



# Πληροφορική

Ενότητα 8: A. Fibonacci Revisited.  
B. Επανάληψη Α' κύκλου.  
Γ. Γραφικά στο Matlab.

Κωνσταντίνος Καρατζάς  
Τμήμα Μηχανολόγων Μηχανικών



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

# Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



# Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης» 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.





# FIBONACCI REVISITED!

28.04.2015

Αναπλ. Καθηγητής Κωνσταντίνος Καρατζάς  
Τμήμα Μηχανολόγων Μηχανικών ΑΠΘ

# Fibonacci

---

$$f(n) = \begin{cases} 0, & n=0 \\ 1, & n=1 \\ f(n-1) + f(n-2), & n>1 \end{cases}$$

---

# 1<sup>η</sup> υλοποίηση (αναδρομική)

```
function fn = fibonacci1(n)

if (n==0)
    fn = 0;
elseif (n==1)
    fn = 1;
else
    fn = fibonacci1(n-2)+fibonacci1(n-1);
end
```

---

# 2<sup>η</sup> υλοποίηση

```
function fn = fibonacci2(n)
```

```
fn(0) = 0;
```

```
fn(1) = 1;
```

```
for i=2:n
```

```
    fn(i) = fn(i-1) + fn(i-2);
```

```
end
```

---

# 3<sup>η</sup> υλοποίηση

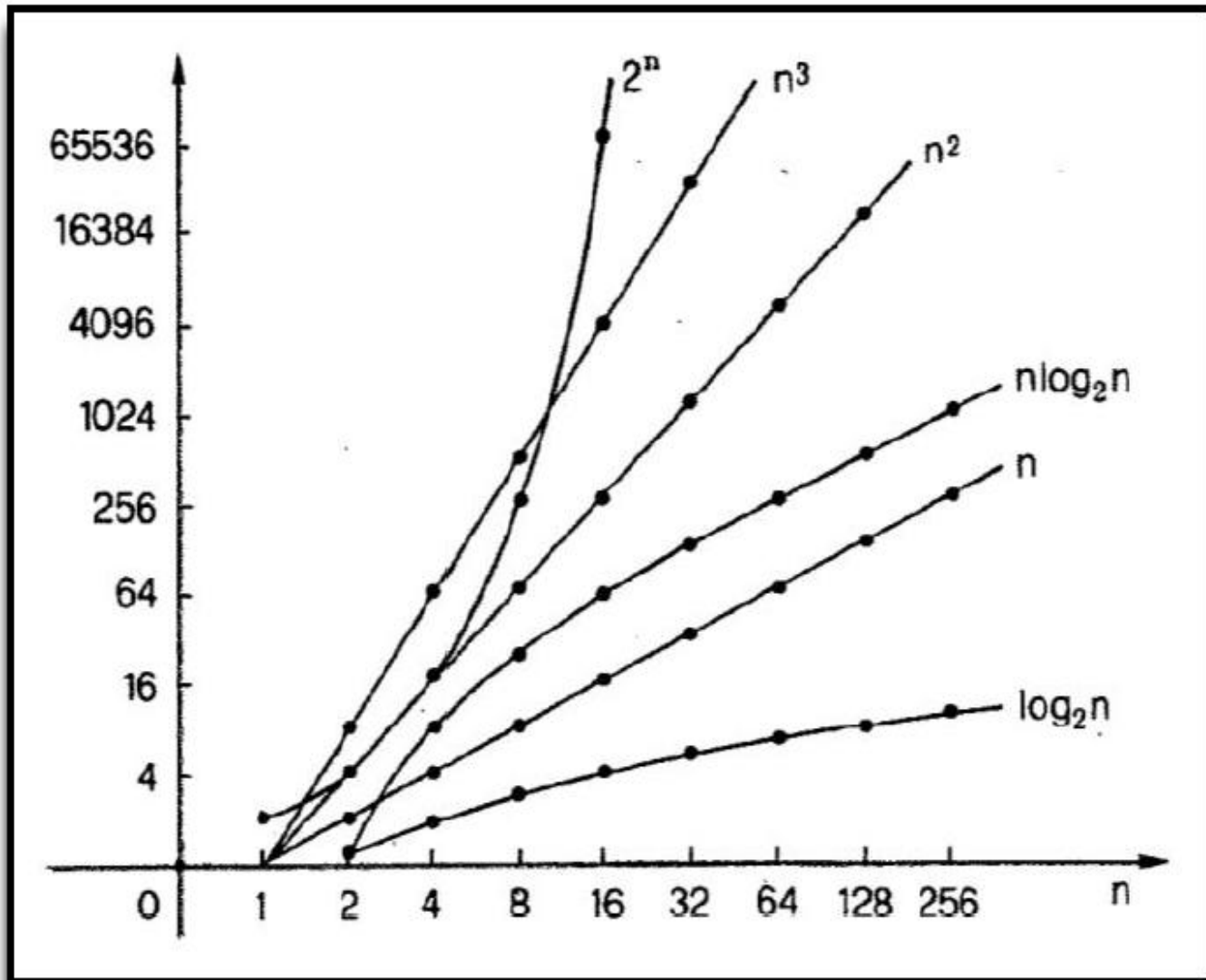
$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n = \begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix}.$$

- function fn = fibonacci4(n)
  - M = [1 1; 1 0];
  - F = M^n;
  - fn = F(1,2);
-



# 3<sup>η</sup> υλοποίηση

- Το “μυστικό” βρίσκεται στην ύψωση σε δύναμη:
    - ▣ Μπορεί από πράξη γραμμικής πολυπλοκότητας να γίνει πράξη  $O(\log_2 n)$
-





# Δ.#ΧΧ: ΠΛΗΡΟΦΟΡΙΚΗ ΕΠΑΝΑΛΗΨΗ Α' ΚΥΚΛΟΥ

Κωνσταντίνος Καρατζάς  
Τμήμα Μηχανολόγων Μηχανικών ΑΠΘ

Ας θυμηθούμε.....



---

# Επαναληπτικό εργαστήριο Α κύκλου

Ασκήσεις επανάληψης με στόχο την αυτόνομη εξάσκηση και τον εντοπισμό «αδυναμιών» κλπ.

---

# 1η άσκηση

- Αναπτύξτε μία συνάρτηση η οποία θα υπολογίζει το μήκος της τρίτης πλευράς ενός τριγώνου ( $c$ ) χρησιμοποιώντας τον νόμο των συνημιτόνων:  $c^2 = a^2 + b^2 - 2.a.b.\cos(t)$ 
    - Δεν παραδόθηκε συνάρτηση (λείπει “function” και “όνομα”)
    - Η συνάρτηση ζητά input
    - Η συνάρτηση περιλαμβάνει τις τιμές εισόδου (hard coding)
    - Όνομα αρχείου  $\neq$  όνομα συνάρτησης
    - Η συνάρτηση δεν έχει μεταβλητή εξόδου
    - Η ανάθεση τιμής γίνεται στο όνομα συνάρτησης και όχι στη μεταβλητή εξόδου
    - Δεν χρησιμοποιείται *end* (προαιρετικό)
-

# Πως δημιουργούμε μία συνάρτηση;

```
function [out1 out2 ...] = name(in1, in2, ..)
```

```
out1 = ...
```

```
out2 = ...
```

```
Εναλλακτικά πράξεις...
```

```
end
```

---

# Συνάρτηση χωρίς ορίσματα εισόδου-εξόδου

```
function doki
```

```
%Summary of this function goes here
```

```
%Detailed explanation goes here
```

```
i=5
```

```
end
```

---



# 2η άσκηση

- Υπολογίστε τον  $n$ -οστό όρο μίας ακολουθίας τιμών κάθε στοιχείο της οποίας είναι άθροισμα των δύο προηγούμενων όρων. Οι δύο πρώτοι όροι της ακολουθίας είναι  $x_1 = 0, x_2 = 1$ .
    - ▣ Fibonacci
    - ▣ Οι περισσότερες λύσεις “συμβατικές”
    - ▣ Αρκετές λύσεις με συνάρτηση
      - $N$ =διαβάζεται ως input εντός συνάρτησης!
      - $N$ : hard coded σε συνάρτηση ( $i=1:30$ )
-

# 2η άσκηση

```
function [x]= code (n)
x (1)=0 ;
x (2)=1 ;
for i=3:n
a (i)=a (i-1)+a (i-2) ;
end
x=a (n) ;
end
```

Τρέχει?

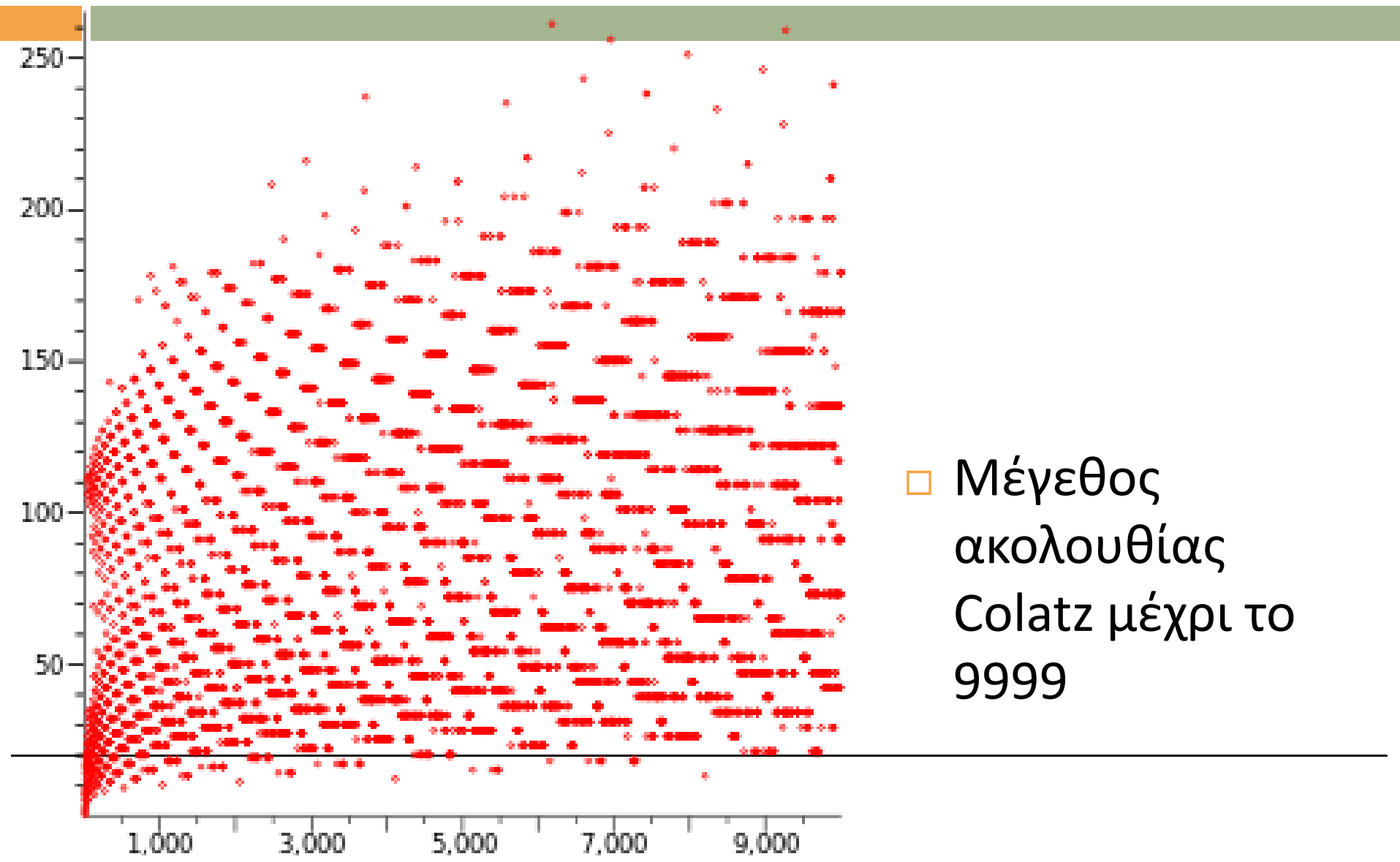
---

# 3η άσκηση

- Εικασία Collatz
  - Για κάθε ακέραιο  $n$
  - Αν άρτιος
    - $n = n/2$
  - Αν περιττός
    - $n = 3 * n + 1$
-

```
function [count]=collatz(n)
count = 0;
while n > 1
count=count+1;
    if rem(n,2)==0
        n = n/2;
    else
        n = 3*n+1;
    end
end
end
```

# Collatz εικασία



# 4η άσκηση

- Πίνακας 648 X 509 με τιμές από 1 έως 128
    - ▣ Μέση τιμή στρογγυλοποιημένη στον εγγύτερο ακέραιο
      - Εγγύτερος = ROUND
    - ▣ κάθε στοιχείο του πίνακα με τιμή μεγαλύτερη από το μέσο όρο ίσο με την μέγιστη τιμή που συναντάμε στον πίνακα, και κάθε στοιχείο του πίνακα με τιμή μικρότερη ή ίση του μέσου όρου, ίσο με την ελάχιστη τιμή που συναντάμε στον πίνακα
-

# 4η άσκηση

- Πολλές/πολλοί δεν πρόλαβαν
  - ~40% των απαντήσεων “μή διανυσματικές”
    - Ασκηθείτε με τις προτεινόμενες δραστηριότητες!
  - 5-10% με πολλά προβλήματα
    - Δείτε όλες τις ασκήσεις από την αρχή και ελάτε να μιλήσετε μαζί μας
-

# 4η άσκηση: the short way

- $mo = \text{round}(\text{mean}(X(:)))$   
 $X(X > mo) = \text{max}(X(:))$   
 $X(X \leq mo) = \text{min}(X(:))$
  - Διότι  $X(:)$  σημαίνει ΟΛΑ τα στοιχεία του  $X$  σε μία στήλη!
-



# 4η άσκηση: the long way

```
s=0
max=X(1,1); min=X(1,1)
for i=1:1:648
  for j=1:1:509
    s=s+X(i,j)
    if X(i,j)>max
      max=X(i,j)
    end
    if X(i,j)<min
      min=X(i,j)
    end
  end
end
end
m=s/(648*509)
if mod(m,1)>=0.5
  m=m-mod(m,1)+1
else
  m=m-mod(m,1)
end
for i=1:1:648
  for j=1:1:509
    if X(i,j)>m
      X(i,j)=max
    else
      X(i,j)=min
    end
  end
end
end
```

# Σταχυολογώντας προβλήματα 1ου θέματος

- Ορθή χρήση παρενθέσεων
  - Αδυναμία διανυσματικής επίλυσης ( $b=10.^{[0:0.1:1]}$ ), hard coding του περιεχομένου του πίνακα  $f$
  - Απουσία μεταβλητής εξόδου συνάρτησης
  - Στην 7η άσκηση (κύβος με συντεταγμένες που δεν πρέπει να είναι πρώτοι αριθμοί), ελέγχονται και οι 3 συντεταγμένες, κρατώντας ως αποτέλεσμα το προϊόν του τελευταίου ελέγχου. Έτσι, εάν η τελευταία συντεταγμένη που ελέγχθηκε δεν είναι πρώτος αριθμός, τότε ο κύβος χαρακτηρίζεται ως ασφαλής ανεξάρτητα από τις τιμές των προηγούμενων συντεταγμένων
-



Τέλος

Αναπληρωτής Καθ. Κωνσταντίνος Καρατζάς  
Τμήμα Μηχανολόγων Μηχανικών ΑΠΘ

# ΠΛΗΡΟΦΟΡΙΚΗ : ΓΡΑΦΙΚΑ ΣΤΟ MATLAB

28 Απριλίου 2015

Κώστας Καρατζάς  
Τμήμα Μηχανολόγων Μηχανικών, ΑΠΘ

# Γιατί γραφικά?

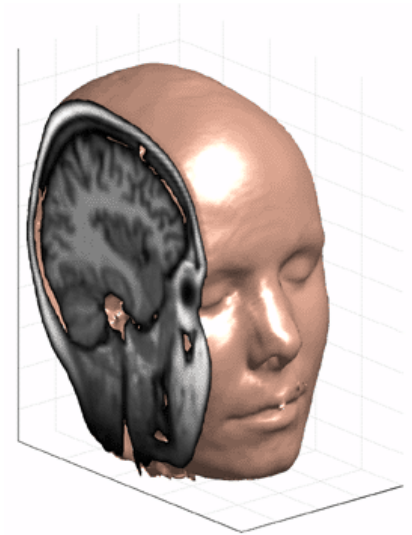
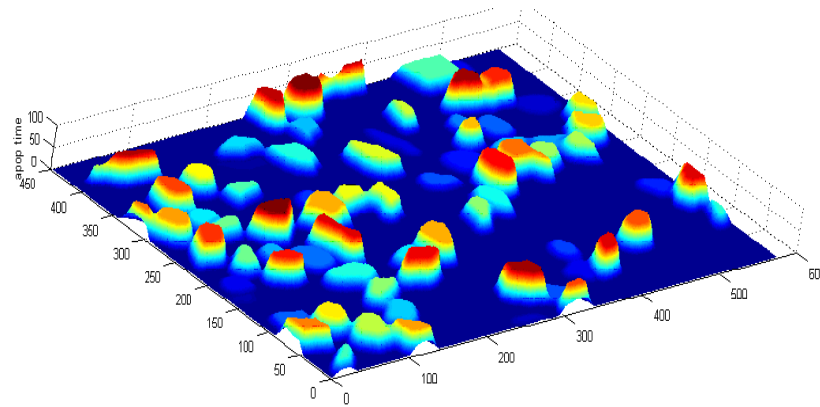
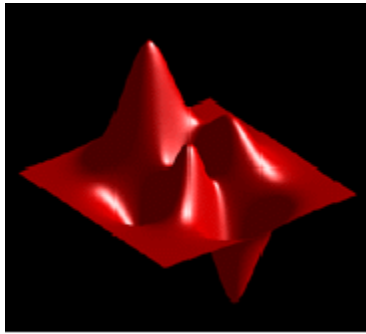
---

- Παρουσίαση δεδομένων
  - Διερεύνηση αποτελεσμάτων (πειραμάτων, υπολογισμών κλπ)
-

# Γραφικά στο Matlab

- ❑ Δισδιάστατα Γραφικά
  - ❑ Τρισδιάστατα Γραφικά
  - ❑ Άλλα Γραφικά
  - ❑ Γραφικές Παραστάσεις Ανώνυμων και Εσωτερικών συναρτήσεων
  - ❑ Movie
-

# Γραφικά....



□ Παραδείγματα

# Γιατί στο Matlab?

Matlab  
visualization



Excel  
visualization





# Η εντολή plot

Η κύρια εντολή κατασκευής γραφικών παραστάσεων στο Matlab είναι η εντολή plot. Η σύνταξή της έχει ως εξής:

***plot(X1,Y1,LineSpec,...)***

X1: διάνυσμα που θα απεικονιστεί στον άξονα-x

Y1: διάνυσμα που θα απεικονιστεί στον άξονα-y

LineSpec: Συμβολοσειρά που καθορίζει τον τύπο της καμπύλης

---

# Παράδειγμα

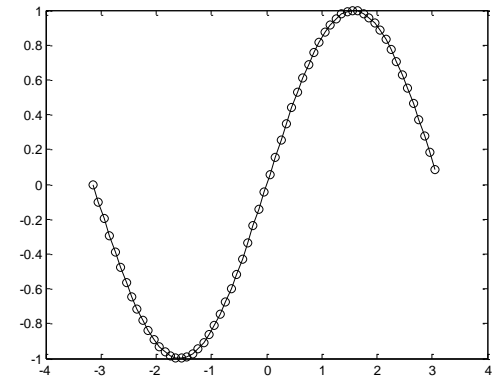
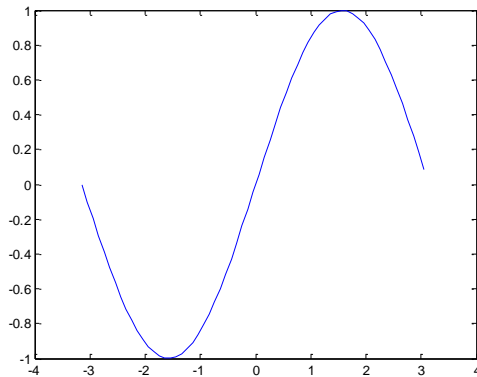
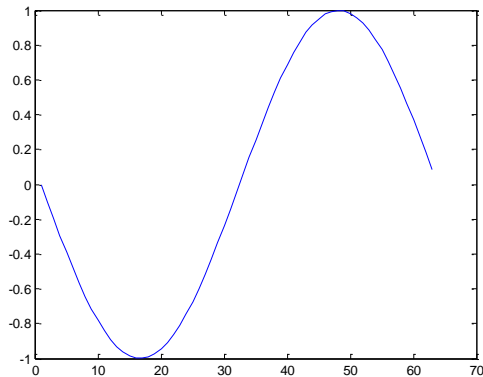
```
>> x=-pi:0.1:pi;
```

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

```
>> plot(y) % Ο άξονας x α/α
```

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

```
>> plot(x,y,'k-o') % αλλαγή του τύπου της καμπύλης
```



# Μορφοποίηση Καμπυλών με την εντολή plot

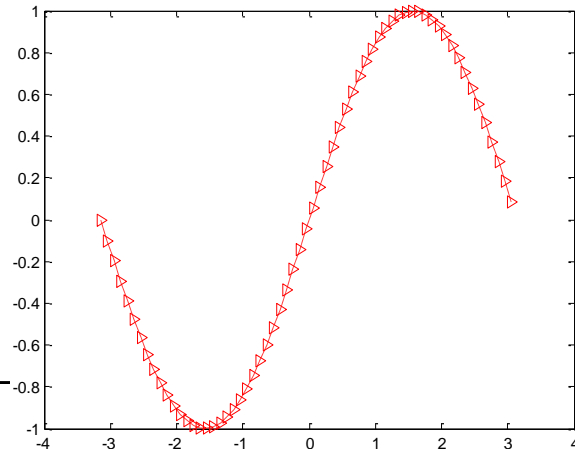
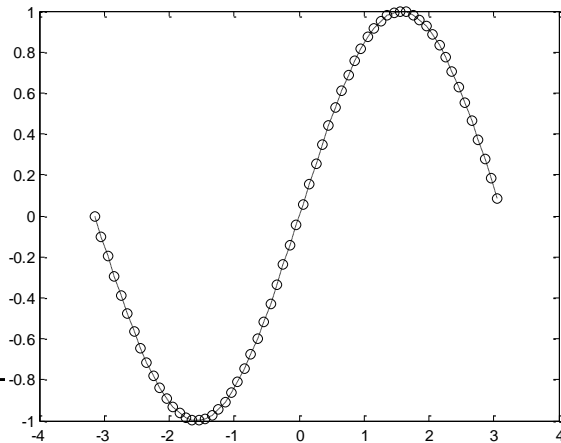
Συνδυάζοντας τους ακόλουθους χαρακτήρες μπορούμε να μορφοποιήσουμε το χρώμα τον τύπο και το σύμβολο μίας καμπύλης.

Χρώμα Γραμμής		Σύμβολο Σχεδίασης		Τύπος Γραμμής	
Χαρακτήρας	Προκύπτει	Χαρακτήρας	Προκύπτει	Χαρακτήρας	Προκύπτει
<b>y</b>	κίτρινο	<b>.</b>	•	<b>-</b>	Συμπαγής
<b>m</b>	ματζέντα	<b>o</b>	◦	<b>:</b>	Στικτή
<b>c</b>	κυανό	<b>x</b>	×	<b>-. </b>	Στικτή – διακεκομμένη
<b>r</b>	κόκκινο	<b>+</b>	+	<b>--</b>	Διακεκομμένη
<b>g</b>	πράσινο	<b>*</b>	*		
<b>b</b>	μπλε	<b>s</b>	□		
<b>w</b>	άσπρο	<b>d</b>	◇		
<b>k</b>	μαύρο	<b>v</b>	▽		
		<b>^</b>	△		
		<b>&lt;</b>	◁		
		<b>&gt;</b>	▷		
		<b>p</b>	★		
		<b>h</b>	εξάγραμμο		

# Παράδειγμα

```
>> x=-pi:0.1:pi;  
>> y=sin(x);  
>> figure(1); plot(x,y,'k-.0')  
>> figure(2); plot(x,y,'r-->')
```

Με την εντολή Figure αριθμούμε τα διαγράμματα, ώστε να «ανοίγουν» σε διαφορετικά παράθυρα. Διαφορετικά, σχεδιάζονται στο ίδιο παράθυρο και χάνονται προηγούμενα σχήματα.



# Πολλαπλά γραφήματα στο ίδιο σχήμα

Υπάρχουν δύο τρόποι για να κατασκευάσουμε πολλά γραφήματα στο ίδιο διάγραμμα.

**1<sup>ος</sup>** `plot(x1,y1,Linspec1,x2,y2,Linspec2,...)`

**2<sup>ος</sup>** `hold on`  
`plot(x1, y1, Linspec1)`  
`plot(x2, y2, Linspec2)`  
`...`  
`hold off`

---

# Παράδειγμα

```
>> x=-pi:0.1:pi;
```

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

```
>> z=cos(x);
```

```
>> plot(x,y,'k--',x,z,'r<')
```

```
% ισοδύναμο με την προηγούμενη
```

```
% εντολή. Κατάλληλο για δομές
```

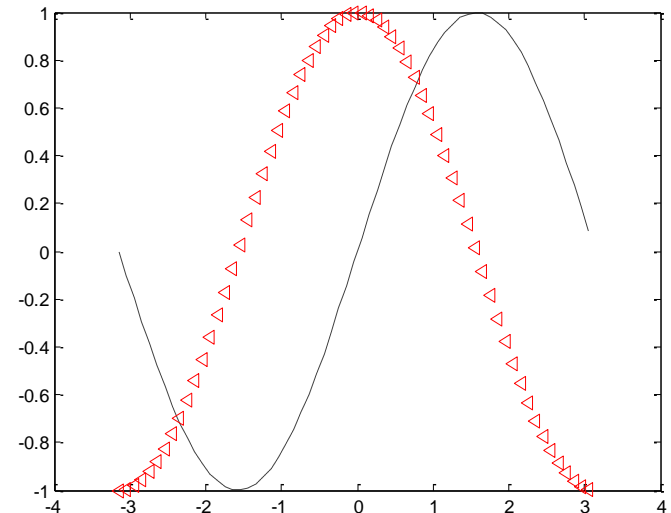
```
% επανάληψης
```

```
>> hold on
```

```
>> plot(x,y,'k--')
```

```
>> plot(x,z,'r<')
```

```
>> hold off
```



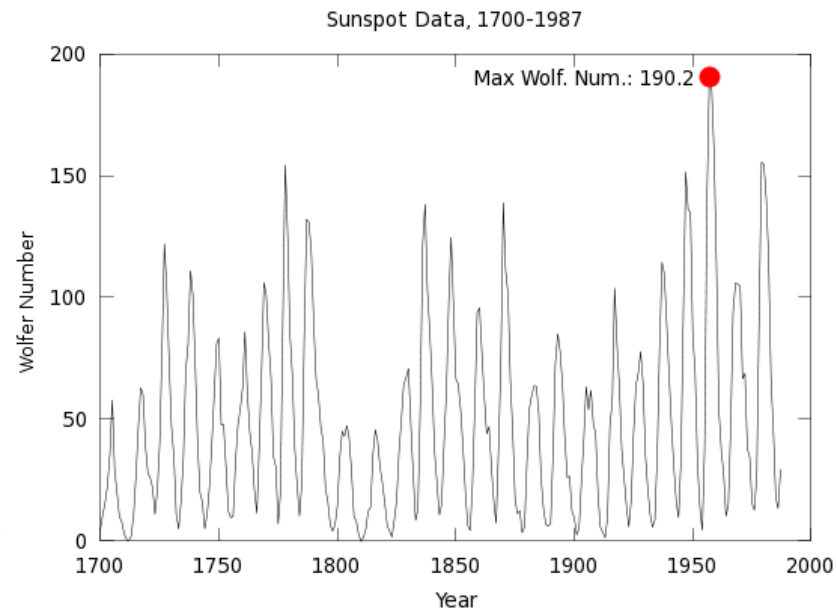
# Μορφοποίηση και διαχείριση των σχημάτων

Το Matlab διαθέτει μία σειρά από εντολές με τις οποίες μπορούμε να μορφοποιήσουμε τις γραφικές παραστάσεις.

Εντολή	Περιγραφή
<b>subplot(nrows,ncols,curplot)</b>	Χωρίζει το παράθυρο σε <i>nrows</i> γραμμές και <i>ncols</i> στήλες (για <i>nrows</i> x <i>ncols</i> σχήματα). Το <i>curplot</i> προσδιορίζει το που θα εμφανιστεί το τρέχον σχήμα.
<b>grid on/off</b>	Προσθέτει (on) ή αφαιρεί (off) πλέγμα από οριζόντιες και κάθετες γραμμές στο σχήμα.
<b>box on/off</b>	Προσθέτει (on) ή αφαιρεί (off) την πάνω και την δεξιά οριακή γραμμή στο γράφημα.
<b>axis([xmin xmax ymin ymax])</b>	Εμφάνιση των αξόνων εντός των: οριζόντιος άξονας= <i>xmin</i> ως <i>xmax</i> , κατακόρυφος άξονας= <i>ymin</i> ως <i>ymax</i> .
<b>axis on/off/auto/normal/ square/equal/tight/</b>	Τυποποιημένες ρυθμίσεις των αξόνων.
<b>title('text')</b>	Βάζει ως τίτλο του γραφήματος (στην κορυφή του) το κείμενο που βρίσκεται μέσα στα εισαγωγικά.
<b>xlabel('text')</b>	Βάζει ως ετικέτα του οριζόντιου (x) άξονα το κείμενο που βρίσκεται μέσα στα εισαγωγικά.
<b>ylabel('text')</b>	Βάζει ως ετικέτα του κατακόρυφου (y) άξονα το κείμενο που βρίσκεται μέσα στα εισαγωγικά.
<b>legend('text')</b>	Βάζει ως λεζάντα του γραφήματος (δεξιά κορυφή) το κείμενο που βρίσκεται μέσα στα εισαγωγικά.
<b>text(x,y,'text')</b>	Βάζει το κείμενο που βρίσκεται μέσα στα εισαγωγικά στη θέση (x,y) του γραφήματος.
<b>gtext('text')</b>	Βάζει το κείμενο που βρίσκεται μέσα στα εισαγωγικά στη θέση του γραφήματος, όπου θα κάνουμε αριστερό κλικ με το ποντίκι.

# Παράδειγμα: Μορφοποίηση Σχήματος

```
load sunspot.dat
plot(sunspot(:,1), sunspot(:,2), 'k')
xlabel('Year')
ylabel('Wolfer Number')
title('Sunspot Data, 1700-1987')
[maxval maxpos]=max(sunspot(:,2));
hold on
plot(sunspot(maxpos,1),maxval,'r.',...
... 'MarkerSize',20);
text(sunspot(maxpos,1)-100,maxval,...
... ['Max Wolf. Num.: ',num2str(maxval)])
```





# Παράδειγμα: Μορφοποίηση Σχήματος

```
load sunspot.dat           % γραφική παράσταση των δύο μεταβλητών
plot(sunspot(:,1), sunspot(:,2), 'k') % μαύρη γραμμή

xlabel('Year')             % Κείμενο στους άξονες
ylabel('Wolfer Number')
title('Sunspot Data, 1700-1987')

[maxval maxpos]=max(sunspot(:,2)); % Υπολογισμός θέσης και τιμής
                                   % του μεγίστου

hold on % επισήμανση του μεγίστου με κόκκινη τελεία (μέγεθος 20)
plot(sunspot(maxpos,1), maxval, 'r.', 'MarkerSize', 20);

text(sunspot(maxpos,1)-100, maxval, % προσθήκη κειμένου
...['Max Wolf. Num.: ', num2str(maxval)]);
hold off
```

# Επιμερισμός σχήματος σε μικρότερα σχήματα

Με την εντολή `subplot` μπορούμε να επιμερίσουμε το τρέχον παράθυρο του Matlab, σε έναν πίνακα από μικρότερα παράθυρα και να προσθέσουμε περισσότερα από ένα σχήματα στα κελιά που δημιουργήσαμε. Η σύνταξη της εντολής `subplot` είναι:

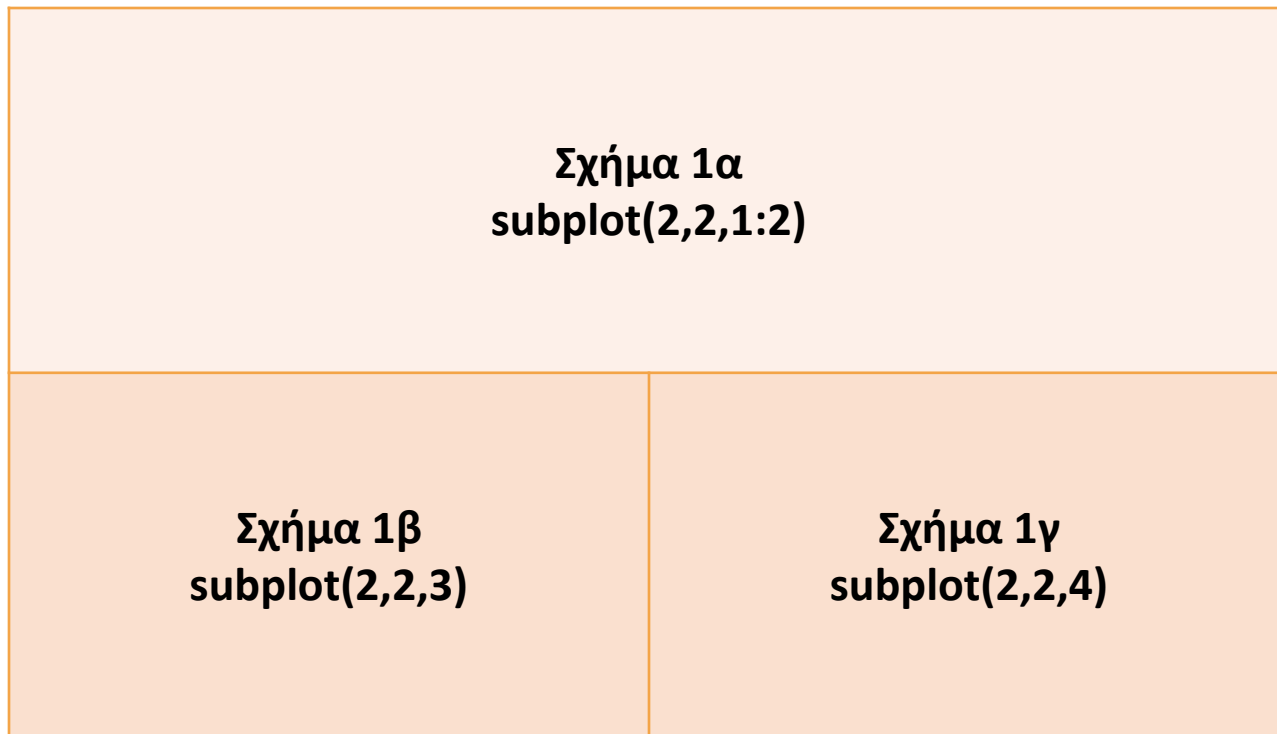
***subplot(m, n, p)***

όπου *m*: ο αριθμός των γραμμών, *n*: ο αριθμός των στηλών του πίνακα και *p* το τρέχον κελί ή κελιά.

---

# Παράδειγμα: Κατασκευή πολλών διαγραμμάτων στο ίδιο παράθυρο

Έστω ότι θέλουμε να δημιουργήσουμε ένα σχήμα όπως το παρακάτω.



# Παράδειγμα: Κατασκευή πολλών διαγραμμάτων στο ίδιο παράθυρο

```
x=-pi:0.1:pi;
```

```
subplot(2,2,1:2)
```

```
plot(x,sin(x),'ko',x,cos(x),'rx');
```

```
title('Γραφικές Παραστάσεις sin(...), cos(...)')
```

```
xlabel('Γωνία [rad]');
```

```
ylabel('Ημίτονο & Συνημίτονο');
```

```
legend('Ημίτονο','Συνημίτονο','Location','NorthWest');
```

```
axis([-3.5 3.5 -1 1]);
```

```
subplot(2,2,3)
```

```
hold on
```

```
plot(x,sin(x),'ko');
```

```
plot(x,cos(x),'rx');
```

```
hold off
```

```
grid on; box off
```

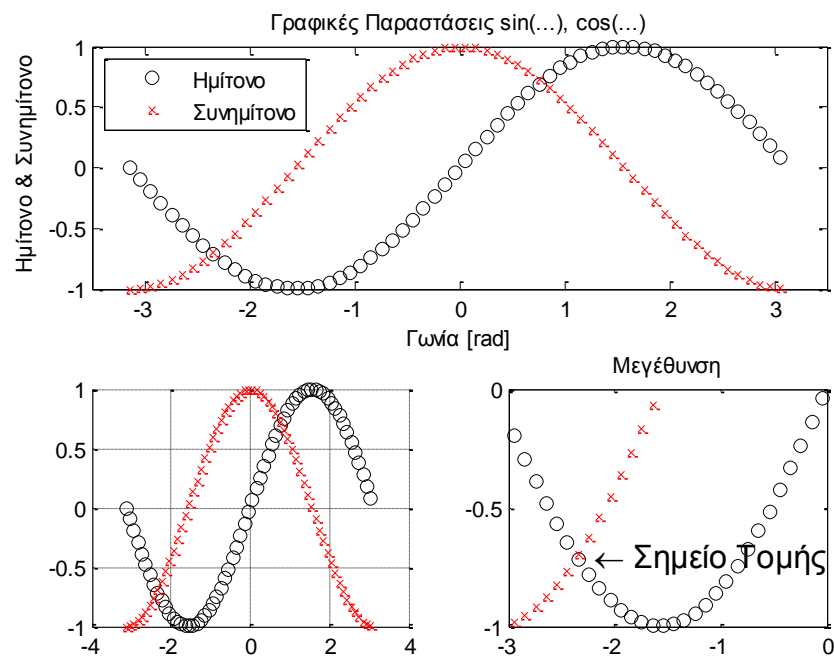
```
subplot(2,2,4)
```

```
plot(x,sin(x),'ko',x,cos(x),'rx');
```

```
axis([-3 0 -1 0]);
```

```
title('Μεγέθυνση')
```

```
text(-2.2,-0.7,'\leftarrow Σημείο Τομής','FontSize',14)
```



# Λογαριθμικά και Ημιλογαριθμικά Διαγράμματα

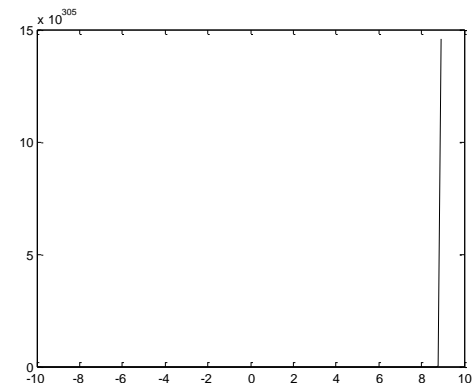
Τα λογαριθμικά και ημιλογαριθμικά διαγράμματα είναι χρήσιμα σε περιπτώσεις που οι τιμές μίας μεταβλητής παρουσιάζουν μεγάλο εύρος. Π.χ. η συνάρτηση  $f(x) = \exp(x^3)$ :

```
>> x=-10:0.1:10;
```

```
>> y=exp(x.^3);
```

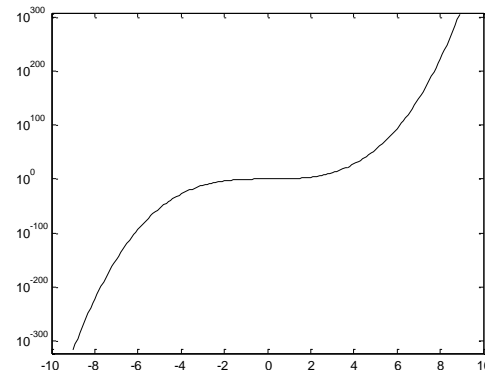
```
% γραμμική κλίμακα
```

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



```
% λογαριθμική κλίμακα y-άξονας
```

```
>> semilogy(x,y,'k')
```



Αντίστοιχα συντάσσονται οι: *semilogx()* και *loglog()*



## Τρισδιάστατες Γραφικές Παραστάσεις

# Καμπύλες στον χώρο

Αντίστοιχη προς την εντολή `plot()` είναι η `plot3()` που επιτρέπει την κατασκευή καμπυλών στο χώρο. Η σύνταξη είναι παρόμοια με αυτή της εντολής `plot()`.

***`plot3(X, Y, Z, LinSpec,...)`***

X: διάνυσμα που θα απεικονιστεί στον άξονα-x

Y: διάνυσμα που θα απεικονιστεί στον άξονα-y

Z: διάνυσμα που θα απεικονιστεί στον άξονα-z

LinSpec: Συμβολοσειρά που καθορίζει τον τύπο της καμπύλης

---

# Παράδειγμα

Έστω η καμπύλη στην παραμετρική μορφή:

$$x = e^{-\frac{t}{10}} \cdot \sin(t)$$

$$y = e^{-\frac{t}{10}} \cdot \cos(t), \quad t \in [0, 10\pi]$$

$$z = t^2$$

```
>> t = 0:0.1:10*pi;
```

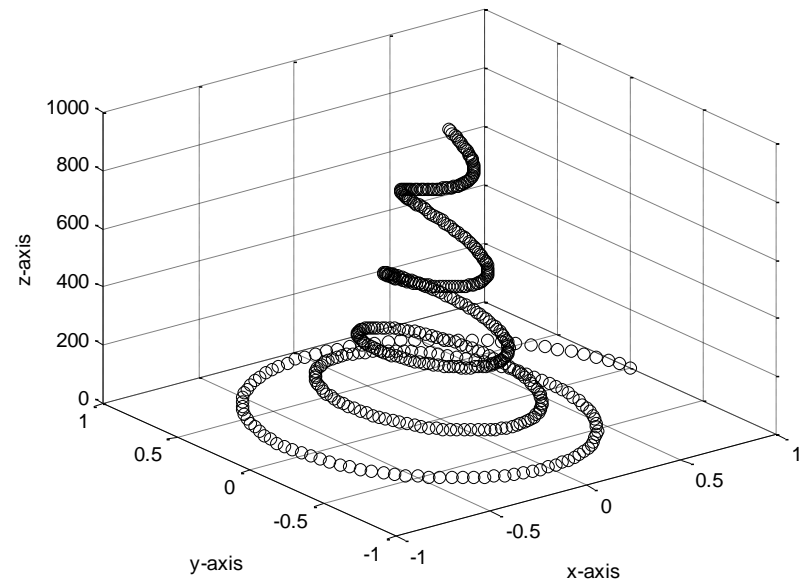
```
>> x = exp(-t/10) .* cos(t);
```

```
>> y = exp(-t/10) .* sin(t);
```

```
>> z = t.^2;
```

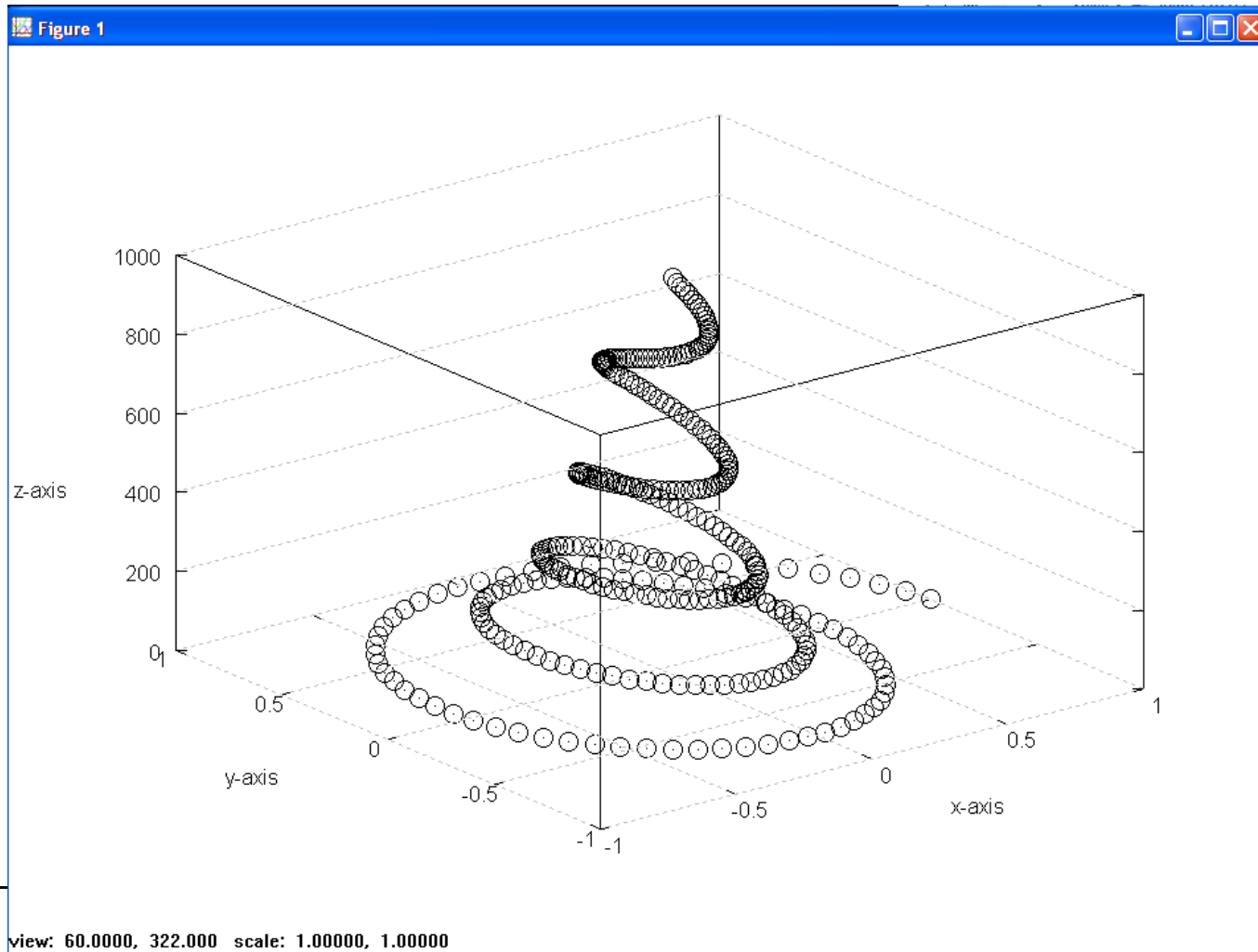
```
>> plot3(x,y,z,'ko'); grid on;
```

```
>> xlabel('x-axis'); ylabel('y-axis'); zlabel('z-axis')
```



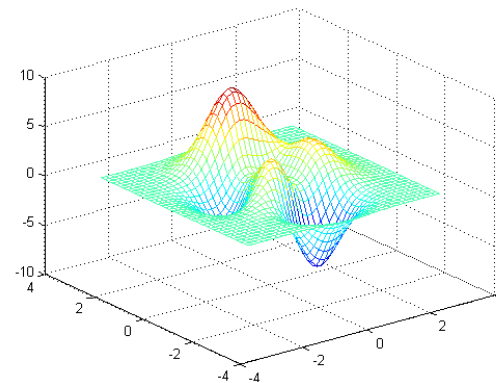
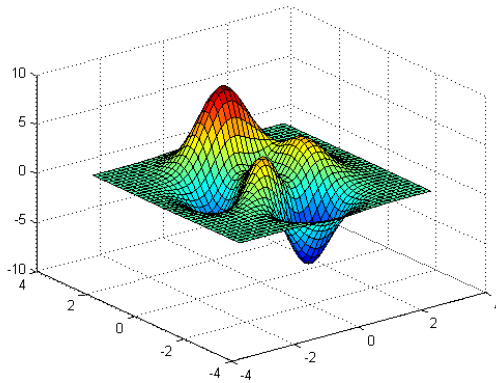


# Όμοιο με Octave!



# Επιφάνειες στον χώρο

Η κατασκευή επιφανειών στον χώρο επιτυγχάνεται με τις εντολές **surf()** και **mesh()**. Π.χ., χρησιμοποιώντας την εγγενή συνάρτηση του Matlab **peaks()** ως παράδειγμα, έχουμε:



```
>> [x,y,z]=peaks(40); % 40x40 πλέγμα  
>> surf(x,y,z) % συμπαγής επιφάνεια  
>> mesh(x,y,z) % επιφάνεια σε μορφή πλέγματος
```

# Οι εντολές *mesh()* και *surf()*

Η σύνταξη των δύο εντολών είναι παρόμοια:

***mesh(X, Y, Z)*** και ***surf(X, Y, Z)***

X: πίνακας που θα απεικονιστεί στον άξονα-x

Y: πίνακας που θα απεικονιστεί στον άξονα-y

Z: πίνακας που θα απεικονιστεί στον άξονα-z (τιμές της συνάρτησης)

---

# Οι εντολές *mesh()* και *surf()*

Οι πίνακες X, Y έχουν την ακόλουθη δομή και μπορούν να κατασκευαστούν εύκολα με την εντολή `meshgrid()`.

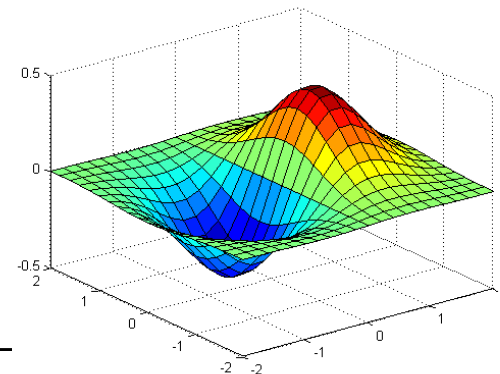
	1	2	3	4	5
1		-1.8000	-1.6000	-1.4000	-1.2000
2	-2	-1.8000	-1.6000	-1.4000	-1.2000
3	-2	-1.8000	-1.6000	-1.4000	-1.2000
4	-2	-1.8000	-1.6000	-1.4000	-1.2000
5	-2	-1.8000	-1.6000	-1.4000	-1.2000
6	-2	-1.8000	-1.6000	-1.4000	-1.2000
7	-2	-1.8000	-1.6000	-1.4000	-1.2000
8	-2	-1.8000	-1.6000	-1.4000	-1.2000
9	-2	-1.8000	-1.6000	-1.4000	-1.2000
10	-2	-1.8000	-1.6000	-1.4000	-1.2000

	1	2	3	4	5
1		-2	-2	-2	-2
2	-1.8000	-1.8000	-1.8000	-1.8000	-1.8000
3	-1.6000	-1.6000	-1.6000	-1.6000	-1.6000
4	-1.4000	-1.4000	-1.4000	-1.4000	-1.4000
5	-1.2000	-1.2000	-1.2000	-1.2000	-1.2000
6	-1	-1	-1	-1	-1
7	-0.8000	-0.8000	-0.8000	-0.8000	-0.8000
8	-0.6000	-0.6000	-0.6000	-0.6000	-0.6000
9	-0.4000	-0.4000	-0.4000	-0.4000	-0.4000
10	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000

```
>> [X,Y] = meshgrid(-2:.2:2, -2:.2:2);
```

```
>> Z = X .* exp(-X.^2 - Y.^2);
```

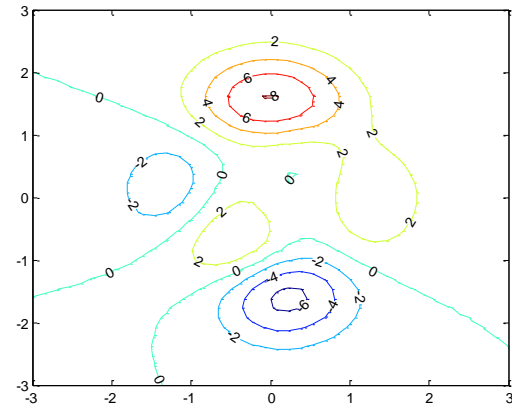
```
>> surf(X,Y,Z)
```



# Ισοϋψείς καμπύλες

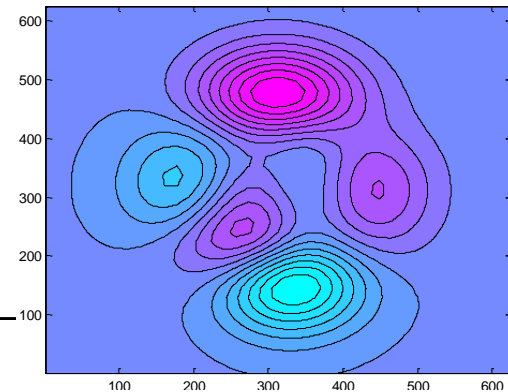
Η εντολή ***contour()*** μπορεί να χρησιμοποιηθεί για την κατασκευή ισοϋψών καμπυλών.

```
>> [x,y,z] = peaks(40);  
>> [c,h] = contour(x,y,z);  
>> clabel(c,h); % τιμή ισοϋψούς
```



% ισοϋψείς με ενιαία χρωματική κλίμακα

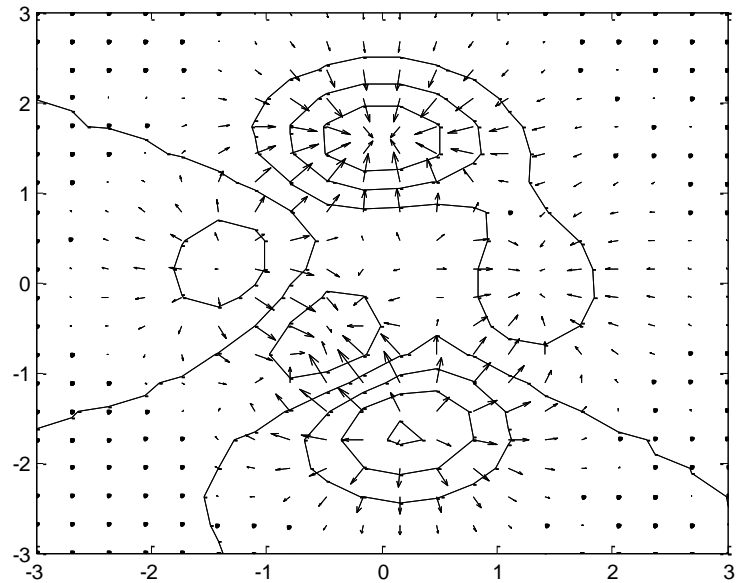
```
>> contourf(interp2(z,4),15)  
>> colormap cool
```



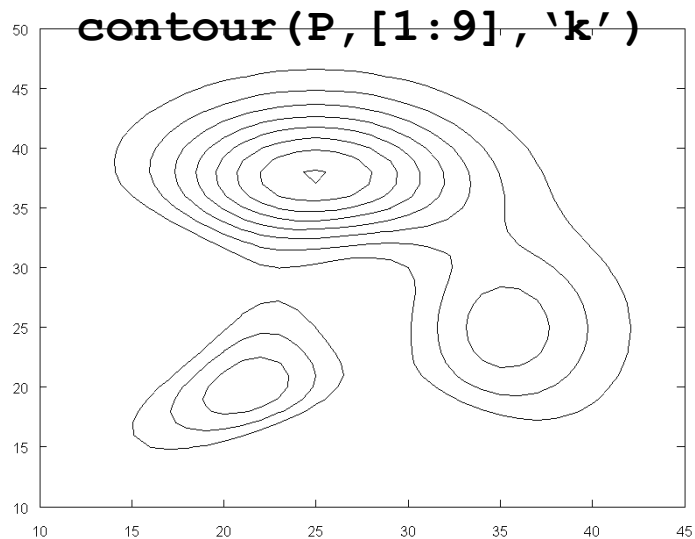
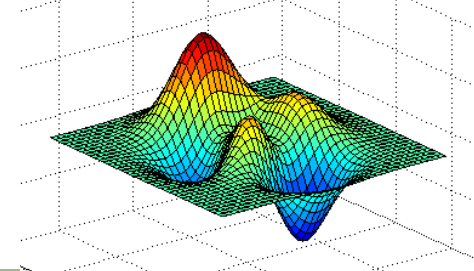
# Διανυσματικά Πεδία

Η απεικόνιση διανυσματικών πεδίων γίνεται συδυάζοντας τις εντολές ***gradient()*** και ***quiver()***. Η μεν πρώτη υπολογίζει τη βαθμίδα ανάμεσα στα στοιχεία ενός πίνακα, η δε δεύτερη απεικονίζει το αποτέλεσμα με μορφή διανυσμάτων, το μήκος των οποίων είναι ανάλογο της βαθμίδας. Π.χ.,

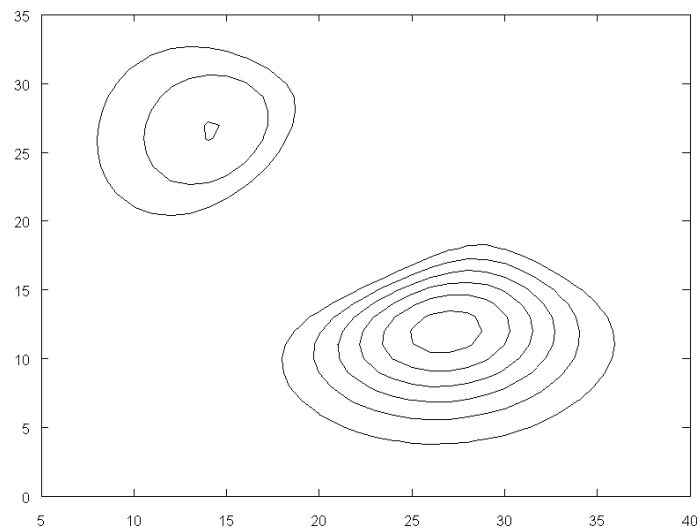
```
>> [x,y,z]=peaks(20);  
>> [dx,dy]=gradient(z,0.5,0.5);  
  
>> hold on  
>> contour(x,y,z,'k');  
>> quiver(x,y,dx,dy,1,'k');  
>> box on  
>> hold off
```



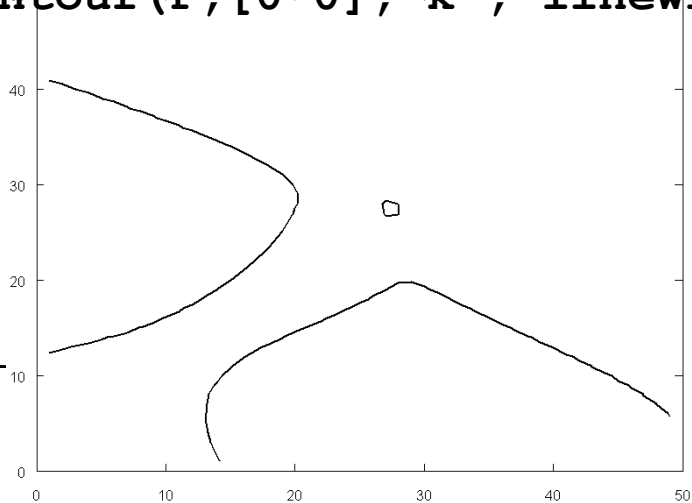
# Πλοτάροντας τμήματα



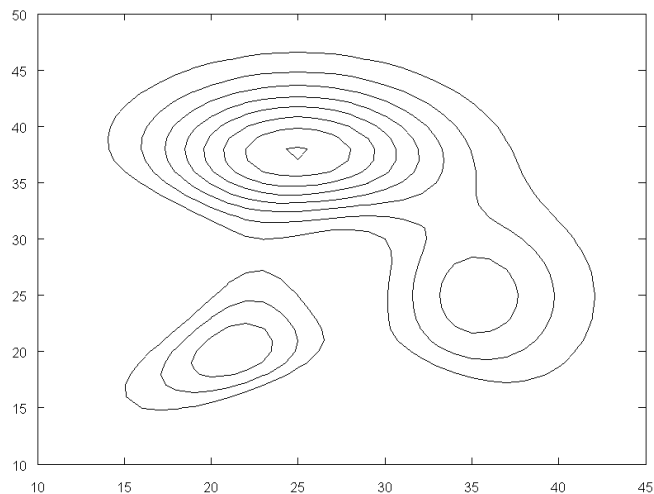
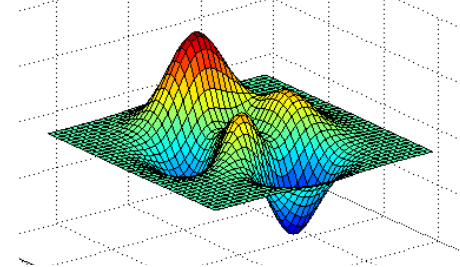
```
contour(P, [-9:-1], 'k-')
```



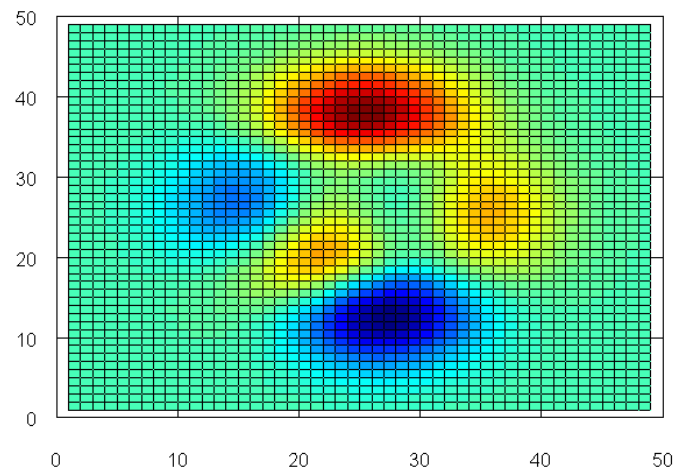
```
contour(P, [0 0], 'k', 'linewidth', 2)
```



# Πλοτάροντας τμήματα

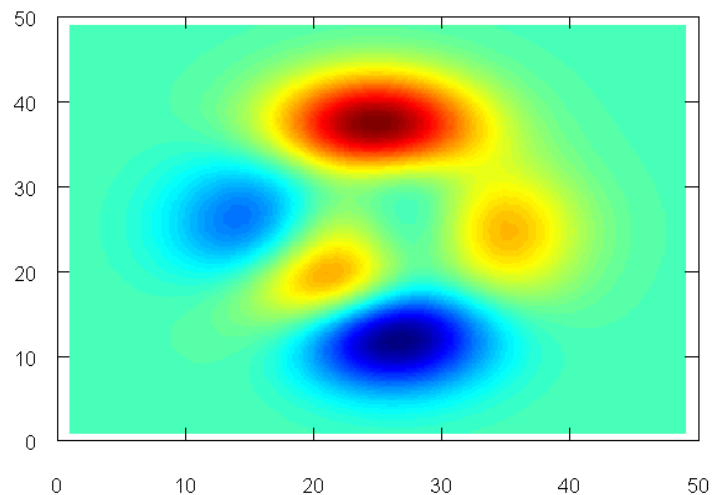


`contour(P, [1:9], 'k')`



`pcolor(P)`

`shading interp`







Άλλα Γραφικά

# Άλλα Γραφικά

Ανάλογα με το είδος της πληροφορίας που θέλουμε να απεικονίσουμε, συχνά χρησιμοποιούμε μία σειρά από άλλου τύπου γραφικές παραστάσεις, Π.χ., ιστογράμματα, πολικά διαγράμματα, κ.λπ.

Όλα τα παραπάνω μπορούν να γίνουν πολύ εύκολα στο Matlab.

---

# Ιστογράμματα

Η εντολή κατασκευής ενός ιστογράμματος είναι η ***hist()***. Η σύνταξή της είναι:

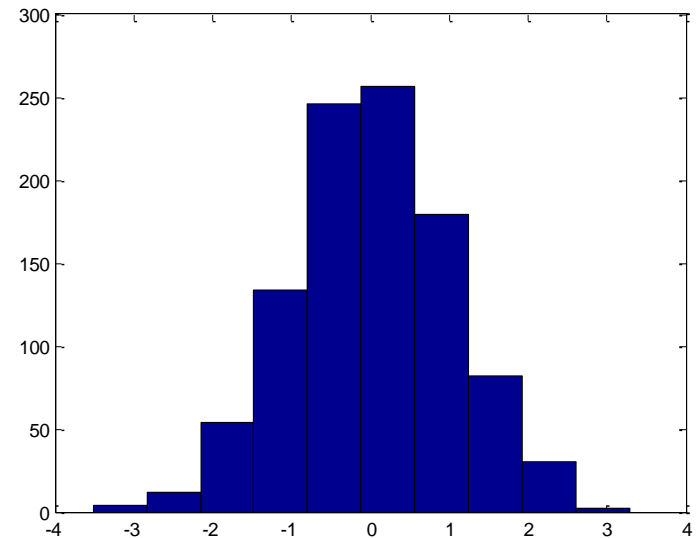
***hist(X, nbins)***

**X**: διάνυσμα με τους αριθμούς

**nbins**: το πλήθος των κλάσεων τιμών

```
>> x = randn(1000,1);
```

```
>> hist(x,10)
```



# Διαγράμματα με ράβδους σφαλμάτων

Τα διαγράμματα τύπου `errorbar` είναι χρήσιμα σε περιπτώσεις που στο σχήμα είναι απαραίτητη η πληροφορία για τη διακύμανση των τιμών γύρω από μία κεντρική τιμή. Η σύνταξη της εντολής είναι:

***errorbar(X,Y,E)***

**X:** Τιμές στον άξονα-x

**Y:** Κεντρικές τιμές στον άξονα-y

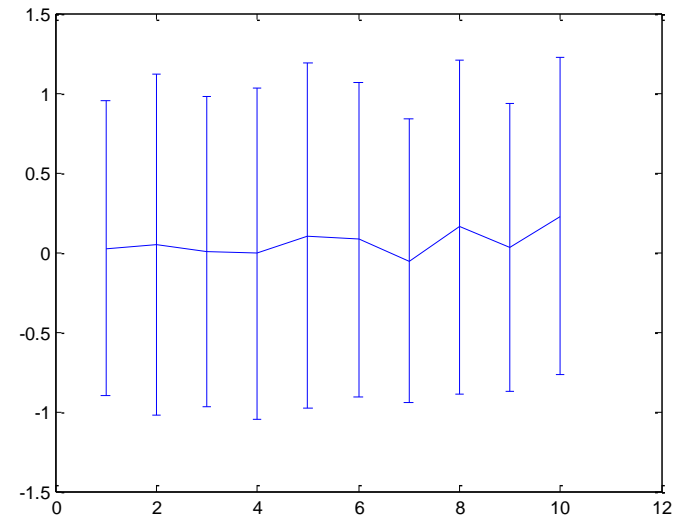
**E:** Διακύμανση γύρω από την κεντρική τιμή (μήκος ράβδου)

---

# Παράδειγμα errorbar

Κατασκευάζουμε πίνακα από τυχαίους αριθμούς με κανονική κατανομή. Σε διάγραμμα τύπου errorbar απεικονίζουμε την μέση τιμή και την τυπική απόκλιση για κάθε στήλη του πίνακα.

```
>> x = randn(100,10);  
>> errorbar(1:10,mean(x),std(x))
```



# Πολικά διαγράμματα

Τα πολικά διαγράμματα χρησιμοποιούνται για γραφικές παραστάσεις σε πολικές συντεταγμένες. Η εντολή στο Matlab είναι η ***polar()*** και συντάσσεται ως ακολούθως

***polar(theta, r, linspec)***

**theta:** τιμές της γωνίας  $\theta$

**r:** τιμές της μεταβλητής  $r$

**linspec:** τύπος καμπύλης

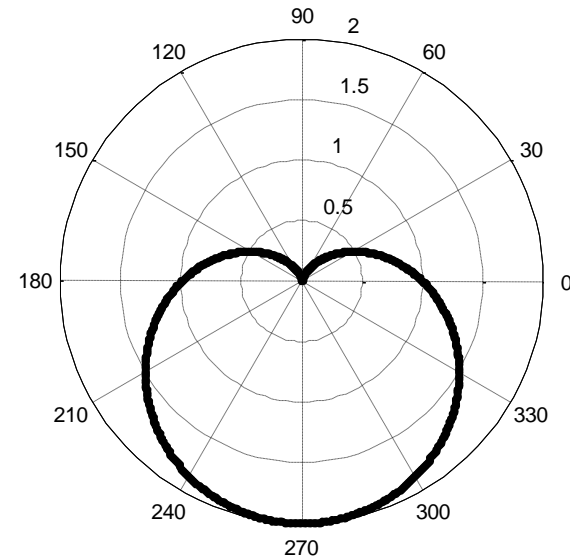
---

# Παράδειγμα πολικού διαγράμματος

Η κορδιοειδής καμπύλης στις πολικές συντεταγμένες περιγράφεται από μία πολύ απλή σχέση:

$$r = 1 - \sin(\theta) , 0 \leq \theta < 2\pi$$

```
>> t = 0:0.01:2*pi;  
>> polar(t,1 - sin(t), 'k.')
```



# Απεικόνιση μιγαδικών αριθμών

Οι μιγαδικοί αριθμοί δύνανται να απεικονιστούν στο επίπεδο σε πολικό διάγραμμα, όπου η γωνία αντιστοιχεί στο όρισμά τους και το μήκος στην απόλυτη τους τιμή. Με την εντολή **compass** είναι δυνατή η απεικόνιση τους σε πολικό διάγραμμα.

```
>> z = 2+3i;
```

```
>> rad2deg(angle(z)) % όρισμα σε μοίρες
```

```
ans =
```

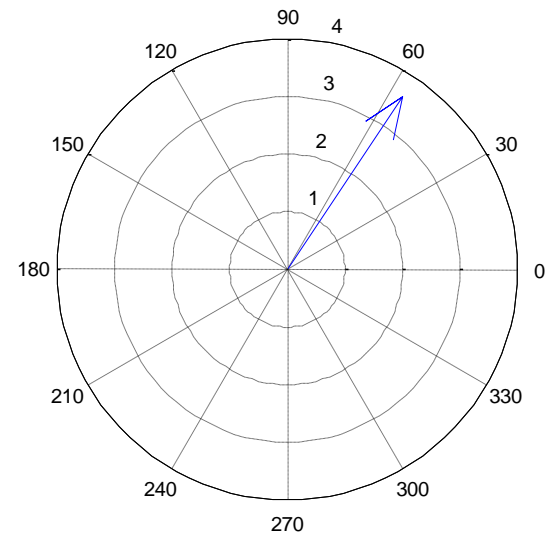
```
56.3099
```

```
>> abs(z) % απόλυτη τιμή
```

```
ans =
```

```
3.6056
```

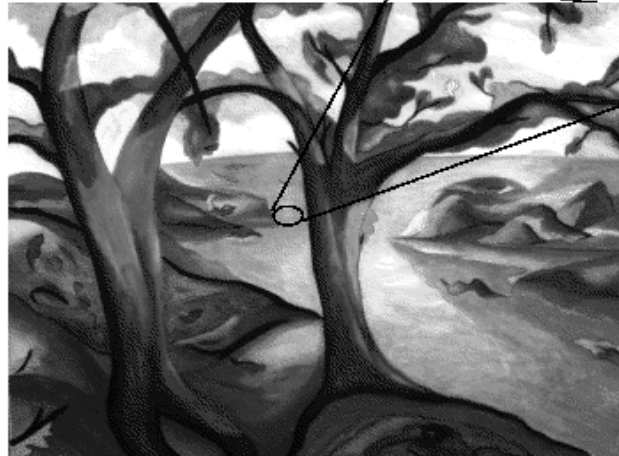
```
>> compass(z)
```





# Απεικόνιση εικόνων

- Binary images :  $\{0,1\}$
- Intensity images :  $[0,1]$  or `uint8`, `double` etc.
- RGB images :  $m \times n \times 3$
- Multidimensional images:  $m \times n \times p$  ( $p$  is the number of layers)



0.2251	0.2563	
0.5342	0.2051	0.2157
0.5342	0.1789	0.1307
0.4308	0.2483	0.2624
0.3344	0.2624	

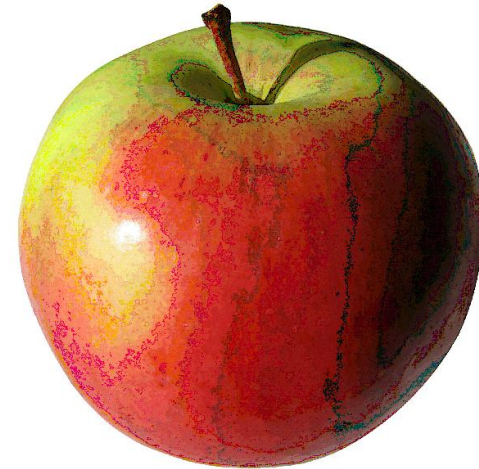
0.2235	0.1294	<b>Blue</b>	0.4196	
0.5804	0.2902	<b>0.0627</b>	0.2902	0.2902
0.5804	0.0627	<b>0.0627</b>	0.0627	0.2235
0.5176	0.1922	<b>0.0627</b>	<b>Green</b>	0.1922
0.5176	0.1294	<b>0.1608</b>	0.1294	0.2588
0.5176	0.1608	<b>0.0627</b>	0.1608	0.2588
0.5490	0.2235	0.5490	<b>Red</b>	0.7412
0.5490	0.3882	<b>0.5176</b>	0.5804	0.7765
0.5490	0.2588	0.2902	0.2588	0.2235
0.2235	0.1608	0.2588	0.2588	0.1608
0.2588	0.1608	0.2588	0.2588	0.2588



# Απεικόνιση εικόνων

Αρχική εικόνα = apple.jpg

```
apple=imread('apple.jpg')  
imshow(apple)
```



```
etna=imread('etna.jpg')  
imshow(etna)
```



# Μελέτη των εικόνων

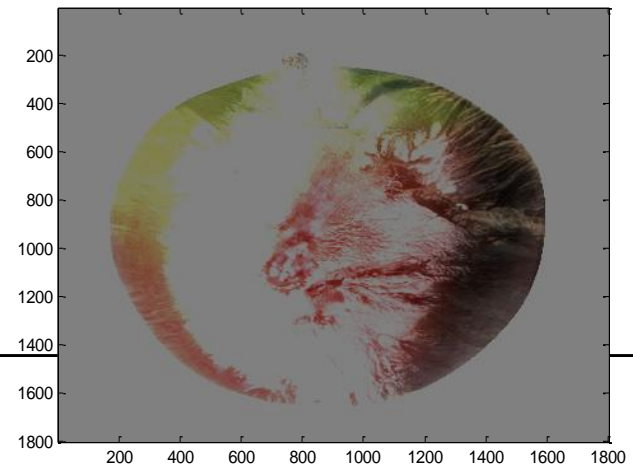
- `size(etna) -> ans = 432 432 3`
  - `Size(apple) -> ans = 1800 1800 3`
  - `B = IMRESIZE(A, [NUMROWS NUMCOLS])`
-

# Επεξεργασία εικόνας

## □ Συνδυάζοντας εικόνες, the old way

```
% Μετρώ τον χρόνο χρησιμοποιώντας stopwatch timer  
tic  
for i = 1 : size(apple, 1)  
    for j = 1 : size(apple, 2)  
        for k = 1 : size(apple, 3)  
            output(i, j, k) = (apple(i, j, k) + etnnew(i, j, k))/2;  
        end  
    end  
end  
toc
```

15.679358 seconds

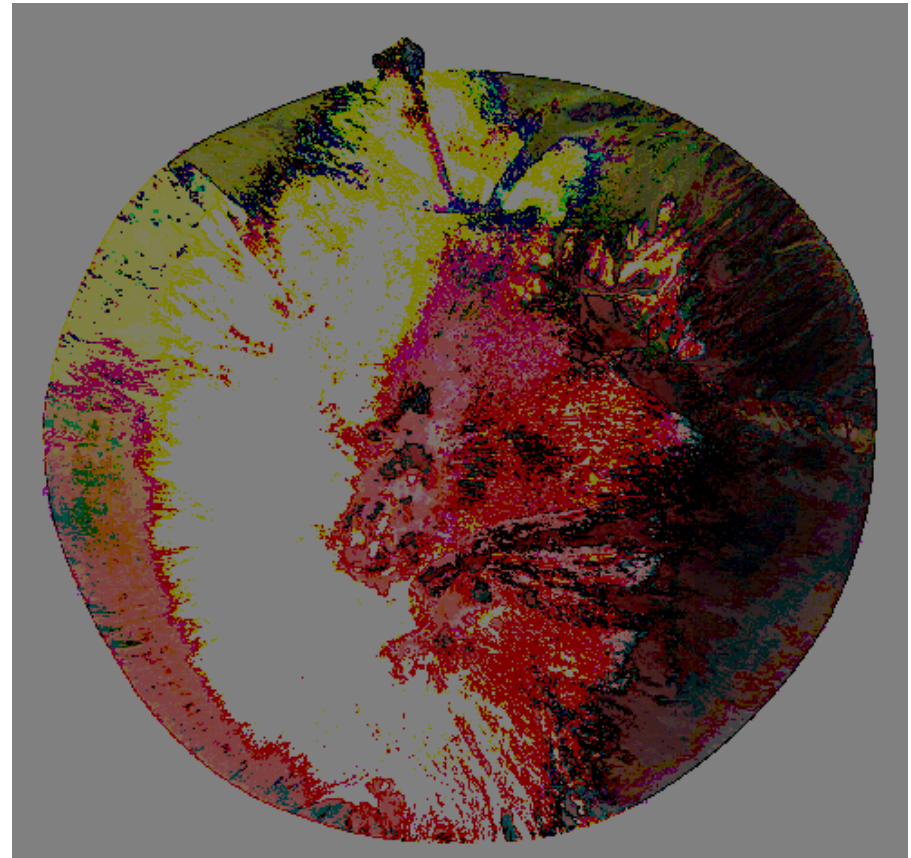


# Επεξεργασία εικόνας

Συνδυάζοντας εικόνες

```
tic  
imnew=(apple+etnnew)/2;  
toc  
  
imshow(imnew)
```

0.042346 seconds



# «Παράγοντας» εικόνα θορύβου

```
row = 1800;  
col = 1800;  
img = rand(row, col);  
img3 = 200*img;  
img3=round(img3)  
imshow(img);
```





Γραφικές Παραστάσεις Ανώνυμων και Εσωτερικών Συναρτήσεων

# Γραφικές Παραστάσεις Ανώνυμων και Εσωτερικών Συναρτήσεων

Οι εντολές κατασκευής γραφικών που παρουσιάστηκαν αφορούν *πίνακες* ή *διανύσματα*. Αντίστοιχες εντολές μπορούν να χρησιμοποιηθούν όταν έχουμε εσωτερικές ή ανώνυμες συναρτήσεις.

Η ομάδα αυτών των εντολών φέρει το πρόθεμα *ez...* και ακολουθεί η αντίστοιχη εντολή. Π.χ., αντί της ***plot()*** η ***ezplot()***

---



# Εντολές κατασκευής γραφικών παραστάσεων

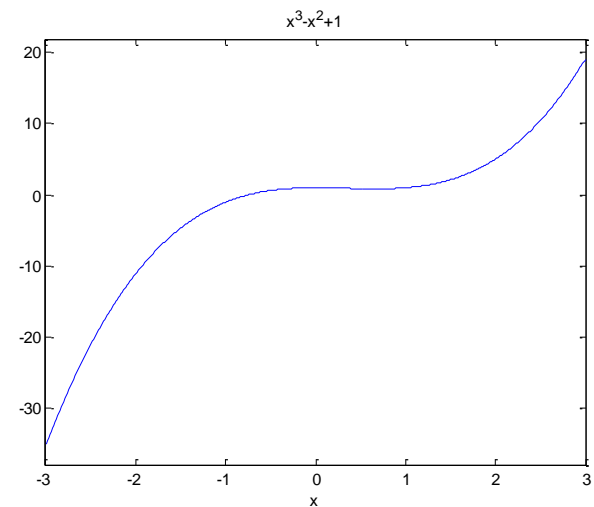
## *Analytic Plots*

<b>ezplot(), ezplot3()</b>	Απεικόνιση δισδιάστατης και τριδιάστατης αναλυτικής συνάρτησης, αντιστοίχως
<b>ezpolar()</b>	Απεικόνιση δισδιάστατης αναλυτικής συνάρτησης, σε πολικές συντεταγμένες
<b>ezcontour()</b>	Διάγραμμα ισοϋψών γραμμών αναλυτικής συνάρτησης
<b>ezcontourf()</b>	Διάγραμμα ισοϋψών γραμμών αναλυτικής συνάρτησης με συνεχές χρωματικό υπόβαθρο (χάρτης)
<b>ezsurf()</b>	Απεικόνιση τρισδιάστατης επιφάνειας αναλυτικής συνάρτησης
<b>ezsurfc()</b>	Απεικόνιση τρισδιάστατης επιφάνειας και ισοϋψών γραμμών αναλυτικής συνάρτησης
<b>ezmesh()</b>	Απεικόνιση τρισδιάστατου πλέγματος αναλυτικής συνάρτησης
<b>ezmeshc()</b>	Απεικόνιση τρισδιάστατου πλέγματος και ισοϋψών γραμμών αναλυτικής συνάρτησης

# Παραδείγματα

```
>> f = @(x) x.^3 - x.^2 + 1;
```

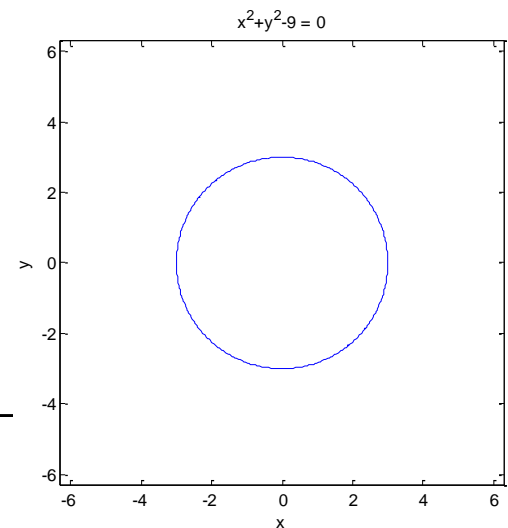
```
>> ezplot(f, [-3, 3])
```



```
>> f = @(x,y) x.^2 + y.^2 - 9
```

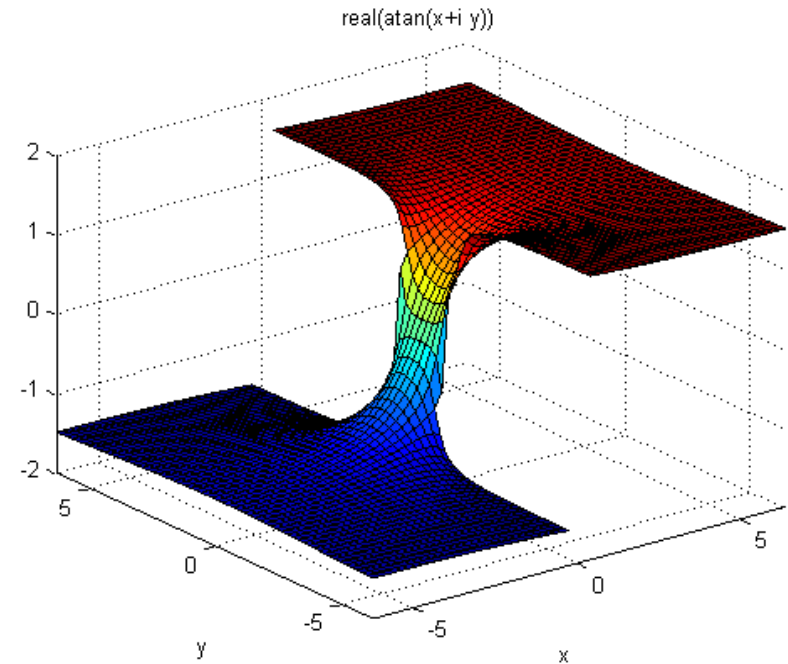
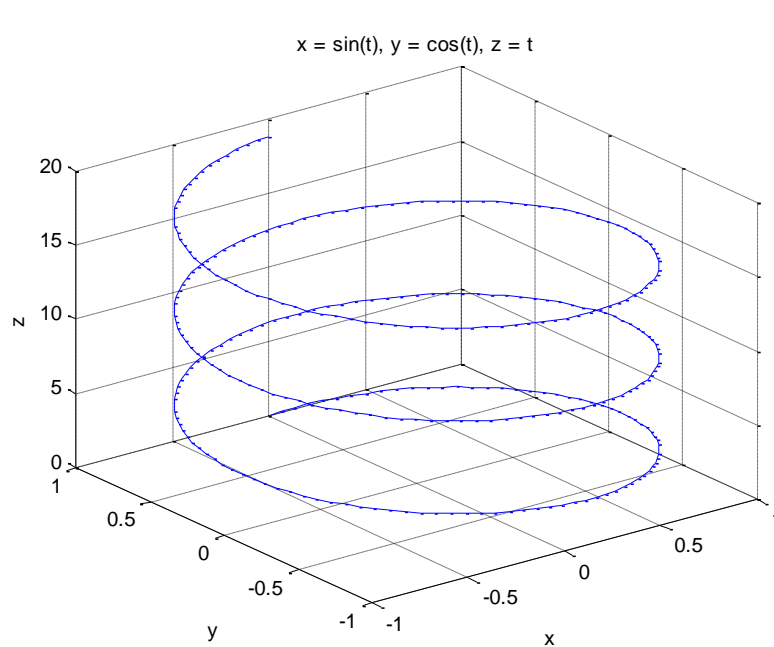
```
>> ezplot(f)
```

```
>> axis square
```



# Παράδειγμα

```
>> ezplot3('sin(t)', 'cos(t)', 't', [0, 6*pi])
```



```
>> f = @(x,y) real(atan(x+i*y));
```

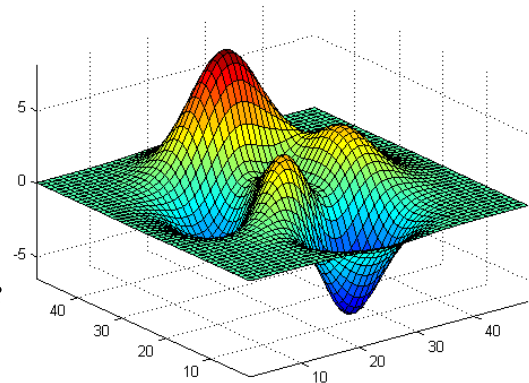
```
>> ezsurf(f)
```



## Εγγραφή και Αναπαραγωγή Ταινίας

# Εγγραφή και Αναπαραγωγή Ταινίας (1/3)

```
Z = peaks; % εγγενής συνάρτηση
surf(Z); % απεικόνιση
axis tight
set(gca, 'nextplot', 'replacechildren');
```



Σύνταξη set: ***set(Object, Property1, PropertyValue1,...)***

**Object:** Το ID (αριθμός) ενός αντικειμένου, π.χ. gca επιστρέφει το ID του τρέχοντος γραφήματος της οθόνης.

**Property:** Ιδιότητα του αντικειμένου που θέλουμε να αλλάξουμε

**PropertyValue:** Νέα τιμή της ιδιότητας

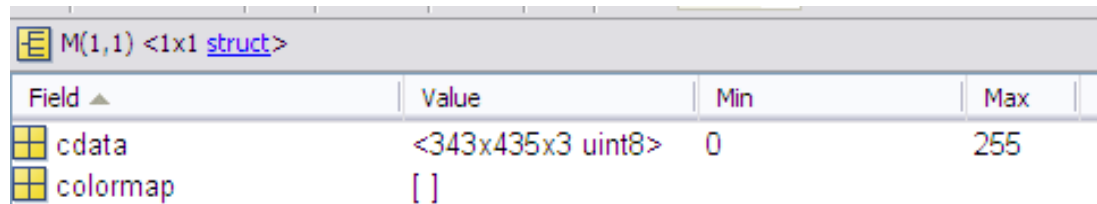
---

***nextplot = replacechildren.*** Remove all child objects, but do not reset figure properties

# Εγγραφή και Αναπαραγωγή Ταινίας (2/3)

```
% Εγγραφή ταινίας (αποτελούμενη από 20 εικόνες)
for j = 1:20
    surf(sin(2*pi*j/20)*Z)    % sin(0->2π)
    M(j) = getframe;
end
```

Η `getframe` «γεμίζει» τον πίνακα `M`. Τι είναι το στοιχείο `M(i)`;



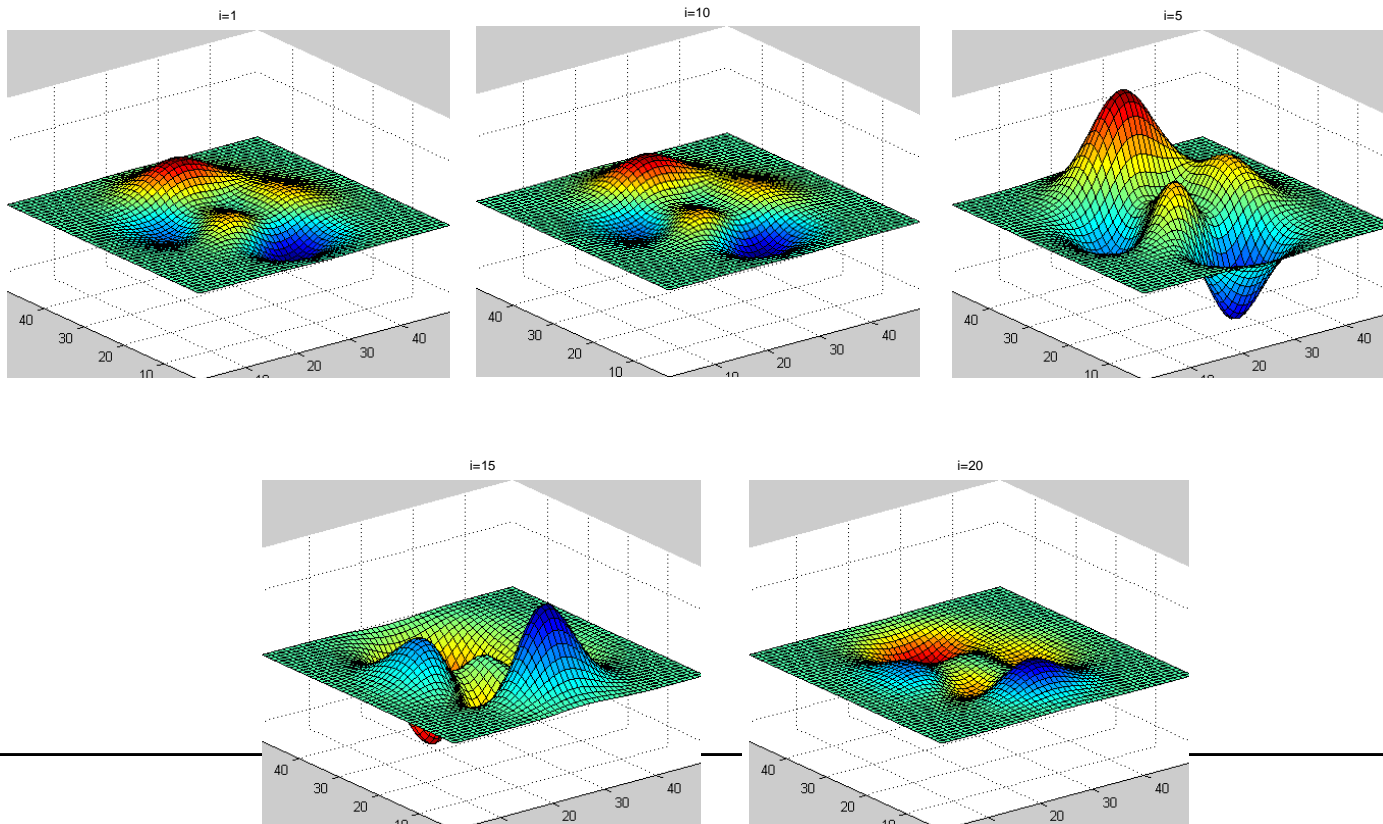
Field	Value	Min	Max
cdata	<343x435x3 uint8>	0	255
colormap	[]		

Μία δομή με δύο πίνακες: `cdata` (3D: Τιμές RGB της εικόνας), `colormap`: Κωδικοποίηση χρωμάτων (εδώ κενός, χρησιμοποιείται η προεπιλεγμένη κλίμακα)

# Εγγραφή και Αναπαραγωγή Ταινίας (3/3)

`% Play the movie twenty times`

`movie(M,20)`





Τέλος

Κώστας Καρατζάς  
Τμήμα Μηχανολόγων Μηχανικών, ΑΠΘ



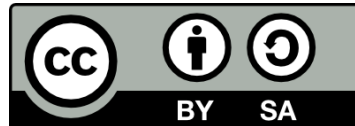
# Σημείωμα Αναφοράς

- Copyright Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Κωνσταντίνος Καρατζάς. «Πληροφορική. Ενότητα 8: A. Fibonacci Revisited. B. Επανάληψη Α΄ κύκλου. Γ. Γραφικά στο Matlab». Έκδοση: 1.0. Θεσσαλονίκη 2014. Διαθέσιμο από τη δικτυακή διεύθυνση: <https://opencourses.auth.gr/courses/OCRS328/>



# Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά - Παρόμοια Διανομή [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



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

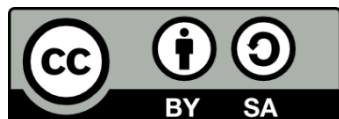
[1] <http://creativecommons.org/licenses/by-sa/4.0/>





# Τέλος ενότητας

Θεσσαλονίκη, Εαρινό Εξάμηνο 2014-2015



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



ΑΡΙΣΤΟΤΕΛΕΙΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΘΕΣΣΑΛΟΝΙΚΗΣ

---

# Σημειώματα

# Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

