



**ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ &
ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ**

**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ**

Εργαστήριο Ψηφιακής Επεξεργασίας Σήματος

ΒΑΣΙΚΕΣ ΑΡΧΕΣ

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΜΕ ΤΟ MATLAB

Ακαδημαϊκό έτος 2018-2019

Έλεγχος ροής προγράμματος

■ Εντολές ελέγχου ροής:

- Εντολή if
- Εντολή switch
- Εντολή For
- Εντολή While
- Εντολή Continue
- Εντολή break

Η εντολή if

- Αποτιμάται μια λογική έκφραση και εκτελείτε ένα σύνολο εντολών όταν αυτή η έκφραση είναι αληθής.
- Η εντολή τερματίζει με την εντολή end.
- Δεν χρειάζονται άγκιστρα ή παρενθέσεις.
- Η εντολές elseif και else παρέχουν πολλαπλές δομές επιλογής.

Η εντολή if

```
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
```

Συσχετιστικοί Τελεστές

- Αυτοί είναι: $<$, $<=$, $>$, $>=$, $==$, και \sim
- Επενεργούν στους πίνακες κάνοντας σύγκριση στοιχείου – στοιχείου.
- Επιστρέφουν ένα πίνακα ιδίου μεγέθους με τα στοιχεία του ίσα με λογικό 1 ή λογικό 0.
- Οι πίνακες που συγκρίνονται πρέπει να έχουν το ίδιο μέγεθος.

Λογικές πράξεις

- & Πράξη AND στοιχείου με στοιχείο.
 $A \& B$ επιστρέφει πίνακα αν με στοιχεία τις λογικές τιμές 1 ή 0.
- | Πράξη OR στοιχείου με στοιχείο.
 $A | B$ επιστρέφει πίνακα αν με στοιχεία τις λογικές τιμές 1 ή 0.
- ~ Πράξη της άρνησης NOT.

Η εντολή if

● Ο κανονικός τρόπος να ελέγχουμε ισότητα μεταξύ δύο μεταβλητών είναι η εντολή `isequal`.

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

● Άλλες χρήσιμες εντολές.

```
>> isempty()
```

```
>> all
```

```
>> any
```

Η εντολή if

■ Τι νομίζετε ότι θα συμβεί στον δίπλα κώδικα:

1. Αν A και B είναι μονόμετρα μεγέθη;
2. Αν A και B είναι πίνακες;

Που αναμένουμε πρόβλημα;

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

Η εντολή SWITCH

- Η εντολή switch εκτελεί ένα σύνολο εντολών βασιζόμενη στην τιμή που έχει μια έκφραση ή μια μεταβλητή.
- Οι εντολές case, otherwise, δείχνουν τα όρια των ομάδων εντολών.
- Τελειώνει πάντα με ένα end.
- Εκτελείται πάντα μόνο μια από τις ομάδες εντολών.
- Δεν χρειάζεται εντολή break.

Η εντολή switch

```
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
```

Η εντολή for

■ Εκτελεί επαναληπτικά ένα σύνολο εντολών και για ένα προκαθορισμένο αριθμό εντολών.

■ Τερματίζει με την χρήση ενός end.

```
>> for n = 3:32
```

```
>> r(n) = rank(magic(n));
```

```
>> end
```

```
>> r
```

Η εντολή while

- Μπορεί να εκτελεί επαναληπτικά ένα σύνολο εντολών και για απροσδιόριστες φορές.
- Μήπως μπορείτε να βρείτε τι υπολογίζει ο κώδικας;

```
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
x
```

Η εντολή continue

- Η εντολή continue περνάει τον έλεγχο στην επόμενη επαναληπτική διαδικασία του ίδιου βρόχου.
- Δηλαδή, παραβλέπει τις εντολές που απομένουν στο σώμα του βρόχου.
- Σε εμφωλευμένους βρόχους ο έλεγχος περνάει στην επόμενη επαναληπτική διαδικασία του βρόχου που περικλείει την εντολή continue.

Η εντολή continue

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

Η εντολή break

- Προκαλεί έξοδο από τον βρόχο.
- Σε εμφωλευμένους βρόχους η εντολή προκαλεί έξοδο από τον βρόχο που περικλείει την εντολή.

```
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
x
```

Δομές Δεδομένων

- Υπάρχουν αρκετοί τύποι δεδομένων που περιλαμβάνονται στο MATLAB.
 - n -διάστατοι πίνακες.
 - Διατάξεις κελιών (cell arrays).
 - Χαρακτήρες και κείμενο.
 - Δομές (structures).

n -διάστατοι πίνακες

- Είναι οι πίνακες των οποίων τα στοιχεία για να προσπελαστούν απαιτούν περισσότερους από δύο διατακτικούς αριθμούς (δείκτες).
- Για την δημιουργία τους μπορούμε να χρησιμοποιήσουμε τις βασικότερες συναρτήσεις του MATLAB, `zeros`, `ones`, `rand`, `randn` με περισσότερα από δύο όρίσματα.

N-διάστατοι πίνακες

■ Δημιουργία ενός πίνακα 3 γραμμών, 4 στηλών και 5 επιπέδων.

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

■ Μπορεί να αναπαριστά την θερμοκρασία σε ένα δωμάτιο.

■ Μπορεί να αναπαριστά μια ακολουθία πινάκων.

■ Μπορεί να αναπαριστά δείγματα μιας χρονικής μεταβολής πινάκων (video).

Παραδείγματα

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

Δημιουργεί $4!=24$ διαφορετικές μεταθέσεις των αριθμών από 1 έως 4. Η k -μετάθεση είναι η γραμμή: του πίνακα $p(k,:)$

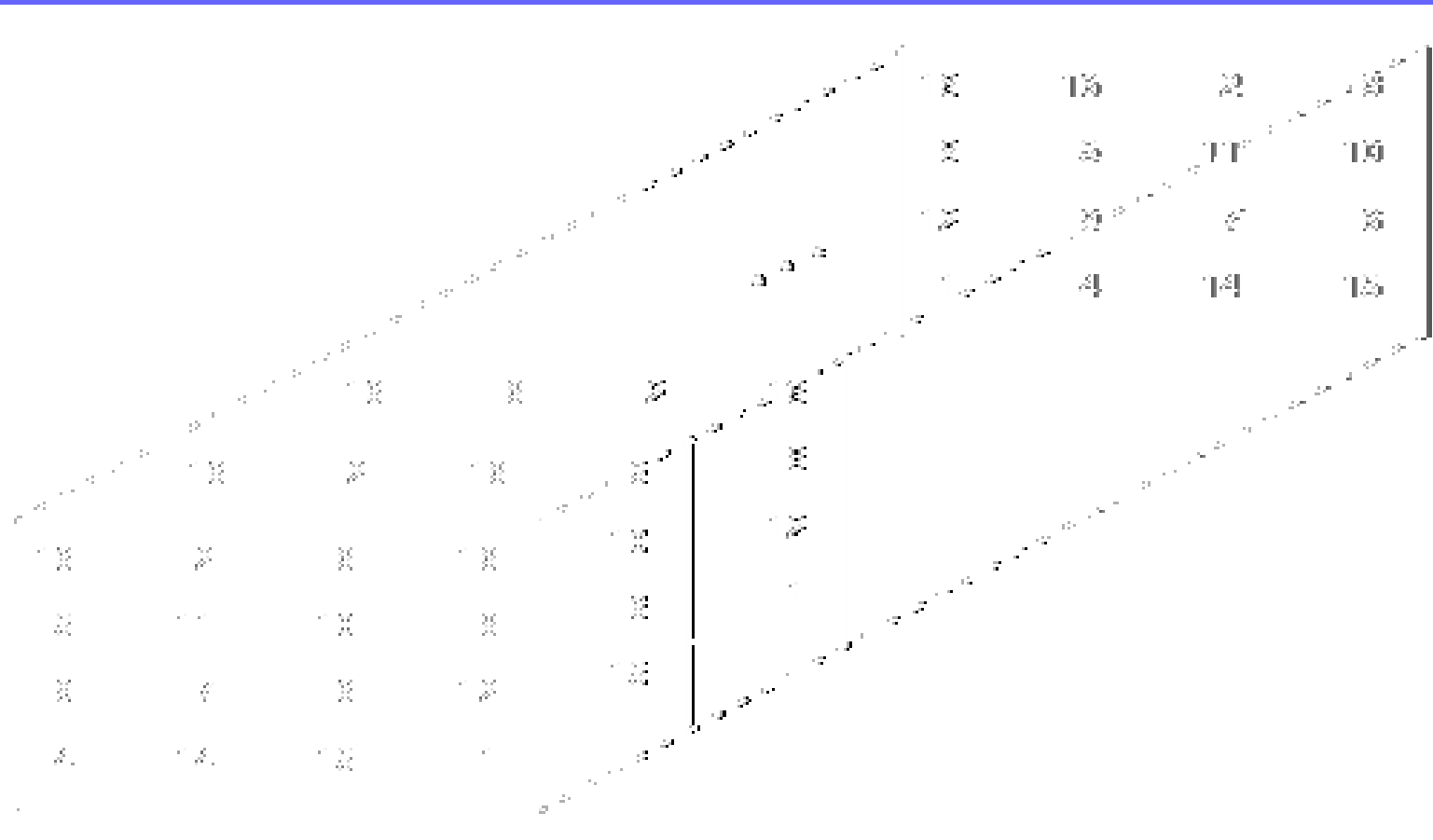
```
>> A = magic(4);
```

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

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

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

```
>> end
```



Παραδείγματα

```
>>size(M)
```

```
ans =
```

```
4 4 24
```

```
>>sum(M,1) % άθροισμα κατά την σάρωση  
του πρώτου δείκτη.
```

```
>> sum(M,2) % άθροισμα κατά την σάρωση  
του δεύτερου δείκτη.
```

```
>> sum(M,3) % άθροισμα κατά την σάρωση  
του τρίτου δείκτη.
```

Παραδείγματα

```
>>sum(M,1) %Παράγει 24 φορές το διάνυσμα  
γραμμής: 34 34 34 34
```

```
>> sum(M,2) % Παράγει 24 φορές το διάνυσμα  
στήλης: 34 34 34 34
```

```
>> sum(M,3) %Παράγει 24 φορές τον πίνακα:  
204 204 204 204  
204 204 204 204  
204 204 204 204  
204 204 204 204
```

Διατάξεις κελιών (cell arrays)

- Χρησιμοποιούνται όταν συμπεριλάβουμε σε αυτήν την δομή μια συλλογή διαφόρων τύπων δεδομένων μέσα σε άγκιστρα {}.

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

```
C =
```

```
[4x4 double] [1x4 double] [2.0923e+013]
```

- Τα δύο πρώτα στοιχεία δεν χωράνε να εμφανιστούν.

Διατάξεις κελιών (cell arrays)

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

M

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

24	3	8	51	30	5	7	57
5	22	24	12	18	51	50	16
17	44	42	20	21	43	42	24
42	22	26	37	36	30	31	30
33	24	22	29	23	33	33	25
47	22	23	44	45	19	18	43
42	12	14	52	53	11	10	56
8	22	22	5	4	32	33	1

...

1	2	3	4
2	1	4	3
3	4	2	1
4	3	1	2

1	2
2	1
3	4

1	2
2	1

1

Χαρακτήρες και κείμενο

- Εισάγουμε κείμενο στο MATLAB με χρήση των απλών αποστρόφων (quotes)

```
>> s = 'Hello'
```

- Εσωτερικά οι χαρακτήρες αναπαρίστανται ως αριθμοί ASCII.

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

```
a =
```

```
72  101  108  108  111
```

Χαρακτήρες και κείμενο

```
>> s = char(a) % επαναφέρει την αρχική  
μορφή.
```

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

```
>> char(F)
```

```
>> char(F+128)
```

Χαρακτήρες και κείμενο

■ Συνένωση

χαρακτήρων:

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

```
h =
```

```
Hello world
```

■ Προσοχή: έχει
εισαχθεί ένα κενό.

■ Ενώ αν:

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

```
v =
```

```
Hello
```

```
world
```

■ Προσοχή: οι γραμμές
πρέπει να έχουν τις
ίδιες στήλες.

Χαρακτήρες και κείμενο

- Για να ελέγξουμε διαφορετικού μήκους χαρακτήρες μπορούμε να προσθέτουμε χαρακτήρες κενούς με την συνάρτηση `char()`.

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

```
S =
```

```
A
```

```
rolling
```

```
stone
```

```
gathers
```

```
momentum
```

Χαρακτήρες και κείμενο

- Για να ελέγξουμε διαφορετικού μήκους χαρακτήρες κάνουμε χρήση διατάξεων κελιών.
- Εδώ τα κελιά είναι διαφορετικού μήκους.

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

Χαρακτήρες και κείμενο

■ Μετατροπή από κείμενο σε διατάξεις κελιών:

```
>> C = cellstr(S)
```

■ Το αντίθετο:

```
>> S = char(C)
```

Δομές (structures)

- Είναι συλλογές από διαφορετικού τύπου δεδομένα.
- Η προσπέλασή τους γίνεται με τον διαχωριστή πεδίου (field designator) που είναι ο χαρακτήρας της τελείας.

Δομές (structures)

```
>> .name = 'Ed Plum';
```

```
>> S.score = 83;
```

```
>> S.grade = 'B+';
```

```
S =
```

```
  name: 'Ed Plum'
```

```
  score: 83
```

```
  grade: 'B+'
```

Δομές (structures)

■ Μπορούμε να προσθέσουμε και άλλα στοιχεία. Δημιουργούμε πίνακες δομών:

```
>> S(2).name = 'Vazelos...';
```

```
>> S(2).score = 91;
```

```
>> S(2).grade = 'A-';
```

Δομές (structures)

- Μπορούμε να δημιουργήσουμε μια δομή με μια εντολή μόνο.

```
>> S(3) = struct('name','Vazelos Again',...  
    'score',70,'grade','C')
```

- Το αποτέλεσμα όμως πλέον δεν εμφανίζεται στην οθόνη λόγω πολλών δεδομένων.

```
S =
```

```
1x3 struct array with fields:
```

```
    name
```

```
    score
```

```
    grade
```

Δομές (structures)

- Υπάρχουν αρκετοί τρόποι για να εμφανίσουμε κάθε ένα πεδίο.
- Οι παρακάτω εντολές είναι ισοδύναμες, όχι όμως και εύχρηστες γιατί δείχνουν το αποτέλεσμα 3 φορές στην μεταβλητή `ans`.

```
>> S.score
```

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

Δομές (structures)

```
>> [S.score]
```

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

■ Οι παραπάνω εντολές είναι εύχρηστες γιατί τοποθετούν τα αποτελέσματα σε ένα πίνακα.

■ Οι συναρτήσεις επιδρούν στα μέλη των δομών.

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

Αρχεία Matlab

- Υπάρχουν δύο είδη αρχείων:
 - Scripts, τα οποία δεν δέχονται ορίσματα εισόδου ούτε επιστρέφουν ορίσματα εξόδου. Επενεργούν στα δεδομένα που υπάρχουν στο πρόγραμμα μέχρι εκείνη την στιγμή. Μετά την εκτέλεση οι μεταβλητές που χρησιμοποιήθηκαν παραμένουν στο πρόγραμμα.
 - Functions, που δέχονται μεταβλητές εισόδου και αποδίδουν τιμές σε μεταβλητές εξόδου. Οι μεταβλητές τους έχουν τοπική εμβέλεια, δηλαδή υπάρχουν μόνο κατά την διάρκεια εκτέλεσης της συνάρτησης.

Συναρτήσεις

```
function r = rank(A,tol) % δευσιμευμένη λέξη.  
% RANK Matrix rank. Σχόλια...  
% 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 % Αριθμός ορισμάτων εισόδου.  
    tol = max(size(A)') * max(s) * eps;  
end  
r = sum(s > tol);
```

Η συνάρτηση eval

- Χρησιμοποιείτε με μεταβλητές τύπου χαρακτήρα για να υλοποιήσει ισχυρές μακροεντολές. Συγκεκριμένα η εντολή `eval(s)`:
 - Υπολογίζει την έκφραση ή εκτελεί το κείμενο που βρίσκεται μέσα στην ακολουθία χαρακτήρων `s`.

```
for d = 1:31
    s = ['load August' int2str(d) '.dat'];
    eval(s)
    % Process the contents of the d-th file
end
```

Μεταβλητές ολικής εμβέλειας

- Αν θέλουμε από περισσότερες συναρτήσεις να χρησιμοποιούν μια μεταβλητή θα πρέπει να την ανακηρύξουμε ολική (global) μέσα σε όλες τις συναρτήσεις.
- Θα πρέπει να δηλώσουμε την μεταβλητή σαν ολική και στην γραμμή εντολών.
- Η δήλωση ολικής εμβέλειας πρέπει να συμβεί πριν την εκτέλεση της πρώτης χρήσης της στην συνάρτηση.
- Συνήθως κάνουμε χρήση κεφαλαίων γραμμάτων για να τις ξεχωρίζουμε από τις κοινές μεταβλητές.

Παράδειγμα χρήσης ολικής μεταβλητής

```
function h = falling(t)
```

```
global GRAVITY
```

```
h = 1/2*GRAVITY*t.^2;
```

Στην γραμμή εντολών γράφουμε:

```
>> global GRAVITY
```

```
>> GRAVITY = 32;
```

```
>> y = falling((0:.1:5)');
```

Συναρτήσεις και χαρακτήρες

- Μπορούμε να γράψουμε συναρτήσεις στο MATLAB που δέχονται ορίσματα τύπου χαρακτήρα χωρίς να κάνουμε χρήση αποστρόφων ή παρενθέσεων.
- Π.χ η έκφραση `foo a b c` είναι ισοδύναμη με την `foo('a','b','c')`
- Όμως, όταν κάνουμε χρήση της παραπάνω μορφής το MATLAB δεν μπορεί να επιστέψει ορίσματα εξόδου.

Συναρτήσεις και χαρακτήρες

- Π.χ η εντολή `legend apples oranges` τοποθετεί απλά λεζάντες στο παράθυρο γραφικών.
- Αν θέλουμε να χρησιμοποιήσουμε την συνάρτηση `legend` ώστε να επιστρέφει ορίσματα εξόδου θα πρέπει να την χρησιμοποιήσουμε με την συναρτησιακή μορφή της.
>> `[legh,objh] = legend('apples','oranges');`

Συναρτήσεις και χαρακτήρες

■ Χρήση χαρακτήρων ως ορίσματα εντολών.

```
for d = 1:31
```

```
    s = ['August' int2str(d) '.dat'];
```

```
    load(s)
```

```
% Επεξεργασία του d-αρχείου.
```

```
end
```

Ειδικές Συναρτήσεις

- Ένα σύνολο συναρτήσεων που ονομάζονται `function functions` μπορούν να χρησιμοποιηθούν για να επιδράσουν με συγκεκριμένο τρόπο σε προκαθορισμένες συναρτήσεις. Τυπικά παραδείγματα σε μη γραμμικές συναρτήσεις:
 - Εύρεση μηδενισμού
 - Βελτιστοποίηση
 - Ολοκλήρωση
 - Επίλυση συνήθων διαφορικών εξισώσεων

Ειδικές Συναρτήσεις

- Το MATLAB αναπαριστά τις μη γραμμικές συναρτήσεις με ένα αρχείο .m τύπου συνάρτησης.
- Αποθηκεύστε ένα αρχείο humps.m

```
function y = humps(x)
```

```
y = 1./((x-.3).^2 + .01) + 1./((x-.9).^2 + .04) - 6;
```

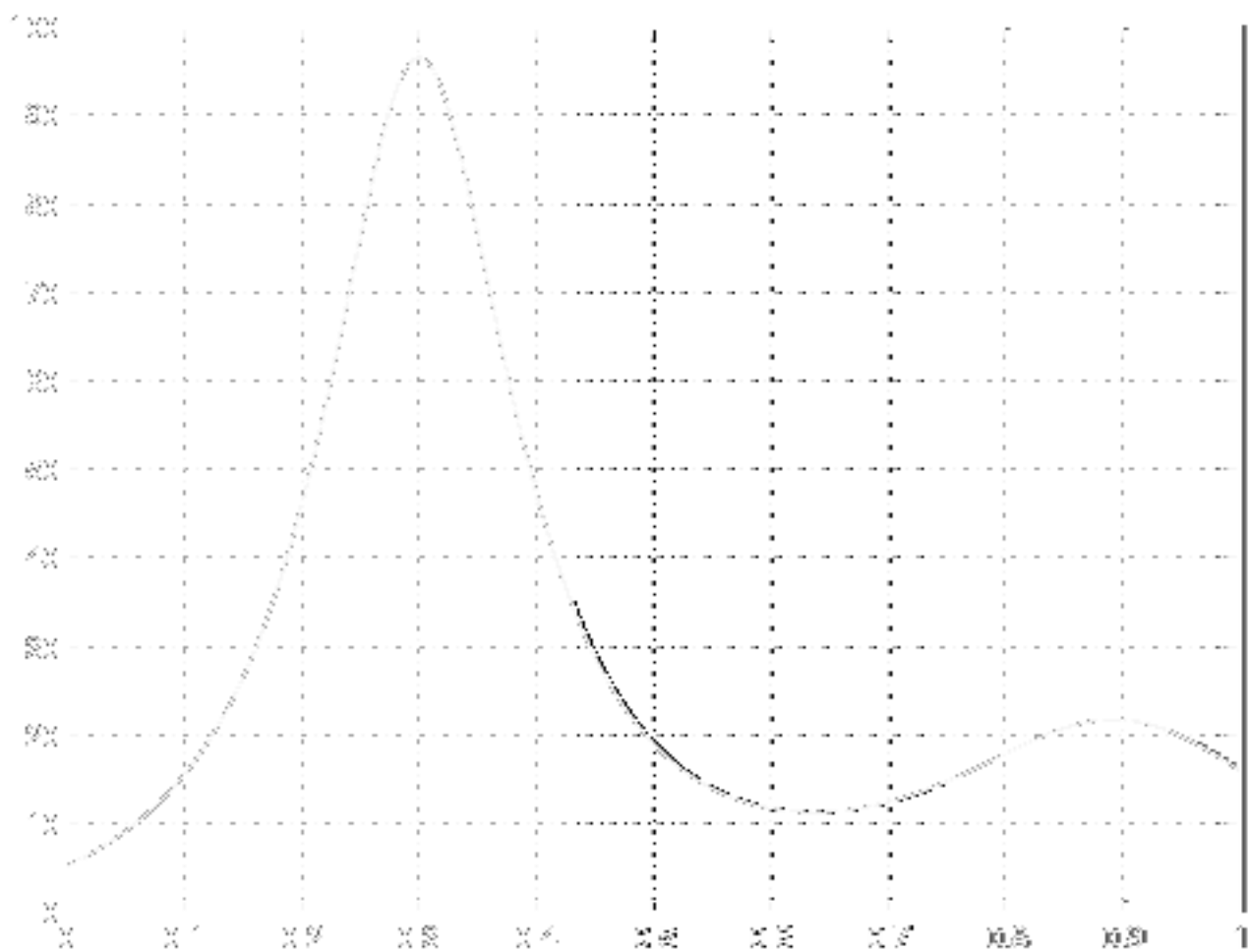
Ειδικές Συναρτήσεις

■ Υπολογίστε την συνάρτηση στο διάστημα $[0,1]$ και σχεδιάστε την γραφική παράσταση.

```
>> x = 0:.002:1;
```

```
>> y = humps(x);
```

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



Ειδικές Συναρτήσεις

- Ο χείριστος της συνάρτησης είναι ο χαρακτήρας @. Εκτιμάμε ότι το ελάχιστο είναι γύρω στο 0.5.

```
>> p = fminsearch(@humps,.5)
```

```
p =
```

```
0.6370
```

Ειδικές Συναρτήσεις

```
>> Q = quadl(@humps,0,1)
```

■ Υπολογίζει το ολοκλήρωμα της συνάρτησης μεταξύ στο διάστημα $[0,1]$

```
Q =
```

```
29.8583
```

Ειδικές Συναρτήσεις

```
z = fzero(@humps,.5)
```

■ Αναζητά μηδενισμό κάνοντας μια πρώτη εκτίμηση.

```
z =
```

```
-0.1316
```