

4. MATLAB'DA GRAFİKLER:

MATLAB grafik sistemi, verilerin hazırlanmasında ve görselleştirilmesinde çok değişik ve kendine has özellikleriyle bir kullanıcılara büyük kolaylık sağlamaktadır.

MATLAB 3.5 ve daha altındaki sürümlerinde; x-y grafikleri, kutupsal grafikler, çubuk grafikler, yüzey grafikleri 3-D yüzey network mesh'i oluşturmak mümkündür. Bunun yanında MATLAB 4.0 ve daha yukarı sürümlerinde grafiksel özellikler çok daha fazla artırılmış ve yukarıdaki bahsedilen özelliklere ilaveten x-y-z grafikleri de dahil olmak üzere her türlü 3 boyutlu çizgi ve yüzey grafikleri oluşturmak mümkün hale gelmiştir.

4.1 2 Boyutlu(2D) Grafikler:

MATLAB'ta verilerin 2 boyutlu olarak çizdirilmesi ve bu grafiklerin düzenlenmesi ile ilgili bir takım hazır fonksiyonlar mevcuttur. Aşağıda bu fonksiyonlar açıklanarak örnekler verilmiştir.

4.1.1 2D Grafik Fonksiyonları:

x-y düzleminde oluşturulan pek çok grafikte x ve y eksenlerinin eşit aralıklarla bölündüğü varsayılır. Ve bu tür grafiklerde lineer grafik adını alır. Bazı istisnai durumlarda ya tek eksen üzerinde ya da her iki eksen (x,y) üzerinde de logaritmik ölçeklendirme kullanılabilir.

x ve y vektörlerinin lineer ve logaritmik fonksiyonları aşağıda verilmiştir (eksenler veri sayısına göre ölçeklendirilirler).

Plot

amaç: 2-D lineer çizim

kullanım:

```
plot(Y)
```

```

plot(X1,Y1,...)
plot(X1,Y1,ÇizgiTürü,...)
plot(...,'Özellikismi',ÖzellikDeğeri,...)
h = plot(...)

```

açıklama:

x ve y eksenlerini lineer olarak ölçeklendirir ve verilerin grafiğini çizer.

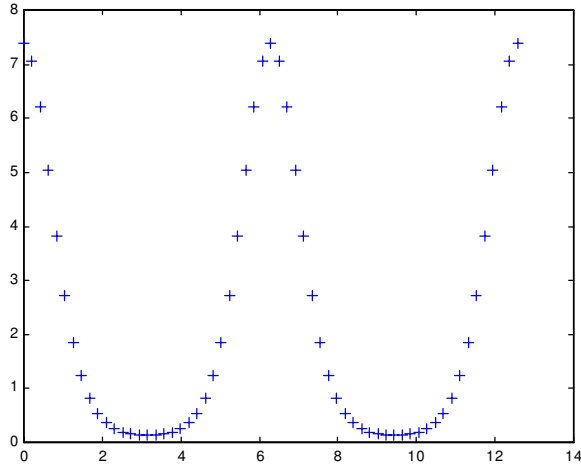
Burada x'ler bağımsız değişken y'ler ise bağımlı değişkeni göstermektedir.

Örnek:

```

X = 0:pi/15:4*pi;
Y = exp(2*cos(X));
plot(X,Y,'b+') herbir veri noktasına mavi artı işareti koyarak çizer

```



Şekil 4.1

Loglog

amaç: Log-log ölçeği çizmek

kullanım:

```

loglog(Y)
loglog(X1,Y1,...)
loglog(X1,Y1,ÇizgiTürü,...)
loglog(...,'Özellikismi', ÖzellikDeğeri,...)
h = loglog(...).plot(Y)

```

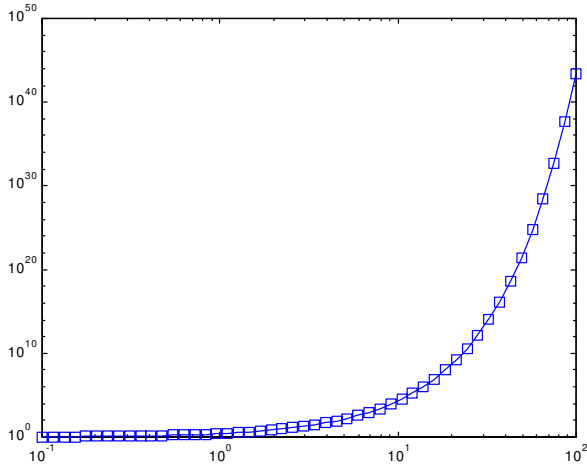
açıklama:

Hem x eksenini hem de y eksenini için logaritmik ölçeklendirme kullanarak x ve y değerlerinin grafiğini oluşturur.

Örnek:

Kare işaretlerle basit bir loglog çizimi oluşturalım.

```
x = logspace(-1,2);
loglog(x,exp(x),'-s')
grid on
```



Şekil 4.2

Semilogx ve semilogy

amaç: Semi-logaritmik grafik çizmek

kullanım:

```
semilogx(Y)
semilogx(X1,Y1,...)
semilogx(X1,Y1, ÇizgiTürü,...)
semilogx(...,'Özellikİsmi',ÖzellikDeğeri,...)
h = semilogx(...)
semilogy(...)
h = semilogy(...)
```

açıklama:

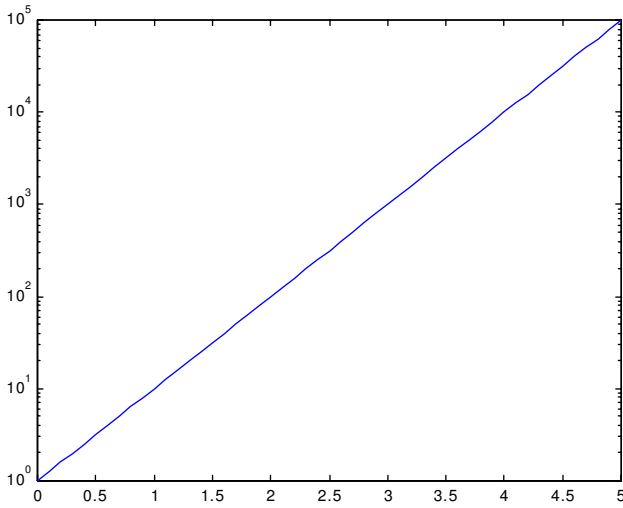
semilogx, x eksenini logaritmik y eksenini lineer olarak ölçeklendirir ve verilerin grafiğini çizer.

semilogy, x eksenini lineer y eksenini doğrusal olarak ölçeklendirir ve verilerin grafiğini çizer.

Örnek:

Örnek bir semilogy plotu:

```
x = 0:.1:10;  
semilogy(x,10.^x)
```



Şekil 4.3

MATLAB'da ana grafik fonksiyonlarının yanında çizilen grafikleri üzerine yazı yazdırabileceğimiz etiket komutları da vardır. Belli başlı etiket komutları ve açıklamaları aşağıda verilmiştir.

Title

amaç: Hali hazırdaki grafiğe başlık eklemek

kullanım:

```
title('başlık yazısı')  
title(fismi)  
title(..., 'Özellikİsmi', ÖzellikDeğeri, ...)
```

```
h = title(...)
```

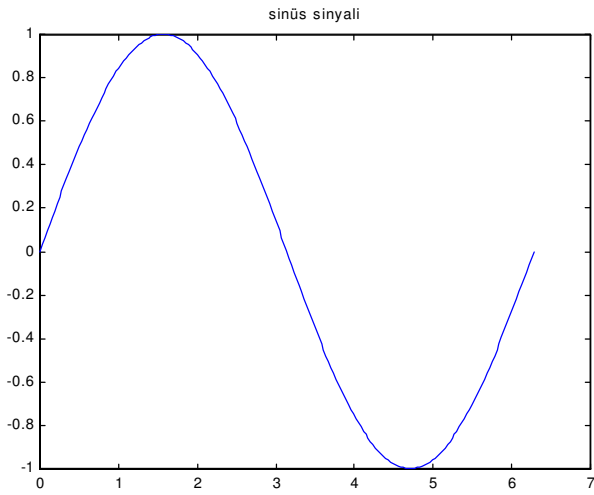
açıklama:

Halihazırdaki grafiğin en üst kısmına merkezleyerek tırnak içindeki kelime ya da kelime grubunu yazdırır.

Örnek:

2π periyotlu bir sinüs sinyali çizdirerek üzerine 'sinüs sinyali' diye yazılması:

```
t = 0:pi/100:2*pi;
y = sin(t);
plot(t,y)
```



Şekil 4.4

xlabel ve ylabel

amaç: x ve y eksenleri için etiket tanımlamak

kullanım:

```
xlabel('yazı')
xlabel(fismi)
xlabel(...,'Özellikİsmi',ÖzellikDeğeri,...)
h = xlabel(...)

ylabel(...)
h = ylabel(...)
```

açıklama:

xlabel, Halihazırdaki grafiğin x eksenini için etiket yazdırır.

ylabel, Halihazırdaki grafiğin y eksenini için etiket yazdırır.

Text

amaç: x ve y eksenleri için etiket tanımlamak

kullanım:

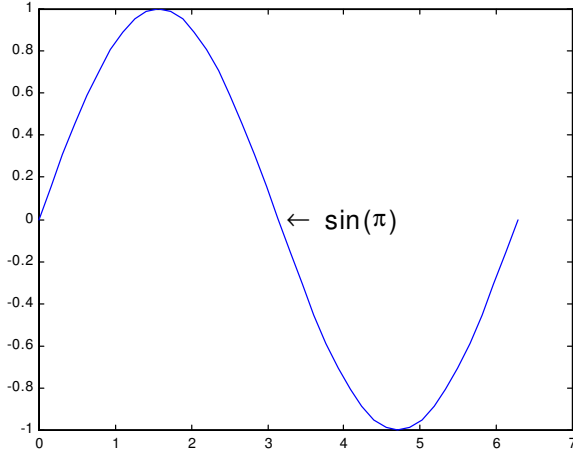
```
text(x,y,'yazı')
text(...'Özellikİsmi',ÖzellikDeğeri...)
h = text(...)
```

açıklama:

Halihazır grafikten alınan eksenleri kullanarak (x,y) koordinatları yolu ile belirlenen grafik eksenine üzerine bir etiket yazdırır.

Örnek:

```
plot(0:pi/20:2*pi,sin(0:pi/20:2*pi))
text(pi,0,' \leftarrow sin(\pi)', 'FontSize',18)
```



Şekil 4.5

Gtext

amaç: Fare yardımıyla grafik üzerine metin yerleştirme

kullanım:

```
gtext('metin')
h = gtext('metin')
```

açıklama:

Bu komut grafik ekran üzerinde fare veya ok tuşları ile gösterilen noktada bir metin veya metinler yazar.

Örnek:

```
gtext('Bu nota dikkat et!')
```

Grid

amaç: Izgara çizgilerini gösterme

kullanım:

```
grid on
grid off
grid
```

açıklama:

Bu komut üzerinde çalışılan grafik için yatay-dikey ölçeklendirme çizgilerini(grid=ızgara) çizdirir.

grid on, üzerinde çalışılan alanda ızgaraları aktif hale getirir.

grid off, aktif olan ızgaraları geri kaldırır.

4.1.2 Bir plotun oluşturulması:

Eğer y bir vektör ise plot(y) fonksiyonu y'nin elemanlarını indislerine göre lineer olarak çizdirir. Argüman olarak iki vektör belirtilirse plot(x,y) fonksiyonu x'e göre y'nin lineer fonksiyonunu çizer.

Bu fonksiyon ile çoklu veri kümelerini çizimin özelliklerini belirterek çizdirmek mümkündür.

Bir plot oluşturulurken izlenecek adımlar ve ilgili örnekler:

Tablo 4.1

Adımlar	İlgili Örnek
1.Verinin hazırlanması	x = 0:.2:12; y1 = bessell(1,x); y2 = bessell(2,x); y3 = bessell(3,x);
2.Pencereyi ve çizdirilecek pozisyonunu seçilmesi	figure(1) subplot(2,2,1)
3.Çizim fonksiyonunu girilmesi	h = plot(x,y1,x,y2,x,y3);
4.Çizgi ve işaret özelliklerinin seçilmesi	set(h,'Çizgigenişliği',2,{ 'Çizstili'},{'-',' ':'-.'}) set(h,{ 'Renk'},{'r','g','b'})
5.Eksen sınırları, tick izleri ve ızgara çizgilerinin belirlenmesi	axis([0 12 -0.5 1]) grid on
6.Eksen etiketi, açıklayıcı bilgi ve text ile grafiğe notların düşülmesi	xlabel('Zaman') ylabel('Yükselme') legend(h,'Birinci','İkinci','Üçüncü') title('Bessel Fonksiyonlar') [y,ix] = min(y1); text(x(ix),y,'First Min \rightarrow',... 'HorizontalAlignment','right')
7.Grafik çıktısı al	print -dps2

4.1.3 Grafik Çiziminde Çizgi Stilleri, İşaretler ve Renkler:

plot(x,y) komutu x ve y vektörleri ile temsil edilen noktaları çizgi dilimleri ile birleştirerek bir çizgi grafiği oluşturur. Kesikli, noktalı, noktalı - kesikli gibi diğer çizgi türlerini de seçmek mümkündür. Diğer taraftan bir çizgi grafiği yerine bir nokta grafiği de çizdirmek mümkündür.

Aşağıdaki tabloda çizgi ve işaret tiplerinin bir listesi verilmiştir.

Tablo 4.2

Sembol	Renk(RGB)	Çizgi stili	Sembol	Nokta stili	
Y	sarı(110)	.	nokta	-	Çizgi
M	magenta(101)	O	yuvarlak	:	Noktalı
C	ciyan(011)	X	çarpı işareti	-.	çizgili ve noktalı
R	kırmızı(100)	+	artı işareti	- -	kesik çizgili
G	yeşil(010)	*	yıldız		

B	mavi(001)	S	karekök		
W	beyaz(111)	D	baklava		
K	siyah(000)	V	üçgen(aşağı)		
		^	üçgen(yukarı)		
		<	üçgen(sola)		
		>	üçgen(sağa)		
		P	pentagram		
		H	hexagram		

Örnek:

plot(x,y,r*) fonksiyonu girildiğinde çizilen grafik kırmızı renkte(r) ve yıldız (*) işaretleriyle çizilmiş olur.

4.1.4 Matrislerin Çizdirilmesi:

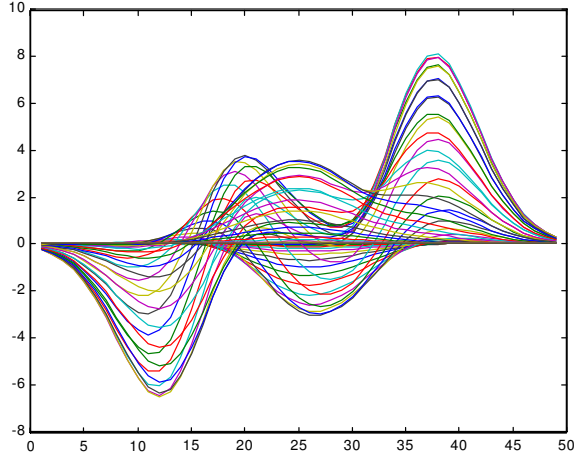
plot fonksiyonu argüman olarak tek bir matris alırsa

```
plot(y)
```

y matrisinin her birisi ayrı ayrı çizdirilir. X-ekseni ise, 1:m'e kadar olan (m:satır sayısı) indis vektörü ile etiketlenir. Örneğin

```
z =peaks
```

iki değişkeli fonksiyonu değerlendirerek 49x49 luk matristen oluşan grafiği çizer.



Şekil 4.6

4.2 3 Boyutlu(3D) Grafikler:

MATLAB'ta verilerin 3 boyutlu olarak çizdirilmesi ve bu grafiklerin düzenlenmesi ile ilgili bir takım hazır fonksiyonlar mevcuttur. Bunlar, yüzeylerin ve kafes çerçevesi şeklindeki ağ grafiklerin çizdirilmesinde kullanılmaktadır. Aşağıda bazı temel 3D grafik fonksiyonlar açıklanarak örnekler verilmiştir(diğer grafik fonksiyonları tezin sonundaki ekte verilmiştir).

4.2.1 3D Grafik Fonksiyonları:

plot3

amaç: 3-D lineer çizim

kullanım:

```
plot3(X1,Y1,Z1,...)
plot3(X1,Y1,Z1, Çizgitürü,...)
plot3(...,'Özellikismi',ÖzellikDeğeri,...)
h = plot3(...)
```

açıklama:

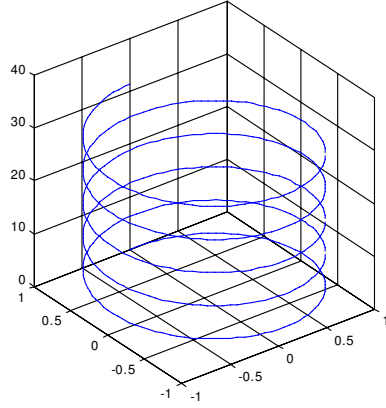
3 boyutlu uzayda noktaların ve çizgilerin grafiğini oluşturur. plot(x,y,z) komutu 3 boyutlu uzayda koordinatları x,y ve z'nin elemanları olan

noktalardan geçen tek bir çizgi grafiği oluşturur. Burada x,y ve z'nin aynı boyutta vektörler olması gerekir. çizer.

Örnek:

Üç boyutlu helazonun çizilmesi:

```
t = 0:pi/50:10*pi;
plot3(sin(t),cos(t),t)
grid on; axis square
```



Şekil 4.7

Mesh,meshc,meshz

amaç: Ağ çizimlerini oluşturmak

kullanım:

```
mesh(X,Y,Z)
mesh(Z)
mesh(...,C)
meshc(...)
meshz(...)
h = mesh(...)
h = meshc(...)
h = meshz(...)
```

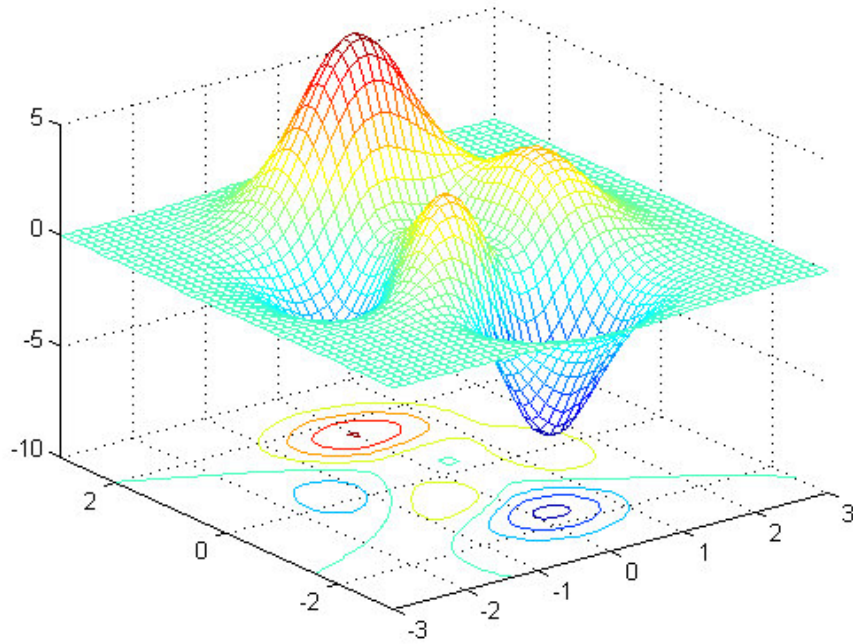
açıklama:

mesh , meshc ve meshz , matris elemanlarının 3 boyulu olarak bşr perspektifinin çizdirirler.

Örnek1:

Birleşme ağını üret ve tepe noktaların kenar çiziminin yapılması.

```
[x,y] = meshgrid(-3:.125:3);
Z = peaks(X,Y);
meshc(X,Y,Z);
axis([-3 3 -3 3 -10 5])
```

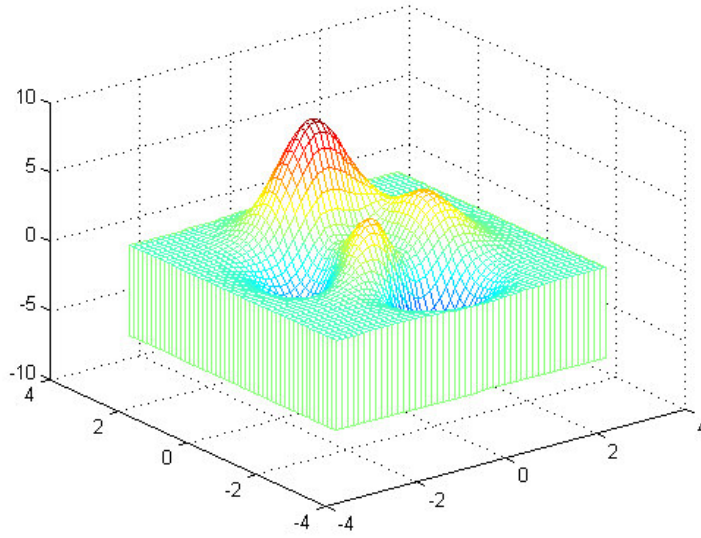


Şekil 4.8

Örnek2:

Tepe noktalar fonksiyonu için perde çiziminin üretilmesi .

```
[X,Y] = meshgrid(-3:.125:3);
Z = peaks(X,Y);
meshz(X,Y,Z)
```



Şekil 4.9

Pcolor

amaç: Renkleri matris elemanlarıyla hesaplanan bir dörtgensel hücre dizisinin çizdirilmek.

kullanım:

```
pcolor(C)
pcolor(X,Y,C)
h = pcolor(...)
```

açıklama:

pseudocolor çizimi , 'C' tarafından belirlenmiş renkler ile hücrelerin dikdörtgen biçiminde sıralanmasıdır .

Surface

amaç: Yüzey nesnesi oluşturmak

kullanım:

```
surface(Z)
surface(Z,C)
surface(X,Y,Z)
```

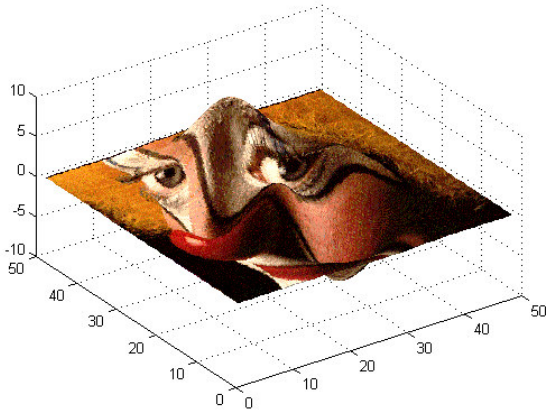
```
surface(X,Y,Z,C)
surface(...'PropertyName',PropertyValue,...)
h = surface(...)
```

açıklama:

surface, yüzey nesnesi oluşturmak için düşük seviyeli bir grafik fonksiyonudur .

Örnek:

```
load clown
surface(peaks,flipud(X),...
        'FaceColor','texturemap',...
        'EdgeColor','none',...
        'CDataMapping','direct')
colormap(map)
view(-35,45)
```



Şekil 4.10

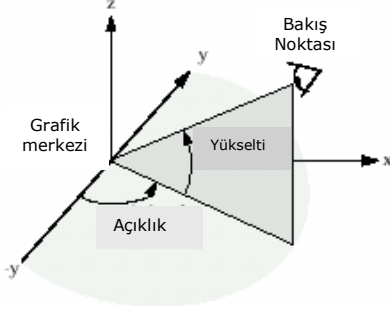
4.2.2 3-Boyutlu Bir Grafiğe Bakış Açısı:

MATLAB, grafiğin yönünü kontrol etmenizde kolaylık sağlar. MATLAB'ın bir grafik için görüş açısını belirtebilir, yönü ve görünüşün izdüşümünü şekil penceresinde görebilirsiniz. Özellikleri görerek bunları bir takım grafik fonksiyonları yoluyla kontrol edebilirsiniz.

`view` fonksiyonu, küresel koordinatlarda eksenlerin merkezine göre (orjine) açılık ve yükselti açıları belirtilerek bakış noktasını ayarlar.

- Açıklık, x-y düzleminde saat yönünde pozitif değerli kutupsal açıdır.
- Yükselti ise bakış noktasının x-y düzleminin üzerinde(pozitif) veya altında(negatif)olmasını belirleyen derece cinsinden bir açıdır.

Aşağıdaki çizim, koordinat sistemini gösterir. Oklar, pozitif yönleri ifade etmektedir.



Şekil 4.11

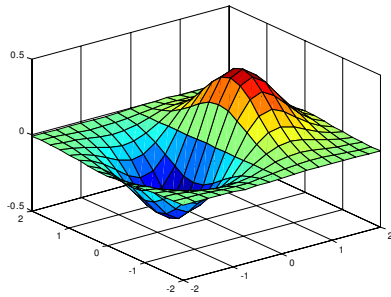
MATLAB , 2-D veya 3-D çizimlerini belirlenmiş görüş açısını otomatik olarak seçer. Buna göre

- 2-D çizimler için, açıklık=0° ve yükselti=90°
- 3-d çizimler için, açıklık=-37.5° ve yükselti=30°

Örnek:Aşağıda peaks matrisinin çizdirilmesinden elde edilen grafiğin dört farklı görünümü yer almaktadır.

1. Açıklık=-37.5° ve yükselti=30° için grafiğin çizimi:

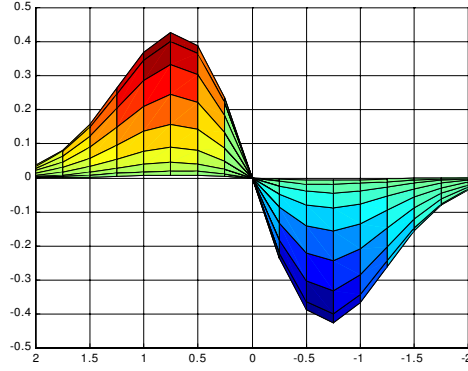
```
[X,Y] = meshgrid([-2:.25:2]);
Z = X.*exp(-X.^2 -Y.^2);
surf(X,Y,Z)
```



Şekil 4.12

2. Açıklık= 180° , yükselti= 0° ve $z=0$ için grafiğin çizimi:

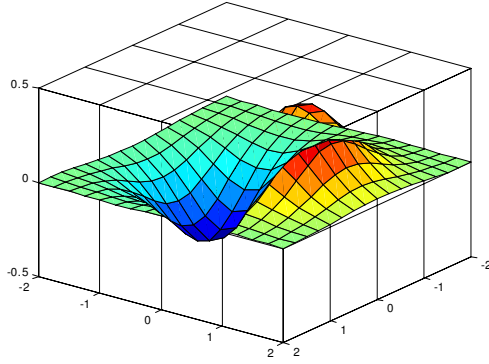
`view[180,0]`



Şekil 4.13

3. Açıklık= -37.5° , yükselti= 30° için grafiğin çizimi:

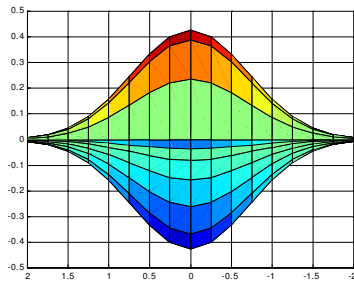
`view([-37.5 -30])`



Şekil 4.14

4. Açıklık= -90° , yükselti= 0° için grafiğin çizimi:

`view([-37.5 -30])`



Şekil 4.15