# Εισαγωγή στους Υπολογιστές

Ενότητα #6: Προγραμματισμός στο MATLAB

Καθ. Δημήτρης Ματαράς
Πολυτεχνική Σχολή
Τμήμα Χημικών Μηχανικών

# Προγραμματισμός στο MATLAB

## Εντολή ελέγχου ροής – if

Γενική μορφή σύνταξης:

```
if   εντολη_ελέγχου_1
          σύνολο εντολών_1
elseif   εντολη_ελέγχου _2
         σύνολο εντολών_2
                 ……
elseif   εντολη_ελέγχου_n-1
         σύνολο εντολών_n-1
else
        σύνολο εντολών_n
end
```

### Παραδείγματα

```
if b>3
   a=33;
else
   a=21;
end
```

```
if x > 6
a = 2;
b = 3;
else
 a = 5;
 b = 6;
end
disp(a), disp(b)
```

### Παραδείγματα

Έστω η συνάρτηση f(x) για την οποία ισχύει:

$$f(x)=\begin{cases}\frac{1}{x} & , \quad -3\le x<-1\\ x^3 & , \quad -1\le x<1\\ 2x-1 & , \quad 1\le x\le 2\end{cases}$$

```
>> x=0;
>> if x<=-1
f=1/x;
elseif x >=-1 & x<1
f=x^3;
else
f=2*x-1;
end
```

# Προγραμματισμός στο MATLAB

## Επαναληπτικές διαδικασίες – βρόχοι (for, Φωλιασμένες for, while)

Γενική μορφή σύνταξης:

```
for δείκτης=λίστα
    σύνολο εντολών προς εκτέλεση
end
```

Οι επαναληπτικές διαδικασίες χρησιμοποιούνται στις περιπτώσεις που απαιτείται η επαναληπτική εκτέλεση μιας ή περισσοτέρων εντολών.

Παραδείγματα

## Παραδείγματα

- Να κατασκευαστεί μια λίστα που να περιέχει τα τετράγωνα των ακεραίων αριθμών 1-10.

```
x=[];                  (δημιουργία κενής λίστας)
for i=1:10             (δημιουργία βρόχου)
   x=[x,i^2];
end
```

- Να υπολογιστεί το άθροισμα των πρώτων εκατό ακεραίων αριθμών 1-100.

```
n=100;
s=0.;
for i=1:n
   s=s+i;
end
```

- Να βρεθεί ο **μεγαλύτερος αριθμός**, απ' αυτούς που είναι αποθηκευμένοι σε μια μονοδιάστατη λίστα, προσδιορίζοντας ταυτόχρονα και τη **θέση** του στη λίστα αυτή. Για την εφαρμογή του παραδείγματος, δημιουργούμε μια μονοδιάστατη λίστα εκατό αριθμών, αξιοποιώντας την εντολή 10*rand(1,100).

```
n=100;
a=10*rand(1,n);
amax=a(1);imax=1;
for i=2:n
   if a(i)>amax
      amax=a(i);
      imax=i;
   end
end
```

## Φωλιασμένες εντολές for

## Παραδείγματα

- Να κατασκευαστεί μια λίστα διαστάσεων 3×4 , τα στοιχεία της οποίας να δίνονται από τη σχέση c(i,j)=(i+j)/2

```
for i=1:3
   for j=1:4
      c(i,j)=(i+j)/2.;
   end
end
```

- Να υπολογιστεί το άθροισμα όλων των στοιχείων της τετραγωνικής λίστας που κατασκευάζεται με την εντολή magic(10).

```
n=10;
a=magic(n);
asum=0.;
for i=1:n
   for j=1:n
      asum=asum+a(i,j);
   end
end
```

## while

Μερικές φορές δεν είναι γνωστός, εκ των προτέρων, ο αριθμός των κύκλων εκτέλεσης ενός βρόχου, ο οποίος ενδέχεται να διαμορφώνεται κατά τη διάρκεια των κύκλων, όπως ενδέχεται και η τιμή του δείκτη για τον επόμενο κύκλο να υπολογίζεται μέσα στον ίδιο το βρόχο, ως μια συνάρτηση των ενδιάμεσων αποτελεσμάτων.

**Σύνταξη:**

αρχική_τιμή_δείκτη

while εντολη_ελέγχου_δείκτη

εντολές προς εκτέλεση

προσδιορισμός_νέας_τιμής_δείκτη

end

Όπως και στο βρόχο for, έτσι και στο βρόχο while, γίνεται επαναληπτική εκτέλεση των πράξεων, ***μέχρι*** να παραβιαστεί ο έλεγχος.

Ο while βρόγχος επαναλαμβάνει μια ακολουθία εντολών όσο μια συνθήκη είναι αληθής. Έστω ότι θέλουμε να βρούμε τον ελάχιστο ακέραιο n για τον οποίο ισχύει $2^n \geq 8$. Αυτό γίνεται μέσω του while βρόγχου με τον ακόλουθο τρόπο:

```
>> n=0
>> while 2^n < 8
n=n+1
end
>> n
3
```

## Παραδείγματα

Χρησιμοποιώντας την εντολή while, να ευρεθεί ο μεγαλύτερος αριθμός από αυτούς που είναι αποθηκευμένοι σε μια μονοδιάστατη λίστα, και να προσδιοριστεί ταυτόχρονα και η θέση του στη λίστα αυτή.

```
n=100;
a=10*rand(1,n);
amax=a(1);imax=1;
ni=1;
while ni<n+0.5
   if a(ni)>amax
      amax=a(ni);
      imax=ni;
   end
ni=ni+1;
end
disp('maximum number= '),disp(amax)
disp('position= '),disp(imax)
```

Θέση μεγίστου

μέγιστο

### Η λύση με for:

```
n=100;
a=10*rand(1,n);
amax=a(1);imax=1;
for i=2:n
   if a(i)>amax
      amax=a(i);
      imax=i;
   end
end
```

# Παραδείγματα – σύγκριση **for** και **while**

- Να υπολογιστεί το άθροισμα όλων των στοιχείων της τετραγωνικής λίστας που κατασκευάζεται με την εντολή magic(10).

(For)

```
n=10;
a=magic(n);
asum=0.;
for n1=1:n
   for n2=1:n
      asum=asum+a(n1,n2);
   end
end
```

(While)

```
n=10;
a=magic(n);
asum=0.;
n1=1;
while n1<n+0.5
   n2=1;
   while n2<n+0.5
      asum=asum+a(n1,n2);
      n2=n2+1;
   end
   n1=n1+1;
end
disp('list element summation= '),disp(asum)n=10;
a=magic(n);
```

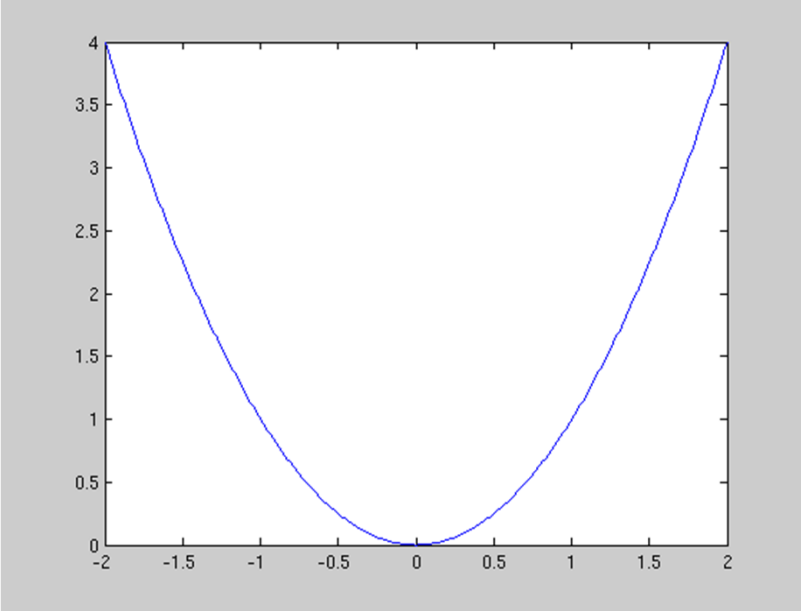
# Προγραμματισμός στο MATLAB

## Λογικοί τελεστές

| **Συσχετιστικοί τελεστές** | | **Λογικοί τελεστές** | |
|---|---|---|---|
| > | Μεγαλύτερο | & | Λογικό και |
| < | Μικρότερο | \| | Λογικό ή |
| >= | Μεγαλύτερο ή ίσο | ~ | Λογικό όχι |
| <= | Μικρότερο ή ίσο | | |
| == | ίσο | | |
| ~= | διάφορο | | |

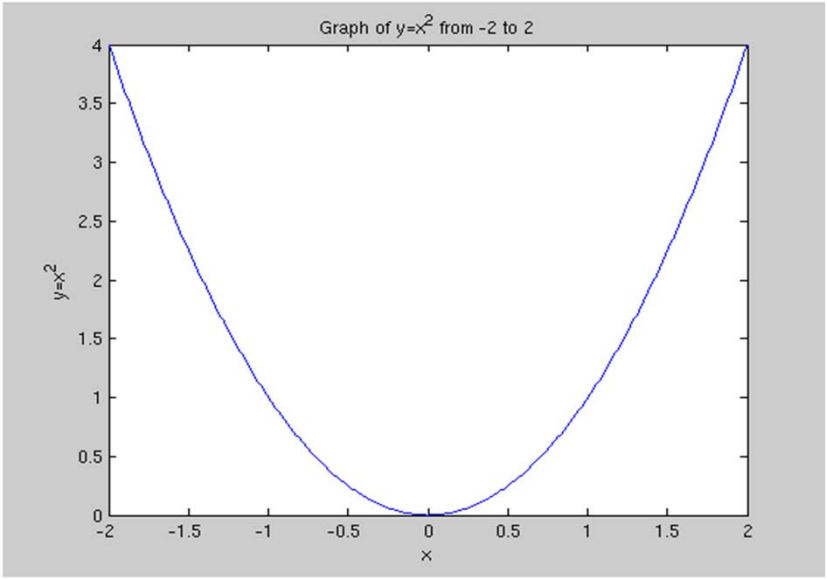
# Γραφικές παραστάσεις 2-D

```
>> x=-2:0.01:2;
>> y1=x.^2;

>> plot(x,y1)
```




Ονόματα αξόνων και τίτλος διαγράμματος

```
>> xlabel('x')
>> ylabel('y=x^2')

>> title ('Graph of y=x^2 from -2 to 2')
```



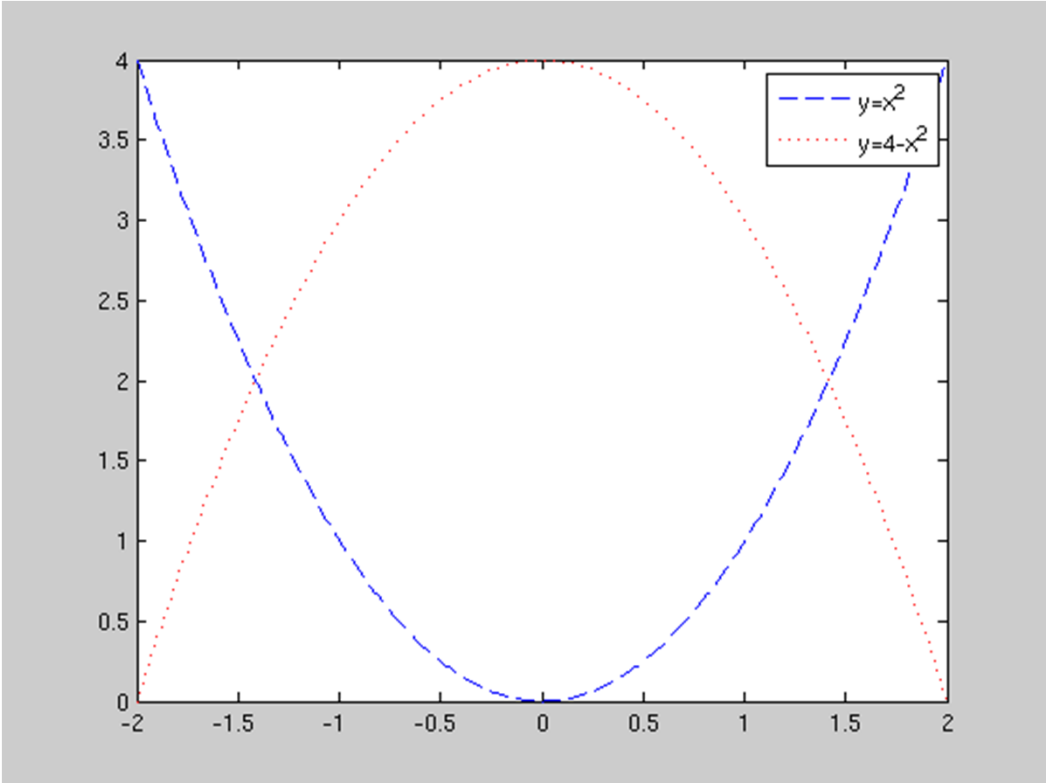
Το MATLAB παρέχει διάφορα **σύμβολα, τύπους γραμμών** και **χρώματα** για την μορφή των γραφικών παραστάσεων μερικά από τα οποία φαίνονται στο παρακάτω πίνακα.

>>plot (x,y, '[color] [stype] [ltype]')  γενική σύνταξη

| COLORS | | SYMBOLS | | LINES | |
|---|---|---|---|---|---|
| b | blue | • | point | - | solid |
| g | green | o | circle | … | dotted |
| r | red | x | x-mark | -· | dshdot |
| c | cyan | + | plus | -- | dashed |
| m | magenta | * | star | (none) | No line |
| y | yellow | s | square | | |
| k | black | d | diamond | | |
| w | white | > | triangle | | |

```
>> x=-2:0.01:2;
>> y1=x.^2;
>> plot(x,y1,'--go')
```
(καμπύλη με πράσινη διακεκομμένη γραμμή και με κύκλους στα σημεία)

Στην ίδια γραφική παράσταση μπορούμε να εμφανίσουμε περισσότερες από μία καμπύλες.
```
>> y2=4-x.^2;
```
μπορούμε να έχουμε τις δύο γραφικές παραστάσεις σε κοινό σύστημα αξόνων:
```
>> plot(x,y1,'--b',x,y2,':r')
```
διαφοροποιώντας τις καμπύλες από τον τύπο της γραμμής, το χρώμα της και βάζοντας λεζάντες στις καμπύλες:
```
>> legend('y=x^2','y=4-x^2')
```



# Λογαριθμικά & ημιλογαριθμικά διαγράμματα

Αν θέλουμε η γραφική παράσταση να γίνει σε **λογαριθμική κλίμακα** στον x-άξονα, στον y-άξονα ή και στους δύο άξονες τότε οι αντίστοιχες εντολές είναι:

- **semilogx**
- **semilogy**
- **loglog**

# Σύνταξη ezplot

>>ezplot('f')

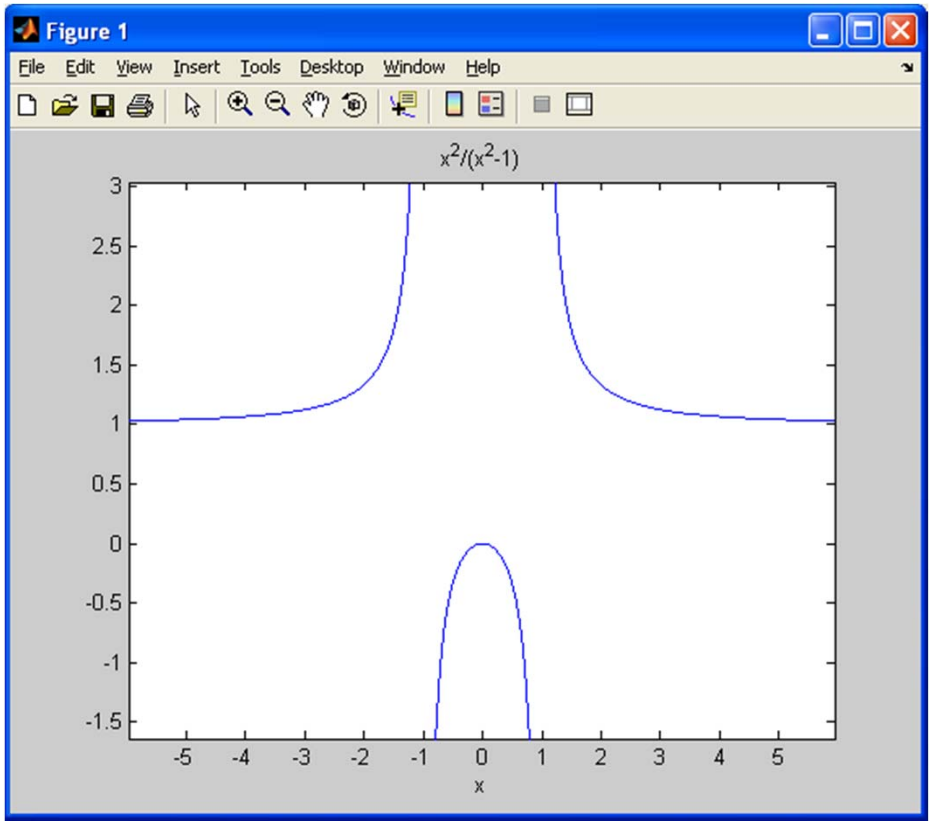
Παράγει το γράφημα της f επιλέγοντας τα διανύσματα των αξόνων των x και y

Y=f(x)

>>ezplot('f', a, b)

Παράγει το γράφημα της f ώστε οι τιμές του x να βρίσκονται στο διάστημα [a, b]

>>ezplot('x^2/(x^2-1)')



# Γραφικές παραστάσεις 3-D

Το MATLAB δίνει τη δυνατότητα αναπαράστασης δεδομένων στις τρεις διαστάσεις μέσω τον εντολών:

- ***surf***
- ***surfl***
- ***mesh***
- ***plot3***

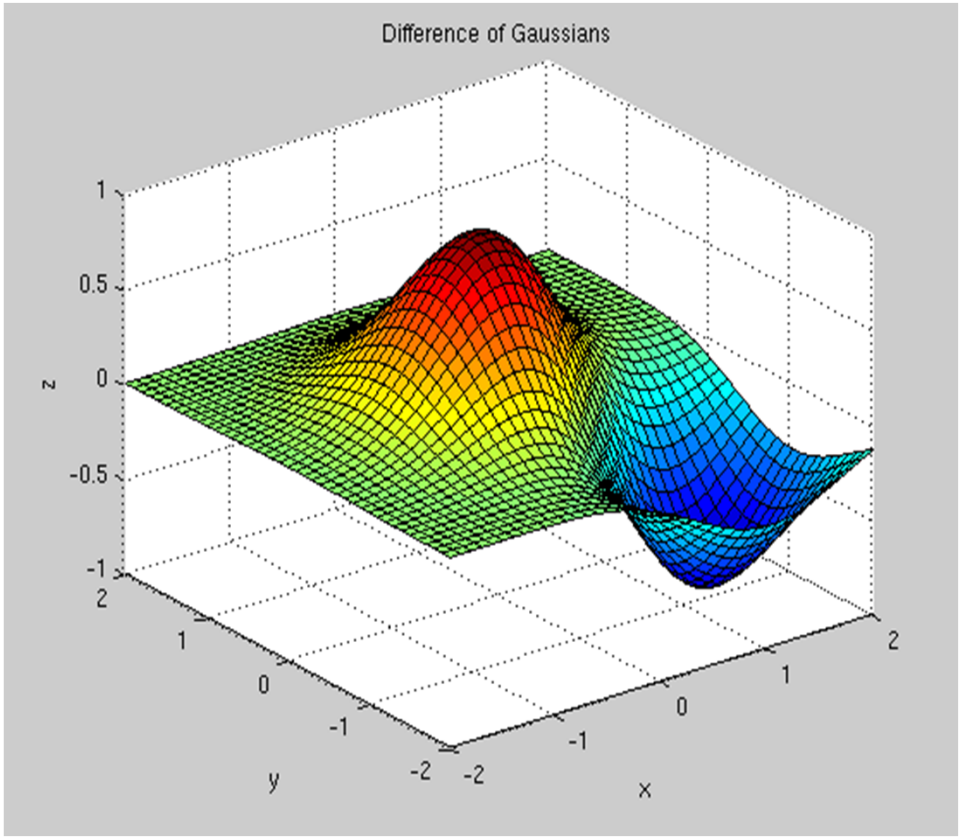
Αν χρειαζόμαστε βοήθεια για τη σύνταξη ανατρέχουμε στο help!

## Παράδειγμα

$f=e^{-(x2+y2)}- e^{-((x-1)2+(y+1)2))}$

η οποία είναι διαφορά δύο Γκαουσιανών συναρτήσεων με δύο μεταβλητές με x ∈ [-2,2] και y ∈ [-2,2]:

```
>> % create x vector
>> x=-2:0.1:2;
>> % create y vector
>> y=-2:0.1:2;
>> % transform x and y vectors int X and Y matrices
>> % with repeated rows and columns respectively
for use
>> % in evaluating functions of two variables
>> [X,Y]=meshgrid(x,y);
>> % define function
>> Z=exp(-(X.^2+Y.^2))-exp(-((X-1).^2+(Y+1).^2));
>> % now plot
>> surf(X,Y,Z)
>> % put axis labels
>> xlabel('x')
>> ylabel('y')
>> zlabel('z')
>> % put title
>> title('Difference of Gaussians')
```



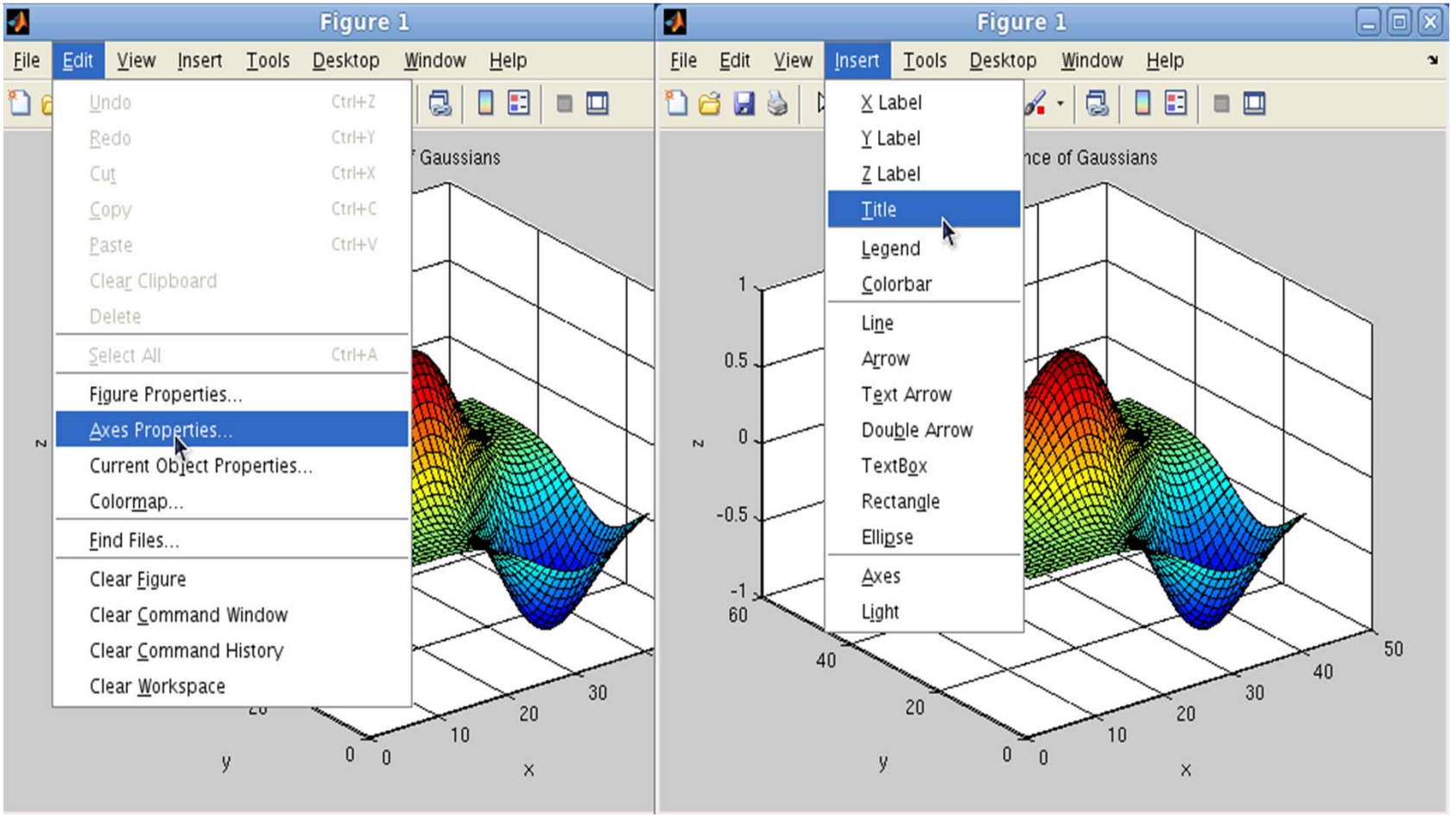


Figure 1
File Edit View Insert Tools Desktop Window Help
Undo Ctrl+Z
Redo Ctrl+Y
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Clear Clipboard
Delete
Select All Ctrl+A
Figure Properties...
Axes Properties...
Current Object Properties...
Colormap...
Find Files...
Clear Figure
Clear Command Window
Clear Command History
Clear Workspace
Gaussians
Figure 1
File Edit View Insert Tools Desktop Window Help
X Label
Y Label
Z Label
Title
Legend
Colorbar
Line
Arrow
Text Arrow
Double Arrow
TextBox
Rectangle
Ellipse
Axes
Light
nce of Gaussians

# Παράδειγμα συνδυασμού δομών επιλογής και επανάληψης

# Παράδειγμα 1

Θέλουμε να υπολογίσουμε το άθροισμα: $\sum_{i=1}^{10}(k+1)^k$

```
>> sum1=0;

>> for i=1:10
     sum1=sum1+(i+1)^i;
   end

>> sum1
sum1 =
  2.6983e+010
```

Μπορούμε εναλλακτικά να χρησιμοποιήσουμε τη συνάρτηση **sum** για διανύσματα αποφεύγοντας το βρόχο **for**:

```
>> sum1=sum(((1:10)+1).^(1:10))
sum1 =
  2.6983e+010
```

# Παράδειγμα 2

Έστω $x_i$, i = 1, …., 11 οι κόμβοι που προκύπτουν αν διαμερίσουμε το διάστημα [0, 1] σε 10 ισομήκη διαστήματα:

$X_i=0.1(i-1), \quad i=1,\ldots,11$

Θέλουμε να υπολογίσουμε το γινόμενο: $\prod_{i=1,\ i\neq 5}^{11} (x_5 - x_i)$

```
>> x=(0:10)/10;
>> product=1;

>> for i=1:4, product=product*(x(5)-x(i)); end
>> for i=6:11, product=product*(x(5)-x(i)); end

>> product
product =
  1.7280e-006
```

Παρατηρούμε ότι χρησιμοποιήσαμε **δύο βρόχους for** για να αποφύγουμε το i = 5. Όπως θα δούμε στη συνέχεια μπορούμε να εργαστούμε με ένα βρόχο και να αποφύγουμε το i = 5 με μια εντολή **if**.

Μπορούμε να αποφύγουμε την περίπτωση i = 5 με την εντολή if:

```
>> x=(0:10)/10;
>> product=1;
>> for i=1:11
     if i~=5
       product=product*(x(5)-x(i));
     end
   end

>> product
product =
  1.7280e-006
```

## Παράδειγμα 3

Η συνάρτηση $f(x) = e^x$ προσεγγίζεται με σειρά Taylor ως εξής: $f(x) = e^x \approx \sum_{n=0}^{N} \frac{x^n}{n!}$.

Μπορεί να δειχτεί ότι $f(x) = e^x = \lim_{N\to\infty} \sum_{n=0}^{N} \frac{x^n}{n!}$, άρα για κάποιο πεπερασμένο $N$, η πιο πάνω σειρά δίνει μια «καλή» προσέγγιση για την συνάρτηση. Το function m-file

exp1.m, που ακολουθεί, υλοποιεί αυτή την προσέγγιση για την $f(x)$ για κάποιο δεδομένο $x$ με ακρίβεια 0.0001.

```
function [S] = exp1(x)
% function [S] = exp1(x)
%
% Briskei mia proseggish gia thv suvarthsh exp(x) me
% akribeia 0.0001.

n = 1;
S = 1;
an = 1;

while abs(an) >= 0.0001
    an = x.^n/factorial(n);
    S = S + an;
    n = n + 1;
end
```

Εδώ χρησιμοποιήσαμε τη συνάρτηση βιβλιοθήκης **factorial(n)** για τον υπολογισμό του παραγοντικού n!.

## Παράδειγμα 4

Το script m-file workerpay.m, που φαίνεται πιο κάτω, υπολογίζει τον μισθό ενός εργαζομένου βάσει των ωρών που έχει εργαστεί. Αφού ζητήσει από τον χρήστη να δώσει τις ώρες εργασίας και το ωρομίσθιο, το πρόγραμμα υπολογίζει το μισθό ως εξής: Μέχρι τις 40 ώρες εργασίας, ο μισθός είναι το γινόμενο των ωρών επί το ωρομίσθιο, ενώ πάνω από τις 40 ώρες (υπερωρίες), οι απολαβές είναι κατά 50% μεγαλύτερες.

```
% Script file workerpay.m

t = input('Poses wres exei ergastei to atomo? ');
h = input('Poios eivai o mis0os ava wra? ');

Pay = t*h;

if t > 40
    Pay = Pay + (t-40)*0.5*h;
end


disp('To eisodeima tou ergazomevou eivai ')
format bank
disp(Pay)

% Telos tou workerpay.m
```

Ας τρέξουμε το πρόγραμμα μερικές φορές:

```
>> workerpay

Poses wres exei ergastei to atomo? 46
Poios eivai o mis0os ava wra? 4.25
To eisodeima tou ergazomevou eivai
        208.25

>> workerpay

Poses wres exei ergastei to atomo? 35
Poios eivai o mis0os ava wra? 4
To eisodeima tou ergazomevou eivai
        140.00
```

## Παράδειγμα 5

Το m-file gee.m, που φαίνεται πιο κάτω, ορίζει την συνάρτηση

$$g(x)=\begin{cases} x^2 & , x \le 0.5 \\ 0.25 & , x > 0.5 \end{cases}$$

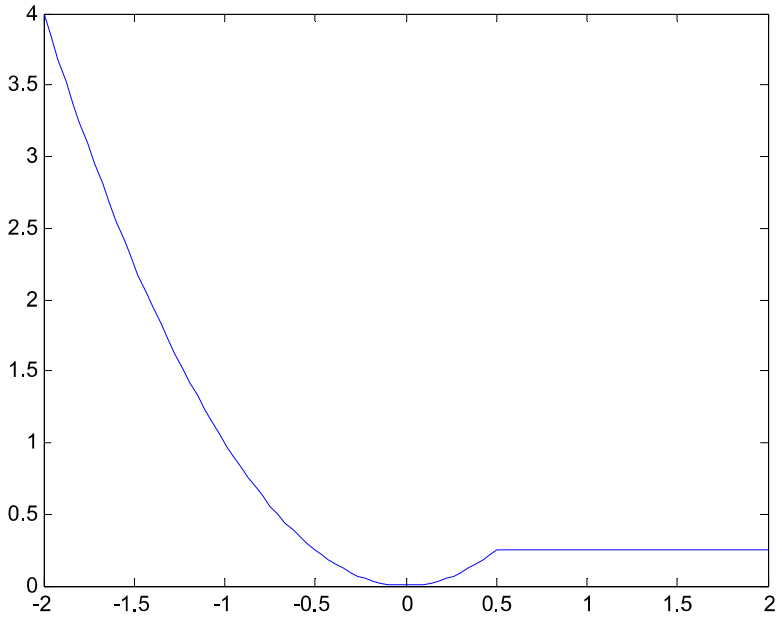
```
function [G] = gee(x)

for i=1:length(x)
    if x(i) <= 0.5
        G(i) = x(i)^2;
    else
        G(i) = 0.25;
    end
end

% Telos tou gee.m
```

Τρέχουμε το m-file και παίρνουμε τις εξής τιμές:

```
>> gee(0)
ans =
             0

>> gee(0.2)
ans =
          0.04

>> gee(3)
ans =
          0.25

>> gee([1:7]')
ans =
    0.2500    0.2500    0.2500    0.2500    0.2500    0.2500
0.2500

>> gee([-3:3])
ans =
    9.0000    4.0000    1.0000         0    0.2500    0.2500
0.2500
```

Μπορούμε, επίσης, να πάρουμε τη γραφική παράσταση της συνάρτησης $g$, π.χ. στο διάστημα [–2, 2], ως εξής:

```
>> x=linspace(-2,2);
>> plot(x,gee(x))
```



# Πολυώνυμα

## Συμβολισμός πολυωνύμων

Στη MATLAB τα πολυώνυμα αναπαριστώνται από διανύσματα που περιέχουν τους συντελεστές τους σε κατιούσα διάταξη.

Για παράδειγμα το πολυώνυμο $p(x)=x^2-3x+5$ αναπαριστάται από το διάνυσμα p = [1, -3, 5 ] ενώ το διάνυσμα q = [ 1, 0, 7, -1, 0 ] παριστάνει το πολυώνυμο $q(x)=x^4+7x^2-x$

## Άθροισμα πολυωνύμων - Γινόμενο πολυωνύμων

```
>> p=[ 1, -3, 5];
>> q=[ 1, 0, 7, -1, 0];
>> 3*p
ans =
    3    -9    15

>> -4*q
ans =
   -4     0   -28     4     0
```

```
>> s=[2, 3, -4];
>> s+p
ans =
     3     0     1
```

Για να προσθέσουμε **πολυώνυμα διαφορετικού βαθμού** που αναπαριστώνται από τα διανύσματα
p = [ 1, 2, 3, 4, 5] και q = [ -2, 0, 1]

γράφουμε το δεύτερο σαν
qnew = [ 0, 0, -2, 0, 1]

και υπολογίζουμε το άθροισμα p+qnew.

# Ρίζες πολυωνύμων - Υπολογισμός τιμών πολυωνύμου

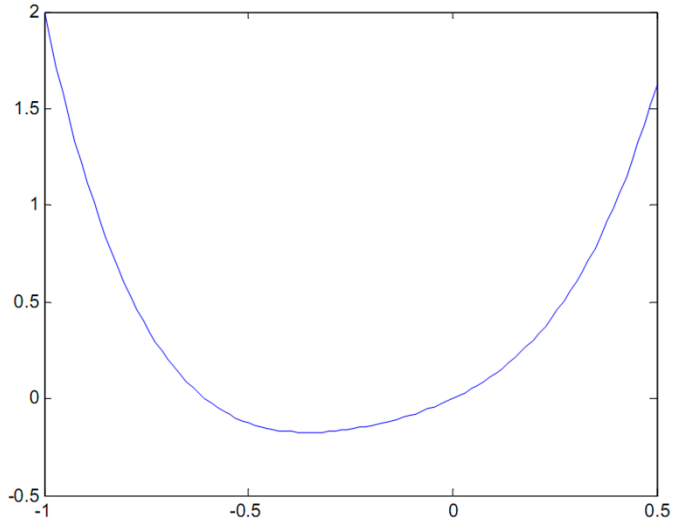
| | |
|---|---|
| **polyval** | Βρίσκει την τιμή ενός πολυωνύμου $p(x)$ σε ένα σημείο $x$.<br>Δομή:<br>polyval(p, x)<br>p: το διάνυσμα που αντιστοιχεί στο πολυώνυμο $p(x)$<br>x: το σημείο όπου υπολογίζεται το πολυώνυμο |
| **roots** | Βρίσκει τις ρίζες ενός πολυωνύμου $p(x)$.<br>Δομή:<br>roots(p)<br>p: το διάνυσμα που αντιστοιχεί στο πολυώνυμο $p(x)$ |

Θα βρούμε τις **τιμές του πολυωνύμου**
$P(x)=4x^4+3x^3+2x^2+x$
σε διάφορα σημεία:

```
>> p=[4, 3, 2, 1, 0];
>> polyval(p,0)

ans =
     0

>> polyval(p,1)

ans =
    10

>> polyval(p,-1)

ans =
     2
```

Άρα, για να πάρουμε τη **γραφική παράσταση** του p(x) στο διάστημα $[-1, \frac{1}{2}]$ γράφουμε

```
>> x=linspace(-1,0.5);
>> plot(x,polyval(p,x))
```




Μπορούμε επίσης να βρούμε τις **ρίζες του πολυωνύμου** με την εντολή roots:

```
>> roots(p)

ans =
         0
   -0.6058
   -0.0721 + 0.6383i
   -0.0721 - 0.6383i
```

# Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Πατρών**» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

# Σημείωμα Αναφοράς



Διαθέσιμο από τη δικτυακή διεύθυνση:
https://eclass.upatras.gr/courses/CMNG2112/

# Σημείωμα Αδειοδότησης




[1] http://creativecommons.org/licenses/by-nc-sa/4.0/

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.