί από 1 έως 16 ταξινομηθούν σε τέσσερις ομάδες με τα ίσα αθροίσματα, εκείνο το άθροισμα πρέπει να είναι

```
sum(1:16)/4
```

που φυσικά, είναι

```
ans =  
    34
```

Η συνάρτηση magic.

Στην πραγματικότητα υπάρχει μία ενσωματωμένη συνάρτηση που παράγει μαγικά τετράγωνα σχεδόν κάθε μεγέθους. Όπως ήταν αναμενόμενο, αυτή η συνάρτηση ονομάζεται *magic*:

```
B = magic(4)
```

```
B =  
    16     2     3    13  
     5    11    10     8  
     9     7     6    12  
     4    14    15     1
```

Αυτός ο πίνακας είναι σχεδόν ο ίδιος με αυτόν στο χαρακτηριστικό του Dürer και έχει τις ίδιες "μαγικές" ιδιότητες. Η μόνο διαφορά είναι ότι οι δύο μεσαίες στήλες έχουν εναλλαχθεί. Για να μετατρέψουμε τον B στον A του Dürer, εναλλάξετε τις δύο μεσαίες στήλες:

```
A = B(:, [1 3 2 4])
```

Με την οδηγία αυτή, για κάθε γραμμή του πίνακα B αναδιατάσσονται τα στοιχεία κατά την σειρά 1, 2, 3, 4, ώστε

```
A =  
    16     3     2    13  
     5    10    11     8  
     9     6     7    12  
     4    15    14     1
```

Χρησιμοποιώντας αυτή την μορφή ο Dürer θέλησε να περιλάβει την ημερομηνία του χαρακτηριστικού του στο κατώτατο άκρο του μαγικού τετραγώνου του: 1514.

Εκφράσεις (expressions)

Όπως και με τις περισσότερες άλλες γλώσσες προγραμματισμού, τα υπολογιστικά μας συστήματα παρέχουν μαθηματικές εκφράσεις, αλλά αντίθετα από τις περισσότερες γλώσσες προγραμματισμού, αυτές οι εκφράσεις περιλαμβάνουν ολόκληρους πίνακες. Οι δομικές μονάδες των εκφράσεων είναι οι «μεταβλητές», οι «αριθμοί», οι «τελεστές» και οι «συναρτήσεις».

Μεταβλητές

Τα MATLAB και OCTAVE δεν απαιτούν οποιοσδήποτε δηλώσεις τύπων ή διάστασης. Όταν συναντούν ένα νέο όνομα μεταβλητής, δημιουργούν αυτόματα τη μεταβλητή και διαθέτουν τον κατάλληλο χώρο στην μνήμη. Εάν η μεταβλητή υπάρχει ήδη, αλλάζουν το περιεχόμενό της και, εάν είναι απαραίτητο, αναμορφώνουν το απαραίτητο χώρο στην μνήμη. Για παράδειγμα, η

```
num_students = 25
```

δημιουργεί ένα πίνακα 1×1 με το όνομα *num_students* και αποθηκεύει την τιμή 25, του μοναδικού στοιχείου του. Για να δούμε τον πίνακα που αντιστοιχεί σε οιαδήποτε μεταβλητή, απλά γράφουμε το όνομα της μεταβλητής. Τα ονόματα των μεταβλητών αποτελούνται από ένα λατινικό χαρακτήρα, ακολουθούμενα από οιονδήποτε αριθμό χαρακτήρων, αριθμητικών ψηφίων ή *underscore*. Τα κεφαλαία και μικρά γράμματα λογίζονται ξεχωριστά: η A και η a δεν είναι η ίδια μεταβλητή.

Αν και τα ονόματα των μεταβλητών μπορούν να έχουν οιοδήποτε μήκος, εσωτερικά χρησιμοποιούνται μόνο οι πρώτοι N χαρακτήρες του ονόματος, όπου N προσδιορίζεται μέσω της συνάρτησης *namelengthmax*, και οι υπόλοιποι αγνοούνται. Για τον λόγο αυτό είναι σημαντικό να καθίσταται το όνομα κάθε μεταβλητής μοναδικό κατά τους πρώτους N χαρακτήρες, ώστε να μην προκαλείται σύγχυση.

```
N = namelengthmax
```

```
N =
```

63

Η συνάρτηση *genvarname* είναι πολύ χρήσιμη στο να δημιουργεί ονόματα μεταβλητών που είναι αποδεκτά αλλά και μοναδικά.

Αριθμοί

Για την δήλωση αριθμών, χρησιμοποιείται ο κατά το αγγλοσαξωνικό σύστημα συμβατικός συμβολισμός δεκαδικών αριθμών, με προαιρετικά το δεκαδικό σημείο και το πρόσημο συν ή πλην. Η επιστημονική παράσταση (*scientific notation*) χρησιμοποιεί τον χαρακτήρα e για να δηλώσει ύψωση σε δύναμη του δέκα. Οι φανταστικοί αριθμοί δηλώνονται με επίθεμα i ή j. Μερικά παραδείγματα ορθής χρήσης αριθμών είναι:

```
3          -99          0.0001
9.6397238  1.60210e-20   6.02252e23
1i         -3.14159j    3e5i
```

Όλοι οι αριθμοί καταχωρούνται στην μνήμη με διπλή ακρίβεια, σύμφωνα με τον τύπο (format) *long*, του προτύπου αριθμών κινητής υποδιαστολής IEEE³, ήτοι με πεπερασμένη ακρίβεια 16 σημαντικών δεκαδικών σημείων και πεπερασμένο εύρος διακύμανσης από 10^{-308} μέχρι 10^{+308} .

Τελεστές

Οι εκφράσεις χρησιμοποιούν τους οικείους αριθμητικούς τελεστές και κανόνες προτεραιότητας κατά την εκτέλεση πράξεων.

```
+      Πρόσθεση
-      Αφαίρεση
*      Πολλαπλασιασμός
```

³ IEEE – Institute of Electrical and Electronic Engineering

- / Διαίρεση
- \ Αριστερή διαίρεση πινάκων.
- ^ Ύψωση σε δύναμη
- ' Μιγαδικός συζυγής. Προκειμένου περί πινάκων, ανάστροφος μιγαδικός συζυγής
- () Ιεραρχία εκτέλεσης – προηγούνται οι πράξεις εντός των εσωτερικότερων παρενθέσεων.

Συναρτήσεις

Παρέχεται μεγάλος αριθμός τυπικών στοιχειωδών μαθηματικών συναρτήσεων, οι οποίες περιλαμβάνουν τις *abs* (απόλυτη τιμή), *sqrt* (τετραγωνική ρίζα), *exp* (νεπέριος εκθέτης), και *sin* (ημίτονο). Η τετραγωνική ρίζα ή ο λογάριθμος ενός αρνητικού αριθμού δεν παράγει σφάλμα, αλλά το κατάλληλο μιγαδικό αποτέλεσμα. Επίσης παρέχονται και πολλές προχωρημένες μαθηματικές συναρτήσεις, όπως οι Bessel και Γάμμα, οι περισσότερες των οποίων δέχονται και μιγαδικά ορίσματα. Για ένα κατάλογο των στοιχειωδών συναρτήσεων, δώσατε την εντολή

```
help elfun
```

Για ένα κατάλογο των προχωρημένων μαθηματικών συναρτήσεων δώσατε

```
help specfun help elmat
```

Μερικές από τις συναρτήσεις, όπως οι *sqrt* και *sin*, είναι ενσωματωμένες. Οι ενσωματωμένες συναρτήσεις είναι μέρος του πυρήνα των συστημάτων και εξαιρετικά αποδοτικές, αλλά οι αλγοριθμικές τους λεπτομέρειες και ο κώδικας δεν είναι προσβάσιμες. Άλλες συναρτήσεις, όπως οι *gamma* (Γάμμα) και *sinh* (υπερβολικό ημίτονο) παρέχονται ως συνοδά προγράμματα (M-files). Στην περίπτωση αυτή ο κώδικας είναι προσβάσιμος και μπορεί να τροποποιηθεί/ βελτιωθεί από τον χρήστη. Στην κατηγορία αυτή συναρτήσεων υπάρχουν ορισμένες διαφορές μεταξύ MATLAB και OCTAVE, αλλά όχι στο επίπεδο που θα μας απασχολήσει.

Αρκετές ειδικές συναρτήσεις παρέχουν τις τιμές χρήσιμων σταθερών:

<i>pi</i>	3.14159265,
<i>1-i</i>	Φανταστική μονάδα,
<i>j</i>	όπως η <i>i</i>
<i>eps</i>	Σχετική ακρίβεια κινητής υποδιαστολής, $\epsilon = 2^{-52}$
<i>realmin</i>	Ελάχιστος αριθμός κινητής υποδιαστολής, 2^{-1022}
<i>realmax</i>	Μέγιστος αριθμός κινητής υποδιαστολής, $2 - \epsilon \approx 2^{1023}$
<i>Inf</i>	Άπειρο
<i>NaN</i>	Μη-αριθμός

Το άπειρο παράγεται με διαίρεση πεπερασμένου αριθμού δια του μηδενός, ή υπολογίζοντας μαθηματικές εκφράσεις που *υπερχειλίζουν* (*overflow*) την δυνατότητα των καταχωρητών του συστήματος, δηλαδή ξεπερνούν την *realmax*. Οι μη-αριθμοί παράγονται από εκφράσεις που καταλήγουν πράξεις τύπου 0/0 ή *Inf - Inf*, οι οποίες έχουν απροσδιόριστη τιμή. Οι συναρτήσεις αυτές δεν είναι αποκλειστικές, δηλαδή υπάρχει δυνατότητα να τους δοθεί διαφορετική τιμή, για παράδειγμα

```
eps = 1.e-6
```

η οποία θα χρησιμοποιηθεί στους επόμενους υπολογισμούς. Η αρχική τιμή μπορεί να αποκατασταθεί με την εντολή

```
clear eps
```

Παραδείγματα εκφράσεων

Έχουμε δει ήδη διάφορα παραδείγματα εκφράσεων. Εδώ δίδονται μερικά ακόμη, και τα προκύπτοντα αποτελέσματα:

```
rho = (1+sqrt(5))/2
```

```
rho =
```

```
1.6180
```

```
a = abs(3+4i)
```

```

a =
    5
z = sqrt(besselk(4/3, rho-i))
z =
    0.3730+ 0.3214i
huge = exp(log(realmax))
huge =
    1.7977e+308
toobig = pi*huge
toobig =
    Inf

```

Εργαζόμενοι με πίνακες

Κατασκευή πινάκων

Διατίθενται τέσσερες συναρτήσεις κατασκευής βασικών πινάκων:

Zeros : Πίνακας με όλα τα στοιχεία του 0.
Ones : Πίνακας με όλα τα στοιχεία του 1.
Rand : Πίνακας με τα στοιχεία του ομοιόμορφα κατανεμημένους τυχαίους αριθμούς.
Randn : Πίνακας με τα στοιχεία του κανονικά κατανεμημένους τυχαίους αριθμούς.

Ιδού ορισμένα παραδείγματα:

```

Z = zeros(2,4)
Z =
    0    0    0    0
    0    0    0    0
F = 5*ones(3,3)
F =
    5    5    5
    5    5    5
    5    5    5
N = fix(10*rand(1,10))
N =
    9    2    6    4    8    7    4    0    8    4
R = randn(4,4)
R =
    0.6353    0.0860   -0.3210   -1.2316
   -0.6014   -2.0046    1.2366    1.0556
    0.5512   -0.4931   -0.6313   -0.1132
   -1.0998    0.4620   -2.3252    0.3792

```

Η συνάρτηση load

Η συνάρτηση load διαβάζει δυαδικά αρχεία περιέχοντα πίνακες ή ανύσματα που έχουν δημιουργηθεί σε προηγούμενο χρόνο, ή ASCII αρχεία που περιέχουν αριθμητικά δεδομένα. Τα τελευταία πρέπει να είναι οργανωμένα ως ορθογώνιοι πίνακες αριθμών διαχωρισμένων με κενά, με μία γραμμή ανά σειρά και ίσο αριθμό στοιχείων ανά γραμμή. Για παράδειγμα, ας κατασκευάσουμε ένα συντάκτη (editor) ένα αρχείο ASCII που περιέχει τις εξής τέσσερες γραμμές,

```

16.0    3.0    2.0    13.0
 5.0    10.0   11.0    8.0
 9.0    6.0    7.0    12.0
 4.0    15.0   14.0    1.0

```

και ας αποθηκεύσουμε το αρχείο υπό το όνομα *magik.dat*. Τότε η εντολή

```
load magik.dat
```

διαβάζει τον φάκελο και δημιουργεί την μεταβλητή *magik*, η οποία περιέχει τον ανωτέρω πίνακα. Ειδικά για το MATLAB, ένας καλός τρόπος ανάγνωσης (εισαγωγής) αρχείων (δυαδικών ή ASCII), είναι να χρησιμοποιήσουμε τον Import Wizard.

M-Files

Η εισαγωγή πινάκων είναι επίσης δυνατή μέσω των *M-files*, οι οποίοι είναι αρχεία ASCII (κειμένου) περιέχοντα κώδικα σε γλώσσα MATLAB. Τα αρχεία αυτά δημιουργούνται με ένα συντάκτη κειμένου και περιέχουν τις ίδιες εντολές, οι οποίες θα εδίδοντο απ' ευθείας στο Παράθυρο Εντολών. Τα αρχεία αυτά σώζονται με (οιοδήποτε) όνομα, αλλά πρέπει να καταλήγουν με *.m*. Για παράδειγμα, ας δημιουργήσουμε ένα αρχείο που περιέχει τις εξής πέντε γραμμές:

```
A = [...  
16.0    3.0    2.0    13.0  
 5.0    10.0   11.0    8.0  
 9.0    6.0    7.0    12.0  
 4.0    15.0   14.0    1.0 ];
```

Αποθηκεύουμε το αρχείο με το όνομα *magik.m*. Κατόπιν η εντολή

```
magik
```

διαβάζει το αρχείο και δημιουργεί μια μεταβλητή *A*, η οποία περιέχει το μαγικό τετράγωνο του Dürer.

Σύνδεση (concatenation)

Σύνδεση είναι η διαδικασία συνένωσης μικρότερων πινάκων προς κατασκευή μεγαλύτερων. Σημειώτέο ότι ο πρώτος πίνακας που κατασκευάστηκε στην αρχή του παρόντος, έγινε με σύνδεση ενός εκάστου των στοιχείων του. Τελεστής σύνδεσης είναι το ζεύγος ορθογωνίων αγκυλών, `[]`. Για παράδειγμα, ας πάρουμε το γνωστό μας 4x4 μαγικό τετράγωνο και σχηματίσουμε:

```
B = [A A+32; A+48 A+16]
```

Το αποτέλεσμα είναι ένας 8x8 πίνακας προκύπτων από την σύνδεση τεσσάρων μερισμάτων:

```
B =  
    16     3     2    13    48    35    34    45  
     5    10    11     8    37    42    43    40  
     9     6     7    12    41    38    39    44  
     4    15    14     1    36    47    46    33  
    64    51    50    61    32    19    18    29  
    53    58    59    56    21    26    27    24  
    57    54    55    60    25    22    23    28  
    52    63    62    49    20    31    30    17
```

Ο πίνακας *B* είναι σχεδόν μαγικό τετράγωνο, με τα στοιχεία του να αποτελούν αναδιάταξη των ακεραίων 1:64. Τα αθροίσματα των στηλών του δίδουν τις σωστές τιμές για το 8x8 μαγικό τετράγωνο:

```
sum(B)
```

```
ans =
```

```
    260    260    260    260    260    260    260    260
```

Τα αθροίσματα των σειρών του όμως, `sum(B)'`, δεν είναι όλα τα ίδια. Ο *B* χρειάζεται περαιτέρω αναδιάταξη προκειμένου να γίνει κανονικό μαγικό τετράγωνο.

Διαγραφή γραμμών και στηλών

Οι γραμμές και στήλες ενός πίνακα μπορούν να διαγραφούν με χρήση του ζεύγους ορθογωνίων αγκυλών, δηλαδή με εξίσωσή τους προς ένα κενό άνωσμο ή πίνακα. Για παράδειγμα, δίδομε

```
X = A;
```

Για να διαγραφεί η δεύτερη στήλη του *X*, εντελλόμαστε

```
X(:,2) = []
```

και ο *X* αλλάζει στον

```
X =  
    16     2    13  
     5    11     8
```

```

9      7      12
4      14     1

```

Εάν επιχειρηθεί αφαίρεση ενός μόνο στοιχείου με αυτό τον τρόπο, το αποτέλεσμα δεν είναι αποδεκτός πίνακας. Έτσι, εκφράσεις του τύπου

```
X(1,2) = []
```

επιστρέφουν σφάλμα! Εντούτοις, η χρησιμοποίηση ενός μόνο δείκτη διαγράφει ένα μόνο στοιχείο ή σειρά στοιχείων και αναδιαμορφώνει τα υπόλοιπα στοιχεία σε ένα οριζόντιο άνυσμα:

```
X(2:2:10) = []
```

```
X =
    16     9     2     7    13    12     1
```

Γραμμική άλγεβρα

Στο σημείο αυτό πρέπει να σημειώσουμε ότι ο όρος ‘πίνακας’ δηλώνει οιαδήποτε ορθογώνια διάταξη αριθμών. Όταν όμως αυτή η ορθογώνια διάταξη αναπαριστά ένα γραμμικό μετασχηματισμό συντεταγμένων, ο ορθός ορισμός είναι ‘μήτρα’. Άτυπα, και σε καθημερινή εφαρμογή, οι όροι ‘μήτρα’ και ‘πίνακας’ χρησιμοποιούνται ως συνώνυμα, και στην Ελληνική ορολογία τείνει να επικρατήσει ο δεύτερος, αλλά δεν πρέπει ποτέ να λησμονούμε την διαφορά. Γραμμική άλγεβρα είναι ο κλάδος των μαθηματικών που ασχολείται με τις ιδιότητες και χειρισμό μητρών. Το μαγικό τετράγωνο, A, του Dürer’, θα χρησιμεύσει και στην επίδειξη αρκετών από τις δυνατότητες των υπολογιστικών μας συστημάτων στον τομέα της γραμμικής άλγεβρας, τον οποίο άλλωστε έχουν βασικά κατασκευασθεί για να εξυπηρετήσουν. Έχουμε ήδη δει τον ανάστροφο του μαγικού τετραγώνου, A'. Προσθέτοντας μία μήτρα πίνακα στην ανάστροφή της, λαμβάνουμε μία *συμμετρική* μήτρα:

```
A + A'
```

```
ans =
    32     8    11    17
     8    20    17    23
    11    17    14    26
    17    23    26     2
```

Το σύμβολο πολλαπλασιασμού, *, δηλώνει και πολλαπλασιασμό μητρών, ο οποίος γίνεται σχηματίζοντας τα εσωτερικά γινόμενα γραμμών και στηλών. Ο πολλαπλασιασμός μίας μήτρας με την ανάστροφή της επίσης παράγει μία συμμετρική μήτρα:

```
A' * A
```

```
ans =
    378    212    206    360
    212    370    368    206
    206    368    370    212
    360    206    212    378
```

Η ορίζουσα της ανωτέρω συμβαίνει να είναι ίση προς μηδέν, πράγμα που σημαίνει ότι αυτή είναι *ιδιάζουσα*:

```
d = det(A)
```

```
d =
    0
```

Εφόσον η μήτρα είναι ιδιάζουσα, δεν έχει αντίστροφη. Εάν προσπαθήσουμε να την αντιστρέψουμε,

```
X = inv(A)
```

λαμβάνουμε μία προειδοποίηση:

```
Warning: Matrix is close to singular or badly scaled.
Results may be inaccurate. RCOND = 9.796086e-018.
```

Ο αλγόριθμος αντιστροφής δεν μπορεί να προσδιορίσει ακριβώς την *ιδιάζουσα τιμή*, δηλαδή την αιτία της αστοχίας, λόγω *σφαλμάτων στρογγύλευσης*. Βλέπομε όμως ότι η τιμή της *rcond*, η οποία υπολογίζει τον *δείκτη κατάστασης (condition number)* της μήτρας, είναι της τάξης της *eps*, δηλ. της σχετικής ακρίβειας κινητής υποδιαστολής, οπότε και η επιστρεφόμενη αντίστροφη μήτρα θα είναι τέτοια, ώστε να μην έχει καμία χρησιμότητα.

Οι *ιδιοτιμές* του μαγικού τετραγώνου παρουσιάζουν ενδιαφέρον:

```
e = eig(A)
e =
    34.0000
     8.0000
     0.0000
    -8.0000
```

Μία από τις ιδιοτιμές είναι μηδενική, συνέπεια και αυτή του ιδιάζοντος της μήτρας μας. Η μεγαλύτερη ιδιοτιμή είναι 34, ίση προς το μαγικό άθροισμα. Αυτό συμβαίνει διότι το κατακόρυφο άνυσμα με όλα τα στοιχεία μονάδα, είναι ένα ιδιοδιανύσμα:

```
v = ones(4,1)
v =
     1
     1
     1
     1
```

```
A*v
ans =
    34
    34
    34
    34
```

Όταν ένα μαγικό τετράγωνο διαιρεθεί (κανονικοποιηθεί) με το μαγικό του άθροισμα,

```
P = A/34
```

Το αποτέλεσμα είναι μία διπλά στοχαστική μήτρα, της οποίας τα αθροίσματα των γραμμών και στηλών είναι όλα 1:

```
P =
    0.4706    0.0882    0.0588    0.3824
    0.1471    0.2941    0.3235    0.2353
    0.2647    0.1765    0.2059    0.3529
    0.1176    0.4412    0.4118    0.0294
```

Μήτρες τέτοιου τύπου παριστούν μεταβατικές πιθανότητες *Μαρκοβιανών διεργασιών*. Διαδοχικές δυνάμεις της μήτρας παριστούν διαδοχικά βήματα της διεργασίας. Για το παράδειγμά μας, η 5^η δύναμη είναι

```
P^5
ans =
    0.2507    0.2495    0.2494    0.2504
    0.2497    0.2501    0.2502    0.2500
    0.2500    0.2498    0.2499    0.2503
    0.2496    0.2506    0.2505    0.2493
```

με λίγα ακόμη πειράματα θα διαπιστωθεί ότι καθώς ο εκθέτης k τείνει στο άπειρο, όλα τα στοιχεία της $k^{\text{ης}}$ δύναμης, P^k , τείνουν στο $\frac{1}{4}$. Τέλος, οι συντελεστές του *χαρακτηριστικού πολυώνυμου*

```
poly(A)
```

είναι

```
ans =
     1    -34    -64    2176     0
```

πράγμα που δείχνει ότι το χαρακτηριστικό πολυώνυμο

```
det(A - λI)
```

είναι

$\lambda^4 - 34\lambda^3 - 64\lambda^2 + 2176\lambda$.

Ο σταθερός όρος είναι μηδέν διότι η μήτρα είναι ιδιάζουσα και ο συντελεστής του κυβικού όρου είναι 34 διότι ο πίνακας είναι μαγικός!

Εργαζόμενοι με πίνακες (αριθμητικές διατάξεις - arrays)

Έξω από τον κόσμο της γραμμικής άλγεβρας, οι πίνακες δεν είναι παρά διδιάστατες αριθμητικές διατάξεις. Στην περίπτωση αυτή, οι αριθμητικές πράξεις γίνονται στοιχείο προς στοιχείο. Αυτό σημαίνει ότι η πρόσθεση και η αφαίρεση γίνονται με τον αυτό τρόπο, αλλά ο πολλαπλασιασμός και η διαίρεση είναι εντελώς διαφορετικοί. Τα υπολογιστικά μας συστήματα χρησιμοποιούν μία τελεία ως μέρος του τελε-

στή πολλαπλασιαστικών πράξεων μεταξύ πινάκων (διατάξεων). Οι τελεστές πράξεων είναι:

+	Πρόσθεση,
-	Αφαίρεση,
.*	Πολλαπλασιασμός στοιχείου προς στοιχείο,
./	Διαίρεση στοιχείου προς στοιχείο,
.\	Αριστερή διαίρεση στοιχείου προς στοιχείο,
.^	Ύψωση στοιχείων σε δύναμη,
.'	Μη συζυγής ανάστροφος.

Αν το μαγικό τετράγωνο του Dürer πολλαπλασιασθεί με τον εαυτό του ως πίνακας,

```
A.*A
```

Το αποτέλεσμα είναι ένας πίνακας περιλαμβάνων τα τετράγωνα των ακεραίων από 1 ως 16, διατεταγμένα σε ασυνήθιστη μορφή:

```
ans =  
    256     9     4    169  
    25    100    121    64  
    81     36     49    144  
    16    225    196     1
```

Οι πράξεις επί πινάκων είναι χρήσιμες για την κατασκευή καταλόγων. Υποθετίσθω ότι n είναι το κατάρυφο άνυσμα

```
n = (0:9)';
```

Τότε, η

```
pows = [n n.^2 2.^n]
```

κατασκευάζει ένα πίνακα των τετραγώνων και δυνάμεων του 2 για τους ακεραίους 0:9:

```
pows =  
     0     0     1  
     1     1     2  
     2     4     4  
     3     9     8  
     4    16    16  
     5    25    32  
     6    36    64  
     7    49   128  
     8    64   256  
     9    81   512
```

Οι στοιχειώδεις μαθηματικές συναρτήσεις λειτουργούν επί των πινάκων στοιχείο προς στοιχείο. Έτσι, οι

```
format short g
```

```
x = (1:0.1:2)';
```

```
logs = [x log10(x)]
```

κατασκευάζουν ένα λογαριθμικό πίνακα.

```
logs =  
    1.0         0  
    1.1    0.04139  
    1.2    0.07918  
    1.3    0.11394  
    1.4    0.14613  
    1.5    0.17609  
    1.6    0.20412  
    1.7    0.23045  
    1.8    0.25527  
    1.9    0.27875  
    2.0    0.30103
```

Δεδομένα πολλών μεταβλητών

Η διαχείριση στατιστικών δεδομένων πολλών μεταβλητών γίνεται κατά στήλες. Κάθε στήλη παριστά μία μεταβλητή και κάθε γραμμή μία παρατήρηση. Το στοιχείο (i,j) είναι η i -στη παρατήρηση της j -στης

μεταβλητής. Για παράδειγμα, θεωρήσατε τα ακόλουθα δεδομένα τριών μεταβλητών:

- Καρδιακός ρυθμός
- Βάρος
- Ώρες άσκησης εβδομαδιαίως

Για ένα σύνολο πέντε παρατηρήσεων, τα κατά στήλες διατεταγμένα αριθμητικά δεδομένα θα έχουν την μορφή:

```
D = [ 72          134          3.2
      81          201          3.5
      69          156          7.1
      82          148          2.4
      75          170          1.2 ]
```

Η πρώτη γραμμή περιέχει τον καρδιακό ρυθμό, βάρος και ώρες άσκησης για τον ασθενή 1, η δεύτερη γραμμή τα αυτά για τον ασθενή 2 κ.ο.κ. Στα δεδομένα αυτά μπορούν να εφαρμοσθούν απ' ευθείας συναρτήσεις ανάλυσης. Για παράδειγμα, για να λάβουμε τον μέσο και τυπική απόκλιση εκάστης στήλης εντελλόμαστε:

```
mu = mean(D), sigma = std(D)
mu =
    75.8    161.8     3.48
sigma =
    5.6303    25.499     2.2107
```

Ο κατάλογος των διαθέσιμων συναρτήσεων ανάλυσης θα εμφανισθεί με την εντολή

```
help datafun
```

Εάν υπάρχει πρόσβαση στην Στατιστική Εργαλειοθήκη του MATLAB επίσης δώσατε

```
help stats
```

Πράξεις με πίνακες και μονόμετρα μεγέθη

Πίνακες και μονόμετρες παράμετροι μπορούν να συνδυασθούν με αρκετούς διαφορετικούς τρόπους. Παραδείγματος χάριν, ένα μονόμετρο αφαιρείται από έναν πίνακα με αφαίρεσή του από κάθε στοιχείο του πίνακα. Η μέση τιμή των στοιχείων του μαγικού τετραγώνου μας είναι 8.5, ούτως ώστε η

```
B = A - 8.5
```

Να δημιουργεί έναν πίνακα του οποίου τα αθροίσματα των στηλών είναι μηδέν:

```
B =
    7.5    -5.5    -6.5     4.5
   -3.5     1.5     2.5    -0.5
    0.5    -2.5    -1.5     3.5
   -4.5     6.5     5.5    -7.5
```

```
sum(B)
```

```
ans =
     0     0     0     0
```

Με ανάλογο τρόπο μπορούν να δοθεί αριθμητική τιμή σε όλα τα στοιχεία εντός δεδομένου εύρους δεικτών. Για παράδειγμα, η

```
B(1:2, 2:3) = 0
```

μηδενίζει zeroes ένα μέρος του B:

```
B =
    7.5     0     0     4.5
   -3.5     0     0    -0.5
    0.5    -2.5    -1.5     3.5
   -4.5     6.5     5.5    -7.5
```

Λογικοί δείκτες

Ένας διαφορετικός τρόπος ανάκλησης μερισμάτων ενός πίνακα είναι τα λογικά ανύσματα που κατασκευάζονται μέσω λογικών και σχεσιακών πράξεων. Υποθετήσθω ότι ο X είναι ένας συνήθης πίνακας και ότι ο L είναι ένας πίνακας του αυτού μεγέθους, ο οποίος προκύπτει μέσω κάποιας λογικής πράξης. Τότε ο X(L) επιλέγει τα στοιχεία του X για τα οποία τα στοιχεία του L είναι μη-μηδενικά. Για παρά-

δειγμα, έστω η ακόλουθη σειρά δεδομένων:

```
x = [2.1 1.7 1.6 1.5 NaN 1.9 1.8 1.5 5.1 1.8 1.4 2.2 1.6 1.8];
```

Στην σειρά αυτή, έστω ότι ο NaN σημειώνει μία απύουσα παρατήρηση, όπως για παράδειγμα την αστοχία ενός οργάνου κάτω από ειδικές συνθήκες μέτρησης. Προκειμένου να αφαιρεθεί η απύουσα παρατήρηση και να μην μολύνει την ανάλυση, μπορούμε να χρησιμοποιήσουμε λογικούς δείκτες που θα λάβομε μέσω της συνάρτησης *isfinite(x)*. Το εξαγόμενο αυτής της συνάρτησης είναι αληθές για όλους τους πεπερασμένους αριθμούς και ψευδές για NaN και Inf:

```
y = isfinite(x)
y =
     1     1     1     1     0     1     1     1     1     1     1     1     1
x = x(isfinite(x))
x =
     2.1     1.7     1.6     1.5     1.9     1.8     1.5     5.1     1.8     1.4     2.2     1.6     1.8
```

Παρατηρήσατε τώρα ότι υπάρχει μία παρατήρηση, 5.1, η οποία φαίνεται πολύ διαφορετική από τις υπόλοιπες. Πρόκειται για μία *έξαρση (outlier)*. Η επόμενη οδηγία αφαιρεί τι εξάρσεις, στην προκειμένη περίπτωση όλα τα στοιχεία με μέγεθος μεγαλύτερο των τριών τυπικών αποκλίσεων από τον μέσο όρο:

```
x = x(abs(x-mean(x)) <= 3*std(x))
x =
     2.1     1.7     1.6     1.5     1.9     1.8     1.5     1.8     1.4     2.2     1.6     1.8
```

Ένα ακόμη παράδειγμα είναι ο εντοπισμός της θέσης των πρώτων αριθμών στο μαγικό τετράγωνο του Dürer, χρησιμοποιώντας λογικούς δείκτες και αντικατάσταση των μη πρώτων αριθμών με μηδέν.

```
A(~isprime(A)) = 0
A =
     0     3     2    13
     5     0    11     0
     0     0     7     0
     0     0     0     0
```

Η συνάρτηση *find* (εύρεση)

Η συνάρτηση *find* ανευρίσκει τους δείκτες εκείνων των στοιχείων, τα οποία υπακούουν δεδομένη λογική συνθήκη. Στην απλούστερη μορφή της, η *find* επιστρέφει ένα κατακόρυφο άνυσμα δεικτών. Για παράδειγμα, ας ξαναρχίσουμε με το μαγικό τετράγωνο του Dürer. Η

```
k = find(isprime(A))'
```

επιλέγει τους δείκτες των πρώτων αριθμών και επιστρέφει το αποτέλεσμα σε ένα οριζόντιο άνυσμα:

```
k =
     2     5     9    10    11    13
```

Οι ανευρεθέντες πρώτοι αριθμοί κατά την σειρά που εμφανίζονται στο k είναι:

```
A(k)
ans =
     5     3     2    11     7    13
```

Όταν το k χρησιμοποιηθεί ως άνυσμα δεικτών σε μία πράξη αντικατάστασης, η δομή του πίνακα διατηρείται:

```
A(k) = NaN
A =
    16    NaN    NaN    NaN
    NaN    10    NaN     8
     9     6    NaN    12
     4    15    14     1
```

Εμφάνιση δεδομένων και αποτελεσμάτων στο Παράθυρο Εντολών.

Η οδηγία format (διαμόρφωση)

Η `format` ελέγχει τον τύπο των αριθμητικών παραστάσεων που εμφανίζονται στο παράθυρο εντολών. Τονίζεται ότι μόνο ο τρόπος με τον οποίο εμφανίζονται οι αριθμοί επηρεάζεται και όχι ο τρόπος με τον οποίο υπολογίζονται και καταχωρούνται στην μνήμη. Ακολουθούν ορισμένα παραδείγματα διαφορετικών αριθμητικών μορφών:

```
x = [4/3 1.2345e-6]
format short
    1.3333    0.0000
format short e
    1.3333e+000  1.2345e-006
format short g
    1.3333  1.2345e-006
format long
    1.3333333333333333  0.00000123450000
format long e
    1.3333333333333333e+000  1.2345000000000000e-006
format long g
    1.3333333333333333  1.2345e-006
format bank
    1.33    0.00
format rat
    4/3    1/810045
format hex
    3ff5555555555555  3eb4b6231abfd271
```

Επιπλέον, η οδηγία

```
format compact
```

καταστέλλει τις κενές γραμμές που εμφανίζονται κατά την παρουσίαση των αποτελεσμάτων, ώστε να χωρά περισσότερη πληροφορία στο περιβάλλον εργασίας. Εάν απαιτείται περισσότερος έλεγχος ή εμφάνιση των αποτελεσμάτων με ειδική μορφή, πρέπει να χρησιμοποιηθούν οι συναρτήσεις *sprintf* και *fprintf*.

Αναστολή παρουσίασης αποτελεσμάτων

Εάν απλά δακτυλογραφηθεί μια εντολή και `↵`, τότε τα αποτελέσματα εμφανίζονται αυτόματα στην οθόνη. Εάν όμως η εντολή τελειώνει με ερωτηματικό, `;`, τότε γίνονται οι υπολογισμοί αλλά δεν εμφανίζεται το αποτέλεσμα. Αυτό είναι πολύ χρήσιμο κατά την διαχείριση μεγάλου όγκου δεδομένων αλλά και για την συγγραφή προγραμμάτων.

Εισαγωγή μακρών εκφράσεων

Εάν μία εντολή δεν χωράει σε μία γραμμή, χρησιμοποιούνται τρεις συνεχόμενες τελείες (...) ακολουθούμενες από `↵` για να δεχθεί ότι η εντολή συνεχίζεται στην επόμενη γραμμή. Για παράδειγμα,

```
s = 1 -1/2 + 1/3 -1/4 + 1/5 - 1/6 + 1/7...
    - 1/8 + 1/9 - 1/10 + 1/11 - 1/12;
```

Τα κενά γύρω από τα σύμβολα `=`, `+`, `and` `-` δεν επηρεάζουν την πράξη, είναι προαιρετικά και βοηθούν στην ανάγνωση.

Command Line Editing

Προσφέρονται αρκετά πλήκτρα ελέγχου που διευκολύνουν την ανάκληση, ανασύνταξη και επαναχρησιμοποίηση εντολών που έχουν δοθεί ενωρίτερα.. Ακολουθεί κατάλογος των πλήκτρων κατεύθυνσης

και ελέγχου που μπορούν να χρησιμοποιηθούν στο παράθυρο εντολών. Σημειώνεται όμως ότι τα πλήκτρα ελέγχου (συνδυασμοί **Ctrl**+πλήκτρο) μπορούν να χρησιμοποιηθούν μόνον εάν έχει επιλεγεί ο **Emacs** ως πρότυπο συντάκτη κειμένου για το παράθυρο εντολών.

Key	Control Key for Emacs (MATLAB standard) Preference	Operation
↑	Ctrl+P	Ανάκληση προηγούμενης γραμμής. Μόνο για εργασίες στο παράθυρο εντολών.
↓	Ctrl+N	Ανάκληση επόμενης γραμμής. Μόνο για εργασίες στο παράθυρο εντολών και εφόσον έχει ήδη χρησιμοποιηθεί το άνω βέλος ή Ctrl+P .
←	Ctrl+B	Μετακίνηση πίσω κατά ένα χαρακτήρα.
→	Ctrl+F	Μετακίνηση εμπρός κατά ένα χαρακτήρα.
Ctrl+→		Μετακίνηση δεξιά κατά μία λέξη.
Ctrl+←		Μετακίνηση αριστερά κατά μία λέξη
Home	Ctrl+A	Μετακίνηση στην αρχή της εντολής.
End	Ctrl+E	Μετακίνηση στο τέλος της εντολής
Ctrl+Home		Μετακίνηση στην αρχή του παραθύρου εντολών.
Ctrl+End		Μετακίνηση στο τέλος του παραθύρου εντολών.
Esc	Ctrl+U	Διαγραφή εντολής.
Delete	Ctrl+D	Διαγραφή χαρακτήρα στον δρομέα (γραμμή εντολής).
Backspace	Ctrl+H	Διαγραφή χαρακτήρα προ του δρομέα (γραμμή εντολής).
	Ctrl+K	Διαγραφή γραμμής εντολής από το δρομέα μέχρι το τέλος.
Shift+Home		Επιλογή από τον δρομέα μέχρι την αρχή.
Shift+End		Επιλογή από τον δρομέα μέχρι το τέλος. Μπορεί να γίνει από οπουδήποτε στο παράθυρο εντολών.

3 Γραφικά

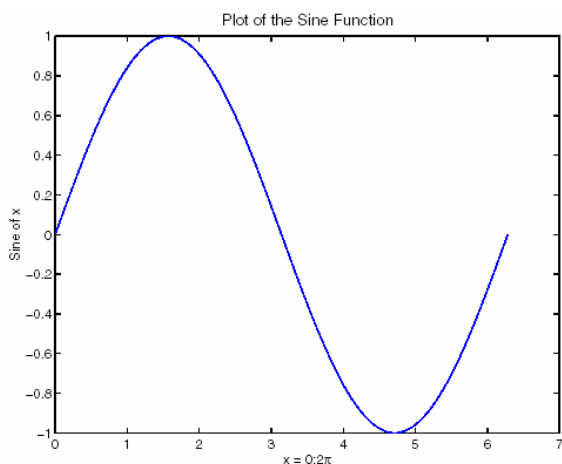
Το MATLAB και η OCTAVE παρέχουν ευρεία ποικιλία τεχνικών γραφικής παρουσίασης δεδομένων και αποτελεσμάτων, εκτύπωσης των γραφημάτων και εξαγωγής τους σε τυπικά αρχεία γραφικών για ηλεκτρονική παρουσίαση. Γενικά, η γραφική παρουσίαση των δεδομένων περιλαμβάνει μία σειρά πράξεων, οι λεπτομέρειες των οποίων διαφέρουν κατά περίπτωση. Αυτό οφείλεται εν μέρει στο γεγονός ότι ο επιλεχθησόμενος τύπος γραφήματος εξαρτάται από την φύση των δεδομένων και την πληροφορία που πρέπει να εξαχθεί από αυτά και να καταδειχθεί. Σε γενικές γραμμές παρέχονται αρκετοί προπρογραμματισμένοι τύποι γραφημάτων, από συμβατικές γραφικές παραστάσεις, ιστογράμματα και pie charts, μέχρι τριδιάστατες απεικονίσεις επιφανειών. Ο βασικός τρόπος κατασκευής γραφημάτων είναι άμεσες εντολές μέσω του παραθύρου εντολών, ή με κατασκευή γραφικών προγραμμάτων μέσω της γλώσσας MATLAB. Στο παρόν κεφάλαιο θα δοθεί μόνο μία γενική εικόνα της διαδικασίας κατασκευής γραφημάτων, με παραδείγματα και λεπτομέρειες για την εκτέλεση των κυριότερων πράξεων.

Βασικές γραφικές συναρτήσεις

Κατασκευή γραφήματος

Η συνάρτηση `plot` λαμβάνει αρκετές διαφορετικές μορφές, ανάλογα με τα ορίσματα που της δίδονται. Εάν `y` είναι ένα οιοδήποτε άνυσμα, τότε η `plot(y)` δημιουργεί μία γραμμική απεικόνιση των στοιχείων του `y` ως προς τους δείκτες των στοιχείων του `y`. Εάν ως ορίσματα προσδιορισθούν δύο ισομήκη ανύσματα, τότε η `plot(x,y)` δημιουργεί μία γραφική παράσταση του `y` (εξαρτημένη μεταβλητή) ως προς `x` (ανεξάρτητη μεταβλητή). Στο ακόλουθο παράδειγμα, κατασκευάζονται ένα άνυσμα `x` γωνιών που μεταβάλλονται από 0 ως 2π , ένα άνυσμα `y` του ημιτόνου αυτών των γωνιών και μία γραφική παράσταση του αποτελέσματος:

```
x = 0:pi/100:2*pi;  
y = sin(x);  
plot(x,y);
```



Τώρα ονομάζομε τους άξονες και προσθέτομε έναν τίτλο. Οι χαρακτήρες `\pi` αντιπροσωπεύουν το σύμβολο π :

```
xlabel('x = 0:2\pi')  
ylabel('Sine of x')
```

```
title('Plot of the Sine Function','FontSize',12)
```

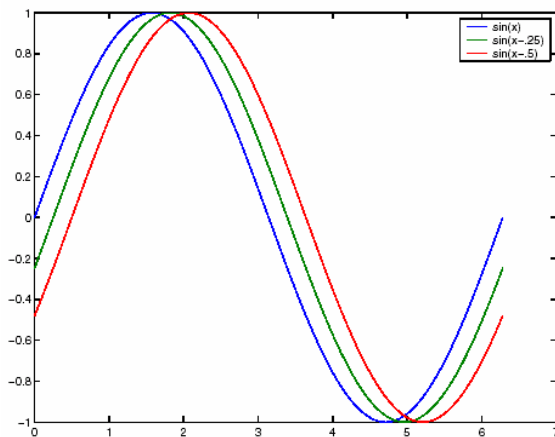
Πολλαπλά δεδομένα σε ένα γράφημα.

Η συνάρτηση plot μπορεί να δεχθεί πολλαπλά ζεύγη x-y στο όρισμά της και να χαράξει πολλαπλά γραφήματα με μία μόνο κλήση. Για εύκολη διάκριση διαφορετικών καμπυλών, το σύστημα αντιστοιχεί ένα χρώμα σε κάθε γραφική παράσταση, αλλά ο χρήστης έχει την δυνατότητα να επεμβεί ορίζοντας χρώματα της επιλογής του (βλ. κατωτέρω). Για παράδειγμα, οι ακόλουθες εντολές χαράσσουν τρία ημιτονοειδή με διαφορά φάσης:

```
x = 0:pi/100:2*pi;  
y = sin(x);  
y2 = sin(x-.25);  
y3 = sin(x-.5);  
plot(x,y,x,y2,x,y3)
```

Η εντολή legend προσφέρει ένα εύκολο τρόπο αναγνώρισης της ταυτότητας εκάστης καμπύλης (μόνο στο MATLAB):

```
legend('sin(x)', 'sin(x-.25)', 'sin(x-.5)')
```



Γραμμές και Χρώματα

Η συνάρτηση plot επιτρέπει τον ορισμό του χρώματος, της μορφής της καμπύλης κατά την κλήση της. Στην περίπτωση αυτή η κλήση έχει την μορφή:

```
plot(x,y,'χρώμα_είδος_σύμβολο')
```

όπου *χρώμα_είδος_σύμβολο* είναι μία σειρά ενός έως τεσσάρων χαρακτήρων που ρυθμίζει το χρώμα, τον τύπο της γραμμής και/ή το σύμβολο με το οποίο θα εμφανισθεί η χαρακτηρισμένη καμπύλη:

- Τα χρώματα προσδιορίζονται με τους χαρακτήρες 'c', 'm', 'y', 'r', 'g', 'b', 'w', and 'k', οι οποίοι αντιστοιχούν στα γαλάζιο (cyan), ιώδες (magenta), κίτρινο, ερυθρό, πράσινο, κυανούν (blue), λευκό και μέλαν (μαύρο).
- Τα είδη της γραμμής⁴ είναι '-' για συνεχή, '--' για διακεκομμένη, ':' για στικτή, '-.' για διακεκομμένη-στικτή. Εάν δεν θα χαραχθεί γραμμή αλλά μόνο σύμβολα, οι χαρακτήρες αυτοί μπορούν να παραλειφθούν.
- Οι τύποι των συμβόλων είναι: '+', 'o', '*', and 'x'. Ειδικότερα για το MATLAB, 's' για τετράγωνο, 'd' για ρόμβο, '^' για τρίγωνο με την κορυφή άνω, 'v' για τρίγωνο κάτω, '>' για τρίγωνο δεξιά, '<' για τρίγωνο αριστερά, 'p' για πεντάγωνο, 'h' για εξάγωνο και τίποτε εάν δεν είναι επιθυμητή η απεικόνιση δια συμβόλων.

Εάν ορισθεί ένα σύμβολο αλλά όχι ένα είδος γραμμής, τότε η καμπύλη χαράσσεται μόνο με το σύμβολο. Για παράδειγμα, η

⁴ Στο MATLAB. Με την OCTAVE η κατάσταση είναι περίπλοκη: οι επιλογές αυτές δεν ισχύουν για την έκδοση της Sourceforge για Windows και του GNUPLOT που χρησιμοποιείται εκεί.

```
plot(x,y,'k+')
```

χαράσσει μαύρους σταυρούς στην θέση κάθε ζεύγους δεδομένων, αλλά δεν τα συνδέει με μία γραμμή.

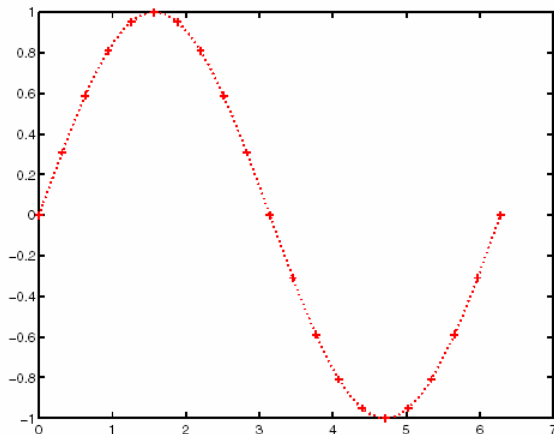
Η εντολή

```
plot(x,y,'r:+')
```

χαράσσει μία ερυθρά στικτή γραμμή και τοποθετεί σταυρούς στην θέση εκάστου ζεύγους δεδομένων.

Στην ίδια κλήση της plot, είναι επιτρεπτοί συνδυασμοί ζευγών x-y με διαφορετικές διαστάσεις. Το ακόλουθο παράδειγμα χαράσσει την καμπύλη δύο φορές, αλλά με διαφορετικό αριθμό σημείων για την στικτή γραμμή και το σύμβολο +:

```
x1 = 0:pi/100:2*pi; % Το άνυσμα x1 έχει 101 στοιχεία  
x2 = 0:pi/10:2*pi; % Το άνυσμα x2 έχει μόνο 10 στοιχεία  
plot(x1,sin(x1),'r:',x2,sin(x2),'r+')
```



Φανταστικά και Μιγαδικά δεδομένα

Όταν τα ορίσματα της plot είναι μιγαδικά, το φανταστικό μέρος παραλείπεται, εκτός εάν το όρισμα είναι ένα και μόνο ένα μιγαδικό άνυσμα. Στην ειδική αυτή περίπτωση, Η plot χαράσσει την γραφική παράσταση του πραγματικού (x) ως προς το φανταστικό (y) μέρος.. Συνεπώς, η

```
plot(Z)
```

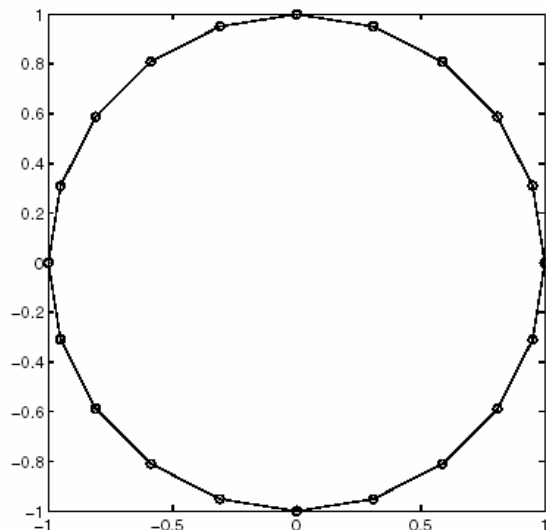
όπου Z μιγαδικό άνυσμα ή πίνακας, είναι ισοδύναμη προς την

```
plot(real(Z), imag(Z))
```

Για παράδειγμα, οι

```
t = 0:pi/10:2*pi; plot(exp(i*t),'-o'); axis equal
```

χαράσσει ένα 20-πλευρο πολύγωνο με μικρούς κύκλους στις κορυφές. Η εντολή *axis equal* ταυτίζει την κλίμακα των αξόνων x και y, πράγμα που προσδίδει 'κυκλική' εμφάνιση στο γράφημα.



Υπέρθωση παραστάσεων σε υπάρχον γράφημα

Η εντολή `hold` επιτρέπει την υπέρθεση νέων γραφικών παραστάσεων σε ένα ήδη υπάρχον γράφημα. Δακτυλογραφώντας

```
hold on
```

το υπάρχον γράφημα δεν αντικαθίσταται όταν δοθούν νέες εντολές γραφικών, αλλά η νέα γραφική παράσταση υπερτίθεται στην παλαιά, με αναμόρφωση των αξόνων όταν αυτό είναι απαραίτητο. Για παράδειγμα, οι ακόλουθες εντολές χαράσσουν το κυανούν ημίτονο του ανύσματος x και υπερθέτουν το ανεξάρτητα υπολογισμένο ερυθρό συνημίτονο ως συνδυασμό γραμμής και συμβόλου)

```
x=0:pi/50:2*pi;
```

```
y=sin(x);
```

```
plot(x,y)
```

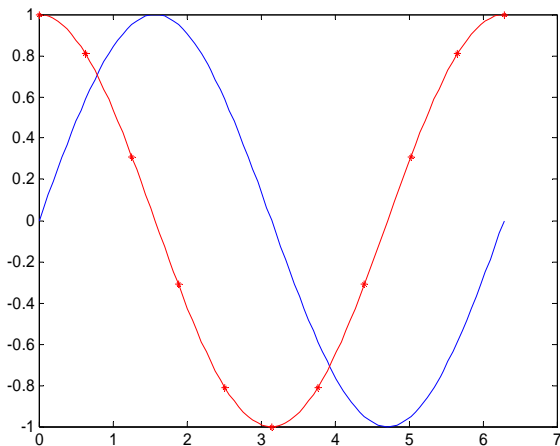
```
hold on
```

```
x1=0:pi/5:2*pi;
```

```
y1=cos(x);
```

```
plot(x,y1,'r-',x1,y1,'r*')
```

Η εντολή `hold` αναγκάζει το γράφημα του συνημιτόνου να συνδυασθεί με το γράφημα του ημιτόνου σε μία εικόνα.



Παράθυρα Γραφικών (Εικόνων)

Οι συναρτήσεις των δημιουργούν αυτόματα νέα παράθυρα γραφικών, εφόσον δεν υπάρχουν ήδη διαθέσιμα παράθυρα στην οθόνη. Εφόσον ήδη υπάρχει ένα ανοικτό παράθυρο, τότε το νέο γράφημα κατευθύνεται προς αυτό. Εάν υπάρχουν πολλά ανοικτά παράθυρα γραφικών, τότε επιλέγεται το τελευταίο χρησιμοποιηθέν (αποτελούμενο και *τρέχουσα εικόνα*- *current figure*). Για να καταστήσουμε ένα ανοικτό παράθυρο «τρέχον», το δείχνουμε με τον δρομέα ή δίδουμε την εντολή

```
figure(n)
```

όπου n είναι ο αριθμός του ανοικτού παραθύρου. Τα μετέπειτα γραφικά όλα θα χαράσσονται σ' αυτό το παράθυρο, μέχρι αντικατάστασής του από ένα άλλο. Για να ανοίξουμε ένα νέο παράθυρο γραφικών και να το καταστήσουμε «τρέχον», δακτυλογραφούμε

```
figure
```

Καθαρισμός παραθύρου για νέο γράφημα:

Όταν υπάρχει ήδη ένα ανοικτό παράθυρο γραφικών, οι επόμενες γραφικές εντολές κατασκευάζουν μεν νέο γράφημα, αλλά δεν αποκαθιστούν τις ιδιότητες του παραθύρου όπως *φόντο*(*background color*),

χρώματα (*colormap*) κ.ά.. Ένα είναι επιθυμούμε την αποκατάσταση του παραθύρου, χρησιμοποιούμε την εντολή *clf* με την επιλογή *reset*,
`clf reset`
 πριν δημιουργήσουμε το νέο γράφημα.

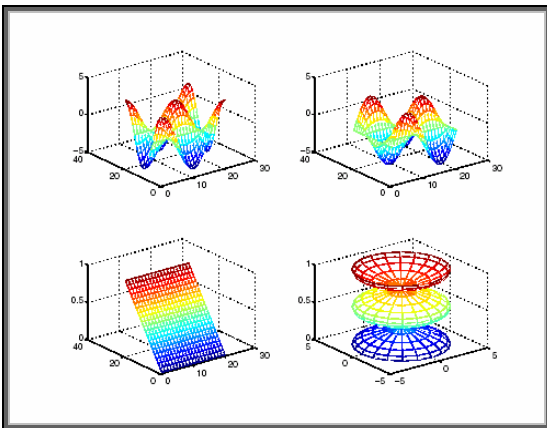
Πολλαπλά γραφήματα σε μία εικόνα (παράθυρο)

Η εντολή *subplot* command επιτρέπει την παρουσίαση πολλαπλών γραφημάτων στο ίδιο παράθυρο (και εκτύπωσή τους στο ίδιο φύλλο χαρτιού). Εντελλόμενοι

```
subplot(m,n,p)
```

διαμερίζουμε το παράθυρο γραφικών σε ένα πίνακα $m \times n$ υπο-εικόνων και επιλέγουμε την p -στη υπο-εικόνα προς την οποία κατευθύνουμε το τρέχον γράφημα. Τα γραφήματα αριθμούνται από την άνω δεξιά γωνία και κατά μήκος της πρώτης σειράς υπο-εικόνων, της δεύτερης κ.ο.κ.. Οι ακόλουθες εντολές κατασκευάζουν γραφήματα σε τέσσερες διαφορετικές υποπεριοχές του παραθύρου γραφικών:

```
t = 0:pi/10:2*pi;
[X,Y,Z] = cylinder(4*cos(t));
subplot(2,2,1); mesh(X);
subplot(2,2,2); mesh(Y);
subplot(2,2,3); mesh(Z);
subplot(2,2,4); mesh(X,Y,Z);
```



Έλεγχος των αξόνων

Η εντολή *axis* προσφέρει ορισμένες δυνατότητες ρύθμισης της κλίμακας, διεύθυνσης και λόγου αξόνων (aspect ratio) των γραφημάτων.

Ρύθμιση ορίων αξόνων: Η αυτόματη ρύθμιση είναι ότι το σύστημα θα εντοπίσει τα μέγιστα και ελάχιστα των δεδομένων και θα επιλέξει άξονες που να χωρούν αυτά τα όρια. Η εντολή *axis* επιτρέπει την ρύθμιση των ορίων των αξόνων από τον χρήστη:

```
axis([xmin xmax ymin ymax])
```

ή, για τριδιάστατα γραφήματα,

```
axis([xmin xmax ymin ymax zmin zmax])
```

Η εντολή

```
Axis('auto')
```

επαναφέρει τον αυτόματο τρόπο ρύθμισης των ορίων.

Ρύθμιση λόγου αξόνων: Η *axis* προσφέρει ορισμένες προεπιλογές. Για παράδειγμα, η

```
axis('square')
```

καθιστά τα μήκη των αξόνων x - και y - ίσα. Η

```
axis('equal')
```

ταυτίζει την κλίμακα των αξόνων x και y . Η

```
axis('normal')
```

επαναφέρει την κλίμακα των αξόνων στην αυτόματη ρύθμιση.

Ρύθμιση ορατότητας αξόνων: Με την εντολή `axis` μπορούμε επίσης να καθορίσουμε εάν οι άξονες θα είναι ορατοί ή όχι. Η

```
axis('on')
```

κάνει τους άξονες ορατούς και είναι η αυτόματη ρύθμιση. Η

```
axis('off')
```

καθιστά τους άξονες αόρατους.

Πλέγματα: Η εντολή `grid` ορίζει εάν θα εμφανισθεί πλέγμα αξόνων ή όχι. Η εντολή

```
grid('on')
```

εμφανίζει το πλέγμα και η εντολή

```
grid('off')
```

το διαγράφει.

Υπομνήματα και τίτλοι

Οι εντολές `xlabel`, `ylabel`, και `zlabel` προσθέτουν υπομνήματα στους άξονες x -, y -, και z - αντιστοίχως. Η εντολή `title` προσθέτει ένα τίτλο στην κορυφή του γραφήματος και η συνάρτηση `text` εισάγει υπομνηματικό κείμενο οπουδήποτε στην εικόνα. Το MATLAB ενσωματώνει διερμηνέα LaTeX, μέσω του οποίου μπορούν να γραφούν Ελληνικοί χαρακτήρες και μαθηματικά σύμβολα στα υπομνήματα, χρησιμοποιώντας αντίστοιχο συμβολισμό:

```
t = -pi:pi/100:pi;
```

```
y = sin(t);
```

```
plot(t,y)
```

```
axis([-pi pi -1 1])
```

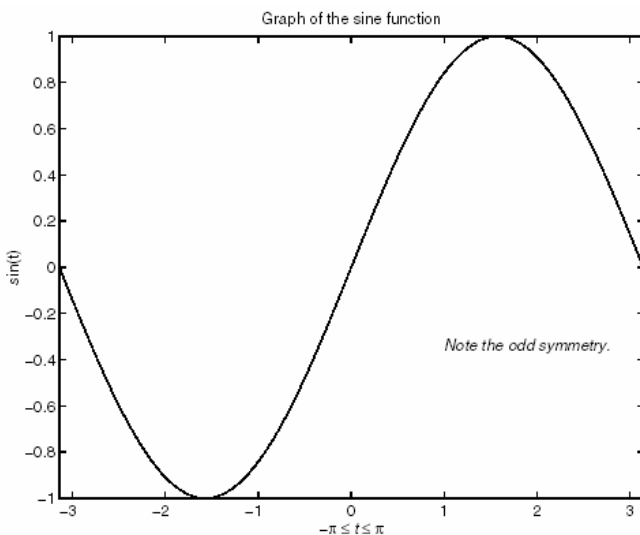
```
xlabel('-\pi \leq t \leq \pi')
```

```
ylabel('sin(t)')
```

```
title('Graph of the sine function')
```

```
text(1,-1/3,{'\itNote the odd symmetry.}')
```

Σημειώσατε ότι η θέση του κειμένου ορίζεται στις μονάδες των αξόνων (δεδομένων).



Τριδιάστατα γραφήματα

Μία επιφάνεια ορίζεται από τις συντεταγμένες z των σημείων επάνω από ένα κάναβο στο επίπεδο $x-y$, και εικονίζεται με ευθύγραμμα τμήματα που συνδέουν γειτονικά σημεία. Οι συναρτήσεις *mesh* και *surf* απεικονίζουν επιφάνειες σε τρεις διαστάσεις. Η *mesh* αναπαριστά την επιφάνεια ως πλέγμα (wireframe) των γραμμών που συνδέουν γειτονικά σημεία. Η *surf* απεικονίζει τόσο το πλέγμα, όσο και τις έδρες (τετράπλευρα) του πλέγματος. Η παλέτα (*colormap*) της εικόνας καθορίζει πως χρωματίζεται η επιφάνεια.

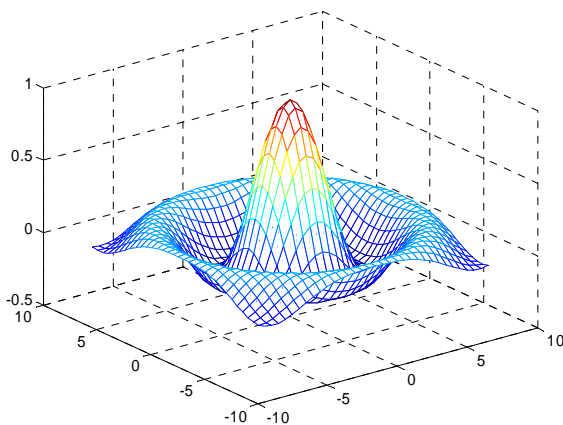
Γραφική παράσταση συναρτήσεων δύο μεταβλητών

Για να παρασταθεί μία συνάρτηση δύο μεταβλητών, $z = f(x,y)$,

1. Χρησιμοποιούμε την συνάρτηση *meshgrid* για να κατασκευάζουμε πίνακες X και Y υπεράνω του πεδίου ορισμού της z , αντίστοιχα αποτελούμενους από επαναλαμβανόμενες γραμμές (σειρές) και στήλες. Το πεδίο ορισμού εισάγεται στην *meshgrid* υπό μορφή ενός ή δύο ανυσμάτων x και y . Οι γραμμές του X είναι αντίγραφα του ανύσματος x και οι στήλες του Y είναι αντίγραφα του y .
2. Χρησιμοποιούμε τους X και Y για να υπολογίσουμε την z .
3. Χρησιμοποιούμε τις *mesh* ή *surf* για να εικονίσουμε την z .

Γραφική παράσταση της συνάρτησης *sinc*. Στο παράδειγμα αυτό θα εκτιμήσουμε και εικονίσουμε την διδιάστατη συνάρτηση *sinc*, $(\sin(R)/R)$, όπου R είναι η απόσταση από την αρχή του συστήματος συντεταγμένων. Πριν τον υπολογισμό του ημιτόνου, προσθέτουμε *eps* προκειμένου να αποφύγουμε απροσδιοριστία του τύπου $0/0$ στην αρχή των συντεταγμένων:

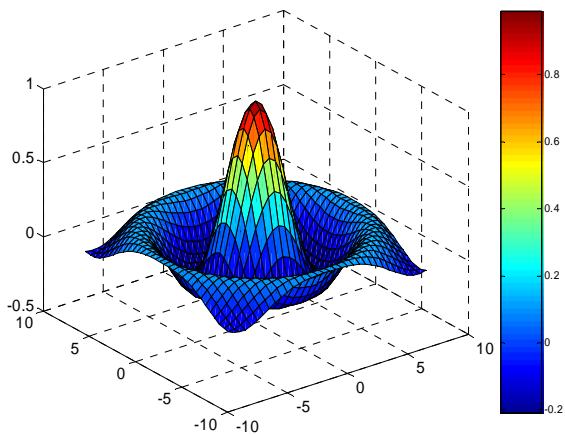
```
[X,Y] = meshgrid(-8:.5:8, -8:.5:8);  
R = sqrt(X.^2 + Y.^2) + eps;  
Z = sin(R)./R;  
mesh(X,Y,Z)
```



Έγχρωμες επιφάνειες: Η γραφική αναπαράσταση μίας επιφάνειας με την συνάρτηση *surf* είναι παρόμοια, με την διαφορά ότι στην περίπτωση αυτή χρωματίζονται και οι τετράπλευρες έδρες της επιφάνειας. Το χρώμα κάθε έδρας καθορίζεται από τις τιμές του Z και την παλέτα (*colormap*), δηλ. την διάταξη χρωμάτων που έχει επιλεγεί. Οι ακόλουθες εντολές εικονίζουν την συνάρτηση *sinc* ως έγχρωμη επιφάνεια και προσθέτουν την χρωματική κλίμακα (*colorbar*) για να δειχθεί η αντιστοίχιση των δεδομένων (τιμών του Z) σε χρώμα⁵:

```
surf(X,Y,Z);  
colorbar
```

⁵ Η συνάρτηση *colorbar* λειτουργεί μόνο στο MATLAB.



Εικόνες (images)

Οι διδιάστατοι πίνακες μπορούν να παρασταθούν ως εικόνες (*images*), στην οποία περίπτωση τα στοιχεία τους ορίζουν την φωτεινότητα ή το χρώμα των εικόνων. Οι εντολές

```
load durer
whos
```

Name	Size	Bytes	Class
X	648x509	2638656	double array
caption	2x28	112	char array
map	128x3	3072	double array

φορτώνουν (διαβάζουν) τον φάκελο `durer.mat`, προσθέτοντας τρεις μεταβλητές στον χώρο εργασίας. Ο πίνακας `X` έχει διαστάσεις 648×509 , ο δε `map` 128×3 και περιέχει την παλέτα (`colormap`) της εικόνας. Οι φάκελοι με επέκταση `.MAT`, όπως ο `durer.mat`, είναι δυαδικοί φάκελοι που μπορούν να δημιουργηθούν με το MATLAB ή την OCTAVE. Τα στοιχεία του `X` είναι ακέραιοι μεταξύ 1 και 128, οι οποίοι χρησιμοποιούνται ως δείκτες για επιλογή χρώματος από την παλέτα. Τότε, οι

```
image(X);
colormap(map);
axis('image')
```

αναπαράγει το χαρακτηριστικό του Dürer. Η εντολή

```
colormap(hot)
```

προσθέτει μοντέρνο χρωματισμό στο χαρακτηριστικό του 16^{ου} αιώνα. Η συνάρτηση `hot` δημιουργεί μία παλέτα που περιέχει αποχρώσεις του ερυθρού, πορτοκαλί και κίτρινου.

Τα τυπικά αρχεία εικόνας (TIFF, JPEG, BMP, κ.λπ.) διαβάζονται χρησιμοποιώντας την συνάρτηση `imread`. Το είδος των δεδομένων που επιστρέφει η `imread` εξαρτάται από το είδος της εικόνας που εισάγεται. Μπορούμε επίσης να γράψουμε δεδομένα (πίνακες) σε ποικίλα τυπικών αρχείων εικόνας χρησιμοποιώντας την συνάρτηση `imwrite`. Λεπτομέρειες για αυτές τις δυνατότητες θα δοθούν προσεχώς.

Εκτύπωση γραφικών

Ένα γράφημα μπορεί να τυπωθεί απ' ευθείας άμεσα στον εκτυπωτή που συνδέεται με τον υπολογιστή ή μπορεί να εξαχθεί ως εικόνα σε ένα από τα τυπικά format γραφικών, περιλαμβανομένων των TIFF, JPEG και PostScript, χρησιμοποιώντας την εντολή `print`. Το ακόλουθο παράδειγμα σώζει το περιεχόμενο της τρέχουσας εικόνας στο αρχείο `magicsquare.eps`, ως έγχρωμο Encapsulated PostScript Επιπέδου 2. Το format αυτό ενσωματώνει και μία εικόνα προεπισκόπησης TIFF, πράγμα που επιτρέπει την αναπαράστασή του από τους περισσότερους επεξεργαστές κειμένου:

```
print -depsc2 -tiff magicsquare.eps
```

Για να σωθεί η ίδια εικόνα σε format TIFF με ανάλυση 200 dpi, χρησιμοποιείται η εντολή

```
print -dtiff -r200 magicsquare.tiff
```

Εάν δοθεί μόνο η εντολή

```
print
```

τότε εκτυπώνεται η τρέχουσα εικόνα στον προεπιλεγμένο εκτυπωτή.

4 Προγραμματισμός

Ένα πρόγραμμα, γενικά *απαρτίζεται* από:

1. Τα δεδομένα και τους τύπους δεδομένων τους οποίους χειρίζεται,
2. Την λογική δομή του εφαρμοζόμενου αλγορίθμου, η οποία υλοποιείται με οδηγίες *ελέγχου της ροής* (εκτέλεσης) και *λήψης αποφάσεων*, και,
3. *Εντολές αντικατάστασης* (υπολογιστικές πράξεις) με τις οποίες παράγονται οι αναγκαίες πληροφορίες για έλεγχο ροής και λήψη αποφάσεων, και τα αριθμητικά αποτελέσματα.

Μέχρι τώρα, έχουμε εξετάσει κυρίως πράξεις αντικατάστασης (απόδοσης αριθμητικής τιμής σε μία μεταβλητή), αν και στις περισσότερες περιπτώσεις η αντικατάσταση γίνεται με το αποτέλεσμα μίας συνάρτησης που αφ' εαυτής είναι ένα αυτοδύναμο πρόγραμμα... Στο παρόν κεφάλαιο θα εστιάσουμε κυρίως στην περιγραφή των τύπων των δεδομένων που μπορεί να χειρισθεί ένα πρόγραμμα και των οδηγιών ελέγχου ροής, και θα επιχειρήσουμε μία εισαγωγή στην σύνταξη απλών προγραμμάτων (συναρτήσεις).

Έλεγχος ροής

Η γλώσσα MATLAB προσφέρει όλες τις δυνατότητες ελέγχου της εκτέλεσης (ροής) ενός προγράμματος και λήψης αποφάσεων, τις οποίες προσφέρουν οι μεγάλες (και παραδοσιακές) γλώσσες προγραμματισμού (C, C++, FORTRAN, BASIC κ.λ.π.). Αυτές είναι οι εξής:

if, else, και elseif

Η οδηγία ελέγχου *if* ελέγχει μια λογική έκφραση και εκτελεί μια ομάδα εντολών όταν η έκφραση είναι *αληθής* (*true*). Οι προαιρετικές ρήτρες *elseif* και *else* επιτρέπουν την εκτέλεση εναλλακτικών ομάδων εντολών. Μία οδηγία *end* που κλείνει ένα βρόχο *if*, τερματίζει την τελευταία ομάδα εντολών. Οι ομάδες εντολών ορίζονται από τις τέσσερις λέξεις κλειδιά – δεν απαιτούνται παρενθέσεις ή αγκύλες. Για παράδειγμα, ο αλγόριθμος κατασκευής ενός μαγικού τετραγώνου τάξης n περιλαμβάνει τρεις διαφορετικές περιπτώσεις: η n να είναι περιττή, η n να είναι άρτια αλλά μη διαιρέσιμη με 4 και, η n να είναι άρτια διαιρέσιμη με 4. Αυτό γράφεται ως εξής

```
if rem(n,2) ~= 0
    M = odd_magic(n)
elseif rem(n,4) ~= 0
    M = single_even_magic(n)
else
    M = double_even_magic(n)
end
```

Στο ανωτέρω παράδειγμα, οι τρεις περιπτώσεις είναι αμοιβαία αποκλειστικές, αλλά εάν δεν ήταν, θα εξετελείτο η *πρώτη αληθής* συνθήκη.

Είναι σημαντικό να γίνει κατανοητό, πώς οι σχεσιακοί τελεστές και οι έλεγχοι *if* δουλεύουν με πίνακες. Όταν θέλουμε να ελέγξουμε την ισότητα μεταξύ δύο μεταβλητών, μπορούμε να χρησιμοποιήσουμε

```
if A == B, ...
```

Η σύνταξη αυτή είναι σωστή και κάνει τα αναμενόμενα όταν οι A και B είναι μονόμετρα μεγέθη. Όταν όμως οι A και B είναι πίνακες, η συνθήκη $A == B$ δεν ελέγχει *εάν* είναι ίσοι, αλλά *που* είναι ίσοι – το αποτέλεσμα είναι ένας άλλος πίνακας αποτελούμενος από 0 (όχι) και 1 (ναι), ο οποίος φανερώνει την ύπαρξη ισότητας στοιχείο προς στοιχείο – εάν οι A και B δεν έχουν το ίδιο μέγεθος, η πράξη $A == B$ καταλήγει σε σφάλμα.

```
A = magic(4);
B = A;
```

```

B(1,1) = 0;
A == B
ans =
    0     1     1     1
    1     1     1     1
    1     1     1     1
    1     1     1     1

```

Ο κατάλληλος τρόπος να ελεγχθεί η ισότητα μεταξύ δύο μεταβλητών είναι να χρησιμοποιηθεί η συνάρτηση *isequal*,

```
if isequal(A, B), ...
```

Αυτή επιστρέφει μία *μονόμετρη* λογική τιμή 1 (αληθής) ή 0 (ψευδής) προς έλεγχο από την *if*, αντί για έναν πίνακα. Χρησιμοποιώντας τους πίνακες A και B του ανωτέρω παραδείγματος, λαμβάνομε

```
isequal(A, B)
ans =
    0

```

Ιδού και ένα άλλο παράδειγμα για να τονίσει αυτό το σημείο. Εάν οι A και B είναι μονόμετρα, το ακόλουθο πρόγραμμα δεν θα φθάσει ποτέ στην «απρόσμενη περίπτωση». Αλλά για τα περισσότερα ζεύγη πινάκων, περιλαμβανομένων και των μαγικών τετραγώνων με ανταλλαγμένες στήλες, καμία από τις συνθήκες $A > B$, $A < B$, ή $A == B$ δεν είναι αληθής για όλα τα στοιχεία και έτσι εκτελείται η ρήτρα *else*:

```

if A > B
    'greater'
elseif A < B
    'less'
elseif A == B
    'equal'
else
    error('Unexpected situation')
end

```

Υπάρχουν αρκετές συναρτήσεις που βοηθούν στην αναγωγή συγκρίσεων μεταξύ πινάκων σε μονόμετρες συνθήκες για χρήση με την *if*, οι κυριότερες των οποίων είναι οι *isequal*, *isempty*, *all* και *any*.

switch και case

Η οδηγία ελέγχου *switch* εκτελεί ομάδες εντολών βασιζόμενη στην τιμή μιας μεταβλητής ή μιας έκφρασης. Οι ρήτρες *case* και *otherwise* ορίζουν τις ομάδες. Μόνο η πρώτη αληθής περίπτωση εκτελείται. Ο βρόχος *switch* πρέπει πάντοτε να κλείνει με *end*. Χρησιμοποιώντας έλεγχο *switch*, η λογική ροή του αλγορίθμου μαγικών τετραγώνων μπορεί να γραφεί ως εξής:

```

switch (rem(n,4) == 0) + (rem(n,2) == 0)
case 0
    M = odd_magic(n)
case 1
    M = single_even_magic(n)
case 2
    M = double_even_magic(n)
otherwise
    error('This is impossible')
end

```

for

Ο βρόχος *for ... end* εκτελεί προκαθορισμένο αριθμό επαναλήψεων μίας ομάδα εντολών. Κάθε οδηγία *for* πρέπει να τερματίζεται με μία αντιστοιχούσα οδηγία *end*:

```

for n = 3:32
    r(n) = rank(magic(n));
end
r

```

Το ερωτηματικό που τερματίζει την εσωτερική εντολή καταστέλλει την εμφάνιση του αποτελέσματος

$r(n)$ κάθε φορά που εκτελείται η συνάρτηση `rank` και η εντολή `r` μετά το τέλος του βρόχου εμφανίζει το τελικό αποτέλεσμα. Το επόμενο παράδειγμα, δείχνει δύο βρόχους `for ... end` να εκτελούν μία αντικατάσταση στα στοιχεία του πίνακα $H(i,j)$. Ο εσωτερικός (ένθετος) βρόχος (μεταβαλλόμενος κατά j υπεράνω των γραμμών του H), είναι ελαφρά μετατοπισμένος σε σχέση με το εξωτερικό, για να βελτιωθεί η αναγνωσιμότητα. Ο τρόπος αυτός γραφής πολλαπλών εσώκλειστων βρόχων, γενικά θεωρείται καλή ιδέα:

```
for i = 1:m
    for j = 1:n
        H(i,j) = 1/(i+j);
    end
end
```

while

Ο βρόχος `while ... end` εκτελεί απροσδιόριστο αριθμό επαναλήψεων μίας ομάδας εντολών, έως ότου ικανοποιηθεί μία λογική συνθήκη. Κάθε οδηγία `while` πρέπει να τερματίζεται με μία αντιστοιχούσα οδηγία `end`. Ακολουθεί ένα πλήρες πρόγραμμα που επιδεικνύει την χρήση των `while`, `if`, `else`, και `end`, και υλοποιεί την μέθοδο της διχοτόμησης προκειμένου να εντοπίσει την ρίζα ενός πολυωνύμου:

```
a = 0;
fa = -Inf;
b = 3;
fb = Inf;
while b-a > eps*b
    x = (a+b)/2;
    fx = x^3-2*x-5;
    if sign(fx) == sign(fa)
        a = x; fa = fx;
    else
        b = x; fb = fx;
    end
end
```

Αποτέλεσμα είναι η ρίζα του πολυωνύμου $x^3 - 2x - 5$, δηλαδή η

```
x =
    2.09455148154233
```

continue

Η οδηγία `continue` μεταφέρει τον έλεγχο στην επόμενη επανάληψη ενός βρόχου `for ... end` ή `while ... end`, εντός του οποίου δίδεται, παραλείποντας τις ενδιάμεσες εντολές. Σε περίπτωση πολλαπλών ένθετων βρόχων, η `continue` μεταφέρει τον έλεγχο στον αμέσως ανώτερο (περιβάλλοντα αυτή) βρόχο. Το επόμενο παράδειγμα δείχνει πως η `continue` βοηθά στο να μετρηθούν οι πραγματικές γραμμές κώδικα στο αρχείο `magic.m`, παραλείποντας όλες τις κενές γραμμές και τα σχόλια: η οδηγία υποχρεώνει το πρόγραμμα να προχωρήσει στην επόμενη γραμμή του `magic.m` χωρίς να αυξάνει τον μετρητή `count`, κάθε φορά που συναντάται μία γραμμή χωρίς περιεχόμενο (η `isempty` αποβαίνει αληθής) ή μία γραμμή αρχίζει από `%` (η `strcmp` αποβαίνει αληθής):

```
fid = fopen('magic.m','r');
count = 0;
while ~feof(fid)
    line = fgetl(fid);
    if isempty(line) | strcmp(line,'% ',1)
        continue
    end
    count = count + 1;
end
disp(sprintf('%d lines',count));
```

break

Η οδηγία *break* επιτρέπει την πρόωμη έξοδο από ένα βρόχο `for` ή `while`, εφόσον ικανοποιούνται δεδομένες συνθήκες. Σε περίπτωση ενθέτων βρόχων, η `break` εξάγει από τον εσωτερικότερο βρόχο εντός του οποίου εμφανίζεται. Το ακόλουθο παράδειγμα προσφέρει μία βελτίωση του κώδικα υπολογισμού της πολυωνυμικής ρίζας που συναντήσαμε νωρίτερα. Η βελτίωση συνίσταται στο γεγονός ότι μόλις βρεθεί η ρίζα, ο βρόχος `while ... end` τερματίζεται και δεν συνεχίζει να ψάχνει μέχρις εξαντλήσεως της ακρίβειας του συστήματος (`eps`).

```
a = 0;
fa = -Inf;
b = 3;
fb = Inf;
while b-a > eps*b
    x = (a+b)/2;
    fx = x^3-2*x-5;
    if fx == 0
        break
    elseif sign(fx) == sign(fa)
        a = x; fa = fx;
    else
        b = x; fb = fx;
    end
end
```

try - catch

Η γενική μορφή μίας σειράς `try-catch` είναι

```
try
    statement
    ...
    statement
catch
    statement
    ...
    statement
end
```

Στην ακολουθία αυτή οι εντολές μεταξύ των `try` και `catch` εκτελούνται έως ότου ανακύψει κάποιο σφάλμα. Αμέσως μετά εκτελούνται οι εντολές μεταξύ `catch` και `end`. Η σειρά αυτή χρησιμοποιείται για να εντοπίζει σφάλματα και να τα αποτρέπει από του να «ρίξουν» το σύστημα. Η συνάρτηση *lasterr* μπορεί να χρησιμοποιηθεί για εντοπισμό της αιτίας σφάλματος. Εάν υπάρξει σφάλμα μεταξύ των εντολών `catch` και `end`, η εκτέλεση του προγράμματος τερματίζεται εκτός και εάν αυτό συμβεί εντός μίας άλλης σειράς `try-catch`.

return

Η *return* τερματίζει την τρέχουσα ακολουθία εντολών και επιστρέφει τον έλεγχο στην συνάρτηση ή στο πληκτρολόγιο. Η καλούμενη συνάρτηση κανονικά μεταφέρει τον έλεγχο στην καλούσα συνάρτηση μόλις ολοκληρωθεί η εκτέλεσή της, αλλά η εντολή `return` μπορεί να εισαχθεί οπουδήποτε στην καλούμενη συνάρτηση για να επιφέρει πρόωμο τερματισμό και μεταφορά του ελέγχου, εφόσον απαιτείται από τις συνθήκες.

Προγράμματα

Τα συστήματα MATLAB και OCTAVE, εκτός από ένα διαλογικό υπολογιστικό περιβάλλον, προσφέρουν και μία ισχυρή γλώσσα προγραμματισμού υψηλού επιπέδου (γλώσσα MATLAB). Τα αρχεία που περιέχουν κώδικα σε γλώσσα MATLAB ονομάζονται *M-files* (*φάκελοι-M*). Οι M-files μπορούν να δημιουργηθούν και μέσω ενός απλού συντάκτη κειμένου και χρησιμοποιούνται όπως κάθε άλλη συνάρτηση ή εντολή. Υπάρχουν δύο είδη M-files:

- **Scripts**, τα οποία δεν δέχονται ορίσματα εισόδου και δεν επιστρέφουν ορίσματα εξόδου, λειτουργούν δε επί των δεδομένων και μεταβλητών του χώρου εργασίας.
- **Συναρτήσεις**, οι οποίες μπορούν να δεχθούν και να επιστρέψουν ορίσματα. Οι μεταβλητές που ορίζονται εντός των συναρτήσεων είναι προσωρινές και διαγράφονται μόλις η συνάρτηση επιστρέψει τον έλεγχο.

Για να δούμε το περιεχόμενο ενός M-file, παραδείγματος χάριν, του myfunction.m, χρησιμοποιούμε την εντολή type:

```
type myfunction
```

Scripts

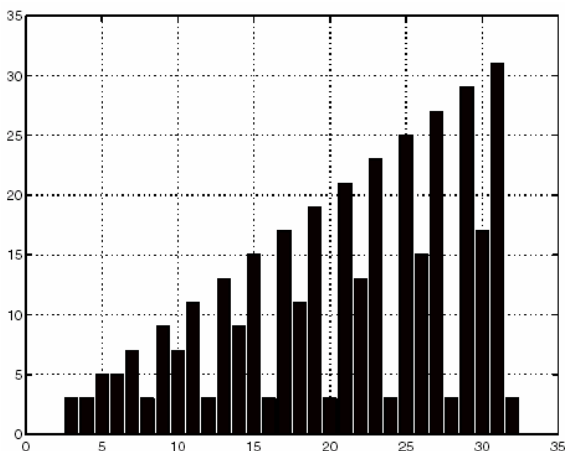
Όταν καλείται ένα *script*, το σύστημα απλά εκτελεί τις εντολές που βρίσκονται στο αρχείο. Τα scripts λειτουργούν μόνο επί των υπάρχοντων δεδομένων στον χώρο εργασίας, ή μπορούν να δημιουργήσουν νέα δεδομένα επί των οποίων θα επενεργήσουν. Αν και τα scripts δεν επιστρέφουν ορίσματα εξόδου, όποιες μεταβλητές δημιουργήσουν ή επεξεργασθούν παραμένουν στον χώρο εργασίας, για να χρησιμοποιηθούν σε επόμενους υπολογισμούς. Επιπλέον, τα scripts μπορούν να παράγουν γραφικά, χρησιμοποιώντας συναρτήσεις όπως η plot. Για παράδειγμα, ας δημιουργήσουμε ένα φάκελο με το όνομα magicrank.m, που να περιέχει τις ακόλουθες εντολές:

```
% Investigate the rank of magic squares r = zeros(1,32);  
for n = 3:32  
    r(n) = rank(magic(n));  
end r  
bar(r)
```

Δίδοντας

```
magicrank,
```

το σύστημα εκτελεί τις εντολές και υπολογίζει τον βαθμό των πρώτων 30 μαγικών τετραγώνων, στο τέλος δε χαράσσει ένα ιστόγραμμα του αποτελέσματος. Μετά την ολοκλήρωση των εργασιών, οι μεταβλητές n και r παραμένουν στον χώρο εργασίας.



Συναρτήσεις (functions)

Οι συναρτήσεις είναι M-files που δέχονται ορίσματα εισόδου και επιστρέφουν ορίσματα εξόδου. Το όνομα του M-file και της συνάρτησης που περιέχει οφείλουν να συμπίπτουν. Οι συναρτήσεις κατασκευάζουν τον δικό τους *προσωρινό* χώρο εργασίας, ξεχωριστό από τον χώρο εργασίας στον οποίο έχουμε πρόσβαση μέσω του παραθύρου εντολών, και επιδρούν επί των μεταβλητών που εισάγονται ή δημιουργούνται εκεί. Για παράδειγμα, μπορούμε να δούμε την συνάρτηση rank.m (διαθέσιμη ως M-file) δίδοντας

```
type rank
και λαμβάνομε:
function r = rank(A,tol)
% RANK Matrix rank.
% RANK(A) provides an estimate of the number of linearly
% independent rows or columns of a matrix A.
% RANK(A,tol) is the number of singular values of A
% that are larger than tol.
% RANK(A) uses the default tol = max(size(A)) * norm(A) * eps.

s = svd(A);
if nargin==1
    tol = max(size(A)') * max(s) * eps;
end
r = sum(s > tol);
```

Η πρώτη γραμμή μίας συνάρτησης M-file αρχίζει με την λέξη-κλειδί *function*, η οποία δηλώνει το όνομα της συνάρτησης και την σειρά των ορισμάτων. Στην συγκεκριμένη περίπτωση έχουμε δύο ορίσματα εισόδου και ένα εξόδου. Η πρώτη γραμμή του κειμένου βοήθειας είναι η γραμμή H1, η οποία εμφανίζεται όταν χρησιμοποιείται η εντολή *lookfor* ή ζητείται βοήθεια σε ένα αρχειοκατάλογο. Οι επόμενες αρκετές γραμμές, μέχρι την πρώτη κενή ή εκτελέσιμη γραμμή, είναι σχόλια που προσφέρουν on line βοήθεια. Οι γραμμές αυτές εκτυπώνονται στην οθόνη όταν εντελλόμαστε

```
help rank
```

Το υπόλοιπο του αρχείου είναι ο εκτελέσιμος κώδικας της συνάρτησης. Η μεταβλητή *s*, η οποία δημιουργείται κατά την εκτέλεση του κώδικα, καθώς και οι *r*, *A* και *tol*, όλες είναι *τοπικές (local)* μεταβλητές της συνάρτησης και ξεχωριστές από τις μεταβλητές του χώρου εργασίας. Οι *A* και *tol* (ορίσματα εισόδου) δημιουργούνται ως αντίγραφα των μεταβλητών του χώρου εργασίας και διαγράφονται μόλις η συνάρτηση επιστρέψει τον έλεγχο. Αυτό το παράδειγμα δείχνει και μια πτυχή των συναρτήσεων της γλώσσας MATLAB, οι οποίες συνήθως δεν απαντώνται σε άλλες γλώσσες προγραμματισμού – μεταβλητό αριθμό ορισμάτων. Η συνάρτηση rank μπορεί να χρησιμοποιηθεί με τρεις διαφορετικούς τρόπους:

```
rank(A)
r = rank(A)
r = rank(A,1.e-6)
```

Εάν δεν δοθεί όρισμα εξόδου, το αποτέλεσμα καταχωρείται στην αυτόματη μεταβλητή *ans*. Εάν δεν δοθεί το δεύτερο όρισμα εισόδου, η συνάρτηση χρησιμοποιεί την αυτόματη τιμή *eps*. Μέσα στο σώμα της συνάρτησης υπάρχουν δύο ποσότητες που ονομάζονται *nargin* και *nargout* και καταμετρούν τον αριθμό των ορισμάτων εισόδου και εξόδου που χρησιμοποιούνται σε κάθε ανεξάρτητη κλήση της συνάρτησης. Η συνάρτηση rank χρησιμοποιεί την *nargin*, αλλά δεν χρειάζεται την *nargout*.

Τύποι Συναρτήσεων

Προσφέρονται αρκετοί διαφορετικοί τύποι συναρτήσεων προς χρήση από τον προγραμματιστή.

Ανώνυμες Συναρτήσεις: *Ανώνυμη* είναι μια απλή μορφή συνάρτησης η οποία δεν απαιτεί ύπαρξη αντιστοίχου M-file. Απαρτίζεται από μία μόνη έκφραση και τον απαιτούμενο αριθμό ορισμάτων εισόδου και εξόδου. Μία ανώνυμη συνάρτηση μπορεί να ορισθεί απ' ευθείας στο παράθυρο εντολών, ή εντός ενός M-file ή script. Η δυνατότητα αυτή προσφέρει ένα ταχύ τρόπο κατασκευής απλών υπολογιστικών

συναρτήσεων (που συνήθως καλούνται πολλές φορές), χωρίς να απαιτείται ξεχωριστός M-file κάθε φορά. Η σύνταξη για τη δημιουργία μίας ανώνυμης συνάρτησης από μία έκφραση είναι:

```
f = @(arglist) expression
```

Η κατωτέρω εντολή δημιουργεί μία ανώνυμη συνάρτηση που υπολογίζει το τετράγωνο ενός αριθμού. Όταν καλείται η συνάρτηση, το σύστημα αποδίδει την τιμή του ορίσματος εισόδου στην μεταβλητή *x*, και μετά χρησιμοποιεί την *x* στην εξίσωση $x.^2$:

```
sqr = @(x) x.^2;
```

Για να εκτελεσθεί η *sqr* όπως ορίζεται ανωτέρω, δίδουμε

```
a = sqr(5)
```

```
a =
```

```
25
```

Πρωτεύουσες (Primary) και Υποσυναρτήσεις (Subfunctions): Όλες οι μη ανώνυμες συναρτήσεις πρέπει να ορίζονται μέσα σε έναν M-file. Κάθε M-file περιέχει έχει την απαιτούμενη *πρωτεύουσα συνάρτηση*, η οποία εμφανίζεται πρώτη στον φάκελο, η οποία μπορεί να ακολουθείται από οιονδήποτε αριθμό *υποσυναρτήσεων*. Οι πρωτεύουσες συναρτήσεις μπορούν να καλούνται από οπουδήποτε είναι χρήσιμο, δηλαδή από το παράθυρο εργασίας ή από άλλες συναρτήσεις μέσα σε άλλους M-files, ενώ οι υποσυναρτήσεις όχι. Οι υποσυναρτήσεις είναι ορατές μόνο στις πρωτεύουσες συναρτήσεις και άλλες υποσυναρτήσεις εντός του *ιδίου* M-file. Η συνάρτηση *rank* ανωτέρω είναι παράδειγμα πρωτεύουσας συνάρτησης.

Ιδιωτικές (Private) Συναρτήσεις: Πρόκειται για ειδικό τύπο πρωτεύουσας συνάρτησης, της οποίας διακριτικό χαρακτηριστικό είναι ότι μπορεί να κληθεί μόνο από μία περιορισμένη ομάδα άλλων συναρτήσεων. Συναρτήσεις αυτού του είδους είναι χρήσιμες όταν πρέπει επιθυμείται περιορισμένη πρόσβαση, ή και απόκρυψη του τρόπου χρήσης της. Οι ιδιωτικές συναρτήσεις οφείλουν να τοποθετούνται σε υποκαταλόγους (subdirectories) με το *ειδικό* όνομα *private* και μπορούν να κληθούν μόνο από συναρτήσεις του μητρικού καταλόγου. Επειδή οι ιδιωτικές συναρτήσεις είναι *αόρατες* έξω από τον μητρικό τους κατάλογο, μπορούν να έχουν τα ίδια ονόματα με συναρτήσεις σε άλλους καταλόγους. Αυτό είναι αρκετά χρήσιμο όταν κανείς θέλει να δημιουργήσει την δική του έκδοση μίας κάποιας συνάρτησης, διατηρώντας την αρχική έκδοση σε έναν άλλο κατάλογο. Επειδή το σύστημα ψάχνει πρώτα για ιδιωτικές συναρτήσεις και μετά για τυπικούς M-files, θα εντοπίσει μία ιδιωτική συνάρτηση ονομαζόμενη *test.m* πριν από μία κοινή συνάρτηση M-file που επίσης ονομάζεται *test.m*.

Ένθετες (Nested) Συναρτήσεις: Αυτές μπορούν να ορισθούν εντός του κυρίου σώματος οιασδήποτε συνάρτησης σε ένα M-file. Μία ένθετη συνάρτηση μπορεί να περιλαμβάνει όλες τις συνιστώσες και χαρακτηριστικά των συναρτήσεων τύπου M-file. Στο ακόλουθο παράδειγμα, η συνάρτηση *B* είναι ένθετη στην συνάρτηση *A*:

```
function x = A(p1, p2)
```

```
... B(p2)
```

```
    function y = B(p3)
```

```
    ...
```

```
    end
```

```
end
```

Όπως και οι άλλες συναρτήσεις, μία ένθετη δημιουργεί τον δικό της χώρο εργασίας όπου καταχωρούνται οι μεταβλητές που χρησιμοποιεί. Επιπροσθέτως, έχει πρόσβαση στον χώρο εργασίας της συνάρτησης εντός της οποίας εντίθεται! Έτσι, μία που ορίζεται εντός μίας πρωτεύουσας συνάρτησης, μπορεί να ανακληθεί ή και να τροποποιηθεί από μία ένθετη συνάρτηση. Ομοίως, μια μεταβλητή που ορίζεται στην ένθετη συνάρτηση μπορεί να ανακληθεί ή και να τροποποιηθεί από οποιαδήποτε συνάρτηση περιβάλλει την ένθετη.

Κοινές (Global) Μεταβλητές

Μία μεταβλητή μπορεί να χρησιμοποιείται από κοινού σε περισσότερες από μία συναρτήσεις, εάν δηλωθεί ως *global* (κοινή) σε όλες τις συναρτήσεις. Με την αυτή δήλωση μίας μεταβλητής στο παράθυρο εντολών, ο βασικός χώρος εργασίας μπορεί να την προσπελάσει και μέσα σε οιαδήποτε συνάρτηση. Η δήλωση *global* πρέπει να εμφανιστεί πριν η μεταβλητή χρησιμοποιηθεί για πρώτη φορά. Αν και δεν απαιτείται, η χρήση κεφαλαίων γραμμάτων για τα ονόματα των *global* μεταβλητών βοηθά στην διάκρισή τους από τις άλλες. Για παράδειγμα, δημιουργούμε έναν M-file με το όνομα *falling.m*:

```
function h = falling(t)
global GRAVITY
h = 1/2*GRAVITY*t.^2;
```

Μετά ,στο παράθυρο εργασίας δηλώνομε και καλούμε:

```
global GRAVITY
GRAVITY = 32;
y = falling((0:.1:5)');
```

Οι δύο δηλώσεις global καθιστούν την τιμή που δίδεται από το παράθυρο εντολών στην μεταβλητή GRAVITY προσβάσιμη και από την συνάρτηση falling. Η GRAVITY μπορεί να τροποποιηθεί οποτεδήποτε και οι υπολογισμοί να επαναληφθούν χωρίς να χρειαστεί η αναθεώρηση του M-file.

Εισαγωγή αλφαριθμητικών (string) ορισμάτων σε συναρτήσεις

Οι κλήσεις συναρτήσεων που δέχονται μόνο αλφαριθμητικά ορίσματα μπορούν να γίνουν χωρίς παρενθέσεις και αποστρόφους. Δηλαδή, το σύστημα ερμηνεύει την

```
foo a b c
```

ως

```
foo('a','b','c')
```

Σημειωτέον όμως, ότι όταν χρησιμοποιείται η ελεύθερη (χωρίς αποστρόφους) μορφή, το σύστημα αδυνατεί να επιστρέψει ορίσματα εξόδου. Για παράδειγμα, η

```
legend apples oranges
```

δημιουργεί ένα υπόμνημα σε ένα γράφημα, χρησιμοποιώντας τις αλφαριθμητικές παραμέτρους *apples* και *oranges* ως ετικέτες καμπυλών. Εάν όμως είναι επιθυμητή η επιστροφή των ορισμάτων εξόδου της συνάρτησης, τότε πρέπει να χρησιμοποιηθεί η μορφή:

```
[legh,objh] = legend('apples','oranges');
```

Κατασκευή αλφαριθμητικών ορισμάτων με κώδικα: Το ακόλουθο παράδειγμα επεξεργάζεται πολλαπλά αρχεία δεδομένων, August1.dat, August2.dat, κ.ο.κ., χρησιμοποιώντας την συνάρτηση *int2str* για να μετατρέψει ακεραίους αριθμούς σε αλφαριθμητικές σειρές προκειμένου να κατασκευάσει ονόματα αρχείων:

```
for d = 1:31
    s = ['August' int2str(d) '.dat'];
    load(s)
    % Κώδικας επεξεργασίας του d-στού αρχείου
end
```

Προσοχή: Σε ορισμένες περιπτώσεις, η άνευ αποστρόφων σύνταξη αλφαριθμητικών σειρών μπορεί χρησιμοποιηθεί λανθασμένα, χωρίς να προκαλέσει σφάλματα. Για παράδειγμα, δεδομένου ενός πίνακα A,

```
A =
     0     -6     -1
     6      2    -16
    -5     20    -10
```

Η συνάρτηση *eig* επιστρέφει τις ιδιοτιμές του A:

```
eig(A)
ans =
    -3.0710
   -2.4645+17.6008i
   -2.4645-17.6008i
```

Η ακόλουθη εντολή είναι λανθασμένη διότι η A δεν είναι αλφαριθμητική μεταβλητή.

```
eig A
ans =
    65
```

Δεν προκαλείται σφάλμα, διότι σφάλμα ερμηνεύει ως όρισμα εισόδου το ASCII αριθμητικό ισοδύναμο του γράμματος A (το οποίο είναι 65) και υπολογίζει την ιδιοτιμή του που (επειδή πρόκειται για μονόμετρο μέγεθος), είναι απαραίτητως 65.

Η συνάρτηση eval

Η συνάρτηση eval εργάζεται με αλφαριθμητικές μεταβλητές και αποτελεί μία πολύ ισχυρή μηχανή εκτέλεσης macro-εντολών κειμένου. Η έκφραση

```
eval(s)
```

χρησιμοποιεί τον διερμηνέα του συστήματος για να αξιολογήσει την έκφραση, ή να εκτελέσει την εντολή που περιέχεται στην αλφαριθμητική σειρά *s*. Το προηγούμενο παράδειγμα επεξεργασίας πολλαπλών αρχείων μπορεί επίσης να υλοποιηθεί με τον ακόλουθο κώδικα, (ο οποίος όμως είναι λιγότερο αποδοτικός διότι χρησιμοποιεί ολόκληρο τον διερμηνέα του συστήματος και όχι μόνο μία συνάρτηση):

```
for d = 1:31
    s = ['load August' int2str(d) '.dat'];
    eval(s)
    % Κώδικας επεξεργασίας του d-στού αρχείου
end
```

Χειριστήρια (handles) συναρτήσεων

Σε ορισμένες εφαρμογές, ένα χειριστήριο συνάρτησης χρησιμεύει ως αποτελεσματικότερο μέσο κλήσης της συνάρτησης. Το χειριστήριο συνήθως δίδεται ως όρισμα σε κάποια άλλη συνάρτηση, η οποία μπορεί μετά να εκτελέσει, ή να αξιολογήσει την συνάρτηση που το χρησιμοποιεί. Ένα χειριστήριο συναρτήσεων κατασκευάζεται προσθέτοντας το σύμβολο *at*, *@*, πριν από το όνομα της συνάρτησης. Το ακόλουθο παράδειγμα δημιουργεί ένα χειριστήριο της συνάρτησης *sin* και το αποδίδει στην μεταβλητή *fhandle*:

```
fhandle = @sin;
```

Μία συνάρτηση μπορεί να εκτελεσθεί χρησιμοποιώντας το χειριστήριό της, κατά τον ίδιο τρόπο που χρησιμοποιείται το όνομά της. Η σύνταξη είναι

```
fhandle(arg1, arg2, ...);
```

Η συνάρτηση *plot_fhandle*, κατωτέρω, δέχεται ένα χειριστήριο συνάρτησης και δεδομένα τετμημένης, κατασκευάζει τα δεδομένα τεταγμένης χρησιμοποιώντας το χειριστήριο της συνάρτησης και τα απεικονίζει σε γράφημα:

```
function x = plot_fhandle(fhandle, data)
plot(data, fhandle(data))
```

Όταν κληθεί η *plot_fhandle* ως κάτωθι, με ένα χειριστήριο για την συνάρτηση *sin* και τα δεδομένα τετμημένης, παράγεται το γράφημα ενός ημιτονοειδούς:

```
plot_fhandle(@sin, -pi:0.01:pi)
```

Ανυσματοποίηση (Vectorization)

Τα προγράμματα σε γλώσσα MATLAB εκτελούνται *πολύ πολύ* ταχύτερα, εάν υπάρχει δυνατότητα *ανυσματοποίησης* των χρησιμοποιούμενων αλγορίθμων, διότι στην περίπτωση αυτή οι πράξεις γίνονται ταυτόχρονα για όλα τα στοιχεία των ανυσμάτων ή πινάκων που περιέχουν αριθμητικά δεδομένα. Ένα απλό παράδειγμα «κλασσικής» κατασκευής λογαριθμικών πινάκων είναι το εξής:

```
x = .01;
for k = 1:1001
    y(k) = log10(x);
    x = x + .01;
end
```

Η ανυσματοποιημένη έκδοση του αυτού κώδικα είναι:

```
x = .01:.01:10;
y = log10(x);
```

Για πιο περίπλοκο κώδικα όμως, οι μέθοδοι ανυσματοποίησης μπορεί να μην είναι τόσο προφανείς.

Προκαθορισμός διαστάσεων (Preallocation)

Εάν ο κώδικας δεν μπορεί να ανυσματοποιηθεί, η εκτέλεση επιταχύνεται εάν είναι δυνατός ο προκαθορισμός των διαστάσεων των ανυσμάτων και πινάκων όπου θα καταχωρηθούν τα αποτελέσματα των υ-

πολογισμών. Στο ακόλουθο παράδειγμα, ο κώδικας χρησιμοποιεί την συνάρτηση `zeros` για να προκαθορίσει το άνυσμα που δημιουργείται εντός του βρόχου `for ... end`. Τότε, ο βρόχος εκτελείται πολύ ταχύτερα:

```
r = zeros(32,1);
for n = 1:32
    r(n) = rank(magic(n));
end
```

Στην αντίθετη περίπτωση το άνυσμα `r` θα έπρεπε να επαναπροσδιορίζεται σε κάθε επανάληψη, επεκτεινόμενο κατά ένα στοιχείο – για μεγάλα ανύσματα και πίνακες αυτό απαιτεί αρκετή υπολογιστική εργασία. Με τον προκαθορισμό των διαστάσεών του το βήμα αυτό εξαλείφεται και η εκτέλεση επιταχύνεται.

ΑΛΛΕΣ ΜΟΡΦΕΣ ΚΑΙ ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Πολυδιάστατοι πίνακες

Οι πολυδιάστατοι πίνακες έχουν περισσότερους από δύο δείκτες. Ένας τρόπος δημιουργίας πολυδιάστατων πινάκων είναι με κλήση των `zeros`, `ones`, `rand`, ή `randn` με περισσότερα των δύο ορίσματα. Για παράδειγμα, η

```
R = randn(3,4,5);
```

δημιουργεί ένα πίνακα $3 \times 4 \times 5$ με συνολικά 60, κανονικά κατανομημένα στοιχεία. Ένας τριδιάστατος πίνακας μπορεί να αντιπροσωπεύει τριδιάστατα φυσικά στοιχεία, για παράδειγμα την κατανομή της ειδικής αγωγιμότητας στο υπέδαφος δειγματολογισμένη επί ορθο-κανονικού καννάβου, ή απλά μία σειρά διδιάστατων πινάκων, $A(k)$, ή δειγμάτων ενός χρονο-εξαρτώμενου πίνακα, $A(t)$. Το (i, j) -στο στοιχείο του k -στού ή του t -στού πίνακα δηλώνεται με $A(i, j, k)$. Για παράδειγμα, τα MATLAB/OCTAVE και ο Dürer παρέχουν δύο εκδόσεις του μαγικού τετραγώνου 4^{th} τάξης, οι οποίες 4 διαφέρουν κατά την αντιμετάθεση δύο στηλών. Πολλά διαφορετικά μαγικά τετράγωνα μπορούν να παραχθούν με αντιμετάθεση στηλών. Η εντολή

```
p = perms(1:4);
```

παράγει τις $4! = 24$ αντιμεταθέσεις των στοιχείων του ανύσματος 1:4. Η k -στη αντιμετάθεση είναι το οριζόντιο άνυσμα $p(k, :)$. Τότε, οι εντολές

```
A = magic(4);
```

```
M = zeros(4,4,24);
```

```
for k = 1:24
```

```
    M(:, :, k) = A(:, p(k, :));
```

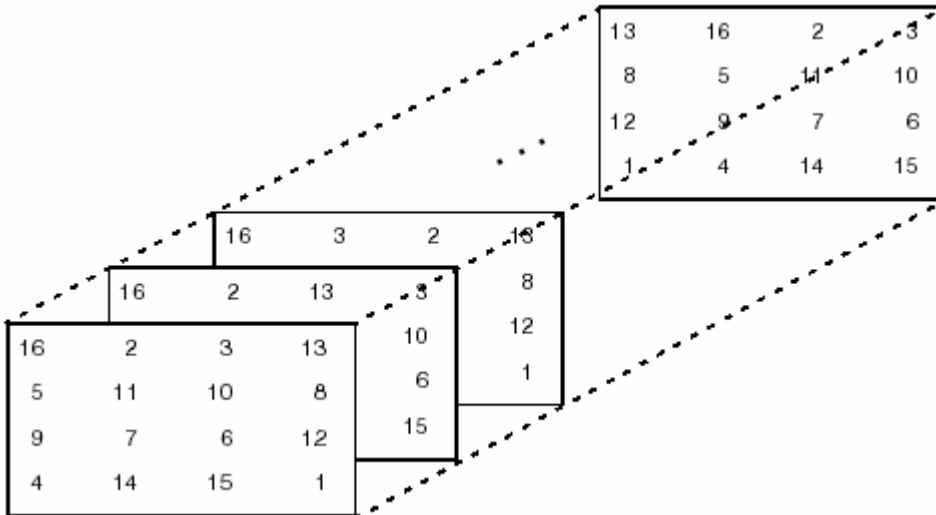
```
end
```

καταχωρούν την ακολουθία των 24 μαγικών τετραγώνων σε ένα τριδιάστατο πίνακα (διάταξη) `M`. Το μέγεθος του `M` είναι

```
size(M)
```

```
ans =
```

```
    4    4   24
```



Η εντολή

```
sum(M, d)
```

υπολογίζει αθροίσματα μεταβάλλοντας τον d-στο δείκτη. Έτσι, η

```
sum(M, 1)
```

επιστέφει ένα πίνακα διαστάσεων $1 \times 4 \times 24$, ο οποίος περιέχει 24 αντίγραφα του οριζόντιου ανύσματος

```
[34 34 34 34]
```

και η

```
sum(M, 2)
```

επιστέφει έναν πίνακα $4 \times 1 \times 24$ array που περιέχει 24 αντίγραφα του κατακόρυφου ανύσματος

```
[34
 34
 34
 34]
```

Τέλος, η ,

```
S = sum(M, 3)
```

προσθέτει τους 24 πίνακες της ακολουθίας. Το αποτέλεσμα έχει μέγεθος $4 \times 4 \times 1$, και έτσι φαίνεται ως πίνακας 4×4 :

```
S =
 204 204 204 204
 204 204 204 204
 204 204 204 204
 204 204 204 204
```

Κυψελοπίνακες (Cell Arrays)

Οι *κυψελοπίνακες* είναι πολυδιάστατοι πίνακες των οποίων στοιχεία είναι αντίγραφα άλλων πινάκων. Ένας κυψελοπίνακας κενών στοιχείων δημιουργείται με την συνάρτηση *cell*. Συχνότερα όμως, οι κυψελοπίνακες δημιουργούνται εγκλείοντας μία συλλογή διαφορετικών μεταβλητών σε άγκιστρα, `{}`. Τα άγκιστρα επίσης χρησιμοποιούνται με δείκτες, προκειμένου να ανακληθεί το περιεχόμενο των διαφόρων κυψελίδων. Για παράδειγμα, η

```
C = {A sum(A) prod(prod(A))}
```

παράγει ένα κυψελοπίνακα 1×3 . Οι τρεις κυψελίδες περιέχουν το μαγικό τετράγωνο, το οριζόντιο άνυσμα των αθροισμάτων των στηλών και το γινόμενο όλων των στοιχείων του. Όταν ζητείται η εμφάνιση του C, το σύστημα επιστρέφει:

```
C =
 [4x4 double] [1x4 double] [20922789888000]
```

διότι οι πρώτες δύο κυψελίδες είναι πολύ μεγάλες ώστε να παρουσιαστούν σε μία μόνο γραμμή, ενώ η τρίτη περιέχει έναν μόνο αριθμό, $16!$, ώστε να υπάρχει χώρος να τον δείξει. Υπάρχουν δύο σπουδαία σημεία που δεν πρέπει ποτέ να λησμονούνται. Πρώτον, για να ανακληθεί το περιεχόμενο μίας κυψελίδας

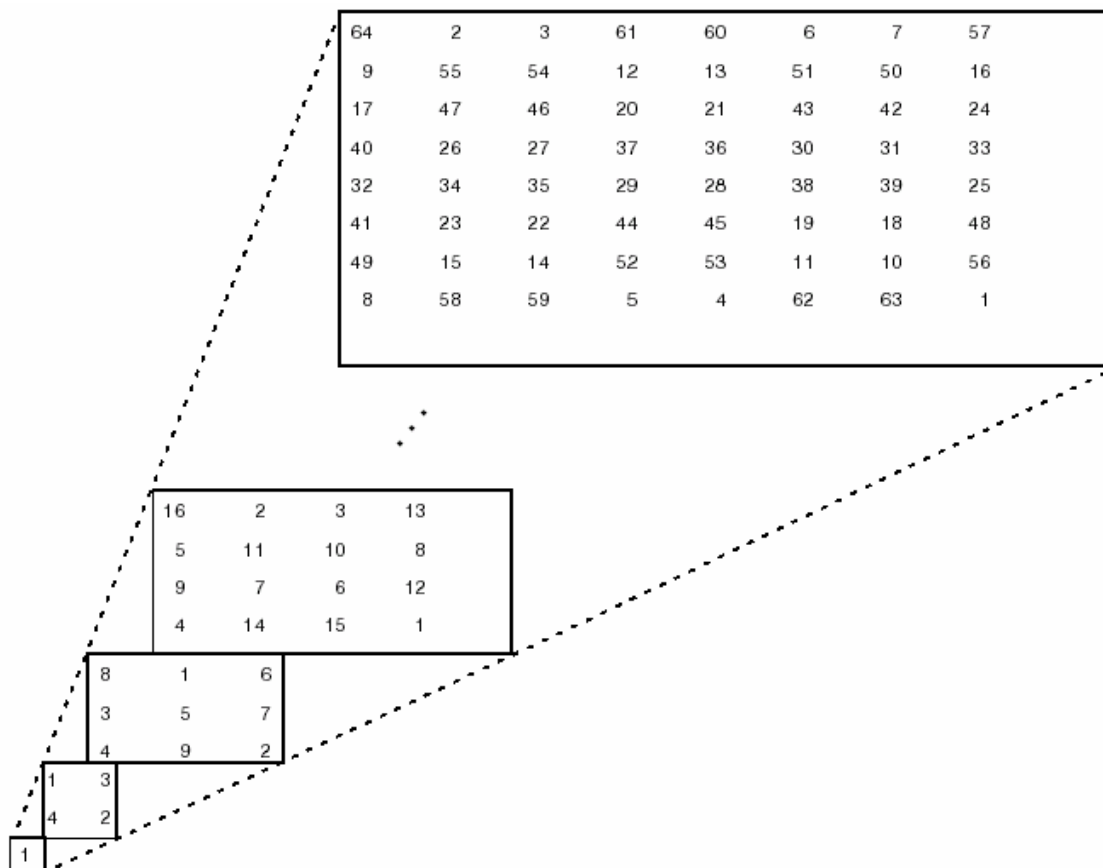
δας, πρέπει να χρησιμοποιηθούν δείκτες εντός αγκίστρων. Για παράδειγμα, η C{1} ανακαλεί το μαγικό τετράγωνο και η C{3} είναι 16! Δεύτερον, οι κυψελοπίνακες περιέχουν αντίγραφα άλλων πινάκων και όχι δείκτες (pointers) προς αυτούς τους πίνακες. Έτσι, αν στο ανωτέρω παράδειγμα ο A αλλάξει μετά την δημιουργία του C, τότε ο C παραμένει αμετάβλητος. Επίσης, οι τριδιάστατοι πίνακες που συζητήθηκαν ανωτέρω μπορούν να χρησιμοποιηθούν για την καταχώρηση διδιάστατων πινάκων του αυτού μεγέθους. Οι κυψελοπίνακες αποτελούν μία γενίκευση: μπορούν να καταχωρήσουν πίνακες διαφορετικών μεγεθών. Για παράδειγμα, οι εντολές

```
M = cell(8,1);
for n = 1:8
    M{n} = magic(n);
end
M
```

παράγουν μια ακολουθία μαγικών τετραγώνων διαφορετικής τάξης:

```
M =
[
    1]
[ 2x2 double]
[ 3x3 double]
[ 4x4 double]
[ 5x5 double]
[ 6x6 double]
[ 7x7 double]
[ 8x8 double]
```

Το 4 × 4 μαγικό τετράγωνο ανακτάται με την M{4}



Χαρακτήρες και κείμενο

Κείμενο (αλφαριθμητικές σειρές) εισάγεται με χρήση αποστρόφων. Για παράδειγμα,

```
s = 'Hello'
```

Το αποτέλεσμα δεν είναι το ίδιο είδος αριθμητικής μήτρας ή πίνακα που έχουμε εξετάσει μέχρι τώρα, αλλά ένας πίνακας *χαρακτήρων* 1×5. Εσωτερικά, το σύστημα καταχωρεί τους χαρακτήρες ως αριθμούς, αν και όχι κινητής υποδιαστολής. Η εντολή

```
a = double(s)
```

μετατρέπει τον πίνακα χαρακτήρων σε ένα αριθμητικό πίνακα περιέχοντα αναπαραστάσεις κινητής υποδιαστολής, των κωδικών ASCII εκάστου χαρακτήρα:

```
a =  
    72    101    108    108    111
```

Η εντολή

```
s = char(a)
```

αντιστρέφει την μετατροπή. Η μετατροπή αριθμών σε χαρακτήρες βοηθά στην διερεύνηση των γραμματοσειρών που διατίθενται στον υπολογιστή σας. Οι εκτυπώσιμοι χαρακτήρες στο βασικό σύνολο χαρακτήρων ASCII, αναπαρίσταται με τους ακέραιους αριθμούς 32:127. Οι ακέραιοι αριθμοί μικρότεροι από 32 αντιπροσωπεύουν μη εκτυπώσιμους χαρακτήρες. Αυτοί οι ακέραιοι μπορούν να διευθετηθούν σε ένα εύχρηστο πίνακα 6×16 με την εντολή

```
F = reshape(32:127, 16, 6)';
```

Οι εκτυπώσιμοι χαρακτήρες στο εκτεταμένο σύνολο ASCII αναπαρίστανται από τους αριθμούς F+128. Όταν αυτοί οι ακέραιοι ερμηνεύονται ως χαρακτήρες, το αποτέλεσμα εξαρτάται από την χρησιμοποιούμενη γραμματοσειρά. Διαφορετικές αλφαριθμητικές μεταβλητές μπορούν να ενωθούν με πράξεις σύνενωσης τη χρήση αγκυλών. Η εντολή

```
h = [s, ' world']
```

ενώνει τα αλφαριθμητικά οριζόντια και παράγει

```
h =  
    Hello world
```

Η εντολή

```
v = [s; 'world']
```

ενώνει κατακόρυφα και παράγει

```
v =  
    Hello  
    world
```

Σημειώσατε ότι στο *h*, οφείλει να τοποθετηθεί ένα κενό προ του 'w' για να διαχωρίσει τις λέξεις και ότι αμφότερες οι λέξεις στο *v* πρέπει να έχουν το ίδιο μήκος. Οι προκύπτοντες πίνακες χαρακτήρων έχουν μεγέθη 1×11 (*h*) και 2×5 (*v*) αντιστοίχως.

Η διαχείριση κειμένων που αποτελούνται από γραμμές μεταβλητού μήκους μπορεί να γίνει με δύο τρόπους: Είτε με εξίσωση του μήκους των γραμμών με προσθήκη κενών (*padding*), είτε με χρήση *κυψελοπίνακων* κειμένου. Στην πρώτη περίπτωση – δημιουργία πινάκων χαρακτήρων – ο χρήστης πρέπει να προσέξει ώστε εκάστη γραμμή να έχει το ίδιο μήκος, το οποίο πρέπει να είναι τουλάχιστον ίσο προς το μήκος της μακρύτερης γραμμής του κειμένου, γεμίζοντας τις ουρές των βραχύτερων γραμμών με κενά.

Η συνάρτηση `char` αυτοματοποιεί αυτή την διαδικασία, όπως στο ακόλουθο παράδειγμα, όπου η

```
S = char('A', 'rolling', 'stone', 'gathers', 'momentum.')
```

παράγει ένα 5×9 πίνακα χαρακτήρων:

```
S =  
    A _____  
    Rolling__  
    Stone____  
    Gathers__  
    momentum.
```

Εναλλακτικά, το κείμενο μπορεί να αποθηκευθεί σε *κυψελοπίνακα* όπως στο παράδειγμα

```
C = {'A'; 'rolling'; 'stone'; 'gathers'; 'momentum.'}
```

Όπου δημιουργείται ένας 5×1 *κυψελοπίνακας* χαρακτήρων που δεν απαιτεί προσθήκη κενών διότι εκά-

στη γραμμή του πίνακα μπορεί να έχει διαφορετικό μήκος::

```
C =
    'A'
    'rolling'
    'stone'
    'gathers'
    'momentum.'
```

Ένας πίνακας χαρακτήρων μπορεί να μετατραπεί σε κυβελοπίνακα με την

```
C = cellstr(S)
και αντιστρόφως
S = char(C)
```

Δομές (Structures)

Οι δομές είναι πολυδιάστατοι πίνακες με στοιχεία προσπελάσιμα μέσω αλφαριθμητικών δηλωτών πεδίων. Για παράδειγμα, η

```
S.name = 'Ed Plum'; S.score = 83; S.grade = 'B+'
```

παράγει creates μία βαθμωτή δομή με τρία πεδία:

```
S =
    name: 'Ed Plum'
    score: 83
    grade: 'B+'
```

Επισημαίνεται ότι οι δομές είναι πίνακες οι οποίοι μπορούν να επαυξηθούν με προσθήκη νέων στοιχείων. Στην περίπτωση αυτή, κάθε στοιχείο του πίνακα είναι μία δομή με πολλά πεδία. Πεδία μπορούν να προστίθενται ένα κάθε φορά. Έτσι, γράφομε

```
S(2).name = 'Toni Miller';
S(2).score = 91;
S(2).grade = 'A-';
```

ή χρησιμοποιούμε την εντολή:

```
S(3) = struct('name','Jerry Garcia', 'score',70,'grade','C'),
προκειμένου να προστεθούν στοιχεία.
```

Όταν η δομή είναι αρκετά μεγάλη, εμφανίζεται μόνο μία σύνοψη του περιεχομένου της:

```
S =
    1x3 struct array with fields:
    Name
    score
    grade
```

Υπάρχουν αρκετοί τρόποι αναδιαμόρφωσης των πεδίων μίας δομής σε άλλους πίνακες ή ανύσματα. Η εντολή

```
S.score
```

Είναι ίδια με την εντολή

```
S(1).score, S(2).score, S(3).score,
```

η οποία αποτελεί ένα κατάλογο διαχωρισμένο με κόμματα (*comma-separated list*). Χωρίς άλλη μέριμνα, η εντολή αυτή δεν προσφέρει τίποτε θυσιαστικά χρήσιμο, διότι εκχωρεί τους τρεις βαθμούς (scores) στην μεταβλητή ans, έναν κάθε φορά και εμφανίζει το αποτέλεσμα κάθε εκχώρησης στην οθόνη χωρίς περαιτέρω δράση. Όταν όμως η ανωτέρω έκφραση κλεισθεί σε αγκύλες,

```
[S.score]
```

είναι το ίδιο όπως

```
[S(1).score, S(2).score, S(3).score]
```

Και δημιουργεί ένα οριζόντιο άνυσμα (row vector) που περιλαμβάνει και τους τρεις βαθμούς ως στοιχεία:

```
ans =
    83    91    70
```

Ομοίως, γράφοντας

```
S.name
```

εκχωρούμε τα ονόματα, ένα κάθε φορά, στην μεταβλητή ans. Αντιθέτως η εντολή

```
{S.name}
```

δημιουργεί έναν 1×3 κυψελοπ'ινακα περιέχοντα τα τρία ονόματα:

```
ans =
```

```
    'Ed Plum'    'Toni Miller'    'Jerry Garcia'
```

και η

```
char(S.name)
```

καλεί την συνάρτηση char με τρία ορίσματα προκειμένου να δημιουργήσει ένα πίνακα χαρακτήρων από τα πεδία ονομάτων

```
ans =
```

```
    Ed Plum  
    Toni Miller  
    Jerry Garcia
```

5. Ασκήσεις

1. Απαντήσατε τις ακόλουθες ερωτήσεις για τον πίνακα:

$$\mathbf{G} = \begin{bmatrix} 0.6 & 1.5 & 2.3 & -0.5 \\ 8.2 & 0.5 & -0.1 & -2.0 \\ 5.7 & 8.2 & 9 & 1.5 \\ 0.5 & 0.5 & 2.4 & 0.5 \\ 1.2 & -2.3 & -4.5 & 0.5 \end{bmatrix}$$

- i. Ποίο είναι το μέγεθος του \mathbf{G} ;
- ii. Παραθέσατε τους δείκτες αναφοράς για τις θέσεις των στοιχείων με τιμή 0.5.

2. Ποίο είναι το μέγεθος των κατωτέρω πινάκων; Ελέγξατε τις απαντήσεις σας εισάγοντας τους πίνακες στο MATLAB με τις εντολές:

- i. $\mathbf{B} = [2; 4; 6; 10]$
- ii. $\mathbf{C} = [5 \ 3 \ 5; 6 \ 2 \ -3]$
- iii. $\mathbf{E} = [3 \ 5 \ 10 \ 0; 0 \ 0 \ \dots$
 $0 \ 3; 3 \ 9 \ 9 \ 8]$
- iv. $\mathbf{T} = [4 \ 24 \ 9]$
 $\mathbf{Q} = [\mathbf{T} \ 0 \ \mathbf{T}]$
- v. $\mathbf{V} = [\mathbf{C}(2,1); \mathbf{B}]$
- vi. $\mathbf{A}(2,1) = -3$

3. Εξηγήσατε το περιεχόμενο των ακολούθων πινάκων, οι τρεις τελευταίοι εκ των οποίων βασίζονται στον πίνακα \mathbf{G} της Άσκησης 1 ανωτέρω:

- i. $\mathbf{C} = 10:15$
- ii. $\mathbf{D} = [4:9; 1:6]$
- iii. $\mathbf{F} = 0.0:0.1:1.0$
- iv. $\mathbf{A} = \mathbf{G}(:,2)$
- v. $\mathbf{T1} = \mathbf{G}(4:5, 1:3)$
- vi. $\mathbf{T2} = \mathbf{G}(1:2:5, :)$

4. Κατασκευάσατε τις εντολές MATLAB που υπολογίζουν τις κάτωθι αριθμητικές εκφράσεις:

- i. $s = \frac{y_2 - y_1}{x_2 - x_1}$ (κλίση ευθείας γραμμής)
- ii. $R = \frac{1}{\frac{1}{r_1} + \frac{1}{r_2} + \frac{1}{r_3}}$ (Αντίσταση παράλληλου κυκλώματος)
- iii. $f = 1 + \frac{b}{u} + \frac{c}{u^2}$ (Συντελεστής διόρθωσης για την πτώση πίεσης λόγω τριβών)
- iv. $L = f \cdot p \cdot \frac{1}{d} \cdot \frac{u^2}{2}$ (Πτώση πίεσης λόγω τριβών)

5. Με χρήση του MATLAB, δώσατε τις τιμές του ανύσματος \mathbf{C} μετά την εκτέλεση των ακολούθων εντολών, στις οποίες τα ανύσματα \mathbf{A} και \mathbf{B} είναι γνωστά.

$$\mathbf{A} = [2 \ -1 \ 5 \ 0], \quad \mathbf{B} = [3 \ 2 \ -1 \ 4]$$

- i. $C = B + A - 3$
- ii. $C = 2 * A + A.^B$
- iii. $C = A./B$
- iv. $C = A.^B$
- v. $C = 2.^B + A$
- vi. $C = 2 * B / 3 .* A$

6. Υπολογίσατε τις ακόλουθες εκφράσεις και ακολούθως ελέγξατε τις απαντήσεις σας χρησιμοποιώντας το MATLAB.

- i. `round(-2.6)`
- ii. `fix(-2.6)`
- iii. `floor(-2.6)`
- iv. `ceil(-2.6)`
- v. `sign(-2.6)`
- vi. `rem(15,2)`
- vii. `floor(ceil(10.8))`
- viii. `log10(100) + log10(0.001)`
- ix. `abs(-5:5)`
- x. `round([0:0.3:2, 1:0.75:4])`

7. Κατασκευάσατε εκφράσεις MATLAB προς υπολογισμό των κάτωθι μεγεθών

- i. $g = \sqrt{ui^2 + 2 \cdot a \cdot x}$ Ομοιόμορφα επιταχυνόμενη κίνηση
- ii. $f = \frac{1}{\sqrt{\frac{2\pi c}{L}}}$ Συχνότητα ταλάντωσης κυκλώματος L
- iii. $r = 2ui^2 \cdot \frac{\sin(b) \cdot \cos(b)}{g}$ Βεληνεκές βλήματος
- iv. $l = k \cdot \sqrt{1 - \left(\frac{y}{c}\right)^2}$ Συστολή μήκους (σχετικιστική κίνηση)
- v. $c = \frac{38.1972 \cdot (r^3 - s^3) \cdot \sin a}{(r^2 - s^2) \cdot a}$ Απόσταση του κέντρου βάρους από ένα επίπεδο αναφοράς εντός κενής κυλινδρικής τομής.

8. Προσδιορίσατε τους πίνακες που αναπαριστώνται με τις ακόλουθες συναρτήσεις (δίδονται w, x και y):

$$w = [0 \ 3 \ -2 \ 7] \quad x = [3 \ -1 \ 5 \ 7] \quad y = \begin{bmatrix} 1 & 3 & 7 \\ 2 & 8 & 4 \\ 6 & -1 & -2 \end{bmatrix}$$

- i. `max(w)`
- ii. `min(y)`
- iii. `min(w,x)`
- iv. `mean(y)`
- v. `median(y)`
- vi. `cumprod(y)`
- vii. `sort(2*w + x)`
- viii. `sort(y)`

9. Δώσατε εντολές MATLAB που να παράγουν 10 τυχαίους αριθμούς εντός του δεδομένου εύρους διακύμανσης. Ελέγξατε τις απαντήσεις σας εκτελώντας τις εντολές και τυπώνοντας τα αποτελέσματά:

- i. Ομοιόμορφα κατανεμημένοι τυχαίοι αριθμοί μεταξύ 0 και 10
- ii. Ομοιόμορφα κατανεμημένοι τυχαίοι αριθμοί μεταξύ -1 και 1
- iii. Ομοιόμορφα κατανεμημένοι τυχαίοι αριθμοί μεταξύ -20 και -10
- iv. Ομοιόμορφα κατανεμημένοι τυχαίοι αριθμοί μεταξύ 4.5 και 5.0
- v. Ομοιόμορφα κατανεμημένοι τυχαίοι αριθμοί μεταξύ $-\pi$ και π .

10. Προσδιορίσατε εάν οι ακόλουθες εκφράσεις είναι αληθείς ή ψευδείς. Κατόπιν ελέγξατε τις απαντήσεις σας με το MATLAB.

$a = 5.5$ $b = 1.5$ $k = -3$

- i. $a < 10.0$
- ii. $a + b > = 6.5$
- iii. $k \sim = 0$
- iv. $b - k > a$
- v. $-(a == 3*b)$
- vi. $-k < = k + 6$
- vii. $a < 10 \& a > 5$
- viii. $\text{abs}(k) > 3 \mid k < b-a$

11. Δώσατε την σειρά των εντολών MATLAB που ανταποκρίνονται στις ακόλουθες διαδικασίες:

- i. Εάν η διαφορά μεταξύ $v1$ και $v2$ είναι μεγαλύτερη του 10, τα΄θωψε τις τιμές $v1$ και $v2$.
- ii. Εάν ο φυσικός λογάριθμος του x είναι μεγαλύτερος ή ίσος του 3, θέσε την μεταβλητή $time$ ίση προς 0 και αύξησε την $count$ κατά μία μονάδα.
- iii. Εάν $dist$ είναι μικρότερη ή ίση του 50.0 και η $time$ είναι μεγαλύτερη του 10.0, αύξησε την $time$ κατά 2, αλλιώς αύξησε την $time$ κατά 2.5
- iv. Εάν $dist$ είναι μεγαλύτερη ή ίση του 100.0, αύξησε την $time$ κατά 2.0. Εάν $dist$ είναι μεταξύ 50.0 και 100.0, αύξησε την $time$ κατά 1. Αλλιώς, αύξησε την $time$ κατά 0.5.