ΠΑΡΑΡΤΗΜΑ Α
Εισαγωγή στο MATLAB

Περιεχόμενα Παραρτήματος Α

Α.1 Γενικά

Το MATLAB είναι υψηλού επιπέδου γλώσσα προγραμματισμού κατάλληλη για επίλυση σύνθετων μαθηματικών προβλημάτων. Περιλαμβάνει ένα σύνολο συναρτήσεων (functions), είτε ενσωματωμένων στον interpreter είτε σε αρχεία εντολών (m-files), οι οποίες καλούνται, μεταφράζονται εντολή προς εντολή και εκτελούνται. Ο προγραμματιστής μπορεί να παράξει δικά του m-files, με αυτόνομα εκτελέσιμα προγράμματα (scripts) ή συναρτήσεις.

Το MATLAB παρέχει παραθυρικό περιβάλλον προγραμματισμού, με τρία βασικά παράθυρα: το *Παράθυρο Εντολών* (*Command window*), το *Παράθυρο Εισαγωγής Κειμένου* (*Editor window*) και το *Παράθυρο Σχημάτων* (*Figure window*). Στο Παράθυρο Εντολών δίνουμε εντολές οι οποίες εκτελούνται αμέσως μετά το πάτημα του πλήκτρου νέας γραμμής (enter). Στο Παράθυρο Κειμένου γράφουμε αρχεία εντολών (m-files), ενώ στο Παράθυρο Σχημάτων σχεδιάζουμε γραφικές παραστάσεις με τη χρήση κατάλληλων σχεδιαστικών εντολών. Υπάρχουν και άλλα δευτερεύοντα παράθυρα, όπως αυτό της *Ιστορίας των εντολών* (*Command History*) και του *Χώρου εργασίας* (*Workspace*). Αν επιλέξουμε ένα σύνολο εντολών, είτε από το Παράθυρο Κειμένου, είτε από το παράθυρο Ιστορίας των εντολών και πατήσουμε δεξί κλίκ, έχουμε (μεταξύ άλλων επιλογών) τη δυνατότητα άμεσης εκτέλεσης του επιλεγέντος συνόλου (F9 ή επιλογή Evaluate Selection).

Όλες οι μεταβλητές οι οποίες έχουν οριστεί από τη στιγμή εκκίνησης του MATLAB είναι αποθηκευμένες σε χώρο μνήμης προσβάσιμο από όλα τα προγράμματα και αποτελούν το Workspace. Μπορούμε να σβήσουμε επιλεκτικά μεταβλητές με την εντολή `clear` (π.χ. `clear x`, για σβήσιμο της x), ή και όλο το workspace με `clear all` (κάτι που συνιστάται πριν την εκτέλεση ενός νέου προγράμματος)

Α.2 Αριθμοί και βασικές δομές δεδομένων στο MATLAB

Το MATLAB χρησιμοποιεί τη συνήθη δεκαδική παράσταση αριθμών. Η περιοχή που μπορεί να παρασταθεί είναι προσεγγιστικά η $\pm(10^{-308}, 10^{308})$. Πολύ μικροί ή πολύ μεγάλοι αριθμοί παριστάνονται σε εκθετική μορφή. Τυπικά παραδείγματα παραστάσεων αριθμών στο MATLAB είναι

67423.97215 2468.1357e-4 -0.1234e3

Η βασική δομή αριθμητικών δεδομένων του MATLAB είναι ο ορθογωνικός πίνακας (nxm) μιγαδικών αριθμών (συμπεριλαμβανόμενων των πραγματικών), ο οποίος και δεν χρειάζεται διαστασιοδότηση, δεν χρειάζεται δηλ. να δηλώσουμε τις τιμές των *n* και m. Εκφυλισμένες περιπτώσεις αυτής της βασικής δομής είναι τα διανύσματα σειράς (n=1) ή στήλης (m=1), αλλά και οι απλοί βαθμωτοί αριθμοί (μιγαδικοί ή πραγματικοί). Έτσι δίνοντας τις παρακάτω εντολές, δημιουργούμε το διάνυσμα-σειρά z=[3.9+2i 5.2 4+6.5i] , διάστασης 1x3, όπως επιβεβαιώνει το εκτυπούμενο απότέλεσμα (άσπρο υπόβαθρο).

```
x=5.2;
y=4+6.5*i;
z=[3.9+2*j x y]
```

```
z =
   3.9000 + 2.0000i   5.2000   4.0000 + 6.5000i
```

Οι μιγαδικοί αριθμοί δίνονται στη μορφή a+b*i ή a+b*j, με a,b πραγματικούς. Το semicolon (ελληνικό ερωτηματικό) **;** στο τέλος μιας εντολής αποτρέπει την εκτύπωση του αποτελέσματος, ενώ χωρίς αυτό έχουμε εκτύπωση, όπως συνέβη πιοπάνω με την τελευταία εντολή για το z. Σε μία γραμμή μπορούν να δοθούν πολλές εντολές, χωριζόμενες βεβαίως με **;** (π.χ. θα μπορούσαν οι δύο πρώτες γραμμές πιοπάνω να γραφούν ως x=5.2; y=4+6.5*i;). Οτιδήποτε εκκινά με το χαρακτήρα **%** μέχρι το τέλος της γραμμής αποτελεί σχόλιο και αγνοείται κατά την εκτέλεση. Αν δώσουμε το όνομα μεταβλητής χωρίς ; εκτυπώνεται η τιμή της (με την προϋπόθεση ότι έχει οριστεί στο workspace του MATLAB). Η συνάρτηση `size(A)` επιστρέφει διάνυσμα με τις διαστάσεις του πίνακα A, ενώ η `length(A)` επιστρέφει τη μεγαλύτερη από τις δύο διαστάσεις:

```
y
```

```
y =
   4.0000 + 6.5000i
```

```
size(z)
```

```
ans =
     1     3
```

```
length(z)
```

```
ans =
     3
```

Κατασκευή και δεικτοδότηση πινάκων

Ένας πίνακας nxm μπορεί να δοθεί με τις n γραμμές του στην ίδια γραμμή εντολής χωριζόμενες με **;** ή σε διαδοχικές γραμμές εντολής. Έτσι, οι πίνακες a και b στο παρακάτω τμήμα προγράμματος είναι ίσοι.

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

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

Η δεικτοδότηση των πινάκων αρχίζει από το 1. Για τα z, b που έχουν ορισθεί νωρίτερα, το `z(3)` θα επιστρέψει την τιμή `4.0000 + 6.5000i`, το `b(2,4)` την τιμή 8, ενώ η αναφορά `z(0)` ή z(4) θα προκαλέσει μήνυμα λάθους.

Ο ανάστροφος (transpose) πίνακας δίνεται με τον τελεστή «τελεία απόστροφος» **.'**, ενώ μόνη η απόστροφος δίνει τον μιγαδικό συζυγή ανάστροφο. π.χ., αν `a` όπως πιοπάνω

```
a.'
```

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

Η έκφραση a:b:c, όπου a,b,c πραγματικοί, δίνει το διάνυσμα [a a+b a+2b…d], με d≤c. Το MATLAB διαθέτει επίσης έτοιμες συναρτήσεις που χρησιμοποιούνται για την κατασκευή σταθερών πινάκων επιθυμητής διάστασης:

```
t=[0:0.1:0.5];
f=3*ones(2,4); g=zeros(1,5);
t
f
g
```

```
t =
         0    0.1000    0.2000    0.3000    0.4000    0.5000

f =
     3     3     3     3
     3     3     3     3

g =
     0     0     0     0     0
```

Σε σχέση με το χαρακτήρα **:** (colon) θα πρέπει να πούμε ότι χρησιμοποιείται στη δεικτοδότηση των πινάκων για να επιλέξει συγκεκριμένα στοιχεία, σειρές ή στήλες τους. Έτσι, για τα t και b όπως ορίστηκαν παραπάνω, το t(3:5) επιστρέφει τα στοιχεία t(3) έως t(5), το b(2,:) επιστρέφει τη δεύτερη γραμμή του b, ενώ το b(:,2:4) επιστρέφει τον πίνακα που απαρτίζεται από τις στήλες 2 έως και 4 του b.

```
b = [1 2 3 4
     5 6 7 8];
t=[0:0.1:0.5];
t(3:5)
b(2,:)
b(:,2:4)
```

```
ans =
    0.2000    0.3000    0.4000

ans =
     5     6     7     8

ans =
     2     3     4
     6     7     8
```

Τέλος, ένας πίνακας μπορεί να αυξηθεί με προσθήκη σειρών ή στηλών, ή ακόμη και να αναδιαταχθεί, ως εξής:

```
t=[t 0.6 0.7];
b =[b [11 12 13 14; 15 16 17 18]];
b=[b(:,5:8) b(:,1:4)];
t
b
```

```
t =
  0  0.100  0.200  0.300  0.400  0.500  0.600    0.700

b =
    11    12    13    14     1     2     3     4
    15    16    17    18     5     6     7     8
```

Α.3 Αριθμητικές πράξεις

Η πρόσθεση (+) και η αφαίρεση (-) γίνονται κανονικά μεταξύ ισομεγέθων πινάκων, στοιχείο προς στοιχείο. Ο πολλαπλασιασμός A*B ορίζεται μεταξύ πίνακα A (οποιασδήποτε διάστασης) και αριθμού B, καθώς και μεταξύ πινάκων A (nxm) και B (mxk), δίνοντας το γραμμικό αλγεβρικό τους γινόμενο, διάστασης nxk. Μεταξύ ισομεγέθων πινάκων ορίζεται ο πολλαπλασιαμός στοιχείο-προς-στοιχείο με τον τελεστή «τελεία αστεράκι» **.***, καθώς και η διαίρεση στοιχείο-προς-στοιχείο, **./**. Ορίζεται επίσης η δύναμη πίνακα στοιχείο-προς-στοιχείο με τον τελεστή **.^**. Η διαίρεση (/) πίνακα με αριθμό δίνει, κατά τα γνωστά, ισομεγέθη πίνακα με όλα τα στοιχεία του διαιρεμένα με τον αριθμό, ενώ η διαίρεση μεταξύ πινάκων ορίζεται σε δύο μορφές: *δεξιά διαίρεση* (/) και *αριστερή διαίρεση* (\), δίνει δε τη λύση γραμμικών συστημάτων. Συγκεκριμένα: αν A τετραγωνικός πίνακας (nxn) και X πίνακας (nxk), με τον ίδιο αριθμό γραμμών όπως ο A, τότε ο Y=A\X είναι η λύση Y του συστήματος AY=X. Μεταξύ των δύο διαιρέσεων ισχύει η ισοδυναμία (A\X)'=X'/A'.
Σε εντολές με πολλές αριθμητικές πράξεις ισχύει η γνωστή προτεραιότητα μεταξύ των πράξεων, καθώς και η δυνατότητα τροποποίησης της προτεραιότητας αυτής με χρήση παρενθέσεων. Τα παραδείγματα που ακολουθούν φωτίζουν τον τρόπο εκτέλεσης αριθμητικών πράξεων μεταξύ πινάκων στο MATLAB.

```
A=[1 5 8; -2 7 3; 5 2 1];
X=[2 3 1; 5 6 1]';
a=2;
A+a*A
```

```
ans =
     3    15    24
    -6    21     9
    15     6     3
```

```
C=A.*A
```

```
C =
     1    25    64
     4    49     9
    25     4     1
```

```
C./A
```

```
ans =
     1     5     8
    -2     7     3
     5     2     1
```

```
A.^2
```

```
ans =
     1    25    64
     4    49     9
    25     4     1
```

```
Y=A\X
```

```
Y =
    0.0265   -0.1327
    0.4513    0.7434
   -0.0354    0.1770
```

```
Z=X'/A'
```

```
Z =
    0.0265    0.4513   -0.0354
   -0.1327    0.7434    0.1770
```

```
A*Y
```

```
ans =
    2.0000    5.0000
    3.0000    6.0000
    1.0000    1.0000
```

A.4 Σχεσιακοί Τελεστές (Relational Operators)

Είναι παρόμοιοι με τους σχεσιακούς τελεστές άλλων γλωσσών προγραμματισμού, και επιστρέφουν λογική τιμή 1 ή 0 («αληθές» ή «ψευδές», αντίστοιχα).

`>` *μεγαλύτερο*
`>=` *μεγαλύτερο ή ίσο*
`<` *μικρότερο*
`<` *μικρότερο ή ίσο*
`==` *ίσο*
`~=` *άνισο*

Για πίνακες (ισομεγέθεις) η σύγκριση γίνεται στοιχείο-προς-στοιχείο και επιστρέφει ισομεγέθη πίνακα λογικών τιμών, 1 ή 0. Έτσι, για τους πίνακες A και X των προηγούμενων παραδειγμάτων έχουμε:

```
(A\X)'~=X'/A'
```

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

Ακόμη

```
C=[2 4 6; 8 10 12];  D=[3 4 5; 10 11 12];
C==D
```

```
ans =
     0     1     0
     0     0     1
```

```
C>=D
```

```
ans =
     0     1     1
     0     0     1
```

Παράδειγμα 1: Να βρεθεί σε πόσα στοιχεία διαφέρουν δοσμένα διανύσματα **x** και **y**.

```
x=[1 2 3 4 5];  y=[1 2 4 4 6]; sum(x~=y)
```

```
ans =
     2
```

Η συνάρτηση sum() υπολογίζει το άθροισμα των στοιχείων διανύσματος.

A.5 Λογικοί τελεστές

Τρείς λογικοί τελεστές χρησιμοποιούνται στο MATLAB:

& λογικό AND
| λογικό OR
~ λογικό NOT

Σε πίνακες εφαρμόζονται στα επιμέρους στοιχεία. Πρέπει να τονιστεί ότι κάθε μη μηδενικός πραγματικός αριθμός θεωρείται στις λογικές πράξεις ως 1. Έτσι

```
A=[1.5 0 -2.3 1]; B=[0 1 2 0];
A&B
A|B
~A
```

```
ans =
     0     0     1     0

ans =
     1     1     1     1

ans =
     0     1     0     0
```

Α.6 Έλεγχος ροής προγράμματος

Οι εντολές **break**, **else**, **elseif**, **end**, **error**, **for**, **if**, **return** και **while** επιτρέπουν την υπό συνθήκη εκτέλεση τμημάτων προγράμματος. Πιο συγκεκριμένα:

- Οι εντολές **for...end** χρησιμοποιούνται για την επανάληψη του τμήματος προγράμματος που περικλείουν.
- Οι εντολές **if**<συνθήκη>**...end** εκτελούν το περικλειόμενο τμήμα εφόσον η σηνθήκη είναι αληθής. Οι εντολές **else** και **elseif** χρησιμοποιούνται σε συνδυασμό με την if για υπό συνθήκη εκτέλεση τμήματος εντός του βρόχου.
- Οι **while**<συνθήκη>**...end** εκτελούν επανειλημμένα το περικλειόμενο τμήμα προγράμματος εφόσον η <συνθήκη> παραμένει αληθής. Στην εκδοχή <while 1 ... end> εκτελείται συνεχώς το περικλειόμενο τμήμα προγράμματος, έως ότου κάποιος εσωτερικός έλεγχος, σε συνδυασμό με τις εντολές **break; end;**, το σταματήσει.
- Γενικά, η εντολή **break** τερματίζει την εκτέλεση ενός βρόχου.
- Η εντολή **error** χρησιμοποιείται για την εμφάνιση λάθους και διακοπή εκτέλεσης των συναρτήσεων.

Τα παρακάτω παραδείγματα δείχνουν τη χρήση των εντολών ελέγχου.

Παράδειγμα 2: Σε διάνυσμα **x** να βρεθεί η θέση του πρώτου στοιχείου που είναι μεγαλύτερο από το προηγούμενό του. Αν δεν υπάρχει, τυπώνεται μηδέν.

```
x=[10 9 8 7 6 8 10 12]; flag =0;
for i=2:length(x)
    if x(i)>x(i-1) flag=1; break; end;
end
i*flag
```

```
ans =
     6
```

Παράδειγμα 3: Να τυπωθούν οι όροι γεωμετρικής προόδου 1, r, r^2,… με r>1, των οποίων το άθροισμα δεν υπεραίνει το 100, καθώς και το άθροισμα.

```
a=1; s=a; x=[a]; r=2.5;
while 1
    a=a*r; s=s+a;
    if s>100 s=s-a; break; end
    x=[x a];
end;
x
s
```

```
x =
    1.0000    2.5000    6.2500   15.6250   39.0625

s =
    64.4375
```

Α.7 Ειδικοί χαρακτήρες και μεταβλητές

Όπως κάθε γλώσσα προγραμματισμού, έτσι και το MATLAB κρατάει συγκεκριμένους χαρακτήρες και λέξεις-κλειδιά ως ονόματα σταθερών και συναρτήσεων, τα οποία ο προγραμματιστής δεν θα πρέπει να αναμίξει με δικές του μεταβλητές. Έτσι, το **pi** παριστάνει το π (3.14159...), τα **i** και **j** παριστάνουν τη φανταστική μονάδα (δεν πρέπει να γίνει μίξη με τους δείκτες βρόχων), το **NaN** σημαίνει Not a Number και λαμβάνεται σε περιπτώσεις απροσδιοριστίας (πχ. ∞-∞), το **inf** παριστάνει το άπειρο και εμφανίζεται στις περιπτώσεις διαίρεσης με μηδέν, κ.ο.κ.

Α.8 Εισαγωγή και εμφάνιση δεδομένων -- μορφότυπο

Μπορούμε να εισάγουμε δεδομένα κατά τη διάρκεια εκτέλεσης του προγράμματος με χρήση της συνάρτησης `input()`. Η εμφάνιση δεδομένων στην οθόνη είδαμε ότι μπορεί να γίνει με εντολές προγράμματος ή αναγραφή μεταβλητών χωρίς το **;** στο τέλος. Για τον ίδιο σκοπό μπορεί να χρησιμοποιηθεί και η εντολή `disp()`.

Όλες οι πράξεις εκτελούνται εσωτερικά με διπλή ακρίβεια. Το μορφότυπο, ωστόσο, εμφάνισης των αποτελεσμάτων στην οθόνη μπορεί να ελεγχθεί με την εντολή `format`. Χωρίς παραμέτρους, η εντολή αυτή εμφανίζει την στάνταρντ μορφή, που

για αριθμούς κινητής υποδιαστολής είναι η σύντομη μορφή (`format short`, 4 δεκαδικών ψηφίων, ή σε εκθετική μορφή, αν ο αριθμός είναι πολύ μεγάλος ή πολύ μικρός). Με την `format long` έχουμε εμφάνιση 15 δεκαδικών ψηφίων, ενώ η προσθήκη του e δίνει εκθετική μορφή.

```
x=input('δώστε αριθμό:  ');
disp(x);
format long;
disp(x);
format long e
disp(x);
```

```
   0.0030
   0.003000000000000
   3.000000000000000e-003
```

A.9 Γραφικές παραστάσεις

Υπάρχει μια πληθώρα από συναρτήσεις για σχεδίαση δισδιάστατων ή τρισδιάστατων γραφικών παραστάσεων, καθώς και για προσθήκη τίτλων, ενδείξεων αξόνων κλπ. Η `plot()` είναι η περισσότερο χρησιμοποιούμενη.

A.10 Αρχεία προγραμμάτων και συναρτήσεων

Τα προγράμματα MATLAB αποθηκεύονται σε αρχεία απλού κειμένου (plain text) με το επίθημα **.m** (**m-files**). Τα ονόματα των m-files πρέπει να αρχίζουν με κάποιο γράμμα ακολουθούμενο από άλλα γράμματα, μέχρι 18, ή/και ψηφία ή και το χαρακτήρα υπογράμμισης (το μείον <–> ή η τελεία <.> δεν επιτρέπονται). Υπάρχουν δύο είδη m-files: **scripts** και **functions**. Ένα script είναι ένα αυτόνομο, εκτελέσιμο πρόγραμμα, ενώ ένα function κωδικοποιεί μια συνάρτηση, η οποία καλείται από άλλες συναρτήσεις ή προγράμματα MATLAB και πρέπει να περιέχει στην πρώτη γραμμή κώδικα τη λέξη-κλειδί function. Μια άλλη σημαντική διαφορά μιας function από ένα script είναι ότι χρησιμοποιεί δικό της χώρο μεταβλητών, ο οποίος και «καθαρίζεται» μετά το τέλος της συνάρτησης, ενώ ένα script χρησιμοποιεί το workspace του MATLAB.

Στο παράδειγμα που ακολουθεί, υπολογίζεται και σχεδιάζεται η rms τιμή σήματος s, σε ολισθαίνον παράθυρο μήκους w δειγμάτων που εισάγεται κατά το χρόνο εκτέλεσης. Η συνάρτηση root_mean_square() δέχεται διάνυσμα και υπολογίζει την rms τιμή των στοιχείων του. Πρέπει να βρίσκεται σε αρχείο με το όνομα root_mean_square.m, αποθηκευμένο στο τρέχον directory

```
s=2+0.5*rand(1,80); % σήμα: σταθερά+θόρυβος
p=[]; % Το διάνυσμα τιμών rms (προς το παρόν κενό)
w=input('δώστε μήκος ολισθαίνοντος παραθύρου (άρτιο):  ');
t=w/2+1:length(s)-w/2; % σημεία για τον υπολογισμό της τιμής rms
for i=1:length(t)
    z=s(t(i)-w/2:t(i)+w/2-1);
    p=[p root_mean_square(z)]; % το p αυξάνεται κατά μία τιμή
end
plot(t, s(w/2+1:length(s)-w/2), ':', t,p)
title(['σήμα και τιμή rms υπολογισμένη σε παράθυρο '...
     num2str(w),' δειγμάτων'])
grid;
axis([0 length(s) min(s)-0.5 max(s)+0.5]);
```

```
function p=root_mean_square(x)
% υπολογίζει την rms τιμη του x
p=sqrt(sum(x.^2)/length(x));
end
```

Το εξαγόμενο του παραπάνω προγράμματος για w=16 είναι το παρακάτω σχήμα.