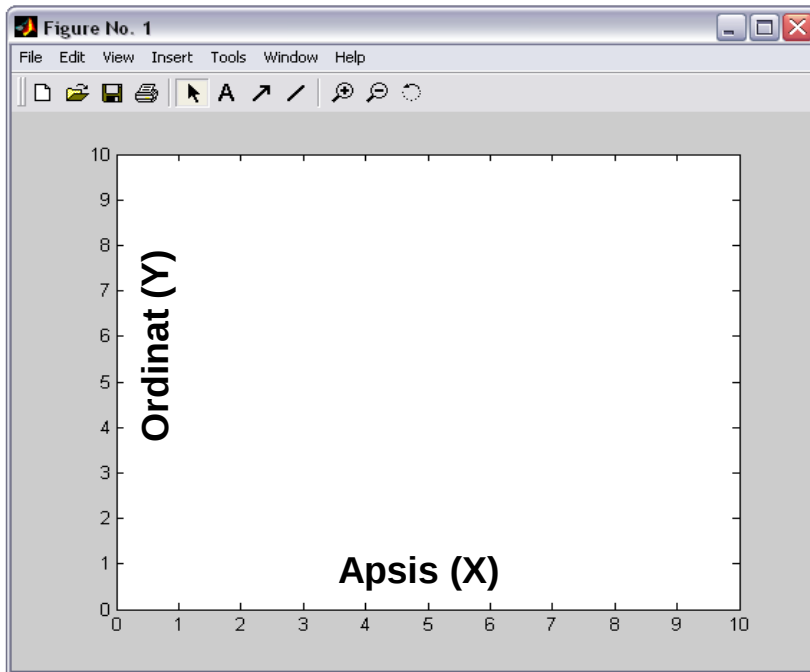


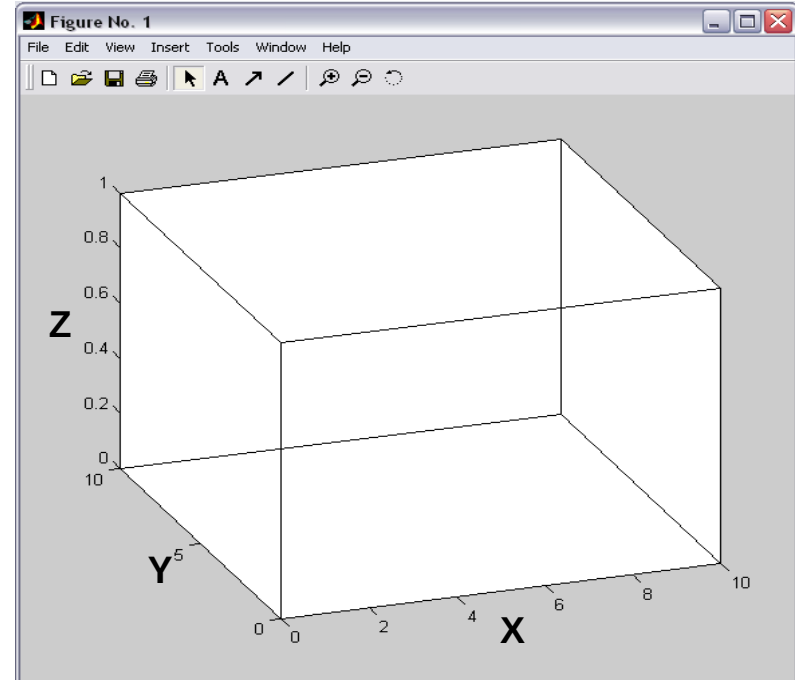
MATLAB/Grafik

- Matlab'de grafikler “figure” penceresinde çizdirilir.
- İki ve üç boyutlu çizim yanı sıra, kutupsal koordinat sisteminde de çizim olanağı bulunur (bak., polar).

İki Boyutlu Koordinat Sistemi



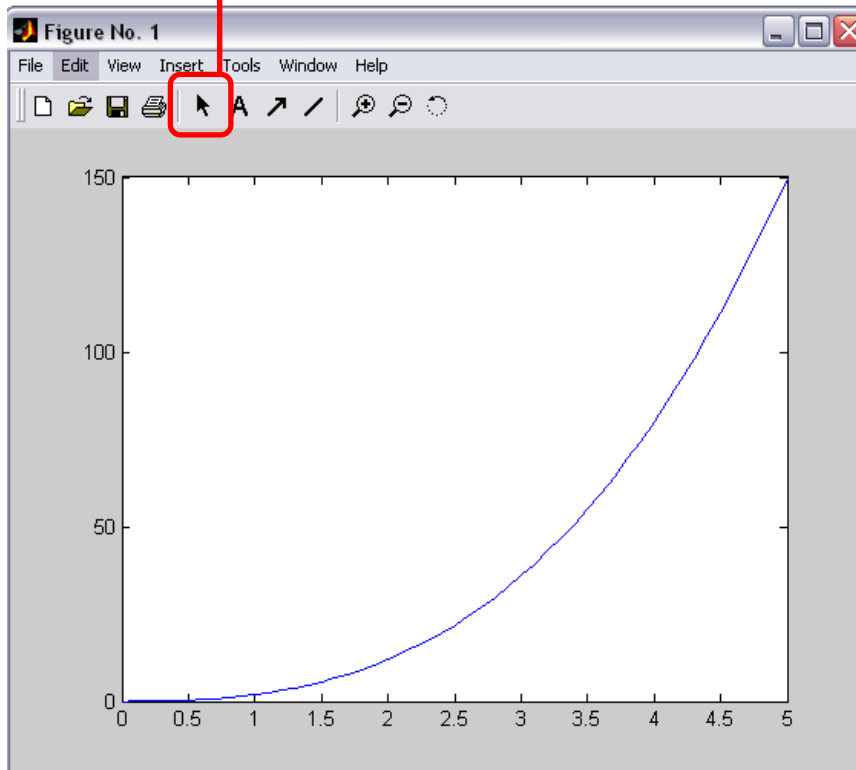
Üç Boyutlu Koordinat Sistemi



MATLAB/Grafik

- Matlab'de en temel çizim fonksiyonu **plot**'dur.
- Örneğin, $x=0:0.1:5$ olan bir dizi vektör elemanlarına karşılık, $y=x.^3+x.^2$ fonksiyon değerleri hesaplatılsın.
- `plot(x,y)` ile aşağıdaki grafik çizdirilir.

Edit plot

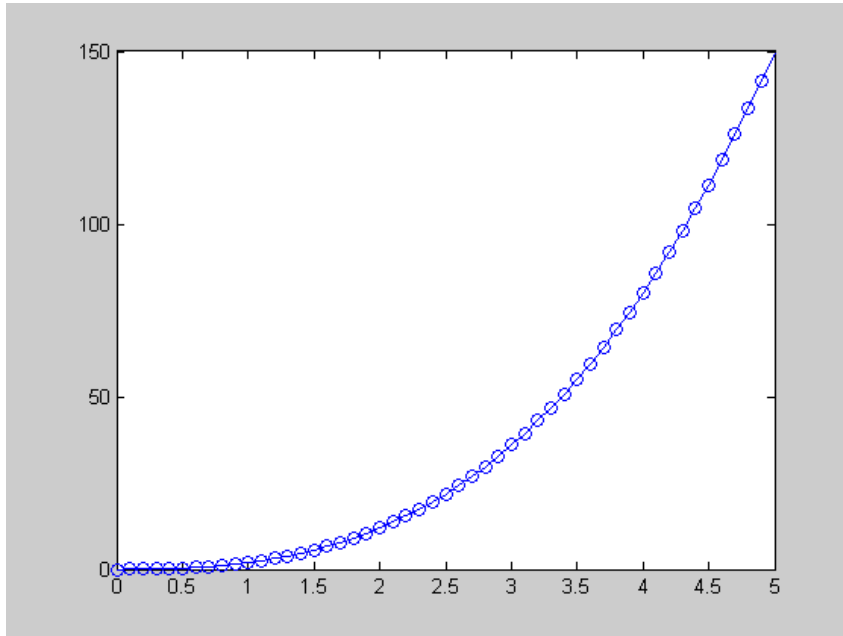


- Çizilen grafiğin üzerinde birçok değişiklik yapmak mümkündür.
- Bunun için **"Edit plot"** düğmesi tıklanır.
- İlgili nesne (çizdirilen eğri, eksenler vb.) iki kez tıklanarak beliren **"Property Editor"** penceresinden istenilen değişiklikler yapılabilir.
- Property Editor penceresinden yapılan her türlü değişikliği, komut olarak yaptırmak mümkündür.

Örneğin, `plot(x,y, '-o')` hem ardışık noktaları şekildeki gibi birleştirir, hem de x,y nokta çiftlerini grafik üzerinde bir "o" sembolü ile işaretler.

MATLAB/Grafik

- `plot(x,y, '-o')` ile ilgili grafik aşağıdaki gibi olur.

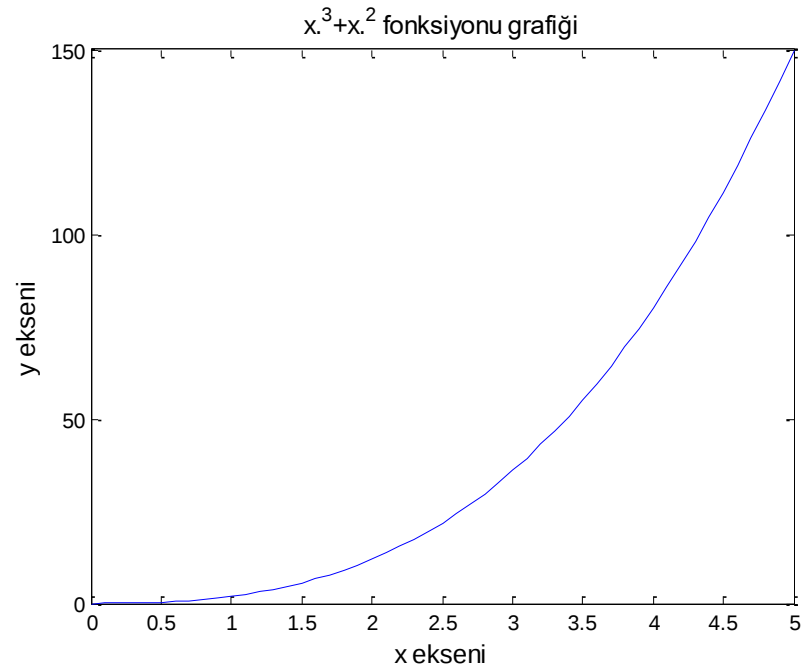


- Aşağıdaki ifadelerle çizimi tekrarlayınız:
`plot(x,y, '-o')`
`plot(x,y, '-*')`
`plot(x,y, '-+')`
`plot(x,y, '-^')`
`plot(x,y, '-.')`
- Sözü edilen grafik üzerindeki o,*,+ gibi sembollere marker denir.
- `plot` fonksiyonu ile ilgili eğrinin rengini değiştirmek de mümkündür:
`plot(x,y, 'r')` kırmızı (red)
`plot(x,y, 'k')` siyah
`plot(x,y, 'b')` mavi (blue)
`plot(x,y, 'g')` yeşil (green)
- `plot(x,y, 'm')` mor (magenta)

MATLAB/Grafik

- Grafik çizimlerinde grafik için başlık oluşturmak, x-y eksen takımlarını isimlendirmek ihtiyaçları duyabiliriz. Bu durumlarda sırası ile “**title**”, “**xlabel**” ve “**ylabel**” fonksiyonları kullanılır.

```
>> x=[0:0.1:5];  
>> y=x.^3+x.^2;  
>> plot(x,y)  
>> title('x.^3+x.^2 fonksiyonu grafiği')  
>> xlabel('x eksen')  
>> ylabel('y eksen')
```



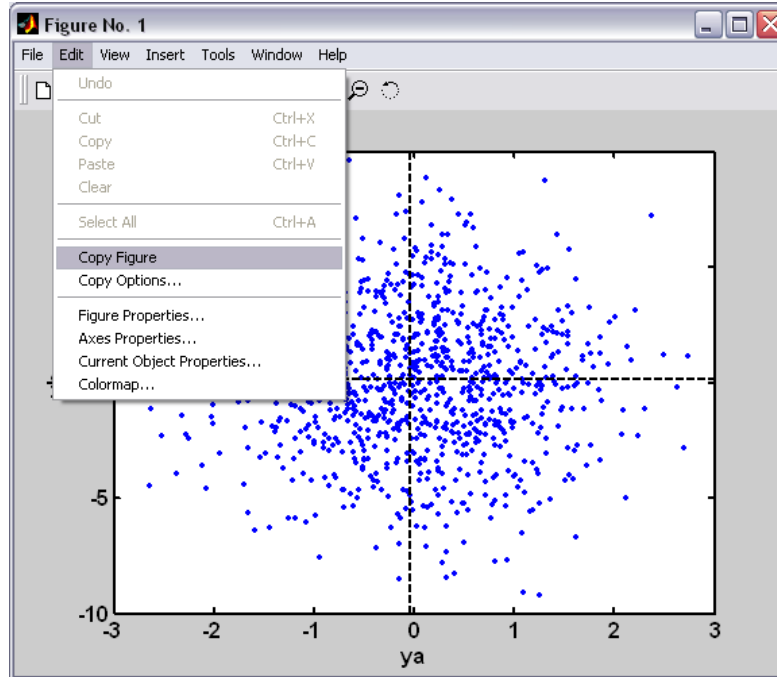
MATLAB/Grafik

- Matlab' da bazı durumlarda değişkenin ve fonksiyonun sadece ilgilenilen aralıklarda değişiminin çizilmesi istenebilir.

<i>Fonksiyon</i>	<i>Tanımı</i>
<code>axis([xmin xmax ymin ymax])</code>	Eksen takımını verilere göre yeniden yapılandırır.
<code>axis equal</code>	Eksen takımlarını eşit ölçekli ve artımlı yapılandırır.
<code>axis square</code>	Eksen takımlarını eşit uzunlukta kare olarak yapılandırır.
<code>axis normal</code>	axis equal ve axis square fonksiyonlarının etkilerini kaldırır.
<code>axis off</code>	Eksen takımlarını, etiketlerini ve eksen üzerindeki sayıları kaldırır.
<code>axis on</code>	Eksen takımlarını ve diğer özellikleri geri getirir.

MATLAB/Grafik-Kaydetme ve kopyalama

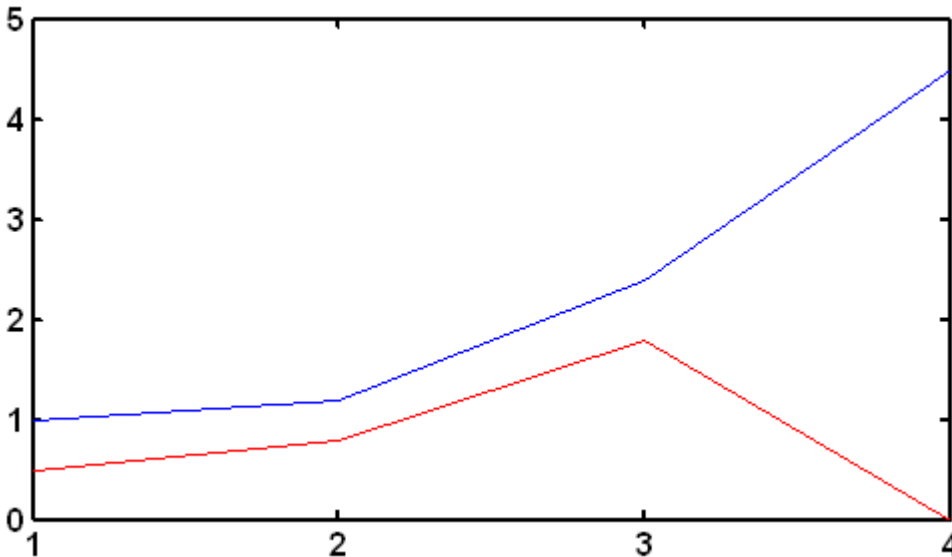
- Çizilen grafikleri kaydetmek için Figure penceresindeki “File” menüsünden “Save” veya “Save As” seçenekleri seçilir. Grafikler, “fig” uzantılı dosyalar olarak ilgili klasöre kaydedilirler.
- Çizilen grafiklerin başka bir ortama aktarılmaları için, “Edit” menüsünün altındaki “Copy Figure” seçeneği seçilir. (Not: Kopyalamanın arka plan rengini ayarlamak için “Copy Options” seçeneğine bakınız.)



MATLAB/Grafik-Aynı eksen takımına farklı grafikler çizdirme

- Aynı eksen takımına farklı grafikleri çizdirmek için **hold on** ve **hold off** komutları kullanılır. Bu iki komut arasına yazılan her türlü grafiğin çizimi aynı eksen takımında gösterilir.
- Örneğin, $y_a = [1; 1.2; 2.4; 4.5]$ ve $y_b = [0.5; 0.8; 1.8; 0]$ vektörleri ile ifade edilen iki farklı ölçü grubunu $x = [1; 2; 3; 4]$ vektörüne göre aynı eksen takımında çizdirmek için aşağıdaki komutları yazmak yeterlidir:

```
hold on, plot(ya), plot(yb, 'r'), hold off
```

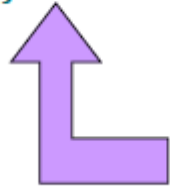


Not: Eğer x eksenini, bu örnekte olduğu gibi, y değerlerinin indisini, yani kaçınıcı değer olduğunu, gösteriyorsa, plot fonksiyonunda x'in yeniden belirtilmesine gerek yoktur.

MATLAB/Grafik-Aynı eksen takımına farklı grafikler çizdirme

- Grafik üzerine **legend** komutu ile açıklama ekleme

legend ('açıklama 1', 'açıklama 2' , konum)



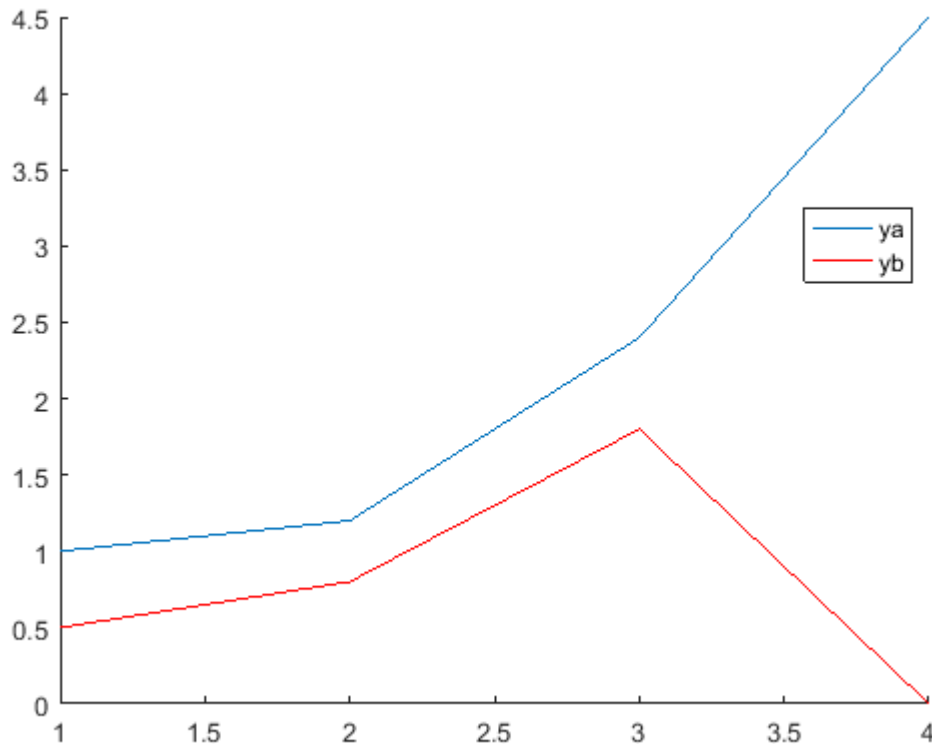
1. grafiğe ait açıklama



2. grafiğe ait açıklama



figüre penceresindeki konum

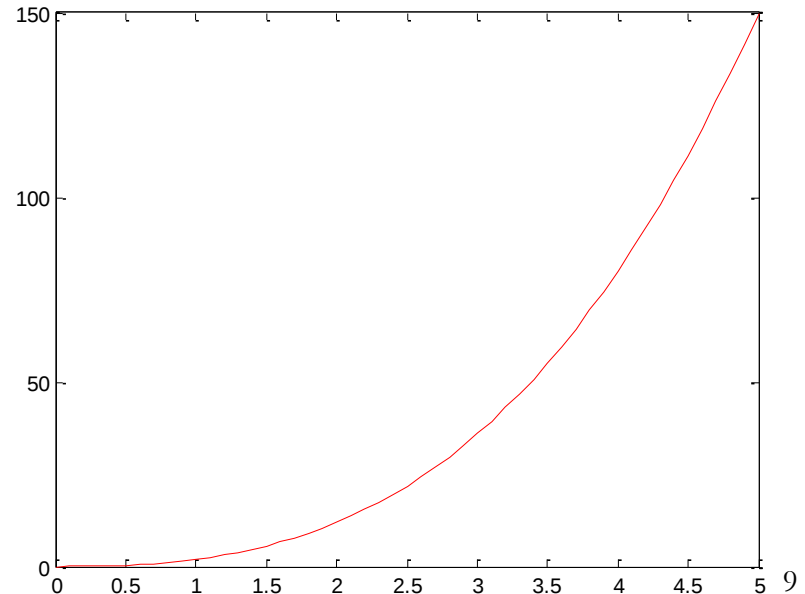
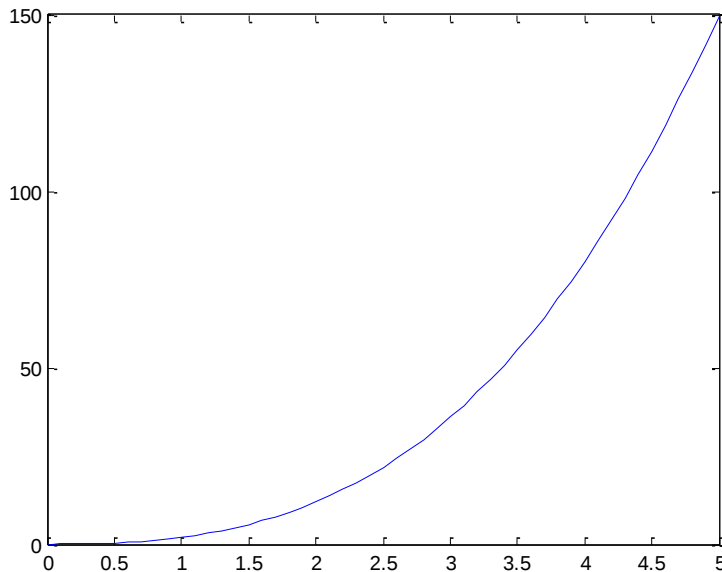


- ya=[1;1.2;2.4;4.5]
- yb=[0.5;0.8;1.8;0]
- x=[1;2;3;4]
-
- hold on
- plot(ya)
- plot(yb,'r')
- hold off
- legend('ya','yb')

MATLAB/Grafik-Çoklu Grafiklerin Oluşturulması

- **figure** fonksiyonu farklı verilere ait grafiklerin farklı pencerelerde gösterilmesine olanak sağlar.

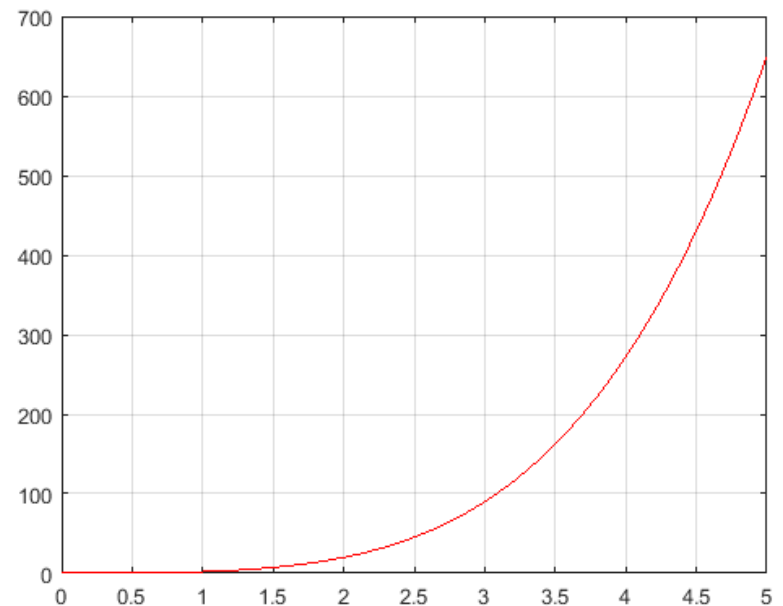
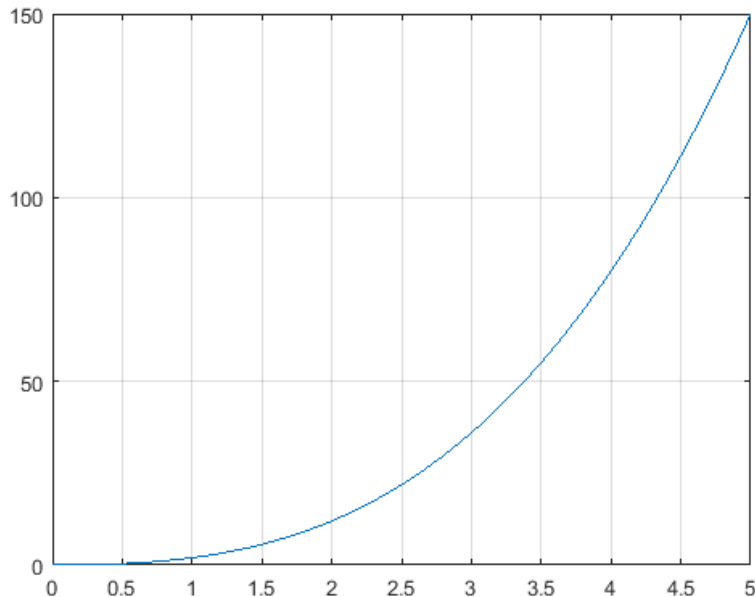
```
x=[0:0.1:5];  
y1=x.^3+x.^2;  
y2=x.^4+x.^2;  
figure(1)  
plot(x,y1)  
figure(2)  
plot(x,y2,'r')
```



MATLAB/Grafik-Çoklu Grafiklerin Oluşturulması

- `grid on` komutu şekildeki grid ağını çizer. `grid off` çizgileri kaldırır.

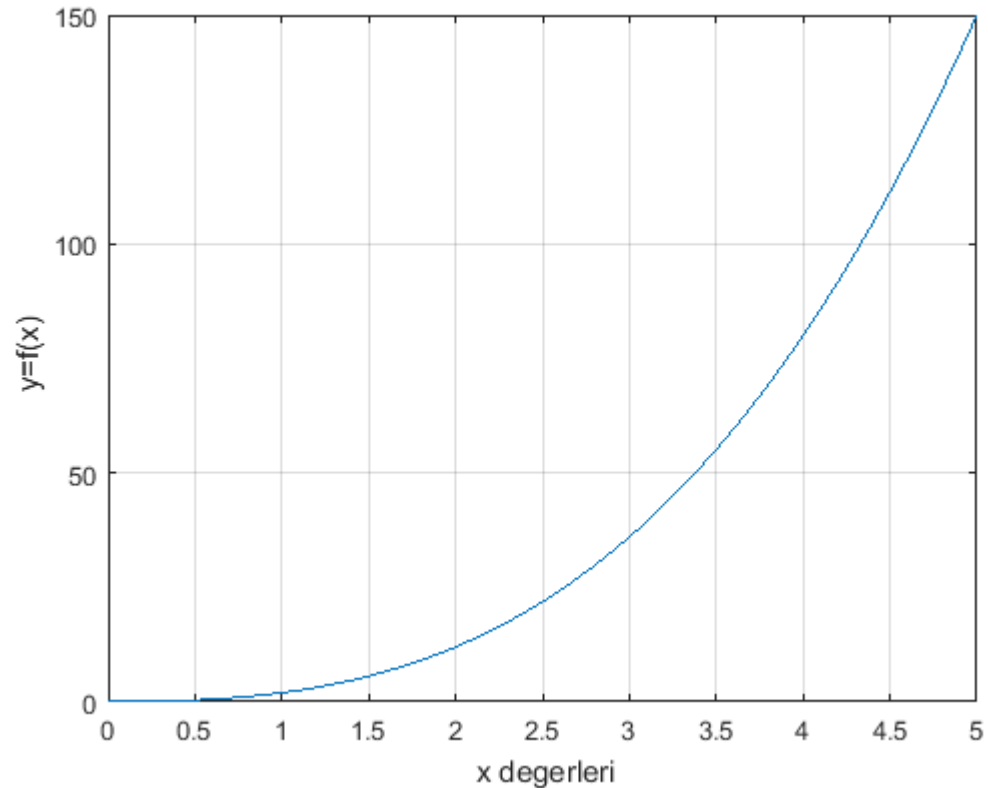
```
x=[0:0.1:5];  
y1=x.^3+x.^2;  
y2=x.^4+x.^2;  
figure(1)  
plot(x,y1)  
grid on  
figure(2)  
plot(x,y2,'r')  
grid on
```



MATLAB/Grafik-Çoklu Grafiklerin Oluşturulması

- `xlabel` `ylabel` komutları x ve y eksenlerinin isimlendirilmesini sağlar.
- `xlabel('x ekseninin etiketi')`
- `ylabel('y ekseninin etiketi')`

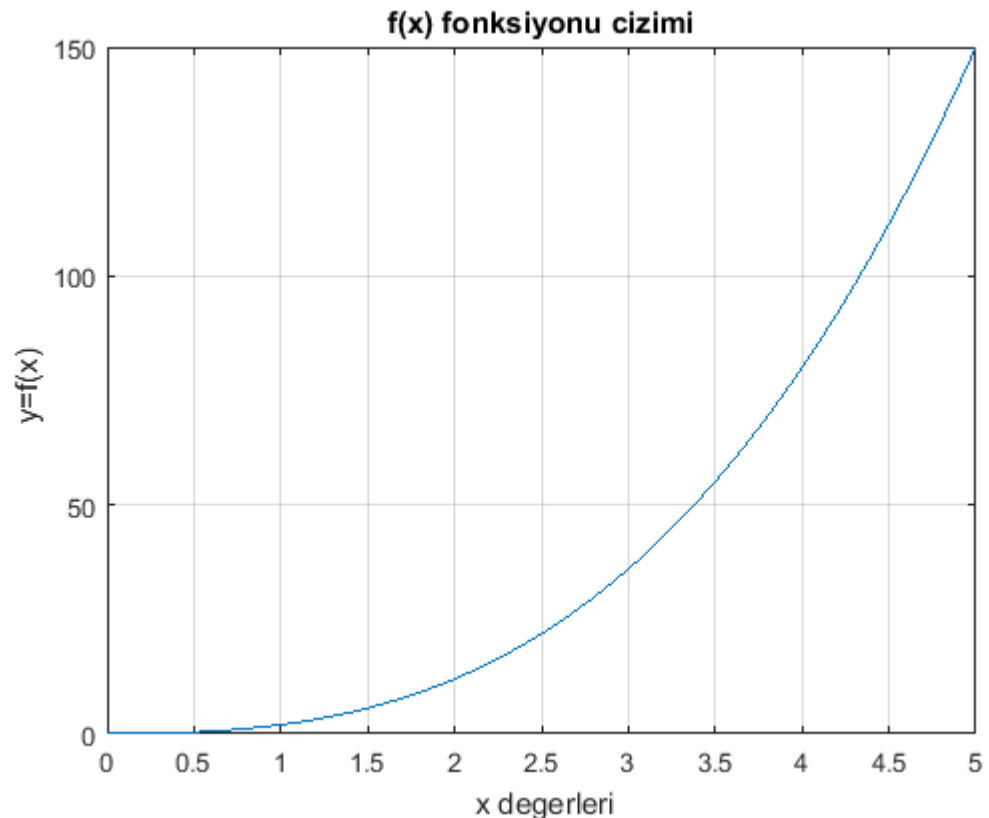
```
x=[0:0.1:5];  
y1=x.^3+x.^2;  
y2=x.^4+x.^2;  
figure(1)  
plot(x,y1)  
grid on  
xlabel(' x degerleri')  
ylabel(' y=f(x)')
```



MATLAB/Grafik-Çoklu Grafiklerin Oluşturulması

- **Title** komutu ile çizilen grafiğe başlık yazdırılması sağlanır.
- **title** ('Grafığın başlığı')

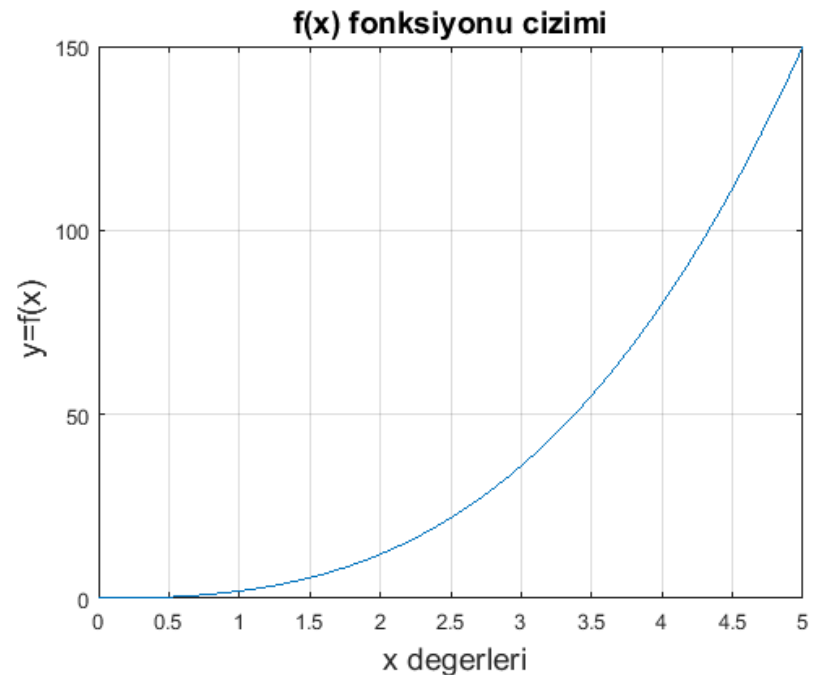
```
x=[0:0.1:5];  
y1=x.^3+x.^2;  
y2=x.^4+x.^2;  
figure(1)  
plot(x,y1)  
grid on  
xlabel(' x degerleri')  
ylabel(' y=f(x)')  
title(' f(x) fonksiyonu cizimi')
```



MATLAB/Grafik-Çoklu Grafiklerin Oluşturulması

- Eksenlerin ve grafiğin başlığının yazı büyüklük (font) ayarı **fontsize** komutu ile gerçekleştirilir
- **fontsize** ('istenen punto büyüklüğü')

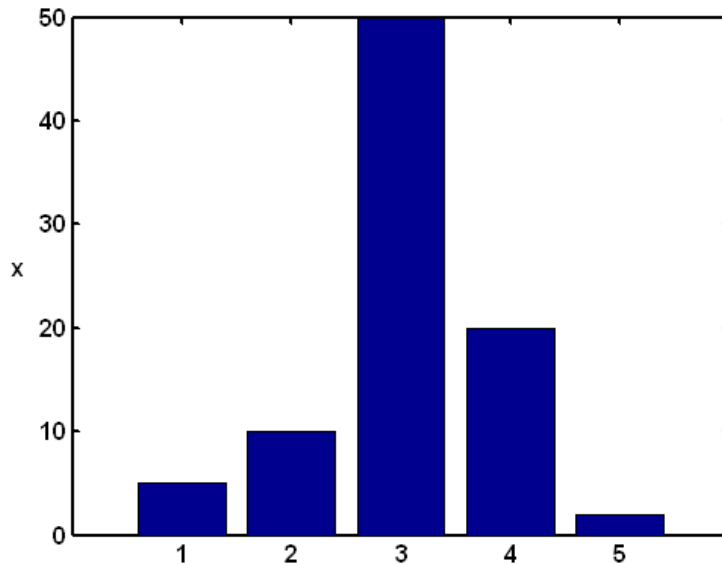
```
x=[0:0.1:5];  
y1=x.^3+x.^2;  
y2=x.^4+x.^2;  
figure(1)  
plot(x,y1)  
grid on  
xlabel(' x degerleri','fontsize',14)  
ylabel(' y=f(x)', 'fontsize',14)  
title(' f(x) fonksiyonu cizimi','fontsize',14)
```



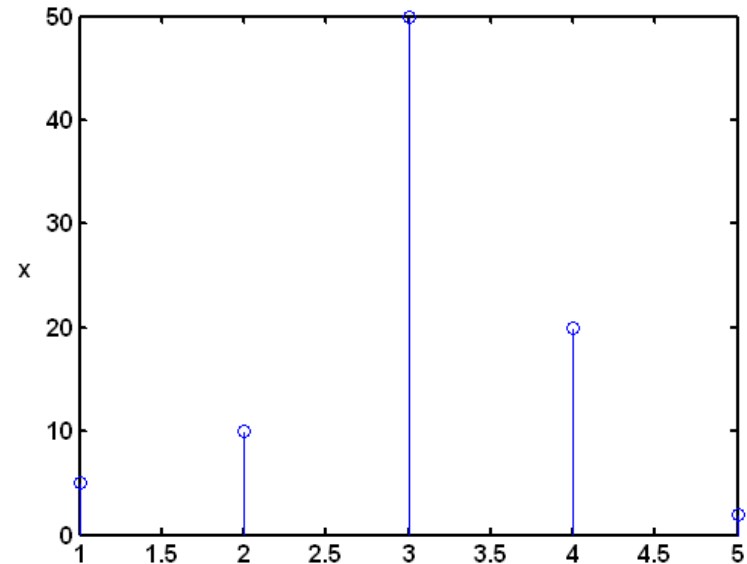
MATLAB/Grafik-Çubuk (bar) ve stem grafiği

- Matlab'de farklı gösterimlere göre çizim yapmak mümkündür. Bunlardan ikisi bar(...) ve stem(...) çizim fonksiyonlarıdır.
- **Örnek:** $\mathbf{x}=[5;10;100;20;2]$ vektör elemanlarının bar ve stem grafik olarak göstermek isteyelim. `bar(x)` ve `stem(x)` aşağıdaki grafikleri çizdirecektir.

Bar grafiği



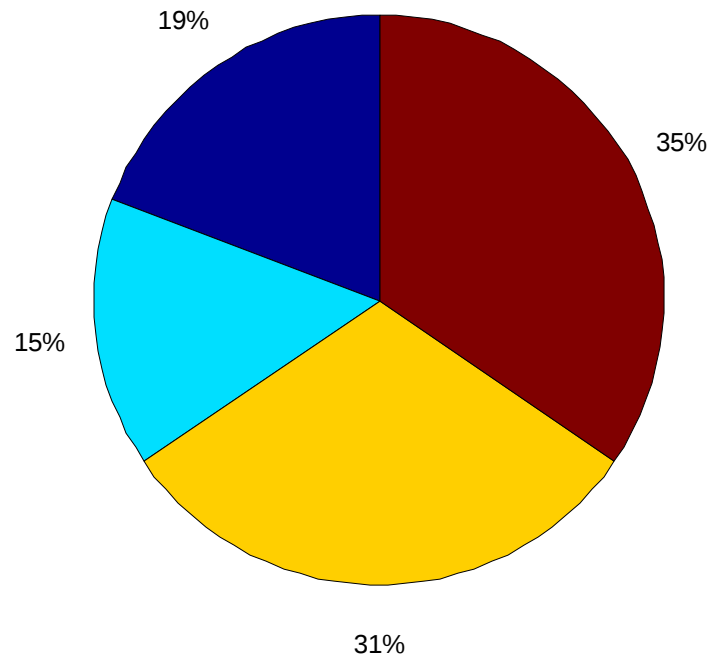
Stem grafiği



MATLAB/Grafik-pie fonksiyonu

- `pie([dizi elemanları])` fonksiyonu dairesel grafikler oluşturmak için kullanılır.

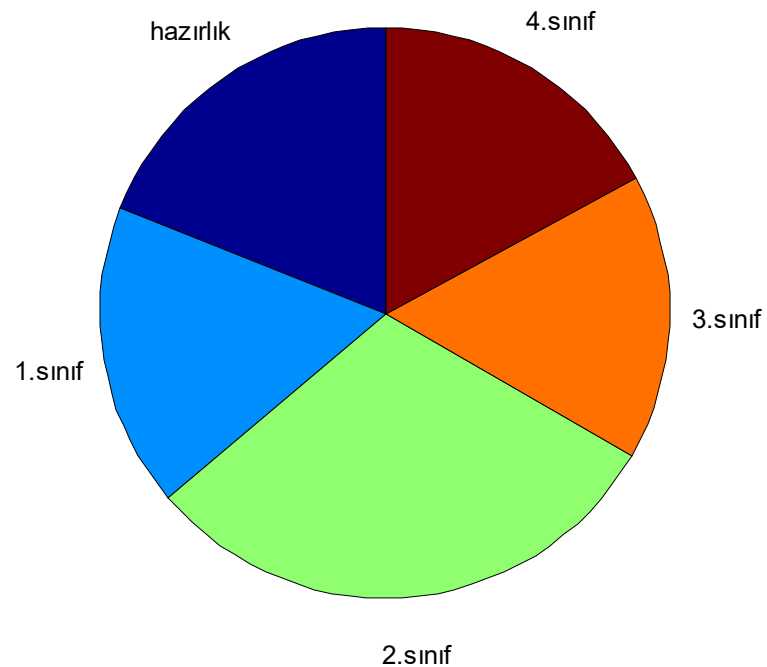
```
a=[5 4 8 9];  
pie(a)
```



MATLAB/Grafik-pie fonksiyonu

- Sınıflara göre öğrenci sayıları verildiğine göre, bu değerlerin yüzdelik dağılımlarını bir grafik üzerinde gösterelim.

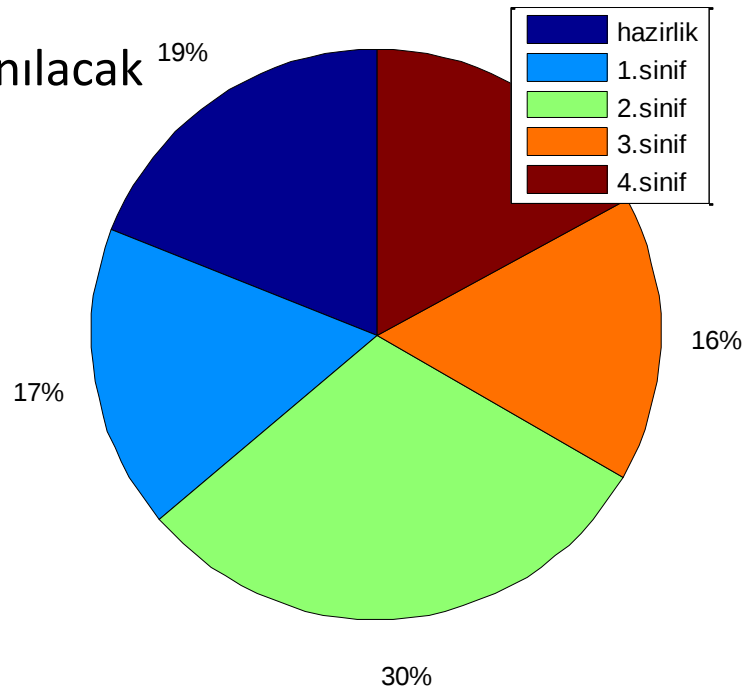
```
clear  
clc  
a=[250, 225, 400, 212, 225];  
b={'hazirlık','1.sinif', '2.sinif','3.sinif','4.sinif'};  
pie(a,b)
```



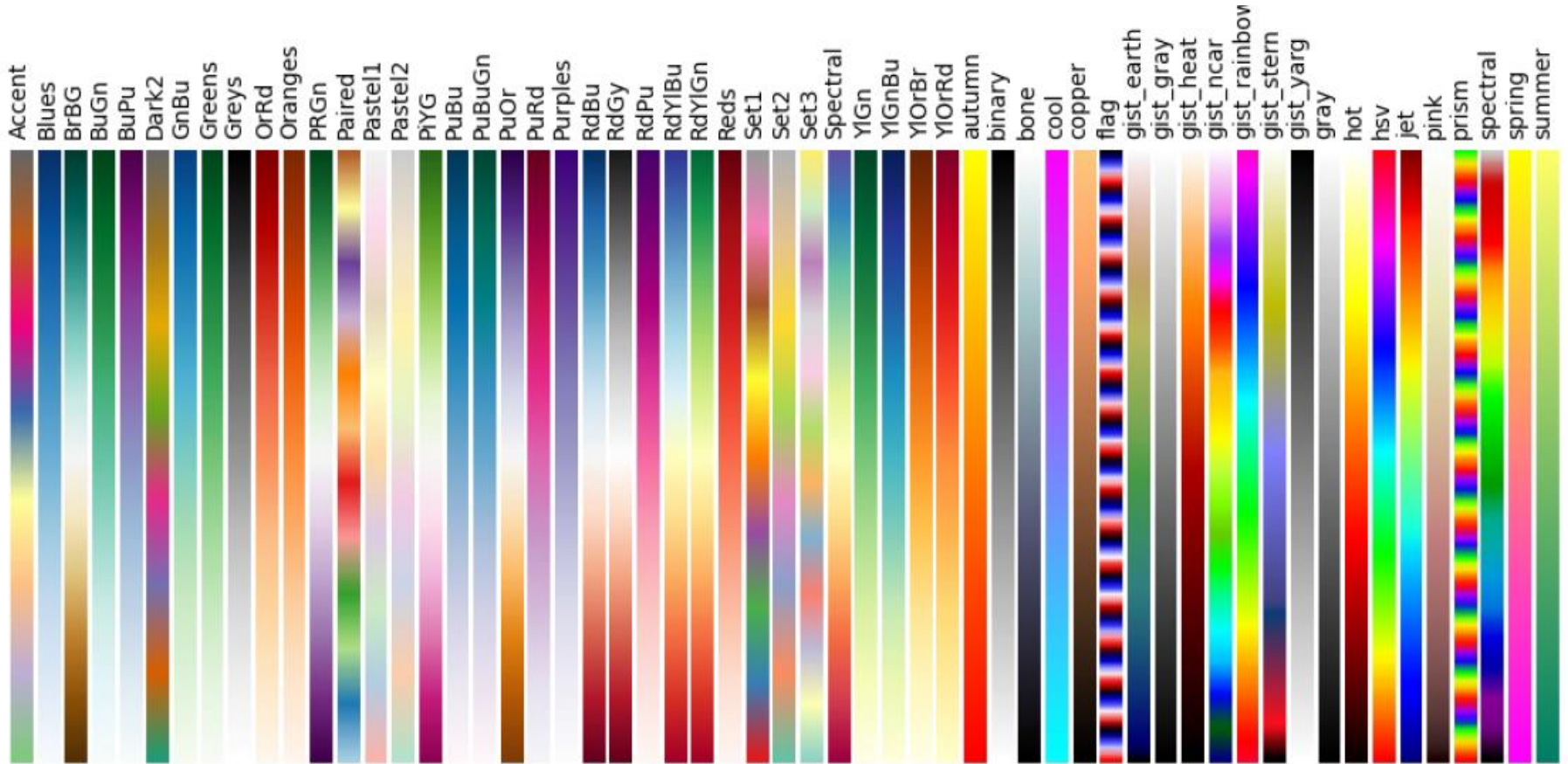
MATLAB/Grafik-pie fonksiyonu

```
clear  
clc  
a=[250, 225, 400, 212, 225];  
pie(a)  
legend('hazirlik','1.sinif', '2.sinif','3.sinif','4.sinif');
```

colormap fonksiyonu grafikte kullanılacak renk haritasını seçmek için kullanılır.

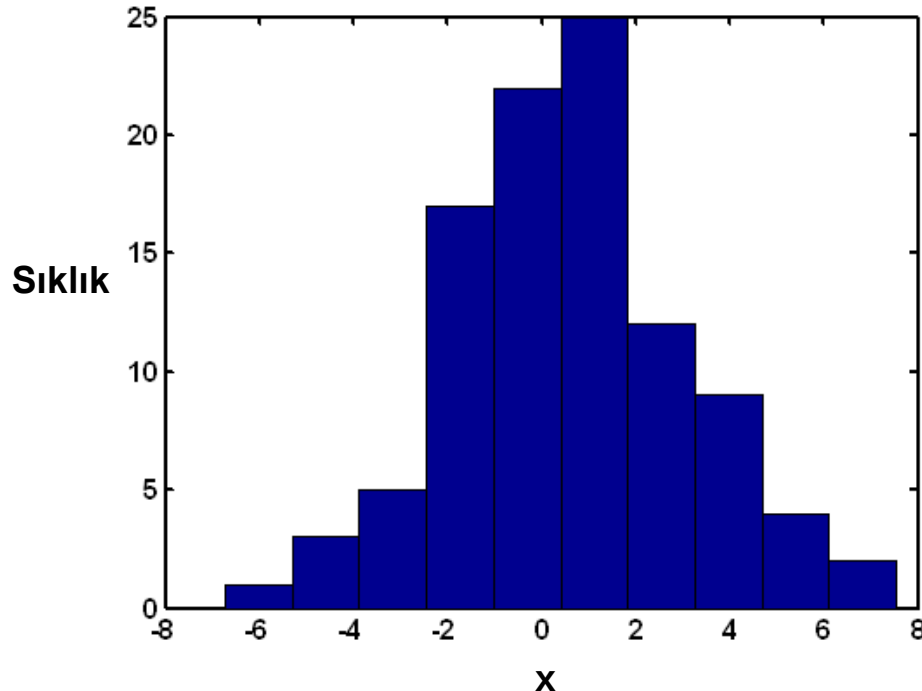


Matlab da kullanan renk haritalari asagida verilmistir.



MATLAB/Grafik-Histogram

- Ölçülerin hangi istatistiksel dağılıma uyduğunu görebilmek için, frekans (sıklık) değerleri hesaplanır ve histogram grafikleri çizilir.
- Elimizde, aynı dağılımda olduğu bilinen bir x ölçü vektörü varsa, `hist(x)` fonksiyonu otomatik olarak bir histogram grafiği çizer.
- Örneğin, `x=randn(100,1)*3` biçiminde normal dağılmış bir ölçü grubu üretelim. `hist(x)` ile aşağıdaki histogram grafiği oluşturulur (Her bir barın üst noktası birleştirildiğinde oluşan eğrinin bir normal dağılım eğrisi veya diğer adıyla çan eğrisi biçiminde olduğu görülecektir.)

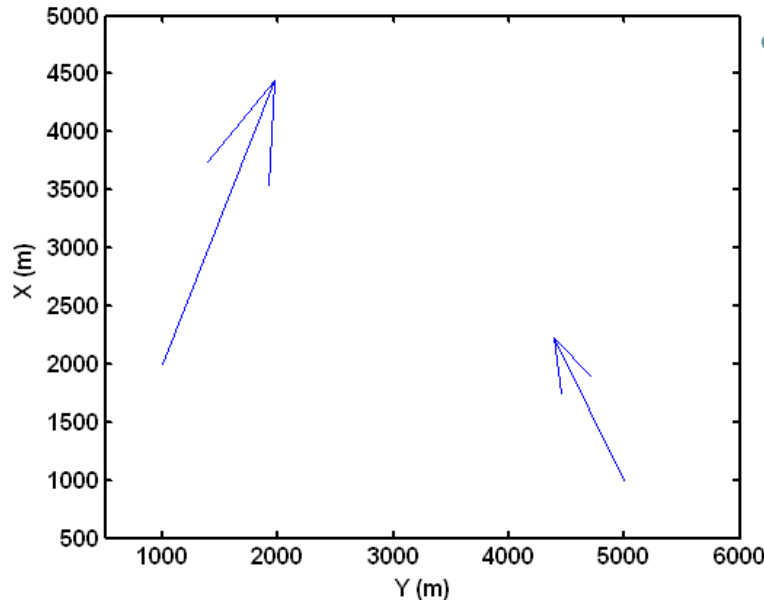


MATLAB/Grafik-Vektör çizimi

- x ve y koordinat değerlerine sahip bir noktanın dx ve dy kadar yer değiştirdiği düşünölsün. Bu noktadaki (dx,dy) vektörünü çizdirmek istediğimizde, **quiver** fonksiyonu kullanılır.
- Örneğin, bir jeodezik dik koordinat sisteminde iki noktanın koordinatları $\mathbf{x}=[1000;2000]$, $\mathbf{y}=[5000;1000]$ vektörleri, bu noktadaki değişimler ise $d\mathbf{x}=[1;2]$ ve $d\mathbf{y}=[-0.5;0.8]$ ile tanımlansın.

quiver(y,x,dy,dx) (**Not:** Bir jeodezik dik koordinat sisteminde x ve y 'nin yer değiştirdiğini hatırlayınız!)

komutu ile bir jeodezik dik koordinat sisteminde vektör çizimi gerçekleştirilir.



- Vektörleri ölçeklendirmek için, s ölçek faktörü quiver fonksiyonuna beşinci bir değişken olarak eklenmelidir;

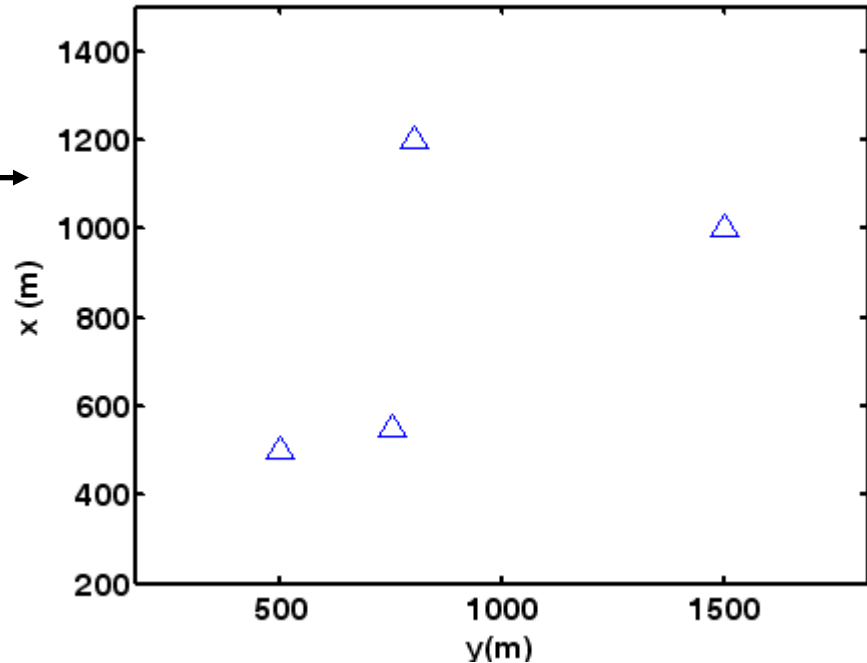
quiver(y,x,dy,dx,s)

MATLAB/Grafik-Kanava Çizimi

- Bir jeodezik dik koordinat sistemindeki x ve y koordinatları verilen jeodezik noktaları, nokta sembolleri üçgen olacak biçimde çizdiriniz.

Nokta	P1	P2	P3	P4
x (m)	500.00	550.00	1000.00	1200.00
y (m)	500.00	750.00	1500.00	800.00

```
plot(y,x,'^ r')  
axis([200 1700 200 1500])  
axis equal
```

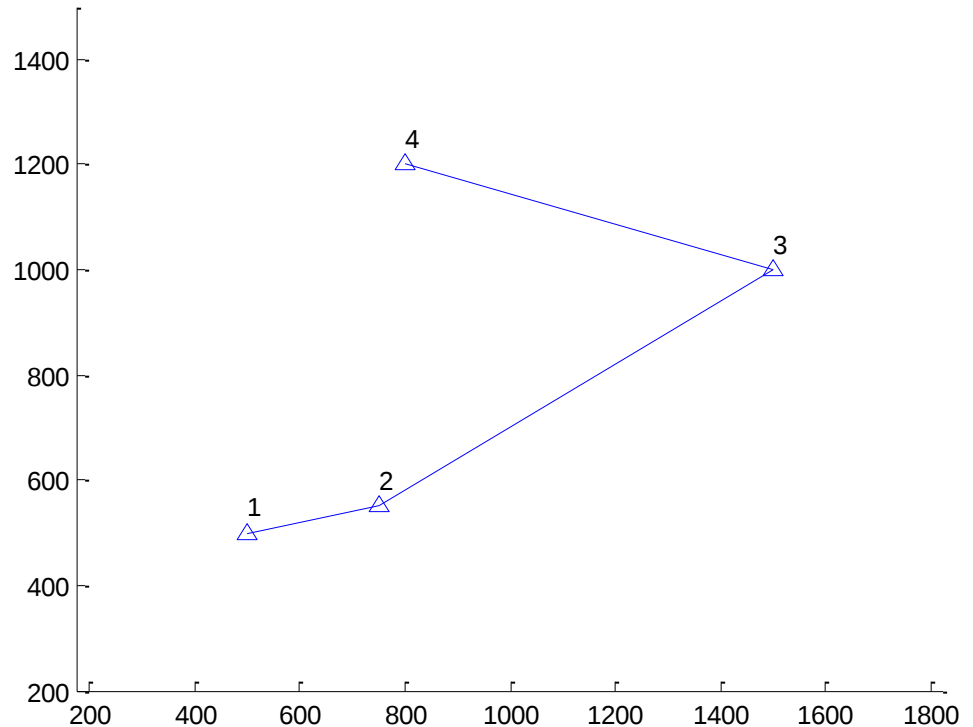


- `axis([Xmin Xmax Ymin Ymax])` fonksiyonu, eksenlerin en küçük ve en büyük değerlerini ayarlar,
- `axis equal` ise x ve y eksenlerindeki ölçek faktörünü (büyüme ve küçülme oranlarını) eşitler.

MATLAB/Grafik-Line Fonksiyonu

- Koordinatları belirli iki noktayı doğrusal olarak birleştirmek için `line([x1 x2],[y1 y2])` şeklinde kullanılır.

```
clear
clc
ad=[1 2 3 4];
x=[500 550 1000 1200];
y=[500 750 1500 800];
hold on
plot(y,x,'^')
axis([200 1700 200 1500])
axis equal
for i=2:length(x)
    line([y(i-1) y(i)],[x(i-1) x(i)]);
end
for i=1:length(ad)
    text(y(i),(x(i)+50),num2str(ad(i)))
end
hold off
```

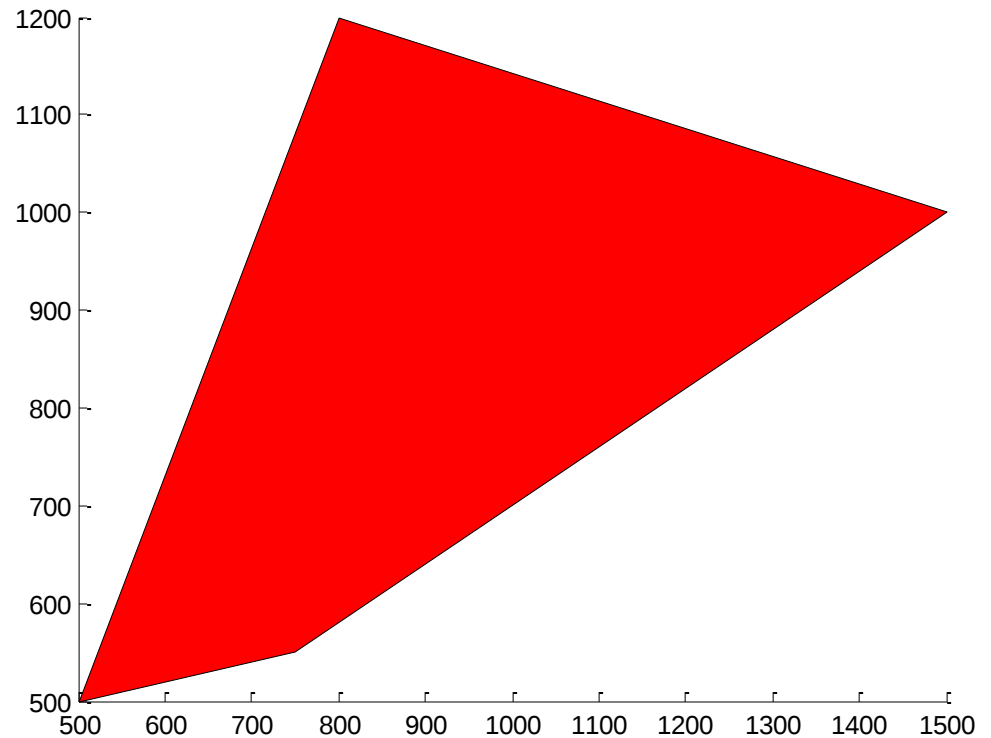


`text(X,Y,'string')`

MATLAB/Grafik-Patch Fonksiyonu

- Koordinatları belirli alanı doldurmaya yarar. Kullanımı `patch(X,Y,C)` şeklindedir. Burada C taranacak rengi belirler.

```
clear  
clc  
x=[500 550 1000 1200];  
y=[500 750 1500 800];  
patch(y,x,'r')
```



Subplot fonksiyonu

Bir pencerede birden çok grafik göstermek için subplot komutu kullanılır.

Subplot(M,N,P);

M: Grafik penceresindeki satır sayısı,

N: Grafik penceresindeki sütun sayısı,

P: Grafik penceresinde hangi grafik olduğunu ifade eder.

`xlabel('x eksenine yazılmak istenen başlık')`

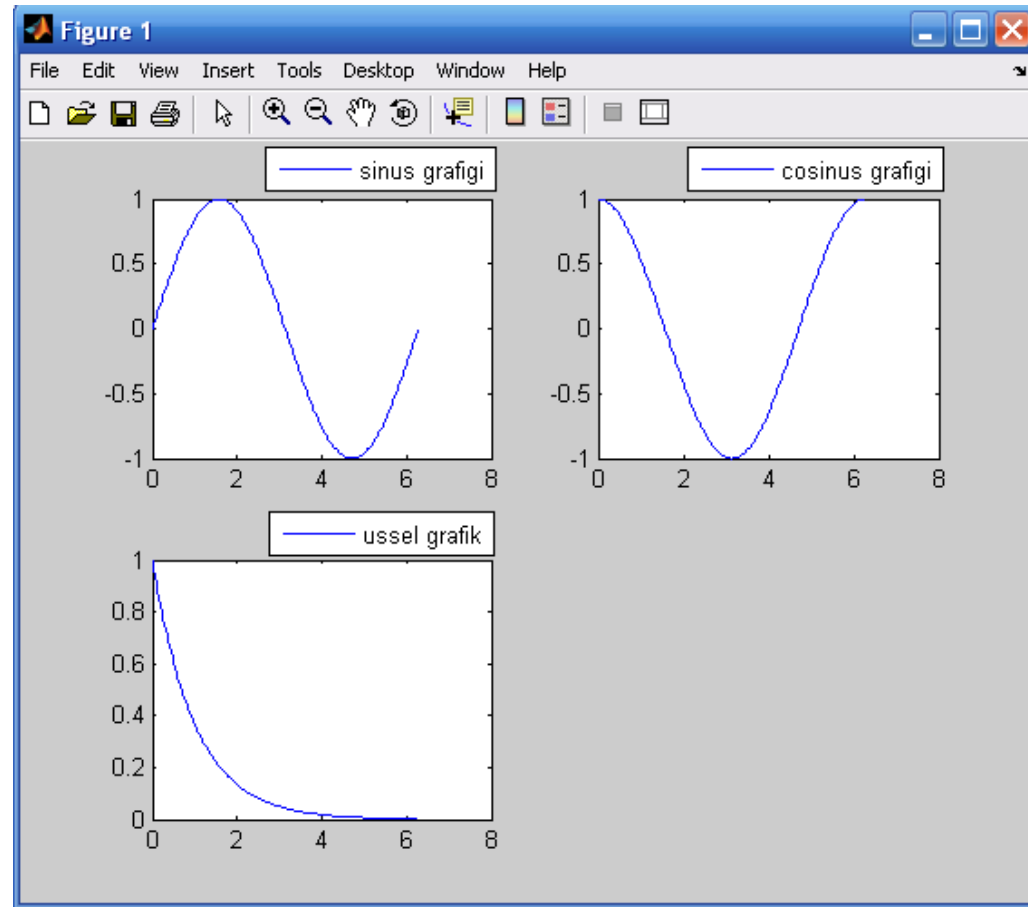
`ylabel('y eksenine yazılmak istenen başlık')`

`legend('oluşan çizimlere açıklama getirmek için kullanılır.')`

`title('grafiğin başlığı')`

Ornek

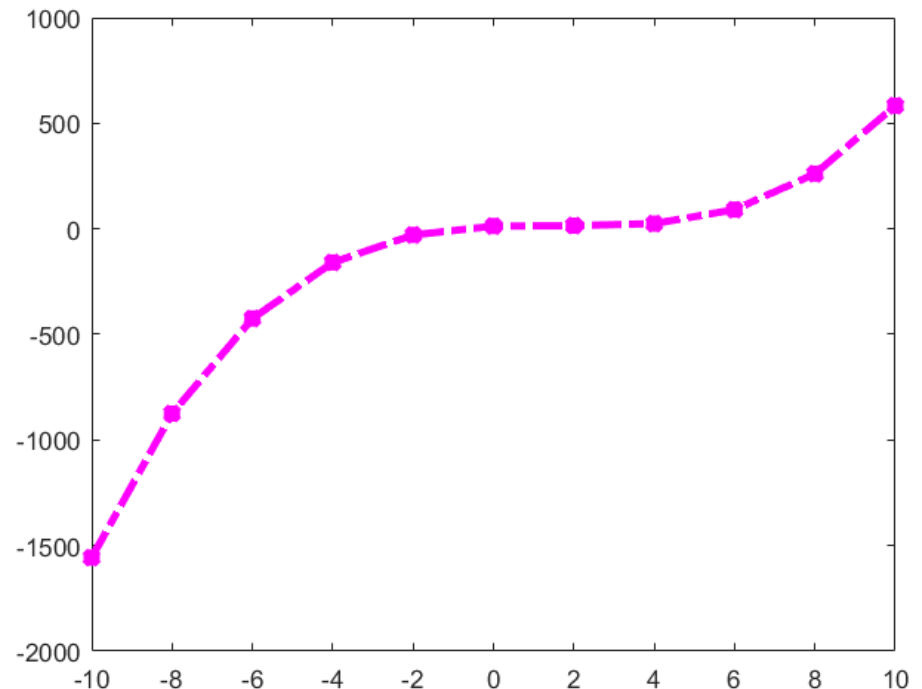
```
x=0:pi/100:2*pi;  
y1=sin(x);  
y2=cos(x);  
y3=exp(-x);  
subplot(2,2,1);  
plot(x,y1);legend('sinus grafigi');  
subplot(2,2,2);  
plot(x,y2);legend('cosinus grafigi');  
subplot(2,2,3);  
plot(x,y3);legend('ussel grafik');
```



Matlab'da çizilen bir grafiğin çizgi kalınlığını degistirmek için **LineWidth** özelliği ve bu özelliğe bağlı nümerik deger kullanılır.

Örnek 58 $x = -10, x = 10$ aralığında $y = x^3 - 5x^2 + 7x + 13$ fonksiyonunun grafigini 3 birim kalınlık, eflatun renk, yıldız kesim noktası isareti ve kesintili noktasal çizgi ile çiziniz.

```
x = -10:2:10;  
y = x.^3 - 5*x.^2 + 7*x + 13;  
plot(x,y, 'm-.*', 'LineWidth', 3)
```

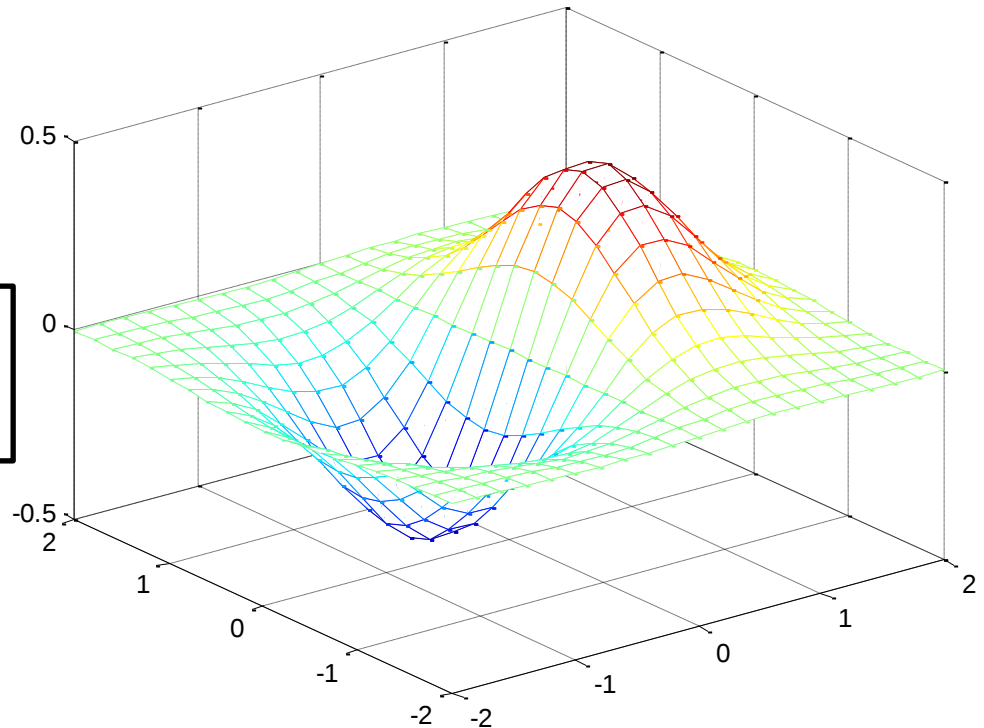


MATLAB/3B Grafik- meshgrid ve mesh fonksiyonları

- **meshgrid** fonksiyonu x ve y vektörleri ile belirtilen alanı, X ve Y matrislerine dönüştürür. **[X,Y]=meshgrid(x,y)** şeklinde kullanılır.
- **mesh** fonksiyonu ise $Z=f(X,Y)$ ile belirli iki değişkenli fonksiyonun belirttiği yüzeyi renkli ağ örgüsü şeklinde çizer. **mesh(X,Y,Z)** şeklinde kullanılır.

$-2 < x < 2$, $-2 < y < 2$, aralığı için

```
[X,Y] = meshgrid(-2:0.2:2, -2:0.2:2);  
Z = X .* exp(-X.^2 - Y.^2);  
mesh(X,Y,Z)
```

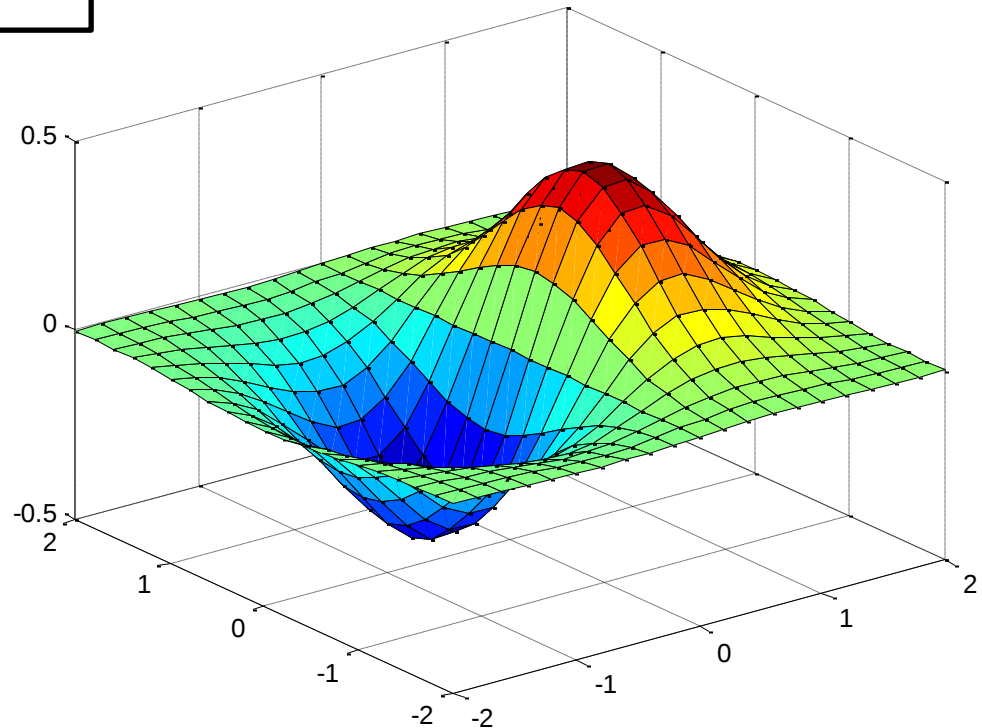


MATLAB/3B Grafik- surf fonksiyonu

- **surf** fonksiyonu ise $Z=f(X,Y)$ ile belirli iki değişkenli fonksiyonun belirttiği yüzeyi çizer. **surf(X,Y,Z)** veya **surf(Z)** şeklinde kullanılır.

$-2 < x < 2$, $-2 < y < 2$, aralığı için

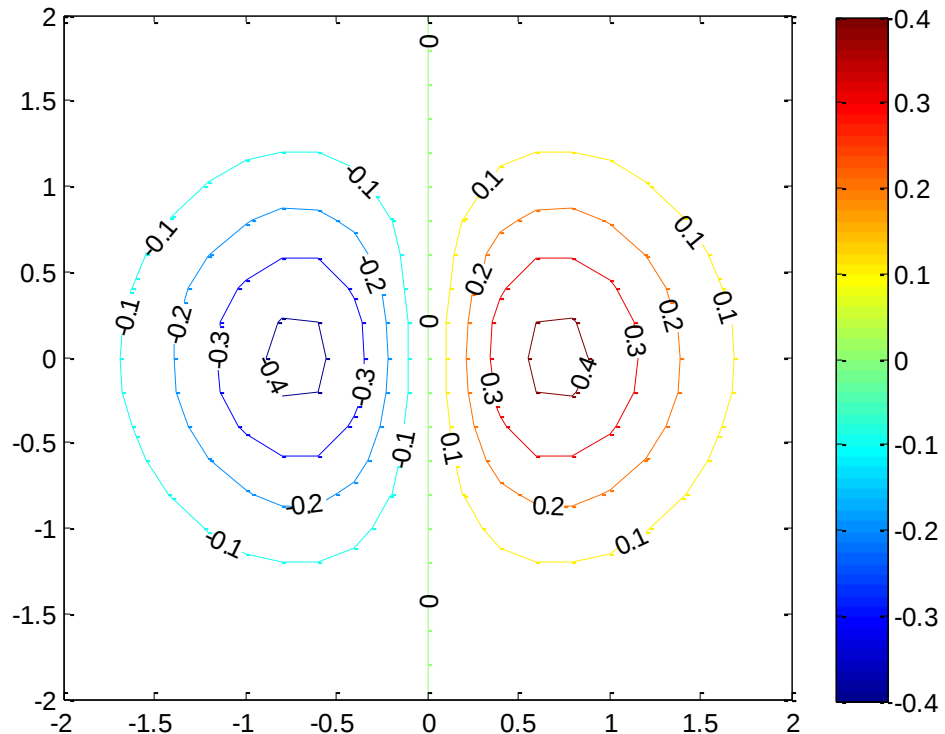
```
[X,Y] = meshgrid(-2:0.2:2, -2:0.2:2);  
Z = X .* exp(-X.^2 - Y.^2);  
surf(X,Y,Z)
```



MATLAB/3B Grafik- contour fonksiyonu

- Yüzeylere ait eşyüksehti eğrilerinin x-y düzleminde çizdirilebilmesi için `contour` fonksiyonundan yararlanılır. `[C,H]=contour(x,y,z)` şeklinde kullanıldığı durumda eş yüksehti değerlerini [C,H] vektörüne atayacaktır.
- Ayrıca, `clabel(C,H)` komutu ile eşyüksehti değerleri grafik üzerinde gösterilebilir.

```
[X,Y] = meshgrid(-2:0.2:2, -2:0.2:2);  
Z = X .* exp(-X.^2 - Y.^2);  
[C,H]=contour(X,Y,Z);  
clabel(C,H)  
colorbar
```

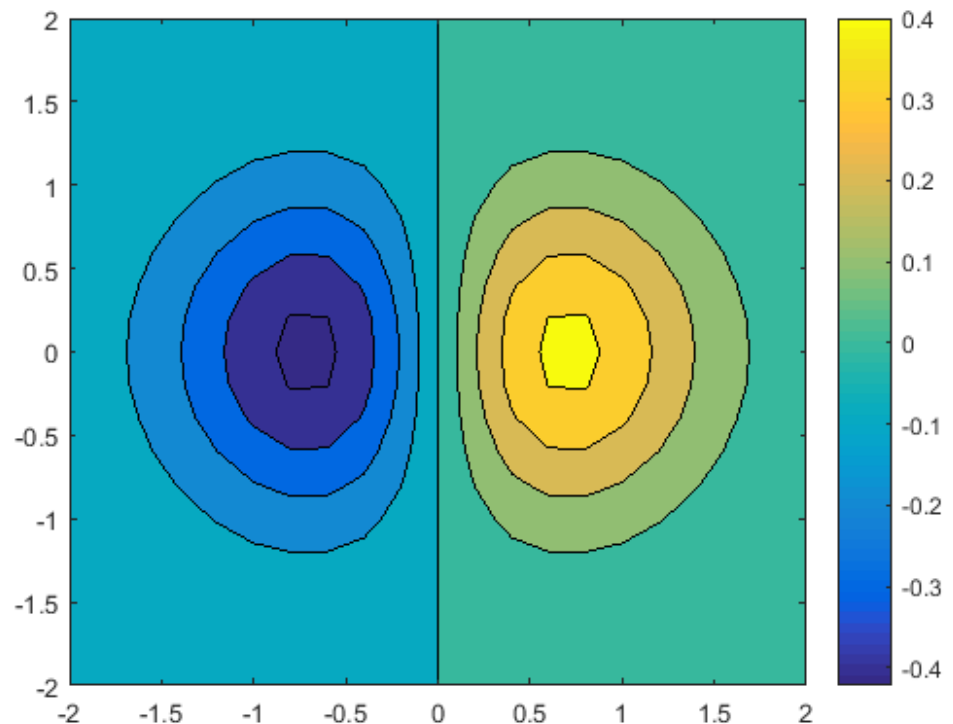


MATLAB/3B Grafik- `contourf` fonksiyonu

- Yüzeylere ait eşyüksehti eğrilerini dolgulu biçimde x-y düzleminde çizdirilebilmesi için `contourf` fonksiyonundan yararlanılır.

Renk ölçeğini yanına eklemek için `colorbar` fonksiyonu kullanılabilir.

```
[X,Y] = meshgrid(-2:0.2:2, -2:0.2:2);  
Z = X .* exp(-X.^2 - Y.^2);  
contourf(X,Y,Z);  
colorbar
```



MATLAB/Dosya Yazdırma-Okuma

- Dosya yazdırma, `fopen`, `fprintf` ve `fclose` fonksiyonlarının kullanımı ile gerçekleştirilir.
- Bu fonksiyonlar ile dosya yazdırmada, yazdırılacak olan metnin “command window” da gösterilmesine gerek yoktur.
- `fopen`, program çıktılarının yazdırılacağı dosyayı açar, `fprintf` yazdırır ve `fclose` ise yazdırma işlemini sonlandırır.

```
ifade=fopen('dosya_adi', 'izin')  
fprintf(ifade,'aciklama',değişken)  
fclose(ifade)
```

Burada:

ifade-> dosya değişkeni

dosya adi->verilerin kaydedileceği dosyanın adı

izin-> dosyanın kullanım izni : **r,w,..., vs.**

MATLAB/Dosya Yazdırma-Okuma

Dosya yazdırma ve okumada kullanılan izinler

Tip	Açıklama
'r'	Dosyayı sadece okumaya açar. (Yazma işlemine izin vermez)
'r+'	Dosyayı yazmaya ve okumaya açar.
'w'	Var olan bir veri dosyası içindeki bilgileri siler, dosya yoksa oluşturur ve dosyayı yazmaya açar.
'w+'	Var olan bir veri dosyası içindeki bilgileri siler, dosya yoksa oluşturur ve dosyayı okumaya ve yazmaya açar.
'a'	Var olan bir veri dosyasını yazmak için açar, dosya yoksa oluşturur ve girilecek bilgileri dosya sonuna ekler.

MATLAB/Dosya Yazdırma-Okuma

- Örneğin, bir a kenarı programda hesaplatılmış olsun. Bu programın a çıktısını, kenar.txt isimli bir dosyaya yazdırmak için, aşağıdaki kodlar düşünülür;

```
a=150.0234234;  
fid=fopen('kenar.txt','w');  
fprintf(fid,'kenar uzunluğu=%1.4f',a);  
fclose(fid);
```

w bu dosyanın üzerine yazılacağını gösterir.

MATLAB/Dosya Yazdırma-Okuma

- **Örnek:** a=[3.12356 4.12456 1;5.8463 6.45111 2;4 5 6] biçiminde verilen bir a matrisini, elemanları virgülden sonra 4 hane olacak biçimde, mat.out dosyasına yazdıran bir program yazınız.

```
a=[3.12356 4.12456 1;5.8463 6.45111 2;4 5 6]
fid = fopen('mat.out','w');
fprintf(fid,'%1.4f%10.4f%10.4f\n',a);
fclose(fid);
```

- **Örnek:** kenar=1500.123 m ve aciklik=103.3367 grad olan değişkenleri, sonuc.out dosyasına alt alta yazdırınız.

```
kenar=1500.123;
aciklik=103.3367;
fid=fopen('sonuc.out','w');
fprintf(fid,'kenar=%1.3f m\n',kenar);
fprintf(fid,'aciklik=%1.4f grad',aciklik);
fclose(fid)
```

MATLAB/Dosya Yazdırma-Okuma

- Bir önceki uygulamada mat.out dosyasına yazdırdığımız matrisi tekrar Matlab ortamında okutalım.

```
fid=fopen('mat.out','r+');  
[dizi,sayi]=fscanf(fid,'%f',inf)
```

dizi =

3.1236
5.8463
4.0000
4.1246
6.4511
5.0000
1.0000
2.0000
6.0000

sayi =

9

```
fid=fopen('mat.out','r+');  
[dizi,sayi]=fscanf(fid,'%f',[3 3])
```

dizi =

3.1236	4.1246	1.0000
5.8463	6.4511	2.0000
4.0000	5.0000	6.0000

sayi =

9

MATLAB/Dosya Yazdırma-Okuma

- Matlab'de dosyaların içindeki kolon yapısındaki metinlerin okunması için `textread` fonksiyonu bulunmaktadır. Örneğin, aşağıda koordinat.txt dosyasındaki verilerin okunması istensin:

P1	1000.1234	1300.23423
P2	1300.5673	1450.98563
P3	2000.1500	2000.11000
P4	3500.3100	1000.12000

Nokta isimleri x koordinatları y koordinatları

- Bunun için,

```
[nokta,x,y]=textread('koordinat.txt','%s%f%f')
```

fonksiyonu kullanılır. `nokta`, nokta isimlerini içeren bir hücre dizisi; `x`, x koordinat vektörü ve `y`, y koordinat vektörü olarak atanır.

```
[a, b, c,...]=textread('dosya_adi','format')
```

MATLAB/Dosya Yazdırma-Okuma

- **Örnek:** Koordinat dosyası, aşağıdaki gibi olan bir koordinat.txt dosyasından, nokta isimlerini, x ve y koordinatlarını textread fonksiyonu kullanarak uygun değişkenlere atayınız.

Nirengi koordinatları		
NN	x (m)	y (m)
P1	1000.1234	1300.23423
P2	1300.5673	1450.98563
P3	2000.1500	2000.11000
P4	3500.3100	1000.12000

[nokta,x,y]=textread('koordinat.txt','%s%f%f','headerlines',2)

'headerlines' komutu ve ardından gelen sayı, dosyanın başlangıcından itibaren kaç tane satırın dikkate alınmayacağını gösterir.

koordinat.txt dosyasında ilk iki satır alınmadan nokta isimleri, x ve y koordinatları okunmuştur.

MATLAB/Dosya Yazdırma-Okuma

- Excel' den veri okutmak amacıyla `xlsread` fonksiyonu kullanılır.

```
num = xlsread('filename', sheet, 'range')
```

```
A = xlsread('deneme.xlsx')
```

```
A = xlsread('deneme.xlsx', 1, 'C4:D8')
```

	A	B	C	D	E	F	G	H	I
1									
2									
3									
4			1	6					
5			2	7					
6			3	8					
7			4	9					
8			5	10					
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									
22									
23									
24									
25									
26									
27									
28									
29									
30									
31									
32									
33									
34									
35									
36									
37									
38									

A =

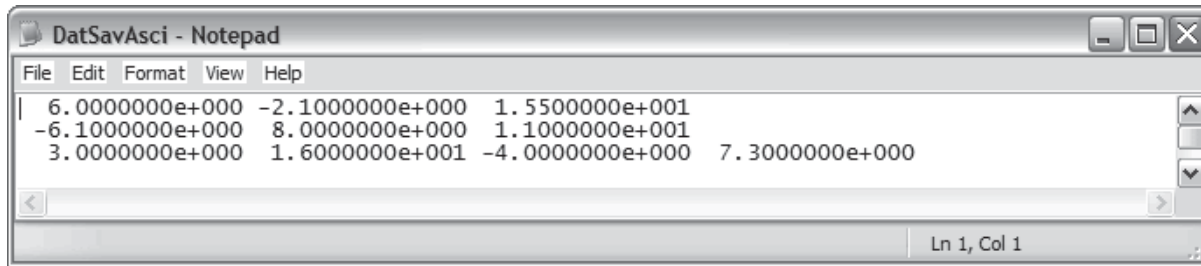
1	6
2	7
3	8
4	9

MATLAB'dan bir Excel elektronik tablosuna veri aktarımı, `xlswrite` komutu kullanılarak yapılır.

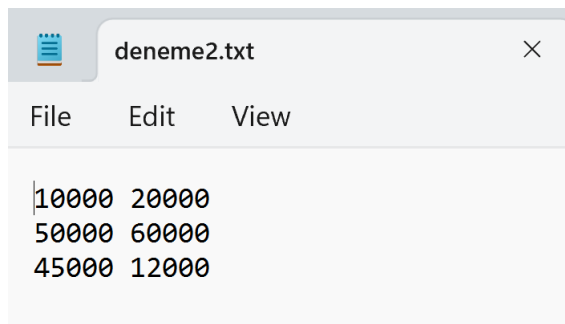
```
xlswrite ("dosya adı", değişken_adı)
```

Load komutu

Load komutu, kaydedilen verileri çalışma alanına aktarmak için de kullanılabilir. Ancak bu, yalnızca dosyadaki veriler MATLAB'da bir değişken biçiminde olduğunda mümkündür. Böylece, dosya bir sayıya (skaler), bir satıra veya bir sayı sütununa (vektör) veya her birinde aynı sayıda sayıya sahip satırlara (matris) sahip olabilir.



```
load file_name veya VarName = load ("dosya_adı")
```



```
load deneme.txt
```

Fonksiyonlar

- Karmaşık ve uzun programları, küçük, basit ve belirli bir amacı olan program parçalarına bölebiliriz. Belirli bir işi yapan bu program parçalarına *fonksiyon* adı verilir.
- Fonksiyon kullanımı, kaynak programın daha modüler (parçalı) bir şekilde inşa edilmesini sağlar.

- **Fonksiyon kullanımının faydaları:**

Kodun gereksiz yere büyümesini engeller: Sıkça tekrarlanan işlemler için bir kere fonksiyon yazıldığında aynı kodlar tekrar yazılmaksızın istenildiği kadar çalıştırılabilirler.

Okunabilirliği artırarak algılamayı kolaylaştırır. Main fonksiyonu içinde birbirinden ayrılmış sadece kendi işlerini yapan fonksiyonların isimleri bulunur. Detay işlemler fonksiyonların içinde halledilir.

Programın test edilmesi ve hata bulmayı kolaylaştırır. Hata araştırılırken aranılması gereken alt program bloğuna bakılır.

Güncelleştirilebilir olmasını ve yeniden kullanabilme kolaylığı sağlar.

Fonksiyonlar, bir m-dosyası biçiminde saklanır ve bu dosyalara, fonksiyon dosyaları adı verilir.

M-Fonksiyon Yapısı

function cikis_ifadesi=**fonksiyon_adi** (giris_ifadesi1, 2, ...n)

M-Fonksiyonlar kullanılırken dikkat edilecek hususlar:

- 1- Kullanıcılar kendi fonksiyonlarını yazmak için m-fonksiyonlarını kullanabilirler.
- 2- **Function** alt programı ve ana program şeklinde iki program yazılarak bu iki program ayrı ayrı kaydedilir.
- 3- Alt programdaki **fonksiyon_adi**, m-dosyasına verilen isimle aynı olmalıdır.
- 4- Ana programdan alt program, function adı kullanılarak çağrılır.
- 5- Parametre aktarımı olması durumunda alt ve ana programda eşit sayıda parametre ve giriş değişkeni olmalıdır.

function cikis_ifadesi =fonksiyon_adi (giris_ifadesi1, 2, ...n)

Örnek: İki nokta arasındaki uzaklığı bulan programı m-fonksiyon (alt program) kullanarak yazınız.

x1=1.noktanın x koordinati; x2=2.noktanın x koordinati

y1=1.noktanın y koordinati; y2=2.noktanın y koordinati

FUNCTION ALT PROGRAMI (uzak.m):

function uzaklik = uzak(x1,y1,x2,y2)

uzaklik=sqrt((x2-x1).^2+(y2-y1).^2);

Bu function alt programı uzak.m olarak kaydedilir.

end

ANA PROGRAM:

ax=3; ay=4; bx=1; by=2;

sonuc = uzak(ax,ay,bx,by); % uzak.m alt programını çağırıyor

fprintf('iki nokta arasindaki uzaklik=%f',sonuc);

Adım adım gerçekleştirilen işlemler:

- Ana program herhangi bir isimle kaydedilir ve koşturulur.
- Program, **function** adına (**uzak**) geldiği zaman alt program çağrılır ve ax, ay, bx, by parametreleri sırasıyla x1, y1, x2, y2 giriş değişkenlerine aktarılır.
- **Function** alt programında hesaplama gerçekleştirilir.
- **Function**'daki çıkış değişkeni olan **uzaklik** alt programda kullanılmakta ana programda aynı değer **sonuc** değişkeninde saklanır.
- Alt programdan ana programa parametre aktarımı zorunlu değildir. İstenirse değişkenlerin değerleri alt programda da girilebilir ve sonuç alt programda yazdırılabilir.

Uygulama: Yukarıdaki örneği ana programdan alt programa parametre aktarımı yapmadan yeniden yazınız.
(Değişkenlerin girilmesi, sonucu hesaplama ve yazdırma işlemi alt programda yapılacaktır)

FUNCTION ALT PROGRAMI:

```
function uzaklik = uzak  
x1=3; y1=4; x2=1; y2=2;  
uzaklik=sqrt((x2-x1).^2+(y2-y1).^2);  
fprintf('iki nokta arasindaki uzaklik=%f', uzaklik);  
end
```

ANA PROGRAM:

uzak; % uzak.m alt programını çağırıyor

Fonksiyonlar

- Birden fazla çıktısı olan fonksiyonlar için
- `function [çıktı 1, çıktı2,...]=fonk-adı(girdi 1, girdi 2,...)`
- Biçiminde tanımlanır.

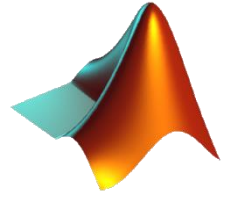
- Eğer tek bir çıktı alınacak ise,
- `function çıktı=fonk-adı(girdi 1, girdi 2,...)`
- biçiminde tanımlanabilir.

Verilen bir değere kadar olan sayıların toplamını bulan Matlab fonksiyonunu yazınız.

```
%toplamfonk.m  
function y=toplamfonk(a)  
y=0;  
for i=1:a,  
y=y+i;  
End
```

Ana program
toplamfonk(4)

Örnek



% rms.m dosyası, x vektörünün karekök ortalama değerini
% ve mutlak ortalama değerini hesaplar.

```
function [r,m] = rms(x)
r = sqrt(sum(abs(x).^2)/length(x));
m = sum(abs(x))/length(x);
```

```
>> x=-4:4;
>> [r,m]=rms(x)
```

```
r =

    2.5820
```

```
m =

    2.2222
```

```
>> |
```

Ornek

Fonksiyonun girdisi ve ciktisi olmayabilir.

```
function [] = stars()  
asteriks = char( abs('*')*ones(1,10));  
disp( asteriks )  
end
```

```
function stars  
asteriks = char( abs('*')*ones(1,10));  
disp( asteriks )  
end
```

```
>> stars()  
*****  
  
>> stars  
*****
```

Ornek

Fonksiyonun girdi ve ciktisi dizi(array) olabilir.

```
function output = g(x,y)
% This function multiply x and y
% x and y must be the same size
output = x.*y;
end
```

```
>> a = [1 3 5];
>> b = [2 3 2];
>> c = g(a,b)
c = 2 9 10
```

Ornek

Lokal fonksiyonlar

```
function [add_result, minus_result] = compute(x,y)
% This function add and subtract the elements
% stored in x and y
add_result = add(x,y);
minus_result = minuss(x,y);

function result = add(x,y)
result = x+y;

function result = minuss(x,y)
result = x-y;
```

```
>> [plus,minus] = compute(2,3)
plus = 5
minus = -1
```

Ornek

Bir silindirin alan ve hacmini hesaplayan matlab fonksiyonunu yaziniz

```
function [alan, hacim]= alanvehacim (r,h)
```

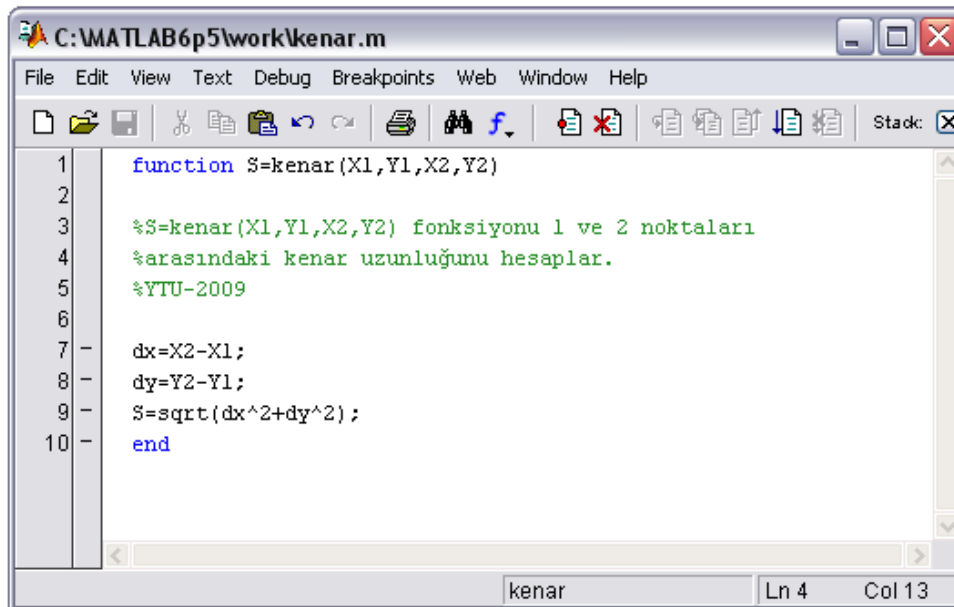
```
hacim=pi*r^2*h
```

```
alan=pi*r^2+2*pi*h
```

```
>> [aa,bb]=alanvehacim (2,3)
```

MATLAB/Fonksiyon Dosyası Oluşturma

- **Örnek:** Koordinatları bilinen iki nokta arasındaki yatay uzunluğu hesaplayan kenar isimli bir fonksiyon oluşturunuz.



The image shows a MATLAB editor window titled 'C:\MATLAB6p5\work\kenar.m'. The window contains the following code:

```
1 function S=kenar(X1,Y1,X2,Y2)
2
3 %S=kenar(X1,Y1,X2,Y2) fonksiyonu 1 ve 2 noktaları
4 %arasındaki kenar uzunluğunu hesaplar.
5 %YTU-2009
6
7 dx=X2-X1;
8 dy=Y2-Y1;
9 S=sqrt(dx^2+dy^2);
10 end
```

The status bar at the bottom indicates the cursor is at line 4, column 13.

- Fonksiyonların, biçim olarak, diğer programlardan tek farkı,

`function output=fonk_ismi(input)`

ile başlaması ve fonksiyon dosyasının sonunda `end` ile bitmesidir.
- function komutunun bulunduğu ilk satırdan hemen sonra gelen açıklama (comment) satırları, ilgili fonksiyonun “yardım” metinleridir.
- Fonksiyon ismiyle, fonksiyon dosyasının ismi aynı olmalıdır.

MATLAB/Fonksiyon Dosyası Oluşturma

- **Örnek:** Hem açıklık açısını hem de kenar uzunluğunu üreten **aci_kenar** isimli bir fonksiyon oluşturunuz.

```
function [ a,S ] = aci_kenar( X1,Y1,X2,Y2)
%[a,S]=aci_kenar(X1,Y1,X2,Y2) fonksiyonu (1-2) açıklık açısını
%(a) ve 1-2 kenar uzunlugunu hesaplar

DX=X2-X1;DY=Y2-Y1;

if (DX~=0)&(DY~=0),a=atan(DY/DX);a=a*200/pi;
    if (DX>0)&(DY>0),a=a;end
    if (DX<0)&(DY>0),a=a+200;end
    if (DX<0)&(DY<0),a=a+200;end
    if (DX>0)&(DY<0),a=a+400;end
end

if (DX==0)&(DY>0),a=100;end
if (DX==0)&(DY<0),a=300;end
if (DX>=0)&(DY==0),a=0;end
if (DX<0)&(DY==0),a=200;end

S=sqrt(DX^2+DY^2);%kenar

end
```

- Bir fonksiyonun birden fazla çıktısı olabilir. Bu örnekte a ve S gibi iki çıktı bulunmaktadır.
- a, açıklık açısını, S ise kenar uzunluğunu göstermektedir.
- aci_kenar(X1,Y1,X2,Y2) komutuyla, ilk output, yani açıklık açısını belirten a değişkeni üretilir.

MATLAB/Uygulama-15

Aşağıda bir kenara ait 15 ölçü kenar.txt dosyasında verilmektedir. Bu ölçülerin ortalamasını, her bir ölçünün ortalamadan farkını (düzeltmeleri), ölçülerin standart sapmasını hesaplayan, $|düzeltme| > 3 * \text{standart sapma}$ olan ölçüleri ölçü kümesinden silen ve geriye kalan ölçüleri temizolcu.txt dosyasına yazdıran matlab kodunu yazınız.

Olculer

15.538
16.834
12.741
15.862
15.319
13.692
14.566
15.343
18.578
17.769
13.650
18.035
25.725
14.937
15.715

Duzeltme=ortalama – ölçü

Standart sapma= $([duzeltme^2]/(\text{ölçü sayisi}-1))^{(1/2)}$

MATLAB/Uygulama-15

```
clear
clc
a=textread('kenar.txt','%f','headerlines',3); %textread fonksiyonu ile veriler okutuluyor.
orta=mean(a); %mean fonksiyonu ile ölçülerin ortalamasi bulunuyor.

for i=1:length(a) % for döngüsüyle duzeltmelerin hesaplanmasi yapiliyor.
    duzeltme(i,1)=orta-a(i);
end
stan=sqrt(duzeltme*duzeltme/(length(a)-1)); %ölçülerin standart sapmasi hesaplaniyor.

hata=0;
artim=0;
for i=1:length(duzeltme) %duzeltmesi standart sapmasindan büyük ölçüler bulunuyor.
    if abs(duzeltme(i))>3*stan
        artim=artim+1;
        hata(artim)=i;
    end
end

hata=sort(hata)
for i=1:artim % hatali olcu siliniyor.
    a(hata(artim+1-i),:)=[];
end

veri=fopen('temizolcu.txt','w+') % temiz olculer yazdiriliyor.
fprintf(veri,'%1.3f\n',a)
fclose(veri)
```

MATLAB/Uygulama-16

Küre üzerinde açı değeri olarak verilen yay değerini uzunluğa çeviren matlab kodunu fonksiyon dosyası şeklinde yazınız.

```
function [ S] = yaykenari(aci,R,ro )
%[ S] = yaykenari(aci,R,ro ) fonksiyonu acisal olarak
%girilen yay deđerinin uzunlugunu
hesaplamaktadır.
% aci ifadesi hesap yapılacak aci degerini, R
degeri kürenin yaricapini,
% ro degeri 180/pi veya 200/pi degerini ifade eder.

S=aci/ro*R;

end
```

Kaynaklar

Doç. Dr. Cüneyt Aydın ders notları, Yıldız Teknik Üniversitesi, 2017