

2 D (2 Boyutlu) Grafikler

x-y eksen takımının yer aldığı düzlemde çizilen grafiklerdir. Bu grafikleri 3 ana bölüm altında inceleyebiliriz.

- Doğru ve veri grafikleri
- Fonksiyon grafikleri
- Özel (çubuk, pasta, halka v.b.) grafikler

Doğru ve Veri Grafikleri

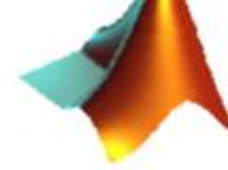
plot fonksiyonu kullanılarak çizilir. *plot* fonksiyonu 2 boyutlu doğru grafikleri çizdirir.

x ve y aynı boyutlu iki vektör ise x'deki sayılar x_ekseni yani absis, y'deki sayılarda y_ekseni yani ordinat üzerinde olsunlar. *plot* fonksiyonu x'in her bir noktasına karşılık gelen y değerlerini çizdirir. Yani *plot* fonksiyonu aynı uzunlukta iki vektörün çizimini sağlar.

x	y	
1	25	x(1), y(1)
2	0	x(2), y(2)
3	20	x(3), y(3)
4	5	x(4), y(4)
5	15	x(5), y(5)

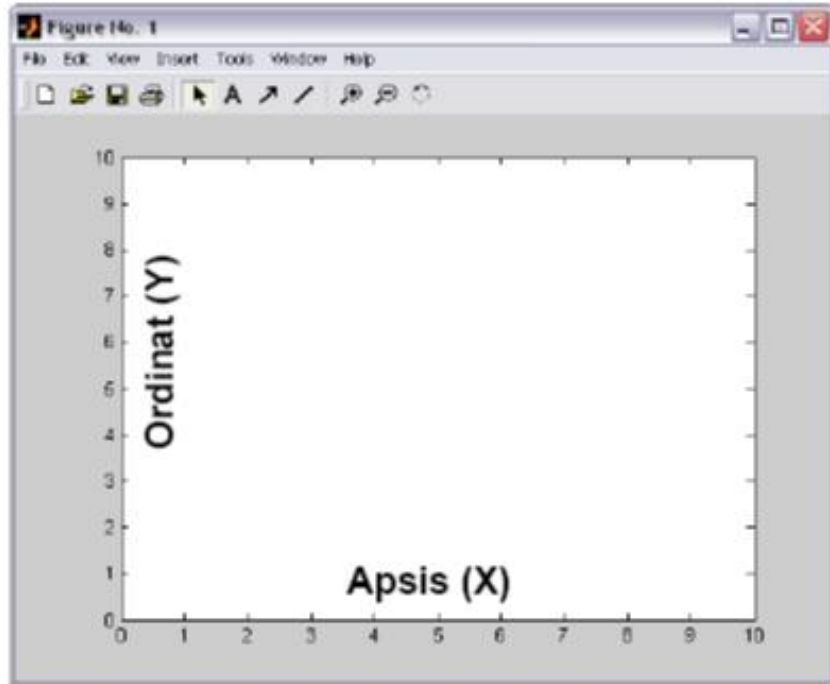
Matlab'de en temel çizim fonksiyonu **plot**'dur.

help graph2d komutu kullanılarak iki boyutlu çizime dair detaylara ulaşılabilir.

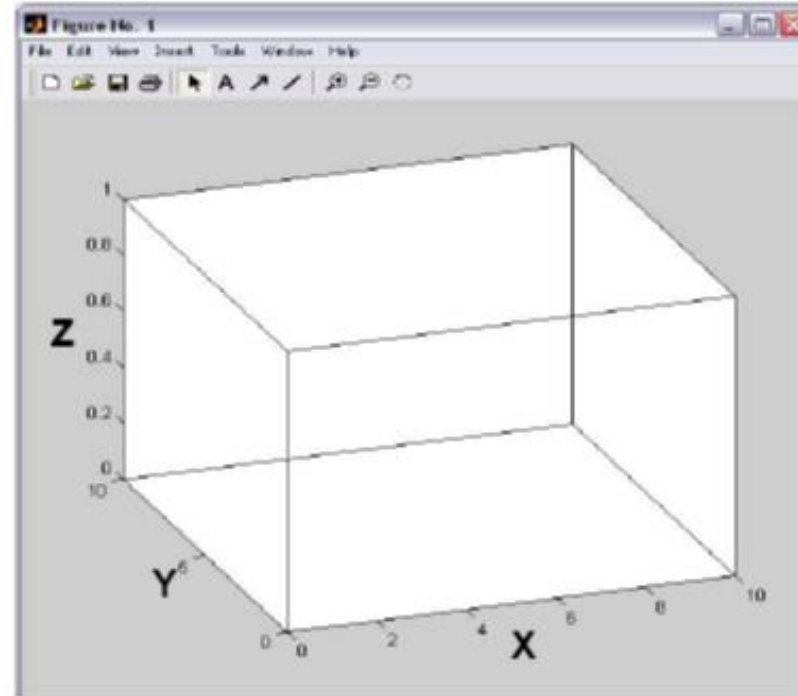


- Matlab'de grafikler “figure” penceresinde çizdirilir.
- İki ve üç boyutlu çizim yanı sıra, kutupsal koordinat sisteminde de çizim olanağı bulunur (bak., polar).

İki Boyutlu Koordinat Sistemi



Üç Boyutlu Koordinat Sistemi

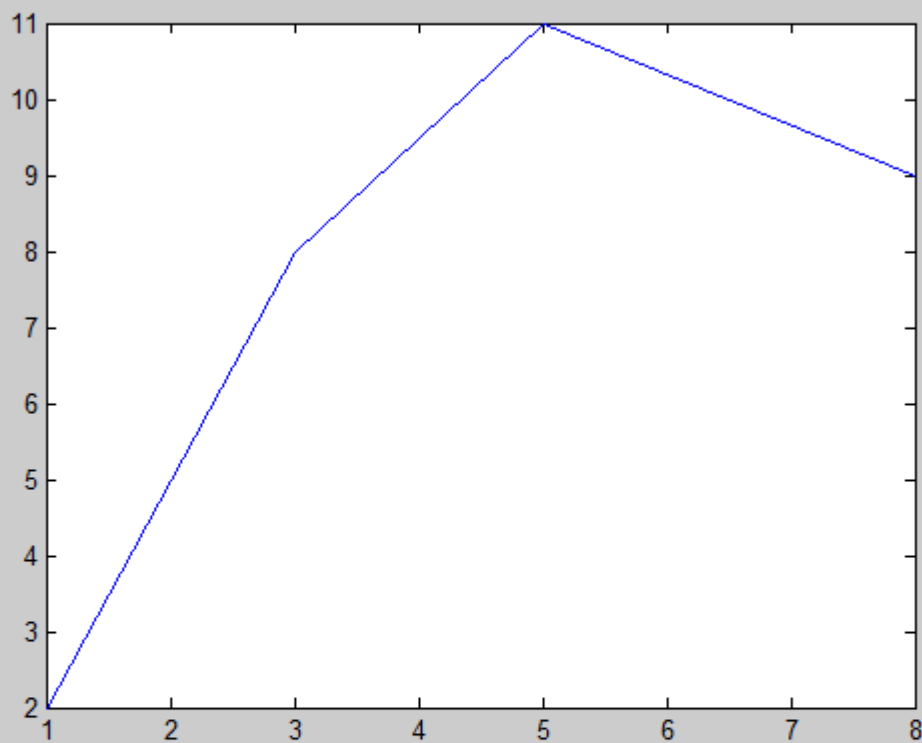


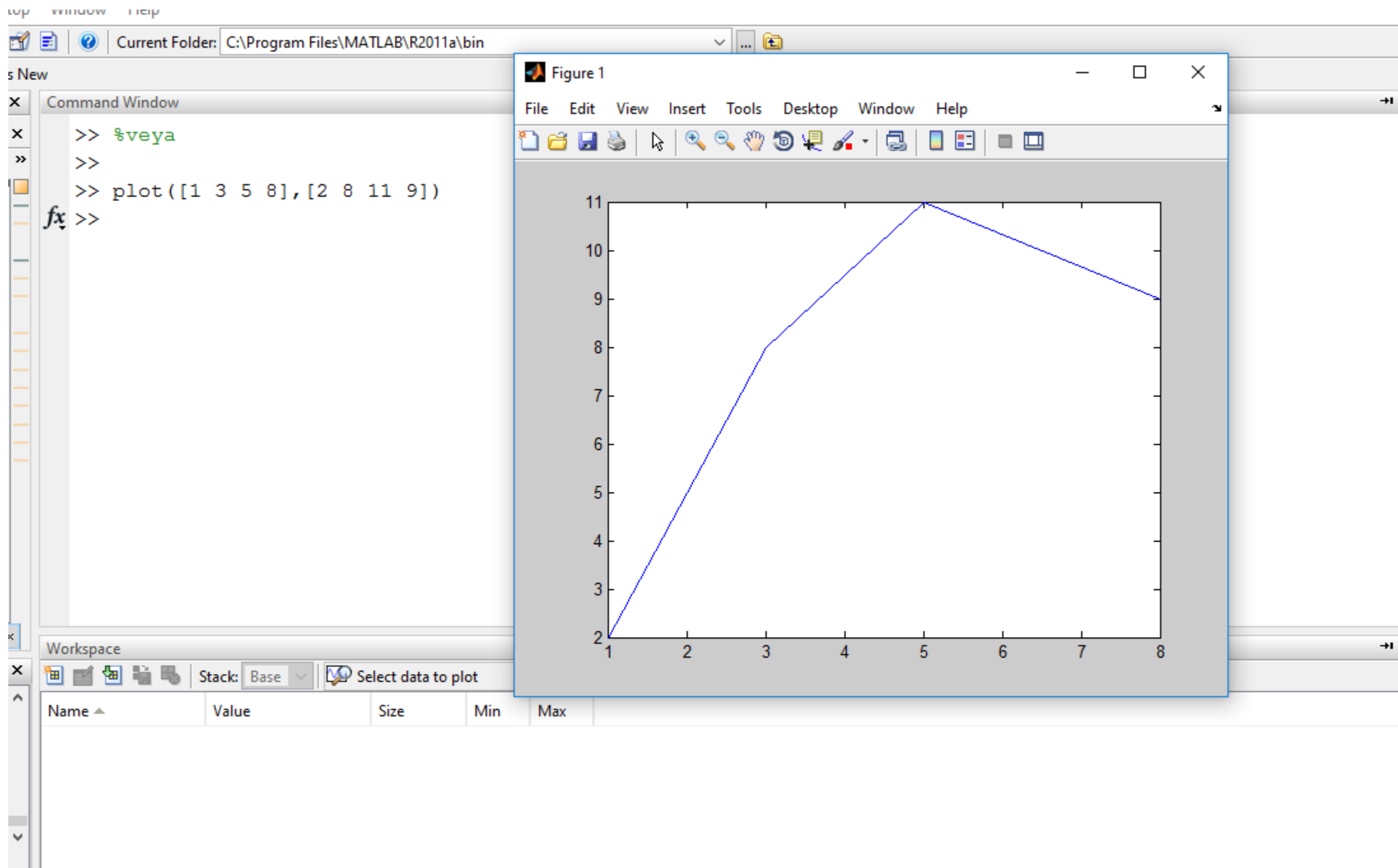
Command Window

Workspace

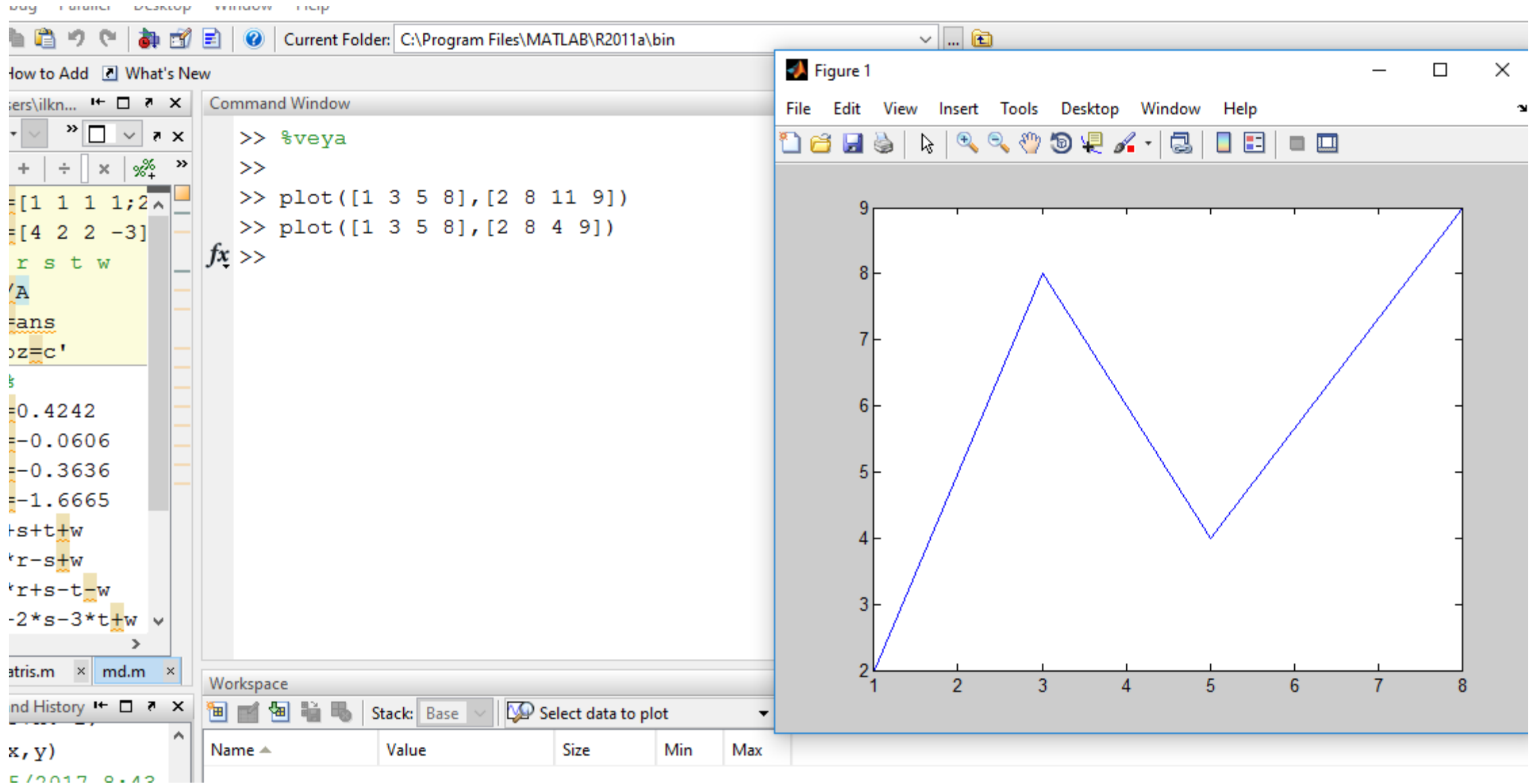
Name ▲	Value	Size	Min	Max
x	[1,3,5,8]	1x4	1	8
y	[2,8,11,9]	1x4	2	11

Name ▲	Value	Size	Min	Max
x	[1,3,5,8]	1x4	1	8
y	[2,8,11,9]	1x4	2	11





- Grafiği çizdikten sonra Figure penceresi açıkken komut penceresinde (command window) değişiklik yapılırsa var olan grafik üzerine değişiklikler doğrudan yansır.



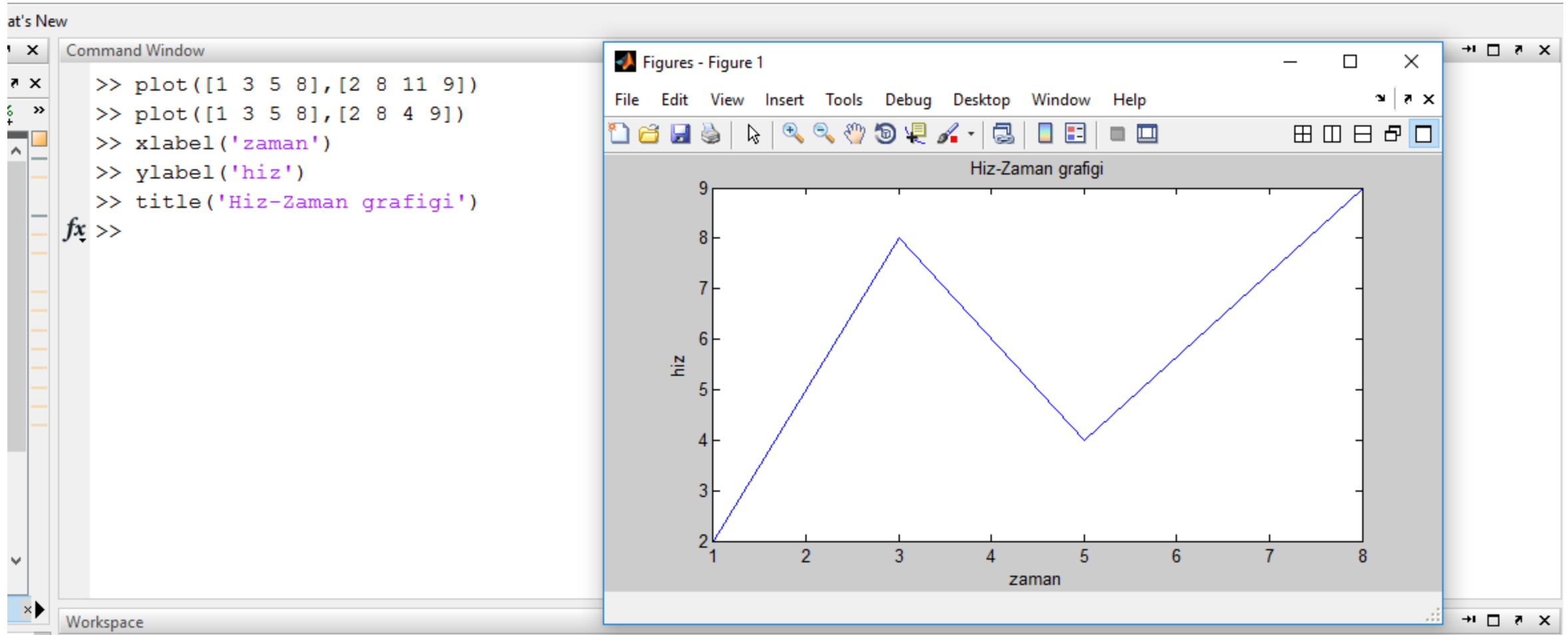
Eksen adlandırma (xlabel, ylabel) ve grafiğe başlık (title) ekleme komutlarının kendi aralarında önceliği yoktur.

❖ **Ancak plot komutundan sonra kullanılmaları gerekir.**

xlabel ('isim') : x_eksenini adlandırır.

ylabel ('isim') : y_eksenini adlandırır.

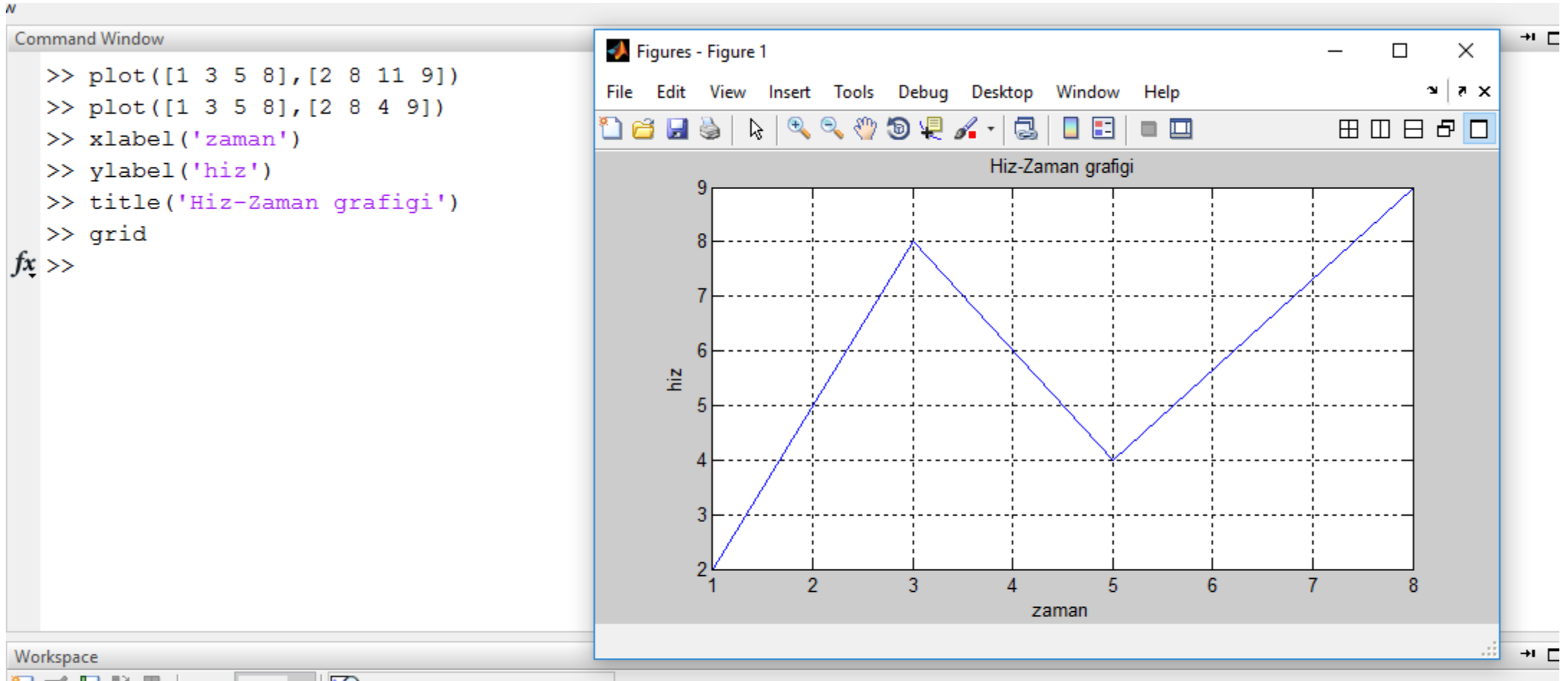
title ('isim') : grafiğe başlık adı verir.



Grafiğin daha anlaşılır olması için, diğer bir deyişle grafiğin okunabilirliğini arttırmak ve görünümünü daha belirgin yapmak amacıyla kılavuz çizgileri eklenir.

grid : geçerli tüm eksenlere kılavuz çizgileri ekler.

grid off : kılavuz çizgilerini kaldırır.



Window Help



Current Folder: C:\Program Files\MATLAB\R2011a\bin



ew

Command Window

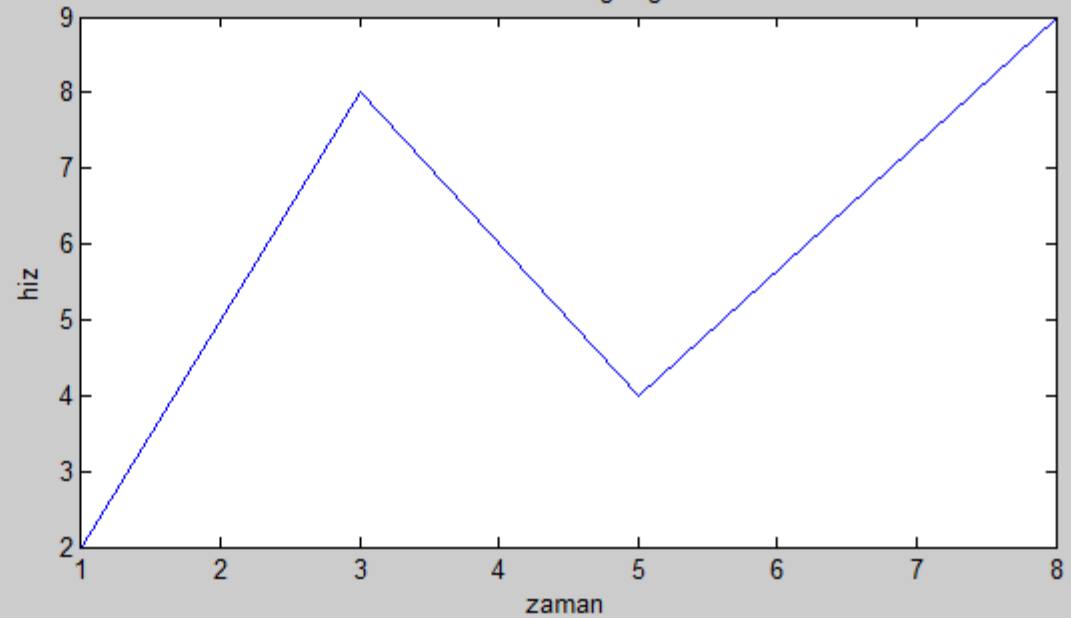
```
>> plot([1 3 5 8],[2 8 11 9])  
>> plot([1 3 5 8],[2 8 4 9])  
>> xlabel('zaman')  
>> ylabel('hiz')  
>> title('Hiz-Zaman grafigi')  
>> grid  
>> grid off  
fx >>
```

Figures - Figure 1

File Edit View Insert Tools Debug Desktop Window Help



Hiz-Zaman grafigi



Grafiklere Sembol ve Yunan Karakterleri Ekleme

Mühendislik ve bilimsel grafiklerde Eksen adlandırma (xlabel, ylabel) ve grafiğe başlık (title) ekleme ya da ileride görülecek olan grafiğe gösterge (legend) ve yazı (text-gtext) ekleme sırasında α , β , γ , θ , ε , ω gibi yunan alfabesi harflerini ve diğer sembolleri eklemek gerekebilir. Bunlardan en çok kullanılanların Matlab kod sistemindeki karşılıkları aşağıdaki tabloda verilmiştir.

Sembol	Matlab'deki Karşılığı
α	\alpha
β	\beta
γ	\gamma
ε	\epsilon
θ	\theta
λ	\lambda
μ	\mu
π	\pi
ϕ	\phi
ω	\omega
üst indis	^
Alt indis	_

Bu semboller sadece grafik düzenleme işlemlerinde değil program kodlarının yazımında da aynı şekilde kullanılabilir.

Command Window

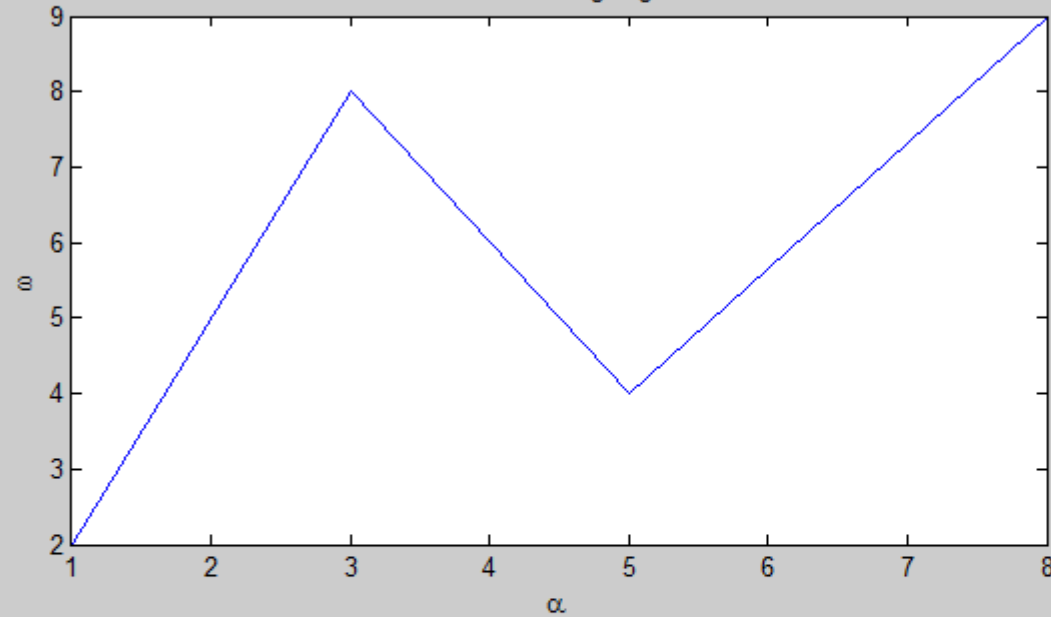
```
>> plot([1 3 5 8],[2 8 11 9])  
>> plot([1 3 5 8],[2 8 4 9])  
>> xlabel('zaman')  
>> ylabel('hiz')  
>> title('Hiz-Zaman grafigi')  
>> grid  
>> grid off  
>> xlabel('\alpha')  
>>  
>> ylabel('\omega')  
fx >>
```

Figures - Figure 1

File Edit View Insert Tools Debug Desktop Window Help



Hiz-Zaman grafigi



Workspace

Stack: Base Select data to plot

Name	Value	Size	Min	Max
------	-------	------	-----	-----

OVR

ENG 9:09 AM

Command Window

```
>> plot([1 3 5 8],[2 8 11 9])
>> plot([1 3 5 8],[2 8 4 9])
>> xlabel('zaman')
>> ylabel('hiz')
>> title('Hiz-Zaman grafigi')
>> grid
>> grid off
>> xlabel('\alpha')
>>
>> ylabel('\omega')
>> xlabel('\alpha_0')
>>
```

Command Window

```
>> plot([1 3 5 8],[2 8 11 9])
>> plot([1 3 5 8],[2 8 4 9])
>> xlabel('zaman')
>> ylabel('hiz')
>> title('Hiz-Zaman grafigi')
>> grid
>> grid off
>> xlabel('\alpha')
>>
>> ylabel('\omega')
>> xlabel('\alpha_0')
>> xlabel('\alpha_0^1')
>>
```

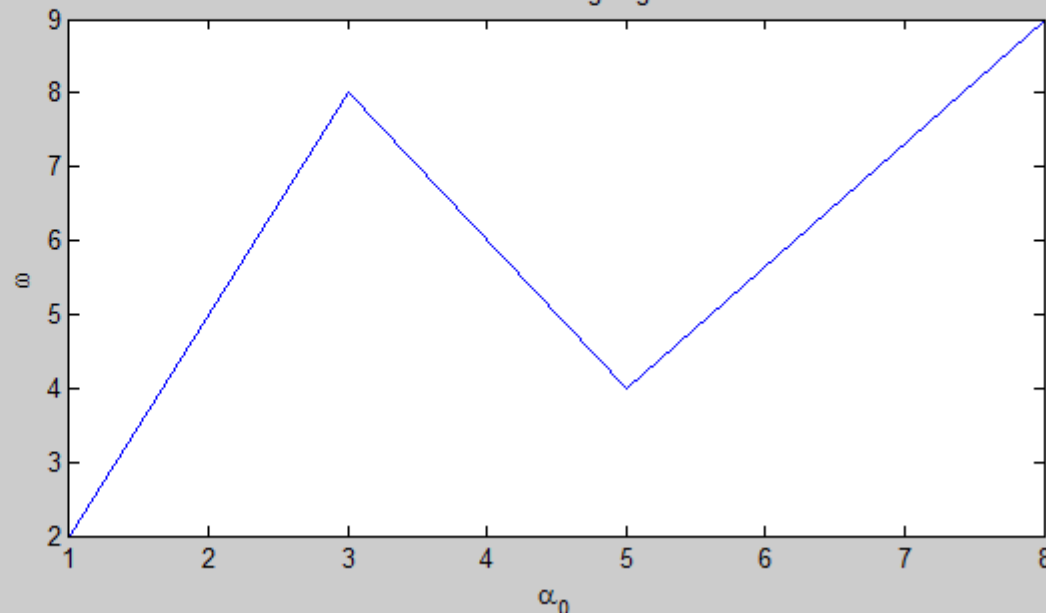


Figures - Figure 1

File Edit View Insert Tools Debug Desktop Window Help



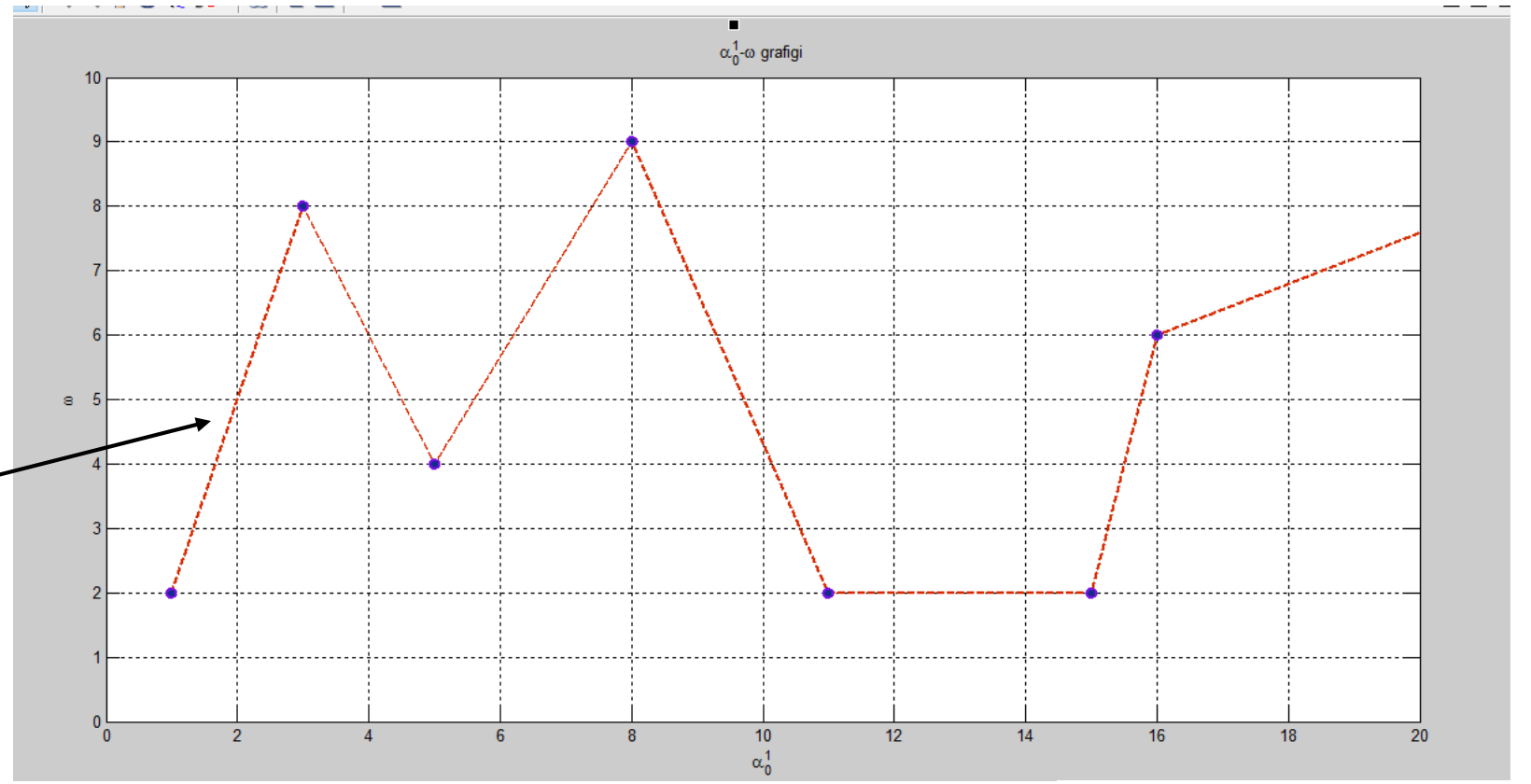
Hiz-Zaman grafigi



Cizgi ve Isaretleme Secenekleri

Genel yazim sekli
`plot(x,y,'indikator')`

Grafik cizgisinin rengi kirmizi,
cizgi turu kesikli
isaretleyici daire seklinde



İşaretleme Sembolü	İndikatör
Nokta	.
Artı	+
Yıldız	*
Daire	o
x	x
Kare	s

1.Çizgi Rengi

Renk	İndikatör
Mavi	b
Yeşil	g
Kırmızı	r
Siyah	k

Default

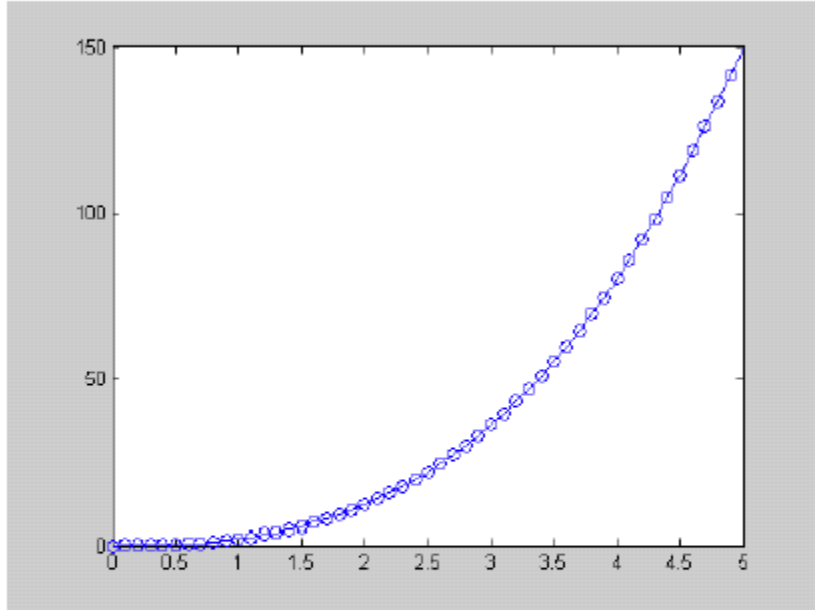
Çizgi Türü	İndikatör
Düz Çizgi	-
Kesikli Çizgi	--
Noktalı Çizgi	:
Kesikli Noktalı	-.

Çoklu grafik çizimlerinde örneğin üç grafik varsa mavi, yeşil, kırmızı olarak otomatik renklendirilir.

`plot(x,y, 'k+:')` siyah renkte + işaretli noktalı çizgi.
`plot(x,y, 'r+')` sadece kırmızı renkte + işaretleri görünür çizgi gözükmez.

Karakter rengi	Karakter Sembolu	Karakter çizgi stili
bblue. ggreen rred ccyan mmagenta yyellow kblack	.point ocircle xx-mark +plus *star square ddiamond vtriangle (down) ^triangle (up) <triangle (left) >triangle (right) ppentagram hhexagram	-solid :dotted -.dashdot --dashed Bu tablodan yararlanarak cizdiginiz grafikleri istediginiz sekilde duzenleyebilirsiniz lutfen deneyiniz <code>plot(x,y,'k+:')</code> plots a black dotted line with a plus at each data point. <code>plot(x,y,'bd')</code> plots blue diamond at each data point but does not draw any line. <code>plot(x,y,'y-',x,y,'go')</code> plots the data twice, with a solid yellow line interpolating green circles at the data points.

- `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)

- **Degerleri 0 ile 5 arasinda 0.1 degiserek artan vektor elemanlarına karşılık $y=x^3+x^2$ fonksiyonunun degerlerini hesaplatarak grafigini ciziniz**

Desktop Window Help

Current Folder:

What's New

Command Window

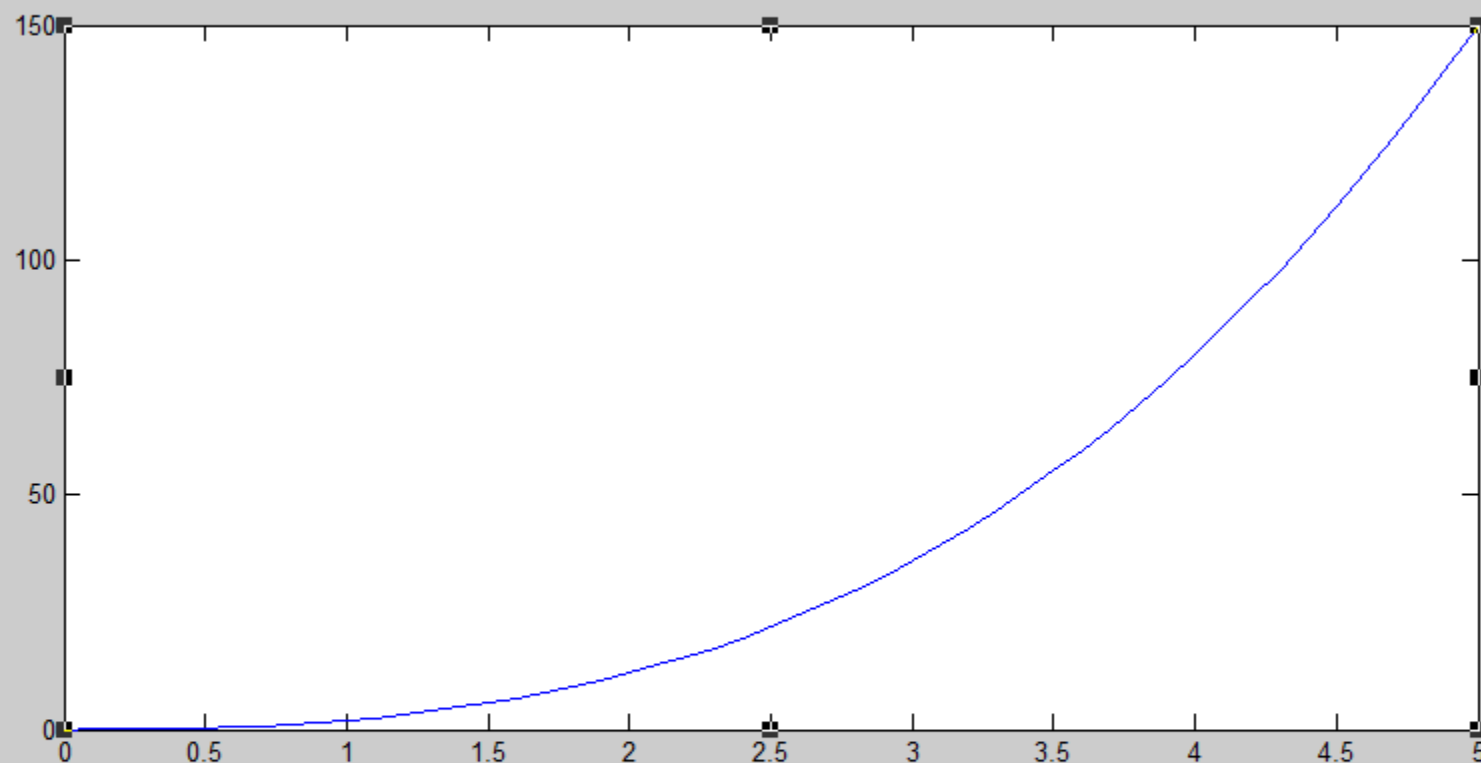
```
>> x=0:0.1:5  
>> y=x.^3+x.  
>> plot(x,y)  
fx >>
```

Workspace

Name	Value
x	<
y	<

Figures - Figure 1

File Edit View Insert Tools Debug Desktop Window Help



Property Editor - Axes

Colors:

Grid: ☐ X ☐ Y ☐ Z

☒ Box

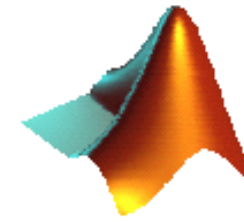
X Axis Y Axis Z Axis Font

X Label:

Ticks...

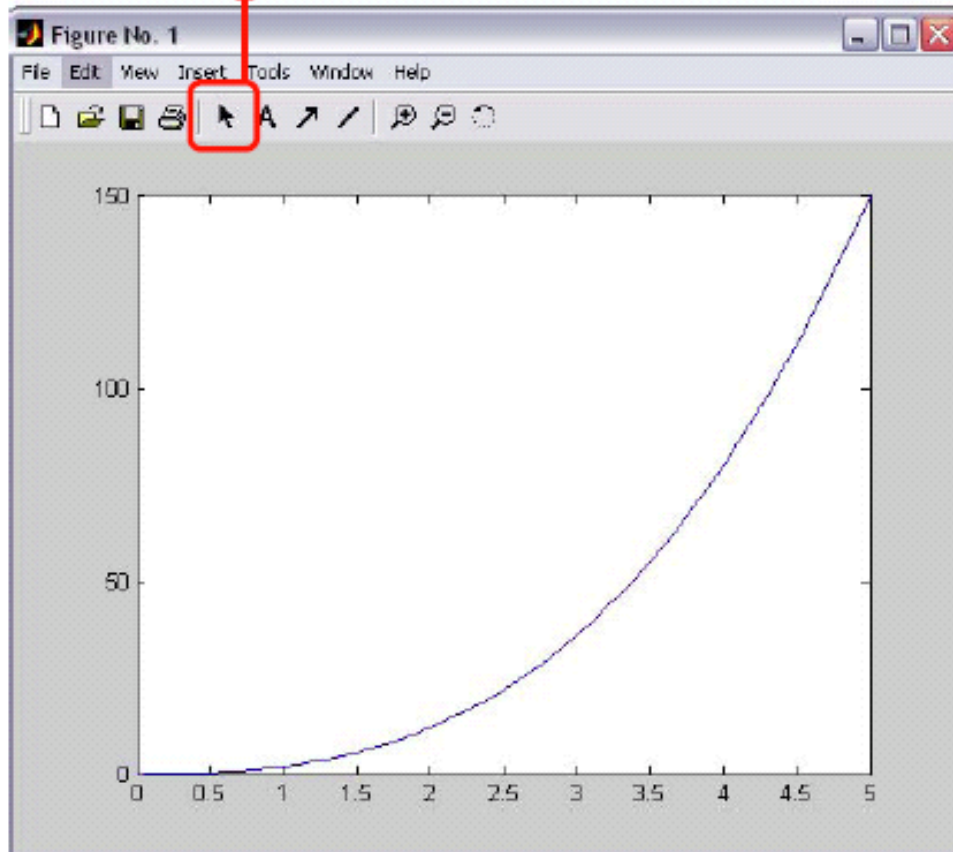
More Properties...

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.

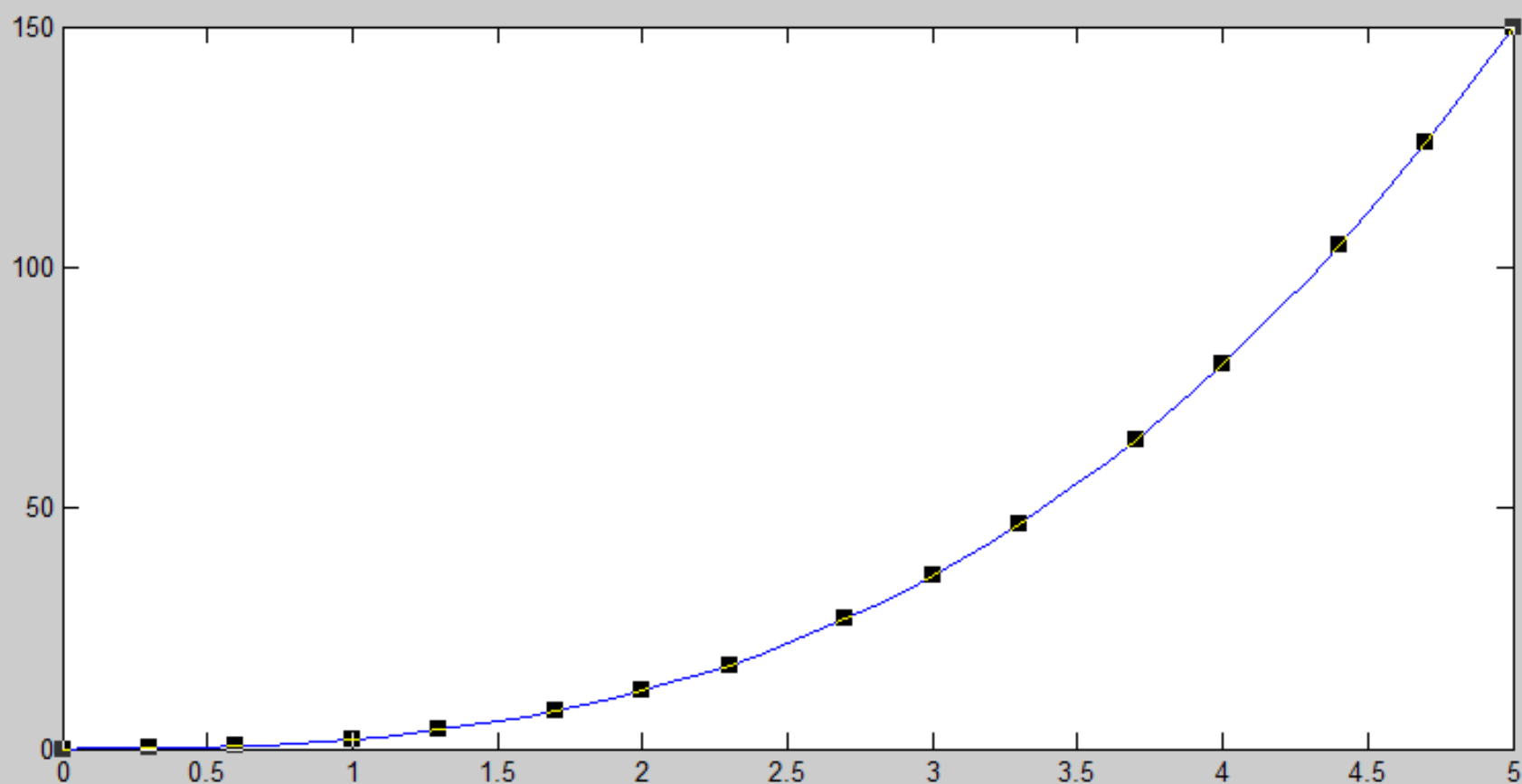
Edit plot



- 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.





Property Editor - Lineseries

Display Name:

Plot Type:

Line

More Properties...

X Data Source:

auto

Y Data Source:

Z Data Source:

Line:

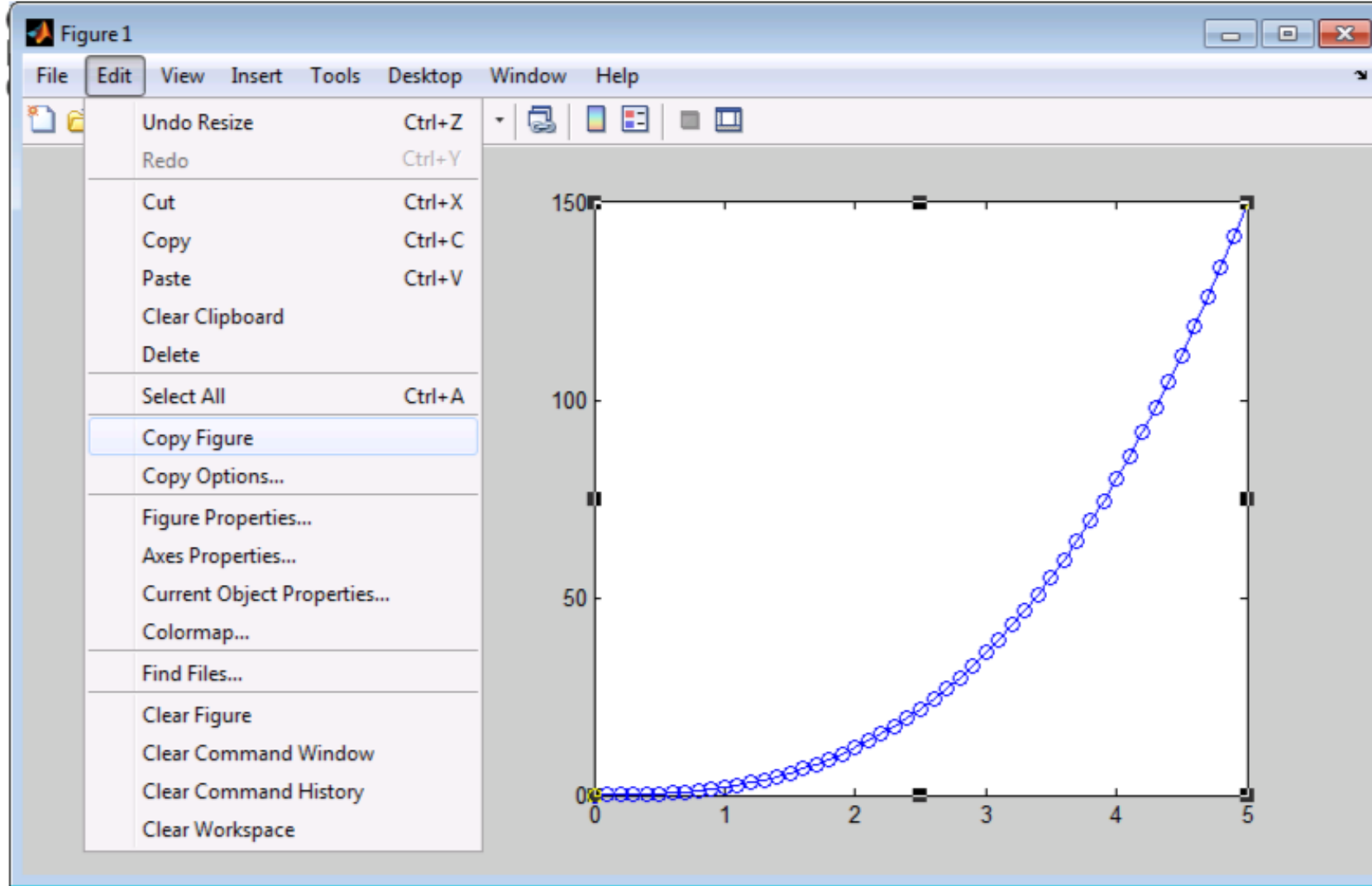
0.5

Marker:

LineStyle

Refresh Data

- Çizilen grafikleri kaydetmek için Figure penceresindeki “File” menüsünden “Save” veya “Save As” seçenekleri seçilir. Grafikler, “fig” uzantılı dosyalar olarak ilgili klasöre kaydedilirler.



Örnek:

$y=e^{2\cos x}$ grafiğini $0 \leq x \leq 4\pi$ aralığında $\pi/15$ artımla, grafik rengi kırmızı, işaretleyici daire ve çizgi türü kesikli olacak şekilde çizdiren `m_file` yazınız.

ezplot Fonksiyonu

Bu fonksiyon iki boyutlu tek değişkenli fonksiyonların grafiklerini bir fikir alabilmek için kabataslak çizdirmeye yarar.

Genel yazım şekli; **ezplot fonksiyon**

$y=\sin(x)/(1+x^2)$ fonksiyonunu çizdirelim.

veya (x ve y ekseninin sınırları da verilmek istenirse)

$y=\sin(x)/(1+x^2)$ fonksiyonunu x ekseninin 0 ile 5 aralığında çizdirelim.

ezplot ('fonksiyon', [xmin,xmax,ymin,ymax])

ezplot da x ve y eksenlerinin minimum ve maksimum değerleri kullanılmaz. Eksenlerin skalası program tarafından otomatik olarak belirlenmiştir. ezplot fonksiyonu ile çizilen grafiklerde fonksiyonun tam karakteristiğini görmek için eksen sınırlarının program tarafından otomatik belirlenmesi önem taşır.

$y=\sin(x)/(1+x^2)$ fonksiyonunu x ekseninin 0 ile 5 aralığında çizdirelim.

Örnek:

$y=\sin(x)/1+x^2$ fonksiyonunu ezplot ile çizdirelim.

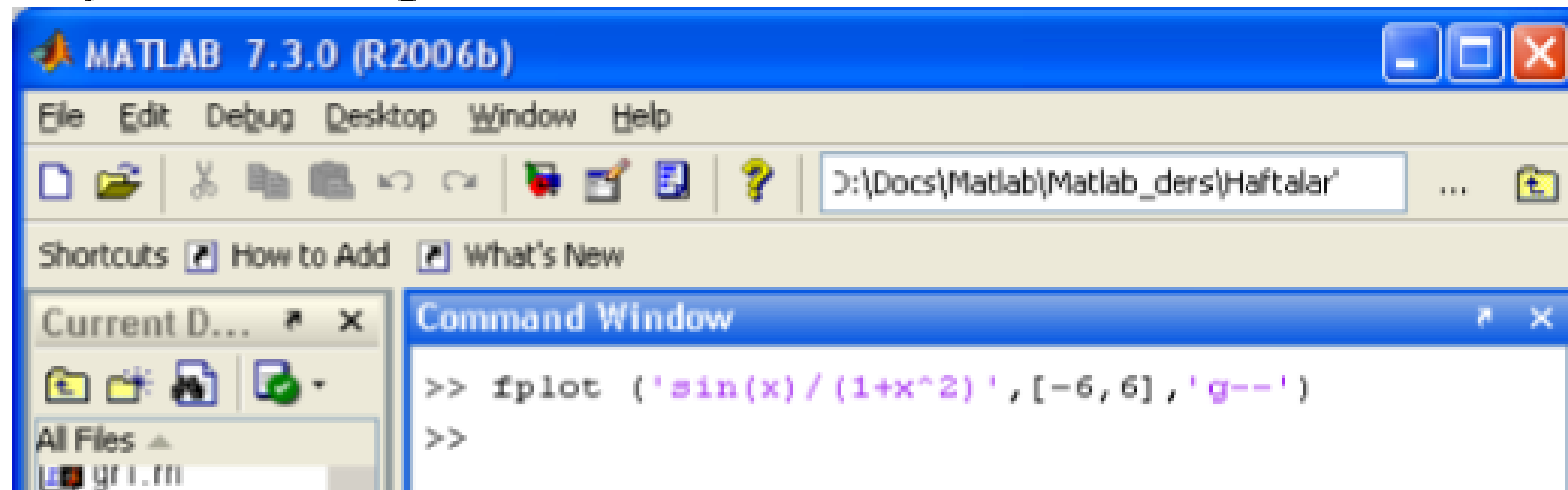
fplot Fonksiyonu

fplot fonksiyonu ezplot fonksiyonu ile aynı işlevi görür ancak ezplot ile yapılamayan çizgi rengi, çizgi türü ve işaretleyici seçenekleri ilave edilebilir.

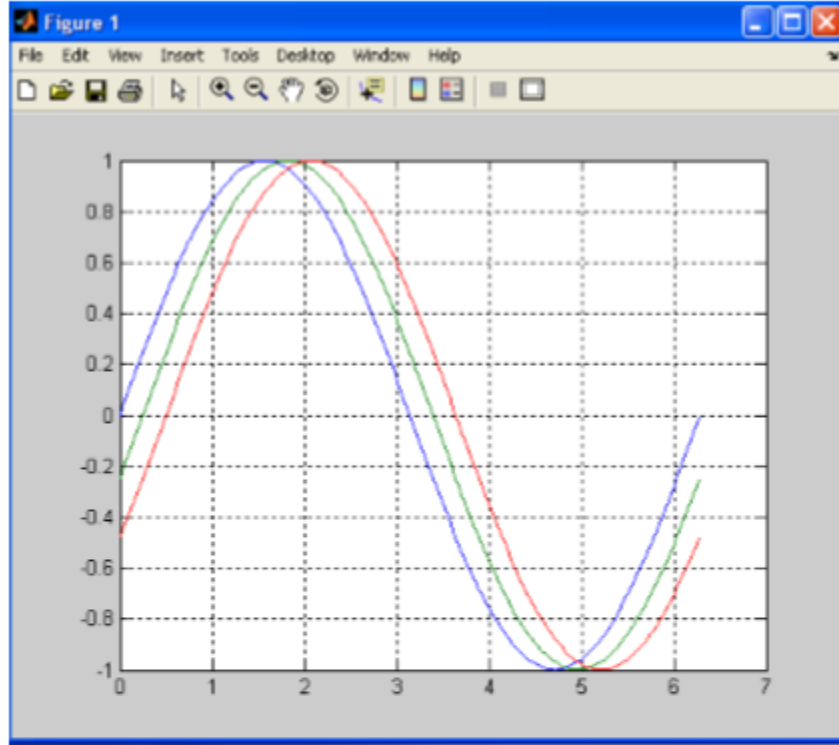
Genel yazım şekli;

fplot ('fonksiyon', [xmin, xmax, ymin, ymax], 'indikatör')

Burada indikatör olarak daha önce çizgi ve işaretleme seçenekleri konusu altında verilen belirleyici karakterler kullanılır. Örneğin çizgi rengini yeşil ve çizgi türünü de kesikli çizgi yapmak için indikatörde 'g--' kullanılır.



Çoklu Grafik Çizimleri



Çoklu grafik çizimi için iki yol vardır:

1. Tek bir plot fonksiyonu kullanarak çizdirmek;
2. Grafiğin üzerine yeni bir grafik eklemek;

1. Tek Bir Plot Fonksiyonuyla Birden Fazla Grafik Çizdirmek

`plot (x,y,w,u,z,t,.....)`

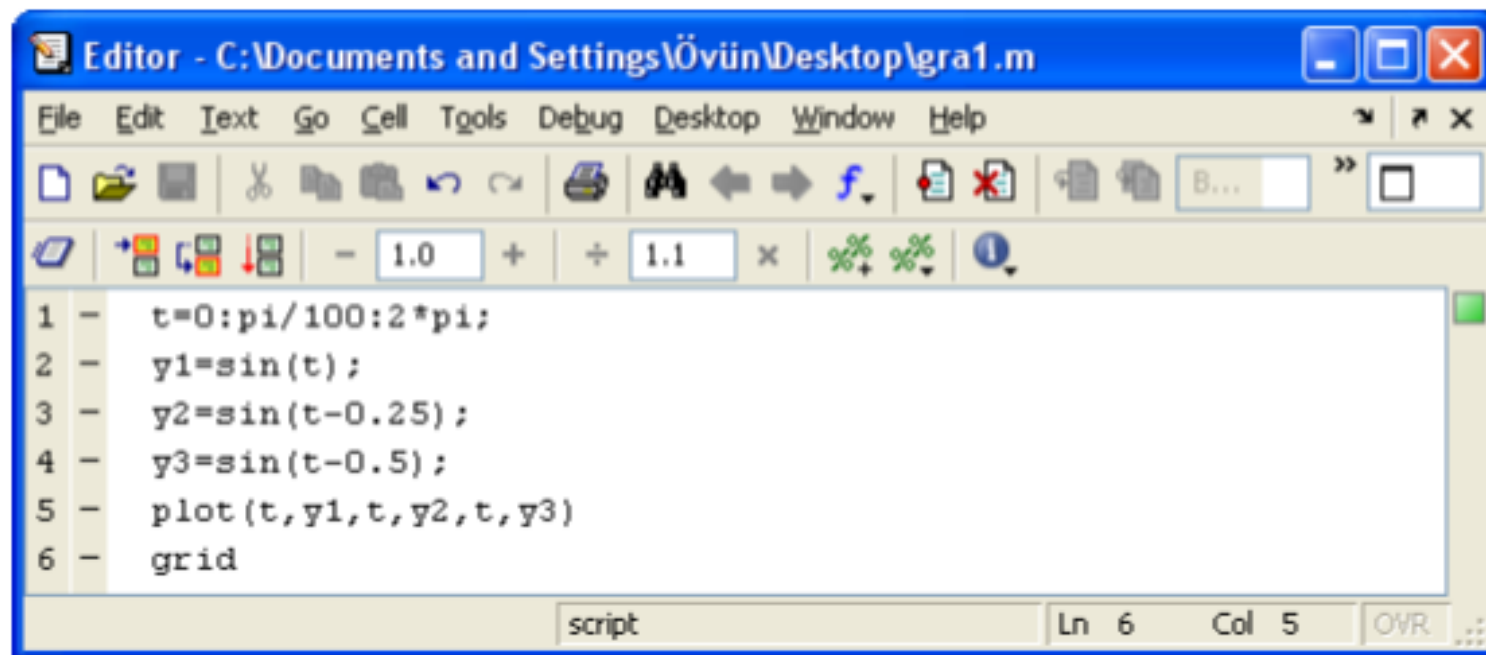
Burada;

x----->y

w----->u

z----->t

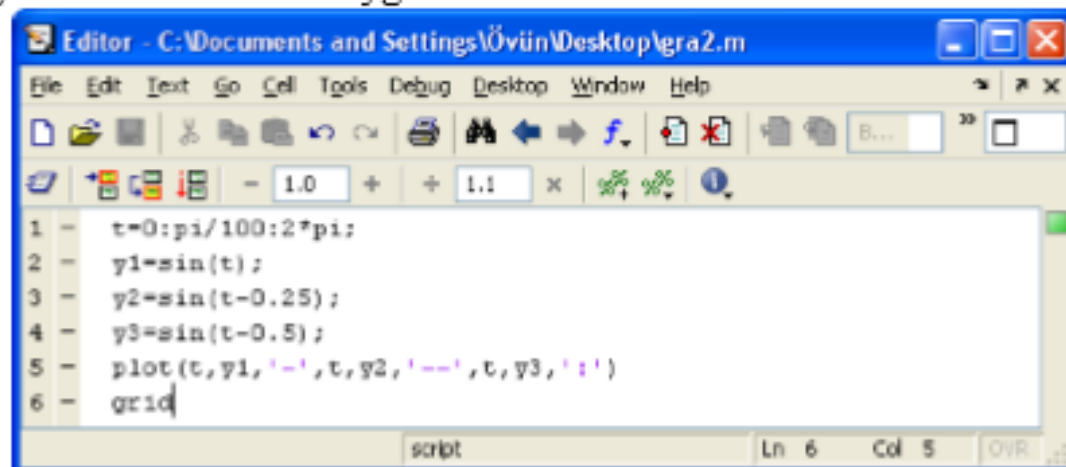
y=sin(t), y=sin(t-0.25) ve y=sin(t-0.5) fonksiyonlarını $\pi/100$ artımla $0 \leq t \leq 2\pi$ aralığında plot fonksiyonunu kullanarak çizdirin.



```
1 - t=0:pi/100:2*pi;
2 - y1=sin(t);
3 - y2=sin(t-0.25);
4 - y3=sin(t-0.5);
5 - plot(t,y1,t,y2,t,y3)
6 - grid
```

Burada varsayılan (default) olarak program tarafından y1 mavi, y2 yeşil ve y3’de kırmızı renklerde çizdirilmiştir.

Çoklu grafiklerde tüm grafikleri daha açık bir şekilde görebilmek için renk, çizgi türü ve işaretleyici simgelerin kullanılması uygun olur.



```
1 - t=0:pi/100:2*pi;
2 - y1=sin(t);
3 - y2=sin(t-0.25);
4 - y3=sin(t-0.5);
5 - plot(t,y1, '-.', t,y2, '-.-', t,y3, ':')
6 - grid
```


$y=\cos(x)$, $y=\cos(2x)$ ve $y=\cos(x/2)$ fonksiyonlarını $2\pi \leq x \leq 2\pi$ aralığında 0.1 artımla çizdiriniz.

2. Grafiğin Üzerine Yeni Bir Grafik Ekleme

Bazı durumlarda çizilmiş bir grafiğin üzerine bir başka grafik eklenmesi gerekebilir. Üst üste birden fazla plot fonksiyonu kullanmak bir önceki grafiği iptal edeceğinden; bunu önlemek için *hold on* fonksiyonu kullanılır.

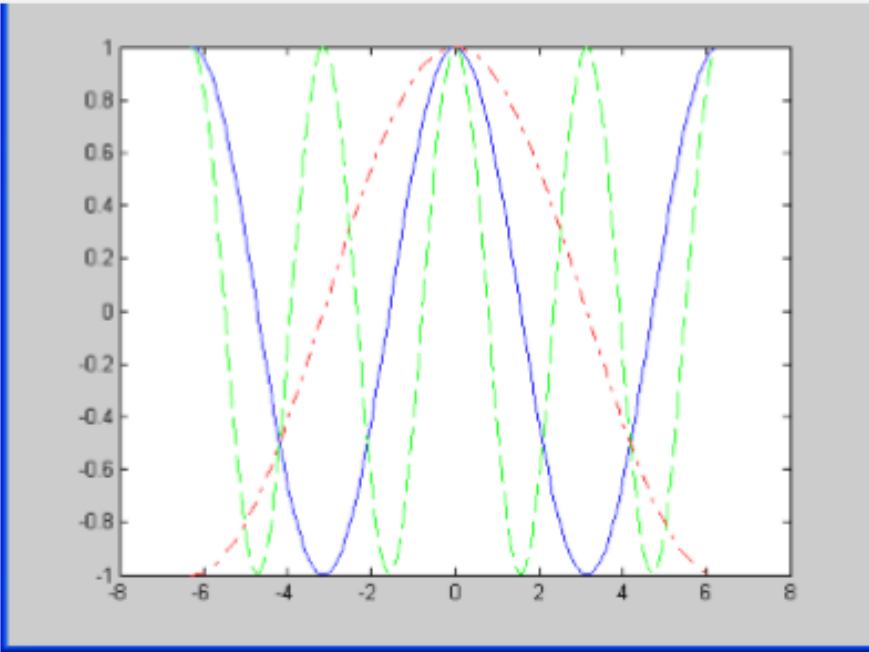
Genel yazım şekli;

plot (x,y1) : ilk grafiği çizer

hold on : grafiği dondurur

plot (x,y2) : yeni bir grafik ekler

hold off : dondurulan grafiği ya da grafikleri serbest bırakır

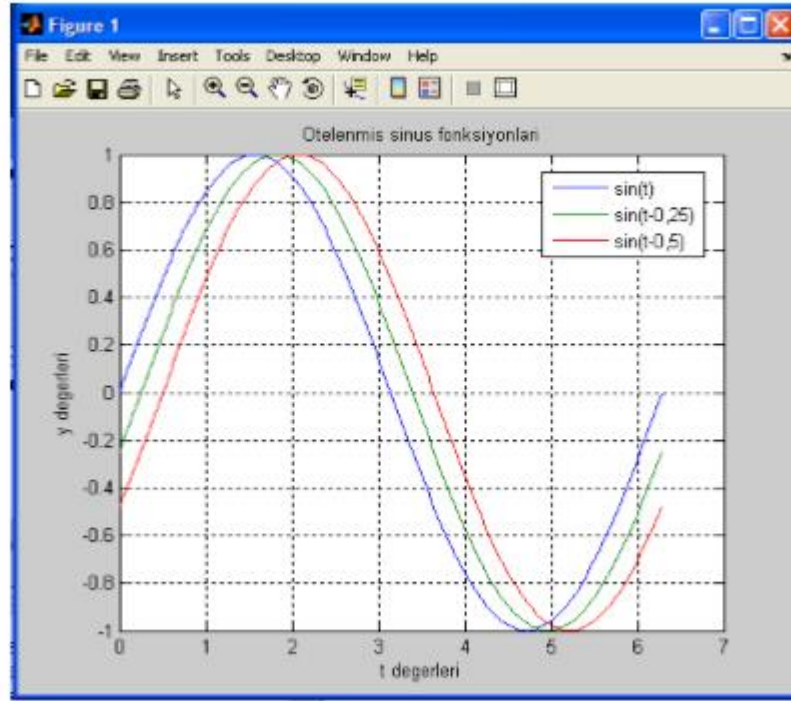


$y=\cos(x)$, $y=\cos(2x)$ ve $y=\cos(x/2)$ fonksiyonlarını `hold on` özelliğini kullanarak $-2\pi \leq x \leq 2\pi$ aralığında 0,1 artımla çizdiriniz.

`hold on` fonksiyonu bir önceki grafiği dondurur ve aynı düzlemde diğer grafiklerin çizilmesini sağlar. `plot` fonksiyonu ile çizilecek olan grafikler ilk grafiğin geçerli eksenlerini kullanır. Ancak çizilecek olan sonraki grafiklerden eksen aralığı en büyük olan kullanılır ve diğer tüm grafiklerin eksenleri bu en büyük aralığa göre otomatik ayarlanır.

```
Editor - D:\Docs\Matlab\Matlab_ders\Haftal...
File Edit Text Go Cell Tools Debug Desktop
[Icons] B...
[Icons] - 1.0 + ÷ 1.1 x % + % -
1 - x=-2*pi:0.1:2*pi;
2 - y1=cos(x);
3 - y2=cos(2*x);
4 - y3=cos(x/2);
5 - plot(x,y1,'b-')
6 - hold on
7 - plot(x,y2,'g--')
8 - hold on
9 - plot(x,y3,'r-.')
10 - hold off
script Ln 10 Col 9 OVR
```

Grafiğe Gösterge Ekleme



Grafikteki gösterge çoklu grafik çizimlerinde ihtiyaç duyulan bir özelliktir. *legend* fonksiyonu kullanılarak uygulanır.

Genel yazım şekli;

legend ('Seri1', 'Seri2', 'Seri3',, Position)

Position gösterge kutusunu belli bir konuma yerleştirir.

**2017 sürümünde
değişti**

Position	İşlevi
0	Verileri kapatmayacak şekilde olası en iyi yeri otomatik belirler.
1	Sağ üst köşe (default yani position'a hiç bir şey yazılmazsa 1 olarak kabul edilir.)
2	Sol üst köşe
3	Sol alt köşe
4	Sağ alt köşe
—1	Grafik alanının dışında sağ taraf

Grafik Üzerine Yazı Ekleme

Grafik üzerinde belirlenen x ve y koordinatlarının kesim noktasından başlayarak yazı eklemek için *text* fonksiyonu kullanılır.

Genel yazım şekli;

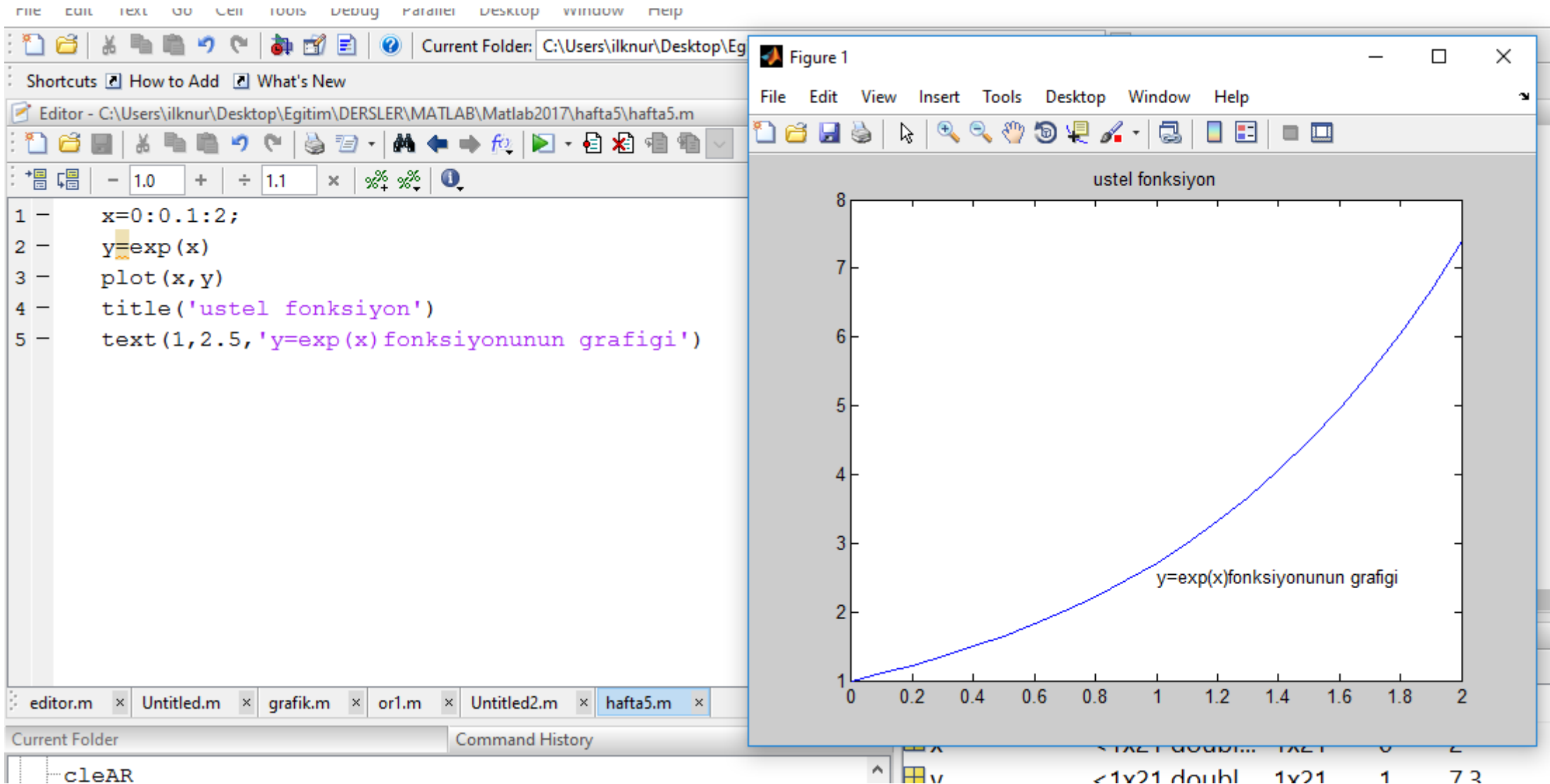
text (xposition, yposition, 'yazı')

xposition : x koordinatı

yposition : y koordinatı

- $y=e^x$ fonksiyonunu $0 \leq x \leq 2$ aralığında 0,1 artımla çizip, (1, 2.5) koordinatlarına 'y=e^x fonksiyonunun grafiği' yazdıralım.

- $y=e^x$ fonksiyonunu $0 \leq x \leq 2$ aralığında 0,1 artımla çizip, (1, 2.5) koordinatlarına ' $y=e^x$ fonksiyonunun grafiği' yazdıralım.

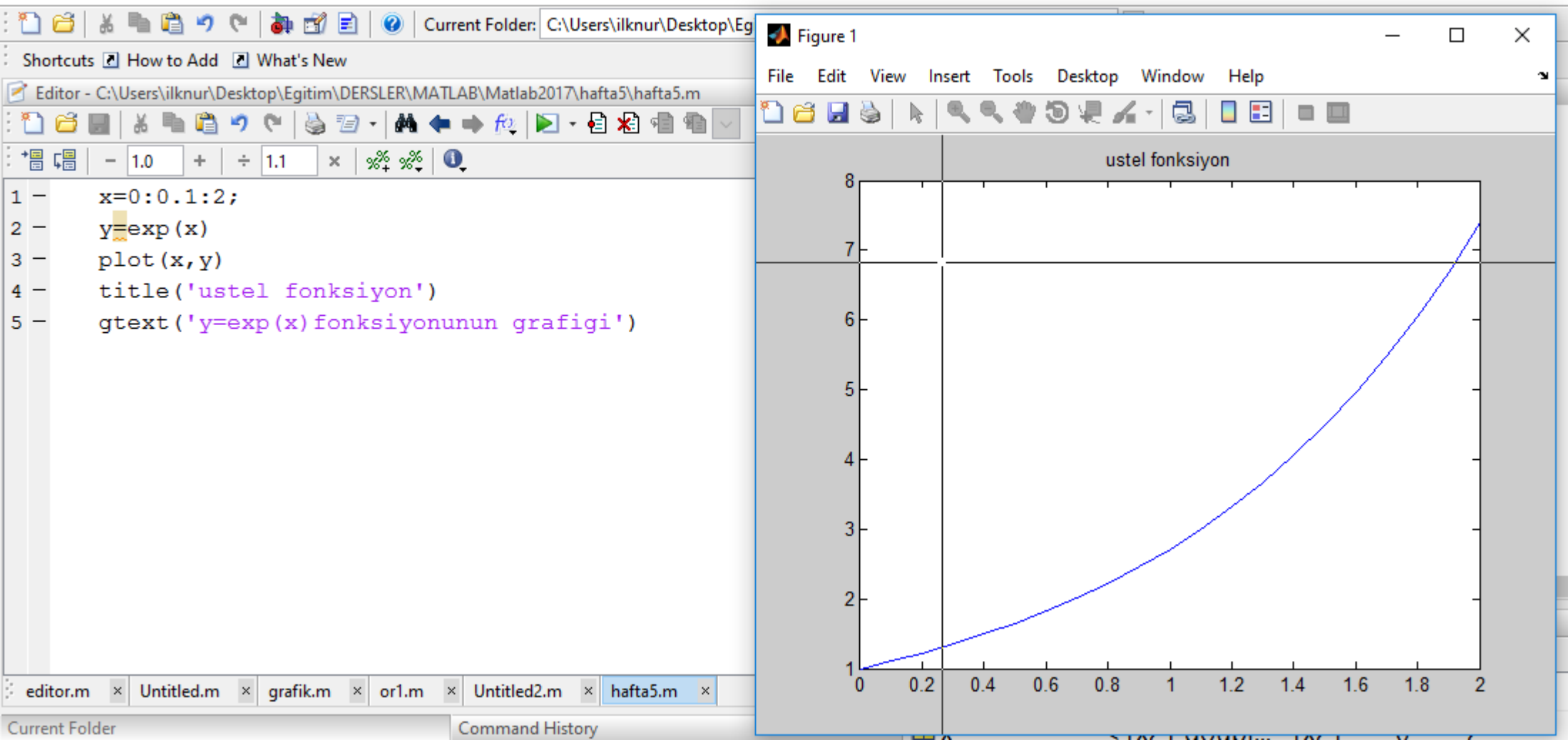


gtext fonksiyonu

Bu fonksiyon text fonksiyonuna göre daha işlevsel ve kullanışlıdır. Grafiği çizdirdikten sonra bu fonksiyon yardımıyla mouse grafik ekranda eksene bir görüntü alır. Eksenlerin kesim noktası yazının konumlandırılacağı yerin başlangıç noktasını gösterir. Arka arkaya bir kaç kez

Genel yazım şekli; gtext ('yazı')

Bu komut tüm grafik biçimlendirme işlemleri (xlabel, ylabel, title v.b.) yapıldıktan sonra en son satırda kullanılır.



gtext fonksiyonu

Genel yazım şekli; gtext ('yazı')

Bu fonksiyon text fonksiyonuna göre daha işlevsel ve

ku
ya
Ek
h

çimlendirme işlemleri (xlabel, ylabel,
onra en son satırda kullanılır.

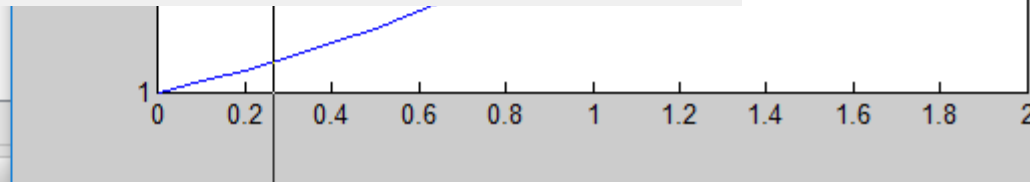
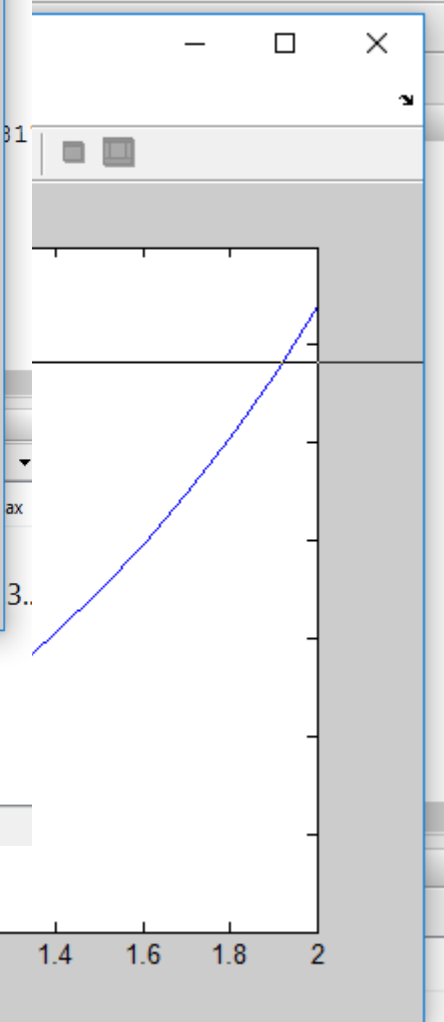
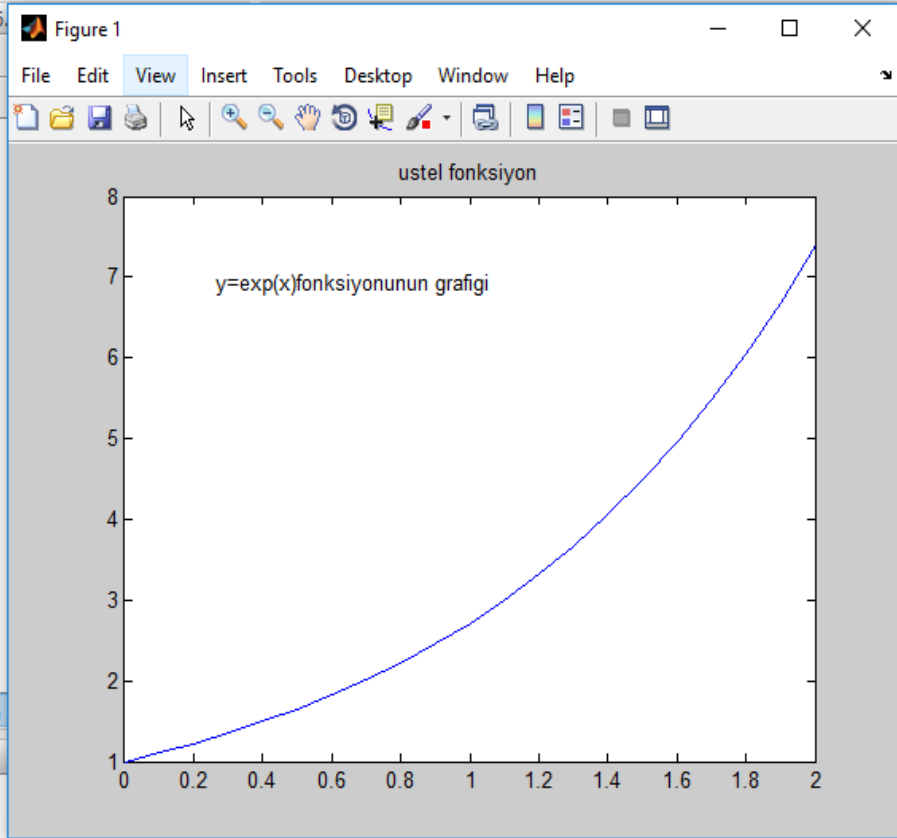
```
1 - x=0:0.1:2;  
2 - y=exp(x)  
3 - plot(x,y)  
4 - title('ustel fonksiyon')  
5 - gtext('y=exp(x) fonksiyonunun grafigi')
```

editor.m x Untitled.m x grafik.m x or1.m x Untitled2.m x hafta5.m

Current Folder Command History

```
clear  
CLEAR  
clear  
ezplot sin(x)/(1+x^2)  
clc  
%-- 3/22/2017 8:37 AM --%
```

Start



Tek Bir Sayfa Üzerine Birden Fazla Grafik Yerleştirme

Aynı düzlem üzerinde ve aynı eksen takımı kullanılarak tek bir grafik (figure) penceresinde birden fazla grafik çizdirmek için subplot fonksiyonu kullanılır

Genel yazım biçimi;

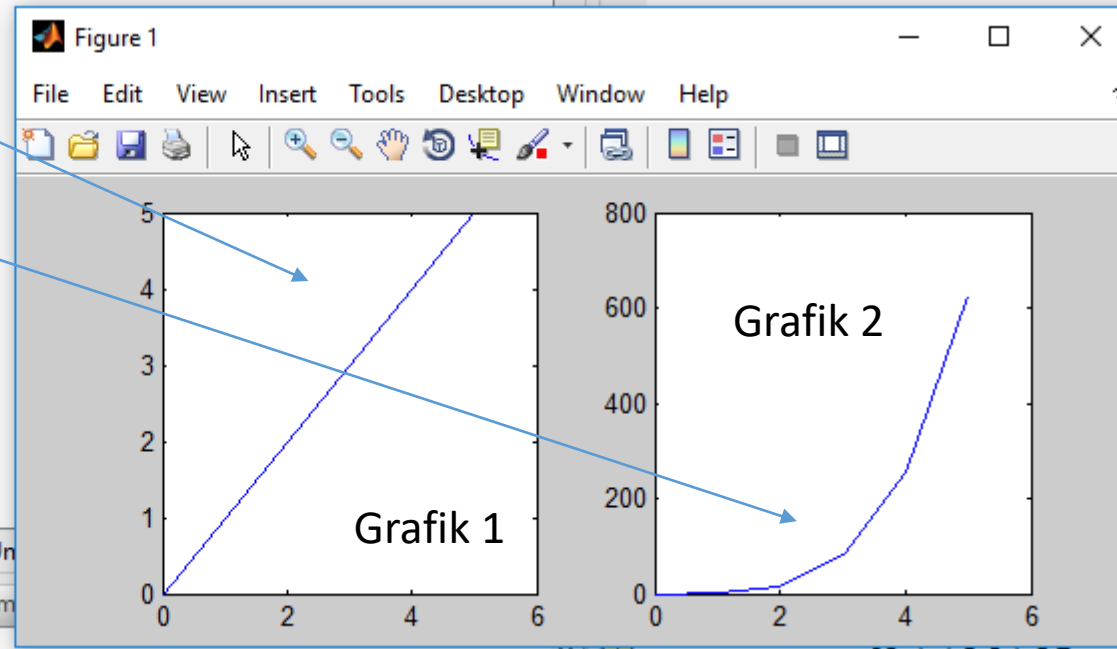
subplot (m,n,p)

m : satır sayısı n : sütun sayısı

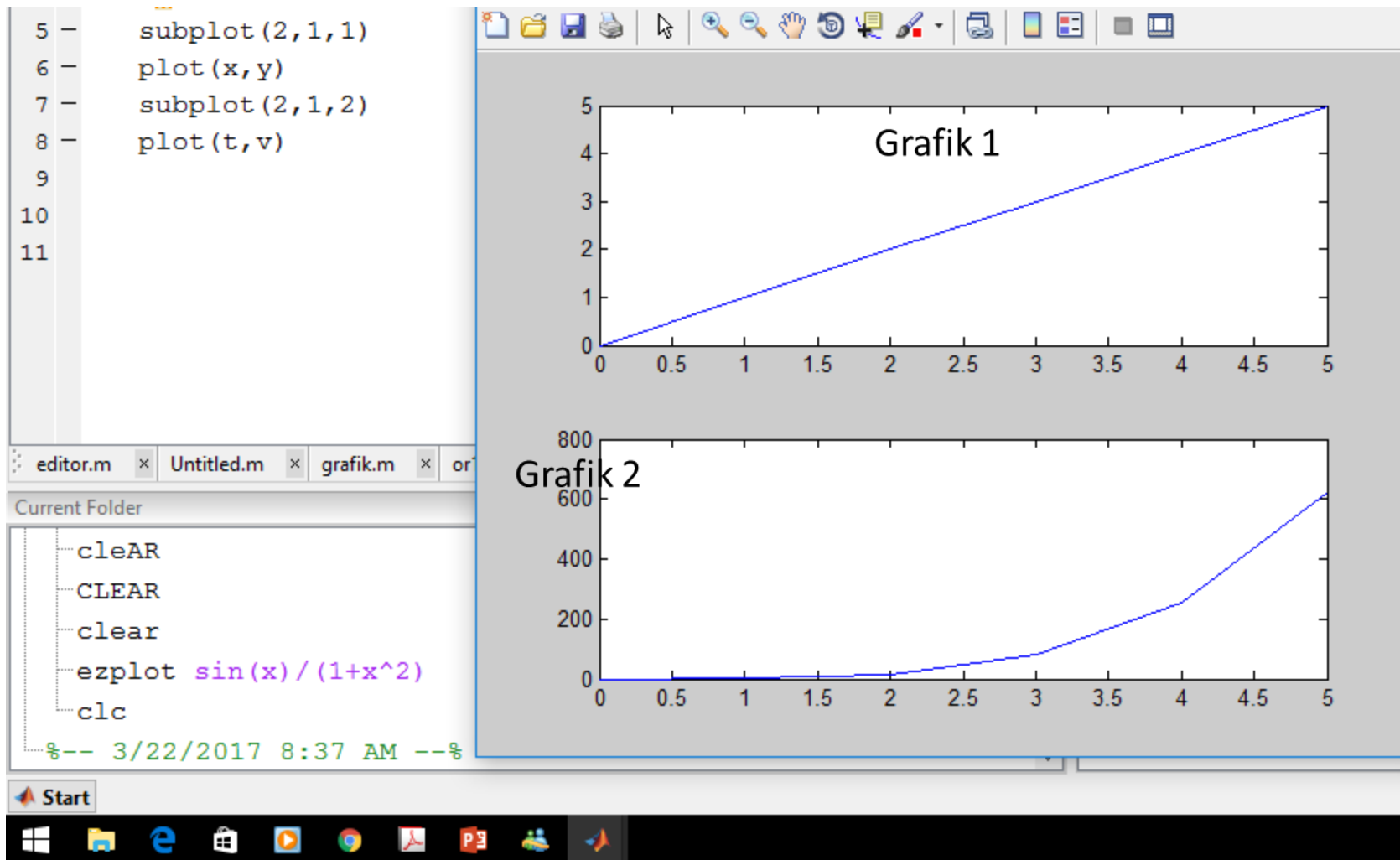
p : indeks (1 ile m.n arasında olmalıdır. Satırda soldan sağa, sütunda yukarıdan aşağıya doğru grafik belirlemesi yapar.)

1 satır, 2 sütun

```
5 - subplot(1,2,1)
6 - plot(x,y)
7 - subplot(1,2,2)
8 - plot(t,v)
```



2 Satir, 1 sutun



subplot(2,3,1) **2 satır 3 sütun**
plot (x,y)

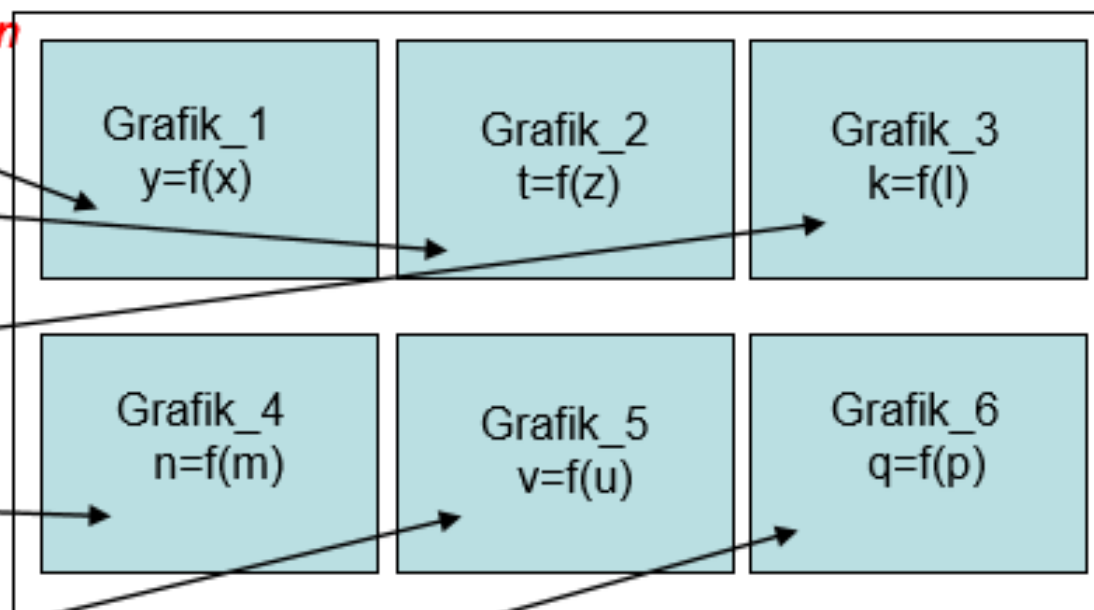
subplot (2,3,2)
plot (z,t)

subplot (2,3,3)
plot (l,k)

subplot (2,3,4)
plot (m,n)

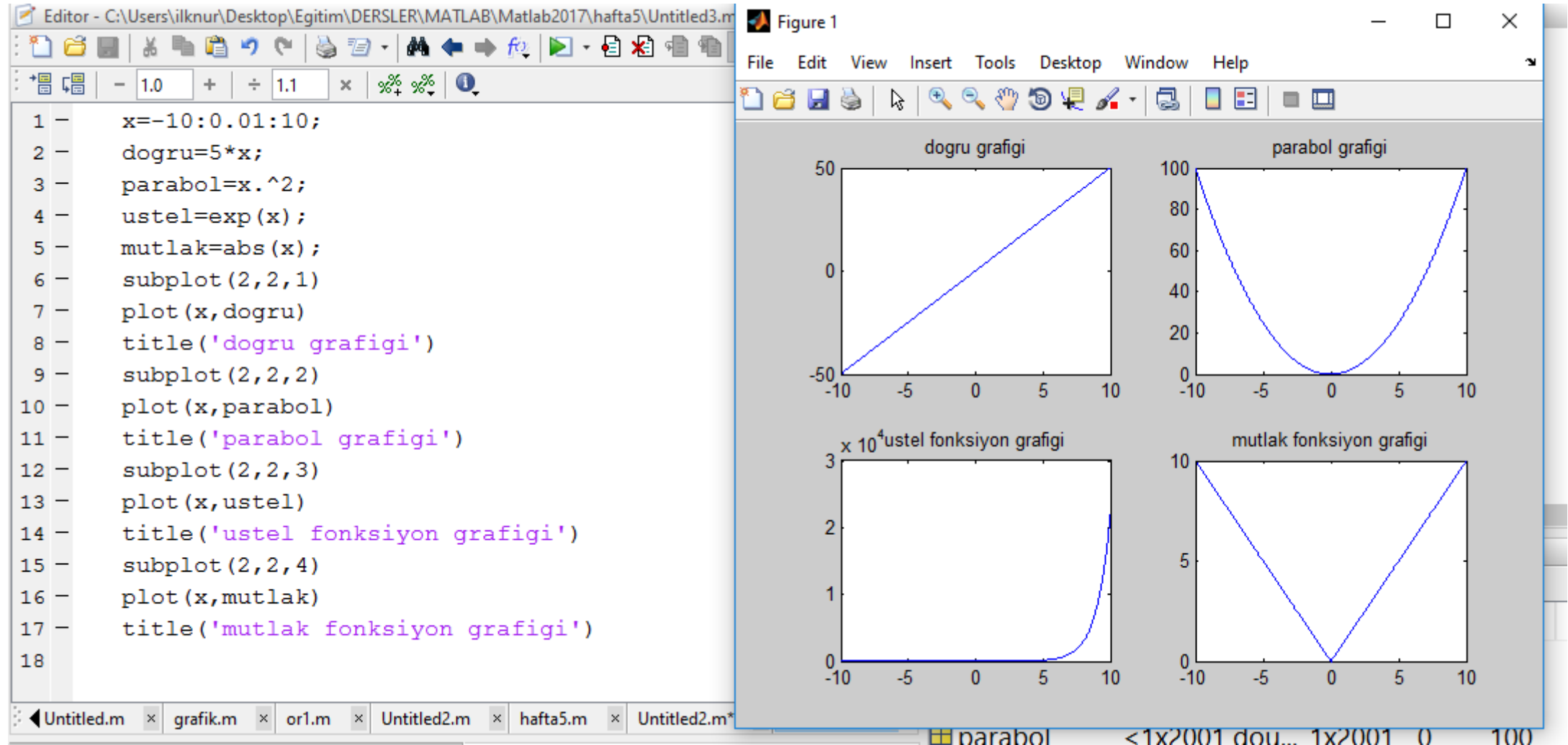
subplot (2,3,5)
plot (u,v)

subplot (2,3,6)
plot (p,q)

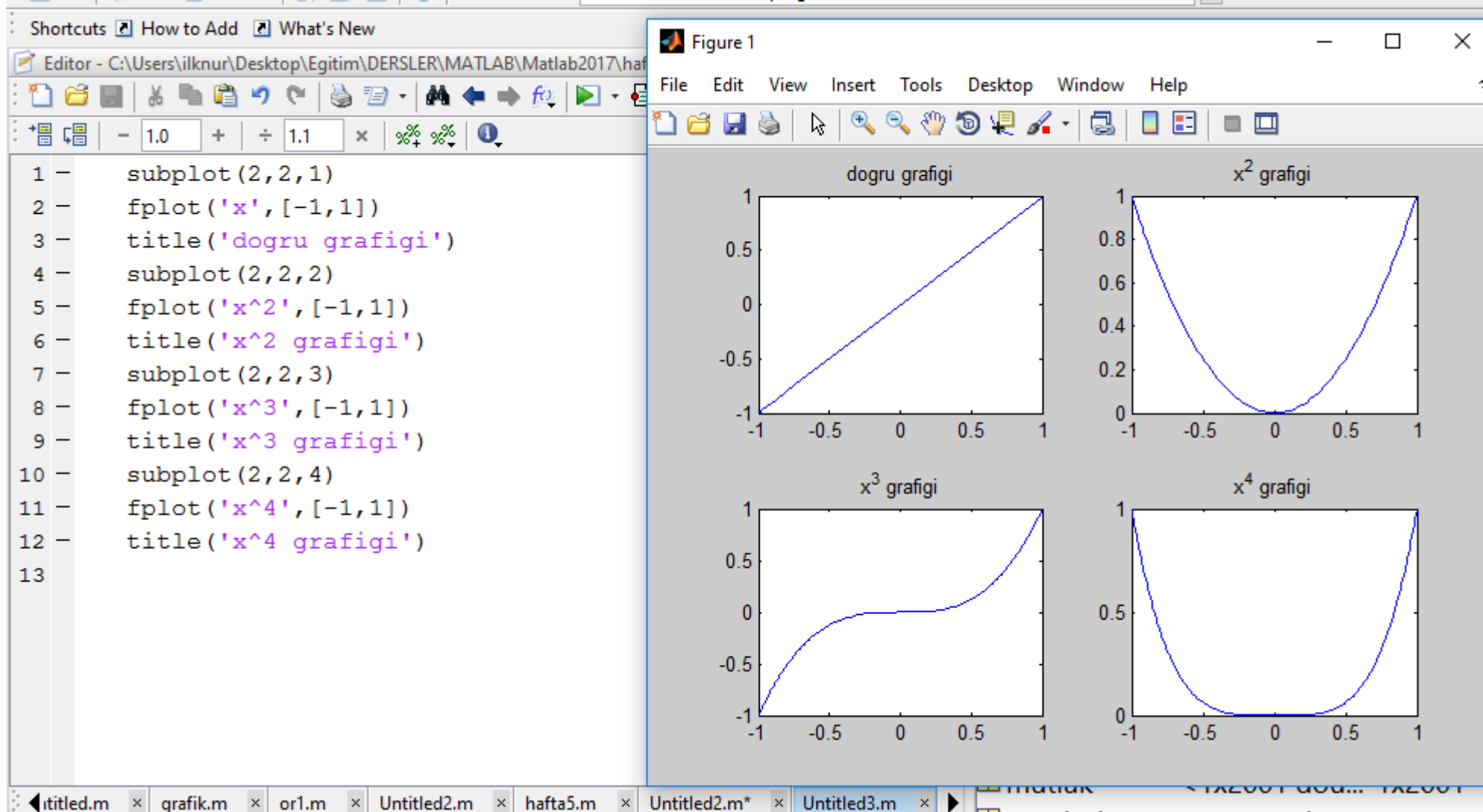


- $y=5x$, $y=x^2$, $y=e^x$ ve $y=|x|$ fonksiyonlarını 2 satır 2 sütun şeklinde aynı figure penceresinde $-10 \leq x \leq 10$ aralığında 0,01 artımla çizdirin. (grafik üzerinde fonksiyon isimlerini belirtiniz)
- fplot fonksiyonunu kullanarak aynı figure penceresinde 2 satır 2 sütun halinde 4 adet grafik çizdirelim. $(-1,1$ aralığında $y=x$, $y=x^2$, $y=x^3$, $y=x^4$)

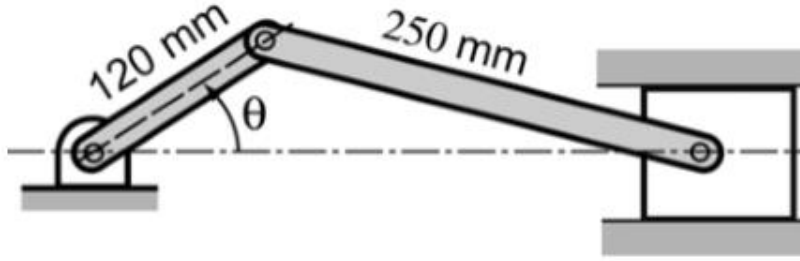
- $y=5x$, $y=x^2$, $y=e^x$ ve $y=|x|$ fonksiyonlarını 2 satır 2 sütun şeklinde aynı figure penceresinde $-10 \leq x \leq 10$ aralığında 0,01 artımla çizdirin. (grafik üzerinde fonksiyon isimlerini belirtiniz)



- **fplot** fonksiyonunu kullanarak aynı figure penceresinde 2 satır 2 sütun halinde 4 adet grafik çizdirelim. (-1,1 aralığında $y=x$, $y=x^2$, $y=x^3$, $y=x^4$)



Piston-çubuk-krank mekanizması pek çok mühendislik uygulamasında kullanılır. Aşağıdaki şekilde gösterilen mekanizmada, krank 500 rpm sabit bir hızda dönüyor.



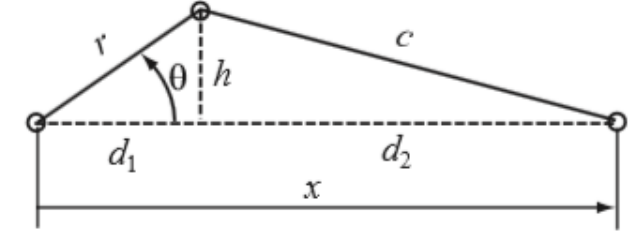
Krank sabit bir açısal hızla dönüyor. Bu, olarak Kabul edersek

$\theta = 0^\circ$ when $t = 0$, then at time t the angle θ is given by $\theta = \dot{\theta}t$, and that $\ddot{\theta} = 0$ at all times.

The distances d_1 and h are given by:

$$d_1 = r \cos \theta \quad \text{and} \quad h = r \sin \theta$$

With h known, the distance d_2 can be calculated using the Pythagorean Theorem:



$$d_2 = (c^2 - h^2)^{1/2} = (c^2 - r^2 \sin^2 \theta)^{1/2}$$

The position x of the piston is then given by:

$$x = d_1 + d_2 = r \cos \theta + (c^2 - r^2 \sin^2 \theta)^{1/2}$$

The derivative of x with respect to time gives the velocity of the piston:

$$\dot{x} = -r\dot{\theta} \sin \theta - \frac{r^2 \dot{\theta} \sin 2\theta}{2(c^2 - r^2 \sin^2 \theta)^{1/2}}$$

The second derivative of x with respect to time gives the acceleration of the piston:

$$\ddot{x} = -r\dot{\theta}^2 \cos \theta - \frac{4r^2 \dot{\theta}^2 \cos 2\theta (c^2 - r^2 \sin^2 \theta) + (r^2 \dot{\theta} \sin 2\theta)^2}{4(c^2 - r^2 \sin^2 \theta)^{3/2}}$$

```

THDrpm=500; r=0.12; c=0.25;
THD=THDrpm*2*pi/60;
tf=2*pi/THD;
t=linspace(0,tf,200);
TH=THD*t;
d2s=c^2-r^2*sin(TH).^2;
x=r*cos(TH)+sqrt(d2s);
xd=-r*THD*sin(TH)-(r^2*THD*sin(2*TH))./(2*sqrt(d2s));
xdd=-r*THD^2*cos(TH)-(4*r^2*THD^2*cos(2*TH).*d2s+
(r^2*sin(2*TH)*THD).^2)./(4*d2s.^(3/2));
subplot(3,1,1)
plot(t,x)
grid
xlabel('Time (s)')
ylabel('Position (m)')
subplot(3,1,2)
plot(t,xd)
grid
xlabel('Time (s)')
ylabel('Velocity (m/s)')
subplot(3,1,3)
plot(t,xdd)
grid
xlabel('Time (s)')
ylabel('Acceleration (m/s^2)')

```

Define $\dot{\theta}$, r , and c .

Change the units of $\dot{\theta}$ from rpm to rad/s.

Calculate the time for one revolution of the crank.

Create a vector for the time with 200 elements.

Calculate θ for each t .

Calculate d_2 squared for each t .

Calculate x for each t .

Calculate $\dot{x}$ and $\ddot{x}$ for each t .

Plot x vs. t .

Format the first plot.

Plot $\dot{x}$ vs. t .

Format the second plot.

Plot $\ddot{x}$ vs. t .

Format the third plot.

$\theta = 0^\circ$ when $t = 0$, then at time t the angle θ is given by $\theta = \dot{\theta}t$, and that $\ddot{\theta} = 0$ at all times.

The distances d_1 and h are given by:

$$d_1 = r \cos \theta \quad \text{and} \quad h = r \sin \theta$$

With h known, the distance d_2 can be calculated using the Pythagorean Theorem:

$$d_2 = (c^2 - h^2)^{1/2} = (c^2 - r^2 \sin^2 \theta)^{1/2}$$

The position x of the piston is then given by:

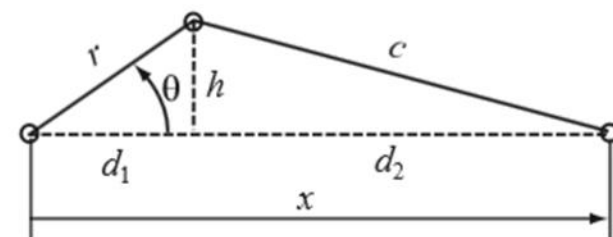
$$x = d_1 + d_2 = r \cos \theta + (c^2 - r^2 \sin^2 \theta)^{1/2}$$

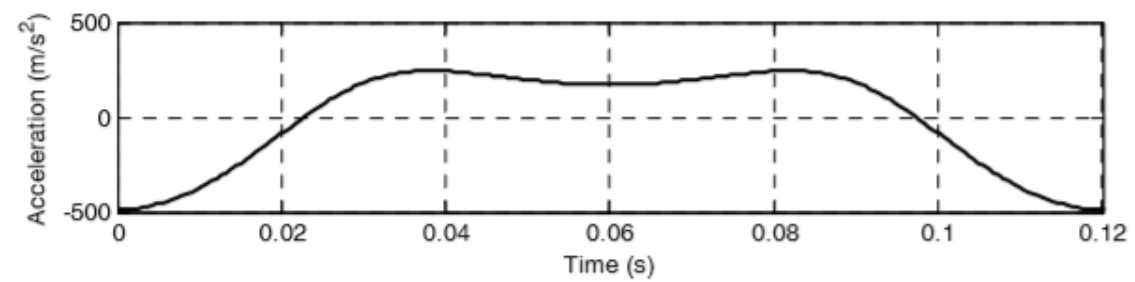
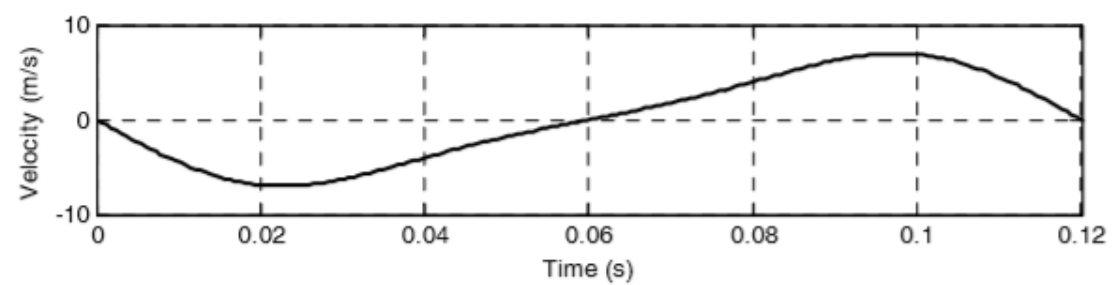
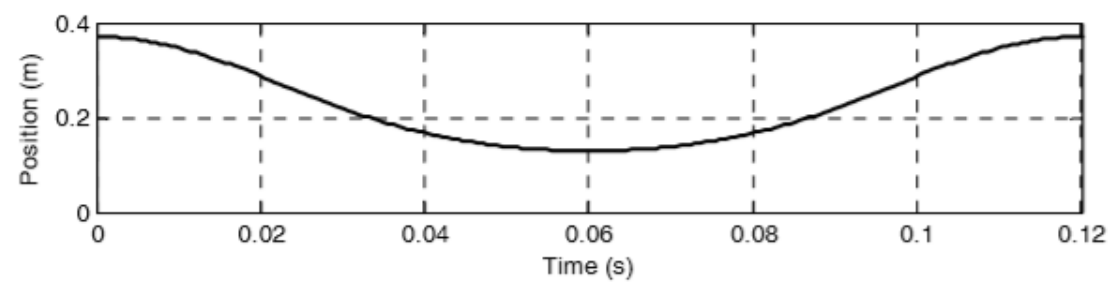
The derivative of x with respect to time gives the velocity of the piston:

$$\dot{x} = -r\dot{\theta} \sin \theta - \frac{r^2 \dot{\theta} \sin 2\theta}{2(c^2 - r^2 \sin^2 \theta)^{1/2}}$$

The second derivative of x with respect to time gives the acceleration of the piston:

$$\ddot{x} = -r\dot{\theta}^2 \cos \theta - \frac{4r^2 \dot{\theta}^2 \cos 2\theta (c^2 - r^2 \sin^2 \theta) + (r^2 \dot{\theta} \sin 2\theta)^2}{4(c^2 - r^2 \sin^2 \theta)^{3/2}}$$





$y=3x^3-26x+10$ fonksiyonunu, 1. ve 2. türevlerini $-2 \leq x \leq 4$ aralığı için çizerek aynı grafikte gösteriniz

çözüm

fonksiyonun : $y=3x^3-26x+10$

fonksiyonun 1. türevi : $y'=9x^2-26$

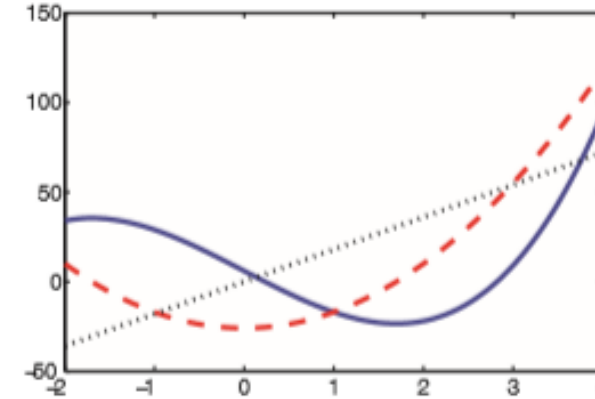
fonksiyonun 2. türevi : $y''=18x$

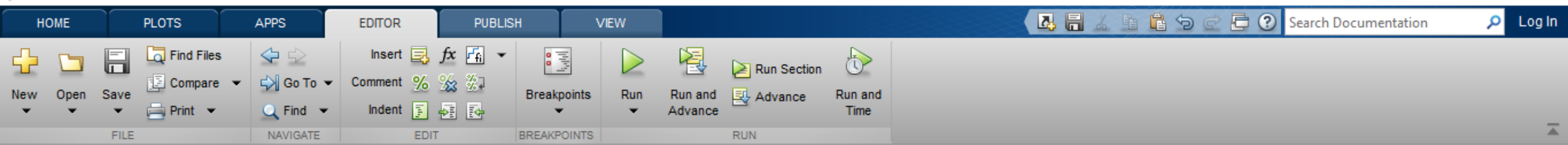
x vektörü oluşturulur ve y, y', y'' değerleri hesaplatılarak grafik çizdirilir.

```
x=[-2:0.01:4];  
y=3*x.^3-26*x+10;  
yd=9*x.^2-26;  
ydd=18*x;  
plot(x,y,'-b',x,yd,'--r',x,ydd,':k')
```

Create vector x with the domain of the function.
Create vector y with the function value at each x .
Create vector yd with values of the first derivative.
Create vector ydd with values of the second derivative.
Create three graphs, y vs. x , yd vs. x , and ydd vs. x , in the same figure.

The plot that is created is shown in Figure 5-7.





C:\Users\ilknur\Documents\MATLAB

Current Folder Editor - grafik.m

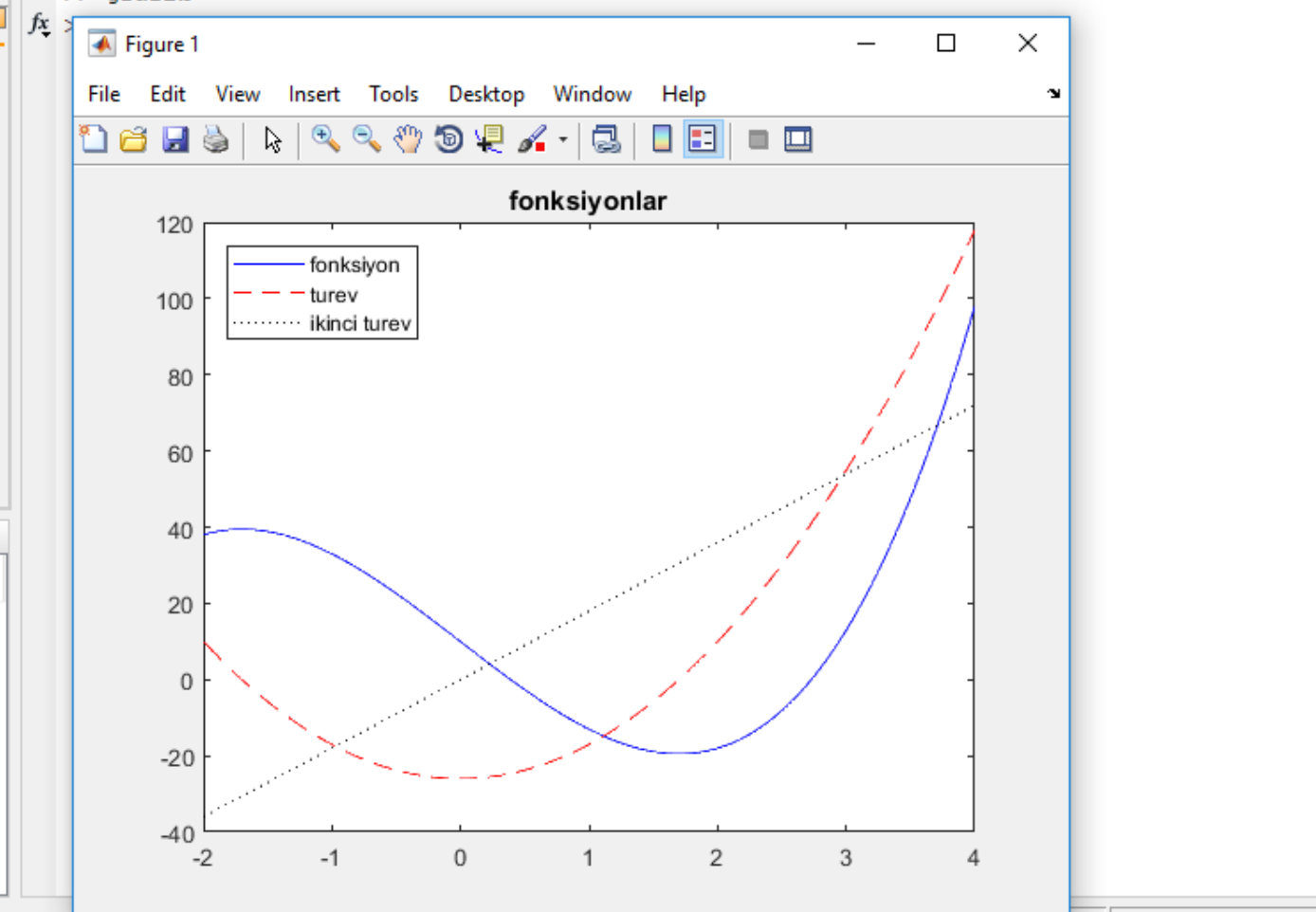
```
1 x = [-2:0.01:4];  
2 y = 3.*x.^3-26.*x+10;  
3 yd = 9.*x.^2-26;  
4 ydd = 18.*x;  
5 plot(x,y,'-b',x,yd,'--r',x,ydd,':k')  
6 title('fonksiyonlar')  
7 legend('fonksiyon','turev','ikinci turev','location','northwest')  
8  
9  
10
```

Workspace

Name	Value
x	1x601 double
y	1x601 double
yd	1x601 double
ydd	1x601 double

Command Window

>> grafik



script

Ln 10 Col 1

```
x=[-2:0.01:4];
y=3*x.^3-26*x+6;
yd=9*x.^2-26;
ydd=18*x;
plot(x,y,'-b')
hold on
plot(x,yd,'--r')
plot(x,ydd,':k')
hold off
```

```
x=[-2:0.01:4];
y=3*x.^3-26*x+6;
yd=9*x.^2-26;
ydd=18*x;
plot(x,y,'LineStyle','-','color','b')
line(x,yd,'LineStyle','--','color','r')
line(x,ydd,'linestyle',':','color','k')
```

5.3.3 Using the *line* Command

With the `line` command additional graphs (lines) can be added to a plot that already exists. The form of the line command is:

`line(x,y,'PropertyName',PropertyValue)`

(Optional) Properties with values that can be used to specify the line style, color, and width, marker type, size, and edge and fill colors.

The format of the `line` command is almost the same as the `plot` command (see Section 5.1). The `line` command does not have the line specifiers, but the line style, color, and marker can be specified with the Property Name and property value features. The properties are optional and if none are entered MATLAB uses default properties and values. For example, the command:

```
line(x,y,'linestyle','--','color','r','marker','o')
```

will add a dashed red line with circular markers to a plot that already exists.

The major difference between the `plot` and `line` commands is that the `plot` command starts a new plot every time it is executed, while the `line` command adds lines to a plot that already exists. To make a plot that has several graphs, a `plot` command is typed first and then line commands are typed for additional graphs. (If a line command is entered before a `plot` command an error message is displayed.)

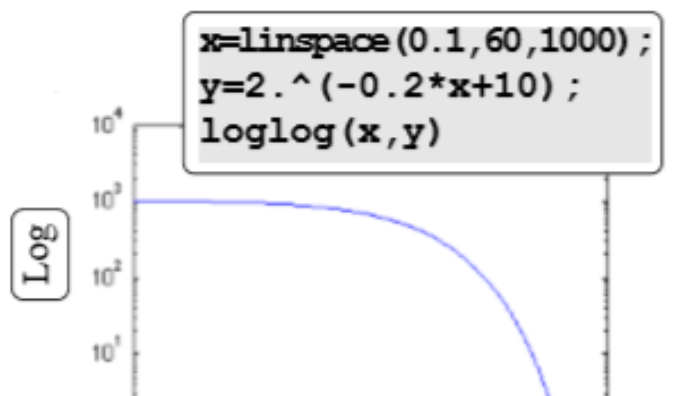
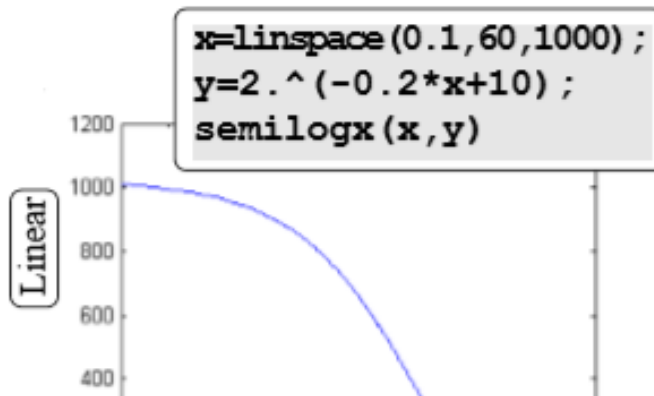
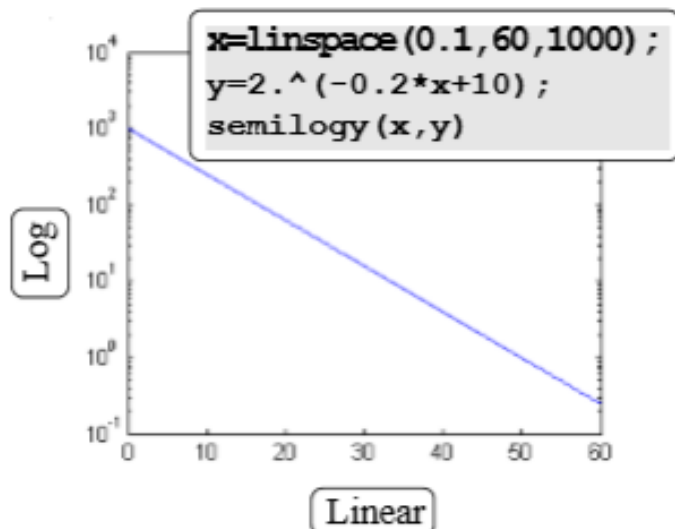
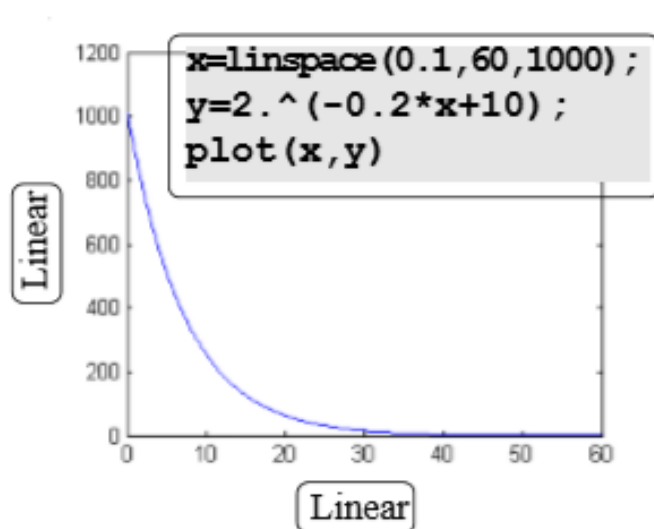
5.5 PLOTS WITH LOGARITHMIC AXES

Many science and engineering applications require plots in which one or both axes have a logarithmic (log) scale. Log scales provide means for presenting data over a wide range of values. It also provides a tool for identifying characteristics of data and possible forms of mathematical relationships that can be appropriate for modeling the data (see Section 8.2.2).

MATLAB commands for making plots with log axes are:

<code>semilogy(x, y)</code>	Plots y versus x with a log (base 10) scale for the y axis and linear scale for the x axis.
<code>semilogx(x, y)</code>	Plots y versus x with a log (base 10) scale for the x axis and linear scale for the y axis.
<code>loglog(x, y)</code>	Plots y versus x with a log (base 10) scale for both axes.

Line specifiers and Property Name and Property Value arguments can be added to the commands (optional) just as in the `plot` command. As an example, Figure 5-9 shows a plot of the function $y = 2^{(-0.2x + 10)}$ for $0.1 \leq x \leq 60$. The figure shows four plots of the same function: one with linear axes, one with a log scale for the y axis, one with a log scale for the x axis, and one with a log scale on both axes.



Notes for plots with logarithmic axes:

- The number zero cannot be plotted on a log scale (since a log of zero is not defined).
- Negative numbers cannot be plotted on log scales (since a log of a negative number is not defined).

İki y eksenli Grafik Çizmek

Bazı durumlarda verileri y ekseninin hem sol hem de sağ tarafında ölçeklemek gerebilir. Bu durumda **plotyy fonksiyonu** kullanılır.

Genel yazım şekli; plotyy (x1,y1,x2,y2)

X1 değerlerine ait y1 grafiği y ekseninin sol tarafında ölçeklendirilir. X2 değerlerine ait y2 grafiği y ekseninin sağ tarafında ölçeklendirilir

- **$y=e^{-x}\sin(x)$ ve $y=e^x\cos(x)$ fonksiyonlarını $1\leq x\leq 10$ aralığında 0,01 artımla y1 değerleri ve y2 değerleri farklı eksenlerde olacak şekilde çizdirin.**

Eksen Sınırlarını Ayarlamak

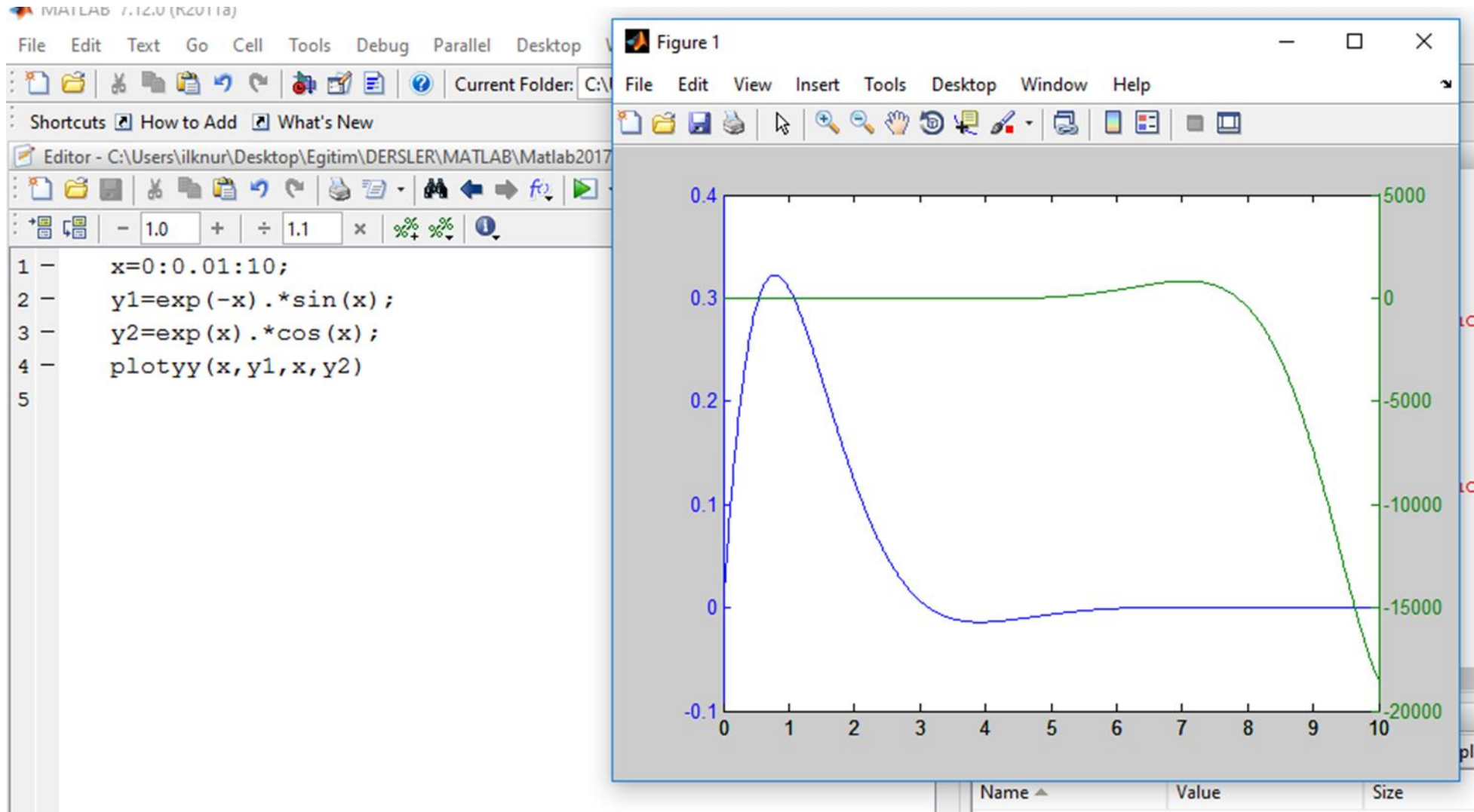
Eksen sınırları programcı tarafından da belirlenebilir. Eksen sınırlarını belirlemek için temel fonksiyon axis fonksiyonudur. Bu fonksiyon var olan bir grafik üzerinde hem x ekseninin hem de y ekseninin geçerli sınırlarını değiştirebilir ve istenilen bir eksen ölçeği belirleyebilir.

Genel yazım şekli; axis ([xmin xmax ymin ymax])

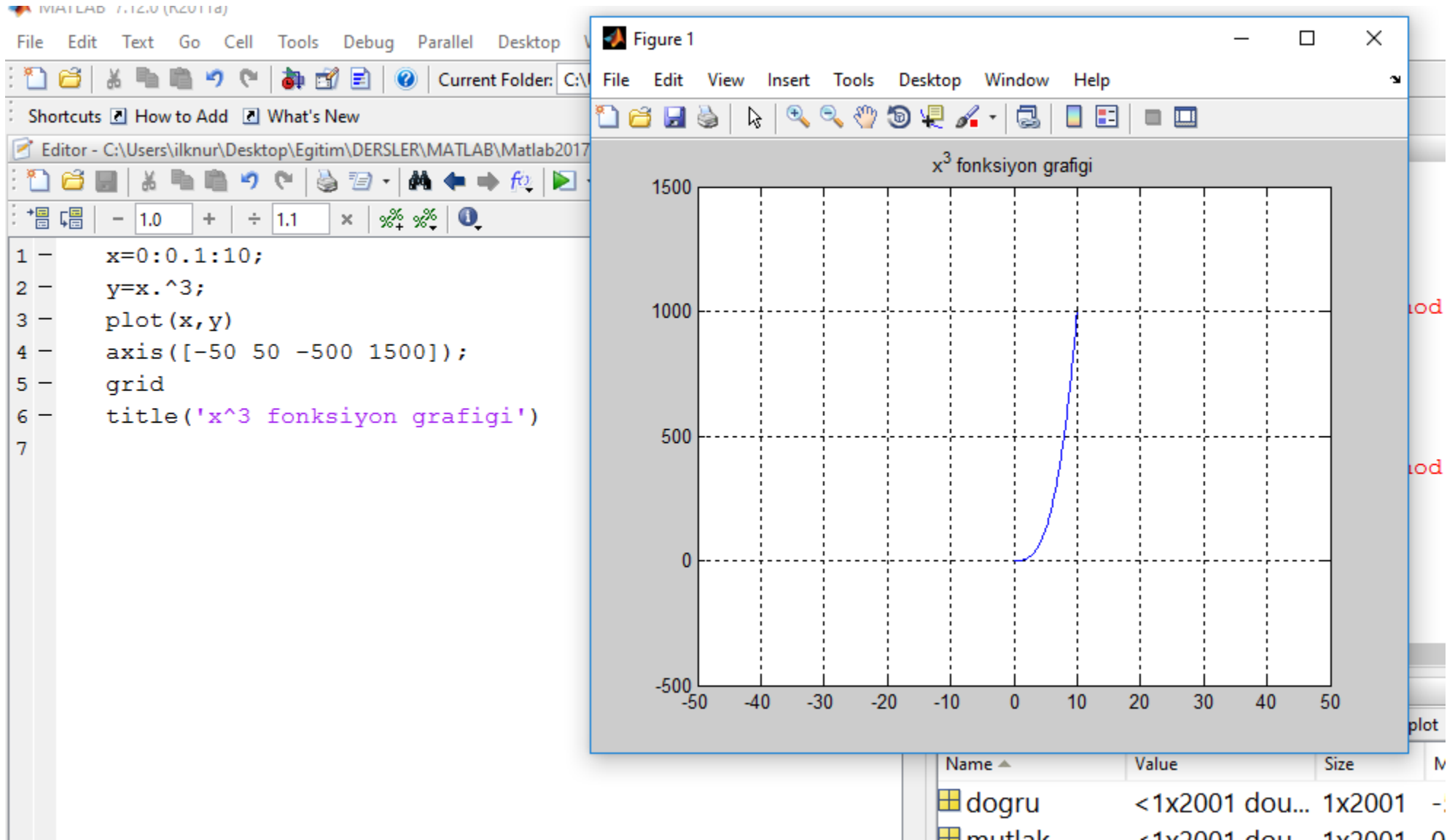
plot fonksiyonundan sonra kullanılmalıdır.

- **$y=x^3$ fonksiyonunu $0\leq x\leq 10$ aralığında 0,01 artımla x ekseninin minimum değeri 50 maksimum değeri 50 ve y ekseninin minimum değeri -500 maksimum değeri 1500 olacak şekilde çizdirelim.**

- $y=e^{-x}\sin(x)$ ve $y=e^x\cos(x)$ fonksiyonlarını $1 \leq x \leq 10$ aralığında 0,01 artımla y1 değerleri ve y2 değerleri farklı eksenlerde olacak şekilde çizdirin.



- $y=x^3$ fonksiyonunu $0 \leq x \leq 10$ aralığında 0,01 artımla x ekseninin minimum değeri -50 maksimum değeri 50 ve y ekseninin minimum değeri -500 maksimum değeri 1500 olacak şekilde çizdirelim.



MATLAB/Grafik-Grafik yoluyla bilgi üretme

Örnek: `ya=randn(1000,1)` ve `yb=randn(1000,1)*3` biçiminde iki ölçü grubu oluşturalım. (`randn` fonksiyonu, beklenen değeri 0, standart sapması 1 olan normal dağılmış sayı üretir). `ya`'nın standart sapması 1, `yb`'nin standart sapması ise 3'tür.

Bu ölçülerin, a ve b kaynaklarından elde edildiğini ve de standart sapmalarını bilmediğimizi düşünelim. Hangi ölçü grubunun daha kaliteli olduğunu (standart sapmasının düşük olduğunu) grafik üzerinden görebilmek için, bir önceki örnekteki hold on/hold off komutlarını kullanarak bunları çizdirmek yeterli olacaktır:

Bir önceki örnekte kullanılan `ya` ve `yb` ölçülerinin birbirleriyle nasıl bir ilişkide olduğunu görmek için,

```
plot(ya,yb, ' . ')
```

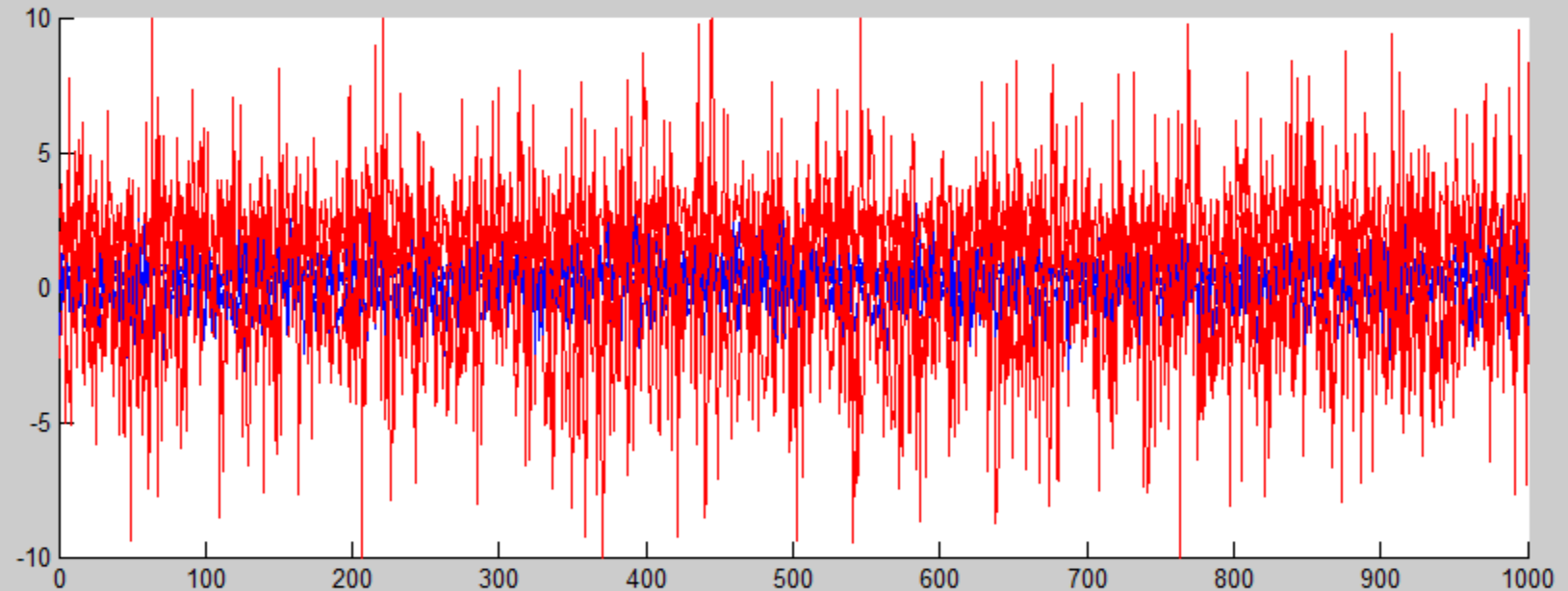
komutunu kullanmak yeterlidir.


```
1 - ya=randn(1000,1);  
2 - yb=randn(1000,1)*3;  
3 - hold on,  
4 - plot(ya),  
5 - plot(yb,'-r'),  
6 - hold off
```

Böylece, kırmızı ile gösterilmiş y_b ölçülerinin sıfırdan daha çok saptıkları, dolayısıyla standart sapmasının daha yüksek olduğu bilgisi grafik üzerinden kolaylıkla okunabilmektedir.

Figure 1

File Edit View Insert Tools Desktop Window Help



Untitled2.m x hafta5.m x Ur

Current Folder

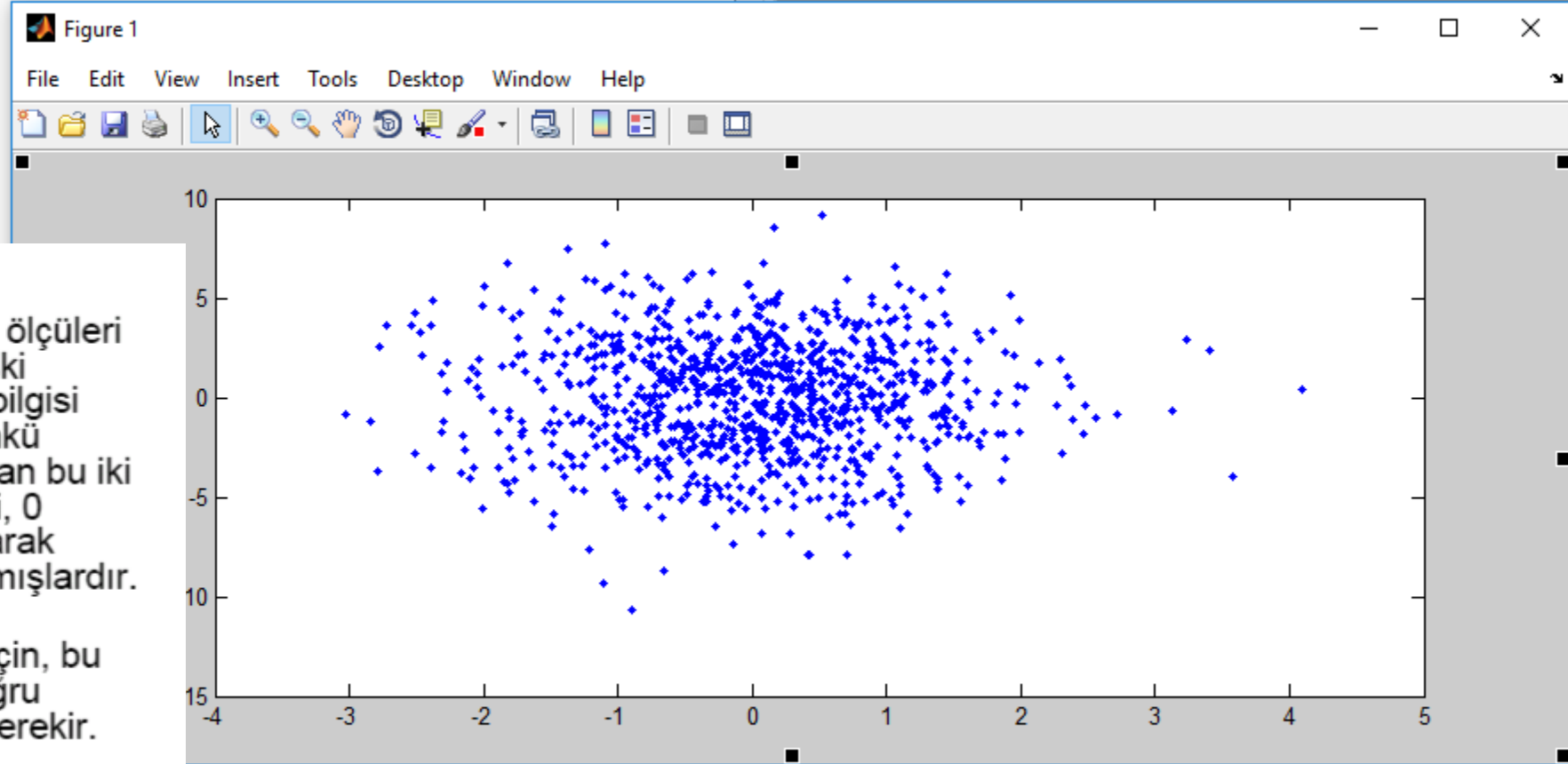
```
ezplot sin(x)/(1+x^2)  
clc
```

%-- 3/22/2017 8:37 AM --%

Start

script

```
1 - ya=randn(1000,1);  
2 - yb=randn(1000,1)*3;  
3 - plot(ya,yb, ' .')  
4
```



İlgili grafikten, ya ve yb ölçüleri arasında anlamlı bir ilişki (korelasyon) olmadığı bilgisi hemen türetilir. Çünkü beklenen değerleri 0 olan bu iki gruba ilişkin ölçü çiftleri, 0 merkezinde düzgün olarak (daire biçiminde) dağılmışlardır. İki ölçü grubu arasında korelasyon olabilmesi için, bu nokta bulutunun bir doğru etrafında gözlenmesi gerekir.

```
clc  
%-- 3/22/2017 8:37 AM --%
```

Start

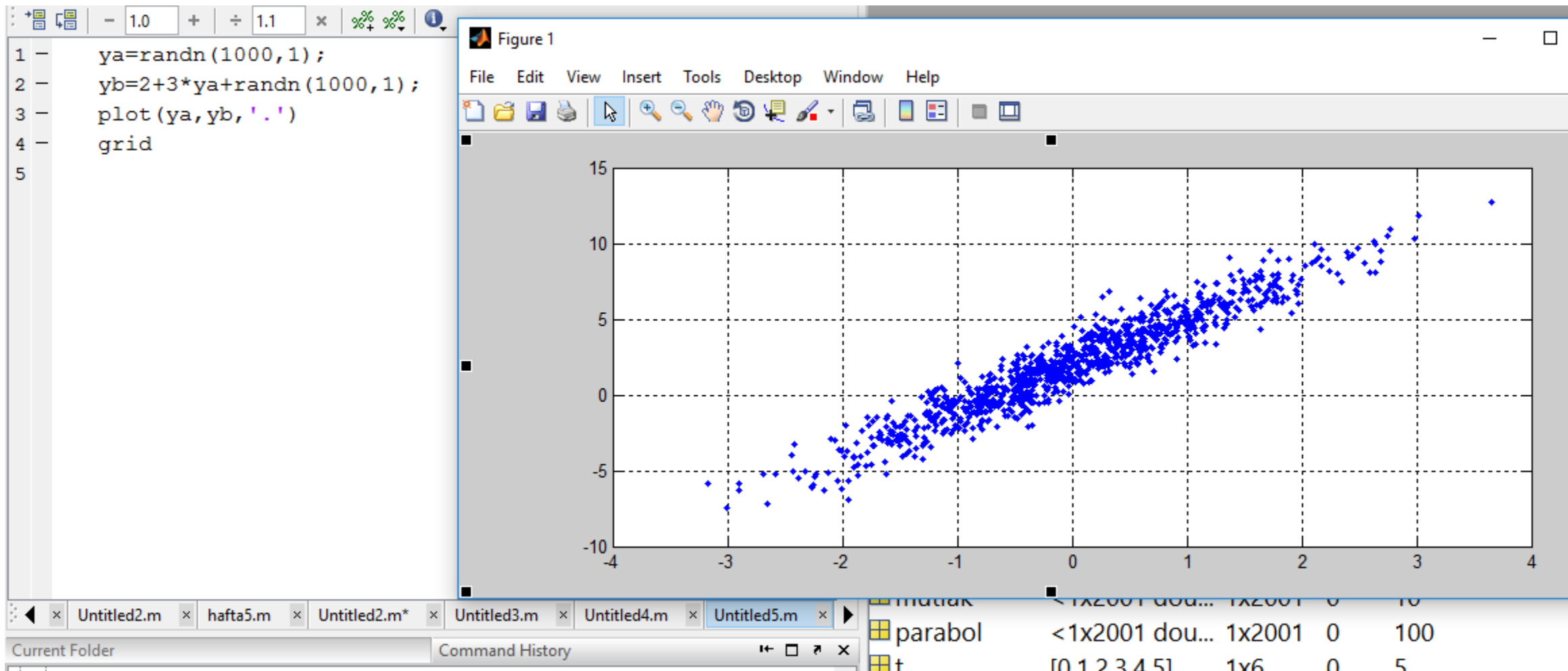


v [0,1,16,81,25... 1x6 0 625
1x101 class 1x101 0 10
script

Bir önceki örnekte kullanılan y_b ölçülerini, $y_b = 2 + 3 \cdot y_a + \text{randn}(1000, 1) \cdot 1$, biçiminde y_a ölçülerine bağlı olarak üretilim. Bu durumda,

```
plot(ya,yb, '.')
```

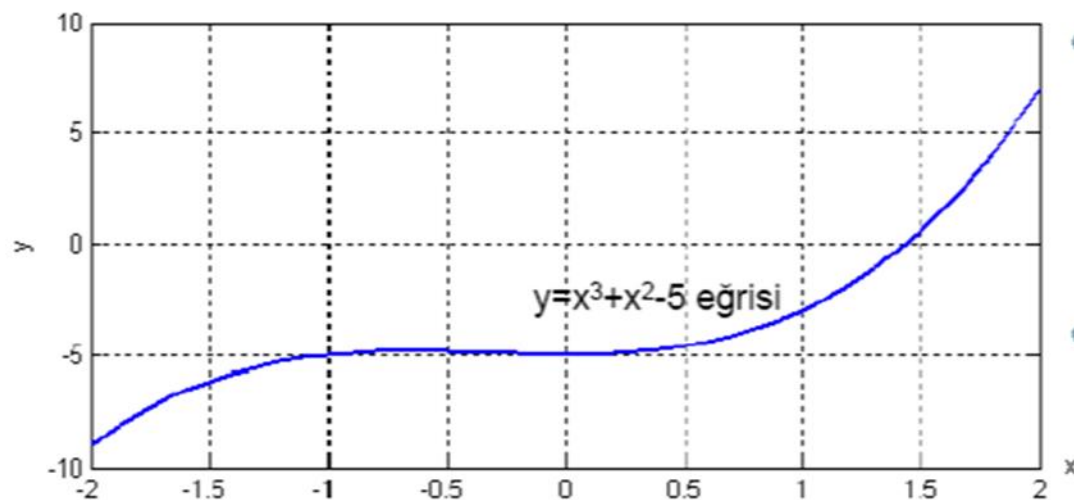
ile oluşturulan grafikten y_a ve y_b ölçüleri arasındaki korelasyonun varlığı hemen görülecektir.



- 1. Mühendislik uygulamalarında en çok karşılaşılan problemlerden biri de $F(x)=0$ biçimindeki bir denklemin ilgili aralıktaki kökünü (fonksiyonu sıfır yapan x değerini) bulmaktır. Sayısal analizde kullanılan Newton-Raphson gibi yöntemlerde kökün yaklaşık değerine ihtiyaç vardır. Bu yaklaşık değeri bulmak için **grafik çizimi** oldukça kullanışlı olmaktadır. Örneğin,
- $F(x)=x^3+x^2-5=0$ gibi bir denklemin -2 ile 2 arasındaki yaklaşık kökünü bulmak için, $x=-2:0.1:2$ biçiminde x değerleri ve $y=x.^3+x.^2-5$ ile de bu x 'lere karşılık y değerleri üretilir.

`plot(x,y),grid on`

komutlarıyla aşağıdaki grafik çizdirilir.(`grid on` komutu şekildeki grid ağını çizer)



- $y=0$ doğrusunun eğriyi kestiği noktadan, x eksenine hayali bir dik inilirse, bu dikin gösterdiği x değeri, $F(x)$ denklemini sağlayan kök olacaktır. Buradan kökün yaklaşık değerinin 1.4 olduğu sonucuna kolaylıkla ulaşılır.

Not: Figure penceresindeki büyütme özelliği ile, ilgili kesişim noktasına zoom yapılarak, yaklaşık kök daha hassas biçimde belirlenir.



- Bir mühendis, bir olayı gözler ve gözlem sonucunda elde ettiği ölçüler yoluyla olayı matematiksel eşitliklerle açıklamaya çalışır. Böylesi eşitliklere, kısaca “model” adı verilir.
- Figure penceresinde yer alan “Tools” menüsü içindeki “Basic Fitting” seçeneği grafik üzerindeki x ve bunlara karşılık gelen y değerlerini kullanarak, bunlara en iyi uyan $y=f(x)$ polinomunu tanımlar. Böylece oldukça pratik bir biçimde model oluşturulur.
- Burada hatırlatılması gereken iki nokta vardır:

(1) Eğer nokta çifti (x,y) sayısı uydurulan polinomun bilinmeyen sayısına eşitse, bulunan fonksiyon bir enterpolasyon polinomudur.

(Not: n. dereceden bir polinomun $n+1$ adet bilinmeyeni olduğunu hatırlayınız: Örneğin 4. dereceden bir polinom; $y=ax^4+bx^3+cx^2+dx+e$ dir ve bilinmeyen sayısı 5'dir)

(2) Eğer nokta çifti sayısı, uydurulan polinomun bilinmeyen sayısından fazlaysa en uygun polinom bir “*en küçük kareler*” kestirim yöntemi sonucudur. y değerleri hatalı büyüklüklerse (yani ölçü ise), basic fitting ile uydurulacak polinomun bilinmeyen sayısı her zaman ölçü sayısından küçük olmalıdır!

➤ Birçok deneylerle elde edilen noktalardan en iyi şekilde geçecek olan eğri denklemi bulmak istendiğinde bulunan eğri her noktadan geçmeyebilir, fakat verilen noktalara en iyi şekilde uyan bir eğri olmalıdır.



➤ Birçok deneylerle elde edilen noktalardan en iyi şekilde geçecek olan eğri denklemi bulmak istendiğinde bulunan eğri her noktadan geçmeyebilir, fakat verilen noktalara en iyi şekilde uyan bir eğri olmalıdır.

➤ En iyi uyan eğriyi bulurken eğrinin her noktaya olan uzaklığı yani hata göz önüne alınır ve bu hatanın yok edilmesine çalışılır. MATLAB'ı kullanarak verilen noktalardan geçen eğrinin denklemi bulunurken genelde bu eğri polinom olarak tanımlanır ve bu polinomun katsayıları bulunur.

- n dereceli bir polinomun n+1 tane katsayısı vardır ve elde edilen n+1 tane veri noktası varsa bu noktalardan n. dereceden bir polinom geçebilir

MATLAB’de bunun için polyfit komutunu kullanılır

Genel yazım şekli :

p=polyfit(x,y,n)

Polinomlar

$$P(x)=a_nx^n + a_{n-1}x^{n-1} + \dots + a_1x + a_0$$

x’in en büyük üssü n polinomun derecesini beliler.

ai(i=1,2,3,...,n) polinomun katsayılarıdır.

➤ Bu fonksiyon (x,y) noktalarından geçen ve n dereceli olan en uygun eğrinin katsayılarını vermektedir. Burada p vektörü alçalan derece sıralamasına göre katsayıları vermektedir.

➤ Polinomun katsayıları bulunduktan sonra 'polyval' fonksiyonu kullanılarak bulunan polinomun değerlerini ve eldeki verilerin karşılaştırarak (genelde aynı eksen üzerinde çizdirerek) polinomun uygunluğunu tespit edilir.

➤Eğer polinom verilen noktalara uygun değilse polinomun derecesi artırılarak yeni katsayılar elde edilir.

Örnek 7 : x ve y değişkenleri arasındaki bağıntı aşağıdaki gibidir.

x=0 1 2 3 4 5 6

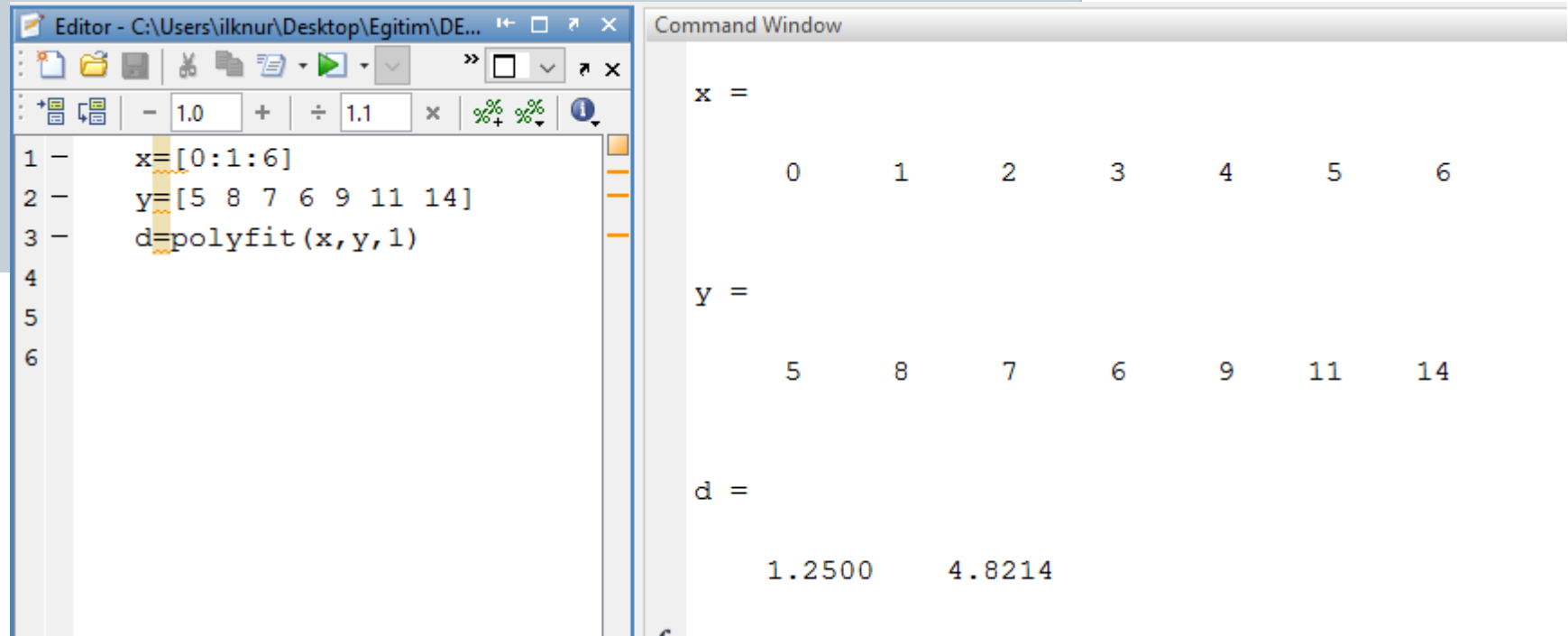
y=5 8 7 6 9 11 14

Önce veri noktalarına 1. Derece bir polinom çizilirse;

x=[0 1 2 3 4 5 6];

y=[5 8 7 6 9 11 14];

d=polyfit(x,y,1)



The screenshot shows a MATLAB Editor window with the following code:

```
1 x=[0:1:6]
2 y=[5 8 7 6 9 11 14]
3 d=polyfit(x,y,1)
4
5
6
```

The Command Window displays the results of the code execution:

```
x =
    0    1    2    3    4    5    6

y =
    5    8    7    6    9   11   14

d =
    1.2500    4.8214
```

Polinom denklemi:

$$p(x)=1.25x+4.8214$$

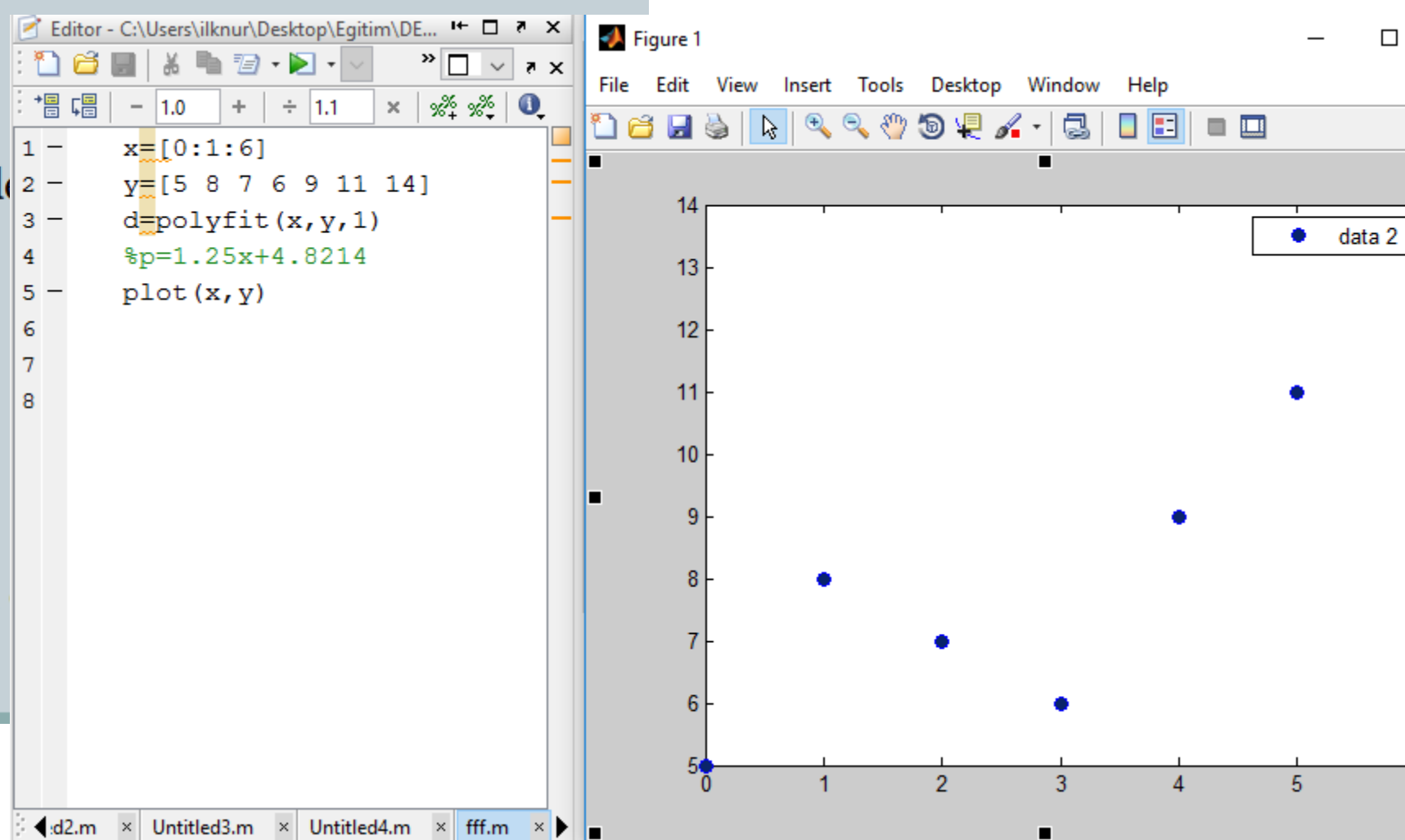
Bu polinomun 0 ile 6 arasındaki de

$z=0:6;$

$s=\text{polyval}(d,z)$

$\text{plot}(x,y, 'o', z,s, ':')$

Şekilden de görüldüğü gibi çizilen
hata fazla olmaktadır.



Polinom denklemi:

$$p(x)=1.25x+4.8214$$

Bu polinomun 0 ile 6 arası

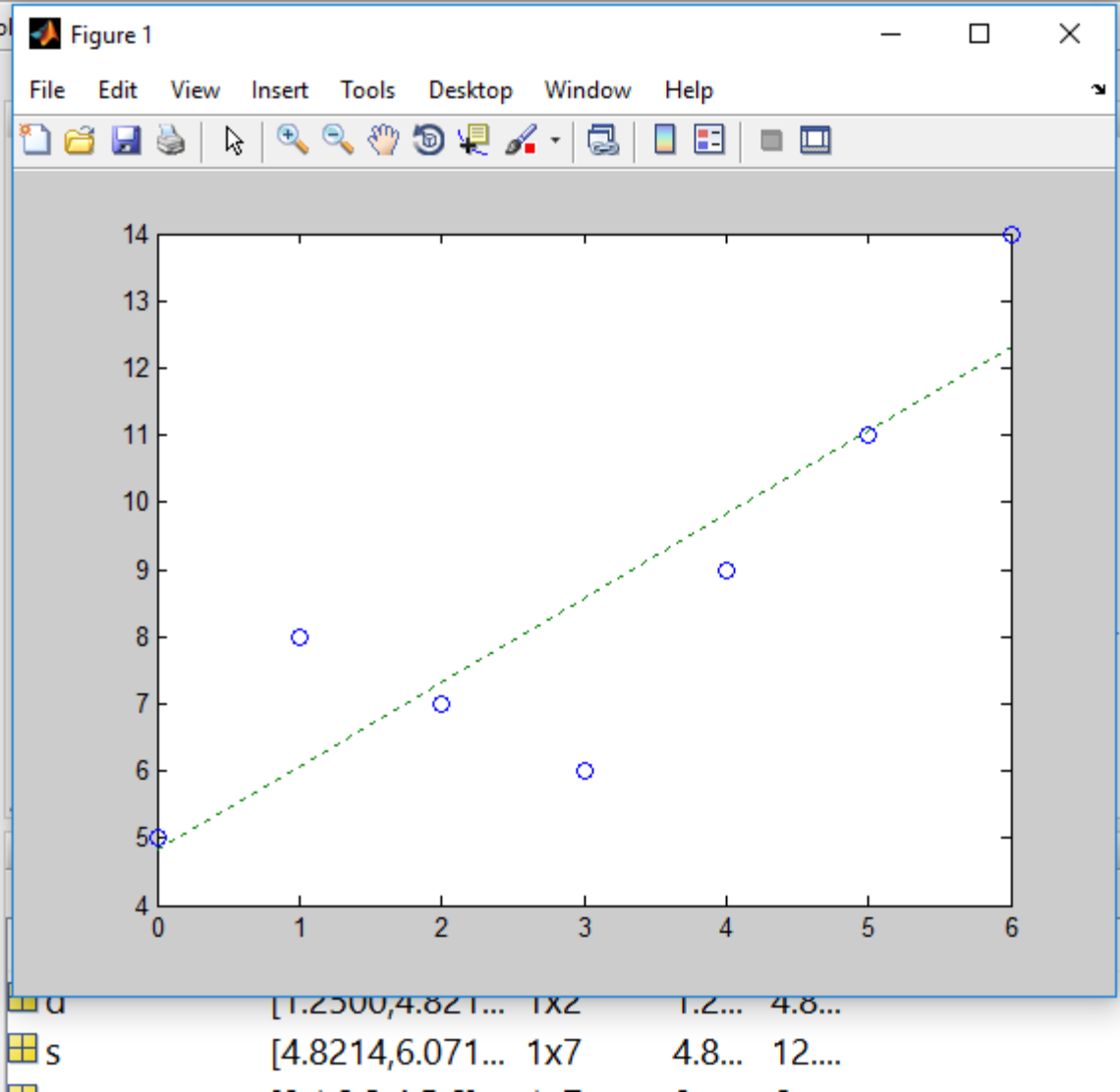
$z=0:6$;

$s=\text{polyval}(d,z)$

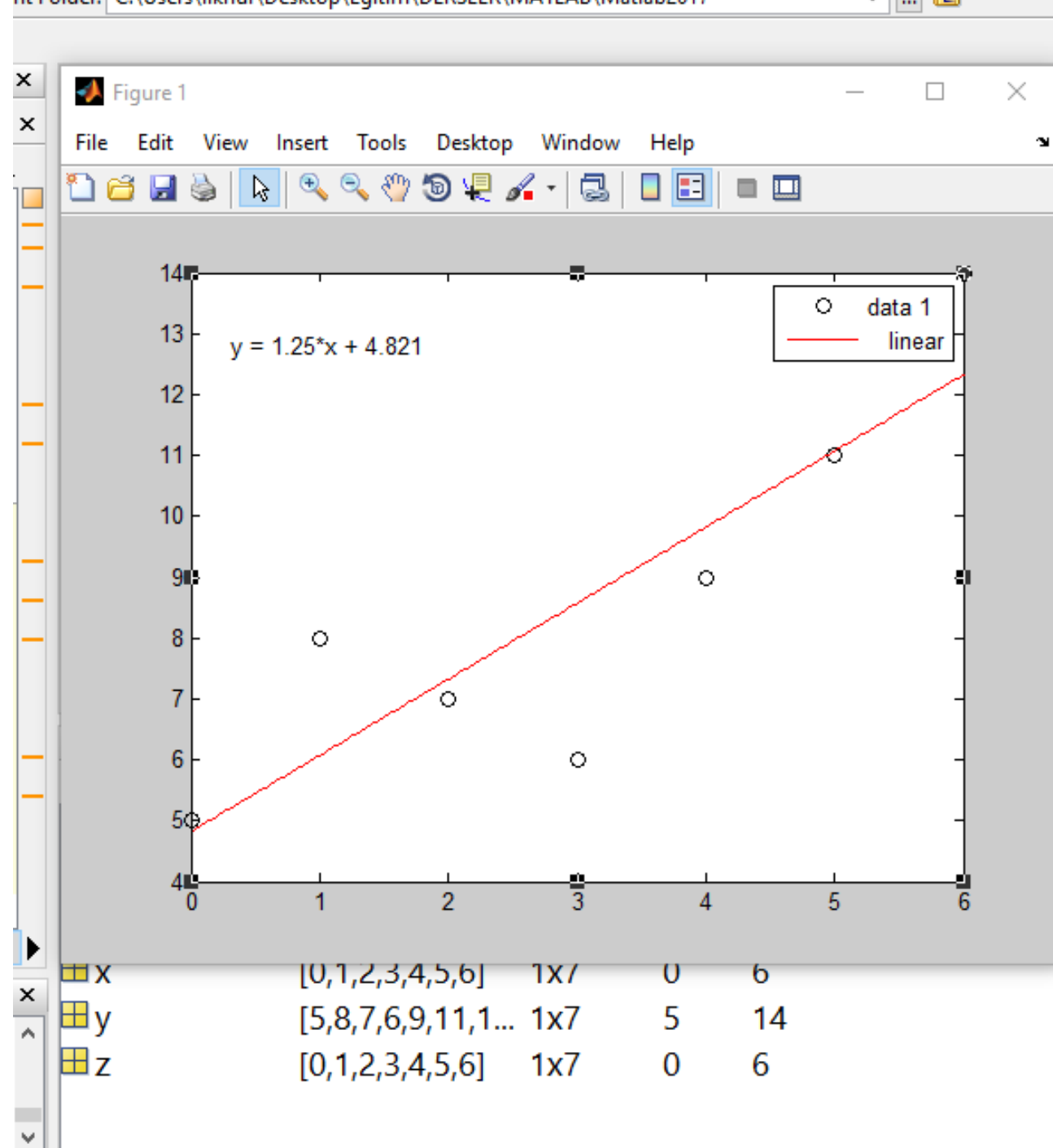
$\text{plot}(x,y, 'o', z,s, ':')$

Şekilden de görüldüğü gibi
hata fazla olmaktadır.

```
Shortcuts How to Add What's New
Editor - C:\Users\ilknur\Desktop\Egitim\DE...
1 - x=[0:1:6]
2 - y=[5 8 7 6 9 11 14]
3 - d=polyfit(x,y,1)
4 -
5 - %d(x)=1.25(x)+4.8214
6 - z=0:6
7 - s=polyval(d,z)
8 - plot(x,y,'o',z,s,':')
9 -
10 -
11 -
```



Aynı degerler için 2, 3, 4... dereceden polinomlari degerlendiriniz



Basic Fitti...

Select data: data 1

☐ Center and scale x data

Plot fits

Check to display fits on figure

- ☐ spline interpolant
- ☐ shape-preserving interpolant
- ☒ linear
- ☐ quadratic
- ☐ cubic
- ☐ 4th degree polynomial
- ☐ 5th degree polynomial
- ☐ 6th degree polynomial
- ☐ 7th degree polynomial
- ☐ 8th degree polynomial
- ☐ 9th degree polynomial

☒ Show equations

Significant digits: 4

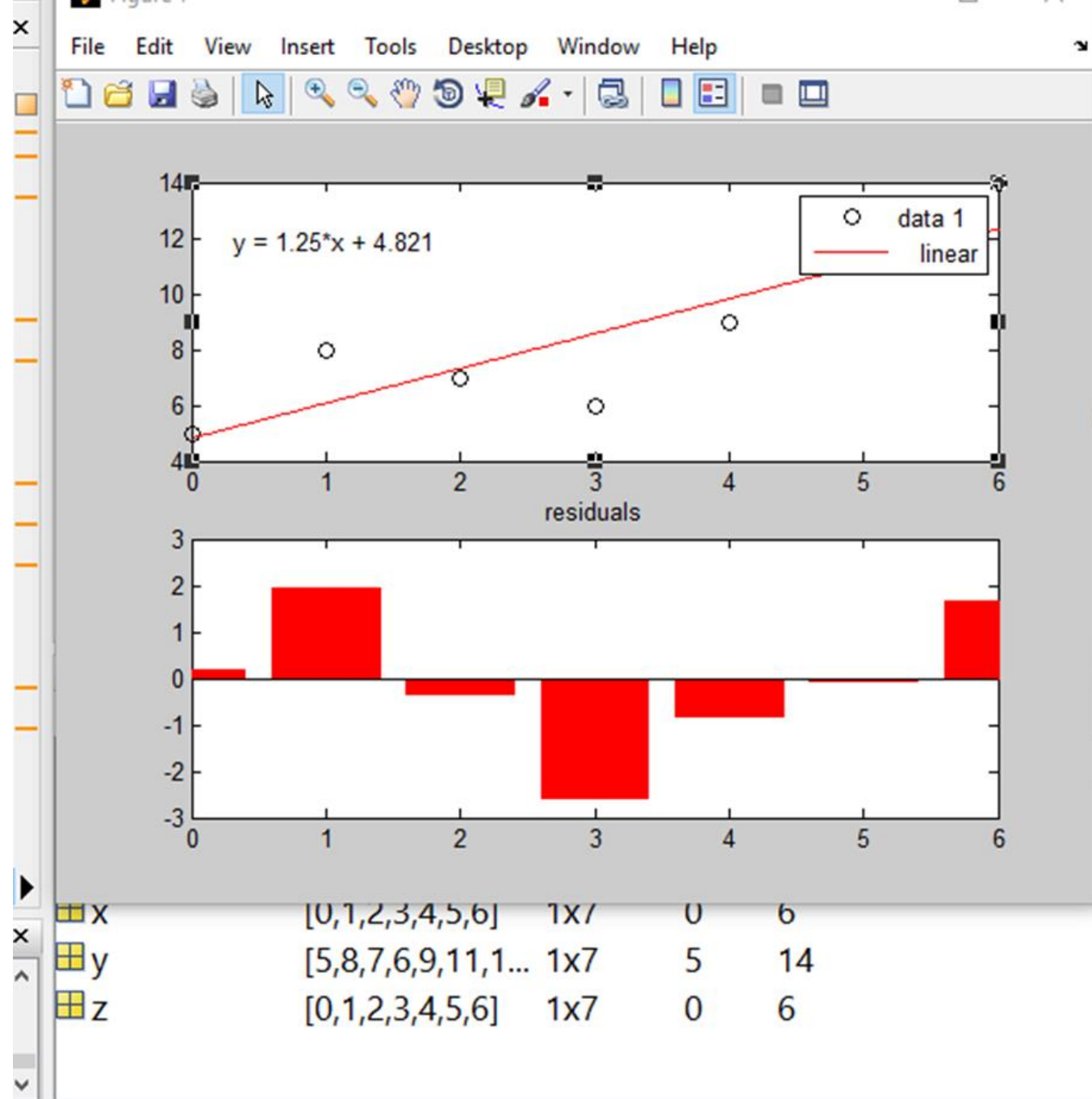
☐ Plot residuals

Bar plot

Subplot

☐ Show norm of residuals

Help Close →



Basic Fitti...

Select data: data 1

☐ Center and scale x data

Plot fits

Check to display fits on figure

- ☐ spline interpolant
- ☐ shape-preserving interpolant
- ☒ linear
- ☐ quadratic
- ☐ cubic
- ☐ 4th degree polynomial
- ☐ 5th degree polynomial
- ☐ 6th degree polynomial
- ☐ 7th degree polynomial
- ☐ 8th degree polynomial
- ☐ 9th degree polynomial

☒ Show equations

Significant digits: 4

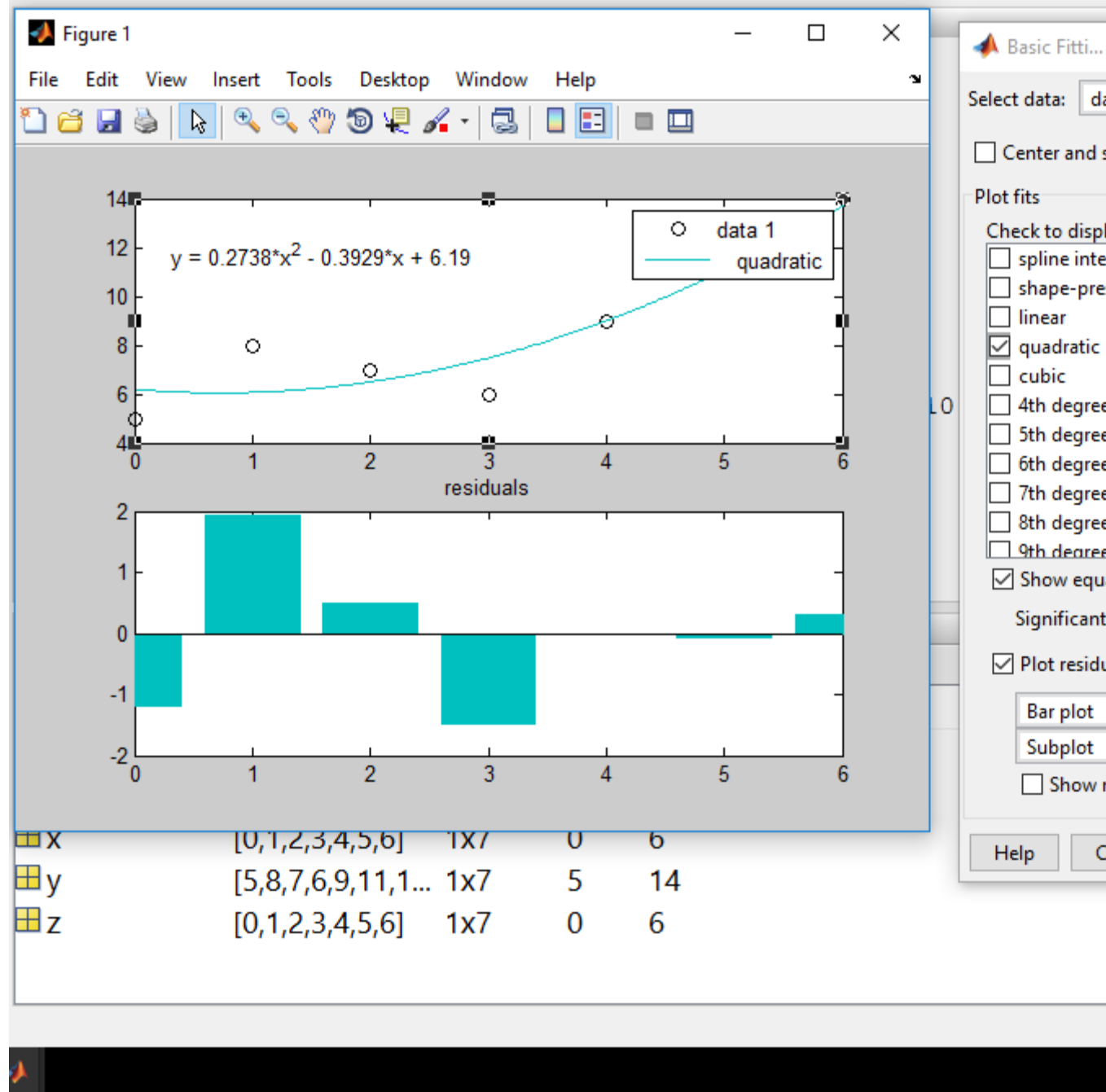
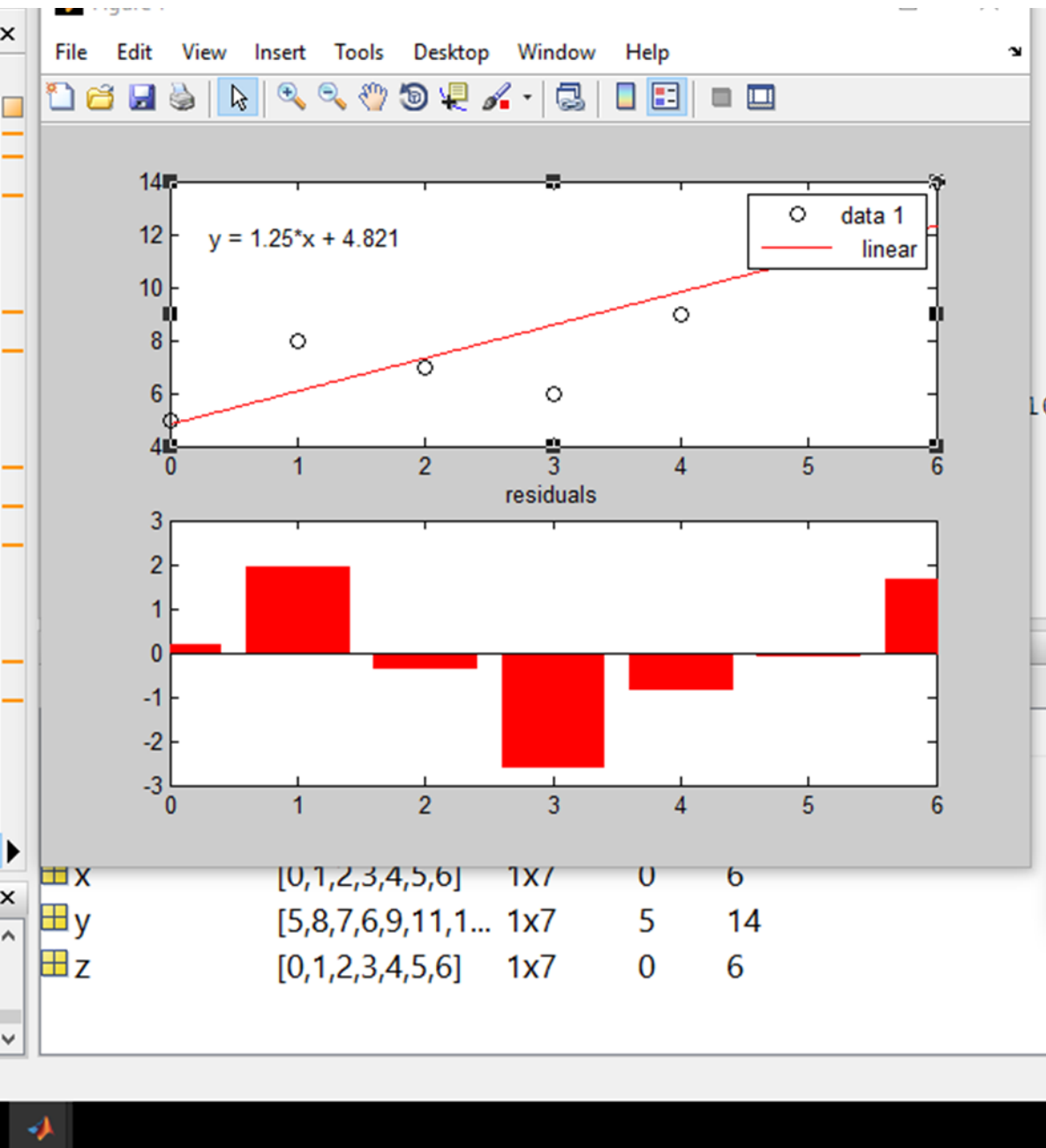
☒ Plot residuals

Bar plot

Subplot

☐ Show norm of residuals

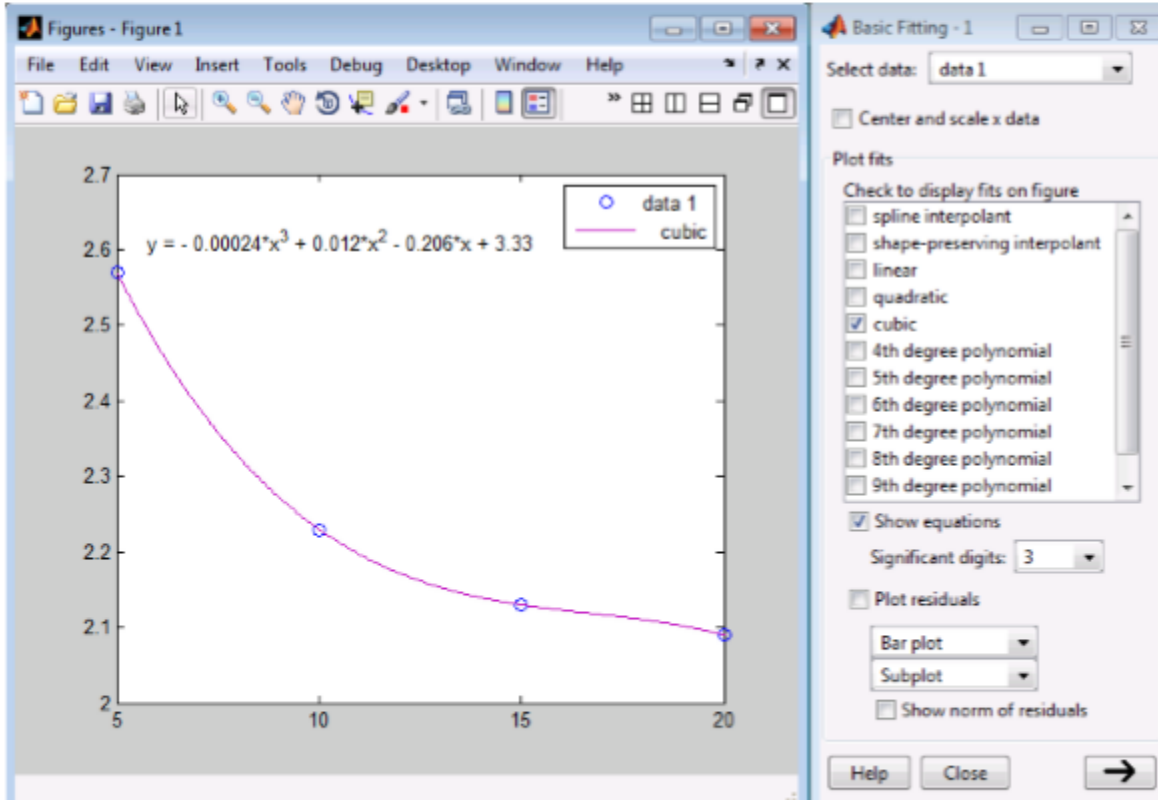
Help Close →



Örnek: Aşağıdaki tabloda $f= 5, 10, 15$ ve 20 değerlerine karşılık t -dağılımının $\alpha=5\%$ güven sınırları (t , değerleri) verilmektedir. $t=af^3+bf^2+cf+d$ polinomunu "basic fitting" özelliğini kullanarak belirleyiniz. $f=9$ için $t=2.26$ olduğuna göre elde edilen enterpolasyon polinomun doğruluğunu test ediniz.

f	5	10	15	20
t	2.57	2.23	2.13	2.09

Çözüm: $x=[5;10;15;20]$ ve $y=[2.57;2.23;2.13;2.09]$ olsun. `plot(x,y)` ile ilgili eğri çizilir.



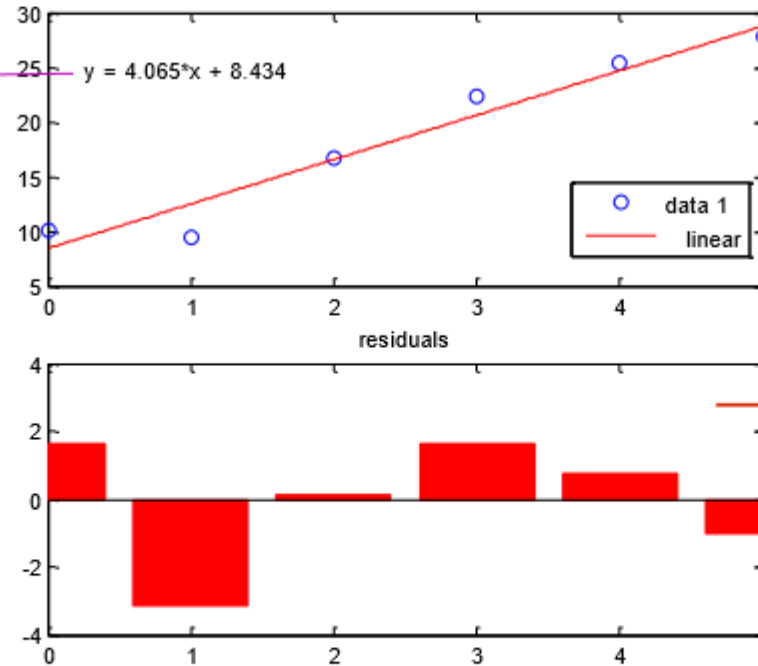
- Figure penceresindeki "Tools" menüsünden, "Basic Fitting" seçeneği seçilir.
- Açılan, "Basic Fitting" penceresinden ilgili polinom (burada, cubic, yani 3.derece) ve ardından, "show equations" seçeneği işaretlenir.
- Şekil üzerinde gösterilen $f(x)$ eşitliği, bize enterpolasyon polinomunu vermektedir.
- Bu denklemde, $x=9$ girilirse, $y=2.24$ değeri elde edilir. $f=9$ için $t=2.26$ olduğu bilindiğine göre, enterpolasyon polinomumuzun doğruluğu-*bu aralık için*-%2'dir.

Örnek: Aşağıdaki tabloda, x zamanlarına karşılık y ölçüleri elde edilmiştir. Ölçülere en iyi uyan $y=a+bx$ doğrusunu belirleyiniz (En küçük kareler kestirim yöntemi)

x	0	1	2	3	4	5
y	10.06	9.36	16.69	22.28	25.44	27.75

Çözüm: Tablodaki değerler x ve y vektörlerine atanır. `plot(x,y, 'o')` ile ilgili eğri çizilir. Basic Fitting penceresinde, "linear", "show equation", "plot residuals" seçenekleri işaretlendiğinde, aşağıdaki grafik oluşturulur.

En küçük kareler
yöntemine göre
belirlenen
en uygun doğru
denklemleri (*model*)



Ölçülerin, belirlenen
doğru denkleminde
sapmasını (düzeltmele
residuals) gösterir.

➤ INTERPOLASYON

- Belirli bir aralıkta değişim gösteren bir fonksiyonun bazı değerleri biliniyorken, bilinmeyen diğer değerleri bulmak bu yöntem kullanılır.
- MATLAB'da bunun için `interp(1)` ve `interp(2)` fonksiyonları kullanılır.
- `interp(1)` fonksiyonu veri analizi ve eğri uydurma işlemlerinde tek boyutlu ara değer hesabı yapar.
- `interp2()` fonksiyonu iki boyutlu ara değer bulma işlemini gerçekleştirir. Bu işlem görüntü işleme ve veri görüntülenmesinde kullanılır

GENEL YAZIM SEKLI

`z=interp1(x,y, [0.5 1.6], 'interpoasyon turu')`

- **parantez içine önce değişken aralığı sonra fonksiyon değeri ve fonksiyon değeri bulunmak istenen sayı ve daha sonra interpolasyon türü girilir.**
- **Tür girilmez ise lineer olarak kabul edilir. Spline ve cubic türleri linear ve nearest türüne göre daha duyarlı olduğundan çoğu uygulamada tercih edilir.**

Lineer Interpolasyon

- Bu metod istenilen x değerinin solundaki ve sağındaki veri noktalarını alıp, bu noktalardan bir doğru geçirir. Bu doğrunun istenilen x değerine karşılık gelen y değerini verir. Linear interpolasyon metodu sadece birinci derece bir polinoma uyan veri noktaları için en uygun sonucu vermektedir.

Cubic Interpolasyon

- Bu metod her iki veri noktasından geçen 3. Derece bir polinom bulur ve bu polinomu kullanarak istenilen noktadaki y değerini hesaplar.

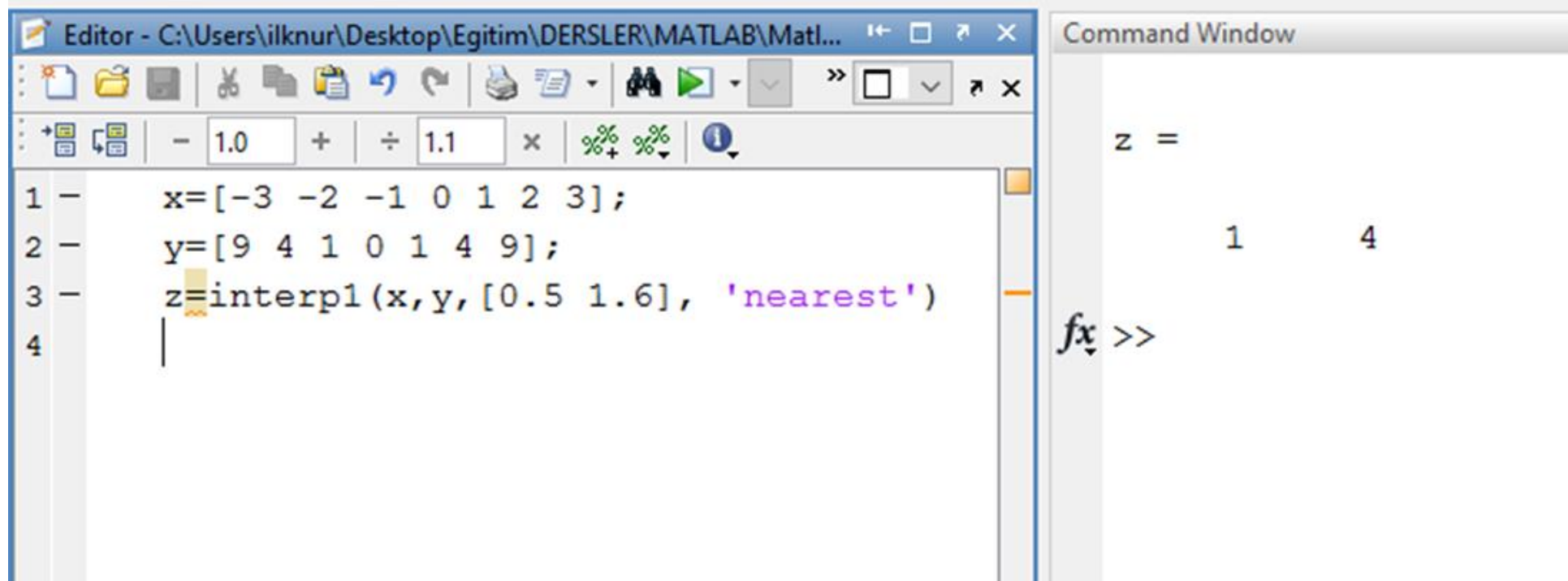
Nearest Interpolasyon

- Bu metod istenilen noktaya en yakın olan veri noktasını verir.

Nearest, cubic, ve lineer interpolasyonu kullanarak $x=0.5$ ve $x=1.6$ olduğu anda y değerleri ne olur?

$x=[-3 \ -2 \ -1 \ 0 \ 1 \ 2 \ 3]$

$y=[9 \ 4 \ 1 \ 0 \ 1 \ 4 \ 9]$



The image shows a MATLAB environment with an Editor window and a Command Window. The Editor window contains the following code:

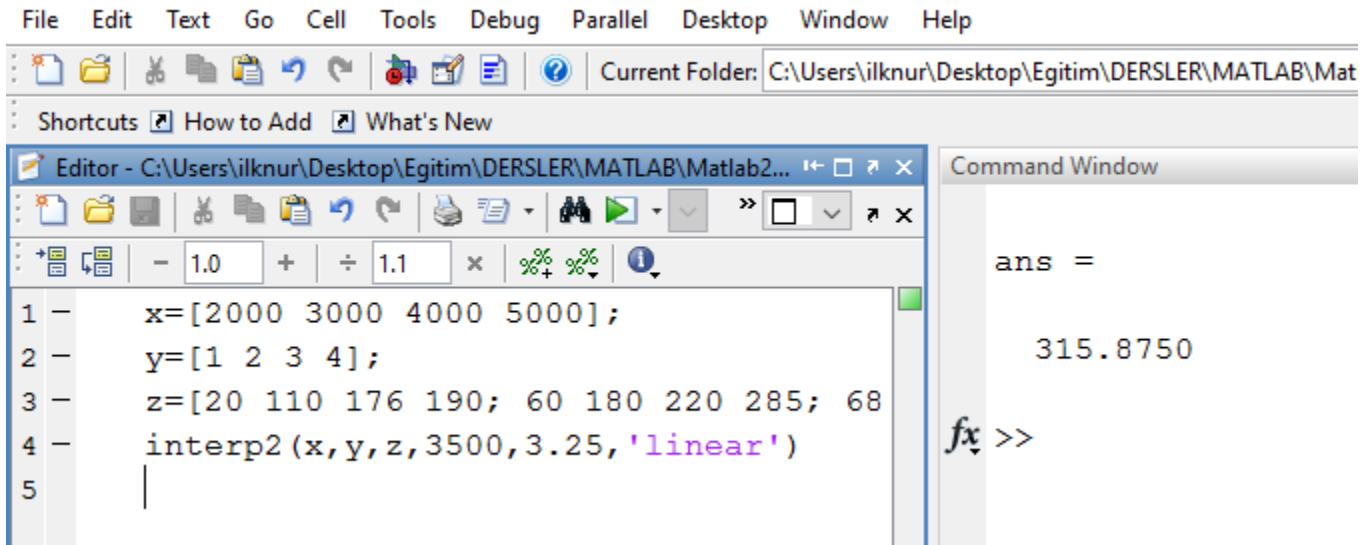
```
1 - x=[-3 -2 -1 0 1 2 3];  
2 - y=[9 4 1 0 1 4 9];  
3 - z=interp1(x,y,[0.5 1.6], 'nearest')  
4 - |
```

The Command Window shows the output of the code:

```
z =  
  
    1    4  
  
fx >>
```

Asağıda verilen tabloda bir motorun belli devirlerinde ve belli zamanlarda silindir başı sıcaklık değerleri gösterilmiştir. Buna göre 3.25 saniyede 3500 d/d karşılık gelen sıcaklık değerini hesaplayın.

Motor Hızıd/d	2000	3000	4000	5000
Zaman(s)	Sıcaklık(Derece)			
0	0	0	0	0
1	20	110	176	190
2	60	180	220	285
3	68	240	349	380
4	77	310	450	510



Kutle akış debisi m (g/s) ile sıcaklık T ($^{\circ}\text{C}$) arasında aşağıdaki ölçümler yapılmıştır. Kutlesel debi m 'in sıcaklığın karekökü ile değiştiğine inanılmaktadır. Buna göre kutlesel akış hızı ve sıcaklık arasındaki bağıntıyı belirleyerek 53°C deki akış hızını bulunuz.

T ($^{\circ}\text{C}$)	10	20	40	80
m (g/s)	14.71	20.14	27.73	38.47

Akış hızı ve akış hızının debimetreden okunarak kalibre edilebilmesi için aşağıdaki ölçümler alınmıştır. V-R kalibrasyon eğrisini çizin. Akış hızı 36 olarak okundığında gerçek debi nedir?

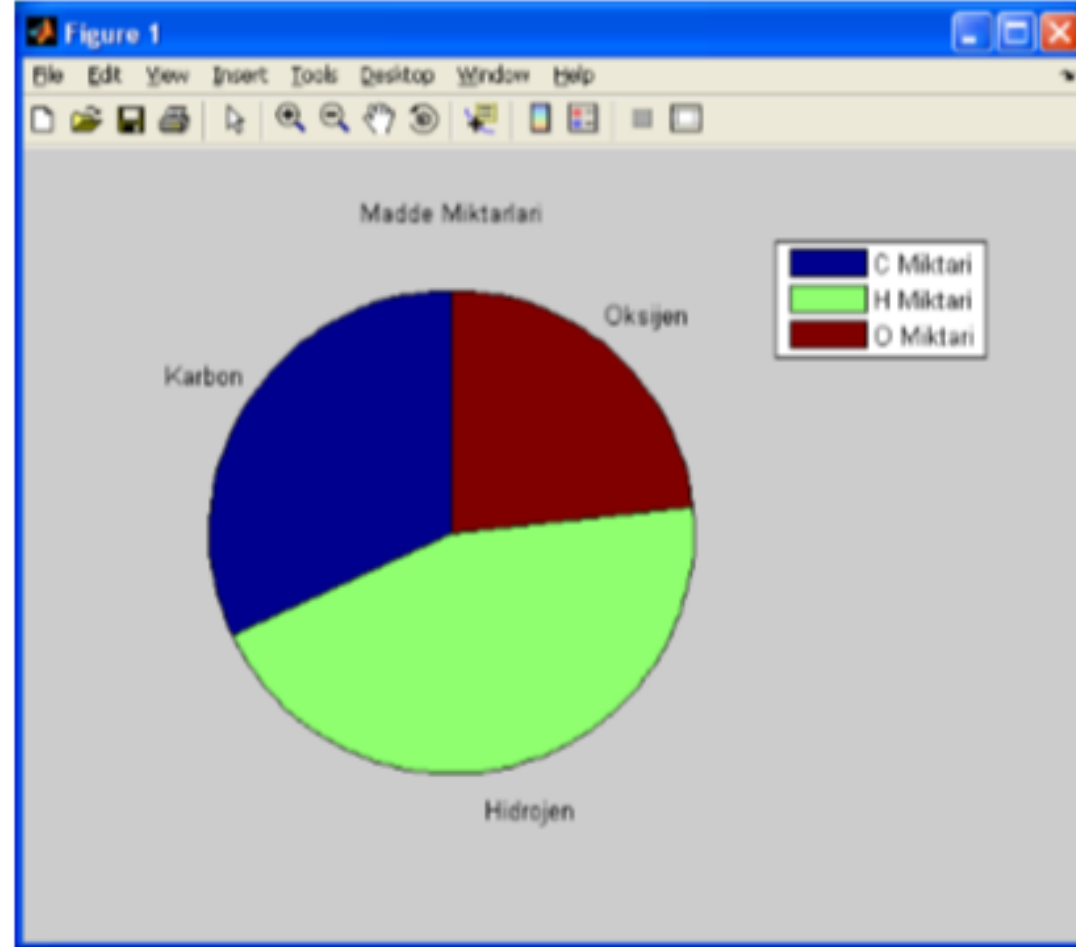
Akış hızı (V)	20	52.1	84.1	118.3	153
Debimetrede okunan değer (R)	10	20	50	70	90

Özel Grafikler

Matlab birçok 2D veya 3D özelleştirilmiş grafik formatını destekler. Bunlardan en çok kullanılan pasta dilimi ve bar formatlarını görelim.

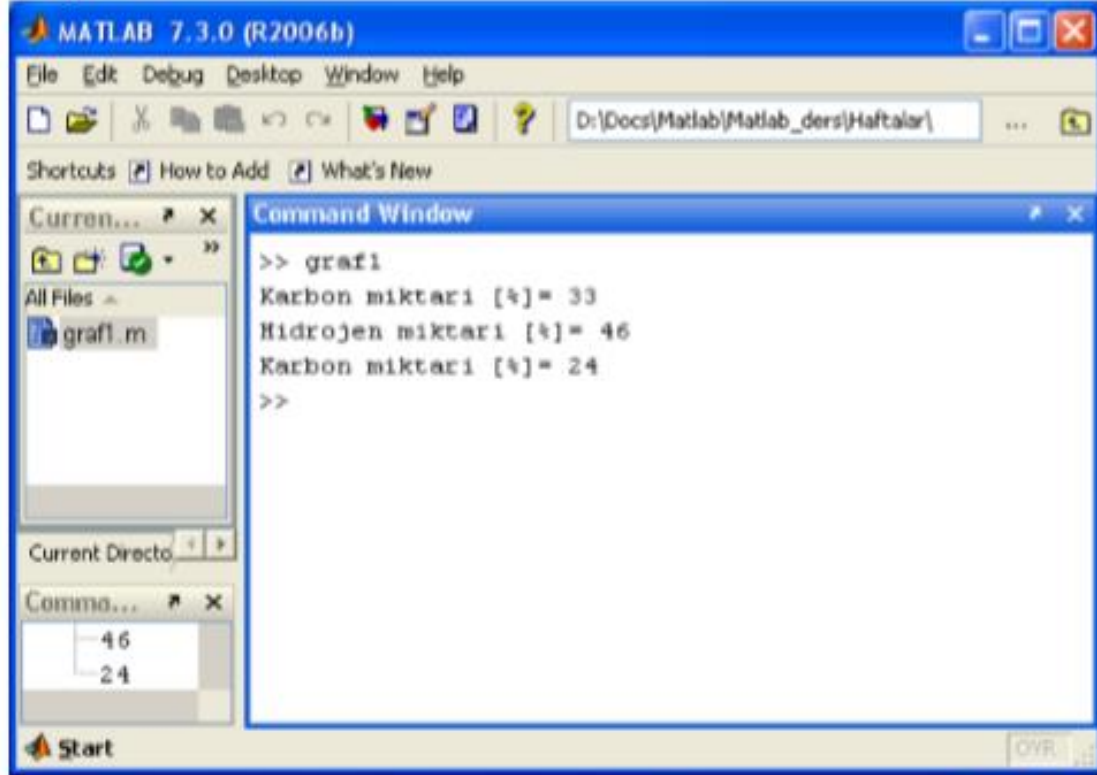
1. Pasta Dilimi Grafikleri (Pie)

Pasta dilimi grafikleri parçaların bir bütün içerisinde ne kadar yer kapladıklarını gösteren grafiklerdir.

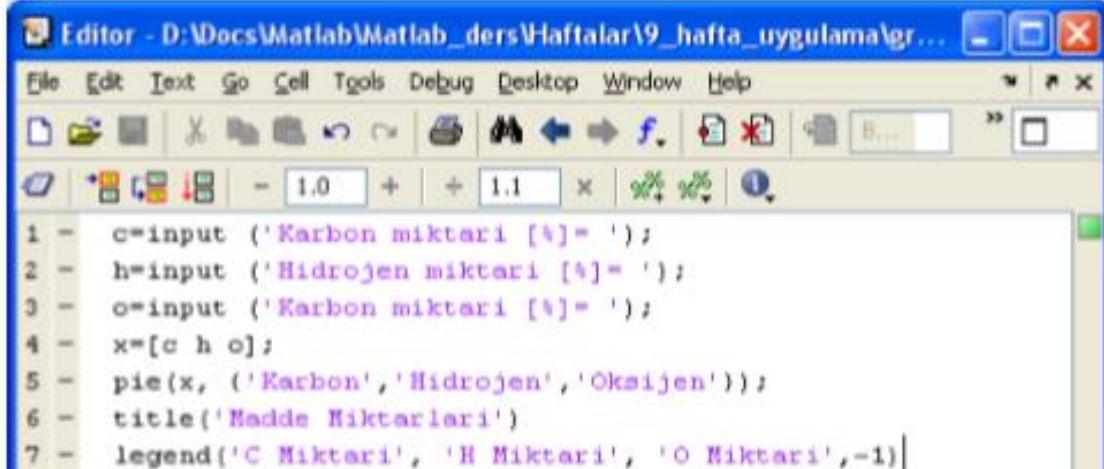


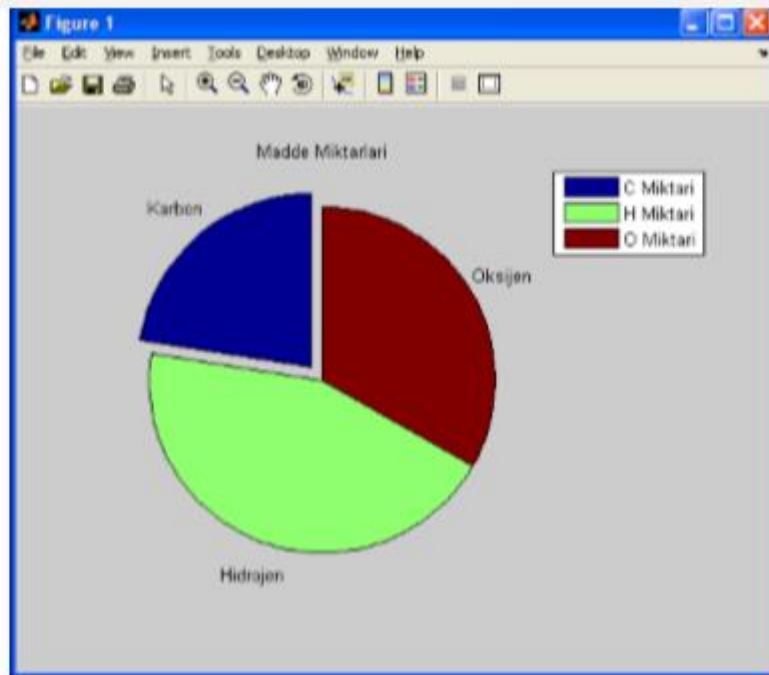
Genel yazım şekli;

Örnek: Karbon, hidrojen ve oksijenden oluşan bir kimyasal maddenin analizini, kullanıcıdan bu maddelere ait yüzde miktarlarını isteyerek pasta dilimi şeklinde gösteren grafiği çizen Matlab kodunu yazınız.



Pasta dilimi formatındaki grafiklerde istenen dilimlerin bütünden ayrılması için explode (patlatma) işlemi uygulanır. pie fonksiyonunda x veri vektöründen sonra sonra [1 1 0 0 0] yazılırsa 5 dilim içerisinde 1. ve 2. dilimler bütünden ayrı olarak gösterilir. Az önceki örnekte 1. dilimi bütünden ayrı olarak gösterelim.





Editor - D:\Docs\Matlab\Matlab_ders\Haftalar\9_hafta_uygulama\gr...

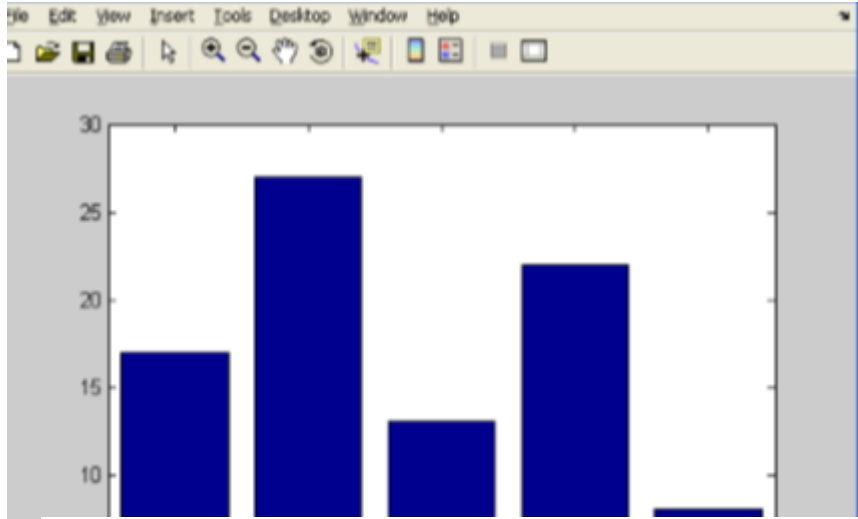
```
1 - c=input ('Karbon miktarı [%]= ');
2 - h=input ('Hidrojen miktarı [%]= ');
3 - o=input ('Karbon miktarı [%]= ');
4 - x=[c h o];
5 - pie(x, [1 0 0], {'Karbon', 'Hidrojen', 'Oksijen'});
6 - title('Madde Miktarlari')
7 - legend('C Miktari', 'H Miktari', 'O Miktari', -1)
```

script Ln 7 Col 49 QVR

Matlab'de farklı gösterimlere göre çizim yapmak mümkündür. Bunlardan ikisi bar ve stem(...) çizim fonksiyonlarıdır.

Çubuk Grafikler (Bar)

Çubuk grafikler verilerin birbirleri ile karşılaştırılması için en iyi seçenektir. Matlab'de dikey çubuk grafikler çizilebilir.



Örnek: 2006 yılında Türkiye deki sektörlerin toplam ciroları aşağıdaki gibi olsun buna göre her bir sektörün toplam içindeki yerini gösteren grafiği çiziniz.

Ticaret	Sanayi	Ulaşım	Turizm	Tarım
17milyon YTL	27milyon YTL	13milyon YTL	22milyon YTL	8milyon YTL

Genel yazım şekli

bar (m:n,x,çubuk kalınlığı)

m : ilk çubuk sayısı n : son çubuk sayısı x : veri vektörü

bar (1:5,x,0.2)

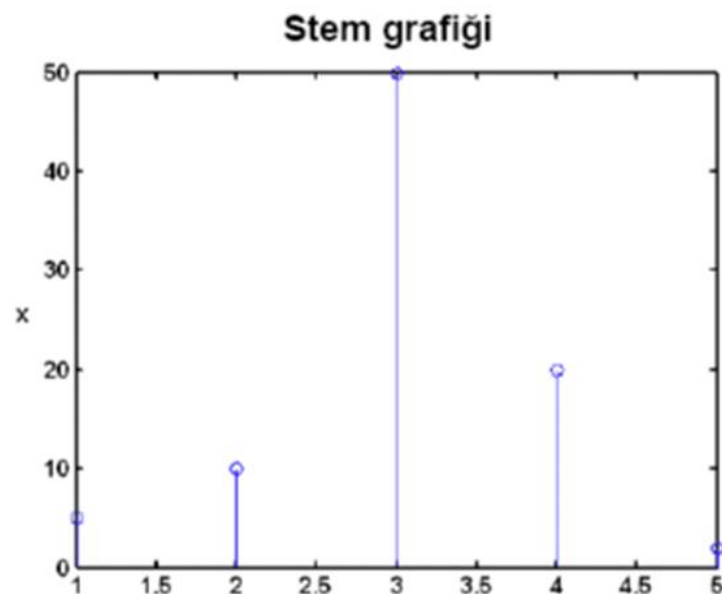
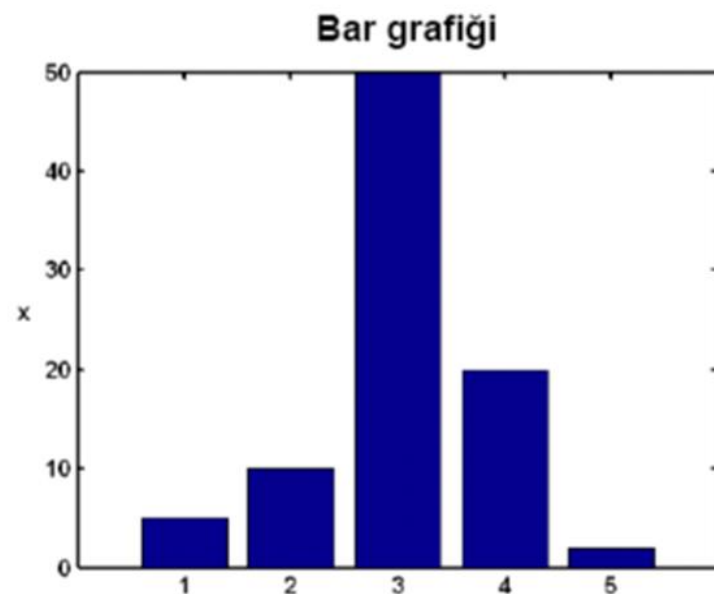
1'den 5'e kadar olan 5 adet çubuğun 0,2 kalınlıkla çizileceği gösterir.

Çubuk sayısı ile veri vektörünün uzunluğu eşit olmalıdır.

Kalınlık değeri yazılmazsa örneğin bar(1:5,x) gibi çubuk genişliğinin default değeri 0,8 alınır

```
Editor - D:\Docs\Matlab\Matlab_ders\Haftalar\9_hafta_uygula...
File Edit Text Go Tools Desktop Window Help
x
1 x=[17 27 13 22 8];
2 bar(1:5,x)
```

- **Örnek:** $x=[5;10;100;20;2]$ vektör elemanlarının bar ve stem grafik olarak göstermek isteyelim. `bar(x)` ve `stem(x)` aşağıdaki grafikleri çizdirecektir.



Örnek: 2006 yılında Türkiye'deki sektörlerin toplam ciroları aşağıdaki gibi olsun buna göre her bir sektörün toplam içindeki yerini gösteren grafiği çizin.

Ticaret	Sanayi	Ulaşım	Turizm	Tarım
17milyon YTL	27milyon YTL	13milyon YTL	22milyon YTL	8milyon YTL

3D (3 Boyutlu) Grafikler

3D Çizgi Grafikleri

3D çizgi grafikleri plot3 fonksiyonu ile oluşturulur.

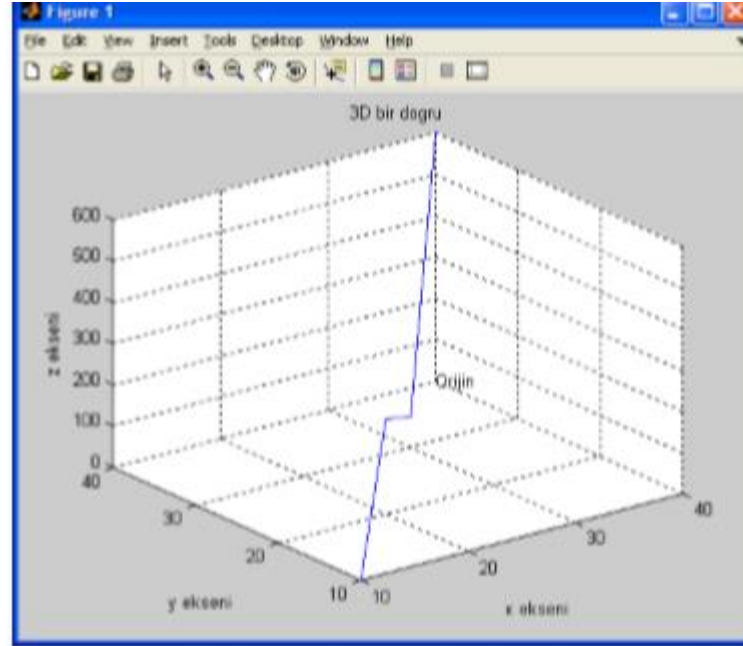
Aynı boyutlu x, y, z olarak adlandırılan

3 vektörün, karşılıklı gelen noktalarının birleştirilmesiyle çizilir.

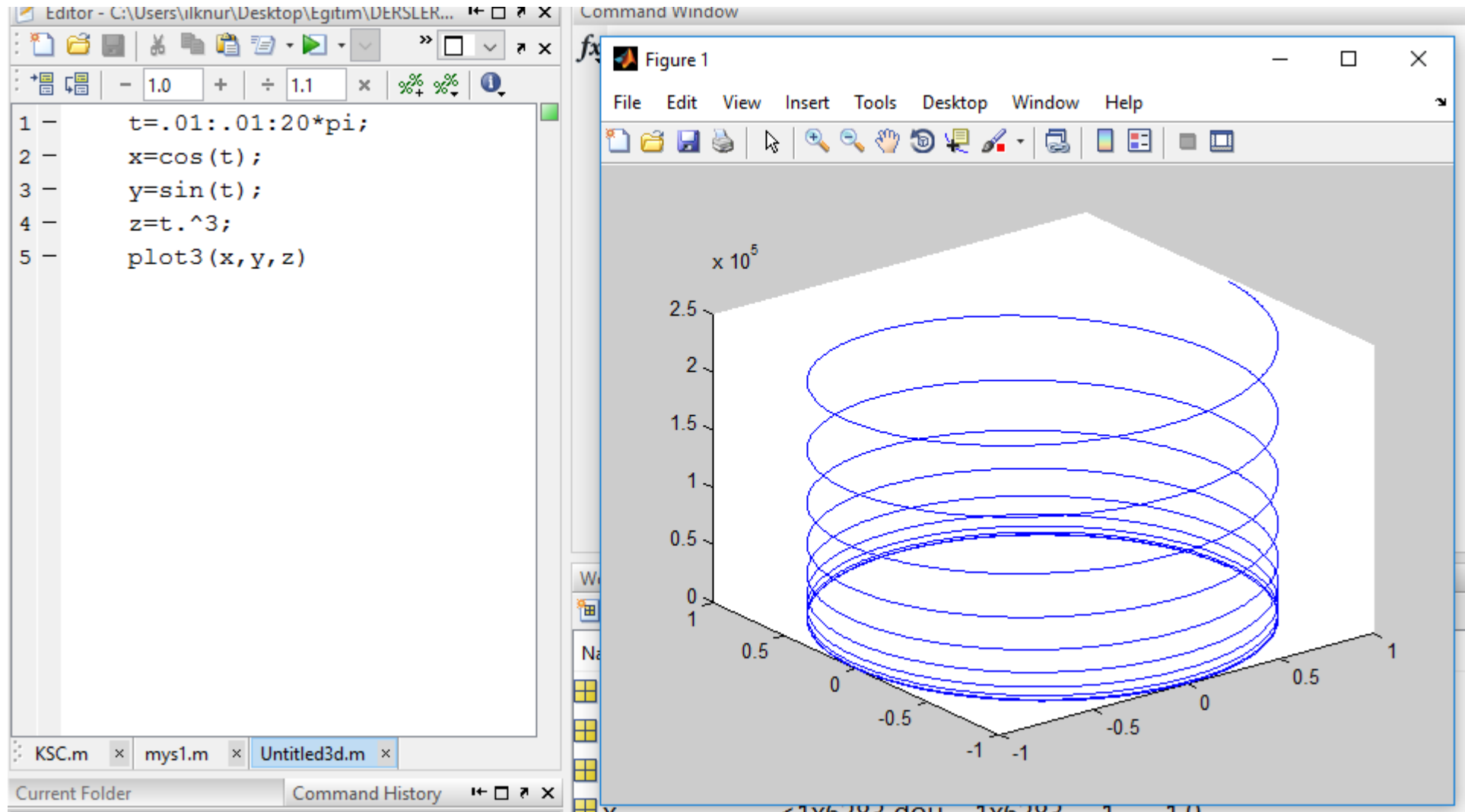
Diğer bir deyişle (x_1, y_1, z_1) , (x_2, y_2, z_2) , (x_3, y_3, z_3) noktaları birleştirilir.

Eksen adlandırılmalarında xlabel ve ylabel fonksiyonlarına ilave olarak zlabel fonksiyonu da kullanılır.

```
Editor - D:\Docs\Matlab\Matlab_ders\Matlab19_hafta_uygulama\graf4.m
File Edit Text Go Tools Desktop Window Help
1 x=[10 20 30 40];
2 y=[10 20 30 40];
3 z=[0 230 75 600];
4 plot3(x,y,z)
5 xlabel('x eksen')
6 ylabel('y eksen')
7 zlabel('z eksen')
8 title('3D bir doğru')
9 text(40,40,0, 'Origin')
10 grid
script Ln 10 Col 5 OK
```

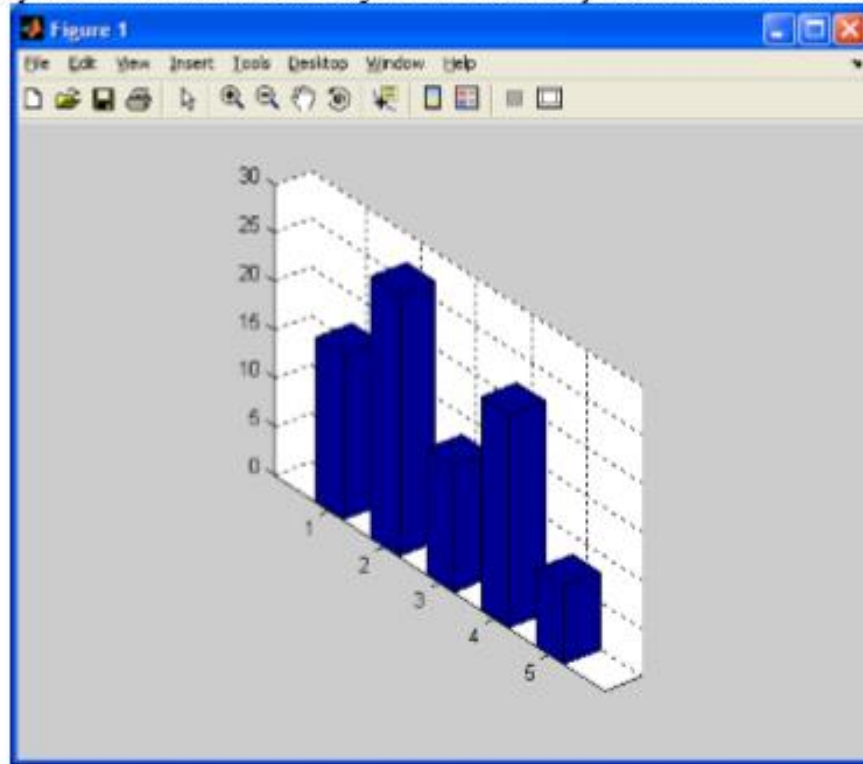


$t=0.01$ ile 20π arasında 0.01 artmakta ise
 $x=\cos(t)$; $y=\sin(t)$; $z=t^3$ uc boyutlu ciziniz



3. 3D Çubuk (bar) Grafikleri

2D çubuk grafik çizimlerinde bar fonksiyonunun *bar3* şeklinde kullanılmasıyla elde edilir.



```
Editor - D:\Docs\Matlab\Matlab_ders\Haftalar\9_...  
File Edit Text Go Tools Desktop Window Help  
1 x=[17 27 13 22 8];  
2 bar3(1:5,x,0.5)
```

3 Boyutlu Ağ (mesh) ve Yüzey (surf) Grafikleri

$z=f(x,y)$ şeklinde üç boyutlu fonksiyonların yüzeyini gösterebilmek için 3D ağ (mesh) grafikleri çizmek gerekir. Önce meshgrid fonksiyonu ile x ve y koordinatlarının oluşturduğu ikililerle bir değerler matrisi hazırlanır daha sonra mesh fonksiyonu ile bu değerler matrisi kullanılarak üç boyutlu grafik çizilir.

Genel yazım şekli;

```
[x,y]=meshgrid (x1:i:x2, y1:j:y2)  
mesh (x,y,z)
```

x1,y1 : x ve y'nin başlangıç değerleri

x2,y2 : x ve y'nin bitiş değerleri

i,j : x ve y'nin artım miktarları

$z=5\cos x.\sin y$ fonksiyonunu $-\pi$ ile π aralığında $\pi/50$ artımla üç boyutlu olarak çizdiriniz.

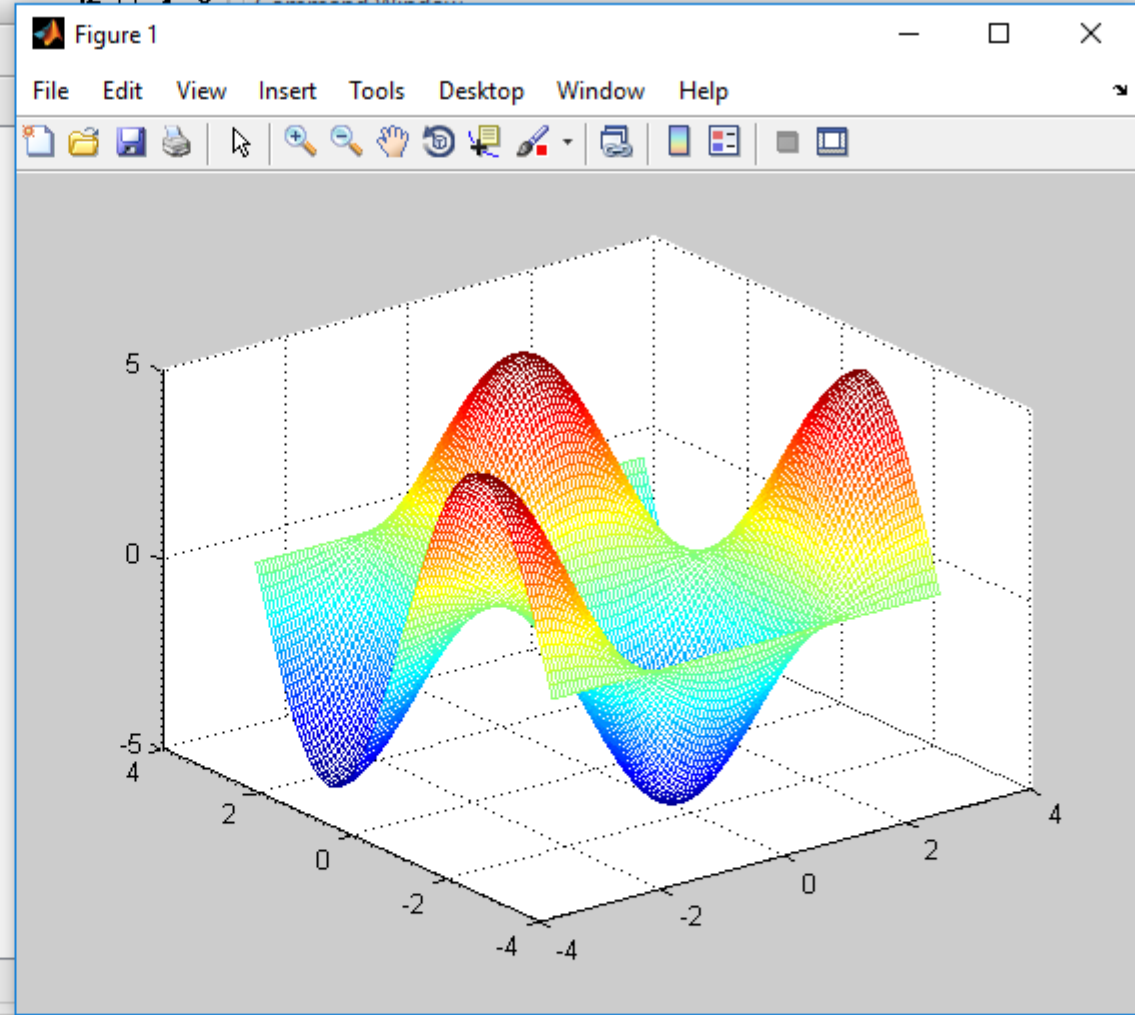

```
Editor - C:\Users\ilknur\Desktop\Egitim\DESLER\MATLAB\Untitled2.m
1 - [x,y]=meshgrid(-pi:pi/50:pi,-pi:pi/50:pi)
2 - z=5*cos(x).*sin(y)
3 - mesh(x,y,z)

KSC.m x mys1.m x Untitled3d.m x Untitled5.m x Untitled2.m x
Current Folder Command History
%-- 3/29/2017 9:44 AM --%
xrandn(100,1)*3
clc
```

meshgrid fonksiyonu ile birlikte mesh komutu yerine surf komutu kullanılırsa çizilen yüzeyin grid boyunca farklı bölgeleri farklı renklerle boyanır.

Genel yazım şekli;

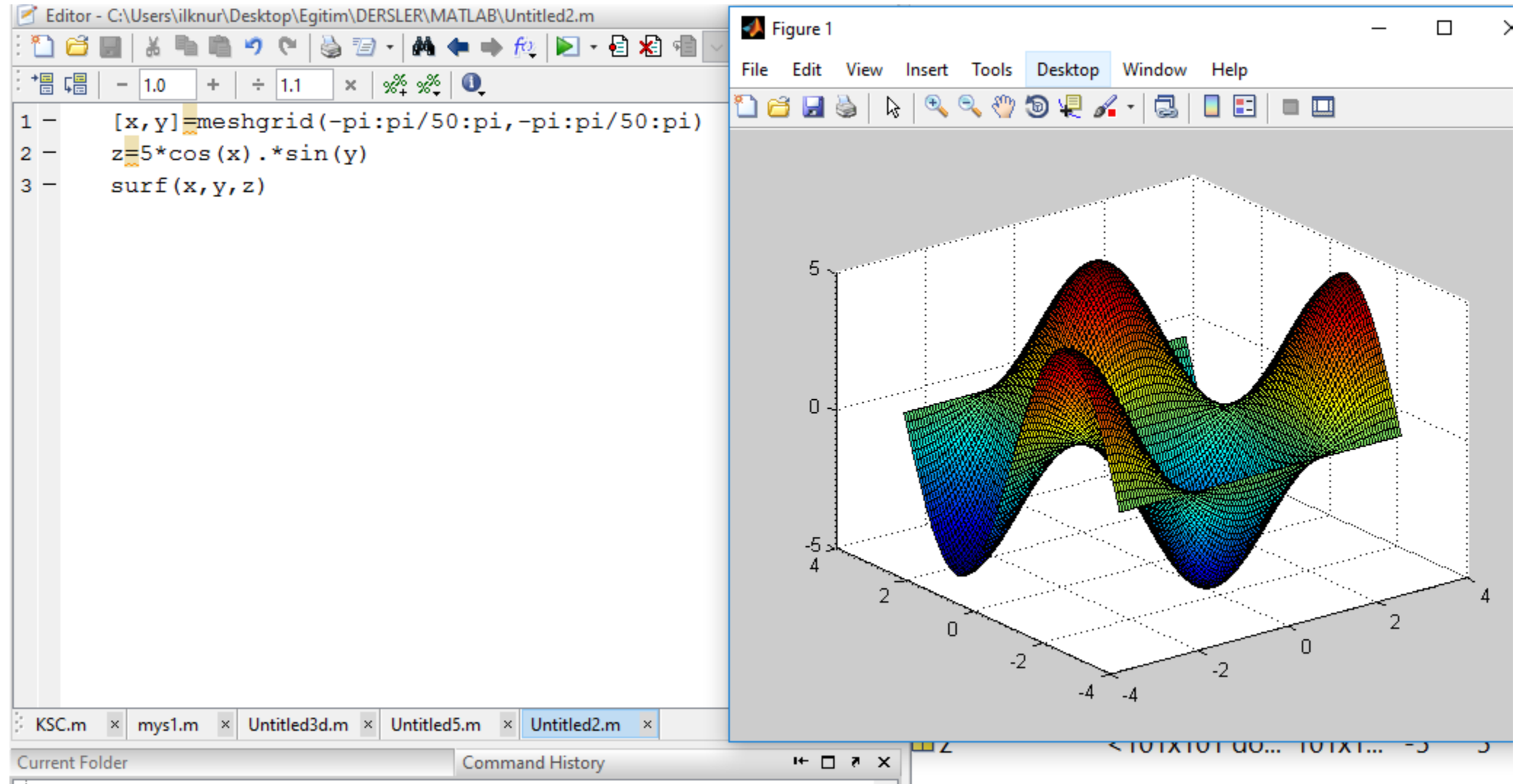
`[x,y]=meshgrid (x1:i:x2, y1:j:y2) surf (x,y,z)`



meshgrid fonksiyonu ile birlikte mesh komutu yerine surf komutu kullanılırsa çizilen yüzeyin grid boyunca farklı bölgeleri farklı renklerle boyanır.

Genel yazım şekli;

`[x,y]=meshgrid (x1:i:x2, y1:j:y2) surf (x,y,z)`



MATLAB/Grafik-İnterpolasyon



- x ve y koordinatları bilinen noktalara ilişkin üçüncü bir bilgi (örneğin, *yükseklik*, *yükseklik değişimi*, *sıcaklık*, *nem*, *basınç*, *gelgit deformasyonu*, *anomali* vb.) olduğunda, noktaların çevrelediği alanın içindeki hayali noktalar için bu bilgiler, çeşitli matematiksel yaklaşımlarla üretilebilir. Bu işleme, kısaca, **interpolasyon** denir.
- Böylesi hayali noktalar, alanın içindeki belirli büyüklükteki kareler ağının köşe noktaları olabilir. Bu noktalara grid noktaları denir. Matlab'de, sonraki interpolasyon işlemlerinde kullanılmak üzere, **meshgrid** fonksiyonu ile bu noktaların x-y koordinatları belirlenir.
- Öncelikle, x ve y eksenleri, söz konusu karenin kenar büyüklüğü kadar parçalara ayrılır. Örneğin, kenar büyüklüğü 10 m olsun: Böylece eksenler aşağıdaki biçimde 10 m'lik parçalara bölünür;

```
x1=xmin:10:xmax;   y1=ymin:10:ymax
```

- Daha sonra, **[XI, YI]=meshgrid(x1,y1)** ile gridlerin köşe nokta koordinatları XI ve YI matrislerine yazdırılır.
 - **HI=griddata(y,x,H,YI,XI,'v4')** fonksiyonu ile x ve y koordinatlarına sahip **jeodezik** noktalardaki üçüncü bilginin toplandığı H, koordinatları XI ve YI'da tanımlanmış grid noktaları için v4 yöntemiyle interpolate edilir; grid noktalarına ilişkin üçüncü bilgi HI vektöründe toplanır. (Not: v4 yönteminden başka, cubic, linear gibi interpolasyon yöntemleri de bulunur)
-

MATLAB/Grafik-İnterpolasyon



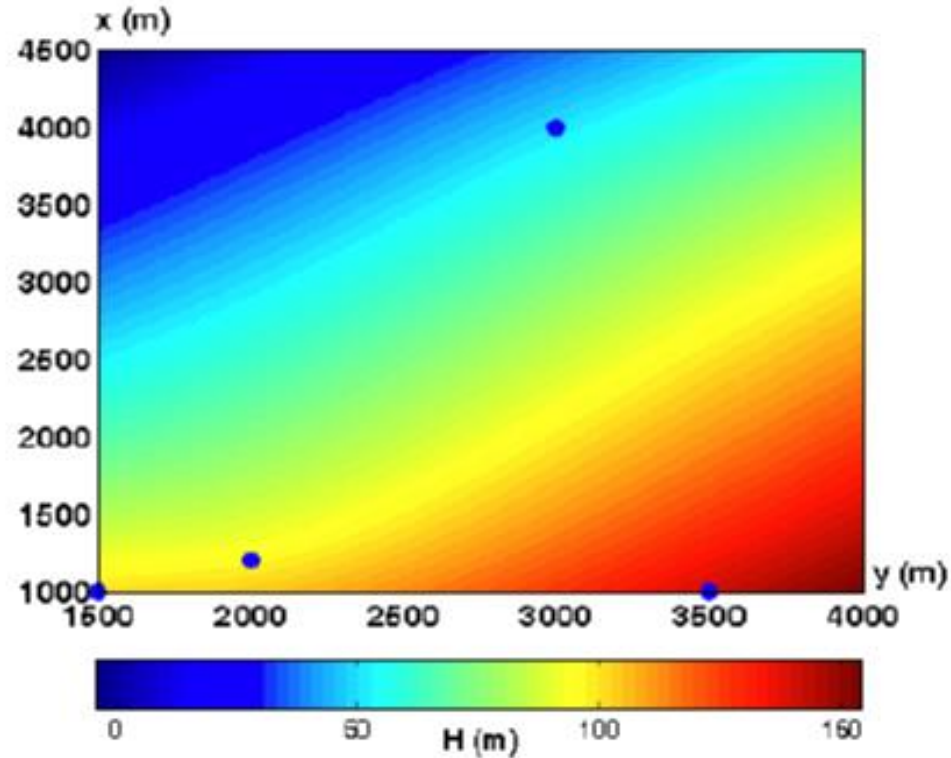
- **Örnek:** Nokta koordinatları, $x=[1000;1200;4000;1000]$, $y=[1500; 2000;3000;3500]$ ile ve bu noktaların yükseklikleri, $H=[100.0000;95.9850;50.5000;140.1200]$ ile tanımlansın. Bölgeyi 10 m'lik gridlere bölerek, bölgenin yükseklik değerlerini gösteren bir renk haritası hazırlayınız.

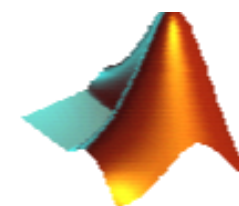
```
clear,clc
x=[1000;1200;4000;1000];
y=[1500;2000;3000;3500];
H=[100.0000;95.9850;50.5000;140.1200];
x1=1000:10:4500;
y1=1500:10:4000;

[XI,YI]=meshgrid(x1,y1);

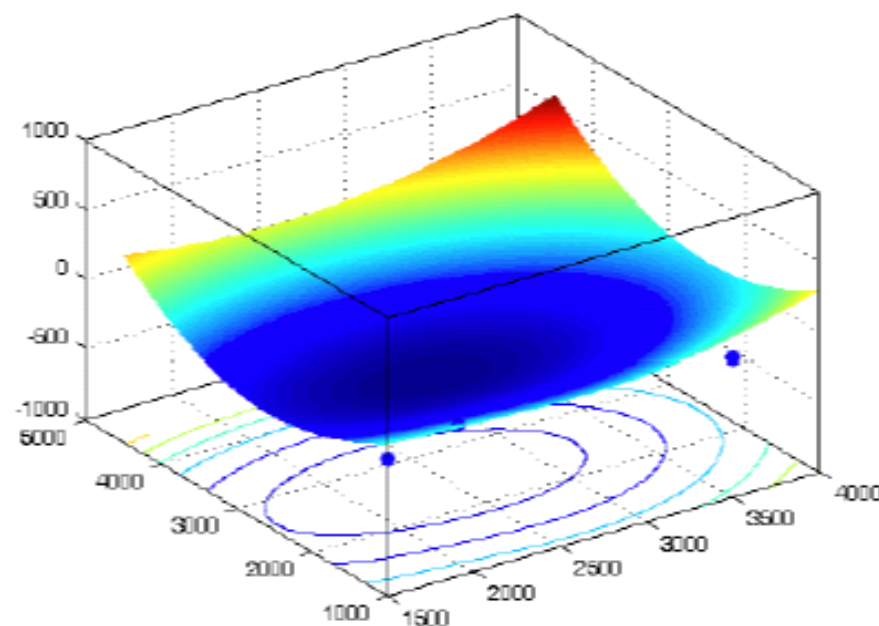
HI=griddata(y,x,H,YI,XI,'v4');
hold on,

pcolor(YI,XI,HI),
shading interp
colormap('jet')
plot(y,x,'o','MarkerFacecolor','b')
hold off
```





- Contour (örneğin, eş yükselti eğrileri) haritası için `contour` ve `clabel` fonksiyonlarına,
- Üç boyutlu yüzey çizimleri için `mesh`, `surf`, `surfl` ve `surfsc` fonksiyonlarına bakınız.

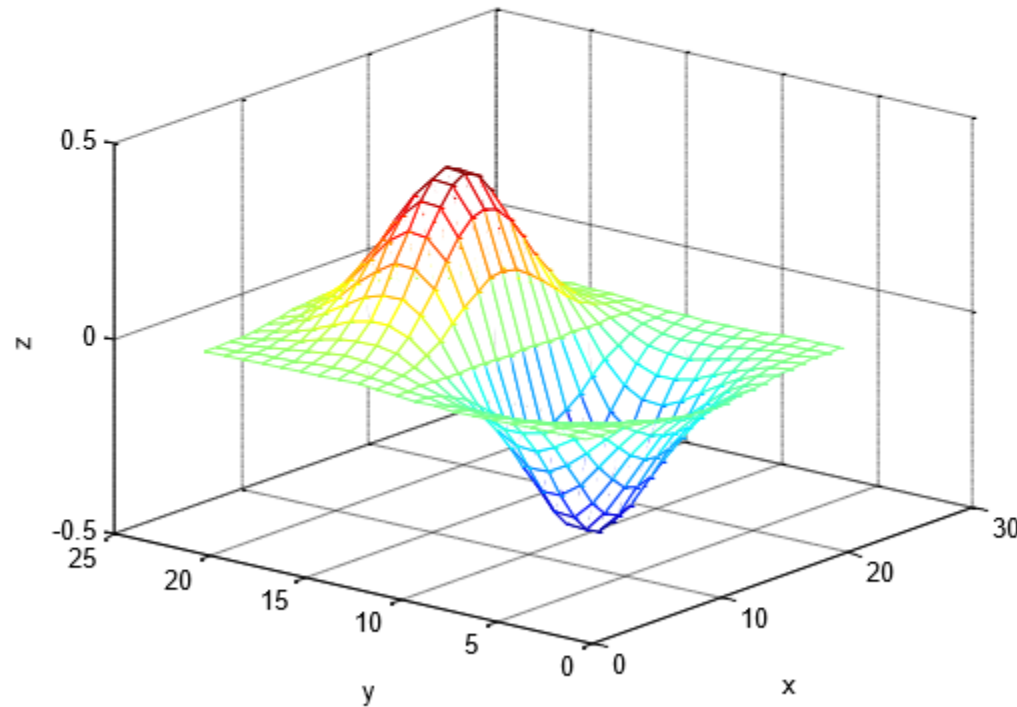


`surfsc` fonksiyonu ile oluşturulmuş bir yüzey grafiği



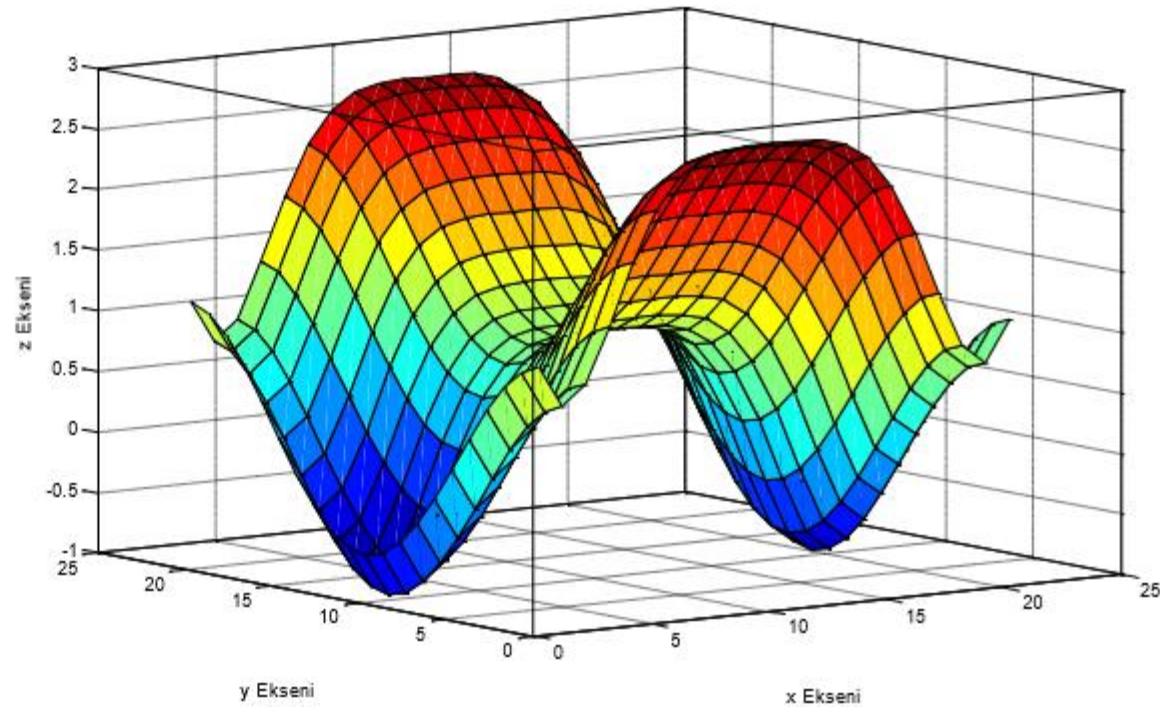
Örnek: x ve y'nin -2 ile 2 arasındaki değerlerinin 0.2 artımı için $z(x,y)=x*\exp(-x^2-y^2)$ yüzey grafiğini çizdirelim.

```
>> [x,y] = ndgrid(-2:.2:2, -2:.2:2);  
>> z = x .* exp(-x.^2 - y.^2);  
>> mesh(z)
```



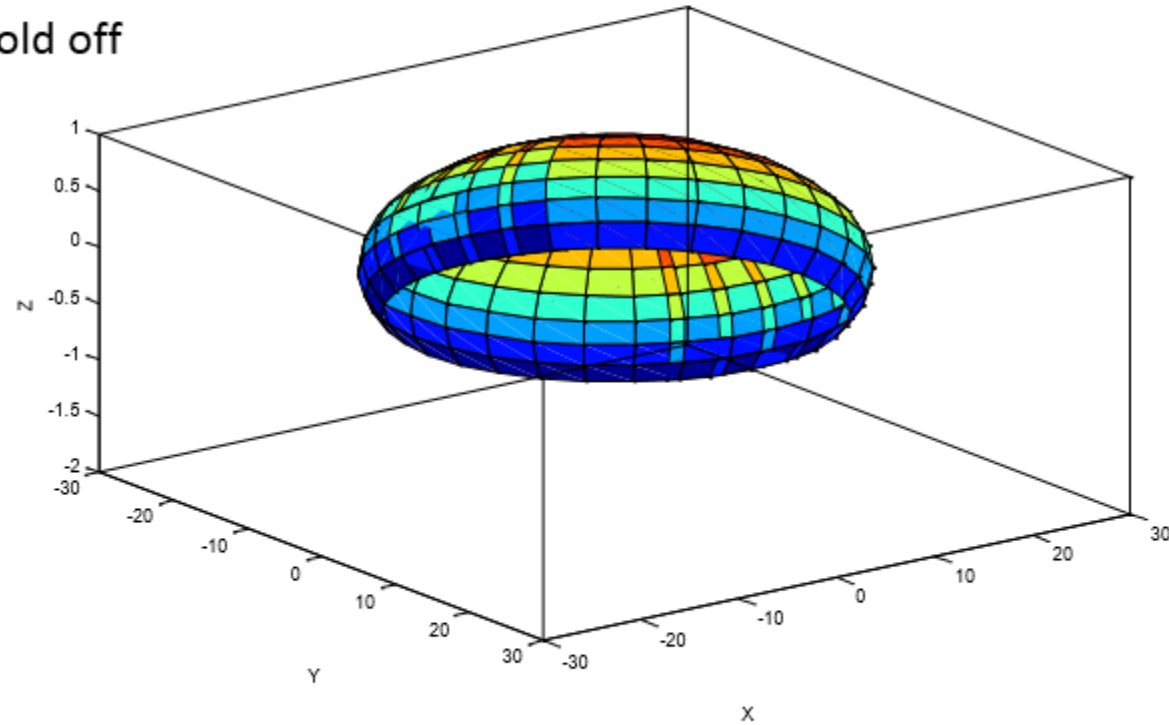
Örnek: x ve y'nin -2 ile 2 arasındaki değerlerinin 0.2 artımı için $z(x,y)=x*\sin(x)+\cos(y^2)$ yüzey grafiğini çizdirelim.

```
>> [x,y] = ndgrid(-2:.2:2, -2:.2:2);  
>> z = x.*sin(x)+ cos(y.^2);  
>> surf(z)
```



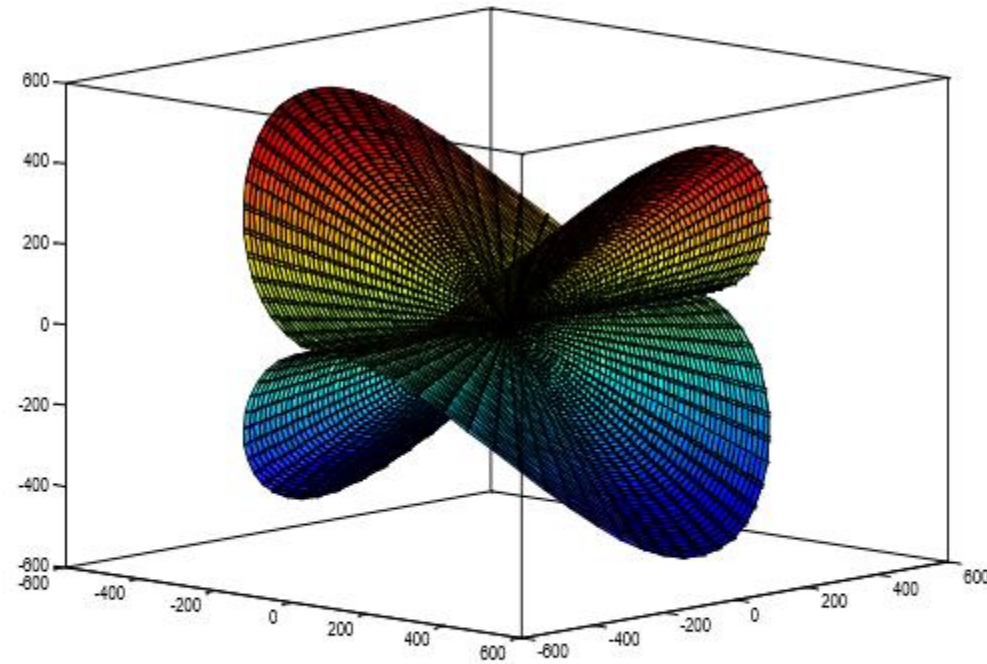
Örnek: x ve y 'nin -2 ile 2 arasındaki değerlerinin 0.2 artımı için $X(x,y)=\sin(x)*\cos(y)$,
 $Y(x,y)=\sin(x)*\sin(y)$ ve $Z(x,y)=\cos(x)$ fonksiyonlarını çözdürüp aynı yüzey grafiğinde
çizdirelim.

```
>> [x,y] = ndgrid(-2:.2:2, -2:.2:2);  
>> X=sin(x)*cos(y);  
>> Y=sin(x)*sin(y);  
>> Z=cos(x);  
>> hold on, surf(X,Y,Z), hold off
```



Örnek: u ve v 'nin -5 ile 5 arasındaki değerlerinin 0.1 artımı için $K(u,v)=u*\sin(v)$, $L(u,v)=u*\cos(v)$ ve $M(u,v)=u*\sin(v)^2$ fonksiyonlarını çözdürüp aynı yüzey grafiğinde çizdirelim.

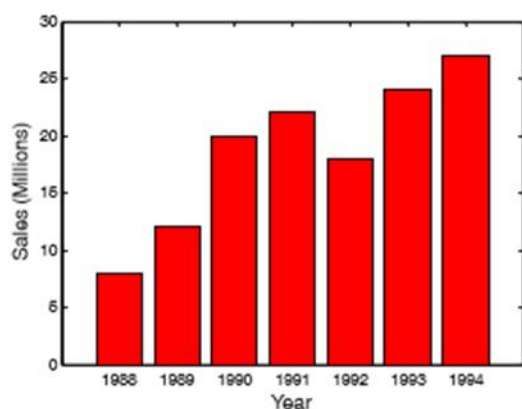
```
>> [u,v] = ndgrid(-5:.1:5, -5:.1:5);  
>> K=u*sin(v);  
>> L=u*cos(v);  
>> M= u*sin(v).^2;  
>> hold on, surf(K,L,M), hold off
```



Vertical Bar Plot

Function
format:

`bar(x, y)`



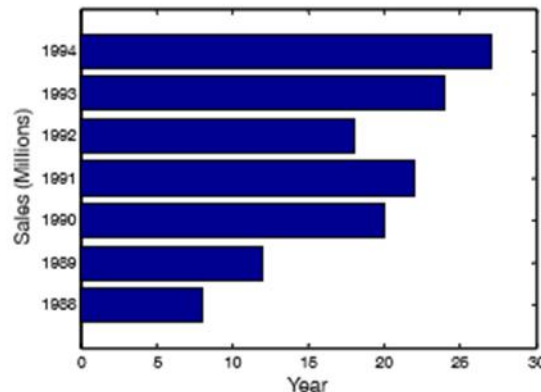
```
yr=[1988:1994];  
sle=[8 12 20 22 18 24 27];  
bar(yr,sle,'r')  
xlabel('Year')  
ylabel('Sales (Millions)')
```

The
bars are
in red.

Horizontal Bar Plot

Function
format:

`barh(x, y)`

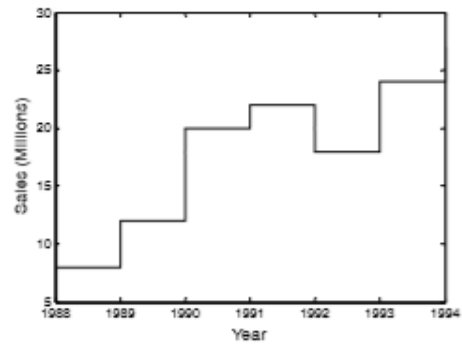


```
yr=[1988:1994];  
sle=[8 12 20 22 18 24 27];  
barh(yr,sle)  
xlabel('Sales (Millions)')  
ylabel('Year')
```

Stairs Plot

Function
format:

`stairs(x,y)`

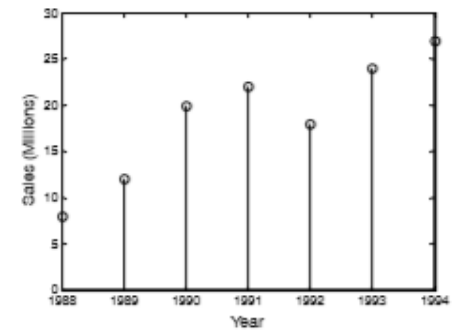


```
yr=[1988:1994];  
sle=[8 12 20 22 18 24 27];  
stairs(yr,sle)
```

Stem Plot

Function
Format

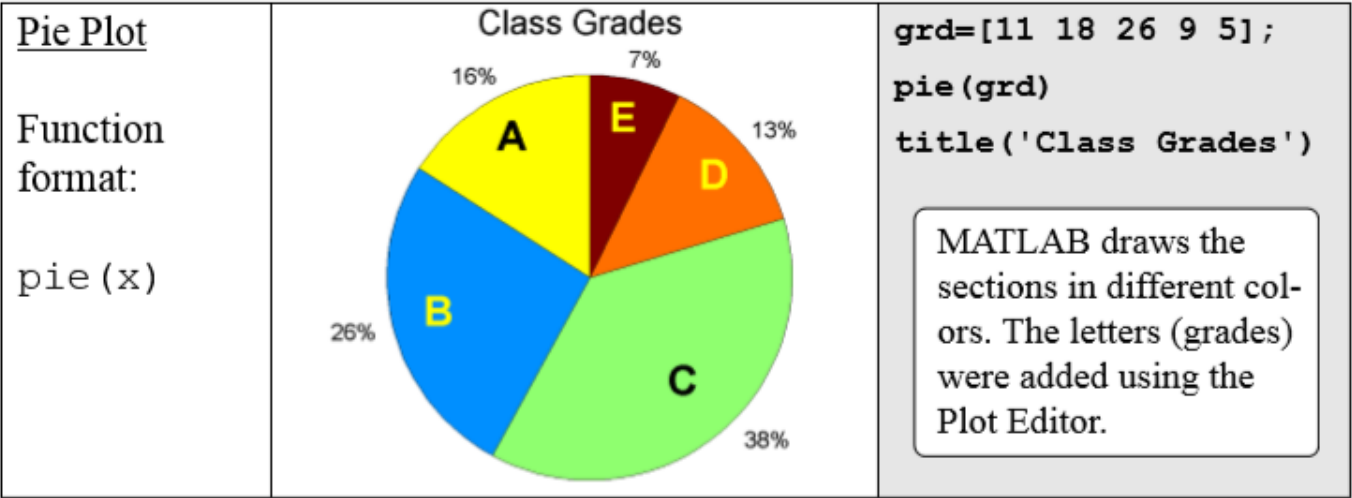
`stem(x,y)`



```
yr=[1988:1994];  
sle=[8 12 20 22 18 24 27];  
stem(yr,sle)
```

Pie charts are useful for visualizing the relative sizes of different but related quantities. For example, the table below shows the grades that were assigned to a class. The data is used to create the pie chart that follows.

Grade	A	B	C	D	E
Number of students	11	18	26	9	5

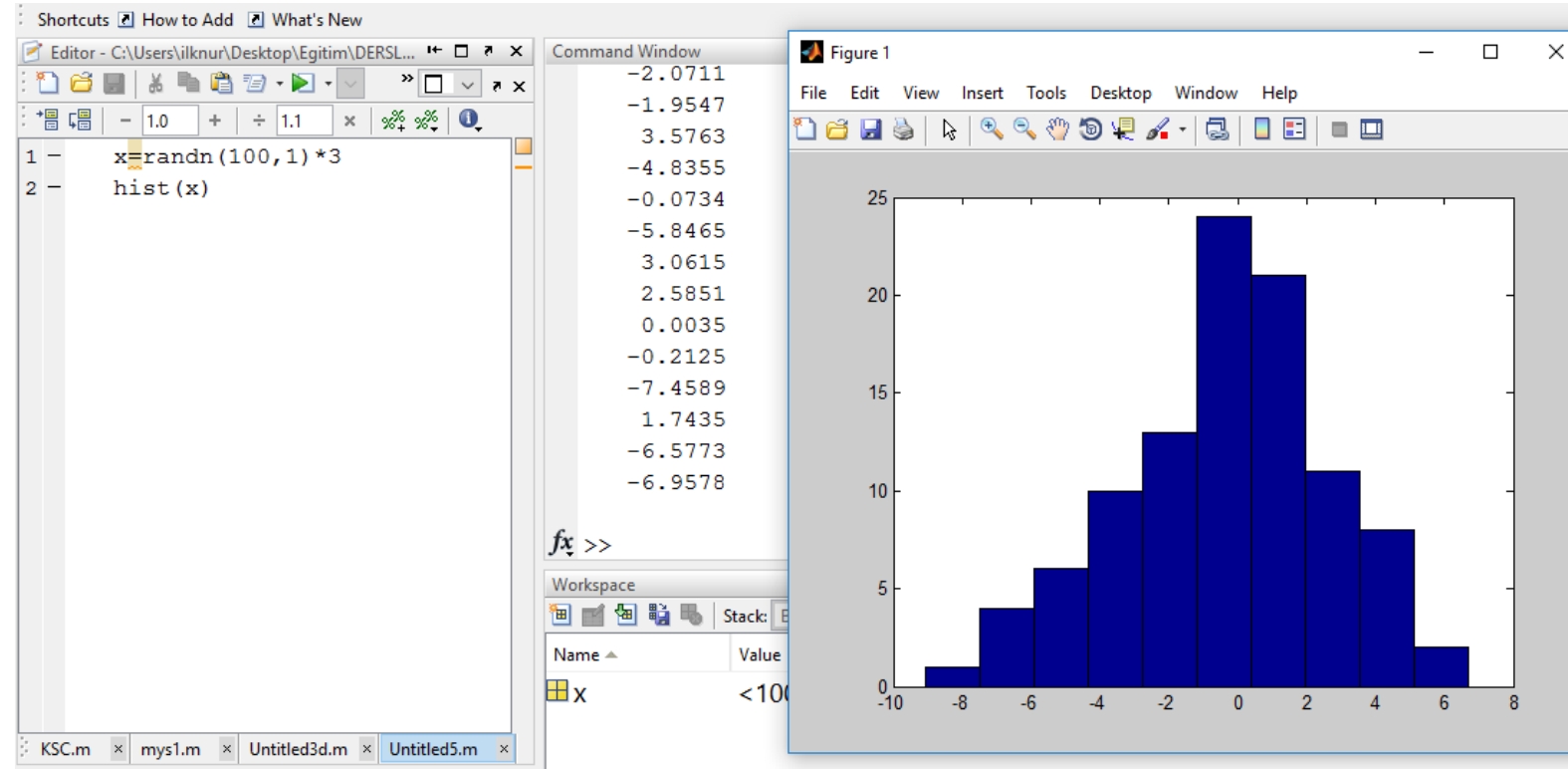


➤ Histogram

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.)



5.8 HISTOGRAMS

Histograms are plots that show the distribution of data. The overall range of a given set of data points is divided into subranges (bins), and the histogram shows how many data points are in each bin. The histogram is a vertical bar plot in which the width of each bar is equal to the range of the corresponding bin and the height of the bar corresponds to the number of data points in the bin. Histograms are created in MATLAB with the `hist` command. The simplest form of the command is:

`hist(y)`

`y` is a vector with the data points. MATLAB divides the range of the data points into 10 equally spaced subranges (bins) and then plots the number of data points in each bin.

For example, the following data points are the daily maximum temperature (in °F) in Washington, DC, during the month of April 2002: 58 73 73 53 50 48 56 73 73 66 69 63 74 82 84 91 93 89 91 80 59 69 56 64 63 66 64 74 63 69 (data from the U.S. National Oceanic and Atmospheric Administration). A histogram of this data is obtained with the commands:

```
>> y=[58 73 73 53 50 48 56 73 73 66 69 63 74 82 84 91 93 89  
91 80 59 69 56 64 63 66 64 74 63 69];  
>> hist(y)
```

The plot that is generated is shown in Figure 5-11 (the axis titles were added using the Plot Editor). The smallest value in the data set is 48 and the largest is 93,

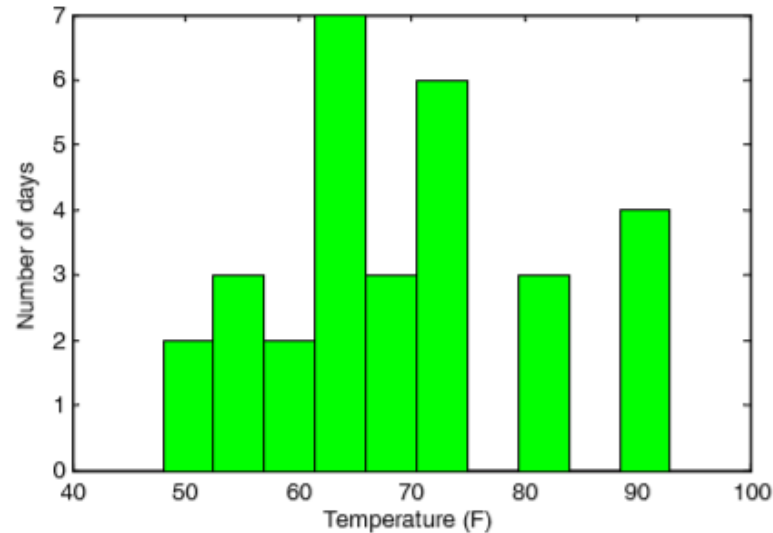


Figure 5-11: Histogram of temperature data.

which means that the range is 45 and the width of each bin is 4.5. The range of the first bin is from 48 to 52.5 and contains two points. The range of the second bin is from 52.5 to 57 and contains three points, and so on. Two of the bins (75 to 79.5 and 84 to 88.5) do not contain any points.

Since the division of the data range into 10 equally spaced bins might not be the division that is preferred by the user, the number of bins can be defined to be different than 10. This can be done either by specifying the number of bins, or by specifying the center point of each bin as shown in the following two forms of the

hist command:

`hist(y,nbins)`

or

`hist(y,x)`

nbins is a scalar that defines the number of bins. MATLAB divides the range in equally spaced subranges.

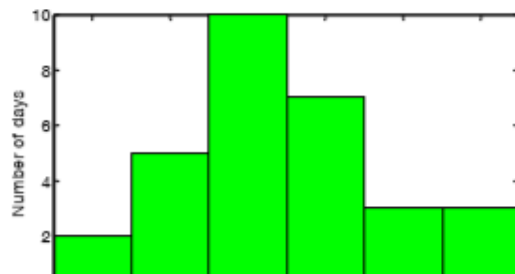
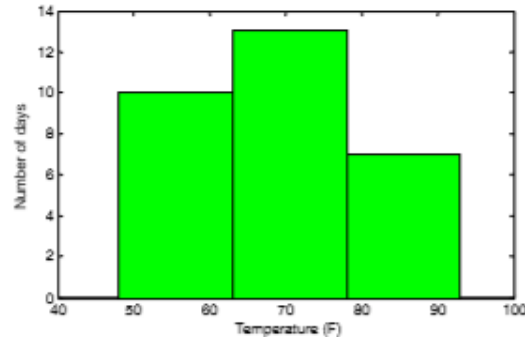
x is a vector that specifies the location of the center of each bin (the distance between the centers does not have to be the same for all the bins). The edges of the bins are at the middle point between the centers.

In the example above the user might prefer to divide the temperature range into three bins. This can be done with the command:

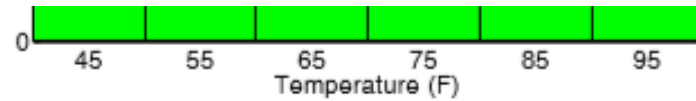
```
>> hist(y,3)
```

As shown in the top graph, the histogram that is generated has three equally spaced bins.

The number and width of the bins can also be specified by a vector **x** whose elements define the centers of the bins. For example, shown in the lower graph is a histogram that displays the temperature data from above in six bins with an equal width of 10 degrees.



The elements of the vector x for this plot are 45, 55, 65, 75, 85, and 95. The plot was obtained with the following commands:



```
>> x=[45:10:95]
x =
    45    55    65    75    85    95
>> hist(y,x)
```

The `hist` command can be used with options that provide numerical output in addition to plotting a histogram. An output of the number of data points in each bin can be obtained with one of the following commands:

`n=hist(y)`

`n=hist(y,nbins)`

`n=hist(y,x)`

The output `n` is a vector. The number of elements in `n` is equal to the number of bins, and the value of each element of `n` is the number of data points (frequency count) in the corresponding bin. For example, the histogram in Figure 5-11 can

also be created with the following command:

```
>> n = hist(y)
```

```
n =
```

```
2 3 2 7 3 6 0 3 0 4
```

The vector `n` shows how many elements are in each bin.

The vector `n` shows that the first bin has two data points, the second bin has three data points, and so on.

An additional, optional numerical output is the location of the bins. This output can be obtained with one of the following commands:

```
[n xout]=hist(y)
```

```
[n xout]=hist(y,nbins)
```

`xout` is a vector in which the value of each element is the location of the center of the corresponding bin. For example, for the histogram in Figure 5-11:

```
>> [n xout]=hist(y)
```

```
n =
```

```
2 3 2 7 3 6 0 3 0 4
```

```
xout =
```

```
50.2500 54.7500 59.2500 63.7500 68.2500 72.7500  
77.2500 81.7500 86.2500 90.7500
```

The vector `xout` shows that the center of the first bin is at 50.25, the center of the second bin is at 54.75, and so on.