

Μέθοδοι Επίλυσης με Η/Υ

7^ο Μάθημα – *Γραφήματα στο MATLAB (2)*

Ν.Δ. Λαγαρός, Α. Στάμος, Χ. Φραγκουδάκης

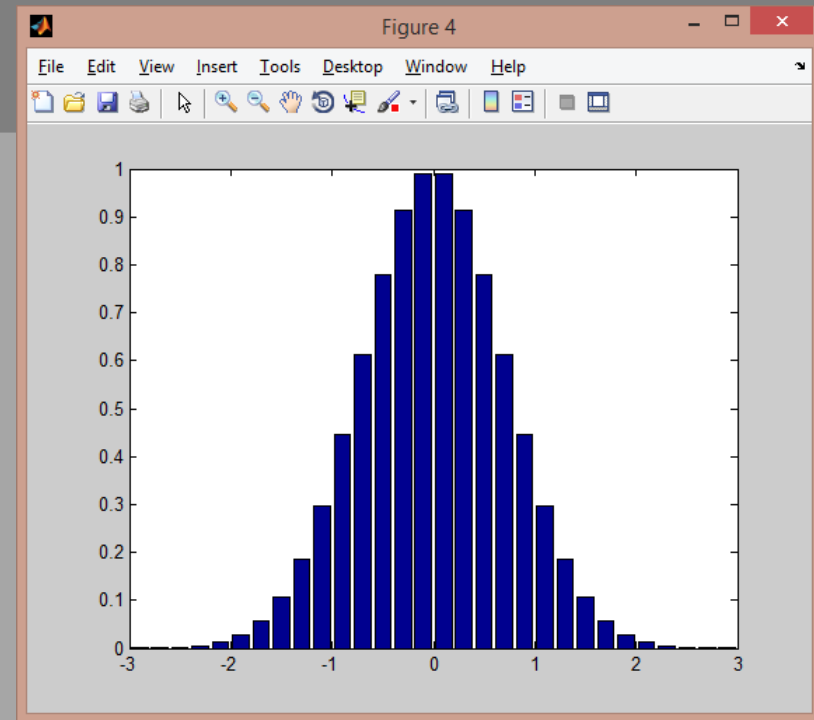
Συναρτήσεις για δημιουργία γραφημάτων

- 1) 3D γραφήματα
- 2) meshgrid
- 3) mesh, surf.
- 4) κλπ.

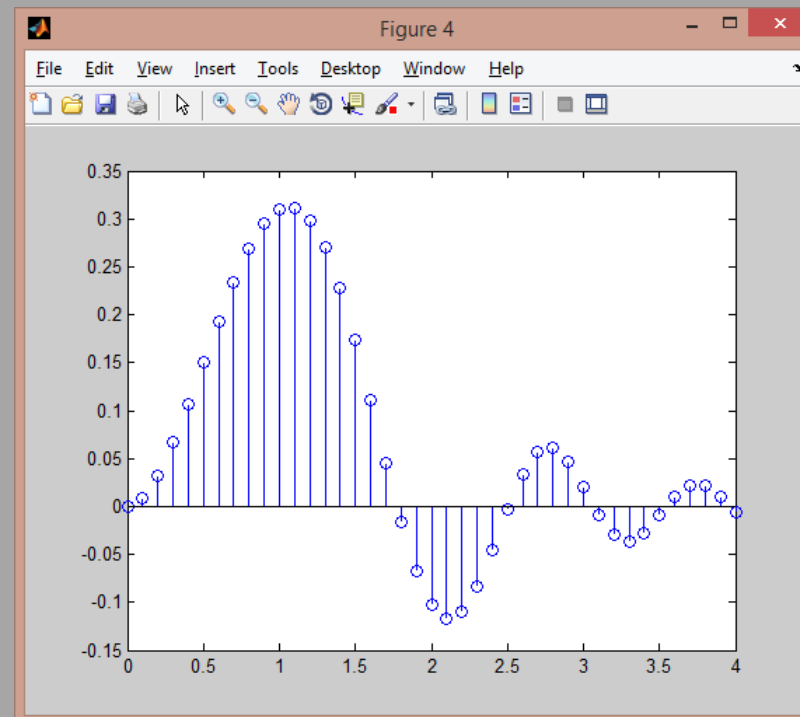
Συναρτήσεις bar & stem

Δεδομένα και γραφήματα

```
>>figure(4)  
>>x = -2.9:0.2:2.9;  
>>bar(x,exp(-x.*x));
```



```
>>x = 0:0.1:4;;  
>>y = sin(x.^2).*exp(-x);  
>>stem(x,y)
```



Συνάρτηση linspace

Σύνταξη:

$y = \text{linspace}(a,b)$ (εξορισμού 100 σημεία)

$y = \text{linspace}(a,b,n)$

Η `linspace` γεννά διανύσματα με στοιχεία γραμμικά κατανεμημένα σε συγκεκριμένο διάστημα. Είναι παρόμοιο με το τελεστή `:`, αλλά δίνει άμεσο έλεγχο επί του αριθμού των σημείων.

Δεδομένα και γραφήματα

```
>>figure(15)
```

```
>>x=linspace(0,(2*pi),100);
```

```
>>y1=sin(x);
```

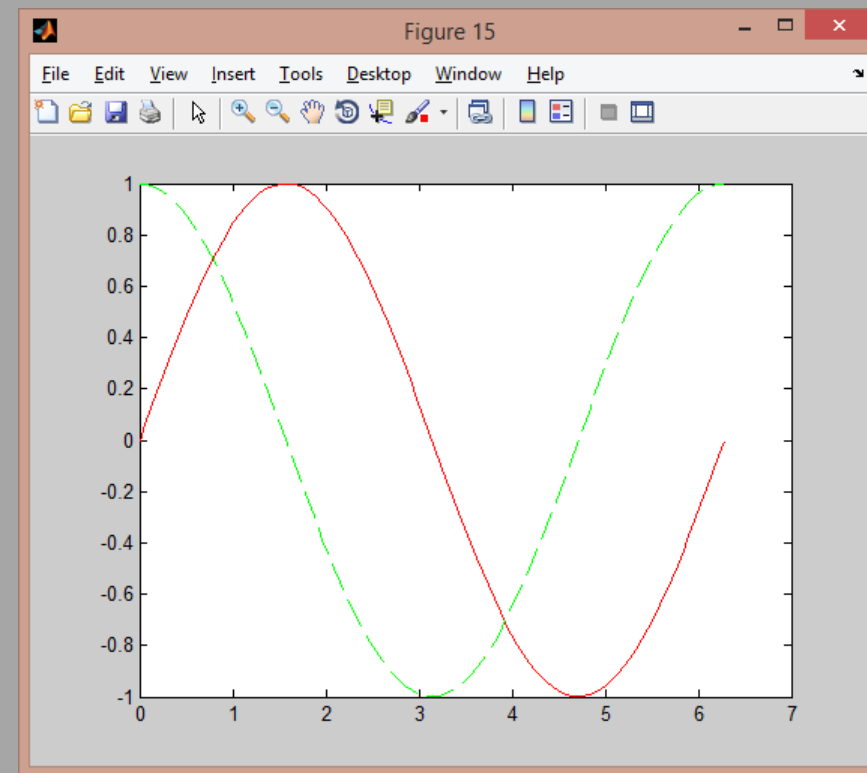
```
>>y2=cos(x);
```

```
>>plot(x,y1,'r-')
```

```
>>hold on
```

```
>>plot(x,y2,'g--')
```

```
>>hold off
```



Αποθήκευση εικόνας

Μία εικόνα μπορεί να αποθηκευθεί με τις εντολές:

- `savefig('PeaksFile.fig')` και
- `print(filename,formattype)`

```
figure()  
hold on, grid on; box on;  
time=dt*(1:nsteps);  
plot(time,Acc,'b-')
```

Αποθήκευση της εικόνας σε αρχείο .fig (ανοίγει μόνο με το Matlab)

```
savefig('seismos.fig')
```

Αποθήκευση της εικόνας σε αρχείο .jpg ή .png

(γενικά αρχεία εικόνας – μπορούν να μπουύν στο Word για μια εργασία)

```
print('seismos.jpg','-djpeg')  
print('seismos.png','-dpng')
```

3D διαγράμματα

- Εκεί όπου η δύναμη του Matlab γίνεται πραγματικά εμφανής είναι στην χάραξη σε 3D και 3D οπτικοποίηση δεδομένων!
- Matlab διαθέτει σειρά από διαφορετικά είδη 3D γραφημάτων αλλά μάλλον θα βρείτε 3 ή 4 για να είναι πιο χρήσιμα:
 - x,y,z 3D line plot: `plot3()`
 - mesh plot: `mesh()`
 - surface plot: `surf()`
 - περίγραμμα (contour) plot: `contour()`
 - συνδυασμένο surface/mesh with contour: `surfc()`

3D διαγράμματα

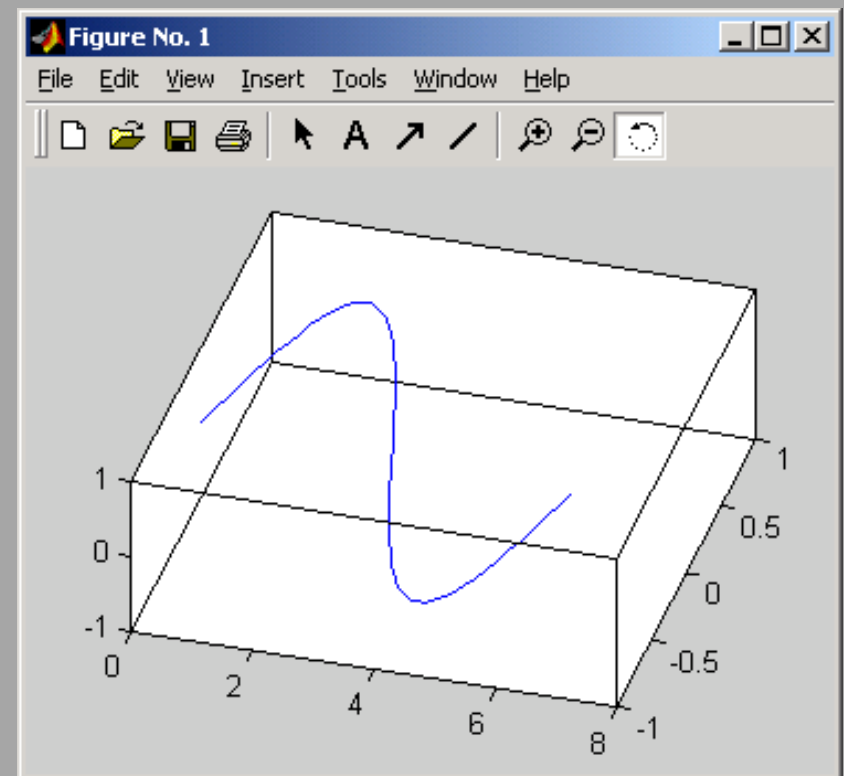
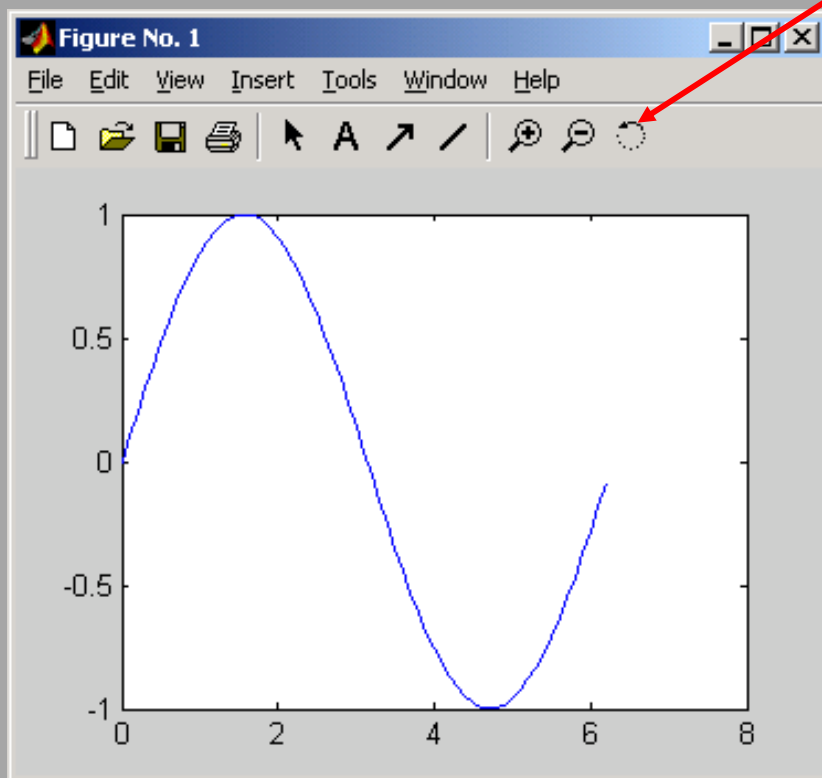
- Υπάρχουν πολλές επιλογές για την χάραξη 3D γραφημάτων, αλλά θα λάβουμε υπόψη αρχικά τα βασικά:
 - Χάραξη καμπύλης σε 3D: `plot3(x,y,z)`
 - Χάραξη επιφάνειας, $z=f(x,y)$, in 3D: `mesh()` & `surf()`
 - Υπάρχουν επίσης πολλά άλλα θέματα σχετικά με την χάραξη γραφημάτων τα οποία δεν θα καλύψει η συζήτηση στην τάξη (αλλά μπορείτε να βρείτε ενδιαφέρον να εξερευνήσετε μόνοι σας):
 - ribbon plots
 - quiver plots (showing vectors)
 - volume plots
 - advanced `colormap` use

3D διαγράμματα

Κάθε 2D plot είναι ένα 3D plot χωρίς τον ορισμό της 3ης διάστασης.

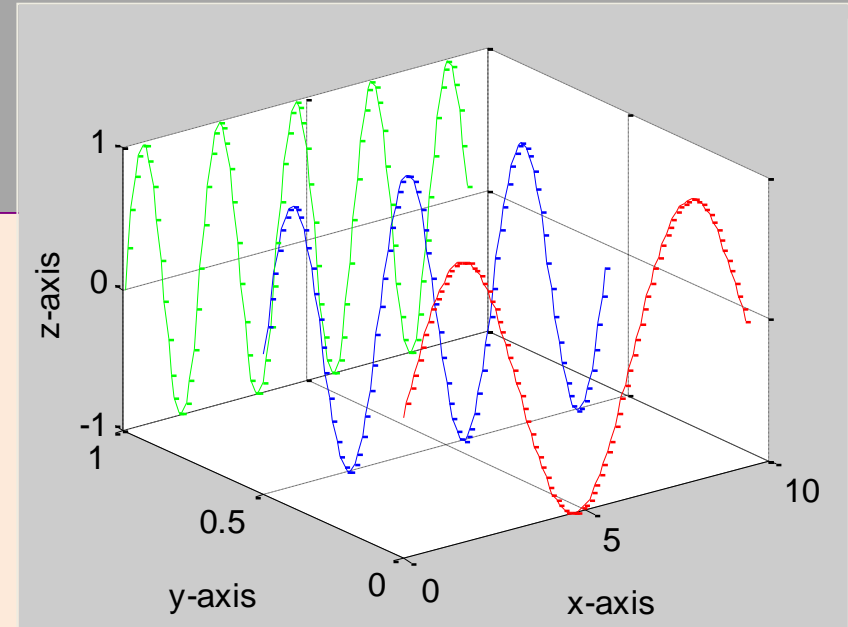
```
>> x=0:0.1:2.*pi;  
>> plot(x,sin(x))
```

Rotate 3D button: δοκιμάστε



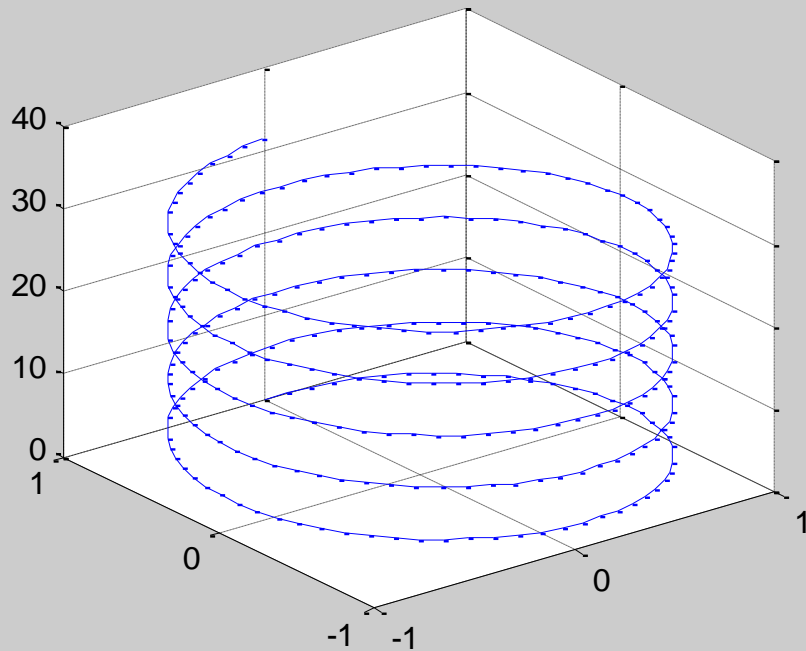
3D διαγράμματα

```
>> x=0:0.1:3.*pi;  
>> z1=sin(x);  
>> z2=sin(2.*x);  
>> z3=sin(3.*x);  
>> y1=zeros(size(x));  
>> y3=ones(size(x));  
>> y2=y3./2;  
>> plot3(x,y1,z1,'r',x,y2,z2,'b',x,y3,z3,'g')  
>> grid on  
>> xlabel('x-axis'), ylabel('y-axis'), zlabel('z-axis')
```

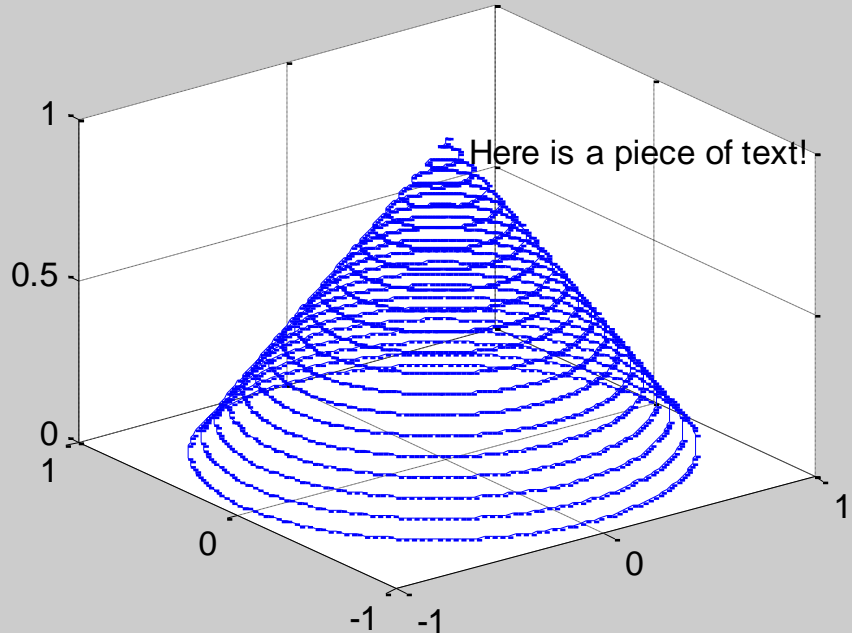


3D διαγράμματα – καμπύλες

```
>> theta=0:0.1:10.*pi;  
>> plot3(sin(theta),cos(theta),theta)  
>> grid on
```



```
>> turns=40.*pi;  
>> theta=linspace(0,turns,4000);  
>> x=cos(theta).*(turns-theta)./turns;  
>> y=sin(theta).*(turns-theta)./turns;  
>> z=theta./turns;  
>> plot3(x,y,z)  
>> grid on  
>> text(0.5,0.5,0.75,'Here is a piece of text!');
```



3D διαγράμματα – επιφάνειες

- Είναι συχνά επιθυμητό να σχεδιαστούν συναρτήσεις της μορφής :
 $z=f(x,y)$
 - για κάθε (x, y) , μπορούμε να υπολογίσουμε μία τιμή για το z
 - Αυτό ορίζει μια επιφάνεια σε χώρο 3D
- Αν μπορούμε να ορίσουμε τα (x, y) σε τακτά διαστήματα, το Matlab παρέχει τρόπους να σχεδιάσετε τη συνάρτηση ως ένα πλέγμα ή επιφάνεια σε 3D.
- Οι (x, y) τιμές αποθηκεύονται σε πίνακες και καθορίζουν ένα πλέγμα σημείων, μέσω των οποίων θα δημιουργηθεί η επιφάνεια.
- Επίσης υπάρχουν εργαλεία στο Matlab για να χειριστούμε την κατάσταση στην οποία τα (x, y) δεν ορίζονται σε ένα πλέγμα.

3D διαγράμματα

Τριδιάστατα γραφήματα:

```
>> figure(8)
```

```
>> z=peaks(25);
```

(έτοιμη συνάρτηση 2 μεταβλητών)

```
>> mesh(z);
```

mesh(X,Y,Z) σχεδιάζει ένα πλέγμα με το χρώμα να καθορίζεται σε σχέση με την τιμή του Z.

```
>> z=peaks(25);
```

```
>> surf(z);
```

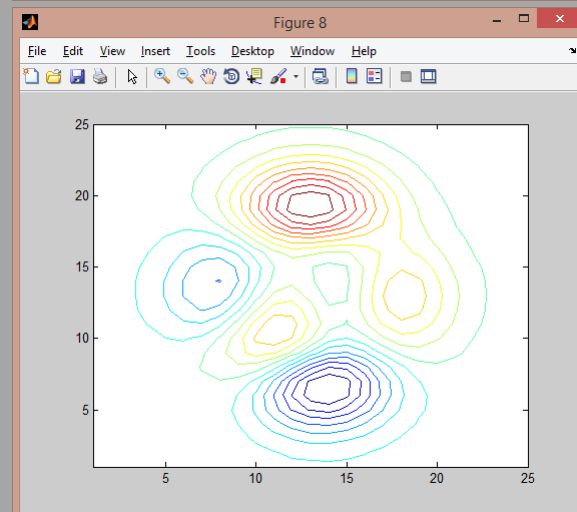
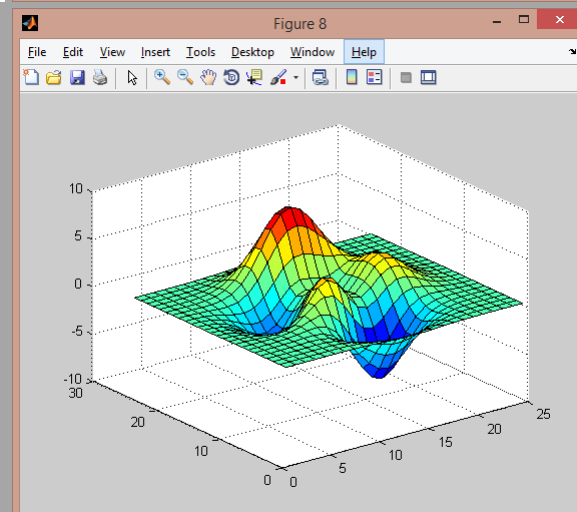
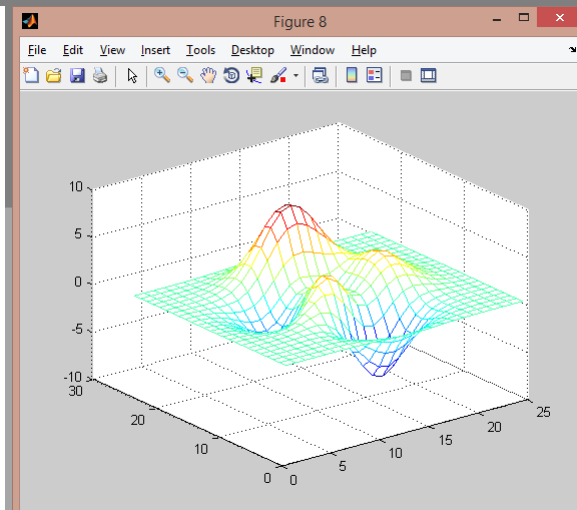
```
>> colormap(jet);
```

colormap είναι ένα μητρώο m-by-3 από πραγματικούς αριθμούς στο 0.0-1.0. Κάθε γραμμή αντιστοιχεί σε ένα διάνυσμα RGB που ορίζει ένα χρώμα.

```
>> z=peaks(25);
```

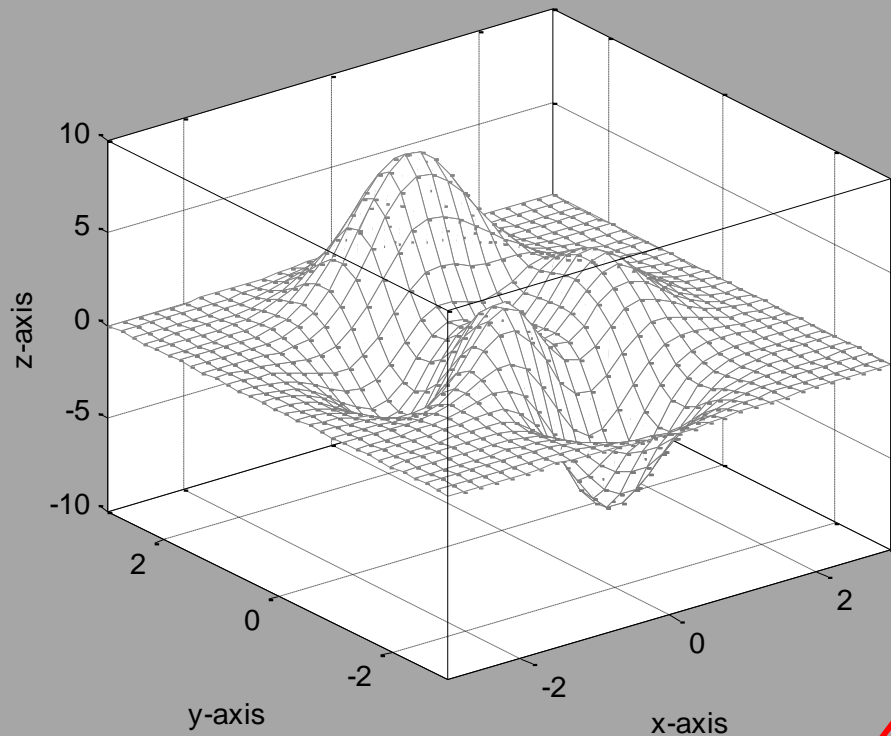
```
>> contour(z,16);
```

contour σχεδιάζει ισοϋψείς με βάση το z, όπου η κλίμακα αυτών είναι 16.



3D διαγράμματα – επιφάνειες

Χρειαζόμαστε έναν τρόπο για να δημιουργήσει το εύρος των (x, y) που απαιτούνται για τον υπολογισμό $f(x, y)$



NOTE:

xx μεταβάλλεται κατά ΣΕΙΡΕΣ ενώ
 yy μεταβάλλεται κατά ΣΤΗΛΕΣ

Αυτό δεν θα λειτουργήσει, γιατί χρειαζόμαστε όλες τις τιμές του y για κάθε τιμή του x και το αντίστροφο

x	-3	-2	-1	0	1	2	3
y	-3	-2	-1	0	1	2	3
$z=f(x, y)$							

Αυτό θα λειτουργήσει: αντίστοιχα στοιχεία του xx & yy περιέχουν το x και y συντεταγμένων για να χρησιμοποιηθούν για την $f(x, y)$ σε εκείνο το σημείο

xx	-3	-2	-1	0	1	2	3
-3	-3	-2	-1	0	1	2	3
-3	-3	-2	-1	0	1	2	3
-3	-3	-2	-1	0	1	2	3
-3	-3	-2	-1	0	1	2	3
-3	-3	-2	-1	0	1	2	3
-3	-3	-2	-1	0	1	2	3
yy	-3	-3	-3	-3	-3	-3	-3
-2	-2	-2	-2	-2	-2	-2	-2
-1	-1	-1	-1	-1	-1	-1	-1
0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3
$z=f(xx, yy)$							

3D διαγράμματα – meshgrid

Το Matlab διαθέτει συνάρτηση για τον υπολογισμό αυτών των συστοιχιών :

```
>> x=-3.5:3.5;  
>> y=-2.5:2.5;  
>> [xx,yy]=meshgrid(x,y)  
  
xx =  
    -3.5    -2.5    -1.5    -0.5     0.5     1.5     2.5     3.5  
    -3.5    -2.5    -1.5    -0.5     0.5     1.5     2.5     3.5  
    -3.5    -2.5    -1.5    -0.5     0.5     1.5     2.5     3.5  
    -3.5    -2.5    -1.5    -0.5     0.5     1.5     2.5     3.5  
    -3.5    -2.5    -1.5    -0.5     0.5     1.5     2.5     3.5  
  
yy =  
    -2.5    -2.5    -2.5    -2.5    -2.5    -2.5    -2.5    -2.5  
    -1.5    -1.5    -1.5    -1.5    -1.5    -1.5    -1.5    -1.5  
    -0.5    -0.5    -0.5    -0.5    -0.5    -0.5    -0.5    -0.5  
     0.5     0.5     0.5     0.5     0.5     0.5     0.5     0.5  
     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5  
     2.5     2.5     2.5     2.5     2.5     2.5     2.5     2.5
```

NOTE #1:

xx μεταβάλλεται κατά ΣΕΙΡΕΣ ενώ yy μεταβάλλεται κατά ΣΤΗΛΕΣ

NOTE #2:

Για κάθε (i, j), η τιμή xx είναι η συντεταγμένη x, ενώ η τιμή στην yy είναι η συντεταγμένη y:

$$xx(3,1) = -3.5$$

$$yy(3,1) = -0.5$$

Οπότε $(x,y) = (-3.5, -0.5)$

NOTE #3:

Matlab ονομάζει αυτούς τους πίνακες "plaids"

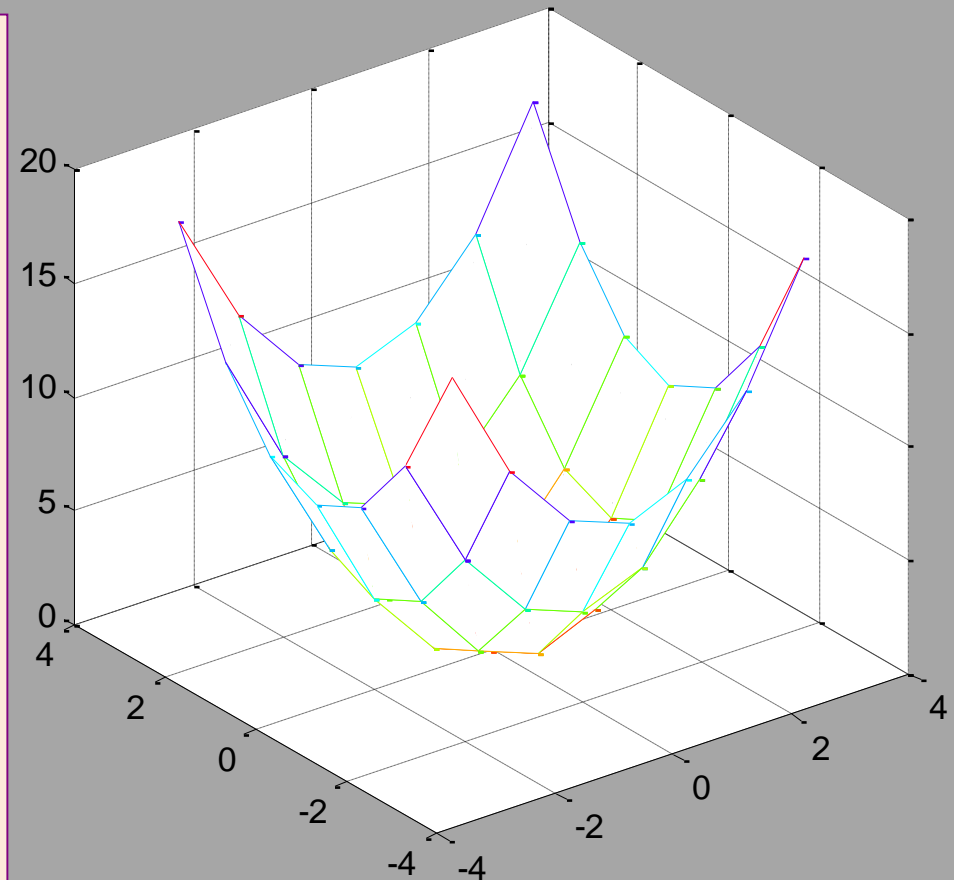
NOTE #4:

Μπορούμε να χρησιμοποιήσουμε τον πίνακα μαθηματικά για να υπολογίσουμε τις τιμές z, όταν $z=f(x,y)$

3D διαγράμματα – meshgrid

Στην περίπτωση που θέλουμε να σχεδιάσουμε την $z = (x^2 + y^2)$ στο διάστημα $-3 \leq x \leq 3$ and $-3 \leq y \leq 3$

```
>> x=-3:3;  
>> y=-3:3;  
>> [xx,yy]=meshgrid(x,y)  
xx =  
     3     -2     -1     0     1     2     3  
    -3     -2     -1     0     1     2     3  
    -3     -2     -1     0     1     2     3  
    -3     -2     -1     0     1     2     3  
    -3     -2     -1     0     1     2     3  
    -3     -2     -1     0     1     2     3  
    -3     -2     -1     0     1     2     3  
yy =  
    -3    -3    -3    -3    -3    -3    -3  
    -2    -2    -2    -2    -2    -2    -2  
    -1    -1    -1    -1    -1    -1    -1  
     0     0     0     0     0     0     0  
     1     1     1     1     1     1     1  
     2     2     2     2     2     2     2  
     3     3     3     3     3     3     3  
>> zz=xx.^2 + yy.^2;  
>> mesh(xx,yy,zz)
```



NOTE:

Απλός υπολογισμός των τιμών της συνάρτησης...

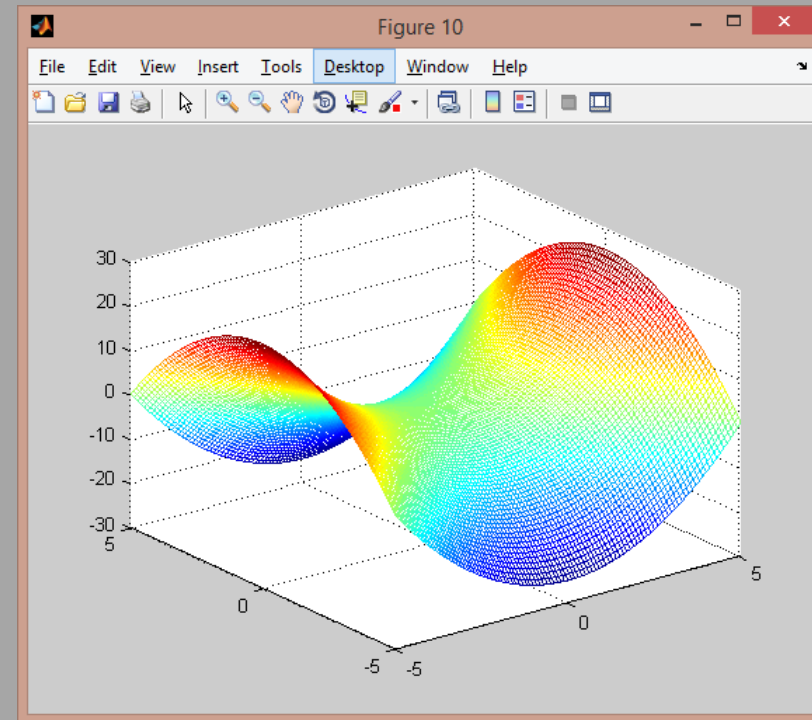
meshgrid

$[X,Y] = \text{meshgrid}(x,y)$: Δημιουργία «πλέγματος» με όλα τα ζεύγη των τιμών (x,y) : Δημιουργούνται οι δισδιάστατοι πίνακες X και Y , που τα αντίστοιχα στοιχεία τους περιέχουν όλα τα ζεύγη των τιμών των μονοδιάστατων πινάκων x και y , με σκοπό την τρισδιάστατη απεικόνιση της επιφάνειας $z=f(x,y)$ μέσω της κατάλληλα διαμορφωμένης εξίσωσης $z=f(x,y)$.

Η χάραξη επιφανειών στο MATLAB γίνεται με παρόμοιο τρόπο με εκείνη των γραμμών:

```
>>[x y]=meshgrid(-5:.1:5,-5:.1:5);  
>>z=x.^2-y.^2;  
>>mesh(x,y,z)
```

Όπου τα **x**, **y**, και **z** είναι μητρώα διάστασης 101×101 .

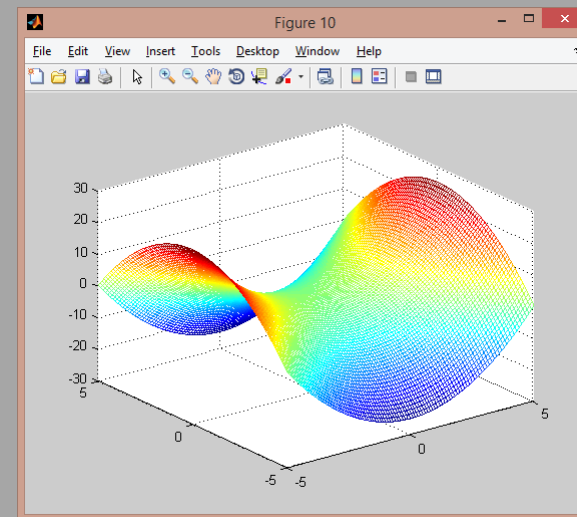


meshgrid

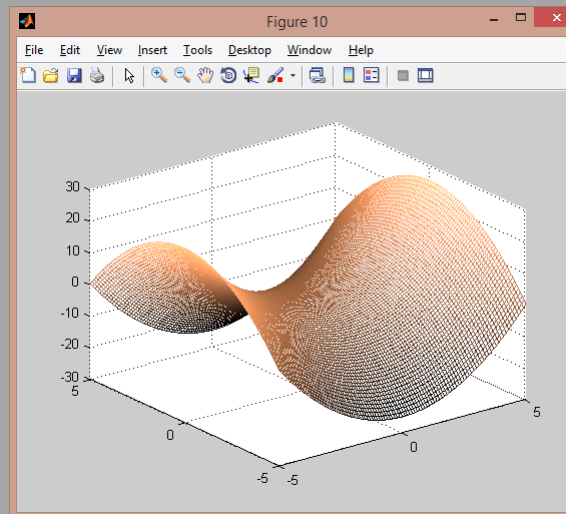
surf(X,Y,Z) : Δημιουργία της επιφάνειας που ορίζεται από τις τριάδες που βρίσκονται στα αντίστοιχα στοιχεία των δισδιάστατων πινάκων X, Y και Z. Για δημιουργία «διαφάνειας», χρησιμοποιείται η εντολή **alpha**.

Τριδιάστατα γραφήματα

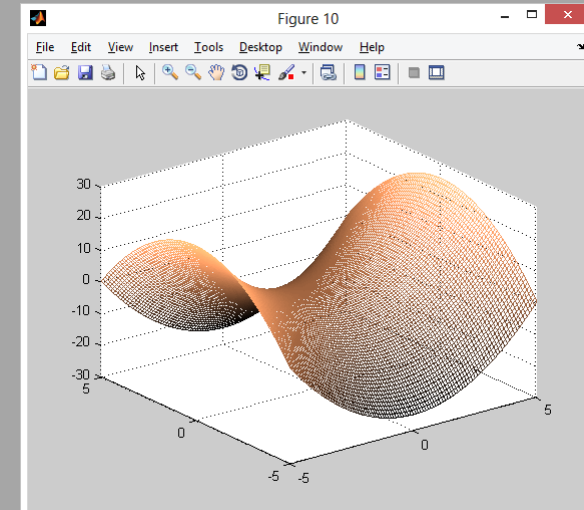
```
>> figure(10)  
>> [x y]=meshgrid(-5:1:5,-5:1:5);  
>> z=x.^2-y.^2;  
>> mesh(x,y,z)
```



```
>> colormap(copper)
```



```
>> alpha(0.1);
```



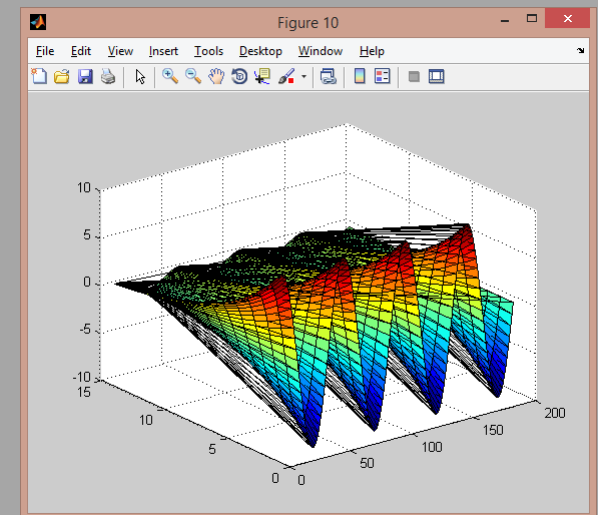
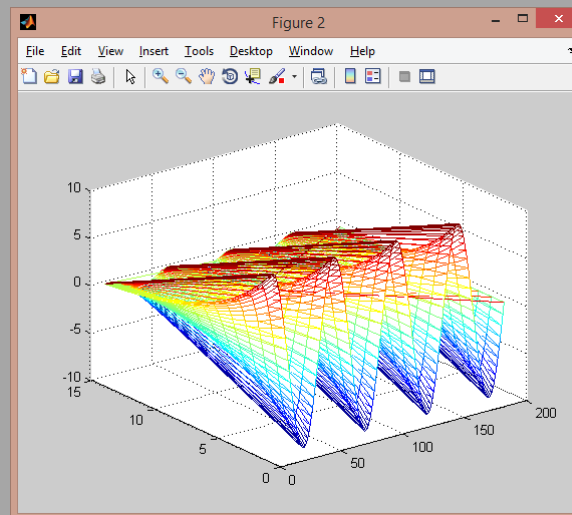
meshgrid

Ας υποθέσουμε ότι θέλουμε να οπτικοποιήσουμε την έκφραση:

$$Z = 10e^{(-0.4a)} \sin(2\pi ft) \quad \text{για } f = 2$$

Όπου τα **a** και **t** λαμβάνουν τιμές στα διαστήματα (0.1-7) και (0.1-2), αντίστοιχα:

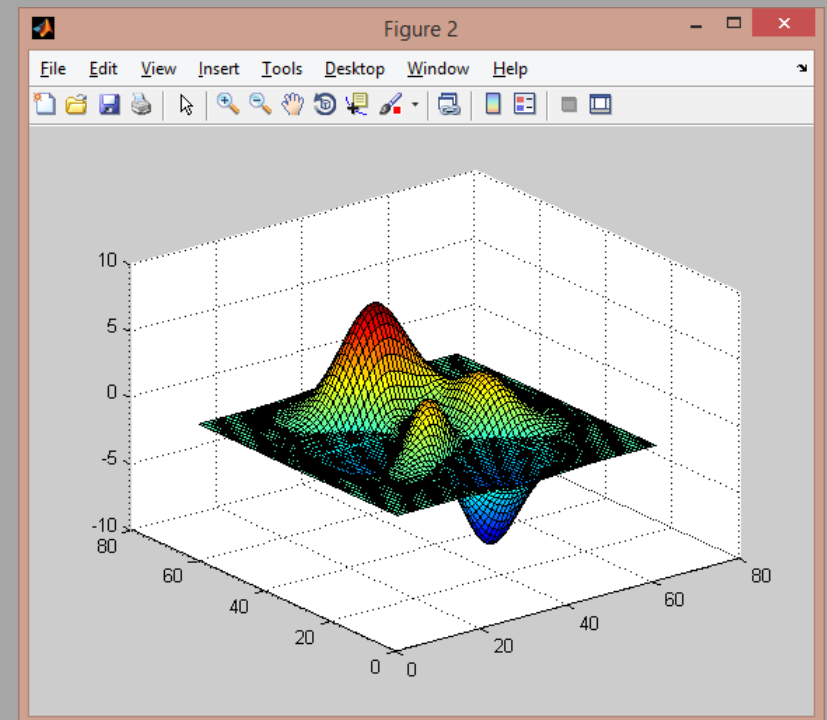
```
>> [t,a] = meshgrid(0.1:.01:2, 0.1:0.5:7);  
>> f=2;  
>> Z = 10.*exp(-a.*0.4).*sin(2*pi.*t.*f);  
>> surf(Z);  
>> figure(2);  
>> mesh(Z);
```



meshgrid

Παράδειγμα *mesh* και *surf* – 3D γραφήματα:

```
>>> [x,y] = meshgrid(-3:.1:3,-3:.1:3);  
>>> z = 3*(1-x).^2.*exp(-(x.^2) - (y+1).^2) ...  
- 10*(x/5 - x.^3 - y.^5).*exp(-x.^2-y.^2) ...  
- 1/3*exp(-(x+1).^2 - y.^2);  
  
>>> surf(z);
```



3D διαγράμματα – υπολογισμός της συνάρτησης

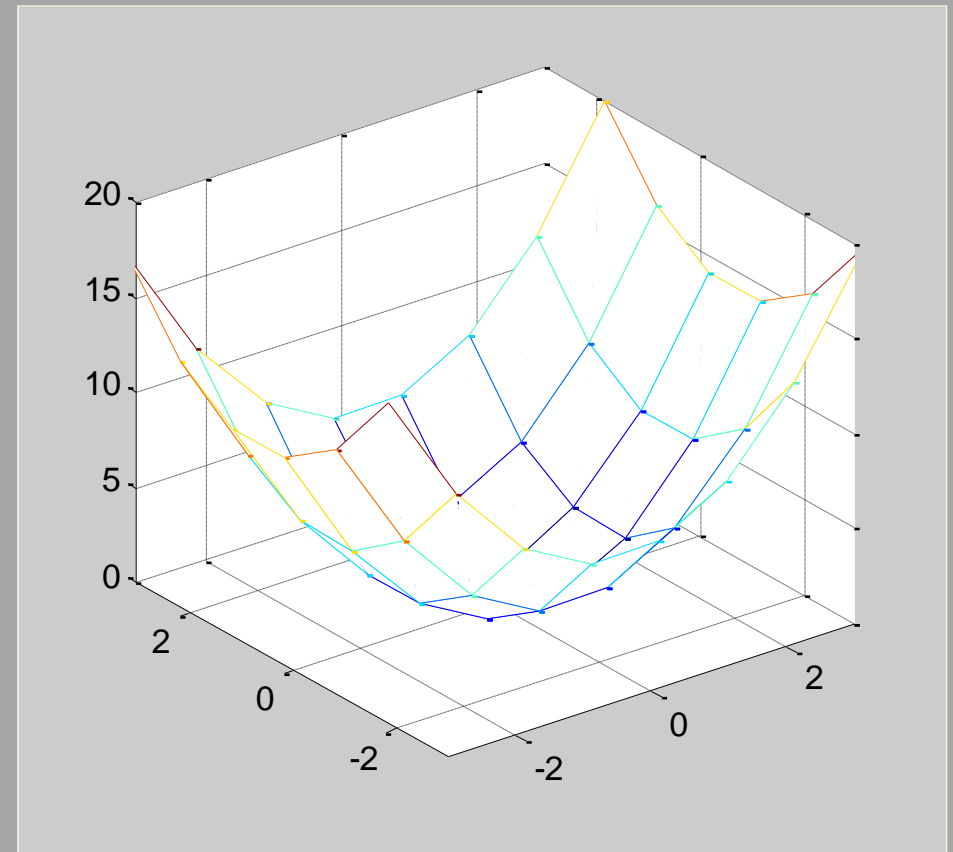
Ο υπολογισμός της $f(x, y)$ μπορεί να είναι δύσκολος, ανάλογα με το πώς ακριβώς ορίζεται ο πίνακας "plaid" (περισσότερα για βρόχους σε επόμενο μάθημα)

Column-wise Calculation of $z = f(x, y)$

```
>> nc=length(x); % number of columns in z
>> nr=length(y); % number of rows in z
>> z=zeros(nr,nc); % initialize z (for speed)
>> for k=1:nc
    z(:,k) = x(k).^2 + y(:).^2;
end
>> mesh(xx,yy,z)
```

NOTE:

Μπορεί να είναι πολύ δύσκολο να παρακολουθείτε τους δείκτες σειράς και στήλης σε αυτού του είδους υπολογισμούς...



3D διαγράμματα – υπολογισμός της συνάρτησης

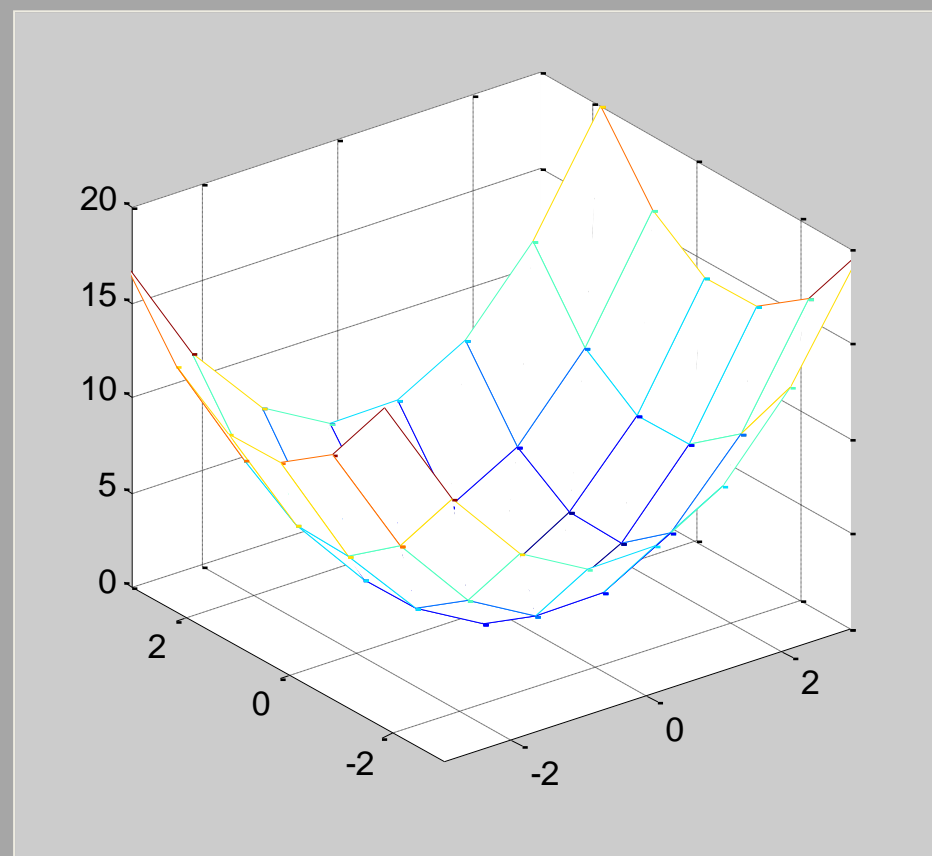
Χειρότερη περίπτωση είναι όταν πρέπει να υπολογιστεί κάθε στοιχείο z ρητά ως εξής :

Element-by-element Calculation of $z = f(x,y)$

```
>> nc=length(x); % number of columns in z
>> nr=length(y); % number of rows in z
>> z=zeros(nr,nc); % initialize z (for speed)
>> for kc=1:nc
    for kr=1:nr
        z(kr,kc) = x(kc).^2 + y(kr).^2;
    end
end
>> mesh(x,y,z)
```

NOTE:

Μπορεί να είναι πολύ δύσκολο να παρακολουθείτε τους δείκτες σειράς και στήλης σε αυτού του είδους υπολογισμούς ...



Ερωτήσεις...