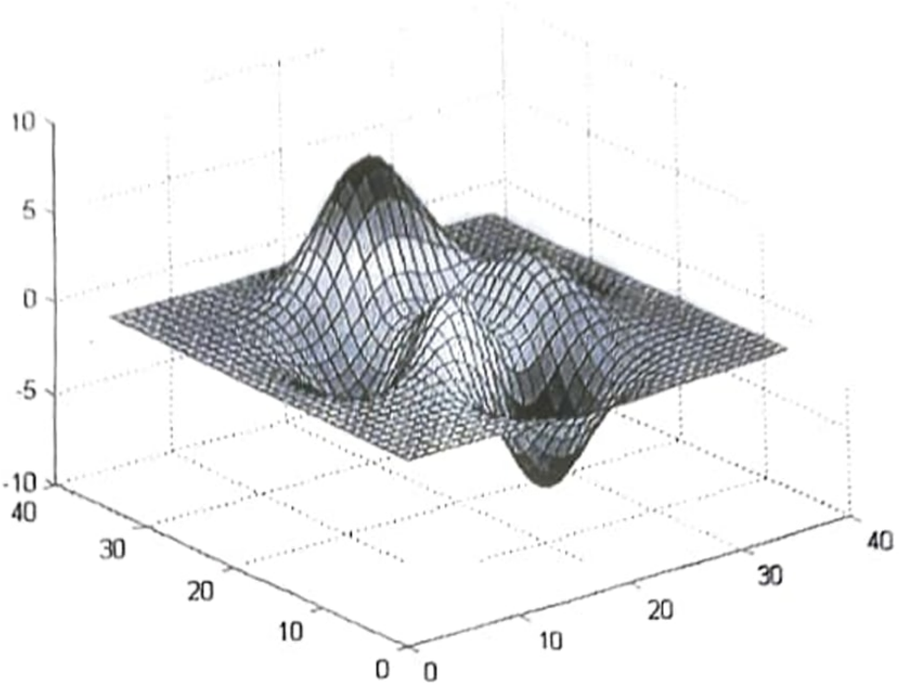
ÖRNEKLERLE

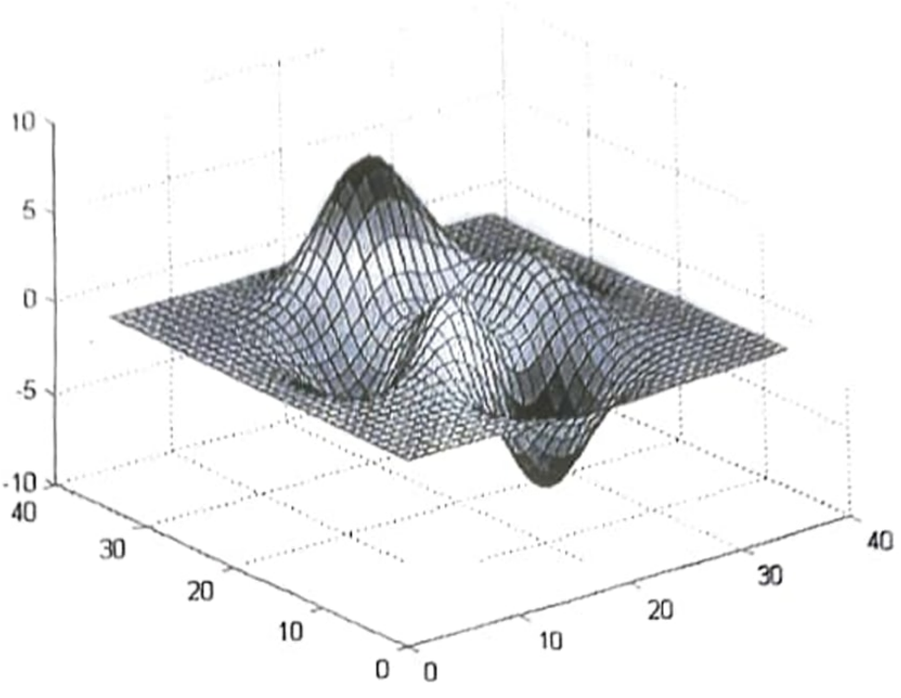
MATLAB VE SİMULİNK



BAHADIR GEDİKBEY

ÖRNEKLERLE

MATLAB VE SİMULİNK



BAHADIR GEDİKBEY

GYTE Yayınları No. 9
ISBN 975-8316-09-5

Yazan
Bahadır GEDİKBEY

Baskı
Celal ÖZBAKIR
GYTE Basımevi

Nisan 2004

Bu eserin kısmen veya tamamen iktibası Gebze Yüksek Teknoloji Enstitüsü Rektörlüğünün yazılı iznine bağlıdır.

© 2004 Gebze Yüksek Teknoloji Enstitüsü

GENEL DAĞITIM:
Gebze Yüksek Teknoloji Enstitüsü Rektörlüğü
Tel: 0(262) 653 84 97

Kızım

AYŞE KEVSER GEDİKBEY'E

Kızım

AYŞE KEVSER GEDİKBEY'E

ÖNSÖZ

Bu kitap, Matlab ve Simulink programlarını, mühendislik eğitimi alan lisans ve lisansüstü öğrencilerine tanıştırmayı amaçlamaktadır.

Kitapta konu anlatımına çok fazla girilmemiş ve örneklerle okuyucunun konuyu kolayca kavraması amaçlanmıştır. Kitapta her komut için en az bir örnek verilmiştir.

Kitabın hazırlanmasında bana her türlü desteği veren eşim Dt. Kadriye Gedikbey'e, akademik çalışmalarımda maddi manevi yardımlarını esirgemeyen babam Prof.Dr. Tevfik Gedikbey'e ve annem Emk.Öğr. Naime Gedikbey'e teşekkürü bir borç bilirim.

Bahadır Gedikbey
Gebze Yüksek Teknoloji Enstitüs
Elektronik Mühendisliği Bölümü
Araştırma Görevlisi

ÖNSÖZ

Bu kitap, Matlab ve Simulink programlarını, mühendislik eğitimi alan lisans ve lisansüstü öğrencilerine tanıştırmayı amaçlamaktadır.

Kitapta konu anlatımına çok fazla girilmemiş ve örneklerle okuyucunun konuyu kolayca kavraması amaçlanmıştır. Kitapta her komut için en az bir örnek verilmiştir.

Kitabın hazırlanmasında bana her türlü desteği veren eşim Dt. Kadriye Gedikbey'e, akademik çalışmalarımda maddi manevi yardımlarını esirgemeyen babam Prof.Dr. Tevfik Gedikbey'e ve annem Emk.Öğr. Naime Gedikbey'e teşekkürü bir borç bilirim.

Bahadır Gedikbey
Gebze Yüksek Teknoloji Enstitüs
Elektronik Mühendisliği Bölümü
Araştırma Görevlisi

DİZİLER 1
Dizi Girişinin Yapılması 1
Özel Diziler 1
randperm 1
linspace 1
logspace 2
Dizilerde Matematik İşlemleri 2
Toplama 2
Çarpma 2
Bölme 3
Üst Alma 3
Dizilerde Yapılan Çeşitli İşlemler 4
Bazı Özel Komutlar 5
length 5
ismember 5
Dizilerde Küme İşlemleri 5
union 5
intersect 5
setdiff 5
Alıştırmalar 5
MATRİSLER 9
Matris Girişleri 9
Matrislerde Matematik İşlemleri 9
Matris Çarpımı 9
Matris Toplamı 10
Matris Çıkarması 10
Matrislerde Karşılıklı Elemanların Çarpılması 10
Bir Matrisin Kuvvetinin Alınması 11
Bir Matrisin Her Elemanının Kuvvetinin Alınması 11
Matrislerde Bölüm 11
Matris İşlemleri 12
İnv 12
Det 12
Özel Matrisler 13
Sıfırlar Matrisi 13
Birler Matrisi 13
Birim Matris 13
Rastgele Değerler Matrisi 13
Dioganal Matris 14
Magic matrisi 14
Magic 15
Matris Değişiklikleri 15
Reshape 15

Matris Boyutunun Bulunması 20
Denklem Çözümlerinde Matrislerin Kullanılması 21
Bazı Matris Çeşitleri 22
Companion matris 22
Hadamard Matris 23
Hankel Matris 23
Hilbert Matris 23
Bazı Özel Matris Komutları 24
Any 24
All 24
Find 25
Insan 25
Finite 25
Rank 25
Trace 25
Alıştırmalar 30
Problemler 30
EKRANDAN SORMA VE EKRANA YAZMA 31
disp 31
fprintf komutu 31
BİLGİ ANALİZİ VE İSTATİSLİK 33
Max 33
Min 33
Sort 33
Sum 33
Prod 33
Cumsum 33
Cumprod 33
Mean 33
Geomean 33
Median 33
Var 33
Std 33
Factor 33
Perms 33
Primes 33
Hist 33

Toplama 50
Çarpma 51
Bölme 51
Komutlar 52
Abs 52
Angle 53
Conj 5
Real 5
İmag 5
Complex 5
Polar 5
Alıştırmalar 5
Problemler 5
POLİNOMLAR 5
Roots komutu 5
Residue komutu 5
Poly komutu 5
Conv komutu 5
Deconv komutu 5
İki polinomun Toplamı 5
Bir Polinomun Skaler İle Çarpımı 5
Polyder Komutu 5
Polyint Komutu 5
Alıştırmalar 5
Problemler 5
MATLAB'DE GRAFİK ÇİZME 5
Çizgi Çeşitleri 5
Nokta Çeşitleri 5
Renk Çeşitleri 5
Text 5
Area 5
Fill 5
Pie 5
Line 5

Compass 82
Rose 83
Stairs 83
Animasyon Hazırlama 84
Getframe 84
Movie 84
Alıştırmalar 90
Problemler 93
ÜÇ BOYUTLU GRAFİK ÇİZİMİ 93
Komutlar 93
Plot3 94
Mesh 95
Surf 97
Surfl 98
Fill3 99
Contour3 99
Bar3 100
Bar3h 101
Pie3 102
Cylinder 103
Sphere 103
Waterfall 104
Peaks 105
Stem 105
Box 105
Ezmesh 106
Ezsurf 107
Scatter3 108
Rotate3d 10
Camdolly 10
Camorbit 10
Camzoom 10
Brighten 11
Işıklandırma 11
TEMEL MATEMATİK FONKSİYONLARI 11
abs 1
acos,acosh 1
acot,acoth 1
acsc,acsch 1
angle 1

conj 117
cos,cosh,csc,csch 117
exp 118
fix 118
flor 118
gcd 119
imag 119
lcm 120
log 120
log2 120
log10 121
mod 121
real 121
rem 122
round 122
sec,sech 122
sin,sinh 122
sqrt 123
tan,tanh 123
Alıştırmalar 128
Problemler *129*
ONKSİYON YAZMA 129
Örnekler 131
Alıştırmalar 132
Problemler *133*
PTİMİZASYON 13
Sıfır Geçiş Noktasını Bulmak 13
Alıştırmalar 13
Problemler *13*
NTEGRAL VE TÜREV 1
Trapz komutu 1
Quad komutu 1
Quad8 komutu 1
Sembolik İntegral Alma .....

Char 166
Str2mat 166
Str2num 166
Bin2dec 166
Dec2bin 166
Dec2hex 167
Hex2dec 167
Alıştırmalar 169
Problemler *171*
SİMULİNK 173
Örnekler *173*
Karşılaştırma Örneği *173*
Verilen iki sayının önce kompleks sayıya çevrilmesi, sonra genlik ve açının bulunması *174*
Girilen bir sinyalin alt ve üst sınırının belirlenmesi. *174*
Nokta Çarpımı. *174*
Sign İşaretinin kullanılışı. *175*
iki Sinyalin toplamı *176*
Bir dizinin ortasındaki sayının bulunması. 179
GUI (GRAFİKSEL KULLANICI ARA YÜZÜ) 184
Örnek: Bir sinüs sinyalinin frekansının değiştirilmesi. 188
Örnek: Hesap Makinesi 19
Örnek: Bir sayının karesinin, karekökünün veya küpünün alınması 19
Örnek: Bir listede şehir isimlerinin seçilmesi ve bu şehirlerin nüfuslarının yazdırılması 19
SAYISAL İŞARET İŞLEME 20
Bazı Önemli Fonksiyonlar 2
Dürtü (Birim Impulse Fonksiyonu) 2
Birim Basamak Fonksiyonu 2
Konvolüsyon .....
Fourier Dönüşümü .....
IIR Filtreler .....
Butterworth Filtreler .....
Chebyshev Filtreler Tip1 .....
Chebyshev Filtreler Tip2 .....
Elliptic Filtreler .....
IIR Filtrelerde Derece Tahmini .....

Firels fonksiyonu 228
Filtre Uygulamaları 228
Filter komutu 229
Filtfilt komutu 23
Medfilt1 fonksiyonu 23
Dalga Biçimi Üretme 23
Sawtooth komutu 2
Sinc Fonksiyonu 2
Square komutu 2
Tripuls komutu 2
Vco komutu .....
Filtre Tasarım ve Analiz Aleti .....
İnteraktif Sayısal İşaret İşleme Aleti .....
OTOMATİK KONTROL SİSTEMLERİ .....
Sistem Modelleme .....
Doğrusal Modeller .....
Modellerin Oluşturulması .....
Modeller Arasında Çevirim Yapma .....
Doğrusal Modeller Arasında Bağıntılar .....
Kökün Geometrik Yeri (Root Locus) .....
Bode Diyagramı .....
Nyquist Diyagramı .....
residue .....
tf2zp .....
Ltiview Grafiksel Kullanıcı Ara Biriminin Kullanılması .....
Ltiview komutları .....
ELEKTRONİK MÜHENDİSLİĞİ PROBLEMLERİ .....
Doğru Akım Devrelerinin Çözüm Yöntemleri .....
Düğüm Gerilimleri Yöntemi .....
Çevre Akımları Yöntemi .....
Süperpozisyon Yöntemi .....
Thevenin Teoremi .....
Doğru Akımda Bobin ve Kondansitör .....
Kondansitörün Dolması ve Boşalması .....
Alternatif Akım Devrelerinde Bobin ve Kondansatörün İncelenmesi .....

Dizi Girişinin Yapılması

Matlab de dizileri kolayca tanımlamak mümkündür. Bu diziler matematiksel işlemlerde, çizimlerde pratik bir şekilde kullanılır.

A =

1 2 3 4 5 6

şeklindeki bir diziyi;

» A=[1 2 3 4 5 6] veya » A=[1,2,3,4,5,6] şeklinde girebiliriz.

"B" dizisi 1 ile 10 arasında değişsin. Bu diziyi şu şekilde tanımlayabiliriz.

» B=1:10 Sonuç olarak da;

B =

1 2 3 4 5 6 7 8 9 10

dizisi elde edilir.

Örnek:

"C" dizisi 0 ile 55 arasında 5'er artarak değişsin. Bu durumda diziyi şöyle tanımlayabiliriz.

C=0:5:55 Sonuç olarak da ;

C =

0 5 10 15 20 25 30 35 40 45 50 55

dizisini elde ederiz.

Özel Diziler

→Randperm(n):
1 ile n arasında, rastgele dizilmiş n tane sayı üretir.
Örnek:
Randperm(7)
ans = 2 7 4 3 6 5 1

→linspace(a, b) :
Bu komut a ve b arasında düzgün olarak değişen ve 100 elamandan oluşan bir dizi oluşturur.
Örnek:
linspace(1,20)

→linspace(a,b,N):
Bu komut a ve b arasında değişen, düzgün aralıklarla artan "N" elemanlı bir dizi oluşturur.
Örnek:
linspace(0,100,5)
ans =
0 25 50 75 100

Bu komut 10 ve 10 ... oluşan bir vektör oluşturur.

→logspace(a,b,N):
Bu komut 10^a ve 10^b arasında logaritmik olarak düzgün değişen "N" elemanlı bir dizi oluşturur.
Örnek:
logspace(1,3,5)
10 31.6 100 316.2 1000

Dizilerde Matematik İşlemleri

Toplama:
a) Aynı eleman sayısına sahip dizilerde karşılıklı elemanlar toplanır.
Örnek:
```
a=[1 2 4];
b=[3 2 1];
c=a+b

c =    4    4    5
```

b) Bir dizi ile bir sayı toplanılırsa dizinin her elemanına bu sayı eklenir.
Örnek:
```
a=[1 4 2 3 5 2 3];
b=a+4
b= 5    8    6    7    9    6    7
```
Çıkarma:
a) Aynı eleman sayısına sahip dizilerde karşılıklı elemanlar çıkarılır.
Örnek:
```
a=[4 1 2 3];
b=[1 2 1 2];
c=a-b

c =    3   -1    1    1
```
Örnek:
b) Bir diziden bir sayı çıkarılırsa dizinin her elemanından bu sayı çıkarılır.

```
a=[4 1  3 2 2 3];
b=a-3
b= 1   -2    0   -1   -1    0
```
Çarpma:
a) Aynı eleman sayısına sahip dizilerde karşılıklı elemanlar birbiriyle çarpılır.
Örnek:
```
a=[1 4 5 2 3];
b=[1 2 3 2 2];
c=a.*b

c = 1    8   15    4    6
```
b) Bir dizi bir sayı ile çarpılırsa dizinin her elemanı bu sayı ile çarpılır.
Örnek:

b =10 4 6 4 4 2 4

c) dot(x,y): Bu komut x ve y dizilerinin karşılıklı elemanlarının çarpımlarını toplar.

Örnek:

```
x=[2 4 5 1 6]
y=[3 1 2 3 1]
dot(x,y)
```

```
x =
    2    4    5    1    6

y =
    3    1    2    3    1

ans =
   29
```

Bölme:

a) Aynı eleman sayısına sahip dizilerde karşılıklı elemanlar birbirleriye bölünür.

```
a=[4 3 5 2 3];
b=[2 1 2 1 3];
c=a./b
```

c = 2.0000 3.0000 2.5000 2.0000 1.0000

Eğer ters bölme işareti kullanılırsa bu sefer 2. dizideki elemanlar 1. dizideki elemanlar bölünür.

```
c = a.\b
```

c = 0.5000 0.3333 0.4000 0.5000 1.0000

b) Bir dizi bir sayıya bölünürse dizinin her elemanı bu sayı ile bölünür.

```
a=[15 20 13 21 2 10 2];
b=a/2
```

b= 7.5000 10.0000 6.5000 10.5000 1.0000 5.0000 1.0000

Üst Alma:

a)

Bir dizi bir sayının üstü olacak şekilde bir işlem yapılırsa, dizideki her eleman ayrı ayrı o sayının üstünü alır. Sonucun eleman sayısı dizinin eleman sayısına eşittir.

```
a=[1 2 3 2 3 1];
b=3.^a
```

b = 3 9 27 9 27 3

b)

Bir dizinin üstü alınacak olursa, o dizinin her elemanının teker teker üstü alınır.

```
a=[3 2 1 2 3];
b=a.^2
```

b = 9 4 1 4 9

c)
Bir dizinin üstünü başka bir dizi ile alırsak iki dizideki karşılıklı elemanlar birbirinin üstünü alır.

```
a=[3 2 1 2 3];
b=[1 2 2 3 2];
c=a.^b

c = 3    4    1    8    9
```

Dizilerde Yapılan Çeşitli İşlemler

1)
Bir dizinin herhangi bir elemanına ulaşabiliriz, silebiliriz veya değiştirebiliriz.

```
a=[13 15 10 4 6];
a(2)=20;
a

a =   13    20    10    4    6
```

Yukarıdaki örnekte a dizinin 2. elemanına 20 atanmıştır. Bu durumda dizinin eski elemanı silinir.

2)

```
a=[3 15 2 14 6 8];
b=[2 32 1 35 9 3];
a(3)=b(2)
b(5)=a(4)

a =   3    15    32    14    6    8
b =   2    32    1    35    14    3
```

Bu örnekte a dizisinin 3. elemanına b dizisinin 2. elemanı ve b dizisinin 5. elemanına a dizisinin 4. elemanı atanır.

3)
Bir dizinin belirtilen aralıktaki elemanlarını alınıp farklı bir dizi oluşturulabilir.

```
a=[3 15 2 14 6 8 34 90 4 2 10];
a(2:5)
ans =    15    2    14    6
```

Bu örnekte a dizisinin 2 ile 5 arasındaki elemanları çekilir.

```
A(7:end)
Ans = 34    90    4    2    10
```

Bu örnekte ise a dizisinin 7 elemanından sonuncu elemana kadar elemanlar çekilir

```
a([7 2 8 1])
ans = 34    15    90    3
```

Bu örnekte a dizisinin istenilen elemanları konumları belirtilerek çekilir..

4)

```
a=[3 34 90 4 2 10];
b=[1 3 2];
c=[b a]
c = 1    3    2    3    34    90    4    2    10
```

Görüldüğü gibi iki dizi yan yana yazarak yeni bir dizi elde etmek mümkündür.

=[2 3 9 6 4 7];
=ismember(a,b)
dizisindeki hangi elemanların b dizisinde olduğunu belirtir.
= 0 1 1 1 1 0 0 0
=ismember(b,a);
dizisindeki hangi elemanların a dizisinde olduğunu belirtir.
= 1 1 1 0 1 0

Dizilerde Küme İşlemleri:

Dizilerle matematiksel kesişim, birleşim ve fark işlemlerini yapmak mümkündür.

→ **union(a,b):** Birleşme işlemini gerçekleştirir.
=[1 3 9 4 2 10];
=[2 3 3 6];
=union(a,b)
= 1 2 3 4 6 9 10
ve b kümelerinin birleşim kümeleri elde edilir.
→**intersect(a,b):** Kesişme işlemini gerçekleştirir.
= intersect(a,b)
= 2 3
ve b kümelerinin kesişim kümeleri elde edilir.
→**setdiff(a,b):** a-b işlemini gerçekleştirir.
=setdiff(a,b)
= 1 4 9 10
'da olup b'de olmayan elemanlar.
=setdiff(b,a)
= 6
'de olup a'da olmayan elemanlar.

----------**Alıştırmalar**----------

1)Aşağıdaki sayıları elde etmek için tek satırlık bir matlab programı yazınız.
4 7 10 13 16 18 22 25
Çözüm:
>> a=4:3:25
a =
4 7 10 13 16 19 22 25
2) Aşağıdaki sayıları elde edebilmek için bir matlab programı yazınız.
1 8 27 64 125 216
a=[1 2 3 4 5 6]
b=a.^3
b =
1 8 27 64 125 216

a=[3 15 2 14 6 8];
b=length(a);
b=6
→ismember(a,b):
a=[1 3 9 4 2 10 5 8];
b=[2 3 9 6 4 7];
c=ismember(a,b)
a dizisindeki hangi elemanların b dizisinde olduğunu belirtir.
c = 0 1 1 1 1 0 0 0
c=ismember(b,a);
b dizisindeki hangi elemanların a dizisinde olduğunu belirtir.
c = 1 1 1 0 1 0

Dizilerde Küme İşlemleri:

Dizilerle matematiksel kesişim, birleşim ve fark işlemlerini yapmak mümkündür.

→ union(a,b): Birleşme işlemini gerçekleştirir.
a=[1 3 9 4 2 10];
b=[2 3 3 6];
c=union(a,b)
c = 1 2 3 4 6 9 10
a ve b kümelerinin birleşim kümeleri elde edilir.
→intersect(a,b): Kesişme işlemini gerçekleştirir.
c= intersect(a,b)
c = 2 3
a ve b kümelerinin kesişim kümeleri elde edilir.
→setdiff(a,b): a-b işlemini gerçekleştirir.
c=setdiff(a,b)
c = 1 4 9 10
a'da olup b'de olmayan elemanlar.
c=setdiff(b,a)
c = 6
b'de olup a'da olmayan elemanlar.

Alıştırmalar

---

1)Aşağıdaki sayıları elde etmek için tek satırlık bir matlab programı yazınız.
4 7 10 13 16 18 22 25
Çözüm:
>> a=4:3:25
a =
4 7 10 13 16 19 22 25
2) Aşağıdaki sayıları elde edebilmek için bir matlab programı yazınız.
1 8 27 64 125 216
a=[1 2 3 4 5 6]
b=a.^3
b =
1 8 27 64 125 216

3) Aşağıdaki sıralamayı elde ediniz.

20 19 18 17 16 15 14 13 12 11 10

```
>> a=20:-1:10
a =
     20   19   18   17   16   15   14   13   12   11   10
```

4) Aşağıdaki sıralamayı elde ediniz.

3 9 27 81 243

```
>> a=1:5
a =
   1   2   3   4   5
>> b=3.^a
b =
   3   9   27   81   243
```

5) Aşağıdaki diziyi elde ediniz:

5 10 15 20 25 30

```
>> a=5:5:30
a =
   5   10   15   20   25   30
```

veya

```
>> a=[1 2 3 4 5 6]
a =
   1   2   3   4   5   6
>> b=5*a
b =
   5   10   15   20   25   30
```

6) 1 ile 10 arasında rastgele 10 tane sayı üretiniz.

```
>> randperm(10)
ans =
   8   2   10   7   4   3   6   9   5   1
```

7) 5 tane rastgele sayı üretiniz ve bu sayıların değerleri:

1 16 25 9 4 olsun.

```
>> a=randperm(5)
a =
   1   4   5   3   2
>> b=a.^2
b =
   1   16   25   9   4
```

8) Aşağıdaki işlemleri yapınız ve sonuçlarını bulunuz:

```
a=[4 6 2 8 10];
b=[2 3 1 2 20];
c=a+b
d=a-b
f=a.*b
g=a./b
h=b./a
```

```
f =
    8    18     2    16   200
g =
    2.0000    2.0000    2.0000    4.0000    0.5000
h =
    0.5000    0.5000    0.5000    0.2500    2.0000
```

9) 0 ile 200 arasında değişen eşit aralıklarla dağılan 6 sayı üretiniz

```
linspace(0,200,5)

ans =
    0   40   80  120  160  200
```

9) Aşağıdaki matlab programlarında sonuçları bulunuz.

```
a=[1 3 2 8 6 5 9 7 10];
b=[2 12 3 13 6 14 7 15 8];
c=ismember(a,b)
d=ismember(b,a)

c =
    0    1    1    1    1    0    0    1    0
d =
    1    0    1    0    1    0    1    0    1
```

10) Aşağıdaki kümeler için kesişim, birleşim kümelerini bulunuz ve a'nın b'den farkı ile b'nin a'dan farkını bulunuz.

```
a=[1 3 2 8 6 5 9 7 10];
b=[2 12 3 13 6 14 7 15 8];

c=union(a,b)
d=intersect(a,b)
e=setdiff(a,b)
f=setdiff(b,a)

c =
    1    2    3    5    6    7    8    9   10   12   13   14   15

d =
    2    3    6    7    8
e =
    1    5    9   10
f =
   12   13   14   15
```

----------Problemler:----------

1)

```
a=[2 4 3 8 6 9 7]
```

a) a(2)=?
b) a(4)=?
c) a(3)=5 => a=?

2)
a=[1 4 1 2 1 4 5]
a) b=a(2:4)=?
b) length(a)

3)
a=[1 4 1 2 1 4 5]
b=[2 3 4 1 2 1 3]

a) c=a.^2=?
b) d=3.^b=?
c) e=a.^b=?

4)
a=[8 6 5 7 7 6]
b=[2 6 8 2 3 4]

a) c=union(a,b)=?
b) d=intersect(a,b)=?
c) e=setdiff(a,b)=?
d) f=setdiff(b,a)=?

5)
a=[8 6 5 7 7 6]
b=[2 6 8 2 3 4]

a) c=ismember(a,b)=?
b) d=ismember(b,a)=?

a) a(2)=?
b) a(4)=?
c) a(3)=5 => a=?

2)

a=[1 4 1 2 1 4 5]

a) b=a(2:4)=?
b) length(a)

3)

a=[1 4 1 2 1 4 5]
b=[2 3 4 1 2 1 3]

a) c=a.^2=?
b) d=3.^b=?
c) e=a.^b=?

4)

a=[8 6 5 7 7 6]
b=[2 6 8 2 3 4]

a) c=union(a,b)=?
b) d=intersect(a,b)=?
c) e=

Matrisler, sayıların satırlar ve sütunlar şeklinde sıralanmasından oluşur. Matrisler, birçok mühendislik problemleri çeşidinde kullanılır. Matlab komutları matris işlemlerinin rahat bir şekilde yapılması sağlar.

Matris Girişleri

```
A =
   1   2   3
   4   5   6
```

şeklindeki bir matrisi;

```
» A=[1 2 3
4 5 6]
```

şeklinde veya;

```
» A=[1 2 3;4 5 6]
```

şeklinde girebiliriz.

» A(1,2) komutu bize A matrisinin 1.satırının 2.sütunundaki elemanı verecektir.

```
ans =

   2
```

Matrislerde Matematiksel İşlemler:

```
A =                B =          C =              D =

   1   2   34      1   2        1   2   1        1   1   2
   4   5   6       1   1        3   2   1        2   3   1
                   3   2        1   3   4        4   2   1
```

Olarak Matlab'de giriş yapılmış olsun.

Matris Çarpımı:

İki matrisin çarpımının yapılabilmesi için 1. matrisin sütun sayısının, 2. matrisin satır sayısına eşit olması gerekir. mxn boyutlu bir matris ile nxp boyutlu bir matrisin çarpımından oluşan bir matrisin boyutu mxp olur.

```
» E=A*B

E =
   12   10
   27   25
```

sonucunu verir.

Bir matrisi bir sayı ile çarparsak matrisin her elemanı o sayı ile çarpılır.

E=2*A

```
1
   2   4   6
   8  10  12
```

Matris Toplamı:

İki matrisin toplanabilmesi için satır ve sütun sayılarının eşit olması gerekir

```
» F=C+D

F =
   2   3   3
   5   5   2
   5   5   5
```

sonucunu verir.

Bir matrise bir sayı eklenirse bu sayı matrisin her elemanına eklenecektir

```
F=A+4
      5   6   7
      8   9  10
```

Matris Çıkarması:

```
» G=C-D

G =
   0   1  -1
   1  -1   0
  -3   1   3
```

sonucunu verir.

Bir matristen bir sayı çıkarılırsa bu sayı matrisin her elemanından çıkartılır.

```
G=C-2
G=
      -1   0   1
       2   3   4
```

Matrislerde Karşılıklı Elemanların Çarpılması:

Bu çarpımda iki matriste karşılıklı elemanlar çarpılır. Her iki matrisinde satır ve sütun sayılarının eşit olması lazımdır.

» H = C.*D işlemi

```
H =

   1   2   2
```

```
 6    6
 4    6    4
```

sonucunu verir.

Bir Matrisin Kuvvetinin Alınması:

Bir matrisin 'n.' Kuvvetinin alınması, o matrisin 'n' kez kendisi ile çarpılması demektir. Örneğin C^3=C*C*C demektir.

» I=C^2 veya » I=C*C işlemleri bize

```
I =
     8     9     7
    10    13     9
    14    20    20
```

sonucunu verir.

Bir Matrisin Her Elemanının Kuvvetinin Alınması:

Bir matrisin her elemanının 'n.' kuvvetinin alınması için matris sembolünden sonra "." işaret konur.

» J=D.^3 işlemi

```
J =
     1     1     8
     8    27     1
    64     8     1
```

sonucunu verir.

Matrislerde Bölme:

E = A/C

```
E =
   -0.4000    0.2000    0.8000
   -0.7000    1.1000    1.4000
```

A matrisi C matrisine bölünür.

```
a=[1 2 3 4; 1 3 2 4]
b=[2 1 2 3;1 2 1 2]
c=a./b

c =

    0.5000    2.0000    1.5000    1.3333
    1.0000    1.5000    2.0000    2.0000
```

a matrisinin ... matrisinde satır ve sütun ...

d=a./b

d =

2.0000 0.5000 0.6667 0.7500
1.0000 0.6667 0.5000 0.5000

Bu sefer b matrisinin her elamanını a matrisinin her bir elemanına böler.

Matris İşlemleri:

→**inv(A)**

Bu komut A kare matrisinin tersini bulur

Örnek:

» A=[1 2 3;1 1 2;3 2 2]

A =

1 2 3
1 1 2
3 2 2

» inv(A)

ans =

-0.6667 0.6667 0.3333
1.3333 -2.3333 0.3333
-0.3333 1.3333 -0.3333

→**B'**

Bu komut herhangi bir B matrisinin transpozunu bulur.

Örnek:

» B=[1 2 3; 4 3 1]

B =

1 2 3
4 3 1

» B'

ans =

1 4
2 3
3 1

→**det(C):**

C kare matrisinin determinantını hesaplar:

Örnek:

» C=[1 2 3 4; 1 2 1 2;3 2 1 1;2 3 2 1]

C =

1 2 3 4
1 2 1 2
3 2 1 1
2 3 2 1

» det(C)

ans =

18

Sıfırlar(Zeros) Matrisi:

zeros(N) komutu, boyutu NxN ve elemanları "0" olan kare matris üretir.
zeros(M,N) veya zeros([M,N]) komutları boyutları MxN ve elemanları "0" olan mantisler üretir.

Örnekler:

```
» zeros(3)

ans =

     0     0     0
     0     0     0
     0     0     0
```

```
» zeros(3,2)

ans =

     0     0
     0     0
     0     0
```

```
» zeros([2,3])

ans =

     0     0     0
     0     0     0
```

Birler(Ones) Matrisi:

ones(N) konutu, boyutu NxN ve elemanları "1" olan kare matris üretir.
ones(M,N) veya ones([M,N]) komutları boyutları MxN ve elemanları "1" olan matrisler üretir.
Örnekler:

```
» ones(2)

ans =

     1     1
     1     1
```

```
» ones(2,3)

ans =

     1     1     1
     1     1     1
```

```
» ones([3,2])

ans =

     1     1
     1     1
     1     1
```

Birim Matris:

eye(N) komutu, boyutu NxN olan birim matris üretir.
eye(M,N) veya eye([M,N]) komutları, boyutları MxN, elemanları diagonal boyunca "1", geri kalan yerlerde "0" olan matrisler üretir.

Örnekler:

```
» eye(3)

ans =

     1     0     0
     0     1     0
```

```
» eye(2,3)

ans =

     1     0     0
     0     1     0
```

```
» eye([3,2])

ans =

     1     0
     0     1
     0     0
```

Rasgele Değerler Matrisi:
Bu matrisin elemanları 0 ile 1 arasındaki

```
>> rand(2)

ans =

    0.9501    0.6068
    0.2311    0.4860

>> rand(3,3)

ans =
    0.6154    0.7382    0.9355
    0.7919    0.1763    0.9169
    0.9218    0.4057    0.4103
```

Diagonal Matris:

```
=[4 2 8 5 3]
=diag(a)

=
 4   0   0   0   0
 0   2   0   0   0
 0   0   8   0   0
 0   0   0   5   0
 0   0   0   0   3
```

manları diagonaldan aşağı veya yukarı kaydırmak mümkündür.

```
diag(a,2)

=
 0   0   4   0   0   0   0
 0   0   0   2   0   0   0
 0   0   0   0   8   0   0
 0   0   0   0   0   5   0
 0   0   0   0   0   0   3
 0   0   0   0   0   0   0
 0   0   0   0   0   0   0
```

```
iag(a,-1)

=
 0   0   0   0   0
 0   0   0   0   0
 2   0   0   0   0
 0   8   0   0   0
```

Elemanların sa...
>> magic(3)

```
ans =
     8     1     6
     3     5     7
     4     9     2
```

pascal(n) matrisi: Bu matris bize pascal katsayılarını üretir.
>> pascal(3)

```
ans =
     1     1     1
     1     2     3
     1     3     6
```

Matris Değişiklikleri

a=[4 2 3 5;1 7 2 3;2 1 0 6]

```
a =
     4     2     3     5
     1     7     2     3
     2     1     0     6
```

>> a(2,4)=1

```
a =
     4     2     3     5
     1     7     2     1
     2     1     0     6
```

a matrisinin 2. satırının 4. sütununa 1 atanır.

>> a(3,6)=2

```
a =
     4     2     3     5     0     0
     1     7     2     1     0     0
     2     1     0     6     0     2
```

>> a(:,2)=5

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

a matrisinin 2. sütununun bütün değerlerine 5 atanır.

>> a(:,3)=[1;2;3]

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

Matrisler

a matrisinin 3. sütununa 1,2,3 değerleri atanır

Bir matrisin belli satır ve sütunlarından yeni bir matris oluşturmak mümkündür.

a=[4 2 3 5 1 7 2 3 2 1 0 6]

a =

4	2	3	5
1	7	2	3
2	1	0	6

b = a(1:3,2:3)

b=

2	3
7	2
1	0

Bu komut ile a matrisinin 1-3 arası satırlar ve 2-3 arası sütunlar seçilir.

Bir matrisin belli bir satırını seçmek mümkündür.

b=a(2,:)

b = 1 7 2 3

a matrisinin 2. satırını seçer

b=a(2:-1:1,4:-1:2)

b =

3	2	7
5	3	2

önce 2. satırı sonra 1. satırı alır, ve sütunları 4'den 2'ye doğru ters sıralar.

>> c=[a([1 3],:) b(:,[2 3])]

c =

4	2	3	5	2	7
2	1	0	6	3	2

a matrisinin 1. ve 3. sütunlarını al, b matrisinin 2. ve 3. sütunlarını al ve c matrisi olarak birleştir.

Bir matrisin satır ve sütun sayısını değiştirmek mümkündür. Yalnız bu işlem, eleman sayısı değişmeyecek şekilde olmalıdır.

>> b=reshape(a,2,6)

b =

4	2	7	3	0	3
1	2	1	2	5	6

```
b =
    4    7    0
    1    1    5
    2    3    3
    2    2    6
```

```
b=a
b =
    4    2    3    5
    1    7    2    3
    2    1    0    6
```

a matrisinin içeriği b matrisine kopyalanmış olur.
Bir matrisin satır veya sütununu silmek mümkündür.

```
>> b(:,3)=[]
b =
    4    2    5
    1    7    3
    2    1    6
```

b matrisinin 3. sütunu silinmiş olur.

```
>> b=a(:,[2 3 2])

b =
    2    3    2
    7    2    7
    1    0    1
```

a matrisinin önce 2. sütunu alınır, sonra 3. sütunu alınır, tekrar 2 . sütunu alınır ve b matrisine atanır.

```
>> repmat(a(:,2),1,3)
ans =
    2    2    2
    7    7    7
    1    1    1
```

a matrisinin 2. sütunu alınır ,1x3 boyutunda tekrarlanılır ve yeni matris elde edilir.

→**Flipud(x)** komutu matrisin üst ve alt satırlarının yerini değiştirir.

```
>> b=[1 2 3 ;4 5 6;7 8 9;10 11 12;6 2 1]
>> flipud(b)
b =
    1    2    3
    4    5    6
    7    8    9
   10   11   12
    6    2    1
```

```
6   2   12
10  11  9
7   8   9
4   5   6
1   2   3
```

→**fliplr(x)** komutu matrisin sağ ve sol sütunlarının yerlerini değiştirir.

```
>> fliplr(b)

ans =
   3   2   1
   6   5   4
   9   8   7
  12  11  10
   1   2   6
```

```
c=[ 2 3 5;1 4 6;5 8 6]
diag(c)

c =
   2   3   5
   1   4   6
   5   8   6
ans =

   2
   4
   6
```

→**diag(x)** komutu verilen matrisin diagonal elemanlarını verir.

```
>> triu(c)

ans =

   2   3   5
   0   4   6
   0   0   6
```

→**triu(x)** komutu matrisin üst üçgen kısmını alır, diğer kısma sıfır atar.

```
>> tril(c)

ans =
   2   0   0
   1   4   0
   5   8   6
```

```
g=kron

d =
   1
   0
e =
   2
   4
f =
```

f=kron(d,e)
g=kron(e,d)

```
d =
   1   2
   0   1
e =
   2   3
   4   1
f =
   2   3   4   6
   4   1   8   2
   0   0   2   3
   0   0   4   1
g =
   2   4   3   6
   0   2   0   3
   4   8   1   2
   0   4   0   1
```

Burada kron = (d,e) komutu şuna eşittir:
[1*e 2*e; 0*e 1*e]

kron(e,d)'de şuna eşittir.
[2*d 3*d;4*d 1*d]

Matris Boyutunun Bulunması

→**size(x)** komutu matrisin satır ve sütun sayısını verir.

```
a=[1 4 2 5; 2 5 6 3;8 6 9 7]
size(a)

a =
   1   4   2   5
   2   5   6   3
   8   6   9   7

ans =
   3   4
```

satır ve sütun sayısı ayrı ayrı da elde edilebilir.

```
>> [satir sutun]=size(a)
```

satır =
3

sutun =
4

→**numel (x)** komutu, x matrisi içindeki toplam eleman sayısını verir.

>> numel(a)

ans =

12

→**length(x)** komutu, x matrisinde satır veya sütundan hangisi uzun ise onun uzunluğunu verir.

>> length(a)

ans =
4

>> b=[1 2;3 4;5 6;7 8;9 10]
b =
1 2
3 4
5 6
7 8
9 10
>> length(b)

ans =
5

Denklem Çözümlerinde Matrislerin Kullanılması

Elimizde şöyle bir denklem bulunmuş olsun.

2x+3y+4z=54
4x+5y+2z=62
x+4y+6z=72

Bu denklemde x,y,z değerlerinin bulunulması isteniyor.
Bu denklemi önce aşağıdaki gibi matris formuna sokalım.

$$\begin{bmatrix} 2 & 3 & 4 \\ 4 & 5 & 2 \\ 1 & 4 & 6 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 54 \\ 68 \\ 72 \end{bmatrix}$$

Birinci matrise A, 2.matrise u, 3.matrise B dersek elimizde

1) u=i
2) u=

Şimdi son

A=[2 3 4
B=[54;6
u1=inv(
u2=A\B

A =
2
4
1
B =
54
68
72

u1 =
2.
10
5

u2

1) u=inv(a).B
2) u=A\B

imdi sorumuzu Matlab'e aktaralım ve çözümü bulalım.

```
A=[2 3 4;4 5 2;1 4 6]
B=[54;68;72]
u1=inv(A)*B
u2=A\B

A =
   2   3   4
   4   5   2
   1   4   6
B =
   54
   68
   72

u1 =
   2.0000
   10.0000
    5.0000

u2 =
    2
   10
    5
```

Bazı Matris Çeşitleri

→ Companion matrisi:

compan(a) : Bu matrisin ilk satırı -a(2:n)/a(1) olacak şekilde değerler alır.

```
a=[2 4 0 -6 1]
b=compan(a)

a =
    2    4    0   -6    1

b =
   -2.0000         0    3.0000   -0.5000
    1.0000         0         0         0
         0    1.0000         0         0
         0         0    1.0000         0
```

→Hadamard matrisi:

hadamard(n): Bu matris 1ve -1 lerden oluşur ve n*n boyutundadır. Bu matris $H'*H = n*I$ eşitliği sağlar. Burada n, kare matrisin boyutudur.

Kullanımda n, n/12 ve n/20 2'nin kuvveti olmalıdır

```
hadamard(8)
ans =
     1     1     1     1     1     1     1     1
     1    -1     1    -1     1    -1     1    -1
     1     1    -1    -1     1     1    -1    -1
     1    -1    -1     1     1    -1    -1     1
     1     1     1     1    -1    -1    -1    -1
     1    -1     1    -1    -1     1    -1     1
     1     1    -1    -1    -1    -1     1     1
     1    -1    -1     1    -1     1     1    -1
```

→ **Hankel matrisi:**

hankel(a): Bu komut ilk sütunu a olan, ve ters diagonalden aşağısı sıfır olan bir matris üretir.

```
b=[1 2 3 4 5]
hankel(c)

b =

     1     2     3     4     5


ans =

     1     2     3     4     5
     2     3     4     5     0
     3     4     5     0     0
     4     5     0     0     0
     5     0     0     0     0
```

H = hankel(c,r) matrisinin ilk sütunu c'ye, son sütunu r'ye eşittir. c'nin son elemanı r'nin ilk elemanından farklı ise c'nin son elemanı seçilir r'nin son elemanı seçilmez. Hankel matris ters diagonaller boyunca simetrik ve sabittir. H(i,j)=p(i+j-1)'dir. Burada p=[c r(2:end)]'dir.

```
a=[6 7 5 9]
b=[2 3 1]
c=hankel(b,a)

a=
     6     7     5     9
```

→Hadamard matrix

hadamard(n): Bu matris 1ve -1'lerden oluşur ve
eşitliği sağlar: H*H' = n*I Burada n, kare matrisin boyutudur.
Kullanımda n, n/12 ve n/20 2'nin kuvveti olmalıdır.

```
hadamard(8)
ans =
     1     1     1     1     1     1     1     1
     1    -1     1    -1     1    -1     1    -1
     1     1    -1    -1     1     1    -1    -1
     1    -1    -1     1     1    -1    -1     1
     1     1     1     1    -1    -1    -1    -1
     1    -1     1    -1    -1     1    -1     1
     1     1    -1    -1    -1    -1     1     1
     1    -1    -1     1    -1     1     1    -1
```

→ Hankel matrisi:

hankel(a): Bu komut ilk sütunu a olan, ve ters diagonalden aşağısı sıfır olan bir matris üretir.

```
b=[1 2 3 4 5]
hankel(c)

b =

     1     2     3     4     5


ans =

     1     2     3     4     5
     2     3     4     5     0
     3     4     5     0     0
     4     5     0     0     0
     5     0     0     0     0
```

H = hankel(c,r) matrisinin ilk sütunu c'ye, son sütunu r'ye eşittir. c'nin son elemanı r'nin
elemanından farklı ise c'nin son elemanı seçilir r'nin son elemanı seçilmez. Hankel mat
ters diagonaller boyunca simetrik ve sabittir. H(i,j)=p(i+j-1)'dir. Burada p=[c r(2:end)]'dir.

```
a=[6 7 5 9]
b=[2 3 1]
c=hankel(b,a)

a=
     6     7     5     9
```

```
b =
    2    3    1
c =
    2    3    1    7
    3    1    7    5
    1    7    5    9
```

→ **Hilbert matrisi:**

hilb(n) komutu Hilbert matrisini oluşturur. Bu martis nxn boyutundadır. Hilbert matrisinin elamanları şu şekilde hesaplanır : H(i, j) = 1/(i+j-1).

```
hilb(3)

ans =
    1.0000    0.5000    0.3333
    0.5000    0.3333    0.2500
    0.3333    0.2500    0.2000
```

→ **invhilb(n):** Bu komut Hilbert matrisinin tersini üretir. (n<15) için kesin sonuçlar bulunur. Daha büyük n değerleri için yuvarlama yapılır.

```
a=hilb(2)
b=invhilb(2)
c=a*b
a =
    1.0000    0.5000
    0.5000    0.3333
b =
     4    -6
    -6    12
c =
     1     0
     0     1
```

Bazı Özel Matris Komutları:

→ **any(A):** A matrisinin sütunlarındaki elamanlarından biri bile sıfır değilse 1 üretir. Aksi halde 0 üretir.

```
A=[0 3  0; 2 0 0; 0 4 0]
B=any(A)

A =

     0     3     0
     2     0     0
     0     4     0
```

B =

1 1 0

→ **B=all(A):** A matrisinin sütunlarındaki elemanlarının hepsi 0 ise 1 üretir aksi halde 0 üretir.

```
A=[0 3  0; 2 5 0; 0 4 0]
B=all(A)
```

A =

0 3 0
2 5 0
0 4 0

B =

0 1 0

→ **find(A):** A matrisinde 0 olmayan elemanların indislerini belirler. Sayma işlemi sütunlar boyunca yapılır.

```
A=[0 3  0; 2 5 0; 0 4 0]
B=find(A)
```

A =

0 3 0
2 5 0
0 4 0

B =

2
4
5
6

→ **isnan(A)**: A matrisinin elemanları "NaN" ise 1 üretir aksi halde 0 üretir.

```
A=[0 3  0/0; 2 5 0/0; 0/0 4 0]
B=isnan(A)
```

A =

0 3 NaN
2 5 NaN
NaN 4 0

B =

```
     0     0     1
     0     0     1
     1     0     0
```

→ **finite(A)**: A matrisinin elemanları "NaN" veya sonsuz ise sıfır, sonlu sayı veya sıfır ise 1 üretir.

```
A=[1/0 3  0/0; 2 5 0/0; 0/0 4 1/0]
B=finite(A)
```

A =

```
   Inf     3   NaN
     2     5   NaN
   NaN     4   Inf
```

B =

```
     0     1     0
     1     1     0
     0     1     0
```

→**rank(a)** komutu verilen matrisin rankını hesaplar.

```
>> a=[1 2 3;5 6 8;1 2 6]
a =
     1     2     3
     5     6     8
     1     2     6

>> rank(a)
ans =

     3
```

→ **trace:** Bu komut diagonal boyunca elemanları toplar.

```
>> a=[3 7 2; 1 4 5; 2 5 6]

a =
     3     7     2
     1     4     5
     2     5     6
```

```
>> b=trace(a)

b =
   13
```

Alıştırmalar:

1) Aşağıdaki matrisi, direkt olarak veri girişi yapmadan bir döngü içinde elde ediniz

```
b =

     2     8    14
     4    10    16
     6    12    18
```

çözüm:

```
b=eye(3);
for i=2:2:18
   b(i/2)=i;
end
b
```

```
b =

     2     8    14
     4    10    16
     6    12    18
```

2) Aşağıdaki matrisi bir diziden elde ediniz.

```
     0     1     0     0
     0     0     2     0
     0     0     0     3
     0     0     0     0
```

cevap:

```
a=[1 2 3];
diag(a,1)
```

3) Aşağıdaki matrisi bir diziden elde ediniz.

```
a=[2 4 -6 8];
```

```
-2   3   4
 1   0   0
 0   1   0
```

4)
Aşağıdaki a matrisinden b sütununu elde ediniz

```
a =

     0     0     0
     4     0     5
     2     0     0


b =

     2
     3
     8
```

cevap:

```
a=[0 0 0;4 0 5;2 0 0 ]
b=find(a)
```

5)
Aşağıdaki A matrisinden b dizisini elde ediniz.

```
a =

     0     0     0
     4     0     5
     2     0     0


b =

     1     0     1
```

çözüm:

```
a=[0 0 0;4 0 5;2 0 0 ]
b=any(a)
```

6) Aşağıdaki matrisi Matlab de tek komutla elde ediniz.

```
 1   0   0
 0   1   0
```

cevap:

```
eye(2,3)
```

7) Aşağıdaki matrisi tek komutla elde ediniz.

```
    1     1/2   1/3   1/4
   1/2    1/3   1/4   1/5
   1/3    1/4   1/5   1/6
   1/4    1/5   1/6   1/7
```

cevap

```
hilb(4)
```

8) a ve b matrisleri aşağıda verilmiştir. A matrisinden b matrisini elde ediniz.

```
a =

     1     1     4
     2     1     3
     2     2     5


b =

     5     5     8
     6     5     7
     6     6     9
```

cevap:

```
a=[1 1 4; 2 1 3; 2 2 5]
b=a+4
```

9) 4x4'lük bir matris oluşturunuz ve matrisin satır,sütun ve diegonal toplamı birbirine eş olsun.

```
>> magic(4)

ans =

    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

10)

a ve b matrisleri aşağıda verilmiştir. A matrisinden b matrisini elde ediniz.

```
b=fliplr(a)
```

11) Aşağıdaki denklemi matlab de çözünüz.

$$x + 2y + 3z = 17$$
$$2x + y + 4z = 22$$
$$x + 3y + 5z = 25$$

cevap:

```
a=[1 2 3; 2 1 4;1 3 5]
b=[17 22 25]'
sonuc=inv(a)*b
```

12) a matrisi aşağıda verilmiştir. a matrisinden 2. satırı siliniz.

```
a=[1 2 3; 2 1 4;1 3 5;3 7 6]
a(2,:)=[]
```

a =

1	2	3
1	3	5
3	7	6

13) a ve b matrisleri aşağıda verilmiştir. a matrisinden, b matrisini elde ediniz.

a =

1	2	3
2	1	4
1	3	5
3	7	6

b =

2	1	4
1	2	3
3	5	6

cevap

```
a=[1 2 3; 2 1 4;1 3 5;3 7 6]
b=a([2 1 4],:)
```

sorunun cevabında önce a matrisinin 2. satırı, sonra1. satırı en sonunda da 4. satırı alınarak
matrisi elde edilir.

14) a ve b matrisleri aşağıda verilmiştir. a matrisinden b matrisini elde ediniz.

a =

1	2	3
2	1	4
1	3	5
3	7	6

b =

1	1	2	3	3	5
2	3	1	7	4	6

cevap:

```
a=[1 2 3; 2 1 4;1 3 5;3 7 6]
b=reshape(a,2,6)
```

15) a ve b matrisleri aşağıda verilmiştir. a matrisinden b matrisini elde ediniz.

a =

1	2	4
2	3	1
5	6	2

b =

5	6	2

```
a=[1 2 4 ; 2 3 1; 5 6 2]
b=repmat(a(3,:),3,1)
```

Problemler:

1)

a =

1	2	4
2	3	1
5	6	2

b =

2	8	5
1	4	2
6	7	4

a ve b matrisleri yukarıda verilmiştir. Buna göre:

a) a*b = ?
b) a.*b = ?
c) a+b = ?
d) a-b = ?
e) a/b = ?
f) a./b = ?
g) a+4 = ?
h) 2*b = ?
i) a^2 = ?
j a.^2 = ?

2)

a =

1	2	4
2	3	1
5	6	2

b =

2	8	5
1	4	2

a ve b matrisleri yukarıda verilmiştir. Buna göre;

a) inv(a) = ?
b) det(a) = ?
c) b' = ?

3) Aşağıdaki

a) eye(3) = ?
b) eye(2,3) = ?
c) ones(2) = ?
d) ones(2,4) = ?
e) zeros(5) = ?
f) zeros(2,4) = ?

4) Aşağıdaki matrisleri bulunuz

a) rand(3,2) = ?
b) pascal(5) = ?
c) magic(4) = ?

5)

a =

2	1	3
6	2	3
1	4	6

a matrisi yukarıdaki gibi verildiğine göre aşağıdaki soruları cevaplandırınız.

a) a(2,:)=4
b) a(1,:)=[2 6 8]
c) a(1:2,2:3) = ?
d) a(3:-1:2,1:2) = ?

6)

a =

2	0	3
6	0	0
1	0	0

a matrisi yukarıdaki gibi verildiğine göre aşağıdaki işlemleri yapınız.

a) any(a) = ?
b) all(a) = ?
c) find(a) = ?

3) Aşağıdaki

a) eye(3) = ?
b) eye(2,3) = ?
c) ones(2) = ?
d) ones(2,4) = ?
e) zeros(5) = ?
f) zeros(2,4) = ?

4) Aşağıdaki matrisleri bulunuz

a) rand(3,2) = ?
b) pascal(5) = ?
c) magic(4) = ?

5)

a =

2	1	3
6	2	3
1	4	6

a matrisi yukarıdaki gibi verildiğine göre aşağıdaki soruları cevaplandırınız.

a) a(2,:)=4
b) a(1,:)=[2 6 8]
c) a(1:2,2:3) = ?
d) a(3:-1:2,1:2) = ?

6)

a =

2	0	3
6	0	0
1	0	0

a matrisi yukarıdaki gibi verildiğine göre aşağıdaki işlemleri yapınız.

a) any(a) = ?
b) all(a) = ?
c) find(a) = ?

Problemler 145
KONTROL KOMUTLARI 146
IF komutu 147
FOR komutu 148
WHILE komutu 149
SWITCH komutu 152
Alıştırmalar 153
Problemler 153
KOORDİNAT SİSTEMLERİNİN BİRBİRİNE ÇEVRİLMESİ 154
cart2pol 154
cart2sph 155
pol2cart 156
Alıştırmalar 15
Problemler 15
İNTERPOLASYON 15
İnterp1 15
Alıştırmalar 15
Problemler 1
KARAKTER DİZİSİ KOMUTLARI 1
Strcmp 1
Strncmp 1
İsletter 1
İsspace 1
Strrep 1
Findstr 1
Strvcat 1
Strmatch 1
Strcat 1
Mat3str 1
Eval 1
Deblank 1
Lower 1
Strjust 1
Strjust 1

```
>> b=70;
>> fprintf('Ben %f kiloyum.',b)
Ben 70.000000 kiloyum.

>> c=80;
>> fprintf('Arabayi %4.2f km. hizla\n sürüyordu',c)
Arabayi 80.00 km. hizla
Sürüyordu



>> b=70;
>> fprintf('Ben %d kiloyum.',b)
Ben 70 kiloyum.

>> q='pazartesi';
>> fprintf('Bugün günlerden %s.',q)
Bugün günlerden pazartesi.


d=4250;
fprintf('Dagin yuksekligi %e metredir',d)

Dagin yuksekligi 4.250000e+003 metredir


a=4250;b=5674;c=1234;d=2346;
fprintf('%10.2f\n',a/4)
fprintf('%10.2f\n',b/4)
fprintf('%30.4f\n',c/6)
fprintf('%30.4f\n',b/6)

   1062.50
   1418.50
                  205.6667
                  945.6667
```

Alıştırmalar

1) Ekrandan sıra ile 4 sayı girilmesini isteyiniz ve bu sayıları yazdırınız.

```
clc
for i=1:4
a(i)=input('sayi giriniz.: ') ;
end
a
```

2) Ekr
hane olacak şekilde yazdırınız.

```
for i=1:3
b(i)=input('sayi giriniz.: ') ;
end
fprintf('%6.2f',b)
```

3) 1'den 10'a kadar sayıları ve karelerini yan yana yazdırınız.

```
i=1:10;
j=i.^2;
for k=1:10
   sprintf('%10d %20d',i(k),j(k))
end
```

Sayı Formatının Düzenlenmesi

Matlab de sayının formatını değiştirmek mümkündür.

```
x = [2/5   5.6     7.23e-4]

>> format short

x =    0.4000    5.6000    0.0007


>> format short e

x =   4.0000e-001  5.6000e+000  7.2300e-004

>> format short g

x =   0.4         5.6     0.000723

>> format long

x =   0.40000000000000   5.60000000000000   0.00072300000000

>> format long e

x =  4.000000000000000e-001    5.600000000000000e+000    7.230000000000000e-04

>> format long g

x =  0.4                    5.6                 0.000723

>> format bank
```

```
x =
    0.40        5.60

>> format rat
>> x
x =  2/5        28/5        8/11065

>> a=[10 20 -30 40 -50 24 56];
>> format +
>> a
a =
++-+-++
```

Problemler

1) 100000 sayısını üstel olarak yazdırınız.

2) 4546.678 sayısını virgülden sonra iki hane olacak şekilde yazdırı

3) Ekrandan bir sayı girilmesini isteyiniz. Bu sayının karesini alın v

4) 7890.234 sayısını tamsayı olarak yazdırınız.

5) 8909.3663637 sayısını virgülden sonra dört hane olacak şekilde

6) 78/12, 789/25, 123/45, 678/45 sayılarından maksimum olanı b
hane olacak şekilde ekrana yazdırınız.

7) 0.75 2.5 3.2 sayılarını kesirli olarak yazdırınız.

8) 23 45 -20 12 30 -40 sayılarını format + komutunu kullanarak

BİLGİ ANALİZİ VE İSTATİSTİK

Elimizde mevcut bulunan bilginin analiz edilmesi ve onunla ilgili istatistiki bilgi çıkarmamız hemen hemen hepimiz için önemli bir konudur. Matlab de data analizi ve istatistik için bulunan komutlar mevcuttur. Bu komutlar ile elde mevcut bulunan verilerin analizi kolayca yapılabilmektedir. Elimizde bulunan istatistik bilgilerden yararlanarak olayları yorumlayabiliriz. Kitabın bu bölümde data analizi ve istatistikte kullanılan komutlar anlatılmıştır.

Elimizde

A=[1 2 5 3 8],
C=[4 5 1 2 7] dizileri ve
B= [1 3 8 9
2 5 4 1] matrisi olsun.

Matlab ile eldeki veriler üzerinden sonuçlar elde etmek mümkündür.
max(A) komutu bize A dizisi içindeki maksimum sayıyı verecektir.

```
max(A)
ans =

   8
```

max(B) komutu ile matrisin her sütunundaki en büyük elemanlarından oluşan bir vektör elde edilir.

```
>> max(B)
ans =
   2    5    8    9
```

max(A,C) komutu iki vektörün karşılıklı elemanlarını karşılaştırır ve büyük elemanlarını alarak yeni bir vektör oluşturur.

```
>> max(A,C)
ans =
   4    5    5    3    8
```

min(A) komutu bize A dizisi içindeki minimum sayıyı verecektir.

```
>> min(A)
ans =
   1
```

min(B) komutu ile matrisin her sütunundaki en küçük elemanlarından oluşan bir vektör elde edilir.

```
>> min(B)
ans =
   1    3    4    1
>> min(A,C)
ans =
   1    2    1    2    7
```

sort(C) komutu C vektörünü artan düzen içinde yeniden düzenler.

```
>> sort(C)
ans =
```

1 2 4 5

bir vektörü büyükten küçüğe şu şek

D=sort(C),

F=D(end -1 1)

E = 7 5 4 2 1

[x,y]=sort(C) komutu bize vektörün düzenlenmiş halini ve y'de bize vektörün eski vektö

içindeki sıralamasını verir.

```
>> [x,y]=sort(C)
x =    1    2    4    5    7
y =    3    4    1    2    5
```

sort(B) komutu B matrisindeki her sütunu yukarıdan aşağı artan biçimde düzenler.

```
>> sort(B)

ans =
   1   3   4   1
   2   5   8   9
```

sum(A) komutu A vektörünün bütün elemanlarını toplar.

```
>> sum(A)
ans =
   19
```

sum(B) komutu bir matriste, her sütun boyunca sayıları toplar ve yeni bir vektör üretir

```
>> sum(B)
ans =
   3   8   12   10
```

prod komutu:

Bu komut vektör içindeki bütün elemanları birbiri ile çarpar. Matriste de her sütun boyunca elemanları çarpar.

```
>> A=[1 5 4 2];
>> prod(A)

ans =
   40

>> B=[1 2 3; 6 2 4; 5 3 4];
>> prod(B)

ans =
   30   12   48
```

cumsum komutu:

Bu komut bir vektör için kullanılırsa, vektörde her eleman ,kendi ve önceki elemanlarının toplamının değerini alır. Eğer elimizde bir matris varsa her satır kendisi ve d önceki satırların toplamıyla yer değiştirir.

A =

7 4 5
8 6 6

cumprod komutu: Bu komut bir vektör için kullanılırsa, vektörde her eleman ,kendi ve önceki elemanlarının çarpımının değerini alır. Eğer elimizde bir matris varsa her satır kendis ve daha önceki satırların çarpımıyla yer değiştirir.

```
A =

   3   2   4   1   5

>> cumprod(A)

ans =

   3   6  24  24 120

B =

   1   3   2
   2   1   3
   1   1   1
   2   4   1

>> cumprod(B)

ans =

   1   3   2
   2   3   6
   2   3   6
   4  12   6
```

mean(A) komutu bir vektördeki elemanların ortalamasını bulur.

```
>> mean(A)

ans =
   3.8000
```

```
>> cumsum(A)
ans =
    1    5   14   21   24   25
B =
    2    1    3
    5    3    2
    1    2    1
>> cumsum(B)
ans =
    2    1    3
    7    4    5
    8    6    6
```

cumprod komutu: Bu komut bir vektör için kullanılırsa, vektörde her eleman ,kendi ve önceki elemanlarının çarpımının değerini alır. Eğer elimizde bir matris varsa her satır kendisi ve daha önceki satırların çarpımıyla yer değiştirir.

```
A =

    3    2    4    1    5

>> cumprod(A)

ans =

    3    6   24   24  120

B =

    1    3    2
    2    1    3
    1    1    1
    2    4    1

>> cumprod(B)

ans =

    1    3    2
    2    3    6
    2    3    6
    4   12    6
```

mean(A) komutu bir vektördeki elemanların ortalamasını bulur.

```
>> mean(A)

ans =
    3.8000
```

mean(B) komutu ile matrisin her sütun boyunca elemanların ortalaması alınır ve yeni bir vektör üretilir.

```
>> mean(B)

ans =

   1.5000   4.0000   6.0000   5.0000
```

geomean(A) komutu bir dizideki elamanların geometrik ortalamasını bulur. n sayının geometrik ortalaması (sayıların çarpımı) $^{(1/n)}$ şeklindedir.

```
A =

   1   2   6   5   4

>> geomean(A)

ans =
   2.9926
```

median komutu:

median komutu bir vektörü kendi içinde sıraya dizer ve ortadaki elamanı seçer. Eğer vektörün eleman sayısı çift ise sıraya dizilmiş elemanların ortadaki iki tanesinin ortalamasını alır. Bir matris için de yukarıdaki işlemleri matrisin sütunları boyunca yapar.

```
>> A=[1 5 3 9 4 7 8];
>> median(A)

ans =

   5

>> B=[8 7 8 9 5 4];
>> median(B)

ans =

   7.5000

>> C=[1 2 3;5 2 4; 2 4 2];
>> median(C)

ans =

   2   2   3

>> D=[1 3 2 4;2 5 3 2;3 1 5 2; 6 2 1 3]
```

EKRANDAN BİLGİ ALMA VE EKRANA YAZMA

Matlab programında ekrandan sorgu yapmak, ekrandan alınan değerler üstünde işlem yapmak ve verileri istediğimiz formatta ekrana yazdırmak mümkündür.

→ clc: Bu komut ekranı temizler.
→ a=input('b'): Bu komut ekrana bir soru sorar. Aldığı cevabı da bir değişkene atar.

Örnek:

```
clc
a=input('yasinizi giriniz .: ')

yasinizi giriniz .: 20

a =
   20
```

Eğer soru kısmına '\n' eklenirse Matlab veriyi sorunun bir alt satırından alır.

Örnek:

```
>> b=input('kilonuzu giriniz\n')
kilonuzu giriniz
80

b =
   80
```

→ disp('b'): disp komutu ekrana bilgi yazdırmak için kullanılır.

```
>> a=48;
>> disp(a)
   48

>> disp(' Kitabini getirdin mi?')
 Kitabini getirdin mi?
```

→ fprintf(format, değer): Bu komut, ekrana yazdırma sırasında kontrol imkanı verir. Yazılacak değerin yeri belirtilebilir, değerin formatı değiştirilebilir. Satırbaşı yapılabilir. Komut içinde %e, %f, %g ve \n kullanılabilir. %e kullanılırsa değer eksponansiyel biçiminde yazılır, %f kullanılırsa değer kesirli şekilde yazılır. %g kullanılırsa %e veya %g'den biri seçilir.(Hangisi daha kısa ise).tamsayılar için %d kullanılır.. Karakter dizileri için %s kullanılır. \n kullanılırsa satır başı yapılır

```
>> a=20
>> fprintf('Senin yasin.: %f',a)
```

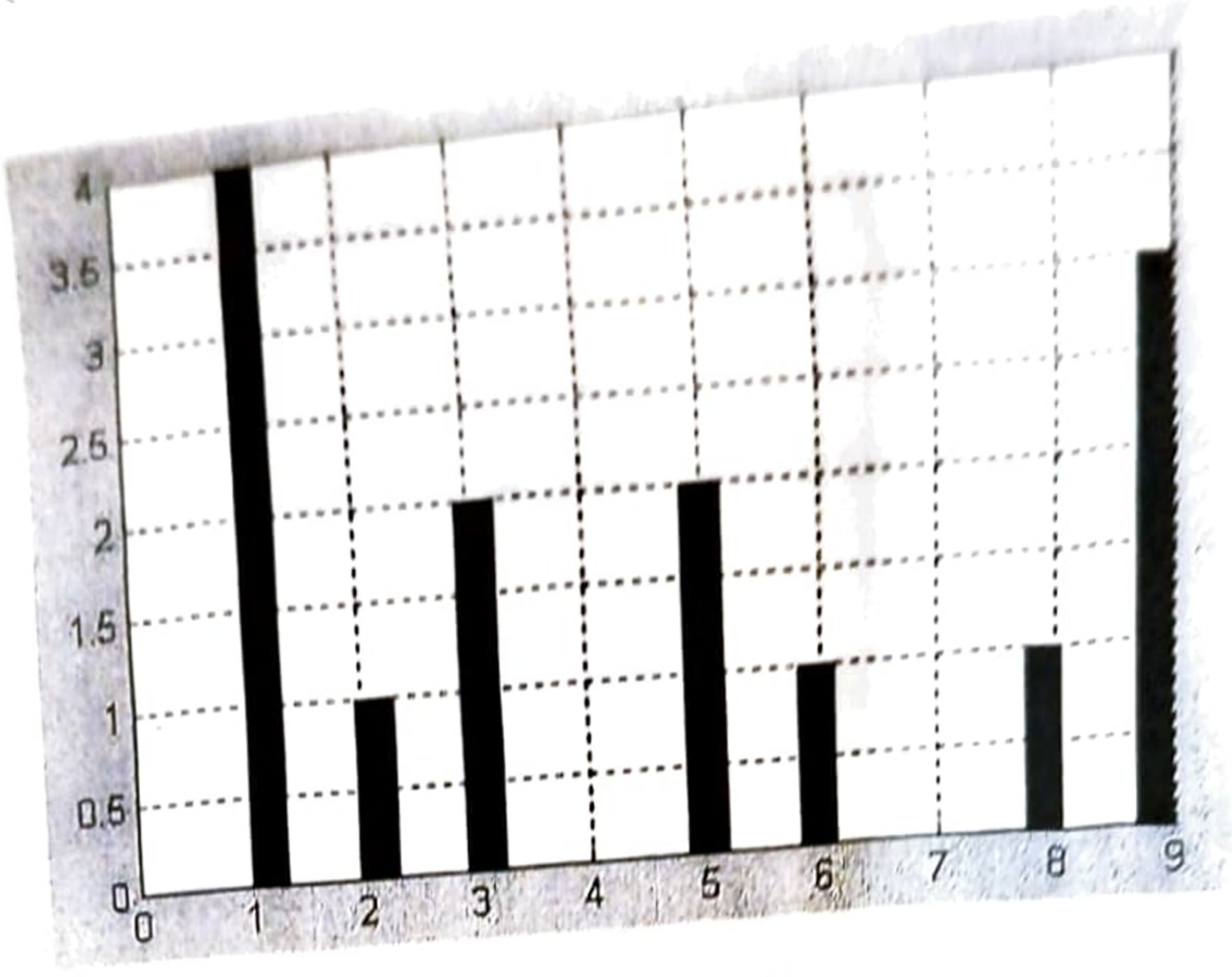
→ **factor(n)**: Bu komut verilen sayıyı asal çarpanlarına ayırır.

```
>> factor(785)

ans =
    5   157

>> factor(1155)

ans =

    3    5    7   11
```

→ **perms(u)**: Bu komut verilen vektörün mümkün olan tüm değişik sıralamaları

```
u=7:9
a=perms(u)
a =
    7    8    9
    7    9    8
    8    7    9
    8    9    7
    9    8    7
    9    7    8

>> perms(2:2:4)
ans =

    2    4
    4    2
```

→ **primes(n)** komutu n sayısına kadar olan asal sayıları verir.



Rastgele Sayılar

→ **rand(n)** komutu; n satırlık ve n sütunluk ve değerleri 0 ile 1 arasında değişen bi
üretir.

```
>> rand(2)

ans =

   0.9355   0.4103
   0.9169   0.8936
```

→ rand(m,n) komutu; m satırlık, n sütunluk ve değerleri 0 ile 1 arasında değişen bi
üretir.

```
>> rand(2,3)

ans =

   0.0579   0.8132   0.1389
   0.3529   0.0099   0.2028
```

```
cevap
>> max(a)

ans =
   12

>> min(a)

ans =
    3
```

3) Aşağıdaki dizini büyükten küçüğe ve küçükten büyüğe doğr

```
a =
    4    8    5    6    3   12    6    7
```

cevap:

```
>> sort(a)

ans =
    3    4    5    6    6    7    8   12

>> b=sort(a);
>> c=b(end:-1:1)

c =
   12    8    7    6    6    5    4    3
```

4) Aşağıdaki dizideki elemanların toplamını ve çarpımın

```
a =
    2    4    3    6    5    1
```

```
>> var(A)

ans =

    4.0000    8.3333    1.3333

>> B = [2 1 3 2 7 5 2 3 8]

B =

     2     1     3     2     7     5     2     3     8

>> var(B)

ans =

     6
```

→ **std komutu:**

Bu komut verilen bir vektörün st

verilmiş ise matrisin sütunları boyunca standart

```
A =

     1     2     3
     3     2     1
     5     6     1
```

Cevap:

```
a=[1 2 3 2 2 4 2]
harmean(a)

ans =
    2.1538
```

8) 9 ve 16 sayılarının geometrik ortalamasını bulunuz.

Cevap:

```
geomean([9 16])

ans =
    12
```

9)

```
a =
    3    1    4    9    6    8    7    2   10   13   12
```

yukarıdaki dizinin sıralanmış halinin ortadaki elemanını bulunuz.

Cevap:

```
a=[3 1 4 9 6 8 7 2 10 13 12]
median(a)
```

10) 374 ve 525 sayılarının 1'den farklı çarpanlarını bulunuz.

Cevap:

```
factor(374)

ans =
    2   11   17

factor(525)

ans =
    3    5    5    7
```

11) aşağıdaki dizinin elemanlarının mümkün olan tüm değişik sıralamalarını veriniz.

```
    3    5    7
```

Cevap:



ans =

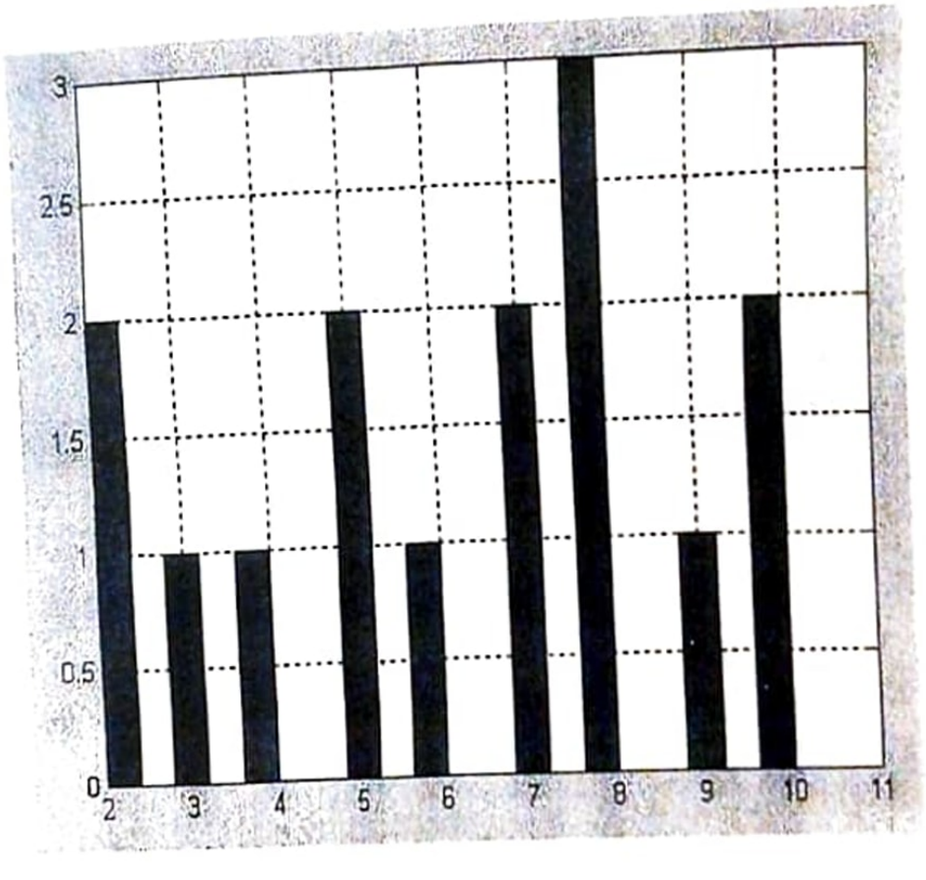
```
3   5   7
3   7   5
5   3   7
5   7   3
7   5   3
7   3   5
```

12)

```
a =
   10    8    2    5    7    3    2    4    7    8   10    8    6    9    5
```

Yukarıdaki dizide sayıların dağılımı bulunuz ve grafikle gösteriniz.

```
a=[10 8 2 5 7 3 2 4 7 8 10 8 6 9 5]
hist(a,20)
grid
```



Problemler

1)

```
a =
   10    8    2
    5    7    3
    2    4    7
```

Yukarıda verilen a matrisi için her sütunun maksimum ve minimum elemanlarını bulunuz

2 1 3 2
6 7 3 4
1 2 3 4

3)

a =

1 8 4 5 2 7 3

Yukarıda verilen dizi için şu işlemleri yapınız.

a) sum(a)
b) prod(b)
c) cumsum(a)
d) cumprod(a)

4) Aşağıda verilen dizinin varyansını ve standart sapmasını hesaplayınız.

a =

6 5 7 8 6 7 6 4 7

5) Aşağıdaki sayıların çarpanlarını bulunuz.

a) 125
b) 378
c) 588

6) Aşağıdaki dizinin elemanlarını değiştirerek mümkün olan bütün dizilimleri

a =

1 4 2 6

7) 47 ve 47'den küçük asal sayıları matlab de bulunuz.

8)

a =

2 1 7 8 6 9 6 4 2 3 6 4 5 2 9

Yukarıdaki dizide sayıların dağılımı bulunuz ve grafikle gösteriniz.

Matlab da kompleks sayıların sanal kısmı "i" veya "j" ile ifade edilir. Burada $i = j = \sqrt{-1}$ 'dir. Kompleks sayılarla ilgili her türlü işlem Matlab'de pratik bir şekilde yapılabilir.

```
>> 2+3i
ans =
  2.0000 + 3.0000i

>> 3+4j
ans =
  3.0000 + 4.0000i
```

Kompleks Sayılarla Yapılan İşlemler

toplama:

```
a=2+3i;
b=3+4i;
c=a+b;

c =
  5.0000 + 7.0000i
```

toplama işlemi matrisler üzerinde de yapılabilir.

```
a=[1+2i 2+2i; 3+2i 4+5i]
b=[2+3i 3+2i;1+4i 6+5i]
c=a+b
c =

  3.0000 + 5.0000i   5.0000 + 4.0000i
  4.0000 + 6.0000i  10.0000 +10.0000i
```

çıkarma:

```
a=5+4i
b=2+2i
c=a-b

c =
  3.0000 + 2.0000i

a=[2+3i 1+2i;2+2i 3+3i];
b=[1+i 2;3i 2+4i]
c=a-b

c =

  1.0000 + 2.0000i   -1.0000 + 2.0000i
  2.0000 - 1.0000i    1.0000 - 1.0000i
```

çarpma:

```
a=2+4i
b=3+2i
c=a*b

c =
  -2.0000 +16.0000i

a=[1+2i 3+i
   2+i i
   1   2+2i]
b=[1+i 2+i i
   2+2i 1+i 2]
c=a*b

a =
   1.0000 + 2.0000i   3.0000 + 1.0000i
   2.0000 + 1.0000i     0 + 1.0000i
   1.0000             2.0000 + 2.0000i
b =

   1.0000 + 1.0000i   2.0000 + 1.0000i     0 + 1.0000i
   2.0000 + 2.0000i   1.0000 + 1.0000i     2.0000
c =

   3.0000 +11.0000i   2.0000 + 9.0000i     4.0000 + 3.0000i
  -1.0000 + 5.0000i   2.0000 + 5.0000i    -1.0000 + 4.0000i
   1.0000 + 9.0000i   2.0000 + 5.0000i     4.0000 + 5.0000i
```

Yukarıda işlem normal matris çarpımıdır. Tek fark elemanların kompleks sayılar olmasıdır.

```
a=[1+2i 2+1i;2+3i 2+2i]
b=[2+2i 1+i;3+2i 1+3i]
c=a.*b
a =
   1.0000 + 2.0000i   2.0000 + 1.0000i
   2.0000 + 3.0000i   2.0000 + 2.0000i
b =
   2.0000 + 2.0000i   1.0000 + 1.0000i
   3.0000 + 2.0000i   1.0000 + 3.0000i
c =
  -2.0000 + 6.0000i   1.0000 + 3.0000i
       0 +13.0000i    -4.0000 + 8.0000i
```

Yukarıdaki çarpma işleminde aynı satır ve sütundaki elemanlar karşılıklı olarak çarpılır. Çarpılan matrislerin boyutlarının aynı olması gerekir.

Bölme:

```
a=4+4i
```

```
b=2+5i
c=a/b

c =
  0.9655 - 0.4138i

a=[1+2i 2+1i;2+3i 2+2i;2+3i 1+1i]
b=[2+2i 1+i]
c=a/b

a =
  1.0000 + 2.0000i  2.0000 + 1.0000i
  2.0000 + 3.0000i  2.0000 + 2.0000i
  2.0000 + 3.0000i  1.0000 + 1.0000i
b =
  2.0000 + 2.0000i  1.0000 + 1.0000i
c =
  0.9000 + 0.1000i
  1.4000 + 0.2000i
  1.2000 + 0.2000i
```

Yukarıdaki bölme işlemi normal matris bölmesidir. Tek fark elemanların kompleks sayı olmasıdır.

```
a=[1+1i 2+2i;2+4i 2+4i;4+6i 2+2i]
b=[1+1i 2;   2  1; 2+3i 2+2i]
c=a./b

a =
  1.0000 + 1.0000i  2.0000 + 2.0000i
  2.0000 + 4.0000i  2.0000 + 4.0000i
  4.0000 + 6.0000i  2.0000 + 2.0000i
b =
  1.0000 + 1.0000i  2.0000
  2.0000            1.0000
  2.0000 + 3.0000i  2.0000 + 2.0000i
c =
  1.0000            1.0000 + 1.0000i
  1.0000 + 2.0000i  2.0000 + 4.0000i
  2.0000            1.0000
```

Yukarıdaki bölme işleminde aynı satır ve sütundaki elemanlar karşılıklı olarak bölünür. Bölünen matrislerin boyutlarının aynı olması gerekir.

Komutlar

abs(x): Bu komut verilen kompleks sayının mutlak değerini hesaplar.

```
>> a=3+4i;
>> abs(a)
```

```
ans = 5

b=[1 3+4i ,5+12i 6+8i]
abs(b)

b =
   1.0000                  3.0000 + 4.0000i
   5.0000 +12.0000i        6.0000 + 8.0000i
ans =
    1    5
   13   10
```

angle(x) = Bu komut verilen kompleks sayının radyan cinsinden değerini hesaplar. (-π ile +π arasında) .

```
>> b=2+2i;
>> angle(b)

ans =    0.7854

>> c=0+2i;
>> angle(c)

ans =
   1.5708

a=[10 11+3i 5+3i ;10+10i 5+9i 3+12i ;5i -2+7i -10+17i;-2 -8-2i -9-5i ]
b=angle(a)
derece=b*180/pi

a =
  10.0000                  11.0000 + 3.0000i    5.0000 + 3.0000i
  10.0000 +10.0000i         5.0000 + 9.0000i    3.0000 +12.0000i
      0 + 5.0000i          -2.0000 + 7.0000i   -10.0000 +17.0000i
  -2.0000                  -8.0000 - 2.0000i   -9.0000 - 5.0000i
b =
         0    0.2663    0.5404
    0.7854    1.0637    1.3258
    1.5708    1.8491    2.1025
    3.1416   -2.8966   -2.6345
derece =
         0   15.2551   30.9638
   45.0000   60.9454   75.9638
   90.0000  105.9454  120.4655
  180.0000 -165.9638 -150.9454
```

conj(x) : Bu komut verilen bir kompleks sayının eşleniğini hesaplar.

```
>> a=5+6i;
>> conj(a)
ans =   5.0000 - 6.0000i
```

```
a=[1+2i 2+3i;2+2i 3+1i;4+5i 5+3i]
b=conj(a)

a =
   1.0000 + 2.0000i   2.0000 + 3.0000i
   2.0000 + 2.0000i   3.0000 + 1.0000i
   4.0000 + 5.0000i   5.0000 + 3.0000i
b =
   1.0000 - 2.0000i   2.0000 - 3.0000i
   2.0000 - 2.0000i   3.0000 - 1.0000i
   4.0000 - 5.0000i   5.0000 - 3.0000i
```

real(x) : Bu fonksiyon verilen bir kompleks rakamın reel kısmını hesaplar.

```
>> b=5+2i;
>> real(b)
ans =    5

a=[2+i 3i;1+3i 4+2i]
b=real(a)

a =
   2.0000 + 1.0000i        0 + 3.0000i
   1.0000 + 3.0000i   4.0000 + 2.0000i
b =
     2     0
     1     4
```

imag(x): Bu komut verilen bir kompleks sayının sanal kısmını hesaplar.

```
>> c=7+8i;
>> imag(c)
ans =    8

a=[2+i 3i;1+3i 4+2i]
b=imag(a)

a =
   2.0000 + 1.0000i        0 + 3.0000i
   1.0000 + 3.0000i   4.0000 + 2.0000i
b =
     1     3
     3     2
```

üstlü olarak ifade edilen bir kompleks sayının açılması.

$a.e^{i.b} = a.\cos(b)+i.a.\sin(b)$

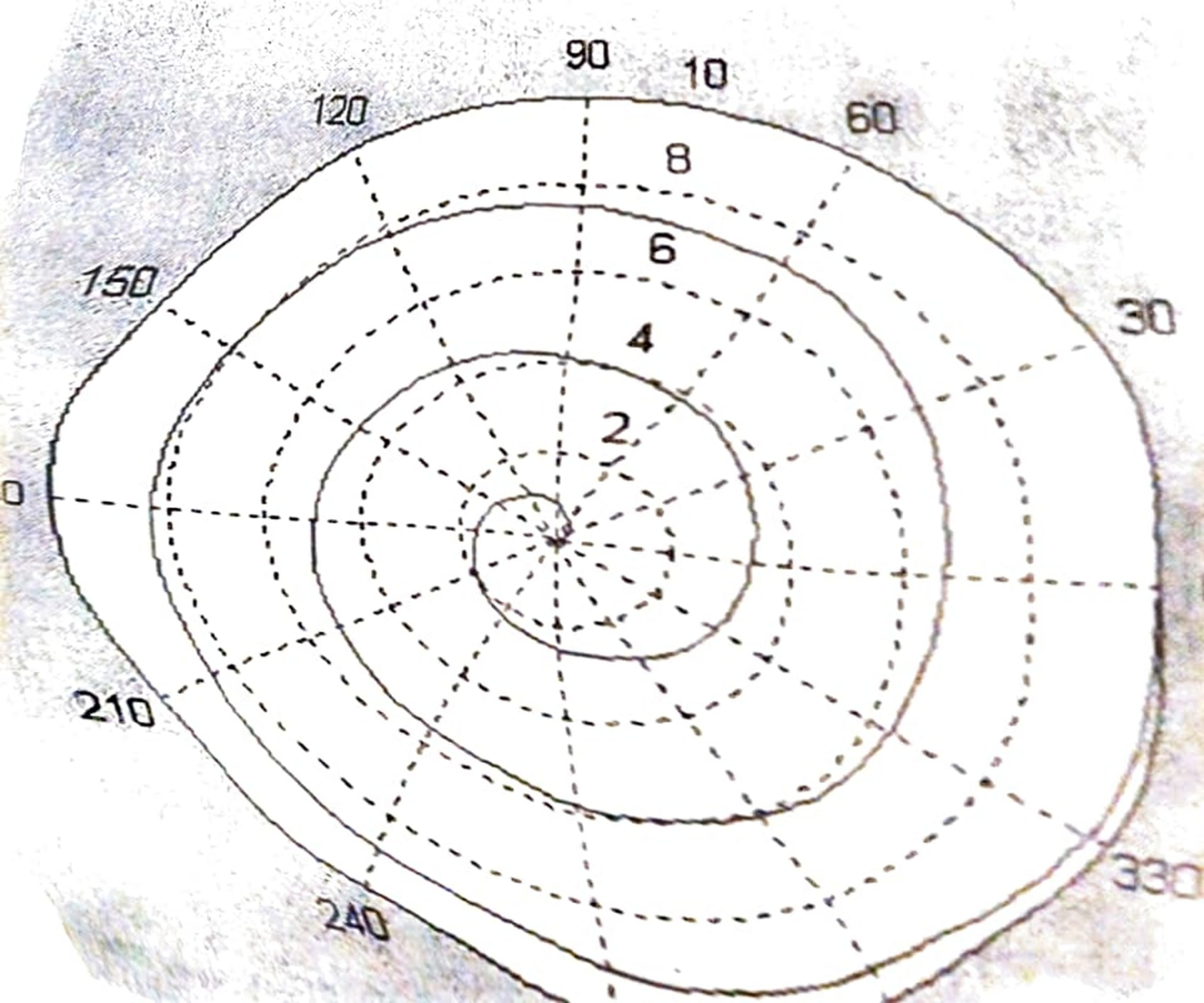
açılım yukarıda gösterilen şekilde yapılır.

p(4*i*pi/6) 10*exp(7*i*pi/6)]

)0i
0i

sayıları kompleks sayı haline çevirir.

ar çizim yapılmasını sağlar.


90
10
120
60
8
6
150
30
4
2
0
210
330
240

b) a-b = ?
c) a*b = ?
d) a/b

cevap:

```
>> a= 2+5i, b= 4+6i


a =
  2.0000 + 5.0000i

b =
  4.0000 + 6.0000i

>> a+b
ans =
  6.0000 +11.0000i

>> a-b
ans =
  -2.0000 - 1.0000i

>> a*b
ans =
 -22.0000 +32.0000i

>> a/b
ans =
  0.7308 + 0.1538i
```

2)
x= 23 + 56i kompleks sayısının mutlak değerini bulunuz.

```
>> abs(23+56i)

ans =
  60.5392
```

3)
y = 9 + 12i kompleks sayısının açısını derece cinsinden bulunuz.

```
>> y=9+12i;
>> aci=angle(y)*180/pi

aci =
  53.1301
```

4)
10+7i sayısının eşleniğini bulunuz.

```
>> conj(10+7i)

ans =
  10.0000 - 7.0000i
```

5) 8+9i sayısının reel ve sanal kısımlarını bulunuz?

```
>> a=8+9i;
>> real(a)

ans =
     8

>> imag(a)
ans =
     9
```

Problemler

1)
x = 2+4i ve y = 3+6i ise
   a) x+y=?
   b) x-y=?
   c) x*y=?
   d) x/y=?

2) k=4+6i ise
   a) k kompleks sayısının mutlak değerini bulunuz.
   b) k kompleks sayısının açısını derece cinsinden bulunuz.
   c) k kompleks sayısının eşleniğini bulunuz.

3) x=23+56 i olduğuna göre
   a) x kompleks sayısının reel kısmını bulunuz.
   b) X kompleks sayısının sanal kısmını bulunuz.

=[1 3 2 2]

→roots komutu:

Bu komut verilen bir polinomun köklerinin bulunması için kullanılır.

Örnek 1:

```
a =[ 1   -5    2    8]
>> roots(e)
ans =
   4.0000
   2.0000
  -1.0000
```

Örnek 2:

```
 b=[1 2 7 4]
>> roots(b)
 ans =
  -0.6732 + 2.3805i
  -0.6732 - 2.3805i
  -0.6536
```

→ **residue komutu**: Bu komut y(s)/x(s) şeklinde verilen bir polinomu kısmi kesirl ayırmaya yarar.

Örnek1:

$\frac{2S^2+4S+6}{S^2+5S+6} = 2 - \frac{12}{S+3} + \frac{6}{S+2}$ dir. Bu işlemi matlab ile yapalım.

```
y=[2 4 6]
x=[1 5 6 ]
[katsayi kutup bolum]=residue(y,x)
 katsayi =
   -12.0000
     6.0000
 kutup =
    -3.0000
    -2.0000
 bolum =
     2
```

Örnek 2:

$\frac{S^3+5S^2+10S+7}{S^2+3S+2}$ fonksiyonunu kısmi kesirlere ayıralım.

```
y=[1 5 10 7];
x=[1 3 2];
[katsayi kutup bolum]=residue(y,x)

katsayi =
     1
     1
kutup =
```

```
   -2
   -1
bolum =
    1    2
```

Bu durumda fonksiyonumuz :$S + 2 + \frac{1}{S+2} + \frac{1}{S+1}$ haline gelir.

Burada katsayı kısmi kesirlerin katsayılarını, kutup paydayı sıfır yapan değerleri, bölüm ise payın paydaya bölümünden elde edilen sonucu göstermektedir.
Kısmi kesirlerden yola çıkarak da y(s)/x(s) fonksiyonunu elde etmek mümkündür.
Örnek 1:

```
katsayı=[2;4];
kutup=[-1;-3];
bolum=[];
[y x]=residue(katsayi,kutup,bolum)
y =
    6   10
x =
    1    4    3
```

Böylece $\frac{2}{S+1} + \frac{4}{S+3}$ toplamından $\frac{6S+10}{S^2+4S+3}$ elde edilir.

Örnek 2:

$2 + \frac{1}{S+3} + \frac{2}{S-1}$ işlemini residue komutunu kullanarak yapınız.

```
katsayi=[1 2];
kutup=[-3 1];
bolum=2;
[y x]=residue(katsayi,kutup,bolum)
y =
    2    7   -1
x =
    1    2   -3
```

yukarıdaki katsayılara göre işlemin çözümü: $\frac{2S^2+7S-1}{S^2+2S-3}$ olur.

→ **poly komutu**: Bu komut kökleri belli olan bir polinomun katsayılarını hesaplar.
Örnek 1:

```
>> kok=[-2 2];
>> poly(kok)
ans =
    1    0   -4
```

Kökleri –2 ve +2 olan polinom x^2-4'dür.
Örnek 2:
Kökleri 1+i,1-i, -1 ve 2 olan polinomun katsayılarını bulunuz.

```
kok=[1+i 1-i -1 2];
a=poly(kok)
a =
    1   -3    2    2   -4
```

kökleri 1+i,1-i,-1 ve olan polinom x^4-$3x^3$+$2x^2$+2x-4'dür.
→**polyval(p,x)** komutu: Bu komut ile katsayıları belli olan bir polinomda girilen değerlere karşılık sonuçlar elde edilir.

Örnek 1:

```
f=[1 3 2 2]
x=[1 2 ]
polyval(f,x)
ans =
    8   26
```

Örnek 2:

```
f=[1 2 1]
x=[1 2;3 1]
polyval(f,x)
ans =
    4    9
   16    4
```

Örnek 3:

Elimizde y = x^3-3x^2-x+3 şeklinde bir denklem olsun. Bu denklemin x=–1.6 ile +3.4 arasına karşılık gelen değerlerini bulalım ve çizdirelim.

```
lc
=[1 -3 -1 3]
=linspace(-1.6,3.4,20);
=polyval(p,x)
ot(x,y)
d
```

```
x=[1 2]
polyval(f,x)
ans =
    8    26
```

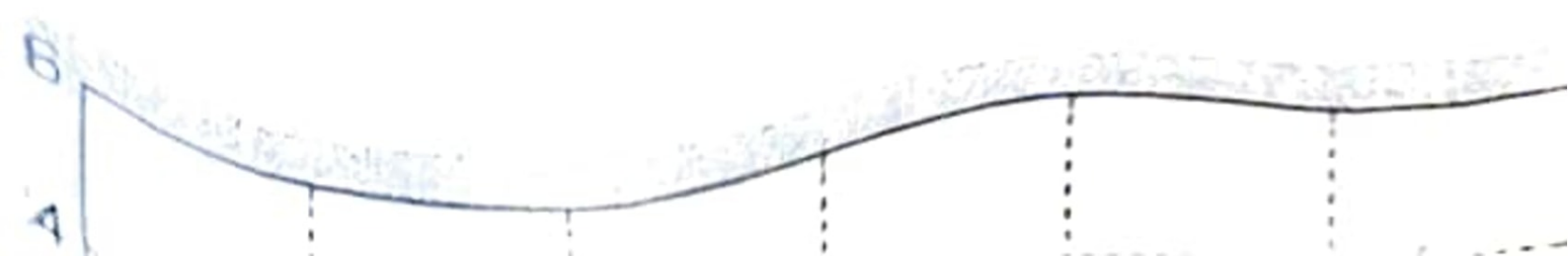
Örnek 2

```
f=[1 2 1]
x=[1 2,3 1]
polyval(f,x)
ans =
     4     9
    16     4
```

Örnek 3:

Elimizde $y = x^3-3x^2-x+3$ şeklinde bir denklem olsun. Bu d karşılık gelen değerlerini bulalım ve çizdirelim.

```
le
p=[1 -3 -1 3]
x=linspace(-1.6,3.4,20);
y=polyval(p,x)
plot(x,y)
grid
```



Örnek 1:

```
f=[1 3 2 2]
x=[1 2 ]
polyval(f,x)
ans =
    8   26
```

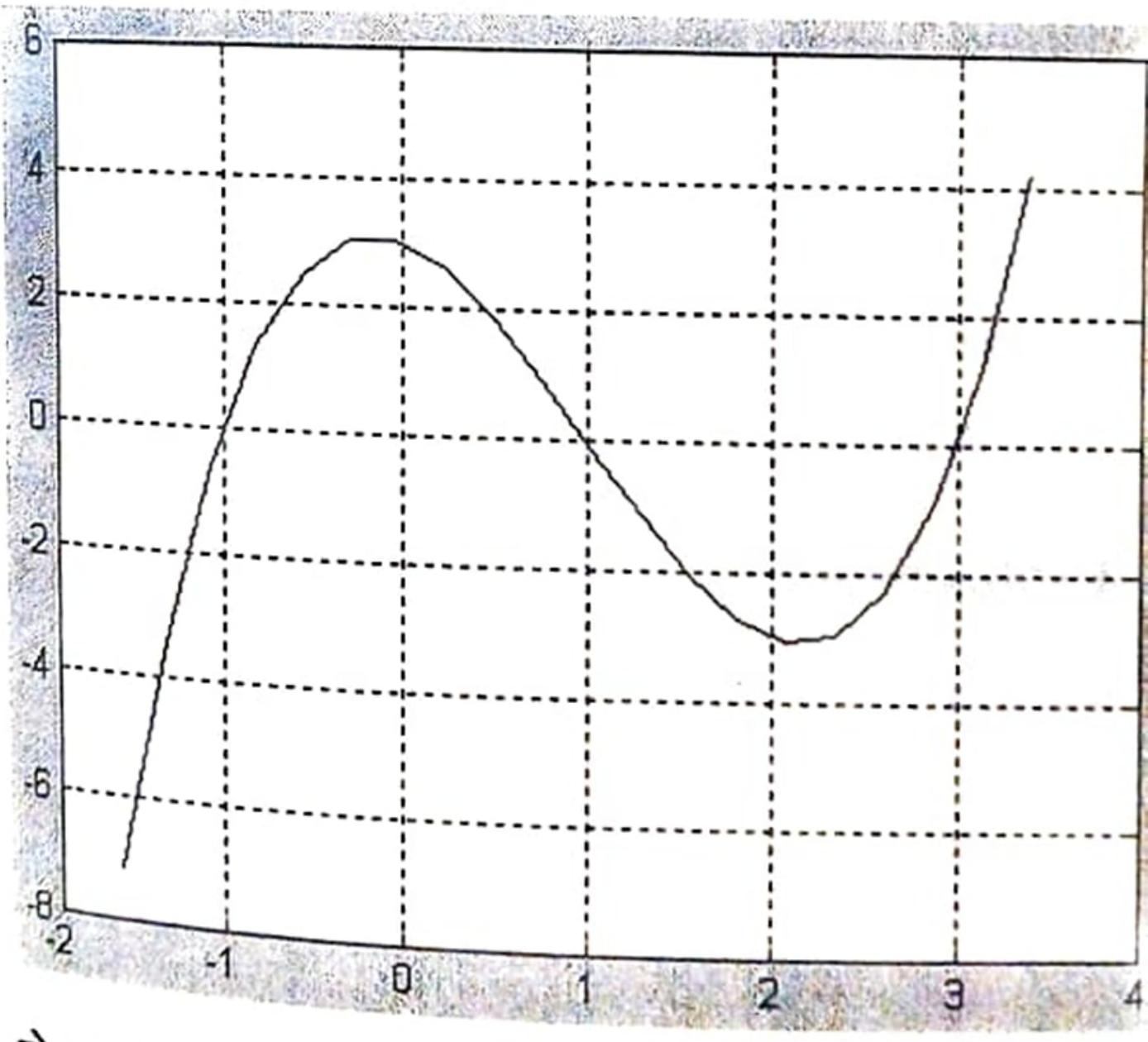
Örnek 2:

```
f=[1 2 1]
x=[1 2;3 1]
polyval(f,x)
ans =
     4     9
    16     4
```

Örnek 3:

Elimizde $y = x^3-3x^2-x+3$ şeklinde bir denklem olsun. Bu de
karşılık gelen değerlerini bulalım ve çizdirelim.

```
clc
p=[1 -3 -1 3]
x=linspace(-1.6,3.4,20);
y=polyval(p,x)
plot(x,y)
grid
```



→**conv komutu**: Bu komut ile iki polinom birbiri ile çarpılır.

Örnek 1:

$f(x)=x^2+2x+1$ ile $g(x)=2x^2+3x+2$ polinomları çarpılıyor.
Sonuç=$f(x).g(x)=2x^4+7x^3+10x^2+7x+2$ olur

Örnekler

Bu işlemi Matlab'de yapalım.

```
f=[1 2 1]
g=[2 3 2]
conv(f,g)

ans =
    2    7   10    7    2
```

Örnek 2:
$f(x)=x^3+2x^2+4x+3$ ile $g(x)=x^2+4x+5$ polinomlarını çarpınız.

```
a=[1 2 4 3];
b=[1 4 5];
c=conv(a,b)
c =
    1    6   17   29   32   15
```

Bu iki fonksiyonun çarpımı $x^5+6x^4+17x^3+29x^2+32x+15$'dir.

→**deconv komutu**: Bu komut ile iki polinom birbirine bölünür. Eğer bölümde kalan varsa onu da görmek mümkündür.
Örnek 1:
$h(x)=2x^2+8x+7$ polinomunu $2x+1$ polinomuna bölürsek bölüm $x+3.5$, kalan 3.5'dir.
Bu işlem matlab'de şu şekilde yapılır.

```
h=[2 8 7]
g=[2 1]
[q r]=deconv(h,g)
q =
    1.0000    3.5000
r =
         0         0    3.5000
```

Örnek 2:
$h(x)=4x^2+18x+8$ polinomu $2x+1$'e bölünürse sonuç $2x+8$'dir ve kalan yoktur

```
h=[4 18 8 ]
g=[2 1]
[q r]=deconv(h,g)
q =
    2    8
r =
    0    0    0
```

Örnek 3:
x^3+3x^2+4x+6 polinomu x^2+x+1 polinomuna bölünürse bölüm ve kalan ne elde edilir.?

```
a=[1 3 4 6];
b=[1 1 1];
[bolum kalan]=deconv(a,b)
bolum =
    1    2
kalan =
```

0 0 1 4

Yukarıdaki polinomlardan bölüm olarak x+2, kalan olarak x+4 elde edilir.

→iki polinomun toplamı:
Örnek 1:
$a(x)=2x^3+x^2+3x$ ve $b(x)=2x^2+1$ şeklinde iki polinomu toplarsak sonuç, $2x^3+3x^2+3x+1$ olur. Bu işlemi Matlab de yaparken olmayan dereceler önüne sıfır kullanılmalıdır.

```
a=[2 1 3 0];
b=[0 2 0 1];
c=a+b
c =
   2   3   3   1
```

Örnek 2:
$p(x)=x^4+3x^3+5x^2+x+3$ ile $g(x)=2x^4+2x^3+3x^2+3x+4$ polinomlarını toplayınız.

```
p=[1 3 5 1 3];
g=[2 2 3 3 4];
a=p+g;
fprintf('sonuc=%dx^4+%dx^3+%dx^2+%dx+%d',a(1),a(2),a(3),a(4),a(5))
sonuc=3x^4+5x^3+8x^2+4x+7
```

→Bir polinomun skaler ile çarpımı:
Örnek:
Elimizde bulunan $b(x)=3x^2+6x+2$ şeklinde bir polinomu 3 ile çarparsak şu sonucu elde ederiz. $a(x)=3*b(x)=9x^2+18x+6$. Bu işlemi matlab ile şu şekilde gerçekleştirebilriz.

```
b=[3 6 2];
a=3*b
a =
   9   18   6
```

→polyder komutu: bu komut bir polinomun türevini hesaplar.
Örnek 1:
$a(x)=x^4+3x^3+x^2+5x+19$ polinomunun türevi $b(x)=4x^3+9x^2+2x+5$ olarak elde edilir. Bu işlemi matlab'de gerçekleştirirsek

```
a=[1 3 1 5 19];
b=polyder(a)
b =
   4   9   2   5
```

Örnek 2:
$a(x)=5x^4+2x^3+3x^2+4x+3$ polinomunun türevini alınız.

```
a=[5 2 3 4 3]
b=polyder(a)
fprintf('a polinomunun türevi=%dx^3+%dx^2+%dx+%d',b(1),b(2),b(3),b(4))
a =
   5   2   3   4   3
b =
   20   6   6   4
a polinomunun türevi=20x^3+6x^2+6x+4
```

→ployint komutu: Bu komut verilen polinomun integralini alır.

Örnek 1: Yukarıdaki örnekte verilen $b(x)=4x^3+9x^2+2x+5$ polinomunun integralini alırsak $a(x)=x^4+3x^3+x^2+5x+19$ olarak bulunur.

```
b=[4 9 2 5];
a=polyint(a,19)
a =
   1   3   1   5   19
```

eğer integral sabiti komut içinde belirtilmez ise bu katsayı sıfır olacaktır.

```
>> polyint(b)
ans =
   1   3   1   5   0
```

Örnek 2:

h(x)=6x^2+8x+5 polinomunun integralini alınız.

```
h=[6 8 5];
g=polyint(h,4)
fprintf('h polinomunun integrali=%dx^3+%dx^2+%dx+%d',g(1),g(2),g(3),g(4))
g =
   2   4   5   4
h polinomunun integrali=2x^3+4x^2+5x+4'dür.
```

Eğri Çizdirme

Elimizde x ve ona karşılık gelen y değerleri olsun. Bu değerlere karşılık gelen yaklaşık eğri denklemini polyfit komutu ile bulabiliriz.

kullanım:

polyfit(x,y,n)

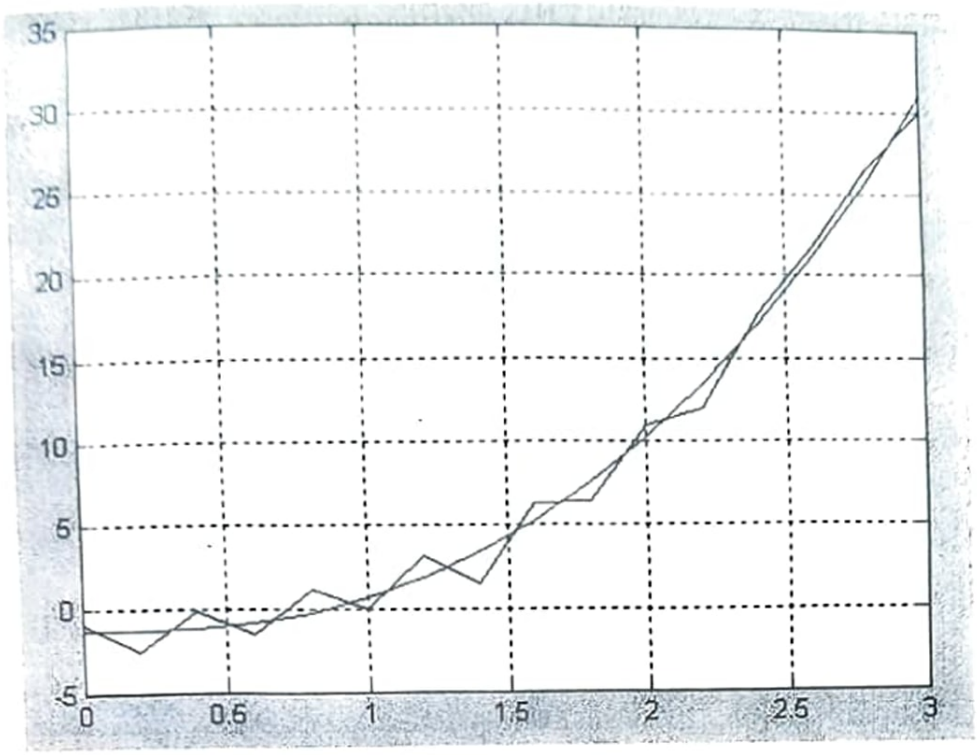
Burada n dereceyi göstermektedir. Örneğin x^2+2x+5 şeklinde bir denklem için n=2'dir.

Örnek1:

```
clc
x=[0 0.2 0.4 0.6 0.8  1.0 1.2 1.4 1.6  1.8  2.0  2.2  2.4  2.6  2.8  3.0];
y=[-1.0 -2.592  -0.136 -1.584  1.112  0 3.128 1.544 6.296 6.432 11.0 12.048  17.624 21.776
26.5520  30.0];
polyfit(x,y,3)
ans =  0.5779   2.0703   -0.6676   -1.3148
```

Bu durumda $y=0.5779x^3+2.0703x^2-0.6676x-1.3148$ şeklinde bir denklem elde ederiz.

Gerçek değerlerle, bulduğumuz 3. dereceden denklemin grafiğini aşağıda verilmiştir.

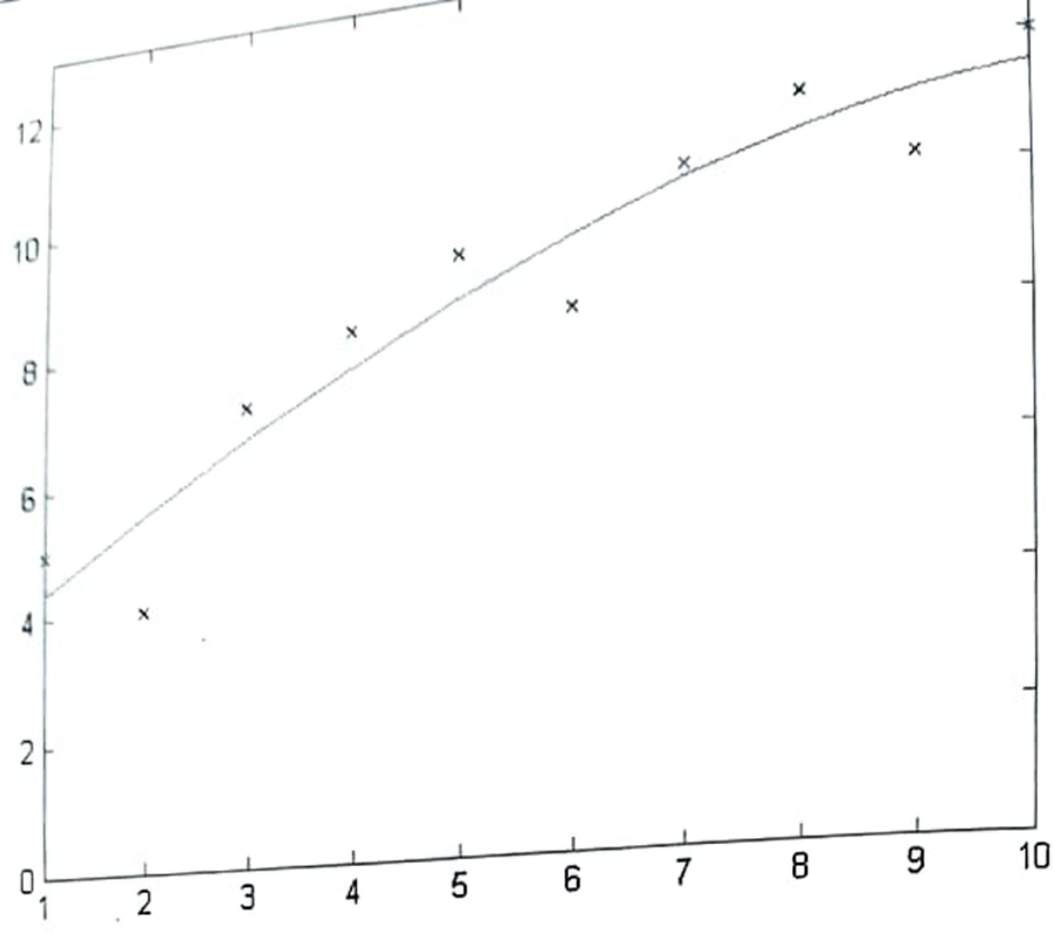


Görüldüğü gibi yeşil renkle gösterilen denklemin grafiği, orijinal değerlerin grafiğine oldukça yaklaşmıştır.

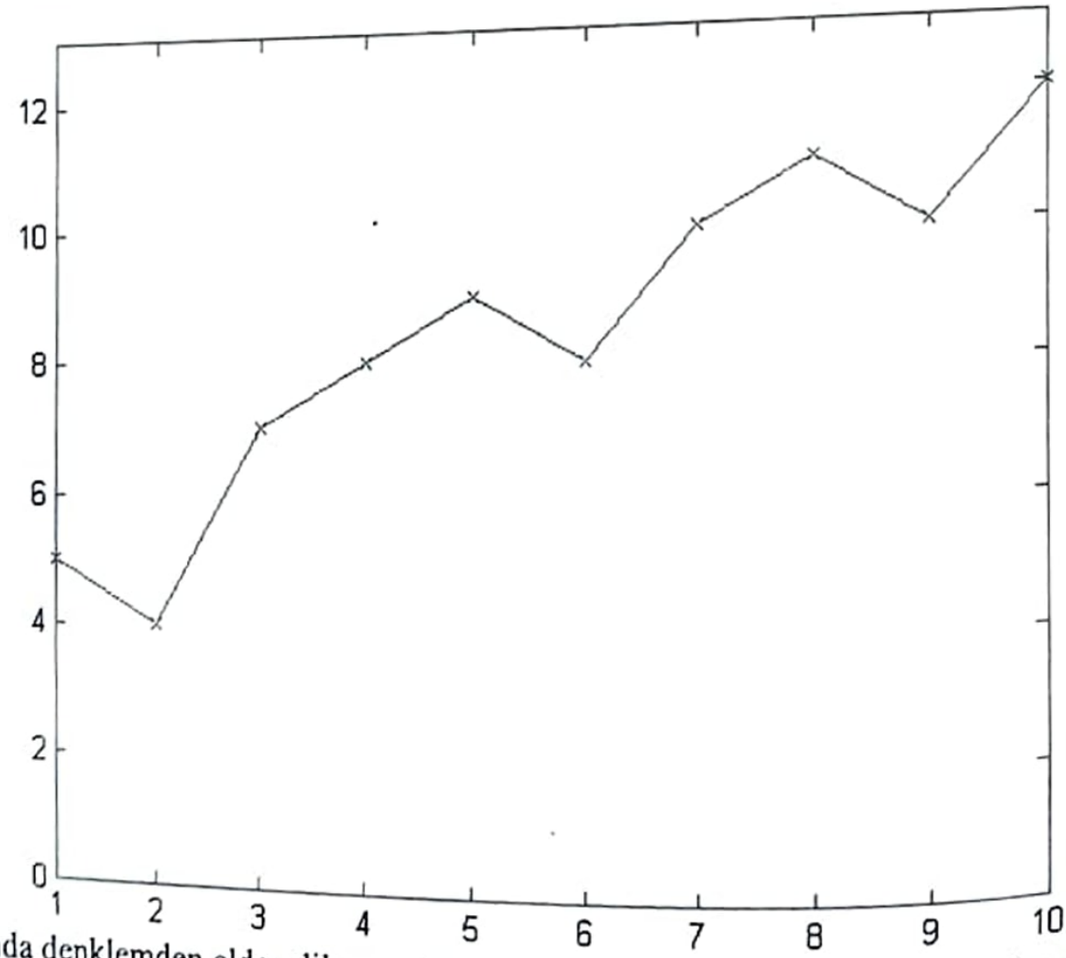
Örnek 2:
A=[1 2 3 4 5 6 7 8 9 10] noktalarına karşılık b=[5 4 7 8 9 8 10 11 10 12] noktaları verilmiştir. Bu noktalara en yakından geçen 3. dereceden eğrinin denklemini bulmaya çalışalım.

```
a=1:1:10;
b=[5 4 7 8 9 8 10 11 10 12];
f=polyfit(a,b,2)
y=polyval(f,a);
plot(a,b,'x',a,y)
axis([1 10 0 13])
```

f değerleri -0.0379 1.2045 3.2333 olarak bulunur. Yani denklem : $-0.0379x^2+1.2045x+3.2333$ dür. Bu durumda grafik aşağıdaki gibi çıkar.



Eğer denklemin derecesi 2'den 9'a çıkartılırsa aşağıdaki grafik elde edilir.

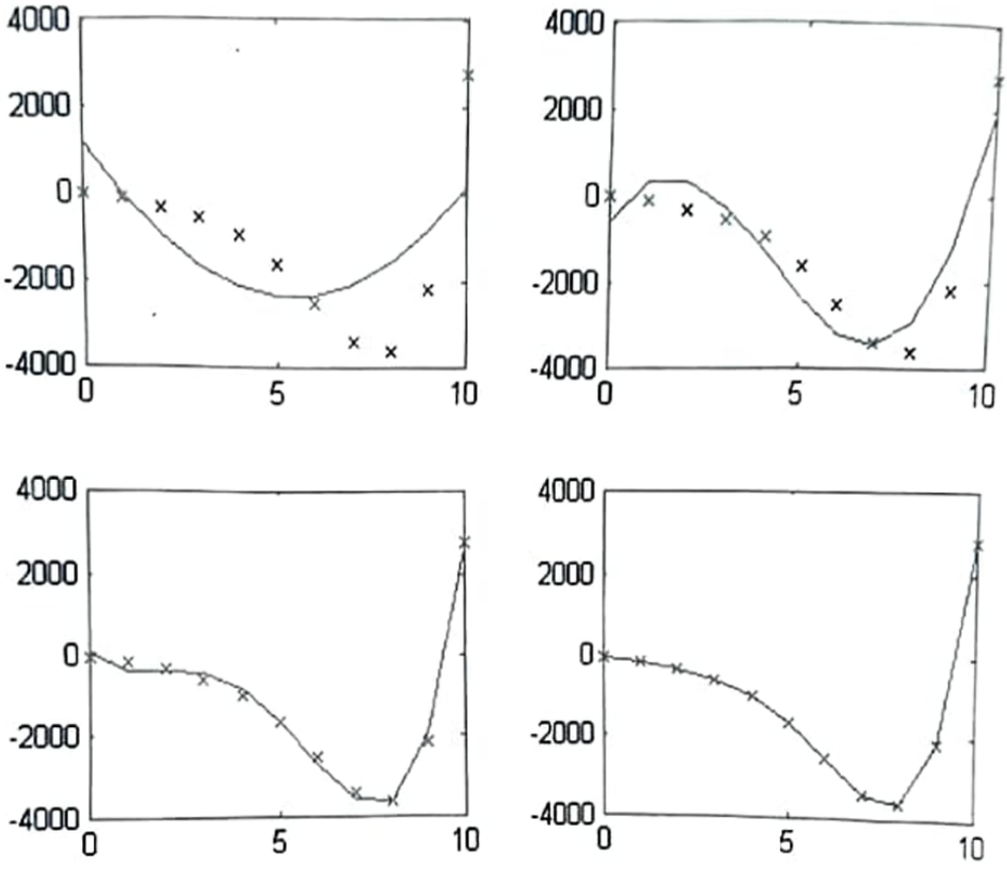


Bu durumda denklemden elde edilen grafik tam olarak verilen noktaların üstünden geçer.

Örnek 3:

x=[0 1 2 3 4 5 6 7 8 9 10] noktalarına karşılık y=[-50 -166 -354 -596 -994 -1650 -2546 -3424 -3666 -1247 2750] noktaları verilmiştir. Bu noktalara en yakından geçen 2., 3., 4. ve 5. dereceden denklemleri çizdiriniz.

```
a=0:10
b=[-50 -166 -354 -596 -994 -1650 -2546 -3424 -3666 -2174 2750];
for i=2:5
   f=polyfit(a,b,i);
   y=polyval(f,a);
   subplot(2,2,i-1),plot(a,b,'x',a,y)
end
```



Alıştırmalar

1) $x^3+x^2-10x+8$ fonksiyonunun köklerini bulunuz.
Cevap:

```
y=[1 1 -10 8];
roots(y)
ans =
  -4.0000
   2.0000
   1.0000
```

2)
Aşağıdaki y(s)/x(s) şeklinde verilen ifadeyi kısmi kesirlere ayırınız.

$$\frac{7S+15}{S^2+6S+5}$$

Cevap:

```
y=[7 15]
x=[1 6 5]
[katsayi,kutup,bolum]=residue(y,x)

katsayi =
    5
    2
kutup =
   -5
   -1
bolum =
    []
```

3) Kökleri –1,2 ve 3 olan polinom nedir?

```
poly([-1 2 3])
ans =
    1   -4    1    6
```

Kökleri –1,2,3 olan polinom S^3-4S^2+S+6'dır.

4) $y = x^3+6x^2-4x+5$ fonksiyonunun x=2 ve x=4 için değerlerini bulunuz.

```
y=[1 6 4 5 ];
x=[2 4]
polyval(y,x)
ans =
   45   181
```

5)

$a(S)=S^2+2S+1$ ile $b(S)=S^3+2$ polinomlarını birbirleri ile çarpınız.

Cevap:

```
a=[1 2 1 ];
b=[1 0 0 2]
convl(a,b)

ans =
    1    2    1    2    4    2
```

6) $a(S)=2S^3+S^2+2$ polinomunu, $b(S)=S^2+S+2$ polinomuna bölünüz.

Cevap:

```
a=[2 1 0 2 ];
b=[1 1 2]
[bolum kalan]=deconv(a,b)
bolum =
    2   -1
```

```
kalan =
    0    0   -3    4
```

7) $a(x)=x^3+4x^2+2x+1$ polinomundan, $b(x)=3x^2+3x+4$ polinomunu çıkartınız.

```
a=[1 4 2 1];
b=[0 3 3 4];
sonuc=a-b

sonuc =
    1    1   -1   -3
```

8) $2x^4+x^3+4x^2+3x+4$ fonksiyonunun türevini alınız.

```
y=[2 1 4 3 4];
polyder(y)

ans =
    8    3    8    3
```

9) $9x^2+4x+5$ fonksiyonun integralini hesaplayınız.

```
y=[9 4 5];
polyint(y)

ans =
    3    2    5    0
```

10) Aşağıda verilen diziye en yakın eğri denklemini bulunuz ve verilen noktalar ile eğriyi aynı grafik üzerinde çizdiriniz.

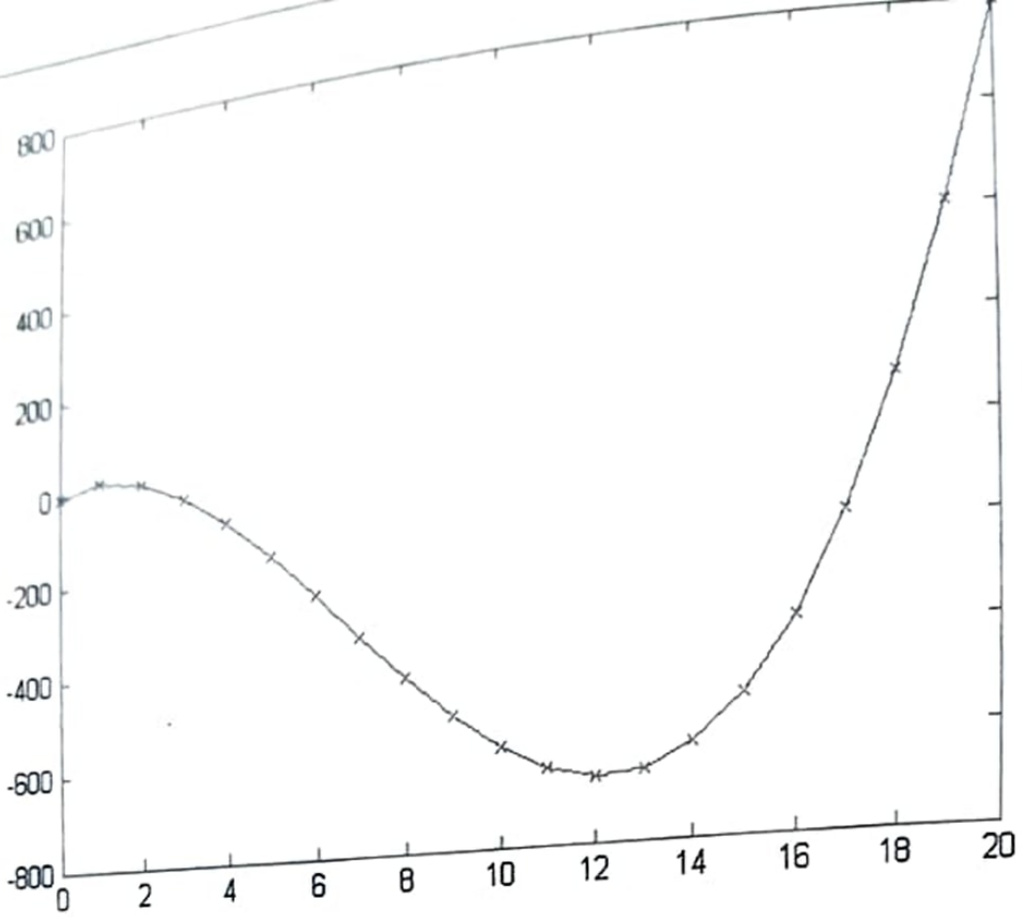
```
x=0:20;
y=[-1 20 7 -34 -97 -176 -265 -358 -449 -532 -601 -650 -673 -664 -617  -526  -385  -188    71
398   799]
```

cevap:

```
x=0:20;
y=[-1 20 7 -34 -97 -176 -265 -358 -449 -532 -601 -650 -673 -664 -617  -526  -385  -188    71
398   799]
f=polyfit(x,y,3)
y2=polyval(f,x);
plot(x,y,'x',x,y2)

f =
   1.0000  -20.0000   40.0000   -1.000
```

Bulduğumuz f katsayıları yukarıdaki gibi olduğu için denklemimiz:

$y=x^3-20x^2+40x-1$ olur.



Problemler

1) $x^5-9x^4+19x^3+21x^2-92x+60$ denkleminin köklerini bulunuz.

2) Aşağıda verilen ifadeyi kısmi kesirlere ayırınız.

$$\frac{6x-10}{x^2-4x+3}$$

3) Kökleri –2, 3,-4 olan polinomu bulunuz.

4) x^3+2x^2+4x-1 polinomu ile x^2+5x+6 polinomlarını çarpınız.

5) $x^4+3x^3-5x^2+x-4$ polinomunu x^2+5 polinomuna bölünüz.

6) x^3+2x^2+3x+4 polinomundan x^2+4x+5 polinomunu çıkarınız.

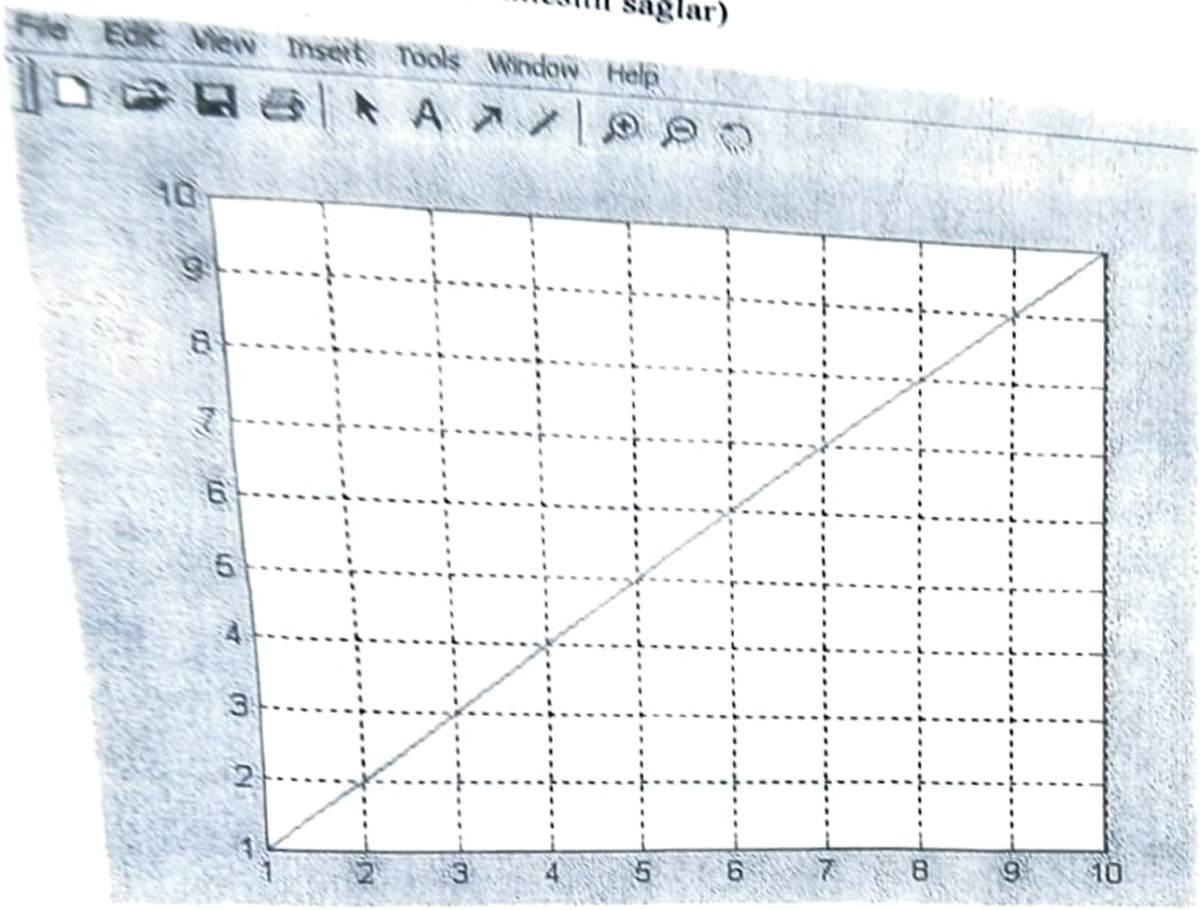
7) $5x^4+4x^3+2x^2+6x+4$ polinomunun türevini alınız.

8) $8x^3+6x^2+8x+4$ polinomunun integralini hesaplayınız.

9)
Aşağıda verilen dizilerden yaklaşık olarak geçen 8. dereceden geçen eğri denklemini bulunuz.
Bu diziyi ve bulduğunuz eğri denklemini aynı grafik üzerinde çiziniz.

```
x=1:10;
y=[1 4 5 5 4 3 4 2 3 6]
```

bölünmesini sağlar)



plot komutuna eklenecek 3. bir değişkenle grafikte, çizgilerin şekli, noktaların şekli veya renk değiştirilebilir.

Çizgi çeşitleri:

-	Düz çizgi
--	Kesikli çizgi
:	İki nokta üst üste
-.	Nokta ve çizgi

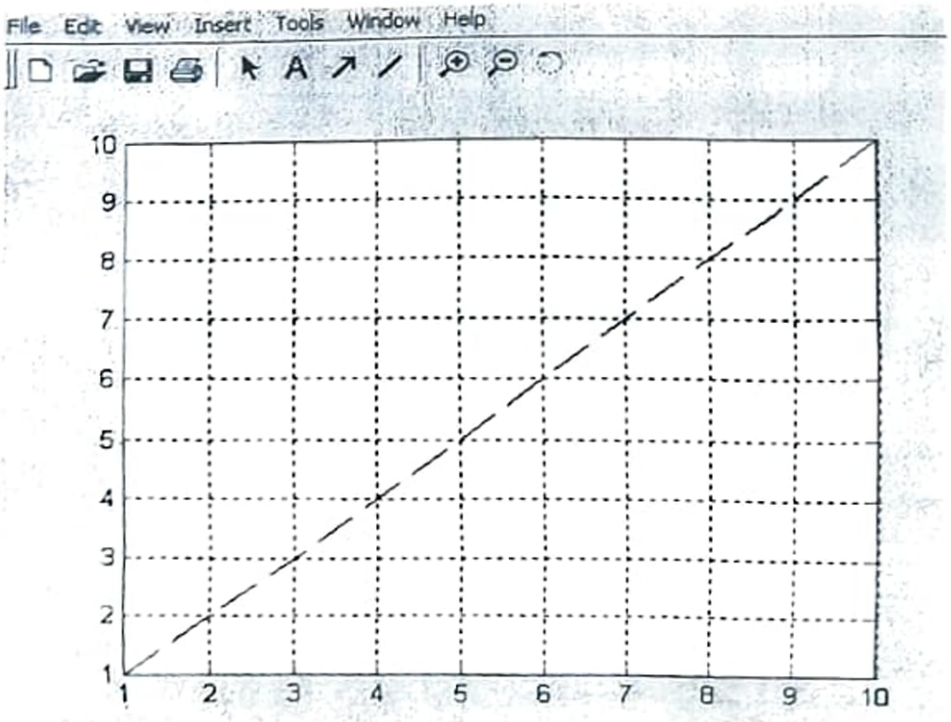
Nokta çeşitleri:

+	Artı
O	Çember
*	Asteriks
.	Nokta
X	Çarpı
S	Kare
D	Elmas
^	Yukarı ok
>	Sağa ok
<	Sola ok
P	Pentagram
H	Hexagram

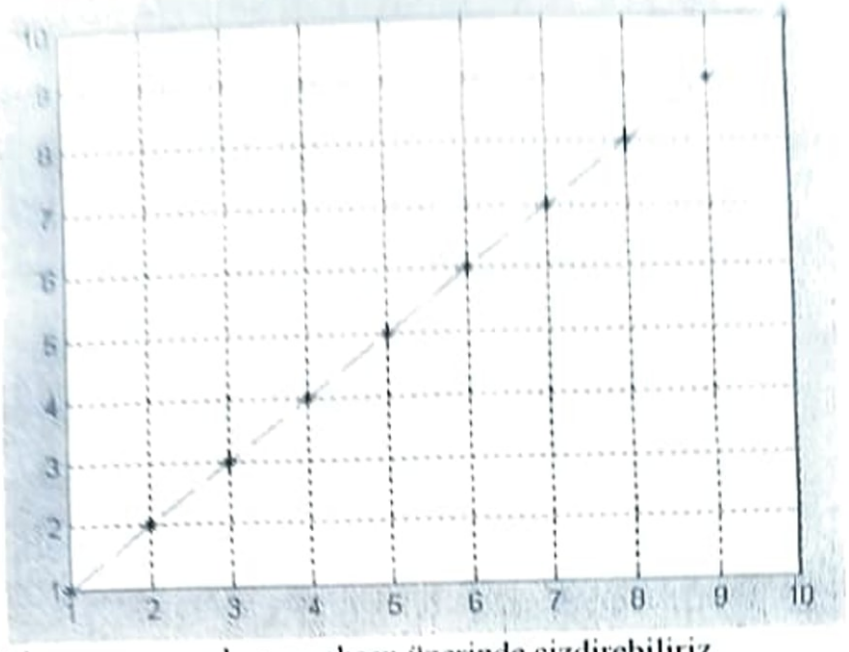
Renk çeşitleri

R	Kırmızı
G	Yeşil
B	Mavi
C	Cyan
M	Koyu kırmızı
Y	Sarı
K	Siyah
W	Beyaz

plot(x,y,'--') komutu ile y=x grafiği kesik çizgiler ile çizilir.

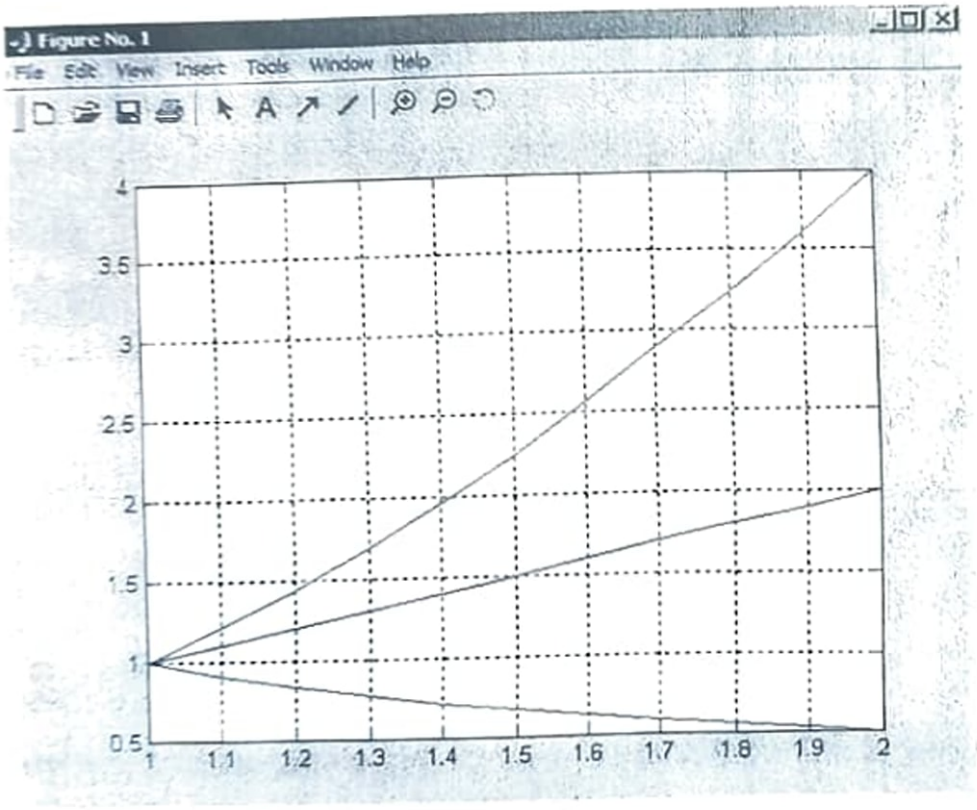


plot komutu içinde aynı anda çizgi çeşidini, nokta çeşidini ve renk çeşidini seçebiliriz
plot(x,y,'--*r') komutu ile kesikli çizgili, asterisk işaretli ve kırmızı renkte grafik çizilecektir.



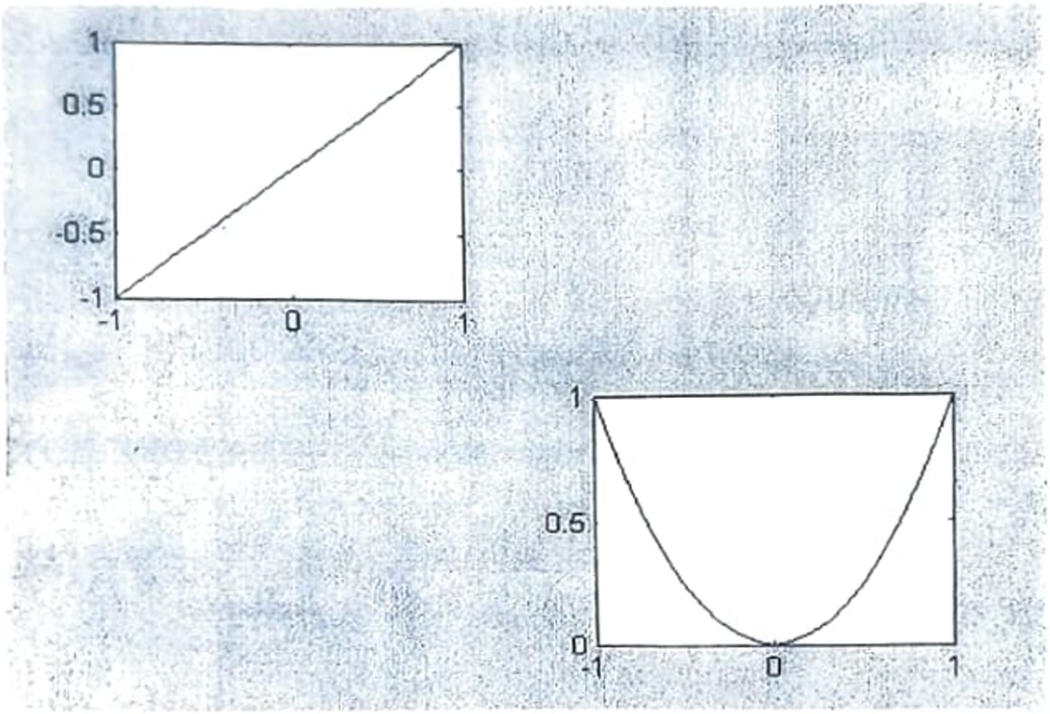
1 den fazla grafiği aynı pencerede aynı eksen üzerinde çizdirebiliriz.
plot(x,y1,x,y2,x,y3.....) şeklinde istediğimiz kadar grafiği çizebiliriz. Her yeni grafik için renk otomatik olarak değişecektir.

```
x=1:0.1:2;
y1=x;
y2=x.^2;
y3=1./x;
plot(x,y1,x,y2,x,y3)
grid
```



Ayrıca MATLAB'de bir pencere n*m boyuta bölünebilir ve istenilen pencere seçilerek grafik oraya çizdirilebilir.

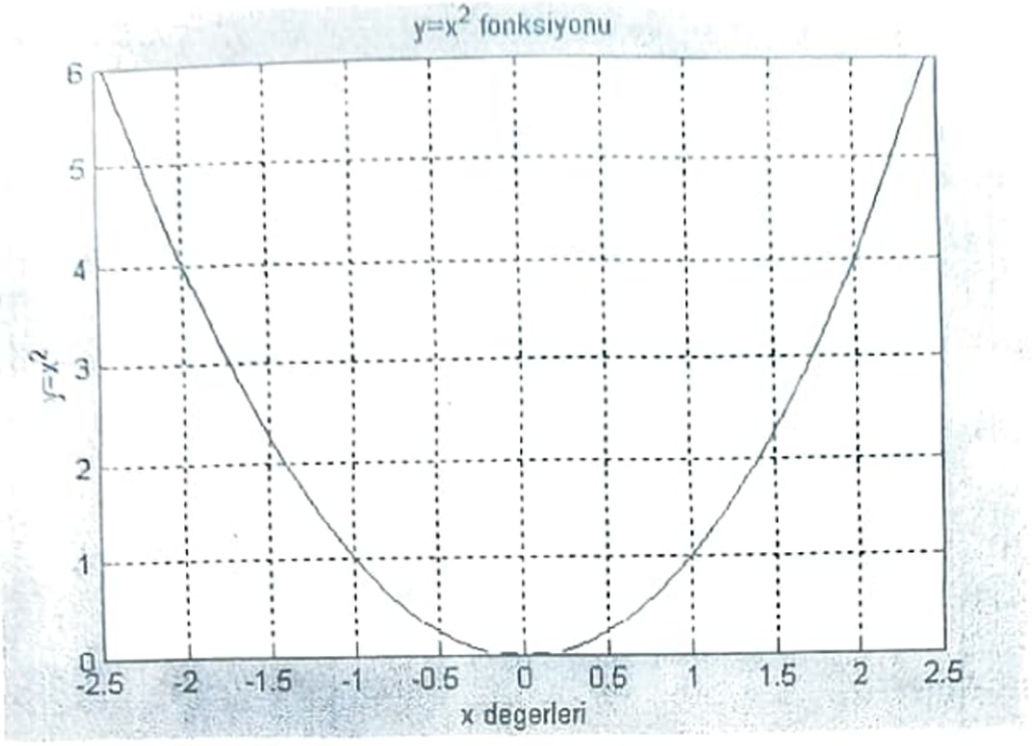
```
x=1:0.1:2;
y1=x;
y2=x.^2;
subplot(221),plot(x,y1);
subplot(224),plot(x,y2);
```



Matlab'de bir grafiğin x ve y eksenine açıklama ilave etmek ve grafiğe başlık eklemek mümkündür. Ayrıca eksenlerin değişim aralığını belirlemek mümkündür.

```
x=-3:.2:3;
y=x.^2;
plot(x,y)
axis([-2.5  2.5 0 6])
grid
xlabel('x degerleri')
ylabel('y=x^2 ')
title('y=x^2 fonksiyonu')
```

xlabel komutu ile x eksenine açıklama, ylabel konutu ile de y eksenine açıklama eklenebilir. Title komutu ile de grafiğin üstüne açıklama eklemek mümkündür.

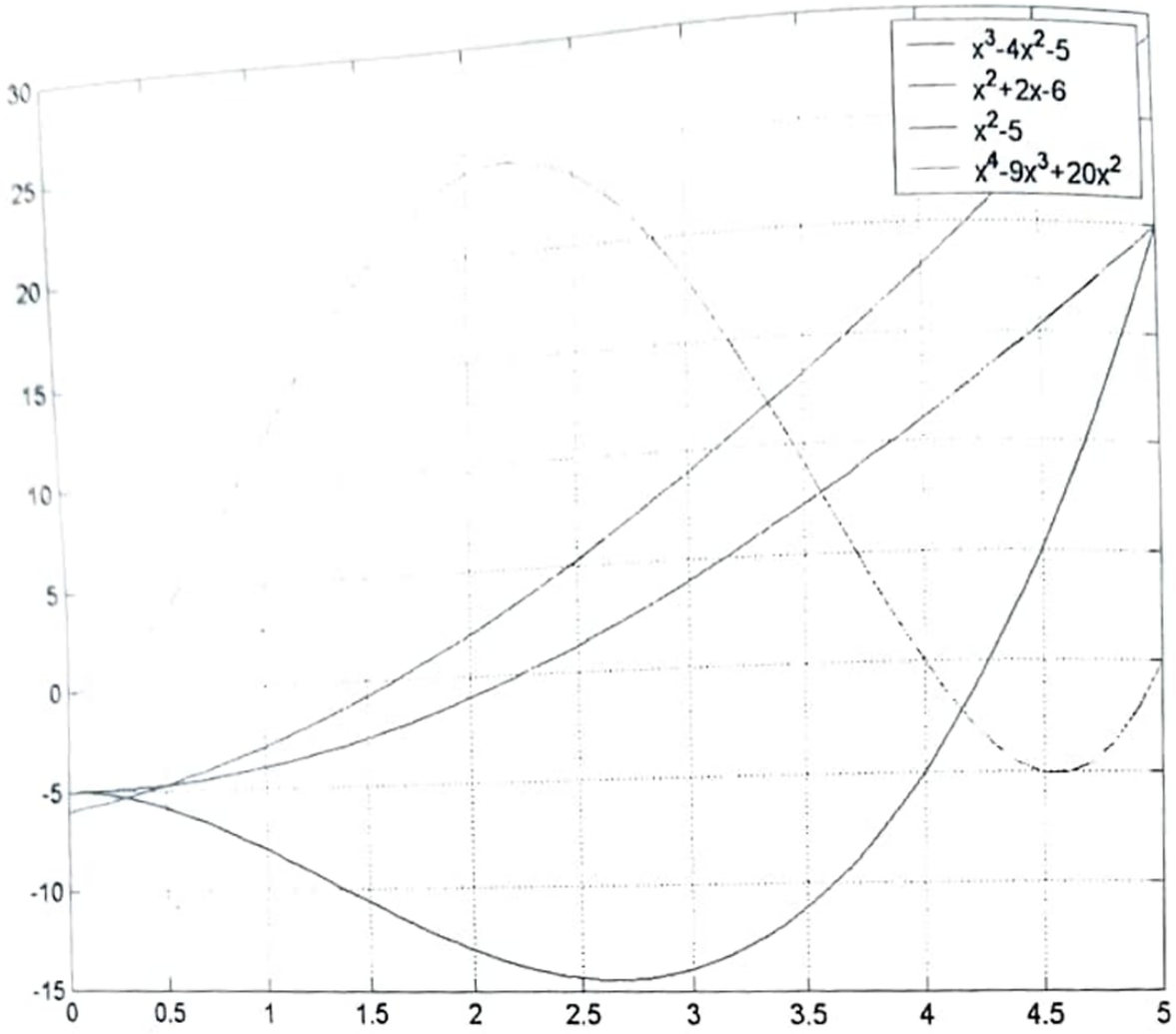


Legend komutu ile grafiklerle ilgili bilgi vermek mümkündür.
Kullanımı:
Legend('açıklama1','açıklama2')
Legend('açıklama1','açıklama2',yer)

Yer=-1 seçilirse açıklamalar eksenlerin dışına yazılır, yer=0 seçilirse eksenlerin içine mümkün olduğu kadar az yer kaplayacak şekilde yazılır, yer=1 seçilirse üst sağ tarafa yazılır, yer=2 seçilirse üst sol tarafa yazılır, yer=3 seçilirse alt sol tarafa yazılır, yer 4 seçilirse alt sağ tarafa yazılır.

Örnek:

```
x = 0:.1:5;
y1=(x.^3-4*x.^2-5);
y2=(x.^2+2*x-6);
y3=(x.^2-5);
y4=(x.^4-9*x.^3+20*x.^2)
plot(x,y1,x,y2,x,y3,x,y4),grid
h = legend('x^3-4x^2-5','x^2+2x-6','x^2-5','x^4-9x^3+20x^2',1);
```



text: Bu komut ile çizilen bir grafiğin istenilen pozisyonuna yazı eklemek mümkündür. Kullanımı:

text(x,y,'yazı')
veya
text(x,y,'özellik',özellik değeri)

yazıya değişik karakterler ekleme imkanı vardır. Bu karakterlerden bazıları aşağıda verilmiştir.

Yazı içinde kullanılan kısım	Karşılık gelen karakter
\alpha	α
\beta	β
\phi	ϕ
\infty	∞
\leftrightarrow	↔
\leftarrow	←
\uparrow	↑
\rightarrow	→
\downarrow	↓

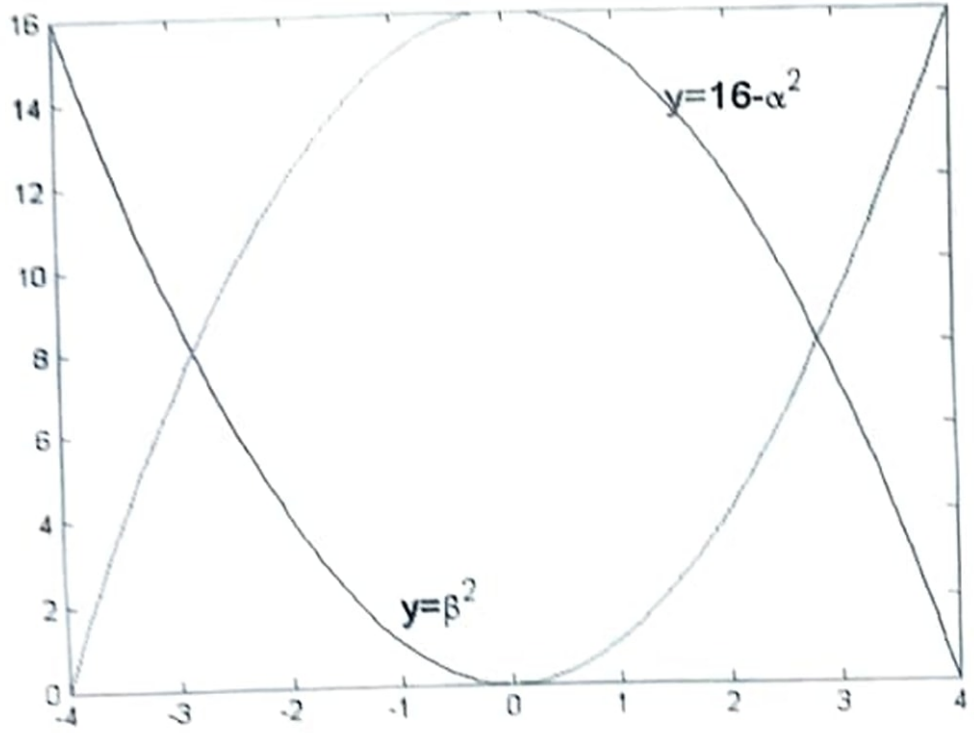
\mu	μ
\pm	±
\pi	π
\div	÷
\approx	≈
\int	∫
\copyright	©

Ayrıca yazıya bazı özellikler ilave etmek mümkündür Aşağıda bu özelliklerden bazıları belirtilmiştir.

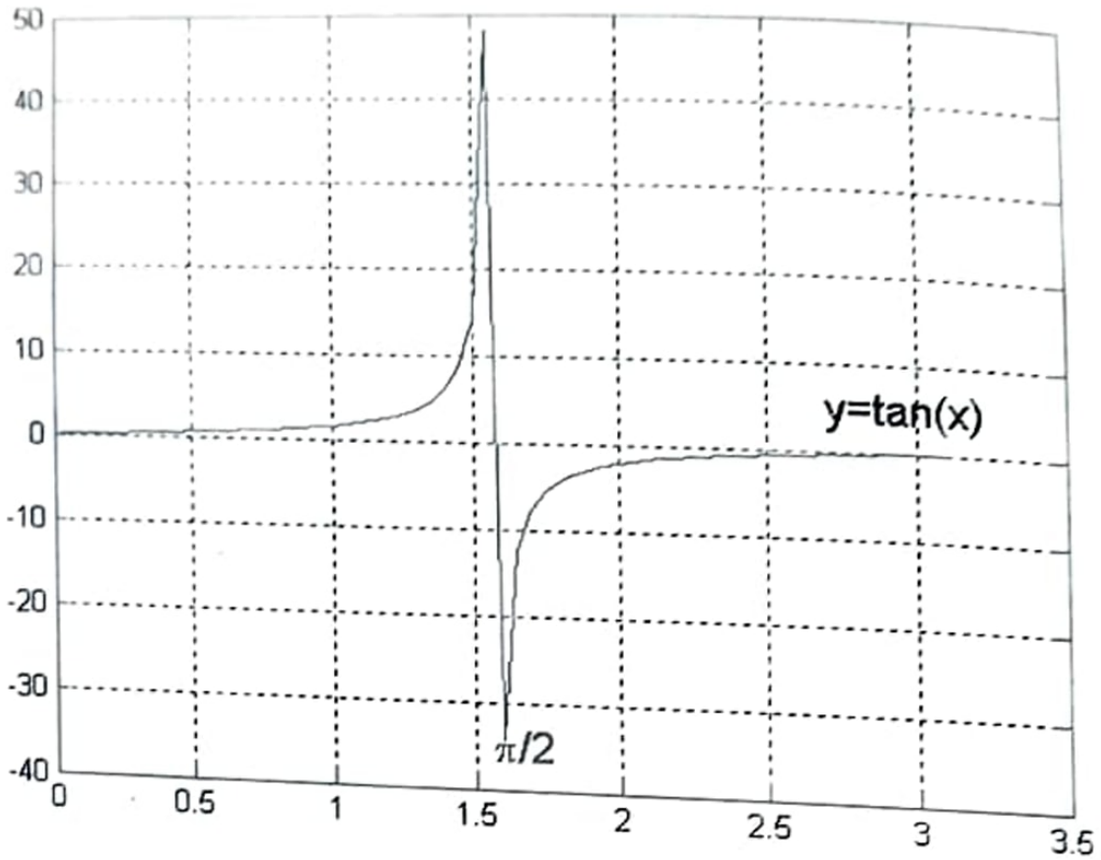
Rotation: Grafikteki yazının saat yönünün tersinde dönmesini sağlar. Dönüş açısı derece cinsindedir.
Fontsize: Grafikte verilen yazının büyüklüğünü belirler. Varsayılan değeri 10'dur.
Color: Yazının rengini belirtir.

Örnekler:

```
x = -4:.1:4;
y1=(x.^2);
y2=16-x.^2;
plot(x,y1,x,y2)
text(1.5,14,'y=16-\alpha^2','Fontsize',14)
text(-1,2,'y=\beta^2','fontsize',14)
```

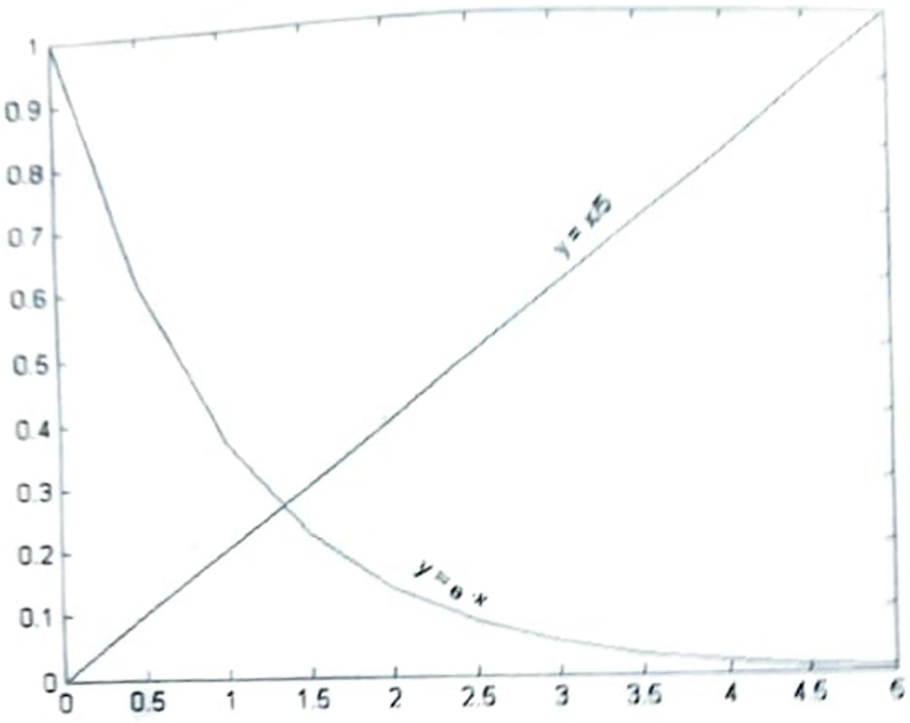


örnek 2:



örnek 3:

```
x = 0:.5:5;
y1=exp(-x);
y2=x/5;
plot(x,y1,x,y2)
text(2.2,0.18,'y = e^-^x','rotation',-40)
text(3,0.65,'y = x/5','rotation',45)
```

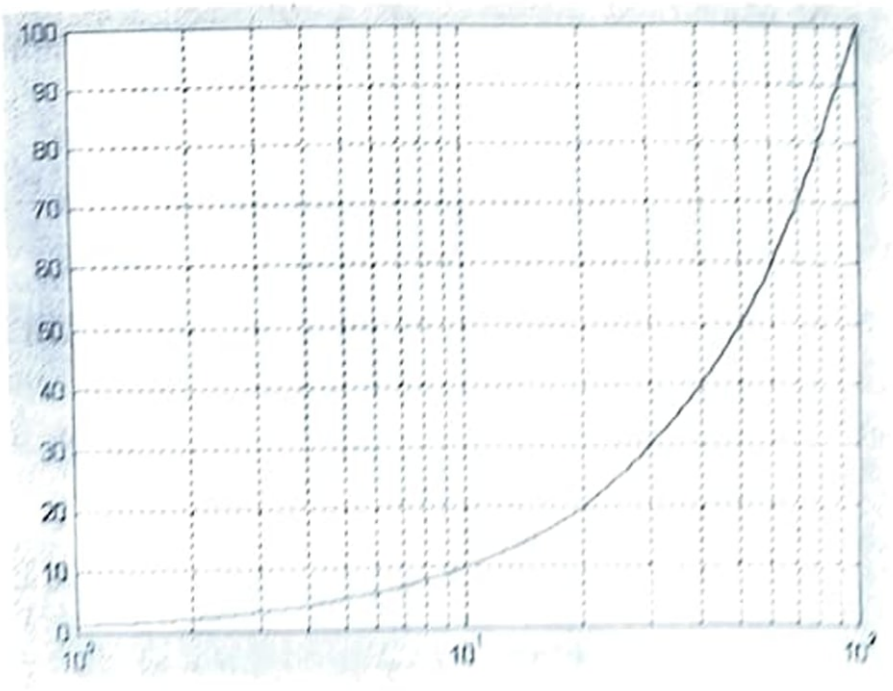


Grafiklerimi logaritmik tabanda çizmek istersek semilogx(x,y) ve semilogy(x,y) komutlarını kullanınız.

Eğer x'i logartitmik olarak, ve y'yi lineer olarak çizmek istersek semilogx(x,y) komutunu kullanınız.

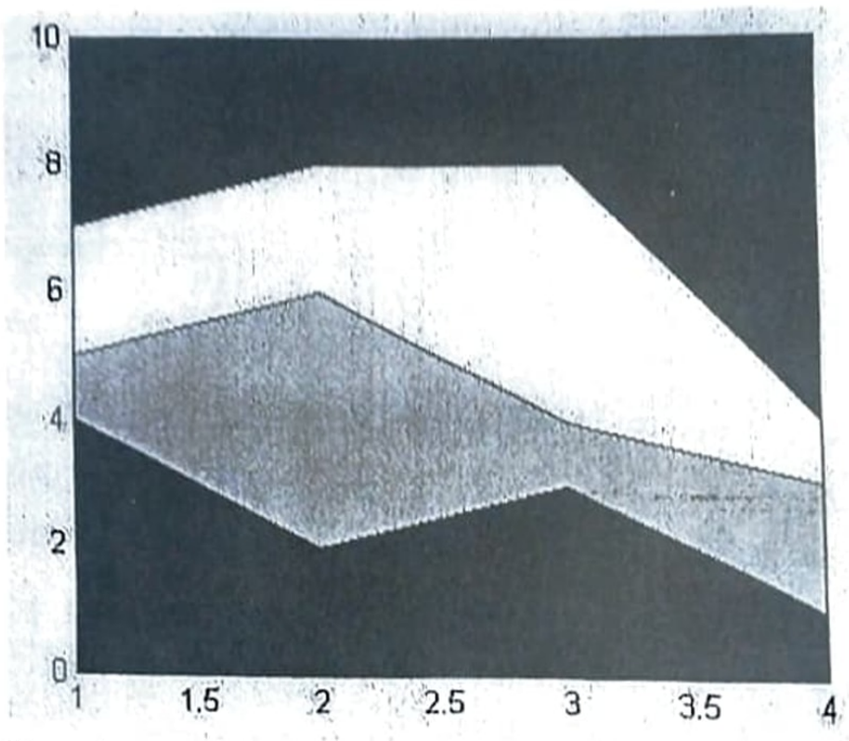
Eğer y'i logartitmik olarak, ve x'i lineer olarak çizmek istersek semilogy(x,y) komutunu kullanınız.

```
X=0:1:100;
y=x;
semilogx(x,y)
grid
```



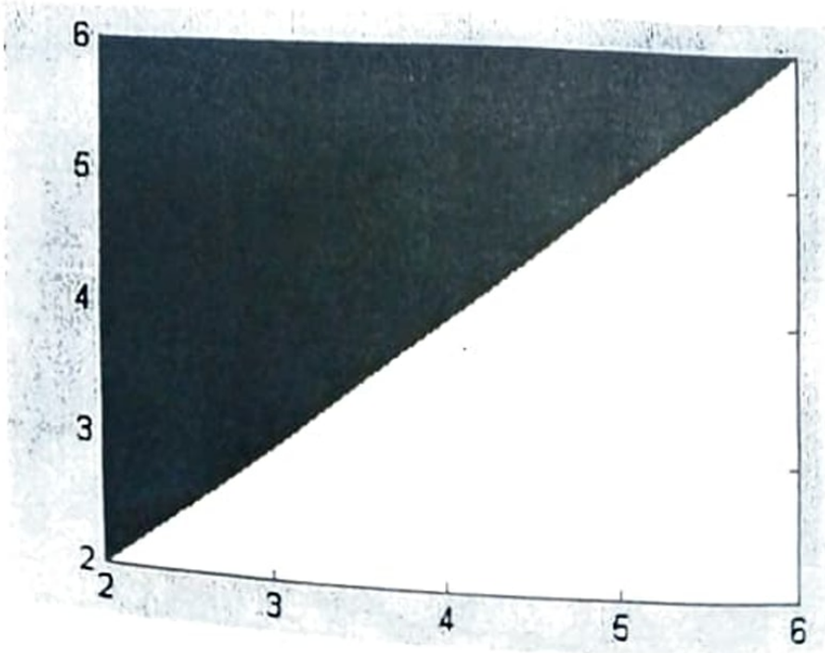
→ area(y): Bu komut çizilen grafiğin altını doldurur. Eğer birden fazla grafik çizilmiş ise grafikleri üst üste toplayarak çizer.

```
y1=[4 1 2 3];
y2=[2 4 2 2];
y3=[3 1 4 2];
y4=[1 2 1 6]
y=[y1;y2;y3;y4]
area(y)
```



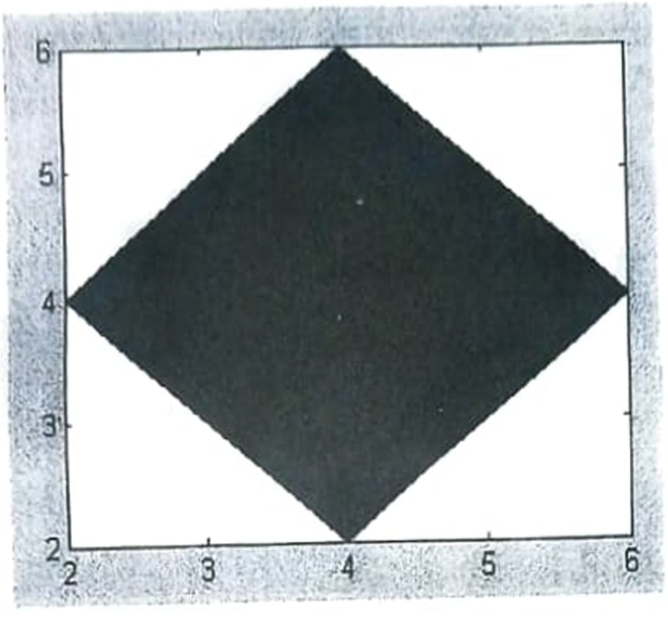
→ fill(x,y,'renk'): Bu komut çizilen grafiğin içini doldurur.

```
x=[2 2 6 ]
y=[2 6 6 ]
fill(x,y,'r')
```



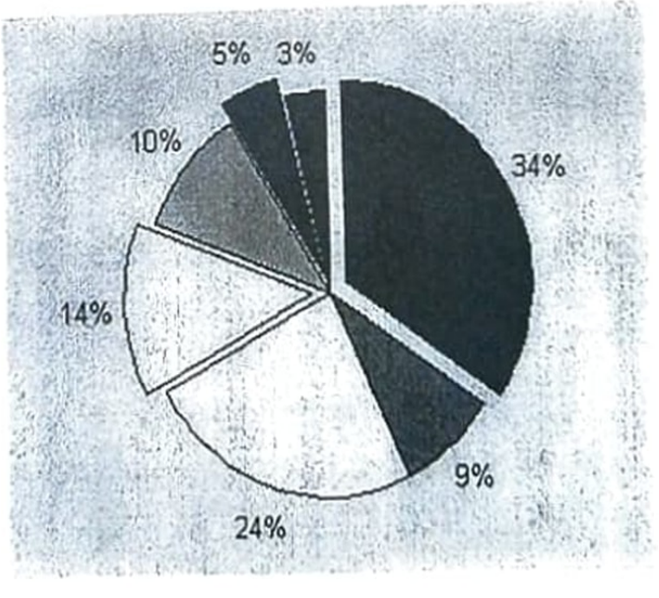
örnek:

```
x=[2 4 6 4];
y=[4 2 4 6];
fill(x,y,'b')
```



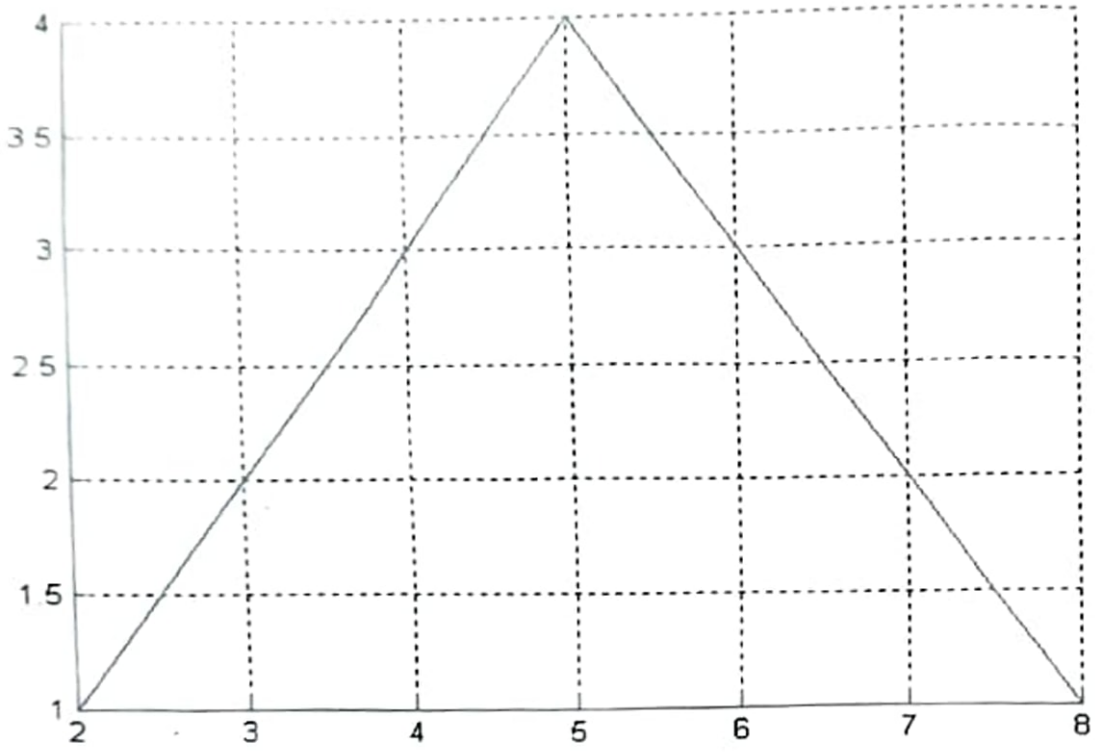
→ **pie(a,b):** Bu komut verilen a vektörün yüzdesini bir daire çizerek gösterir. B vektörüde daire dışına çıkarılacak parçaları belirtir.

```
a=[2 3 6 8 14 5 20 ]
b=[0 1 0 1 0 0 1 ]
pie(a,b)
```



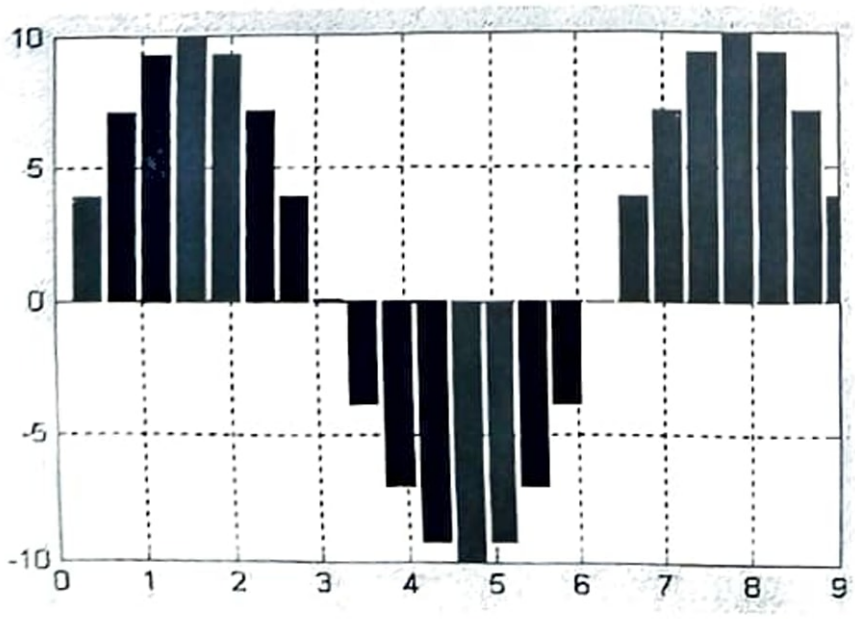
→ **line(x,y):** Bu komut verilen koordinatlarda çizgiler çizer.

```
x=[2 5 8]
y=[1 4 1]
line(x,y)
grid
```



→ bar(x,y): Bu komut plot komutuna benzer. Karşılık gelen noktalara çubuklar yerleştirir

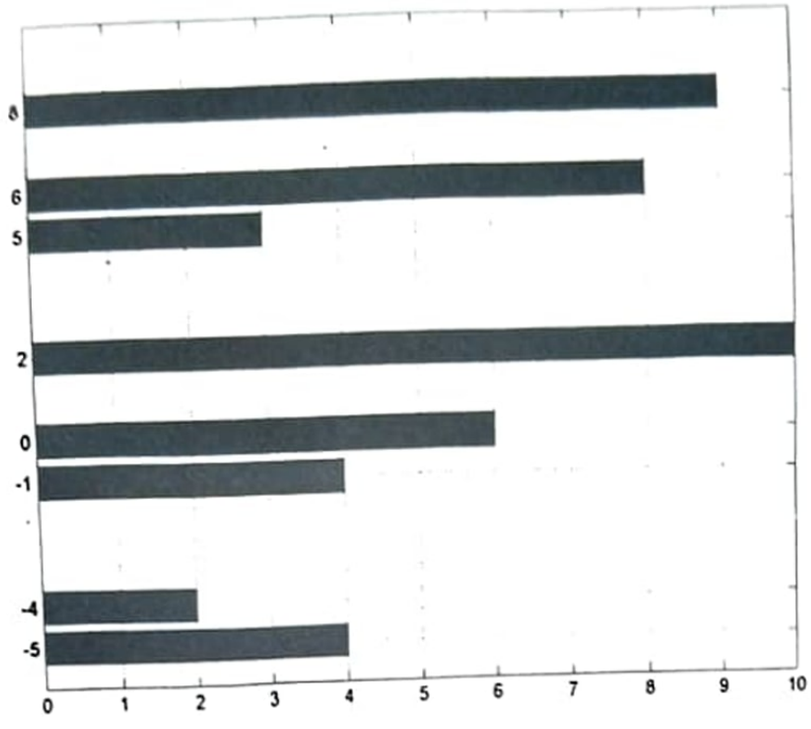
```
x=0:pi/8:3*pi;
y=10*sin(x)
bar(x,y)
axis([0 9 -10 10])
grid
```



→ barh(x,y): Bu komut çubukları yatay olarak çizer. y çubukların uzunluğunu belirtirken, x de y eksenindeki sayıları belirtir.

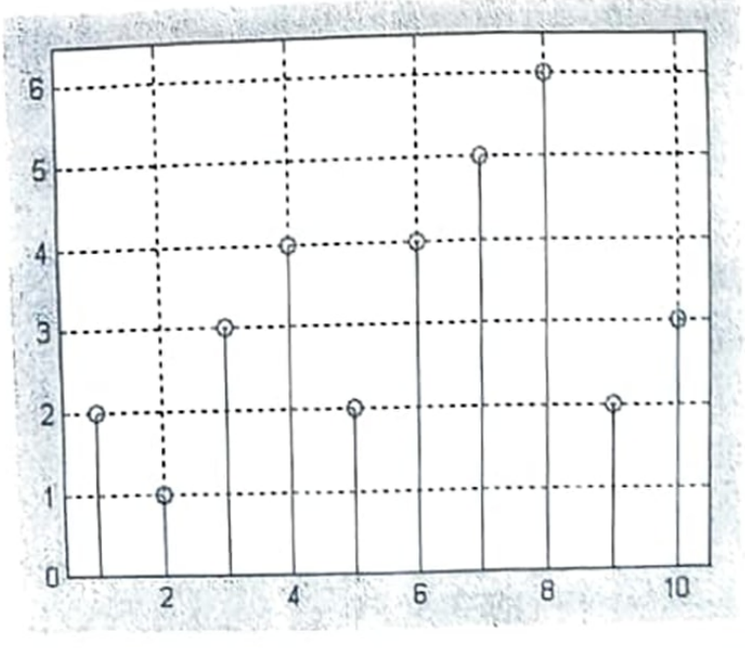
Örnek:

```
x=[-5 -4 -1 0 2 5 6 8];
y=[4 2 4 6 10 3 8 9 ];
barh(x,y)
grid
```



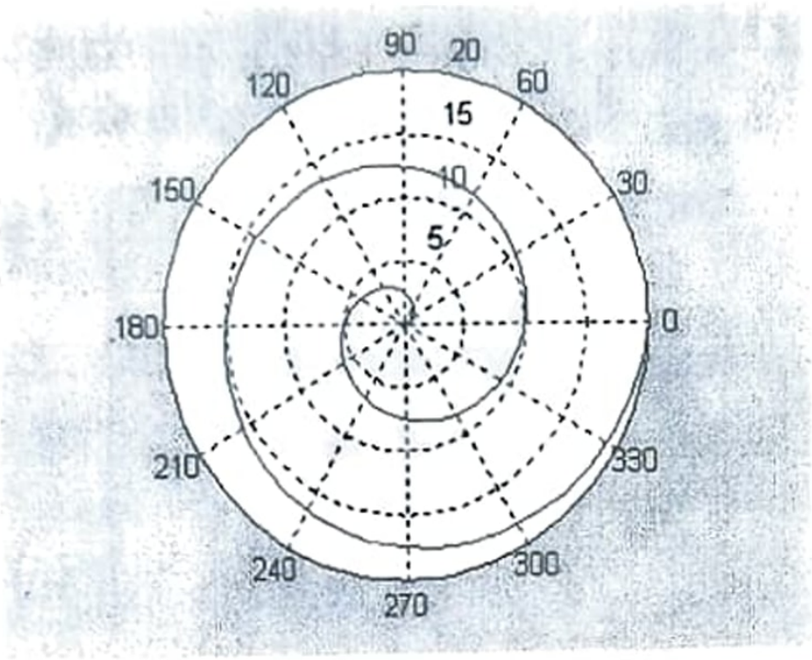
→ stem(x,y): Bu komut kesikli bilgilerin çizdirilmesi gerektiği zaman kullanılır.

```
x=1:10;
y=[2 1 3 4 2 4 5 6 2 3 ]
stem(x,y)
axis([0.5 10.5 0 6.5])
grid
```



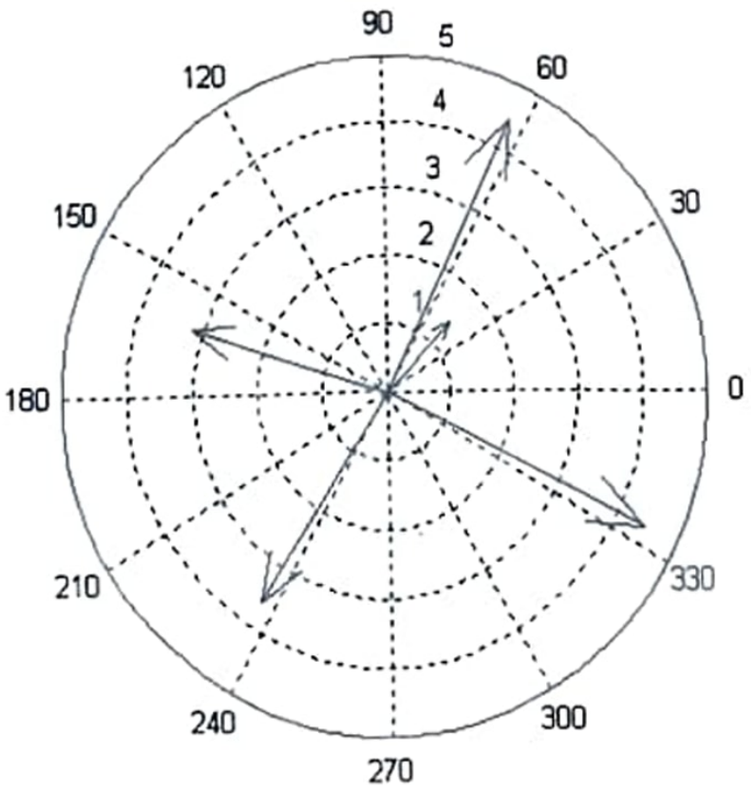
→ polar(a,b) : Bu komut verilen a açısı ve b büyüklüğüne göre polar çizim yapar.

```
b=linspace(0,20,80)
a=linspace(0,4*pi,80)
polar(a,b)
```



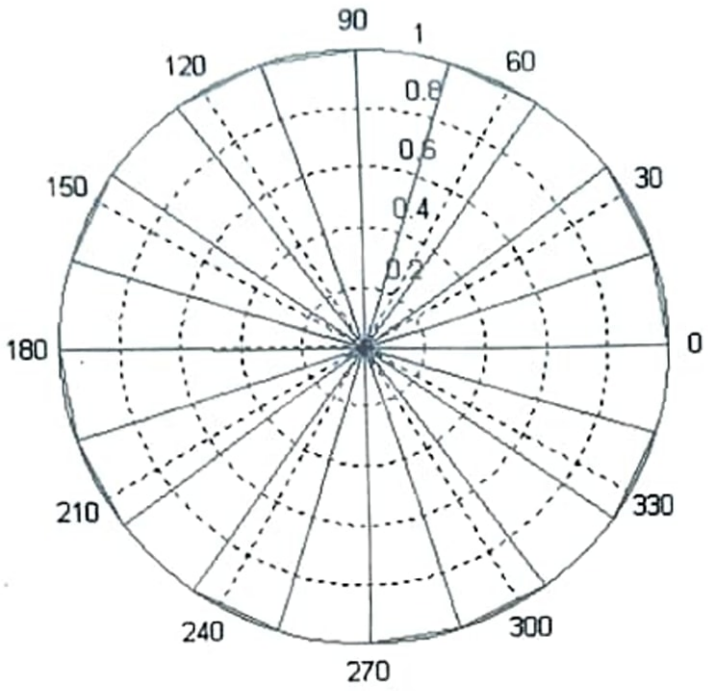
→ compass(x,y): Bu komut x ve y ile verilen koordinatlara vektör biçiminde çizim yapar.

```
x=[1 2 -3 -2 4]
y=[1 4  1 -3 -2]
compass(x,y)
```



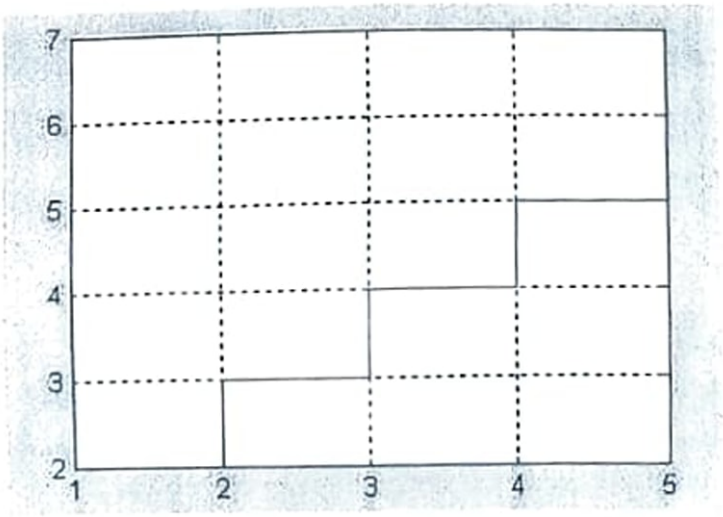
→ rose(x) : Bu komut açı histogramını çizer.

```
x=[0 30 60  90 120  150 180 210 240 270 300 330 ];
x=x*pi/180;
rose(x)
```



→ **stairs(x,y)**: Bu komut grafiği basamak şeklinde çizer.

```
x=1:5;
y=[2 3 4 5 7];
stairs(x,y)
grid
```



Animasyon Hazırlama

Matlab de animasyon hazırlama için şu komutları kullanılır.

→ getframe: Bu komut animasyon yapmak için ekrandan bir çerçeve alır.
→ movie: Bu komut alınan çerçeveleri oynatmak için kullanılır.
movie(a) :komutu a'ya atanmış olan çerçeveleri oynatır.
movie(a,b): animasyonu b kez tekrarlar.
movie(a,b,c): saniyede c defa çerçeve oynatılır. C, animasyonun hızını ayarlamak için kullanılır. c belirtilmemişse 12 kabul edilir.

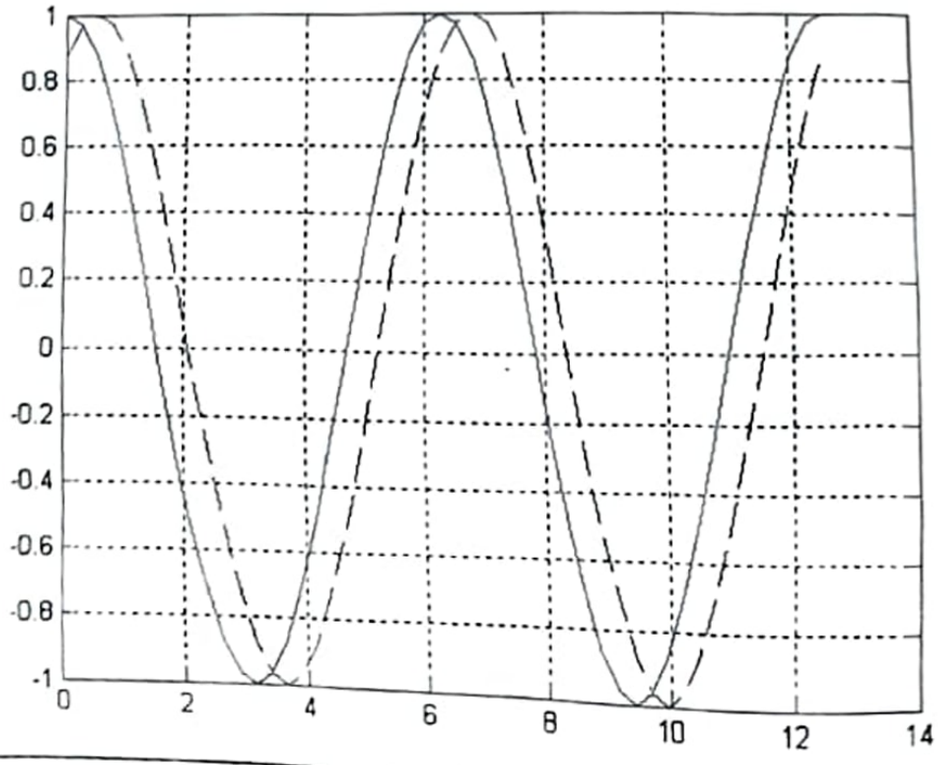
Örnek:

```
clc
j=0;
for i=0:pi/8:4*pi
t=0+i:pi/16:2*pi+i;
y=sin(t);
plot(t,y);
axis([0 12 -1.1 1.1])
grid
j=j+1;
r(j)=getframe;
end
movie(r,1,3)
```

Alıştırmalar

1) iki tane cos fonksiyonunu 30 derece faz farkı ile çizdiriniz. Cosinüslerden birini düz çizgi ile diğerini kesikli çizgi ile gösteriniz.

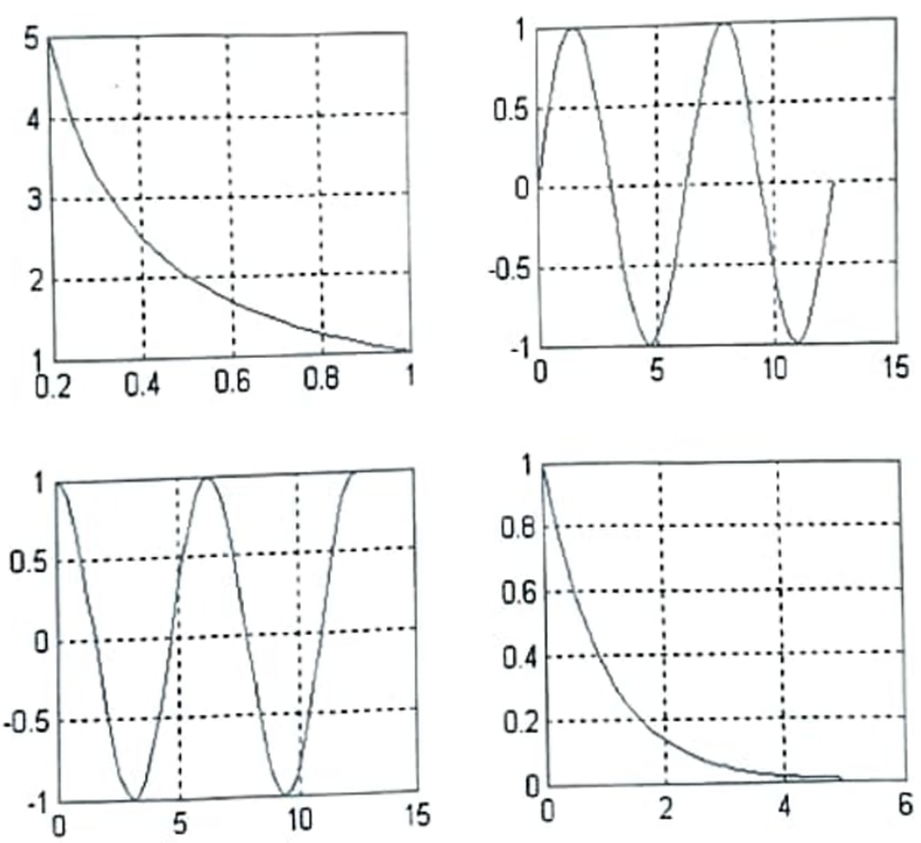
```
x = 0:pi/12:4*pi
y1=cos(x)
y2=cos(x-pi/6)
plot(x,y1,x,y2,'--')
grid
```



2) Grafik ekranını dört parçaya bölünüz ve 1. pencereye y=1/x, 2.pencereye y=sin(x), 3.pencereye y=cos(x) ve 4. pencereye $y=e^{-x}$ grafiğini çizdiriniz.

Cevap:

```
x1=.2:.05:1;
x2 = 0:pi/12:4*pi;
x3=0:.1:5;
y1=1./x1;
y2=sin(x2);
y3=cos(x2);
y4=exp(-x3);
subplot(221),plot(x1,y1),grid
subplot(222),plot(x2,y2),grid
subplot(223),plot(x2,y3),grid
subplot(224),plot(x3,y4),grid
```

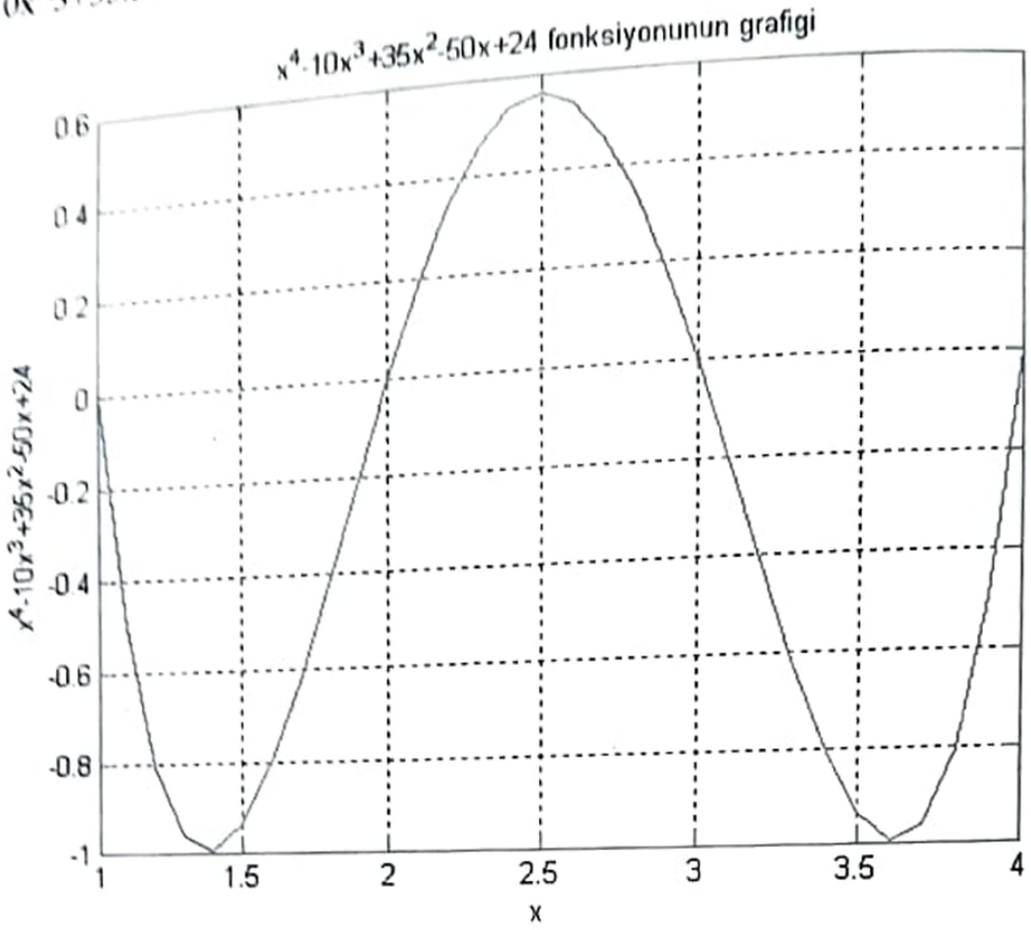


3)
$y=x^4+10x^3+35x^2-50x+24$ fonksiyonunun çizdiriniz. x eksenine 'x', y eksenine '$x^4+10x^3+35x^2-50x+24$', etiket olarak da '$x^4-10x^3+35x^2-50x+24$ fonksiyonunun grafiği' yazdırınız.

Cevap:

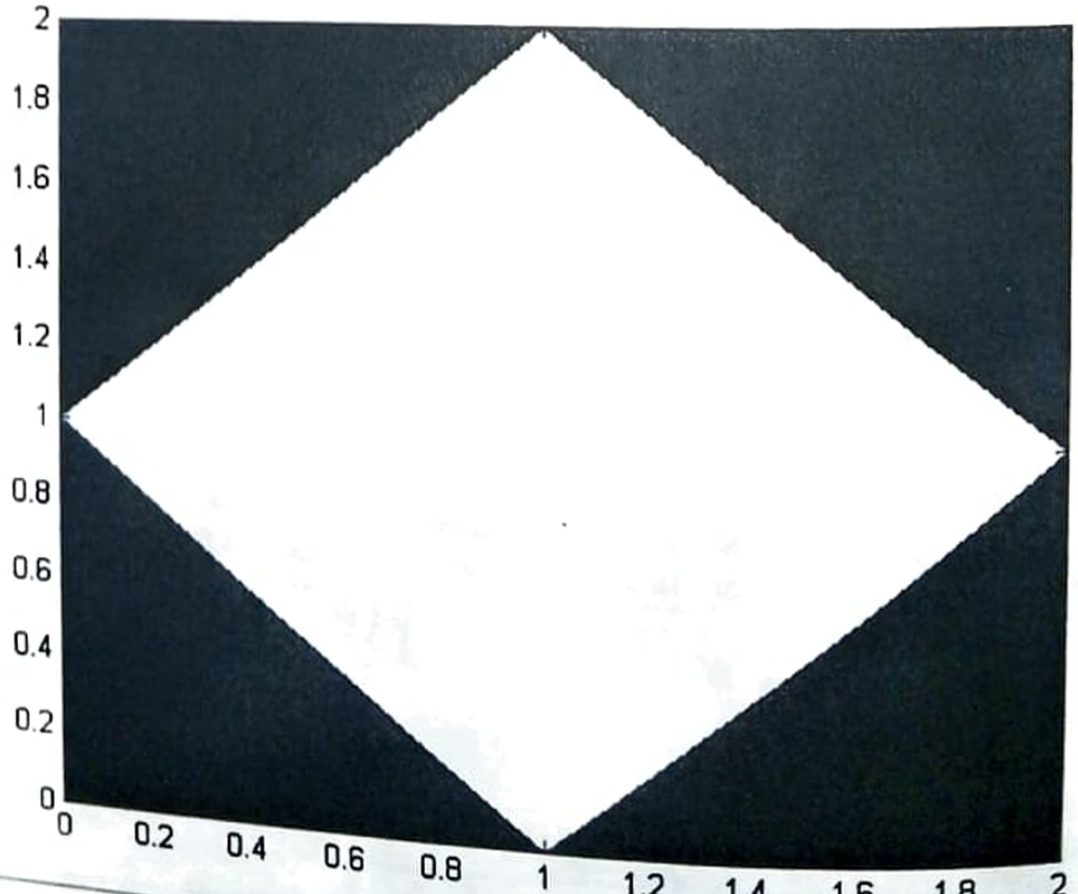
```
x=1:.1:4;
y=(x.^4-10*x.^3+35*x.^2-50*x+24)
plot(x,y)
```

```
grid
xlabel('x')
ylabel('x^4-10x^3+35x^2-50x+24')
title('x^4-10x^3+35x^2-50x+24 fonksiyonunun grafigi')
```



4)

Aşağıdaki grafiği 'fill' fonksiyonunu kullanarak elde ediniz.

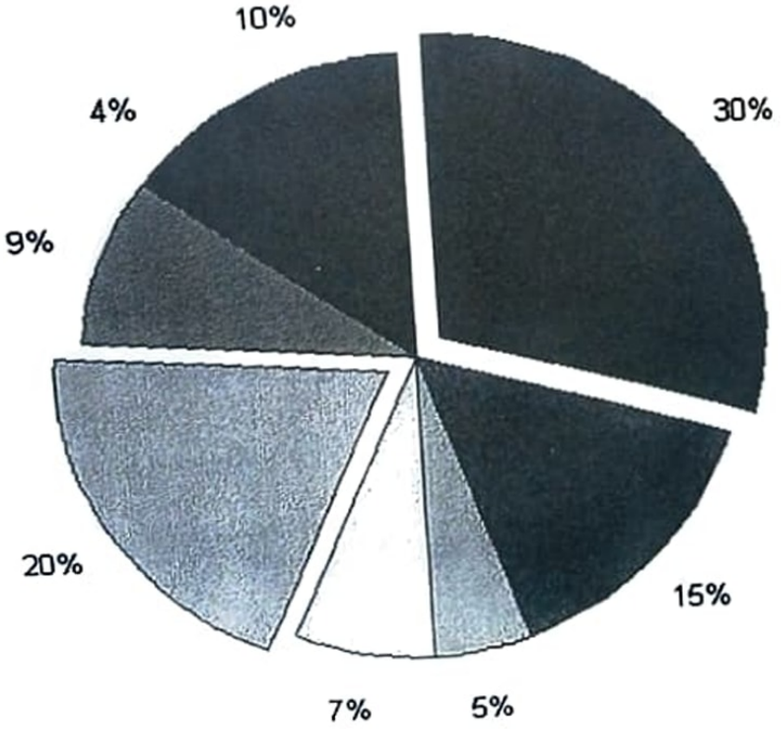


```
]
]
)]
1]
1,'r',x2,y2,'r',x3,y3,'r',x4,y4,'r')
```

lmış bir pasta grafiği çiziniz. Dilimlerin yüzdeleri 10, 4, 9, 20, 7, 5, 15, 30 olsun.
k ve %30'luk dilimleri pastadan ayırınız.

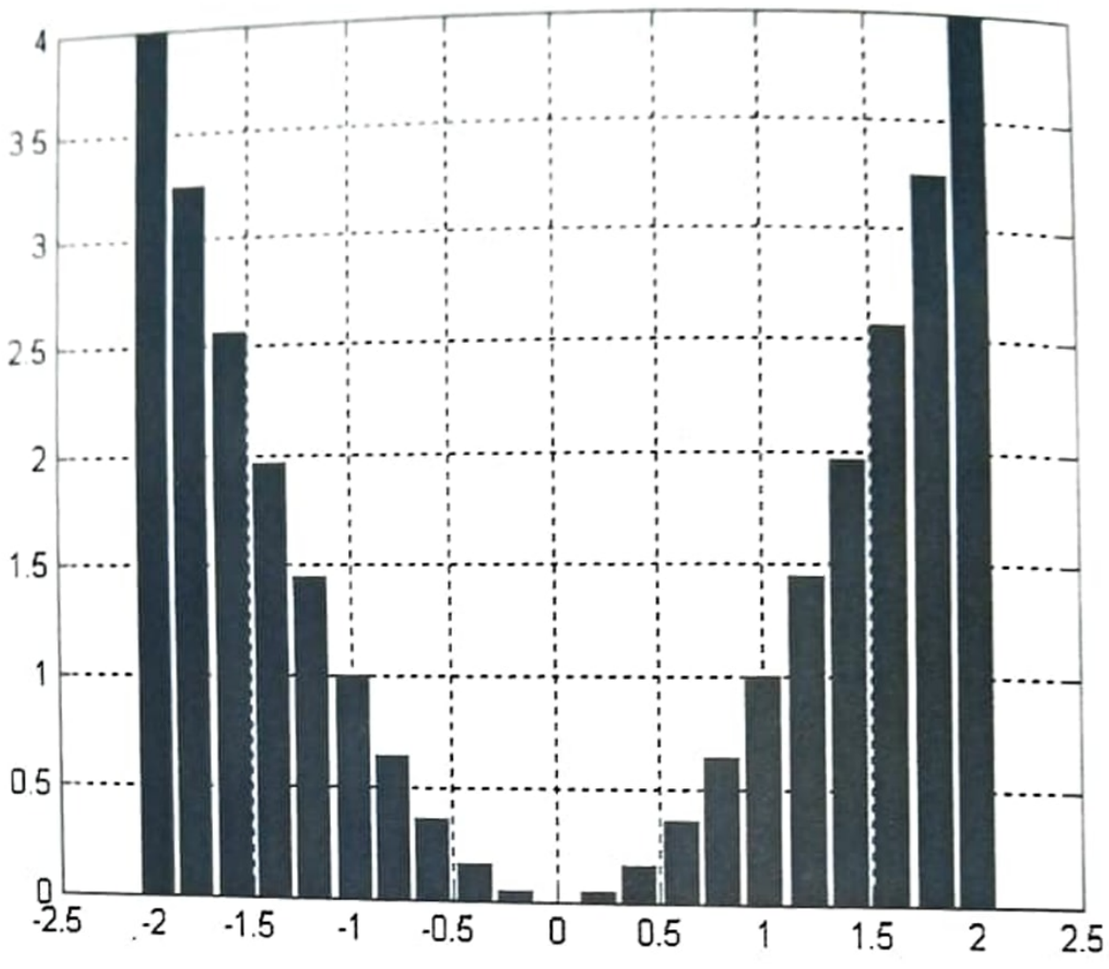
):

```
) 4 9 20 7 5 15 30]
 0 0 1 0 0 0 1] .
a,b)
```



6) $y=x^2$ fonksiyonunu –2 ile +2 arasında 'bar' komutunu kullanarak çizdiriniz.

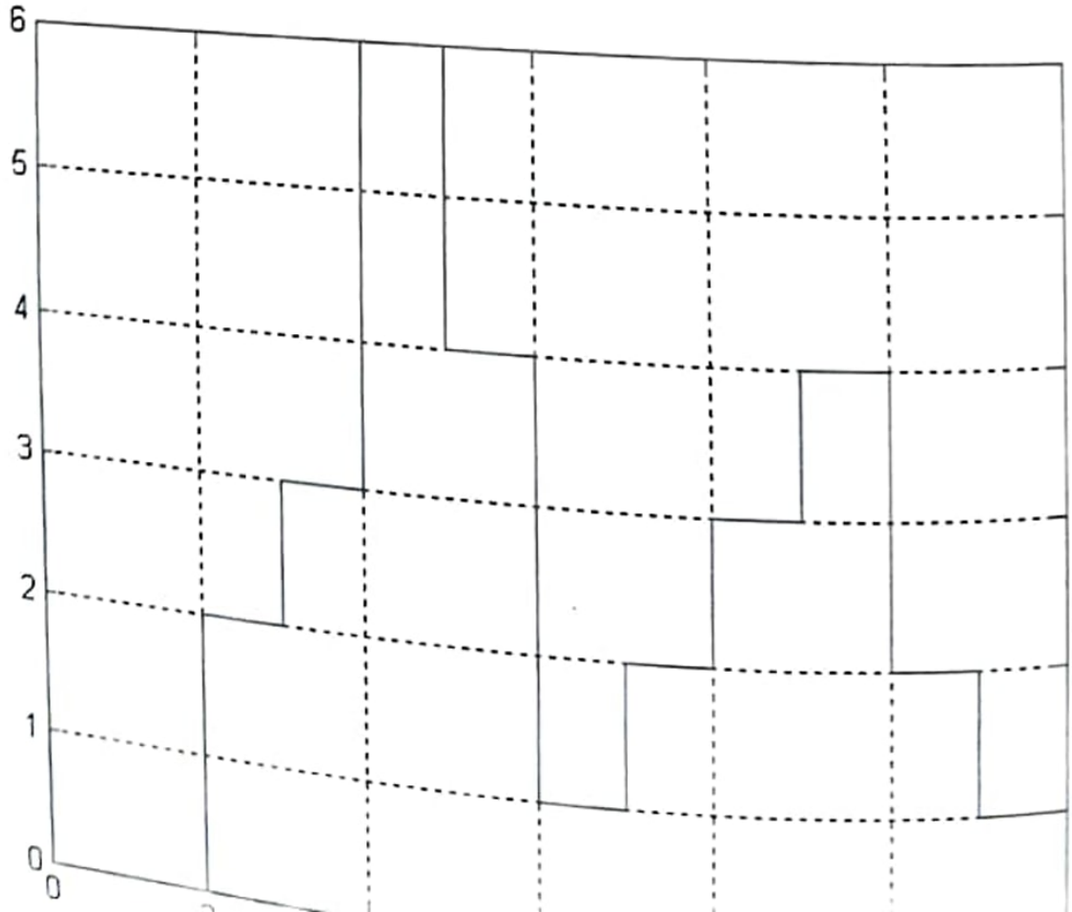
```
x=-2:0.2:2
 y=x.^2
 bar(x,y),grid
```



7) cos(x) fonksiyonunu 0 ile 4pi arasında 'stem' fonksiyonunu kullanarak çizdiriniz.

```
x=0:pi/8:4*pi;
y=cos(x)
stem(x,y),grid
```

8) Aşağıdaki grafiği 'stairs' komutunu kullanarak elde ediniz.



cevap:

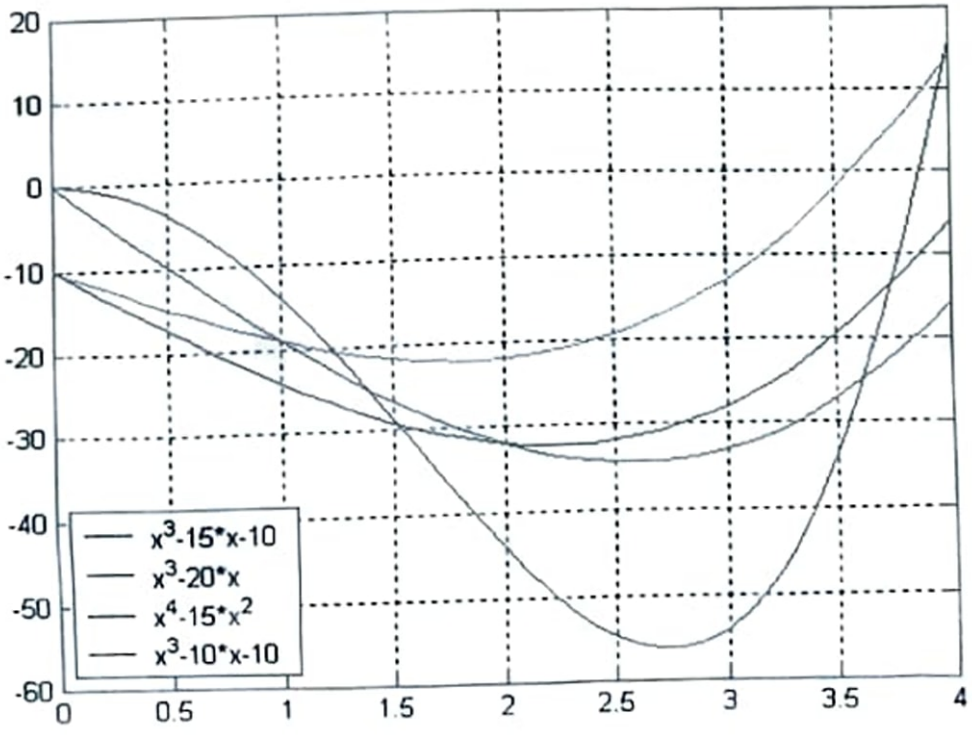
```
b=[0 2 3 6 4 1 2 3 4 2 1 0]
a=[1 2 3 4 5 6 7 8 9 10 11 12]
stairs(a,b)
grid
```

9)
Aşağıdaki fonksiyonları aynı grafik ekranına 0 ile 4 arasında çizdiriniz. Ayrıca ekranın sol alt köşesinde grafik denklemlerini ve grafik renklerini belirtiniz.

a) $y=x^3-15x-10$
b) $y=x^3-20x$
c) $y=x^4-15x^2$
d) $y=x^3-10x-10$

Cevap:

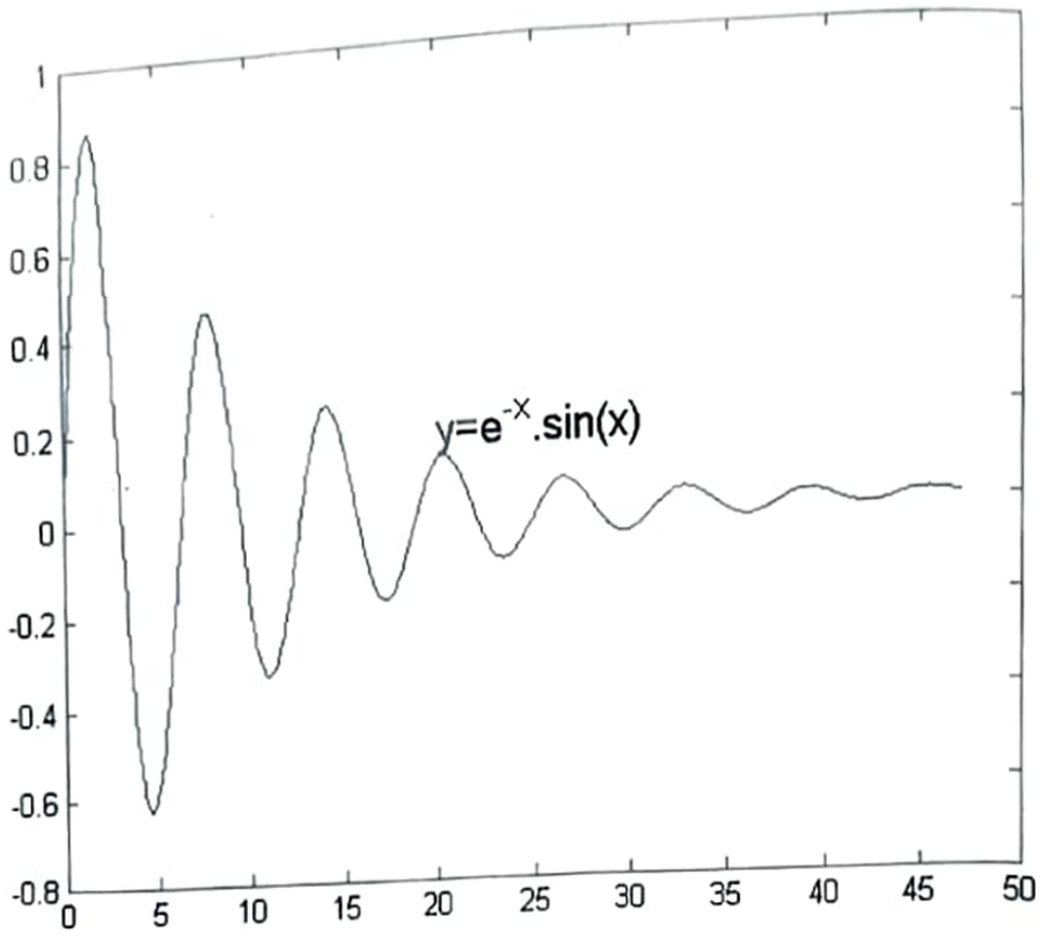
```
x=0:.1:4
y1=x.^3-15*x-10
y2=x.^3-20*x
y3=x.^4-15*x.^2
y4=x.^3-10*x-10
plot(x,y1,x,y2,x,y3,x,y4),grid
legend('x^3-15*x-10','x^3-20*x','x^4-15*x^2','x^3-10*x-10',3)
```



10) e^.sin(x) fonksiyonunu çizdiriniz ve grafiğin üzerine fonksiyonun ismini yazdırınız.

Cevap:

```
x=0:.05:15*pi
y=sin(x).*exp(-x/10)
plot(x,y)
text(20,0.2,'y=e^-^x.sin(x)','fontsize',14)
```



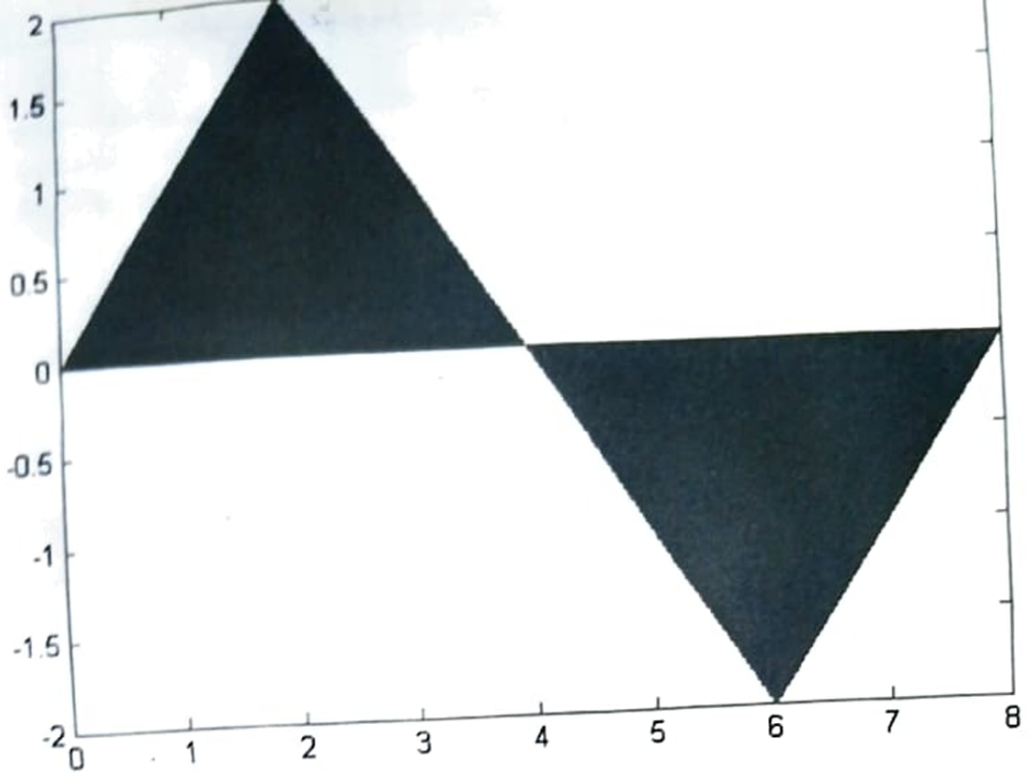
Problemler

1) y=sin(x) fonksiyonunu kırmızı renkte, '*' işareti ile çizdiriniz.

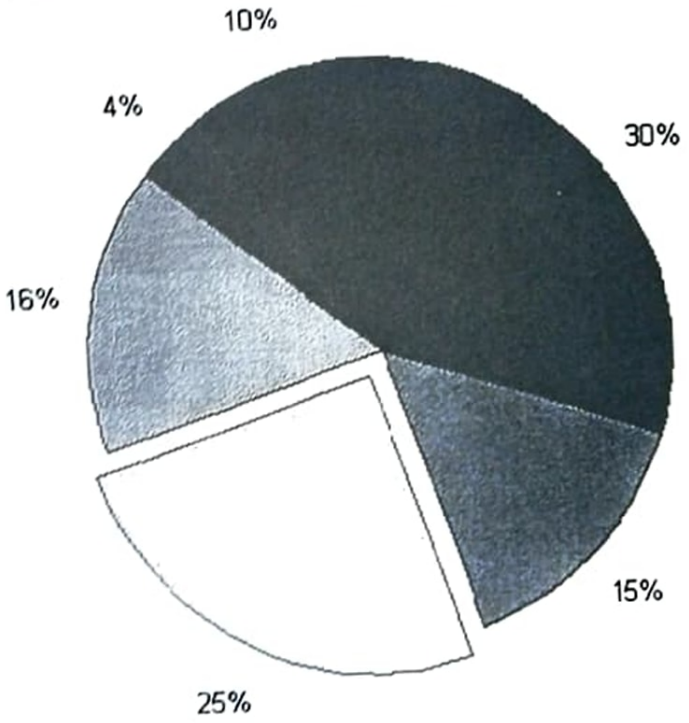
2) Grafik ekranını 4 parçaya bölünüz; 1. bölüme $y=x^2+2$, 2. bölüme $y=x^3-8$, 3. bölüme $y=x^2-7x+5$, 4. bölüme x^3-8x^2+5 grafiğini çizdiriniz.

3) y=sinc(x) fonksiyonunu 0.05 artırımlarla –2pi ile +2pi arasında çizdiriniz. Grafiğin x eksenine 'x', grafiğin y eksenine 'y=sinc(x)' ve grafiğin başlık kısmına da 'y=sinc(x) fonksiyonun grafiği' yazdırınız.

4) Aşağıdaki grafiği fill komutunu kullanarak elde ediniz.

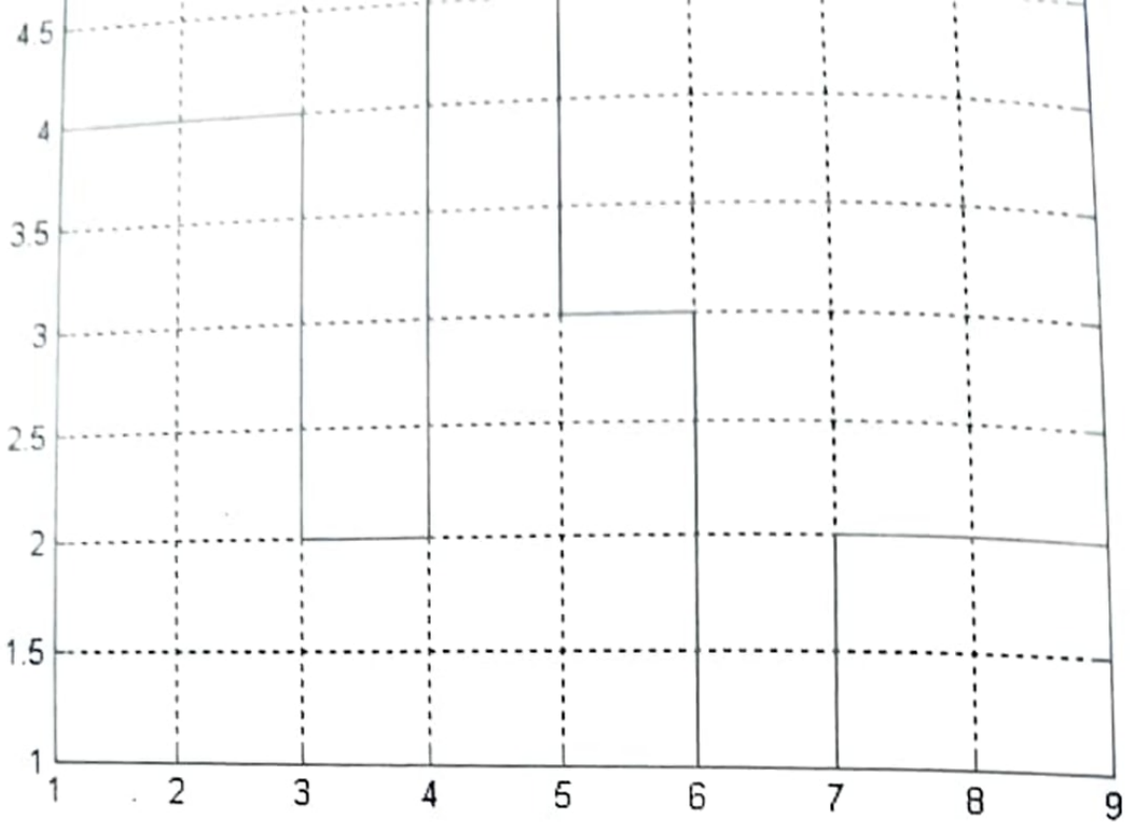


5) Aşağıdaki grafiği 'pie' komutunu kullanarak elde ediniz.



6) y=x fonksiyonunu 1 ile 10 arasında bar komutunu kullanarak çiziniz.

7) Aşağıdaki grafiği 'stairs' komutunu kullanarak çiziniz.



8) Aşağıdaki fonksiyonları aynı grafik ekranına 0 ile 5 arasında 0.1 artırımlarla çizdiriniz. Ayrıca ekranın sağ üst köşesinde grafiklerin denklemlerini ve grafik renklerini belirtiniz.

a) $y=12.\sin(x)$
b) $y=x^3-3x^2-8x+10$
c) $y=15\text{sinc}(x)$
d) $y=x^4-8x^3+16x^2-10$

9) $y=x.\sin(x)$ fonksiyonunu 0 ile 30 arasında 0.1 arttırışlarla çiziniz ve grafiğin üzerine $x.\sin(x)$ yazınız.

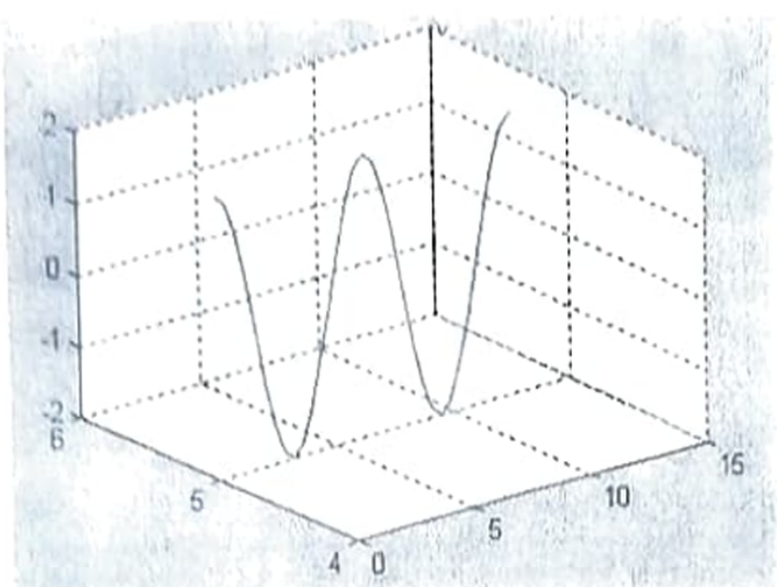
ÜÇ BOYUTLU GRAFİK ÇİZİMİ

Matlab programında üç boyutlu grafikler çizmek mümkündür. Bu grafikler üzerinde renk ve ışık ayarı yapılabilir. Grafiklerin istenilen açı miktarınca yerlerini değiştirmek mümkündür. Matlab'ın bu özelliklerinden yararlanılarak çeşitli olayların daha kolay yorumlanması sağlanır.

Komutlar:

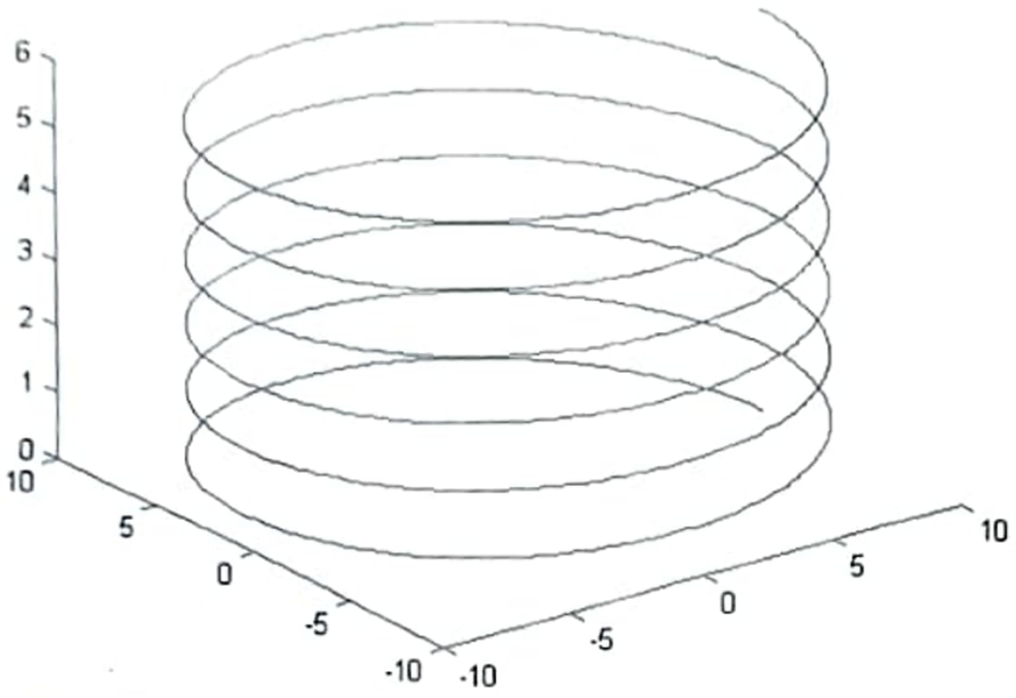
→ **plot3(x,y,z)** : Bu komut x ve y değerlerine karşılık gelen z değerlerini çizer.

```
x=0:pi/8:4*pi;
z=2*cos(x);
for i=1:length(x)
   y(1,i)=5
end
plot3(x,y,z)
grid
```



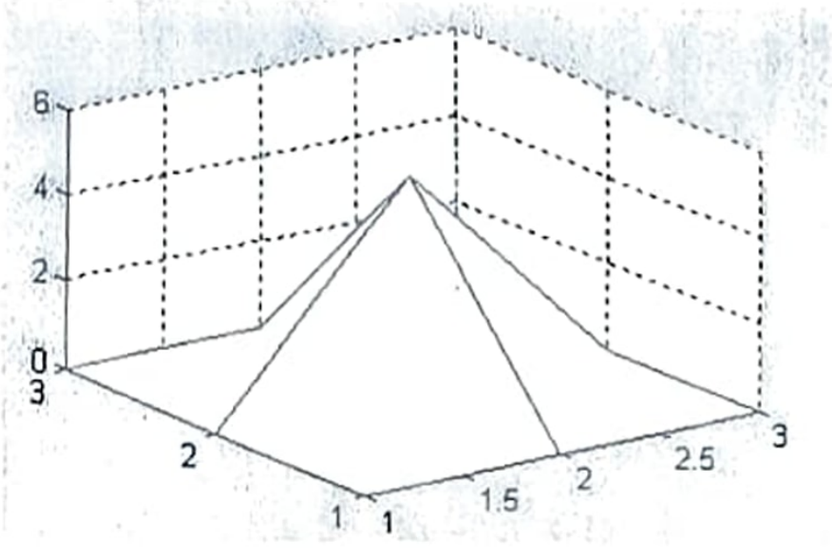
Örnek:

```
x=0:pi/128:4*pi;
x1=10*cos(x);
y1=10*sin(x);
z1=linspace(0,2,513);
plot3(x1,y1,z1)
x=4*pi:pi/128:8*pi;
x2=10*cos(x);
y2=10*sin(x)
z2=linspace(2,4,513);
plot3(x1,y1,z1,x2,y2,z2)
x=8*pi:pi/128:12*pi;
x3=10*cos(x);
y3=10*sin(x);
z3=linspace(4,6,513);
plot3(x1,y1,z1,x2,y2,z2,x3,y3,z3)
```



→ **mesh(x,y,z):** Bu komut verilen x ve y noktalarına karşılık gelen z değerlerini çizer. Çizme işlemi komşu noktalara birleştirme şeklinde olur. z'nin değeri değiştikçe renkte de değişme olur. Aşağıda basit bir örnekle bu işlemin nasıl yapıldığı gösterilmiştir. x ve y vektör ise z matris olmalıdır. z matrisinin satır sayısı y'nin eleman sayısına, z matrisinin sütun sayısı x'in eleman sayısına eşit olmalıdır.

```
x=1:3
y=1:3
z=[0 0 0
   0 5 0
   0 0 0]
mesh(x,y,z)
```

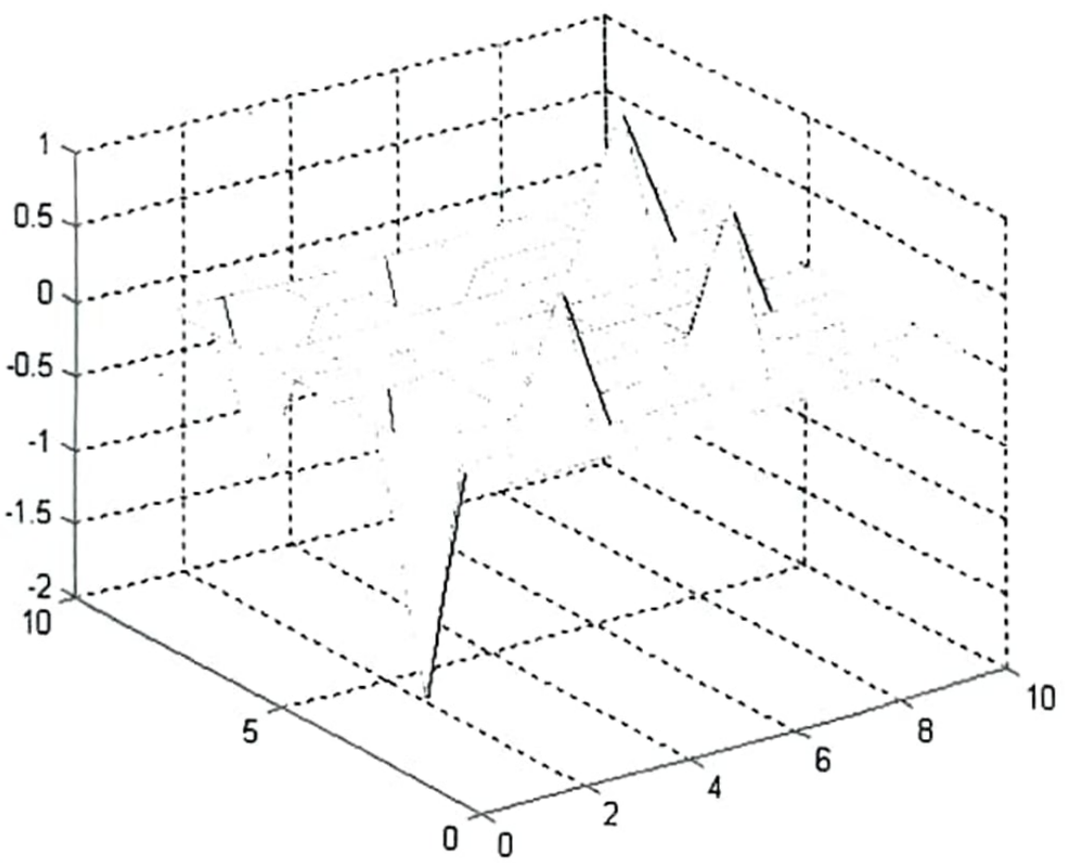


Örnek:

```
x=1:10
y=1:9
```

```
z=[0 0 0 0 0 0 0 0 0 0 0
  0 0 1 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 1 0 0 0 0
  0 -2 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 1 0 0 0
  0 -1 0 0 -1 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0]
```

mesh(x,y,z)



→ **surf(x,y,z):** Bu komut mesh(x,y,z) komutuna benzer. Bu komuttan farkı, renkli yüzeyler oluşturmasıdır.

örnek:

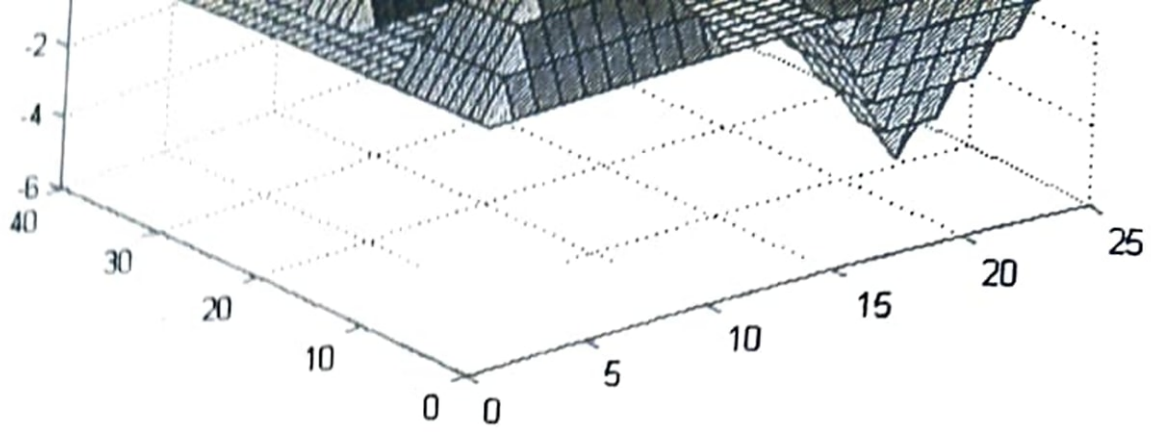
```
x=1:25;
y=1:38;
z=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0
  0 1 2 2 2 2 2 2 2 1 0 0 0 0 0 -1 -2 -2 -2 -2 -2 -2 -2 -1 0
  0 1 2 3 3 3 3 3 2 1 0 0 0 0 0 -1 -2 -3 -3 -3 -3 -3 -2 -1 0
  0 1 2 3 4 4 4 3 2 1 0 0 0 0 0 -1 -2 -3 -4 -4 -4 -3 -2 -1 0
  0 1 2 3 4 5 4 3 2 1 0 0 0 0 0 -1 -2 -3 -4 -5 -4 -3 -2 -1 0
  0 1 2 3 4 4 4 3 2 1 0 0 0 0 0 -1 -2 -3 -4 -4 -4 -3 -2 -1 0
  0 1 2 3 3 3 3 3 2 1 0 0 0 0 0 -1 -2 -3 -3 -3 -3 -3 -2 -1 0
  0 1 2 2 2 2 2 2 2 1 0 0 0 0 0 -1 -2 -2 -2 -2 -2 -2 -2 -1 0
  0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
 0 0 0
```

→ **surfl(x,y,z):**Bu komut surf(x,y,z) komutuna benzer. Bu komuttan farkı, ışığın geliş yönüne göre renk değiştirmesidir.

Örnek:

```
x=1:25;
y=1:38;
z=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0
  0 1 2 2 2 2 2 2 2 1 0 0 0 0 0 -1 -2 -2 -2 -2 -2 -2 -2 -2 -1 0
  0 1 2 3 3 3 3 3 2 1 0 0 0 0 0 -1 -2 -3 -3 -3 -3 -3 -2 -1 0
  0 1 2 3 4 4 4 3 2 1 0 0 0 0 0 -1 -2 -3 -4 -4 -4 -3 -2 -1 0
  0 1 2 3 4 5 4 3 2 1 0 0 0 0 0 -1 -2 -3 -4 -5 -4 -3 -2 -1 0
  0 1 2 3 4 4 4 3 2 1 0 0 0 0 0 -1 -2 -3 -4 -4 -4 -3 -2 -1 0
  0 1 2 3 3 3 3 3 2 1 0 0 0 0 0 -1 -2 -3 -3 -3 -3 -3 -2 -1 0
  0 1 2 2 2 2 2 2 2 1 0 0 0 0 0 -1 -2 -2 -2 -2 -2 -2 -2 -2 -1 0
  0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 1 2 2 2 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 1 2 3 2 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 1 2 2 2 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 1 1 1 1 1 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -0 0 -1 -1 -1 -1 -1 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -0 0 -1 -5 -5 -5 -1 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -0 0 -1 -5 -6 -5 -1 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 -5 -5 -5 -1 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 -1 -1 -1 -1 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
surfl(x,y,z)
```

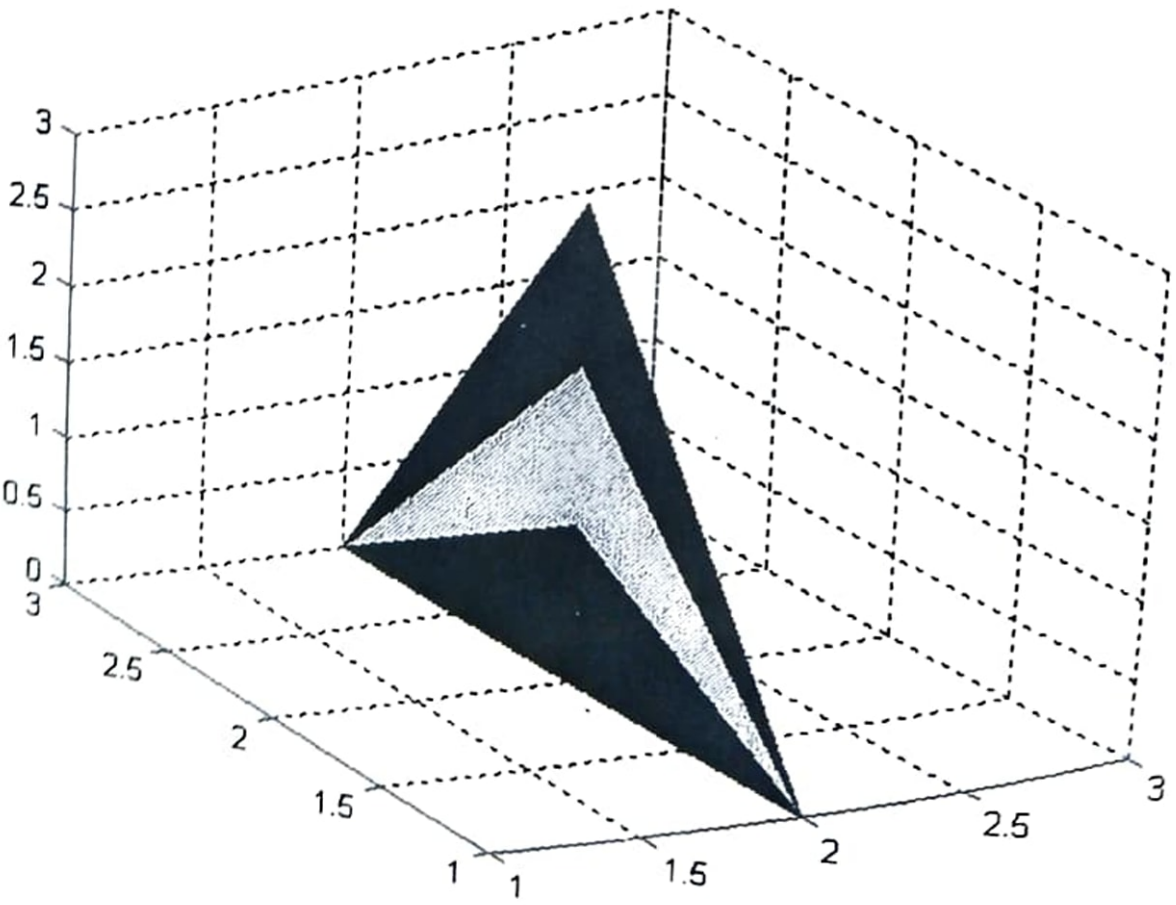


→fill3(x,y,z,c):

x,y,z koordinatları ile belirtilen şeklin içini doldurur.

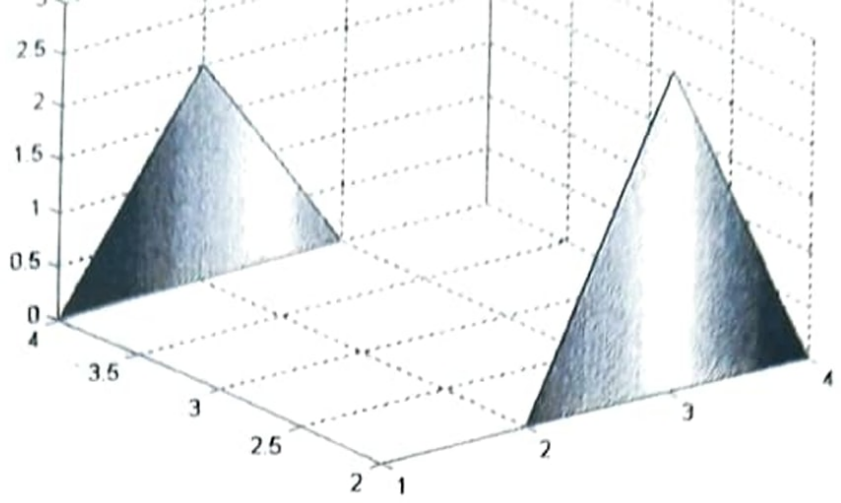
Örnek:

```
x = [2 2 2]
y = [1 2 3]
z = [0 0 0;1 2 3;0 0 0]
c = [1 2 3;1 2 3;1 2 3]
fill3(x,y,z,c)
grid
```



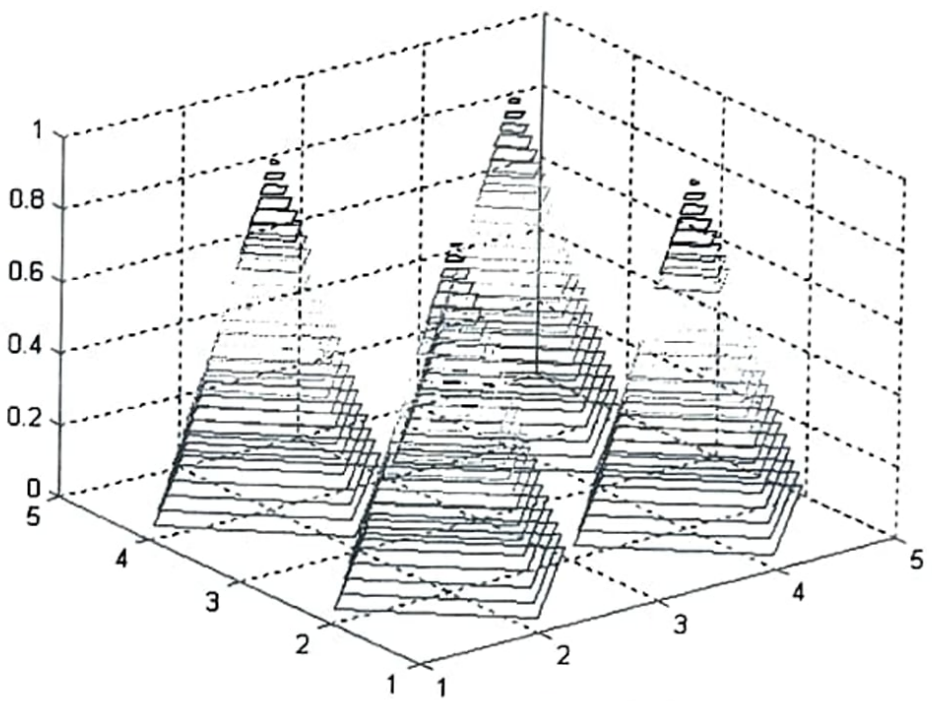
Örnek:

```
clc
x = [1 2 ;2 3;3 4]
y = [4 2 ;4 2;4 2]
z = [0 0 ;2 3; 0 0]
```



→contour3(a):
Bu grafik a matrisine göre çevrimler gösterir. A matrisinin elemanlarının değeri yüksekliği gösterir. Matrisin boyutu en az 2x2 olmalıdır.
contour3(a,n) komutunda ise n, çevrimlerin sayısını gösterir.

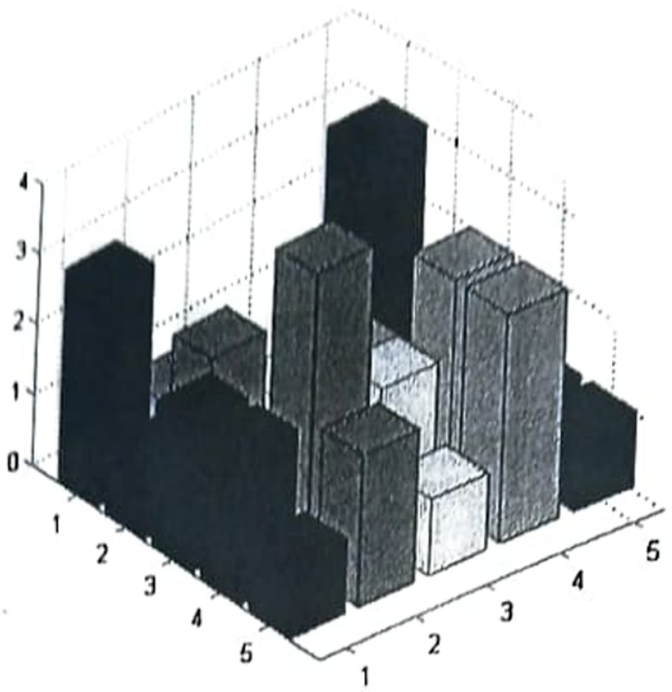
```
a=[0 0 0 0 0;
  0 1 0 1 0
  0 0 0 0 0
  0 1 0 1 0
  0 0 0 0 0 ]
contour3(a,25)
```



→ bar3(a) komutu verilen matris değerlerini 3 boyutlu çubuklar olarak çizer.
örnek:

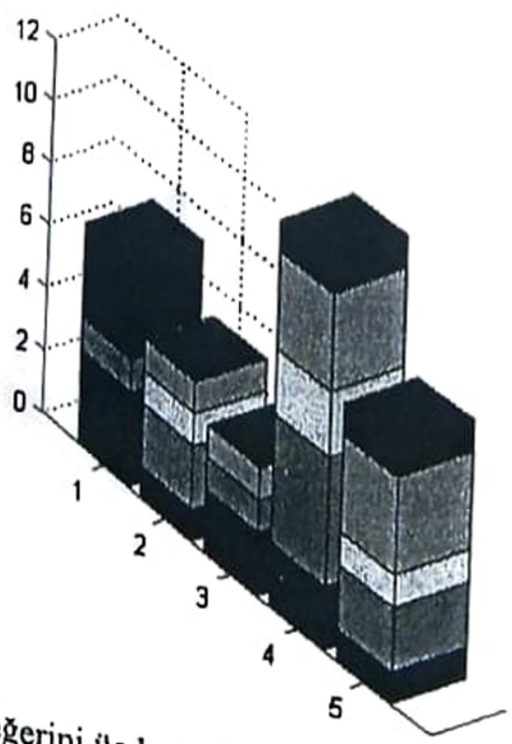
```
x=[ 3 1 0 0 3
   1 2 1 1 0
   2 1 0 1 0
   2 4 2 3 1
```

```
 1 2 1 3 1]
bar3(x)
```



bar3(a,genişlik) komutu ile çizilen çubukların genişliğini değiştirmek mümkündür. Genişlik 0 ile 1 arasında değişir. Matlab tarafından varsayılan değer 0.8'dir.
bar3(a,'stil') komutu ile çizilen çubukların stilini değiştirmek mümkündür.
Stil olarak 'detached' seçilirse grafikler ayrı ayrı çiziler, 'grouped' seçilirse grafikler satır satır seçilir, 'stacked' seçilirse çubuklar üst üste seçilir.
Örnek:

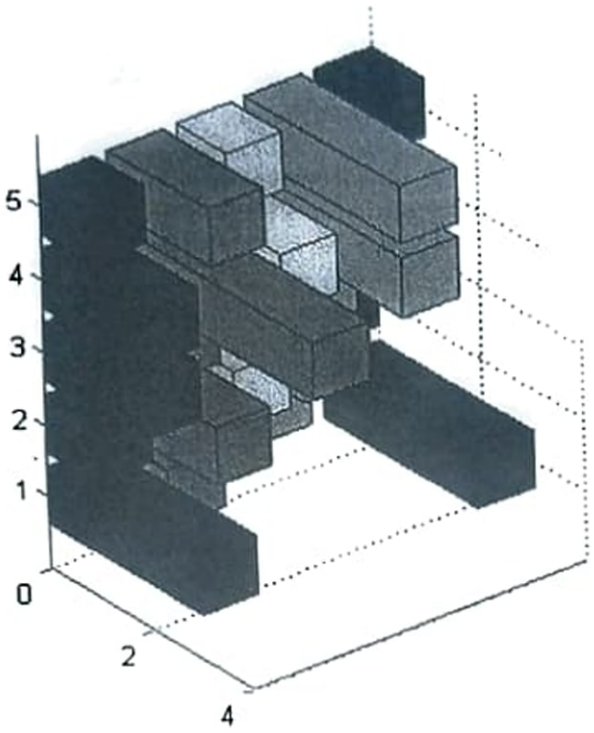
```
x=[ 3 1 0 0 3
   1 2 1 1 0
   2 1 0 1 0
   2 4 2 3 1
   1 2 1 3 1]
bar3(x,'stacked')
```



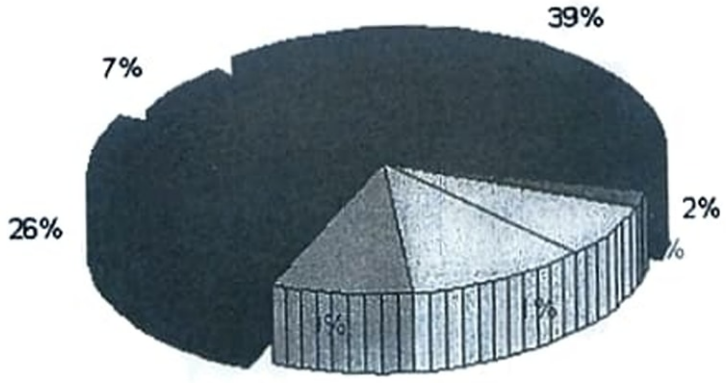
→ **bar3h(x):** verilen matris değerini üç boyutta yatay olarak çizer.
örnek:

```
x=[ 3 1 0 0 3
   1 2 1 1 0
   2 1 0 1 0
   2 4 2 3 1
```

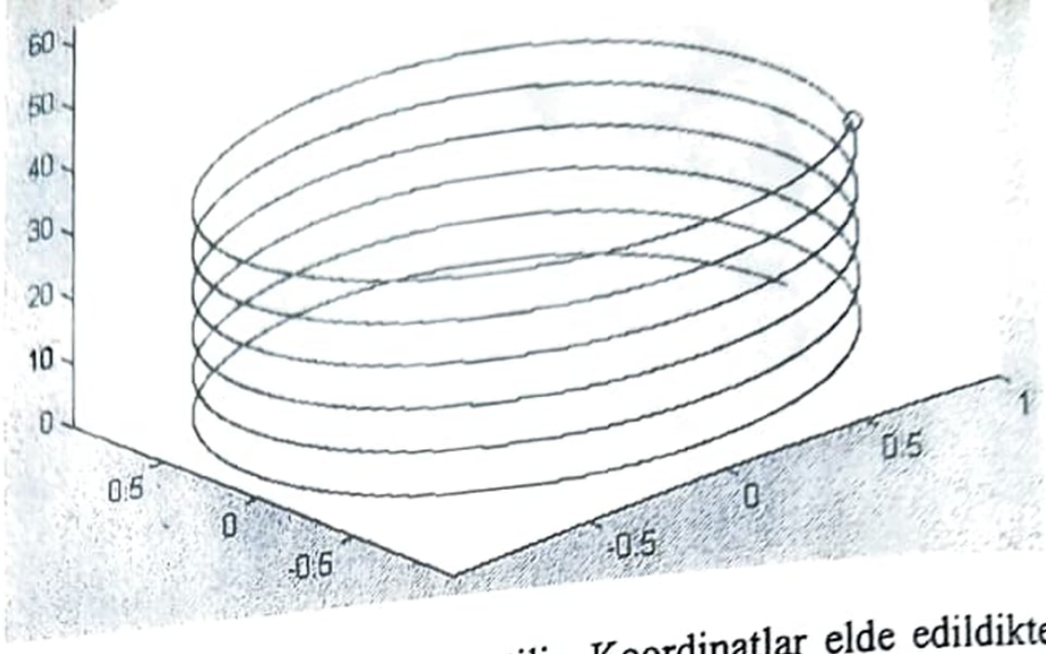
1 2 1 3 1]
bar3h(x)



→ pie3(x,y) :
Bu komut ile 3 boyutlu olarak pasta grafiği çizilir. x dizisi dilimlerin büyüklüğünü, y dilimi de pastadan ayrılan dilimleri gösterir.
Örnek:
x=[3 12 4 5 3 1 18]
y=[0 1 0 0 0 0 1]
pie3(x,y)



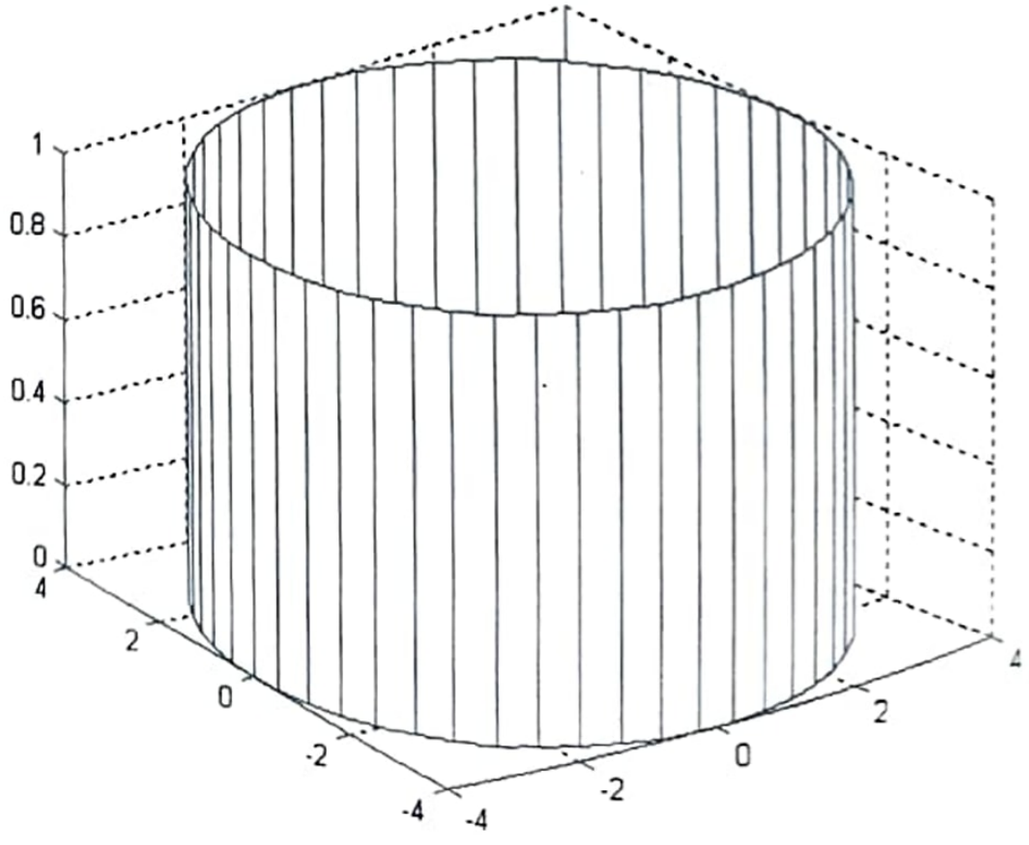
comet3(x,y,z) komutu 3 boyutlu olarak grafiği animasyon şeklinde çizer.
örnek:
t=0:0.005:20*pi;
x=cos(t);
y=sin(t);
z=t;
comet3(x,y,z)



→ **cylinder:**
Bu komut ile birim silindirin koordinatlar üretilir. Koordinatlar elde edildikten sonra çizi
yapılabilir.

Kullanımı:
[x,y,z] = cylinder
Silindirin yarıçapını değiştirmek mümkündür. Bunun için yarıçapı belirtmek gerekir.
Kullanımı:
[x,y,z]=cylinder(r)
Silindiririn normalde daire çevresi boyunca 20 eşit parçaya bölünür. Parça sayısını arttırm
veya azaltmak mümkündür.
Kullanımı:
[x,y,z]=cylinder(r,n)
Örnek:

```
[x,y,z] = cylinder(4,50);
mesh(x,y,z)
```



kullanımı: sp

Bu komut bir küre çizer. Küre de 20x20 yüz bulunmaktadır. Küredeki yüz

arttırmak veya azaltmak mümkündür.

Kullanımı: Sphere(n)

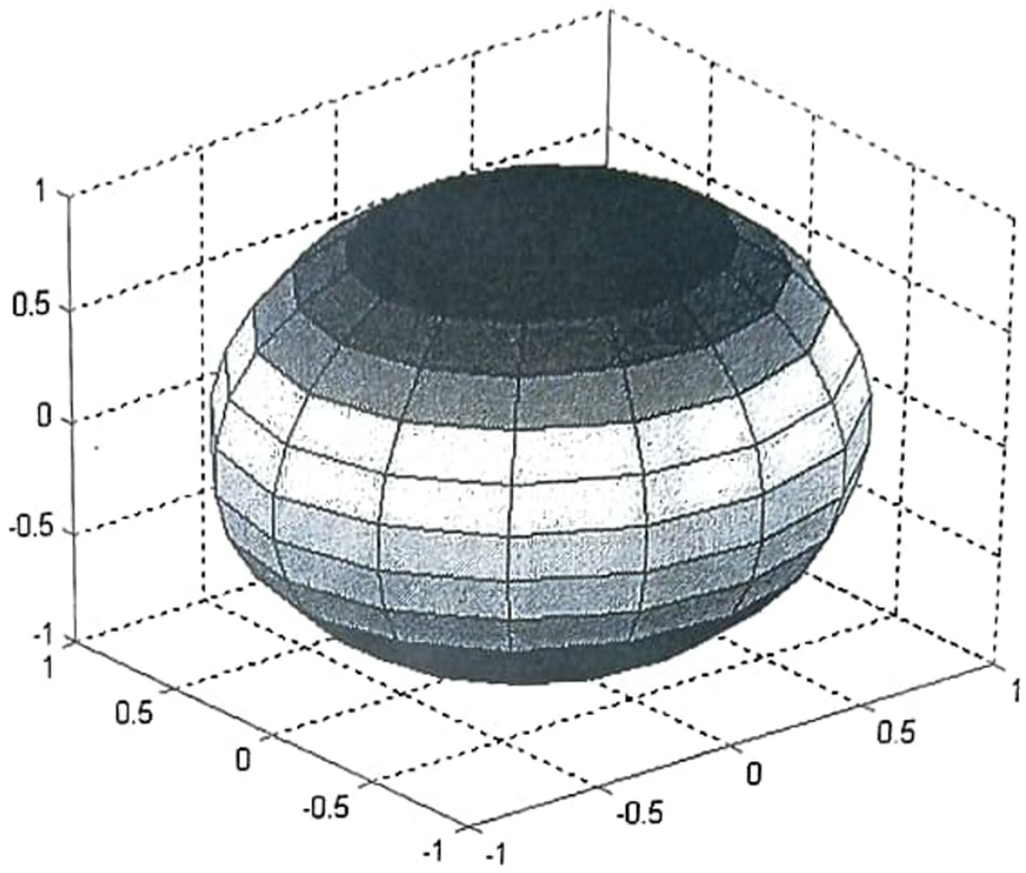
Bu durumda nxn boyutluk bir küre cizilir.

Kürenin koordinatlarını x,y,z cinsinden almak mümkündür.

Kullanımı: [x,y,z]=sphere(n)

Örnek:

sphere(15)



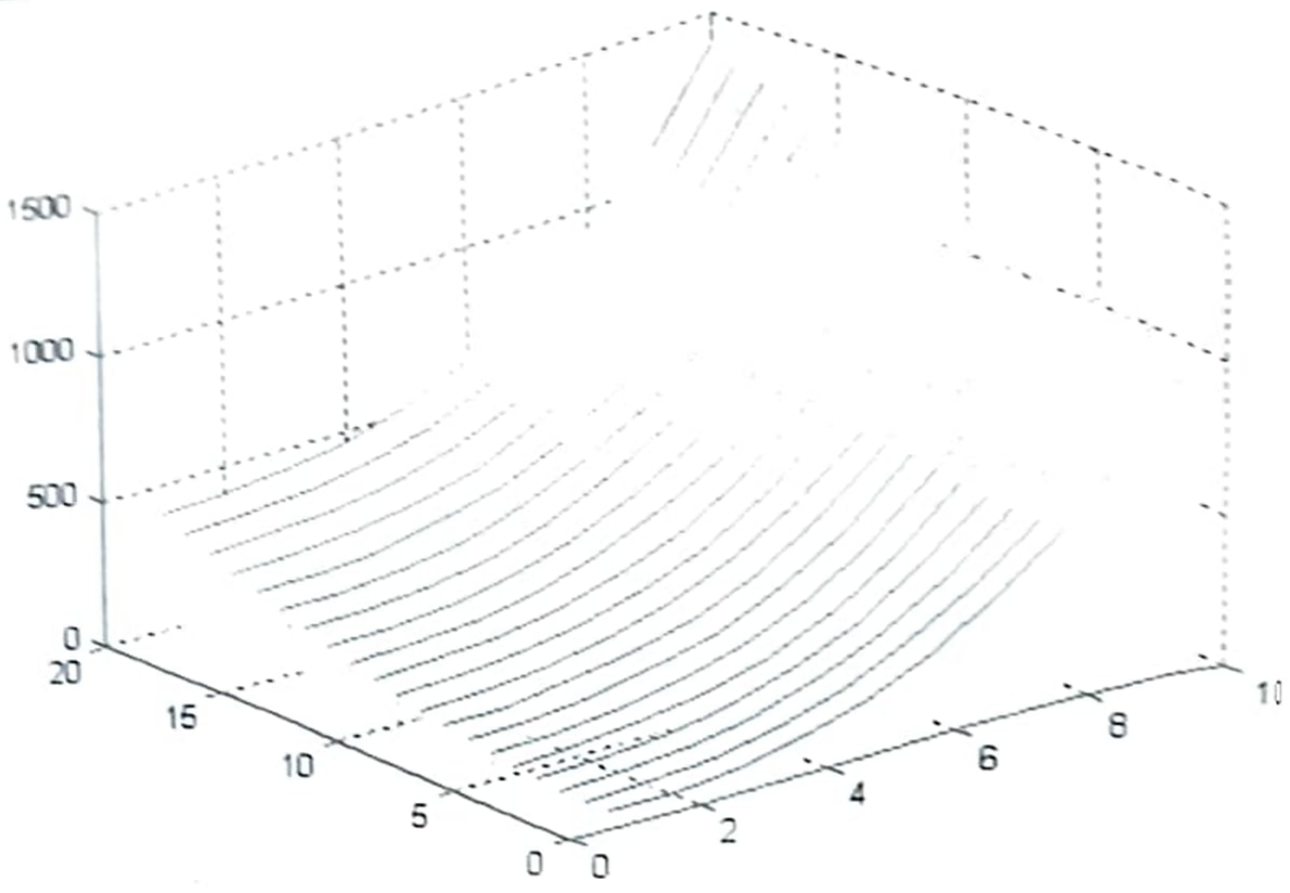
→ waterfall:

Bu komut şelaleden su akışını andıran bir grafik üretir. Ortaya çıkan grafik mesh komutu

çizilen grafiğe benzer. Yalnız mesh komutundan farklı olarak sütunlar birleştirilmez.

Örnek:

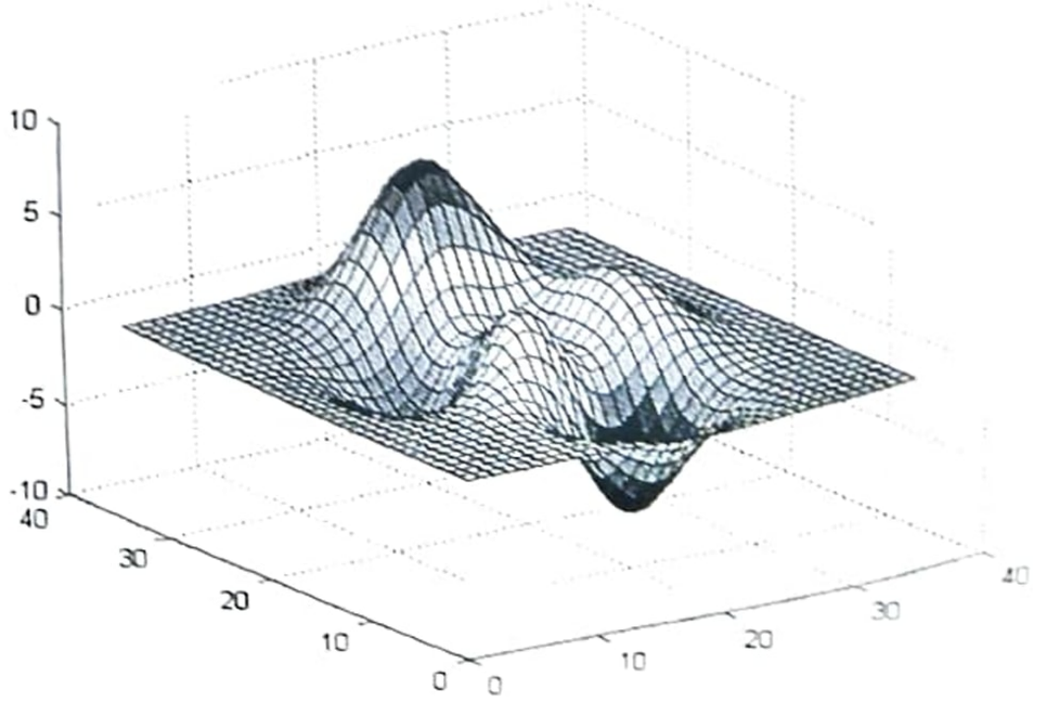
```
clc
   for y=1:20
      for x=1:10
         z(y,x)=x.^3+y.^2;
      end
   end
x=1:10
y=1:20
   waterfall(x,y,z)
```



→**peaks :** iki değişkenin fonksiyonudur. Gauss dağılımından elde edilir.
Kullanımı
a=peaks. Bu durmda 49x49 boyutunda bir matris üretilir.
Matrisin boyutunu değiştirmek mümkündür.
Kullanımı: a=peaks(N)
Bu durumda NxN boyutumda bir matris üretilir.

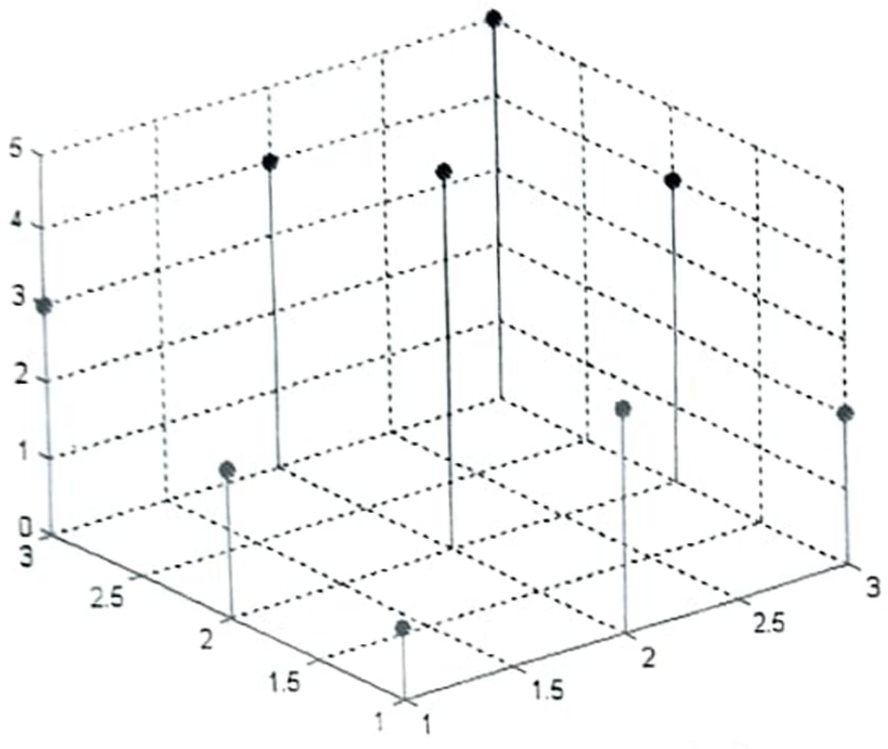
Örnek:

```
a=peaks(35)
surf(a)
```



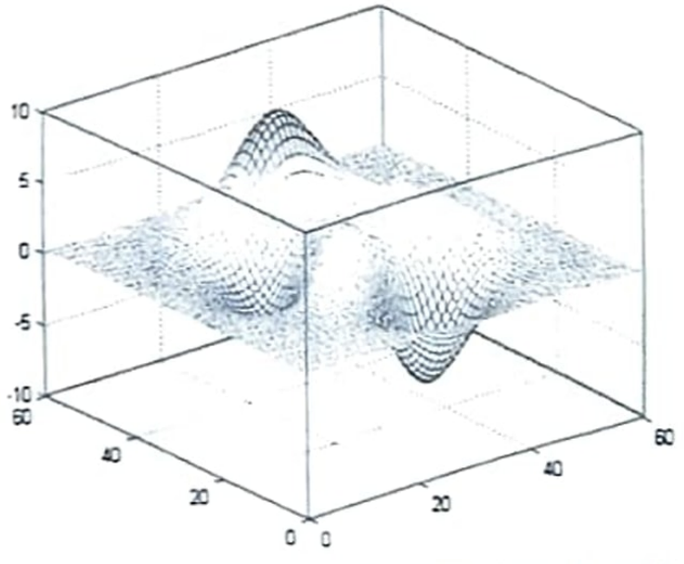
→ **stem(x,y,z):** x ve y'ye karşılık gelen z değerlerini çizgiler şeklinde gösterir. Eğer komut stem3(x,y,z,'fill') şeklinde kullanılırsa çizgilerin içindeki çemberin içi doldurulur.

```
x=[1 2 3;1 2 3;1 2 3]
y=[1 1 1;2 2 2;3 3 3]
z=[1 3 2;2 5 4;3 4 5]
stem3(x,y,z,'fill')
```



→ **box :** Bu komut eksenlerin gösterilip gösterilmeyeceğini kontrol eder.
box on komutu ile eksenler gösterilir, box off komutu ile eksenler gösterilmez.
Örnek:

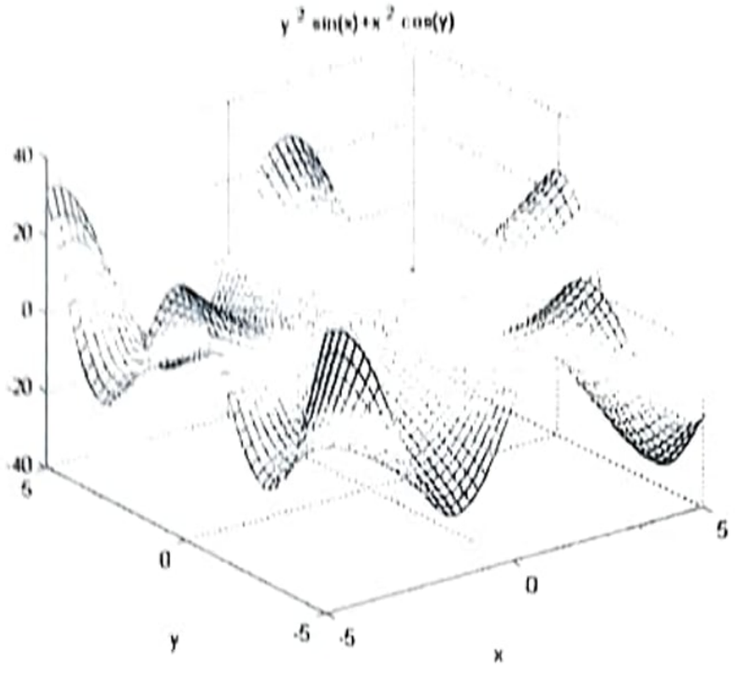
```
a=peaks(60)
mesh(a)
box on
```



→**ezmesh:** Bu komut ile verilen iki değişkenli bir fonksiyonu kolay bir şekilde çizdirmek mümkündür.

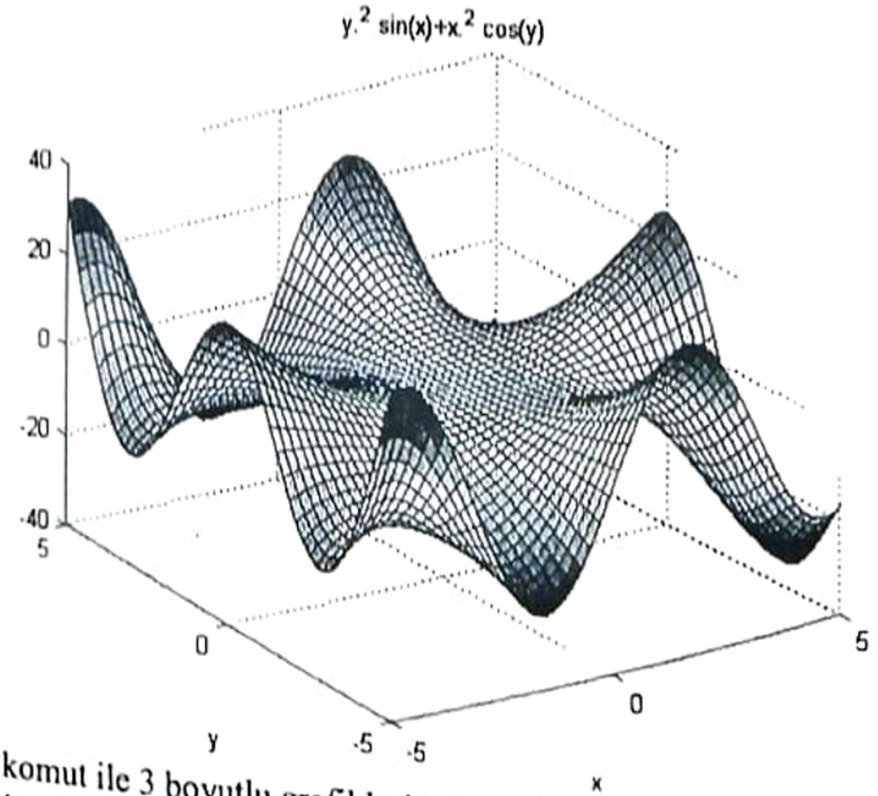
Kullanımı:
ezmesh('fonksiyon')
iki değişkenin hangi aralıklarda değiştiğini belirtmek istersek komut şu şekilde kullanılır:
ezmesh('fonksiyon',[1.başlangıç değeri 1.bitiş değeri],[2.başlangıç değeri 2.bitiş değeri])
grafikteki yüzey sayısını arttırabiliriz. :
ezmesh('fonksiyon',[1.başlangıç değeri 1.bitiş değeri],[2.başlangıç değeri 2.bitiş değeri],sayı)
bu durumda sayı x sayı miktarında bir yüzey oluşur.

Örnek:
ezmesh('y.^2*sin(x)+x.^2*cos(y)',[-5,5],[-5,5],50)



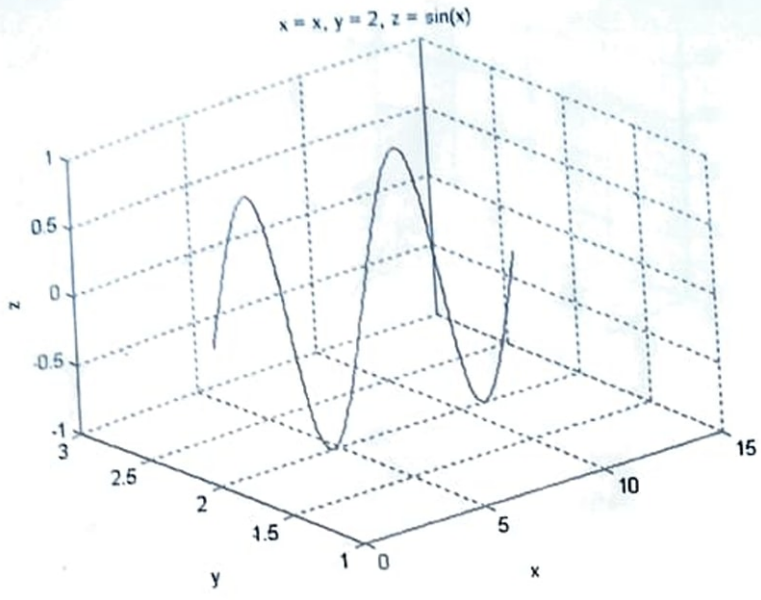
→**ezsurf:** Bu komut ezmesh komutuna benzer. O komuttan farkı yüzeylerin renkli olmasıdır.

ezsurf('y.^2*sin(x)+x.^2*cos(y)',[-5,5],[-5,5],50)



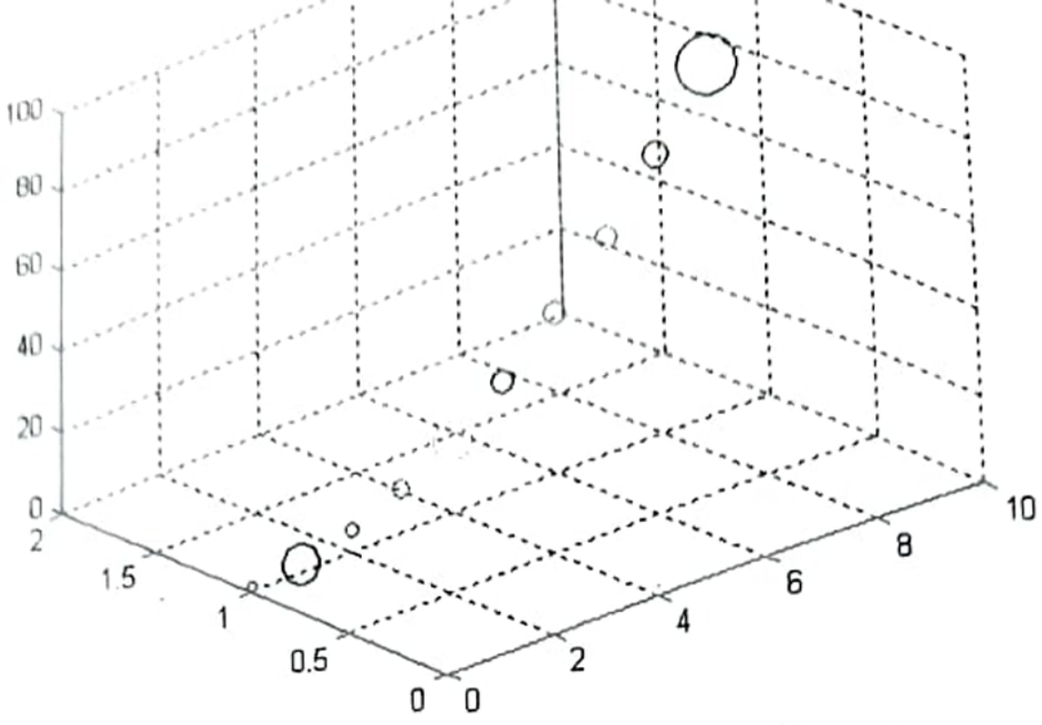
→**ezplot3 :** Bu komut ile 3 boyutlu grafikleri kolay bir şekilde çizmek mümkündür.
Kullanımı: ezplot3(x,y,z)
Çizilecek grafiğin başlangıç ve bitiş değerlerini belirtmek mümkündür:

('x','2','sin(x)',[0,4*pi]



catter3: Bu komut x,y,z koordinatları ile belirtilen noktaya çemberler çizer. Çemberlerin
utlarını ve renklerini değiştirmek mümkündür.
llanım: scatter3(x,y,z)
mberlerinin boyutlarını değiştirmek için x,y,z vektörleriyle aynı uzunlukta bir boyut
ktörü eklenmelidir.
llanımı: scatter3(x,y,z,b)
emberlerin renklerini değiştirmek için ayrıca bir renk vektörü eklenmelidir.
ullanımı: scatter3(x,y,z,b,r)

```
=0:9;
=ones(size(x))
=x.^2+y.^2;
=[10 200 30 40 200 60 70 80 90 500]
=[0 1 2 7 4 9 2 7 8 9];
scatter3(x,y,z,b,r)
```



→ rotate3d komutu bir grafiğin mouse vasıtası ile döndürülmesini sağlar.

Kullanımı:

rotate3d on

rotate3d off

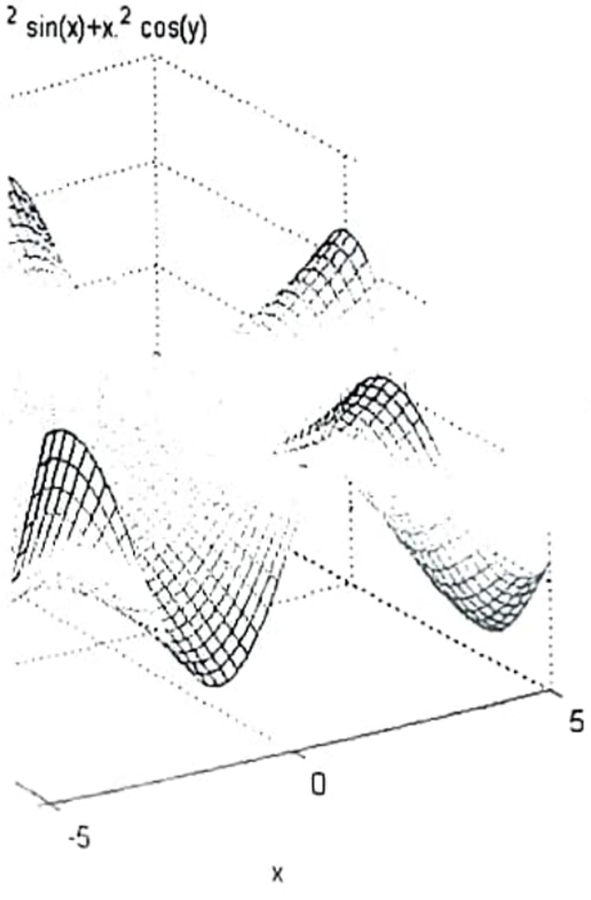
→**camdolly:** Bu komut görüntünün yerini değiştirir.

Kullanımı:camdolly(x,y,z):

Eğer x değeri değiştirilirse görüntü x ekseni boyunca yer değiştirir, y değeri d
görüntü y ekseni boyunca yer değiştirir. z değeri değiştirilirse görüntü ekseni b
değiştirme olur.

Örnek:

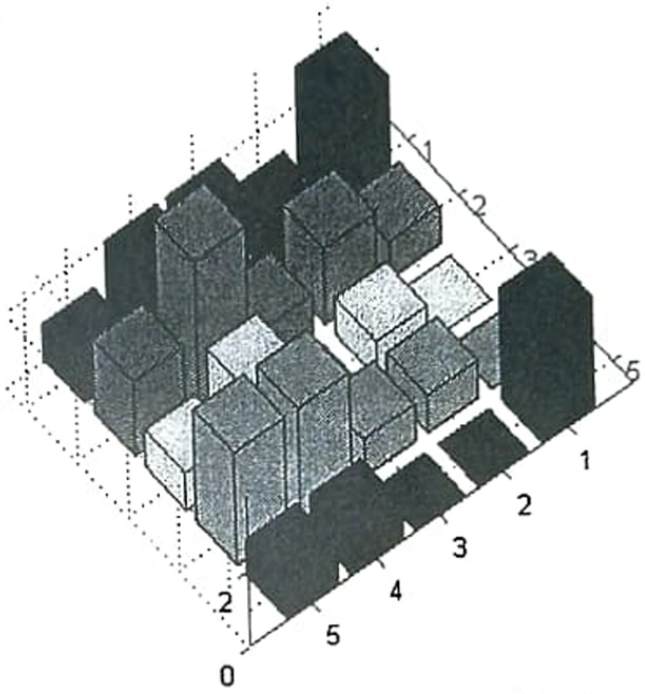
```
ezmesh('y.^2*sin(x)+x.^2*cos(y)',[-5,5],[-5,5],50)
camdolly(1,0,0)
```



→**camorbit(x,y)**: Bu komut görüntüyü yatay yönde hareket ettirir. x değeri değiştirilirse yatay yönde hareket olur. x değeri arttıkça görüntü derece cinsindendir. x değeri değiştirilirse yatay yönde sola doğru hareket eder. y değeri değiştirilirse düşey yönde hareket olur.y değeri arttıkça görüntüde düşey yönde aşağı doğru hareket olur.

örnek:

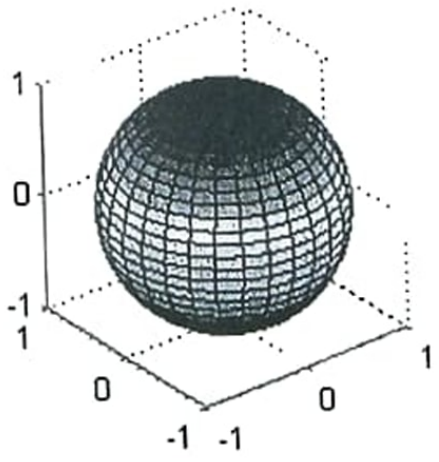
```
x=[ 3 1 0 0 3
    1 2 1 1 0
    2 1 0 1 0
    2 4 2 3 1
    1 2 1 3 1]
bar3(x)
camorbit(90,30)
```



→ **camzoom**: Bu komut görüntüye yaklaşılmasını veya uzaklaşılmasını sağlar.
Camzoom(oran): Eğer oran 1den büyük ise görüntü büyür. Eğer görüntü 0 ile 1 arasında ise görüntü küçülür.

Örnek:

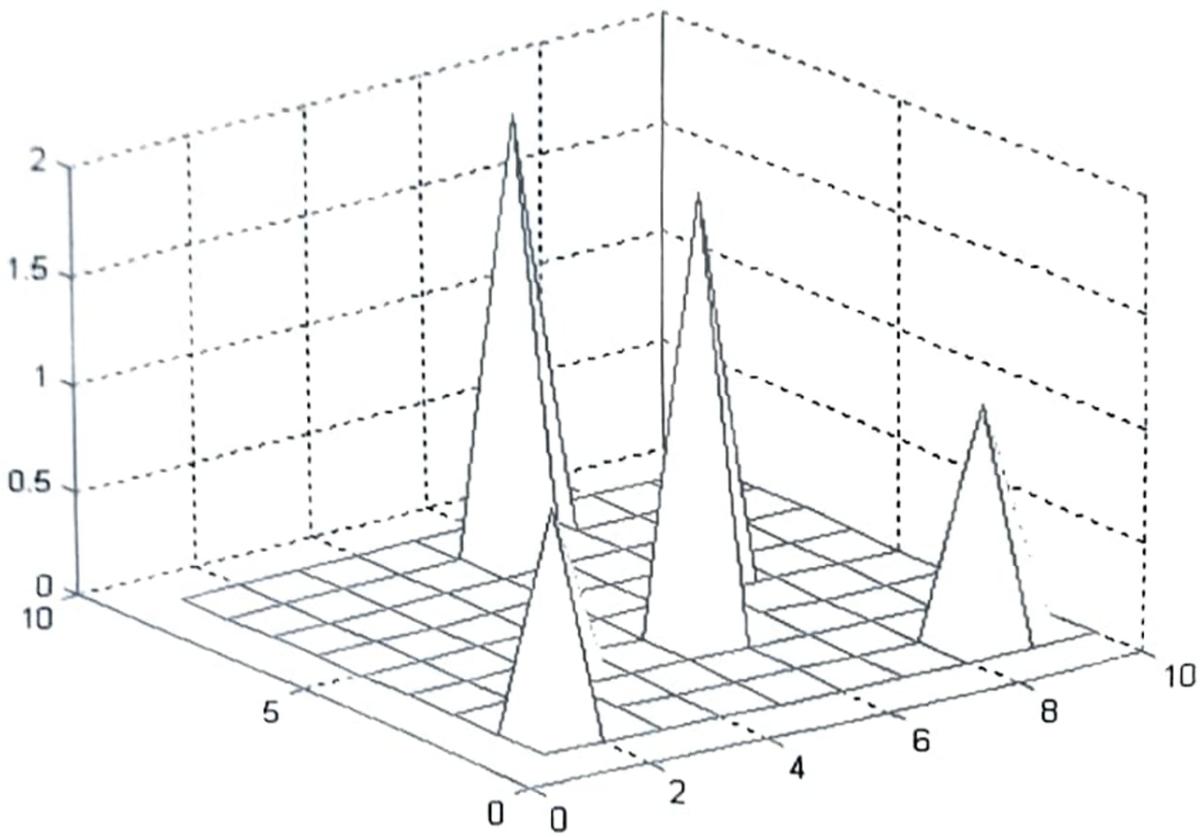
```
sphere(30)
camzoom(0.5)
```



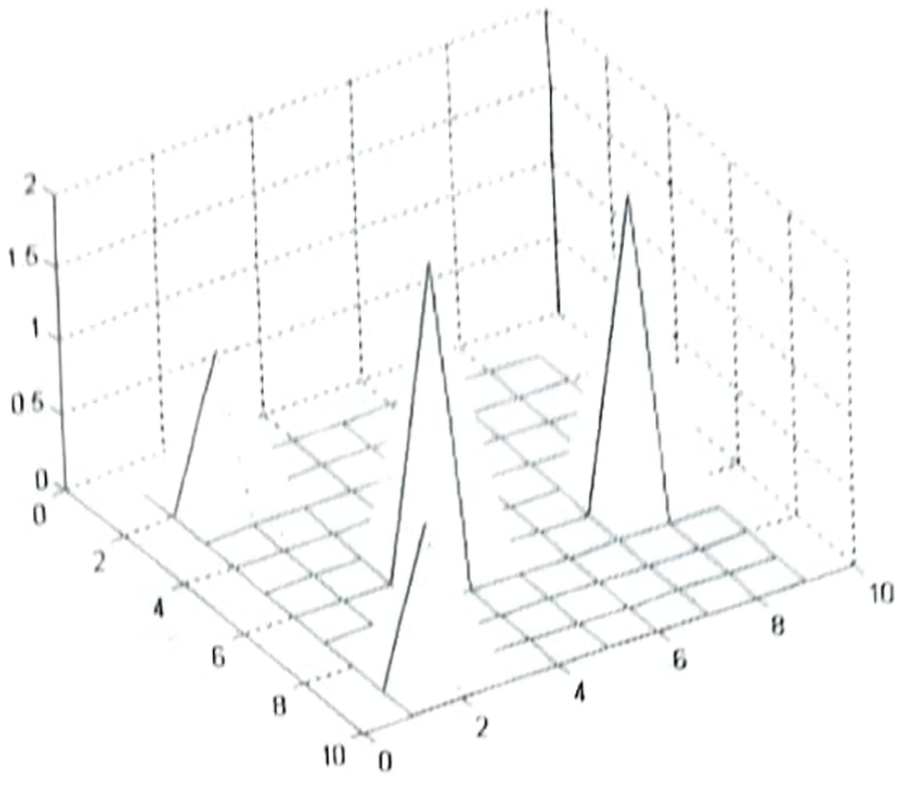
dönüş
üstüne doğru hareket, negatif
Örnek:

```
x=1:10
y=1:9
z=[0 0 0 0 0 0 0 0 0 0
   0 1 0 0 0 0 0 0 1 0
   0 0 0 0 0 0 0 0 0 0
   0 0 0 0 2 0 0 0 0
   0 0 0 0 0 0 0 0 0 0
   0 0 0 0 0 0 0 0 0 0
   0 0 0 0 0 0 0 0 0 0
   0 0 0 0 2 0 0 0 0
   0 0 0 0 0 0 0 0 0 0]
```

mesh(x,y,z)



view(60,45) komutu ile şekli kaydırırsak aşağıdaki grafik elde edilir.



Renklendirme:

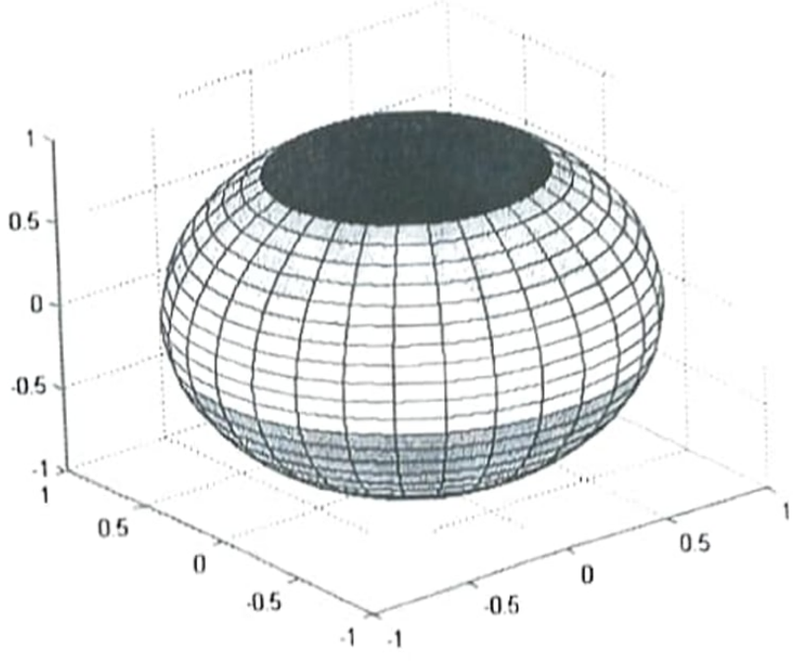
→ brighten: Bu komut grafiğin daha açık veya daha koyu gözükmesini sağlar.

Kullanımı:brighten(sayı):

Eğer sayı 0 ile 1 arasında ise resim daha açık gözükür, eğer sayı –1 ile 0 arasında ise sayı daha koyu gözükür.

Örnek:

```
sphere(30)
brighten(0.95)
```



Işıklandırma:

→Camlight: Bu komut , kameranın pozisyonuna göre grafiğe ışık verir.

Kullanımı:

camlight('headlight'): kamera pozisyonunda grafiğe ışık verir.

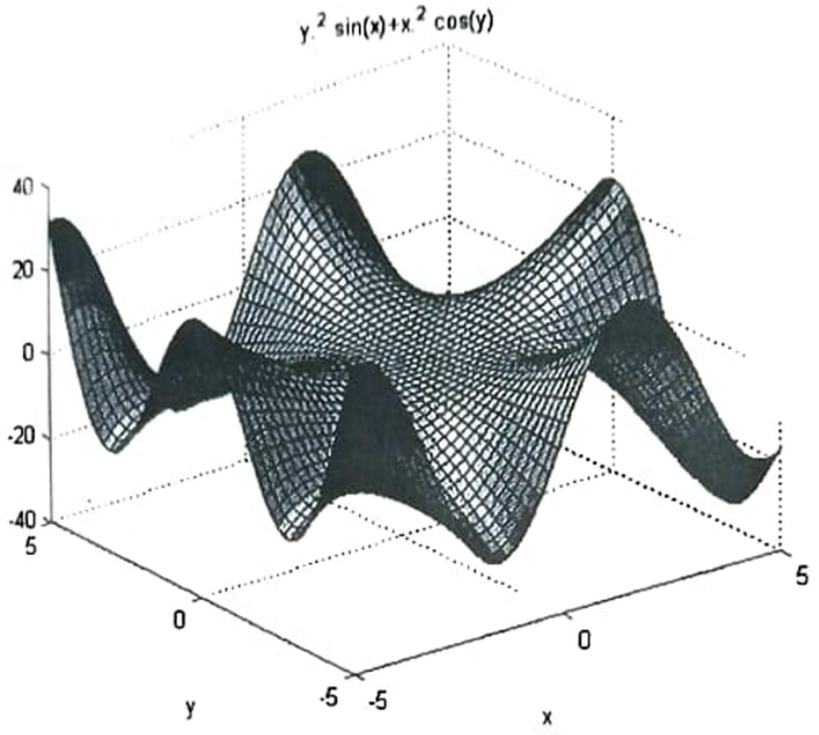
camlight('right') : Kameraya göre sağ üst taraftan ışık verir.

camlight('left') : Kameraya göre sol üst taraftan ışık verir.

camlight(a,b) : Belirtilen noktadan şekle ışık verilir. Burada a ve b, yatay ve düşey olarak ışık verilecek bölgeyi gösterir.

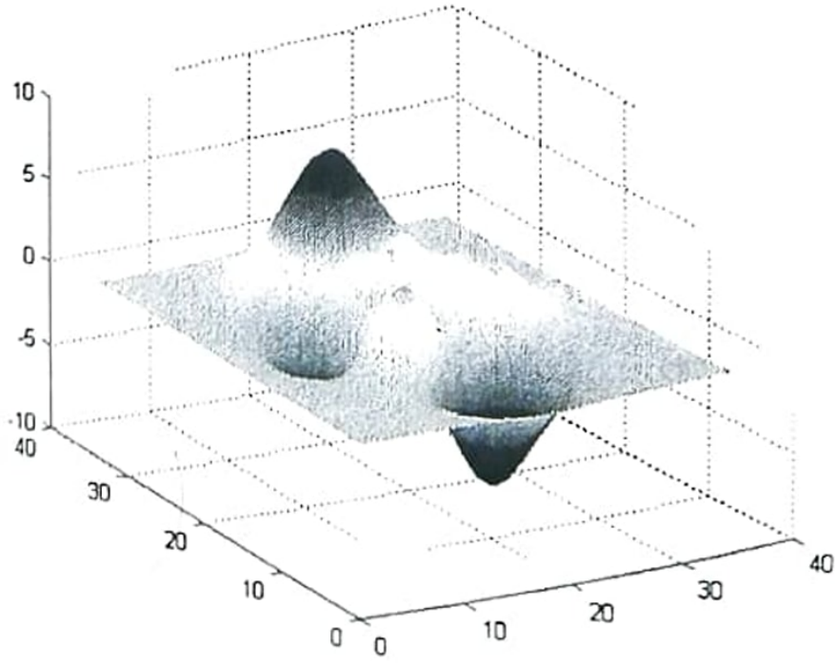
Örnek:

```
ezsurf('y.^2*sin(x)+x.^2*cos(y)',[-5,5],[-5,5],50)
camlight('left')
```



shading: Bu komut yüzeyler arası renk geçişin ayarlamak için kullanılır:
Kullanımı:
shading faceted :Bu durumda ger yüzeyin çevresinde siyah çizgiler vardır. Bu Matlab tarafından varsayılan durumdur.
shading flat : Bu durumda yüzeyler birbirinden çizgilerle ayrılmamıştır. Fakat yüzeyler kendini belli eder.
shading interp: Yüzeylerlerdeki renkler arasında oldukça yumuşak geçişler olur. Bu durumda yüzeyler ayrı ayrı gözükmez.
Örnek:

```
a=peaks(35)
surf(a)
shading interp
```



TEMEL MATEMATİK FONKSİYONLARI

→ **abs(x)**: Bu komut bir sayının mutlak değerini verir. Eğer sayı kompleks ise bu sayının büyüklüğünü verir.

Örnek 1:
```
a=-2;
b=abs(a)

b =
    2
```

Örnek 2:
```
a=-9+12i;
b=abs(a)

b =
   15
```

Örnek 3:
```
a=[-2 4 3+4i; -3 4+3i 8; -4 -5 6+8i]
b=abs(a)

a =
  -2.0000            4.0000            3.0000 + 4.0000i
  -3.0000            4.0000 + 3.0000i  8.0000
  -4.0000           -5.0000            6.0000 + 8.0000i

b =
    2    4    5
    3    5    8
    4    5   10
```

→ **acos(x), acosh(x):**

Örnek 1:
```
a=acos(sqrt(2)/2);
b=180*a/pi

b=45
```

Örnek 2:
```
a=[1 sqrt(3)/2 0.5 0 -0.5 -sqrt(3)*2 -1];
b=acos(a);
c=rad2deg(b)

a =
    1.0000    0.8660    0.5000         0   -0.5000   -0.8660   -1.0000
c =
    0   30.0000   60.0000   90.0000  120.0000  150.0000  180.0000
```

```
a=acot(
b=180*a/pi

b =
   30.0000
```

Örnek 2:

```
a=[inf  sqrt(3) 1/sqrt(3)  ];
b=acot(a);
c=rad2deg(b)

c =
        0   30.0000   60.0000
```

→ acsc(x), acsch(x):

Örnek 1:

```
a=acsc(2);
b=180*a/pi

b =
   30.0000
```

Örnek 2:

```
a=[inf 2 2/sqrt(3)];
b=acsc(a);
c=rad2deg(b)

c =
        0   30.0000   60.0000
```

→ angle(x): Bu komut verilen kompleks sayının radyan cinsinden açısını hesaplar.

Örnek 1:

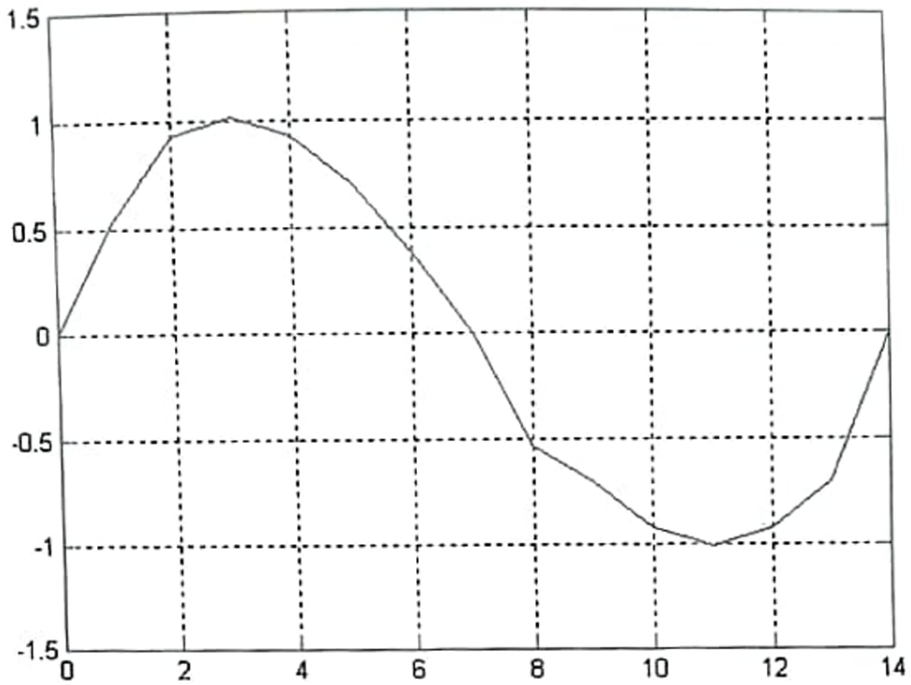
```
a=2+2i
b=angle(a)
b=b*180/pi  %Bu komut radyandan dereceye çevirir.
b =    45
```

Örnek 2:

Bu komut bir matrisin tüm elamanlarınının açısını hesaplar.

```
a=[ 2+2i sqrt(3)+i 1+ i*sqrt(3)
    3   3+3i 2i]
b=angle(a);
b=b*180/pi
45.0000   30.0000   60.0000
         0   45.0000   90.0000
```

```
a=[0 5+5i 
aci=angle(a)
b=0:14;
plot(b,aci)
grid
aci =

Columns 1 through 7
    0    0.5404    0.9273    1.0122    0.9273    0.7086    0.3805
Columns 8 through 14
    0   -0.5404   -0.7086   -0.9273   -1.0122   -0.9273   -0.7086
Column 15
    0
```



→asec(x), asech(x):

Örnek:

```
a=asec(sqrt(2))
b=a*180/pi

b=45
```

→ asin(x), asinh(x):

```
a=asin(1)
b=a*180/pi

b =
   90
```

→ **atan(x), atanh(x):**

```
a=atan(1)
b=a*180/pi

b=
   45
```

→ **ceil(x):**
Bu komut ondalık bir sayıyı kendinden büyük bir sayıya yuvarlar.

```
a=[ -2.1 2.1 -0.3; 3.5  4.2  4.6;  -4.4 5.2 1.1]
b=ceil(a)

a =
   -2.1000   2.1000  -0.3000
    3.5000   4.2000   4.6000
   -4.4000   5.2000   1.1000
b =
    -2     3     0
     4     5     5
    -4     6     2
```

→ **c = complex(a,b):** Bu komut verilen 2 sayıdan kompleks bir sayı üretir

```
c=complex(4,5)

c =
   4.0000 + 5.0000i

a=[ 1 2;3 4]
b=[2 5;6 7]
c=complex(a,b)

c=
   1.0000 + 2.0000i   2.0000 + 5.0000i
   3.0000 + 6.0000i   4.0000 + 7.0000i
```

→ **b= conj(a):** Bu komut verilen kompleks sayının eşleniğini bulur.

Örnek 1:

```
a=4+6i
b=conj(a)

b =
   4.0000 - 6.0000i
```

Örnek 2:

```
a =
   2.0000 + 2.0000i   4.0000 - 3.0000i   8.0000 + 4.0000i
   7.0000 - 2.0000i   1.0000 + 3.0000i   8.0000 - 4.0000i
   4.0000 + 3.0000i   5.0000 - 2.0000i   7.0000 + 6.0000i
```

7.0000 + 2.0000i
4.0000 - 3.0000i 5.0000 + 2.0000i 7.0000 - 6.0000i

→ cos(x), cosh(x):

Örnek:

```
a=[0 60 90 120 180]
a=a*pi/180
b=cos(a)
```

```
b =
   1.0000    0.5000    0.0000   -0.5000   -1.0000
```

→ cot(x), coth(x)

Örnek:

```
a=[45 90 135]
a=a*pi/180
b=cot(a)
```

```
b =
   1.0000    0.0000   -1.0000
```

→ csc(x), csch(x):

Örnek:

```
a =[30 90 150]
a=a*pi/180
b=csc(a)
```

```
b =
   2.0000    1.0000    2.0000
```

→ exp(x): verilen sayının e. kuvvetini verir.

Örnek 1:

```
a=[0 1 2]
b=exp(a)
```

```
b =
   1.0000    2.7183    7.3891
```

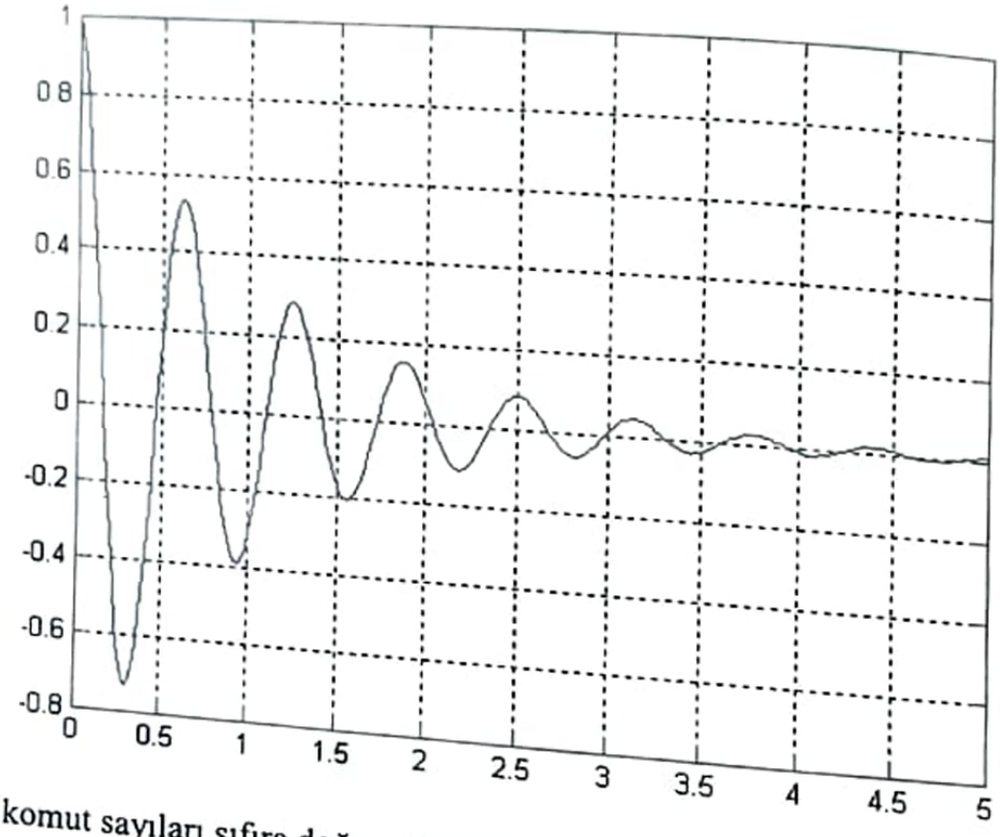
eğer sayı kompleks ise şu şekilde hesaplanma yapılır
c=ai olmak üzere:
exp(c)=cos(a)+isin(b) olur.
Örnek 2:

```
a=i*(pi/6)
c=exp(a)
```

```
c =
   0.8660 + 0.5000i
```

Örnek 3:

```
x=0:.02:5;
y=exp(-x).*cos(10*x);
plot(x,y)
grid
```



→ **fix(x)**: Bu komut sayıları sıfıra doğru yuvarlar.

Örnek:

```
a=[-2.1 -1.5 -0.8 0.8 2.2 2.9 4.6]
b=fix(a)

a =
   -2.1000   -1.5000   -0.8000    0.8000    2.2000    2.9000    4.6000
b =
   -2    -1     0     0     2     2     4
```

→ **floor(x)**:Bu komut ondalık bir sayıyı kendinden küçük bir sayıya yuvarlar.

```
a=[-2.1 -1.5 -0.8 0.8 2.2 2.9 4.6]
b=floor(a)

a=
   -2.1000   -1.5000   -0.8000    0.8000    2.2000    2.9000    4.6000
b =
   -3    -2    -1     0     2     2     4
```

→ **gcd(x,y)**: Bu komut en büyük ortak böleni verir.

Örnek 1:

```
>> gcd(100,30)
ans =
   10
```

Örnek 2:

```
a=[10 20 30;28 80 54;45 48 56]
b=[4 8 20;21 60 18;18 36 42]
gcd(a,b)

a =
   10   20   30
   28   80   54
   45   48   56
b =
    4    8   20
   21   60   18
   18   36   42
ans =
    2    4   10
    7   20   18
    9   12   14
```

→ **imag(c):** Bu komut kompleks sayının sanal kısmını bulur.

Örnek 1:

```
>> imag(4+9i)

ans =
    9
```

Örnek 2:

```
a=[4+2i 5-3i 6+4i;2+i 5+4i  4+2i;6+5i 3+2i 6+4i]
b=imag(a)

a =
   4.0000 + 2.0000i   5.0000 - 3.0000i   6.0000 + 4.0000i
   2.0000 + 1.0000i   5.0000 + 4.0000i   4.0000 + 2.0000i
   6.0000 + 5.0000i   3.0000 + 2.0000i   6.0000 + 4.0000i
b =
    2   -3    4
    1    4    2
    5    2    4
```

→ **lcm(a,b):** Bu komut en küçük ortak katı verir.

Örnek 1:

```
>> lcm(24,30)

ans =
   120
```

Örnek 2:

```
a=[2 6 4; 1 8 4; 6 4 10]
b=[5 4 2; 3 6 3;5 8 8]
lcm(a,b)
```

```
a =
     2     6     4
     1     8     4
     6     4    10
b =
     5     4     2
     3     6     3
     5     8     8
ans =
    10    12     4
     3    24    12
    30     8    40
```

→ **log(a):** Bu komut e tabanına göre logaritmayı hesaplar:

```
a=1:5
b=exp(a)
c=log(b)
```

```
a =
     1     2     3     4     5
b =
   2.7183   7.3891  20.0855  54.5982 148.4132
c =
     1     2     3     4     5
```

→ **log2(x):** 2 tabanına göre logaritma hesaplar.

Örnek:

```
a=[1 2 4 8 16 32 64]
b=log2(a)
```

```
a =

     1     2     4     8    16    32    64
b =
     0     1     2     3     4     5     6
```

→ **log10(x):** 10 tabanına göre logaritma hesaplar.

```
 a=[10 10^(3/2) 100 10^(5/2) 1000]
b=log10(a)
```

```
a =
  1.0e+003 *
    0.0100    0.0316    0.1000    0.3162    1.0000
```

```
b =
   1.0000   1.5000   2.0000   2.5000   3.0000
```

→ **mod(x,y):** x sayısının mod y'ye göre karşılığını verir.

Örnek 1:

```
>> mod (14,5)
ans =
    4
```

Örnek 2:

```
a=[12 20 34;23 65 40;28 65 37]
b=[5 7 8;5 12 30;13 30 14]
rem(a,b)

a =
    12    20    34
    23    65    40
    28    65    37
b =
     5     7     8
     5    12    30
    13    30    14
ans =

     2     6     2
     3     5    10
     2     5     9
```

→ **real(c):** Bir kompleks sayının reel kısmını verir.

Örnek 1:

```
>> real(6+5i)
ans =
    6
```

Örnek 2:

```
a=[2+3i 3+2i 4-4i;1+i 3+5i 4-6i;4+2i 8+3i 9+5i]
b=real(a)

a =
   2.0000 + 3.0000i   3.0000 + 2.0000i   4.0000 - 4.0000i
   1.0000 + 1.0000i   3.0000 + 5.0000i   4.0000 - 6.0000i
   4.0000 + 2.0000i   8.0000 + 3.0000i   9.0000 + 5.0000i
b =
     2     3     4
     1     3     4
     4     8     9
```

→**rem(a,b):** a sayısının b'ye bölümünden kalanını verir.

```
ans =
    3
```

→ **round(x)**: verilen ondalıklı sayıyı en yakın tamsayıya tamamlar.

```
a=[-4.3 -3.6 -0.5 -0.2 0.4 0.5 3.6 4.3]
b=round(a)

a =
   -4.3000   -3.6000   -0.5000   -0.2000    0.4000    0.5000    3.6000    4.3000
b =
    -4    -4    -1     0     0     1     4     4
```

→ **sec(x), sech(x):**

```
a=[pi/6 pi/3 pi/4]
b=sec(a)

a =
    0.5236    1.0472    0.7854
b =
    1.1547    2.0000    1.4142
```

→ **sign(x):** Bu komut negatif sayılar için –1, pozitif sayılar için +1, 0 için 0 üretir. komutu

```
a=-5:5
b=sign(a)

a =
    -5    -4    -3    -2    -1     0     1     2     3     4     5
b =
    -1    -1    -1    -1    -1     0     1     1     1     1     1
```

→ **sin(a), sinh(a):**

```
a=0:pi/6:pi
b=sin(a)

a =
         0    0.5236    1.0472    1.5708    2.0944    2.6180    3.1416
b =
         0    0.5000    0.8660    1.0000    0.8660    0.5000    0.0000
```

→ **sqrt(a)**: verilen sayının karekökünü hesaplar.

```
a=-4:2:4
```

```
b=sqrt(a)

a =
    -4    -2     0     2     4
b =
  0 + 2.0000i  0 + 1.4142i  0  1.4142  2.0000
```

→tan(x), tanh(x):

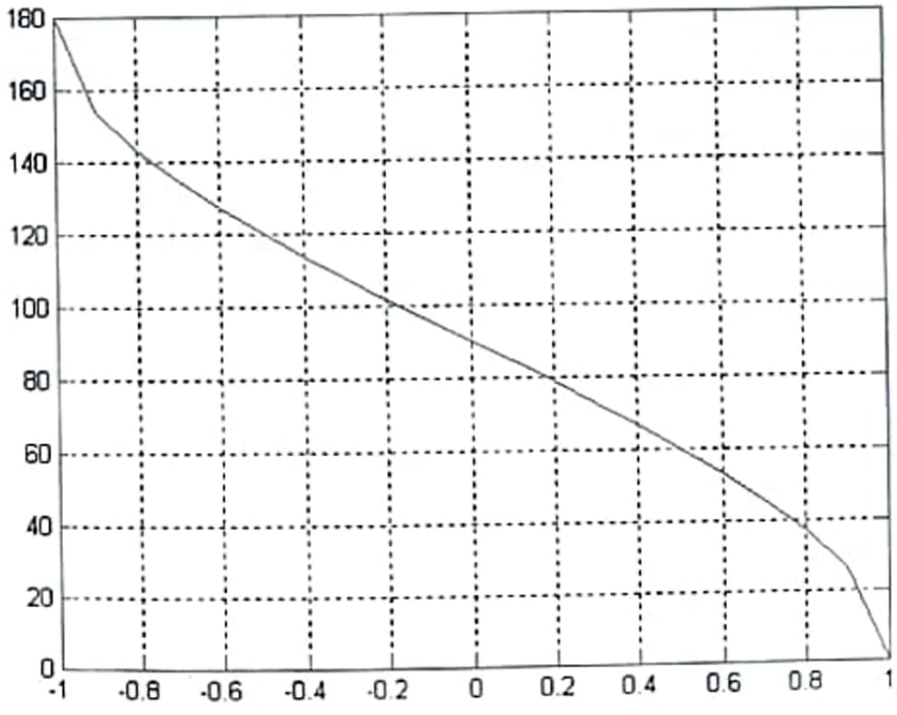
```
a=[0 30 45 60 ]
b=a*pi/180
c=tan(b)

c =

    0    0.5774    1.0000    1.7321
```

Alıştırmalar

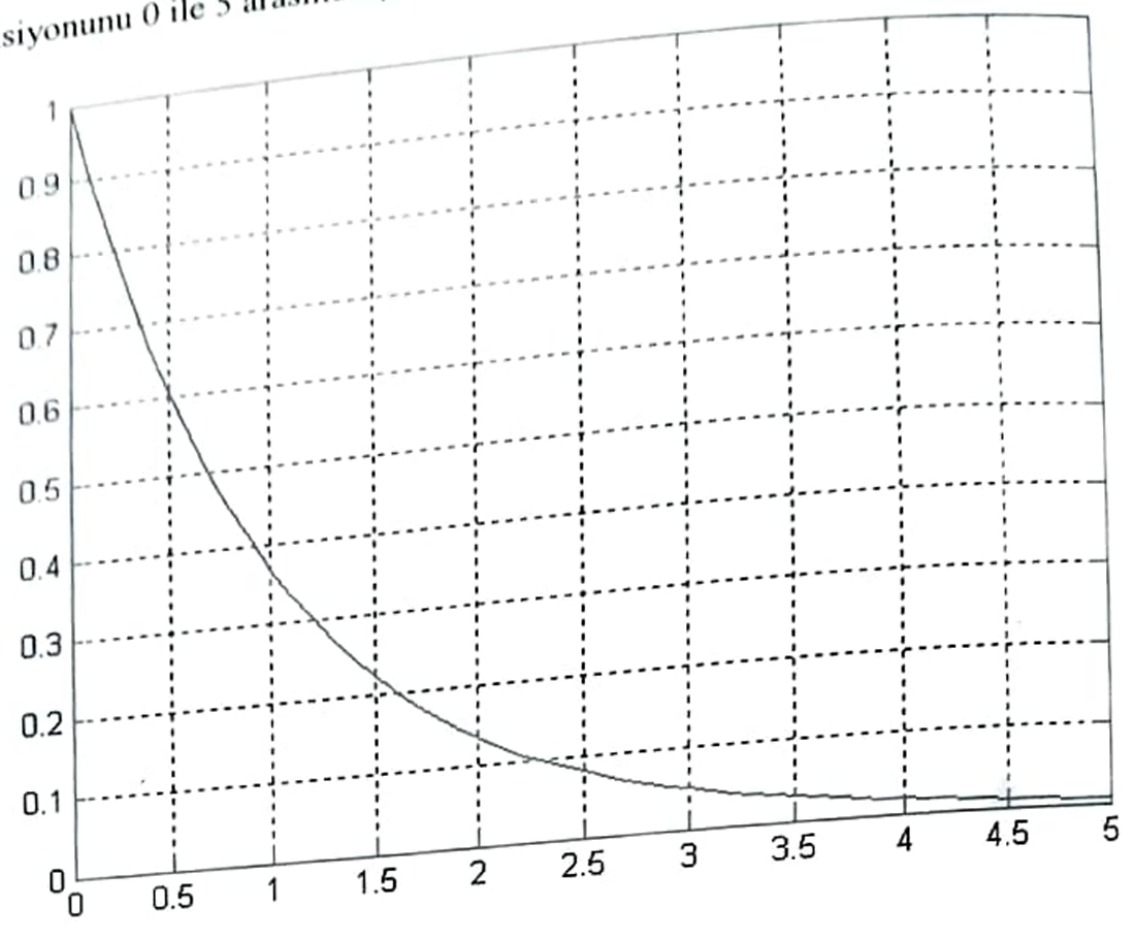
1) acos(x) fonksiyonunu –1 ile +1 arasında çizdiriniz.

cevap:

```
a=-1:.1:1;
b=acos(a)
b=b*180/pi
plot(a,b)
grid
```



2) e^{-x} fonksiyonunu 0 ile 5 arasında çizdiriniz.



3) Aşağıdaki a dizisini fix ve floor komutu ile yuvarlayınız.

a=[3.4 4.3 5.6 7.8 –2.2 –3.4 –3.9 –0.4 2.4 3.5 5.6]
cevap:

```
clc
a=[3.2 4.7 5.6 7.8 -2.2 -3.4 -3.9 -0.4 2.4 3.5 5.6]
b=fix(a)
c=floor(a)

b =
    3    4    5    7   -2   -3   -3    0    2    3    5
c =
    3    4    5    7   -3   -4   -4   -1    2    3    5
```

4) 100 sayısının 6'ya bölümünden kalanını bulunuz.
Cevap:

```
rem(100,6)

ans =
    4
```

5) Aşağıdaki dizinin mutlak değerini abs komutu kullanmadan alınız.

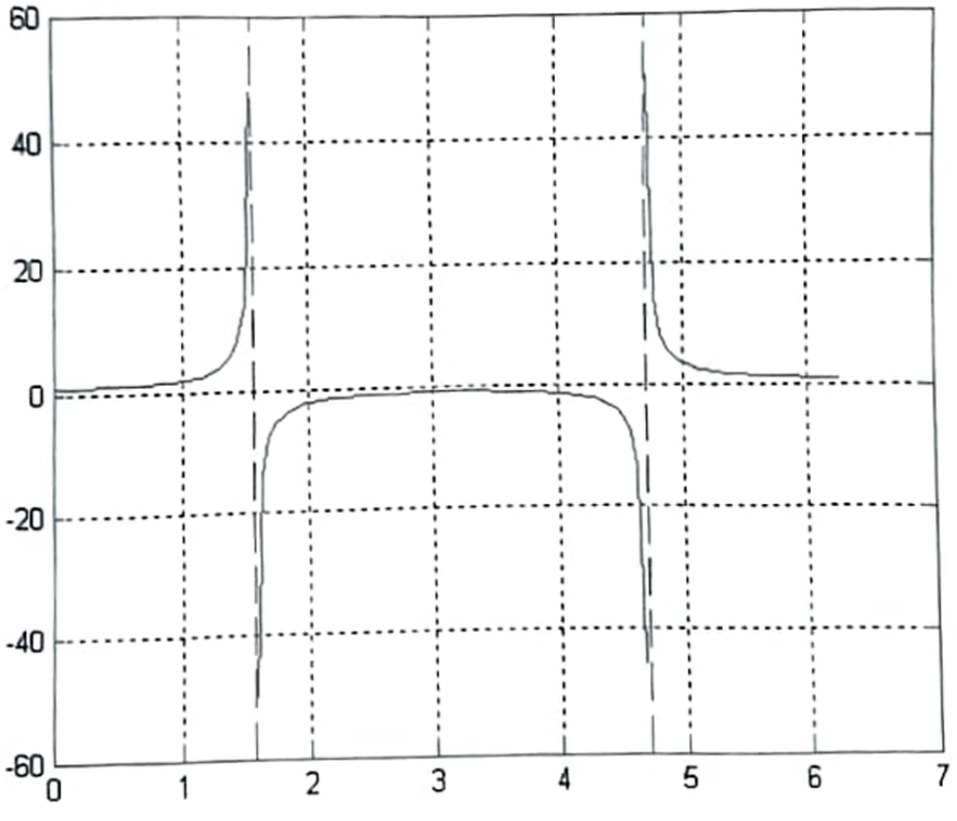
```
a=[-2 3 5 8 -9 0 -4 6 -4 1 2];
b=sign(a);
sonuc=a.*b
```

```
sonuc =
    2    3    5    8    9    0    4    6    4    1    2
```

6) sec(x) fonksiyonunu 0 ile 2π arasında çizdiriniz.

cevap:

```
a=0:.05:pi/2;
b=pi/2+0.02:.05:3*pi/2;
c=3*pi/2+0.02:0.05:2*pi;
y1=sec(a);
y2=sec(b);
y3=sec(c);
y=[-60 60];
t1=[pi/2 pi/2];
t2=[3*pi/2 3*pi/2]
plot(a,y1,b,y2,'b',c,y3,'b',t1,y,'r--',t2,y,'r--')
grid
```



7) Aşağıda verilen kompleks sayıların açılarını derece cinsinden bulunuz.

$\sqrt{3}$ +i, 4+4i, -1+$\sqrt{3}$ i

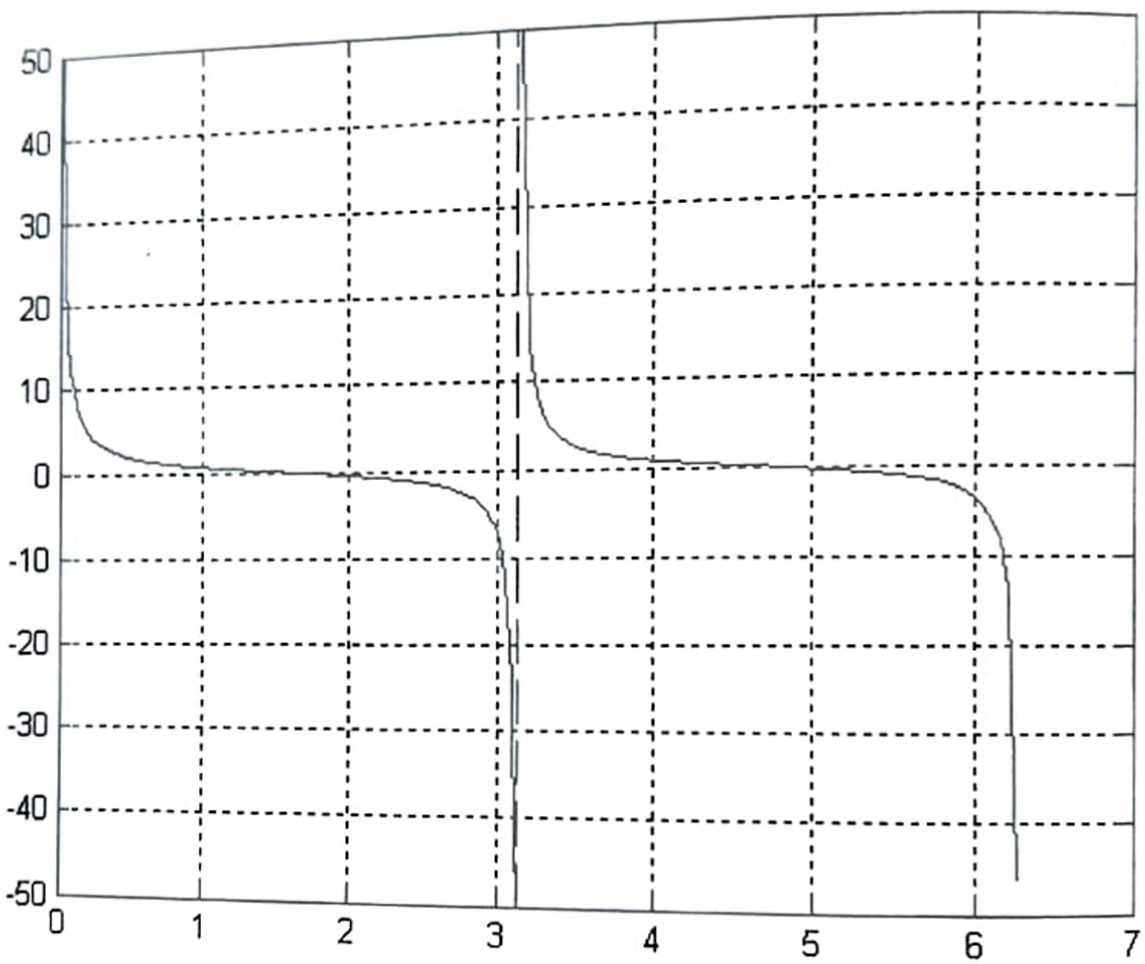
Cevap:

```
a=[sqrt(3)+i 4+4i -1+sqrt(3)*i];
radyan=angle(a);
derece=180*radyan/pi
```

30.000

8) cot(x) fonksiyonunu 0 ile 2π arasında çizdiriniz.

```
a=0.02:.02:pi-0.02;
b=pi+0.02:.05:2*pi;
y1=cot(a);
y2=cot(b);
y=[-50 50];
t=[pi pi];
plot(a,y1,b,y2,'b',t,y,'r--')
grid
```



9) 18, 30, 15 sayılarının en küçük ortak katlarını bulunuz.

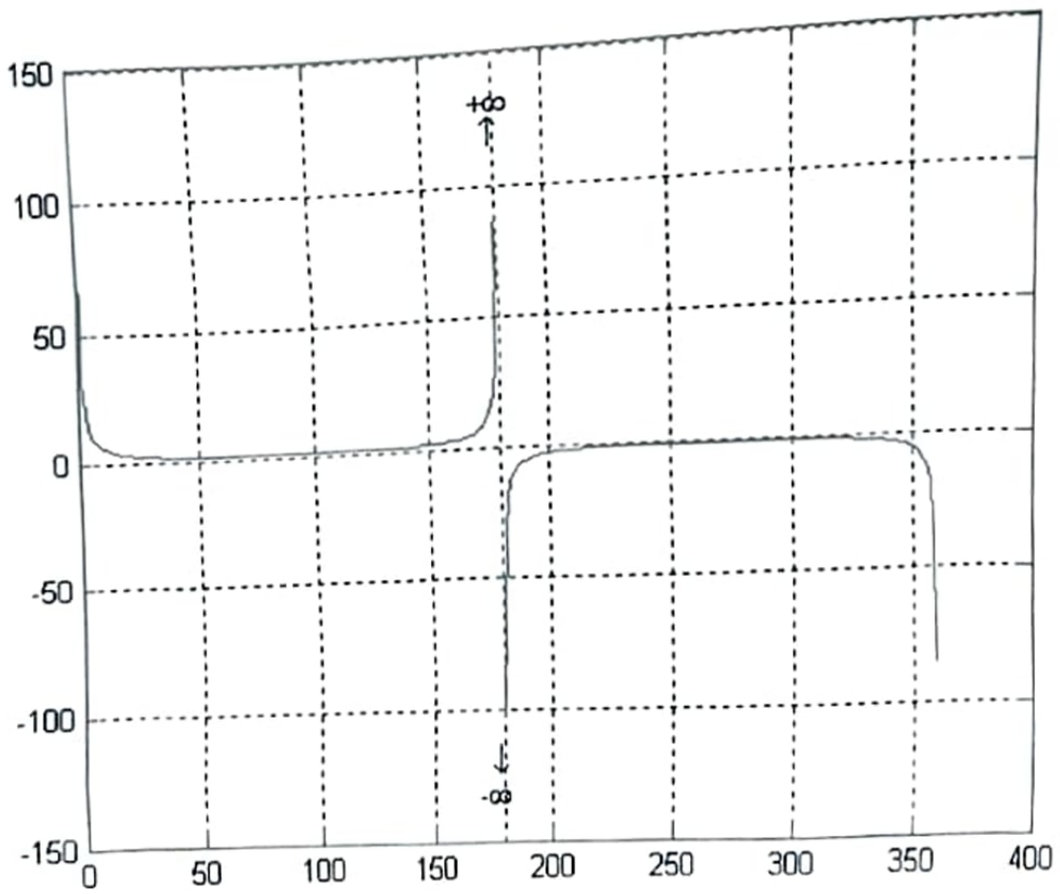
```
a=lcm(18,30);
sonuc=lcm(a,45)

sonuc =

          90
```

10) 14, 28, 49 sayılarının en büyük ortak bölenini bulunuz.

```
a=gcd(14,28);
sonuc=gcd(a,49)

sonuc =

           7
```

11) csc(x) fonksiyonunu 0 ile 2π arasında çizdiriniz.

```
m=0.01;
a=0+m:.02:pi-m;
b=pi+m:.02:2*pi-m;
y1=csc(a);
y2=csc(b);
a=a*180/pi;
b=b*180/pi;
c=[180 180];
y3=[-150 150];
plot(a,y1,b,y2,'b',c,y3,'r:')
text(175,120,'\uparrow')
text(170,132,'+\infty')
text(175,-120,'\downarrow')
text(170,-132,'-\infty')
grid
```



Problemler

1) Aşağıda verilen matrisin elemanlarının mutlak değerlerini bulunuz.

$$\begin{bmatrix} 1+2i & 2+4i & 5 \\ 6i & 3+4i & 5-3i \\ 4+8i & 8-8i & 12 \end{bmatrix}$$

2)acot(x) fonksiyonunu 0 ile 2π arasında çizdiriniz.

3) 1. soruda verilen matrisin elemanlarının açılarını derece cinsinden bulunuz.

4) acsc(x) fonksiyonunu 0 ile 2π arasında çizdiriniz

5) Aşağıda verilen dizinin elemanlarını yukarı doğru sıralayınız.
A=[0.3 3.4 1.2 4.7 –0.5 0.2 –4.3 5.2 6.2 3.5]

6) asec(x) fonksiyonunu 0 ile 2π arasında çizdiriniz

7) 1. soruda verilen matrisin elemanlarının eşleniğini bulunuz.

8) asin(x) fonksiyonunu 0 ile 2π arasında çizdiriniz

9) 5. soruda verilen dizinin elemanlarını sıfıra doğru yuvarlayınız.

10) atan(x) fonksiyonunu 0 ile 2π arasında çizdiriniz

11) 5. soruda verilen dizinin elemanlarını aşağıya doğru yuvarlayınız.

12) 24, 60, 84 sayılarının en büyük ortak bölenlerini bulunuz.

13) 1. soruda verilen matrisin elemanlarının sanal kısmını bulunuz.

14) 18, 27, 45 sayılarının en küçük ortak katını bulunuz.

15) log10(x) fonksiyonunu 0.1 ile 100 arasında çizdiriniz.

16) 1. soruda verilen matris elemanlarının gerçek kısmını bulunuz.

17) 5. soruda verilen dizinin elemanlarını en yakın sayıya doğru yuvarlayınız.

18) 0 ile 10 arasında 0.01 aralıklarla sin(x) fonksiyonunu hesaplatınız. Bulduğunuz değerleri sign(x) fonksiyonuna yerleştirerek çizdiriniz.

FONKSİYON YAZMA

Matlab'de fonksiyon yazmak mümkündür. Bu fonksiyon yazıldıktan sonra isim.m seklinde kaydedilmelidir. Bundan sonra bu fonksiyonu istediğimiz zaman geri çağırmamız mümkün olacaktır. Böylece programın her defasında tekrarlanmasına gerek kalmayacaktır.

Örnekler:

1)

```
% Bu fonksiyon bir sayının faktöryelini hesaplar.
function a=faktoryel(b)
a=1;
for i=1:b
   a=a*i;
end
```

Bu fonksiyon yazılıp kaydedildiği için istenildiği zaman kullanılabilir.
Örnek:

```
>> faktoryel(5)

ans =
  120
```

2)

```
% Bu fonksiyon bir sayının karesini hesaplar.
function a=kare(b)
a=b*b;
```

örnek:

```
>> kare(3)

ans =
   9
```

3)

```
% Bu fonksiyon verilen bir diziyi büyükten küçüğe doğru sıralar.

function a=sirala(b)
for i=1:length(b)-1;
   for j=i+1:length(b);
      if b(j)>b(i)
         sayi=b(i);
         b(i)=b(j);
         b(j)=sayi;
      end
   end
end
a=b;
```

örnek:
```
>> q=[ 6 1 3 8 5 9 14]

q =
   6   1   3   8   5   9   14

>> sirala(q)

ans =
   14   9   8   6   5   3   1
```

4)
```
% Bu fonksiyon radyan cinsinden verilen sayıyı dereceye çevirir.

function a=rade(b)
a=b*180/pi;
```

örnek:
```
>> rade(pi/2)
ans =
   90
```

5)
```
% Bu fonksiyon verilen bir dizinin ortalamasını hesaplar.
function b=ort(c)
toplam=0;
for i=1:length(c)
   toplam=toplam+c(i);
end
b=toplam/length(c);
```

örnek:
```
a=[1 10 89 11 20 30 12 19]
ort(a)
ans =
   24
```

6)
```
% Bu fonksiyon verilen bir sayının tek mi, çift mi olduğunu söyler.
function tekcift(c)
if rem(c,2) ==0
   disp('Bu sayi çift sayidir')
else
   disp('Bu sayi tek sayidir')
end
örnek:
>> tekcift(10)
Bu sayi çift sayidir
>> tekcift(5)
Bu sayi tek sayidir
```

7)
% Bu fonksiyon iki kenarı verilen dik üçgenin hipotenüsünü hesaplar.

```
function a=hip(b,c)
a=sqrt(b^2+c^2);
```

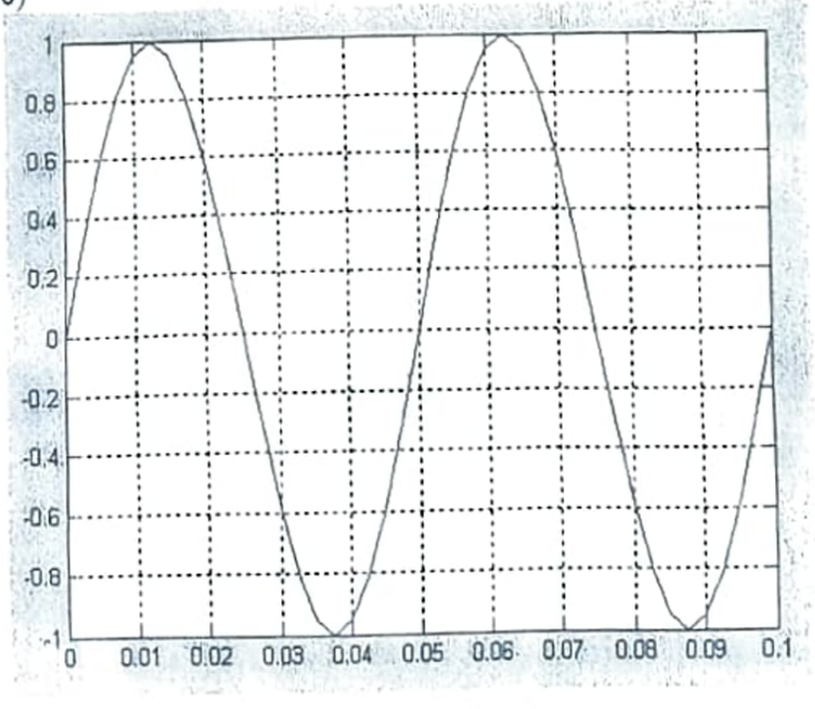
örnek:
```
>> hip(5,12)
ans =
   13
```

8)
```
% Bu fonksiyon istenilen yarıçapta bir daire çizer.
function daire(a)
i=0:.1:2*pi
plot(a*cos(i),a*sin(i))
```

Alıştırmalar

1) Verilen frekansa göre 2 periyotluk sinüs eğrisi çizen bir fonksiyon yazınız.
```
function sinus(fre)
t=0:1/(20*fre):2/fre
y=sin(2*pi*fre*t)
plot(t,y)
grid
```

uygulama: sin(20)



2) Bir matrisin en büyük elemanını bulan bir fonksiyon yazınız.

```
function a=buyuk(b)
[r,k]=size(b);
```

```
c=r*k;
for i=1:c-1;
   for j=i+1:c;
      if b(j)>b(i)
         sayi=b(i);
         b(i)=b(j);
         b(j)=sayi;
      end
   end
end
a=b(1);

uygulama:
a=[1 4 2
   9 8 2
   1 2 3]
buyuk(a)

a =
    1    4    2
    9    8    2
    1    2    3
ans =
    9
```

3) Bir dizideki elemanların çarpımını bulunuz.

```
function a=carp(b)
carp=1;
for i=1:length(b)
   carp=carp*b(i);
end
a=carp;

uygulama:
a=[1 3 4 2 5]
carp(a)

a =
    1    3    4    2    5
ans =
  120
```

Problemler

1) Bir matrisin en küçük elemanını bulan bir fonksiyon yazınız.

2) Bir dizideki elemanların toplamını bulan bir fonksiyon yazınız.

3) Bir dizideki elemanları küçükten büyüğü doğru sıralayan bir fonksiyon yazınız.

4) Derece olarak verilen bir sayıyı radyana çeviren fonksiyon yazınız.

Sıfır Geçiş Noktasını Bulmak:

Bir fonksiyondaki sıfır geçiş noktası "fzero" fonksiyonu ile bulunur. Sıfır geçiş noktası tahmini bir başlangıç noktasının etrafında veya iki değer arasında araştırılır.

Örnekler:

1)

```
gecis=fzero('5*cos(x)',1)

gecis =  1.5708
```

Geçiş noktasına karşılık gelen fonksiyon değeri şu şekilde bulunabilir.

```
[gecis deger]=fzero('5*cos(x)',1)

gecis =  1.5708        deger =  3.0616e-016
```

2)

```
gecis=fzero('x.^2-x-6',[-3 -1])

gecis =  -2
```

3)

```
gecis=fzero('x.^3+3*x.^2-x-3',-4)

gecis=-3
```

Bir fonksiyonun minimum değerini bulmak.

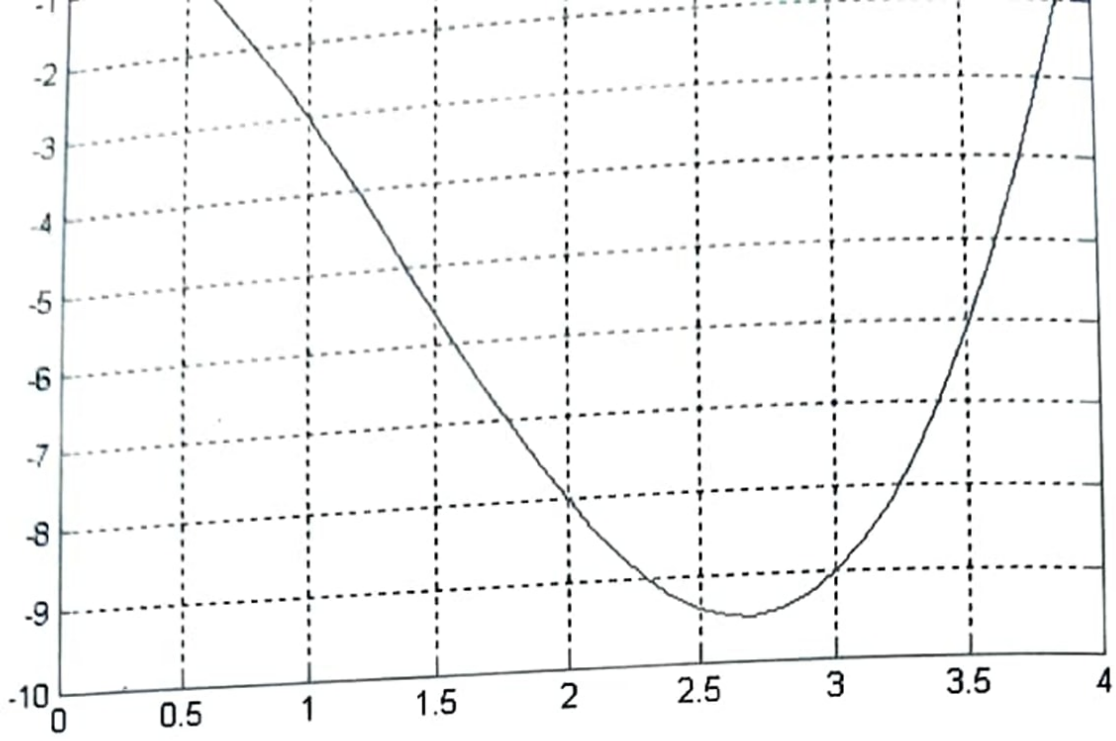
Bir fonksiyonu minimum yapan x değeri fminbnd komutu ile bulunur. Fonksiyonu minumum yapan değer iki değer arasında araştırılır.

Örnek:

$y=x^3-4x^2$ fonksiyonunu minimum yapan x değerini bulunuz.

Fonksiyon aşağıdaki komutlarla çizdirilirse şu grafik elde edilir.

```
x=0:.1:4;
y=x.^3-4*x.^2;
plot(x,y)
grid
```



Görüldüğü gibi fonksiyonun minimum noktası 2 ile 3 arasında değişiyor. Bu durumda aşağıdaki komut ile fonksiyonu minumum yapan x değeri bulunabilir.

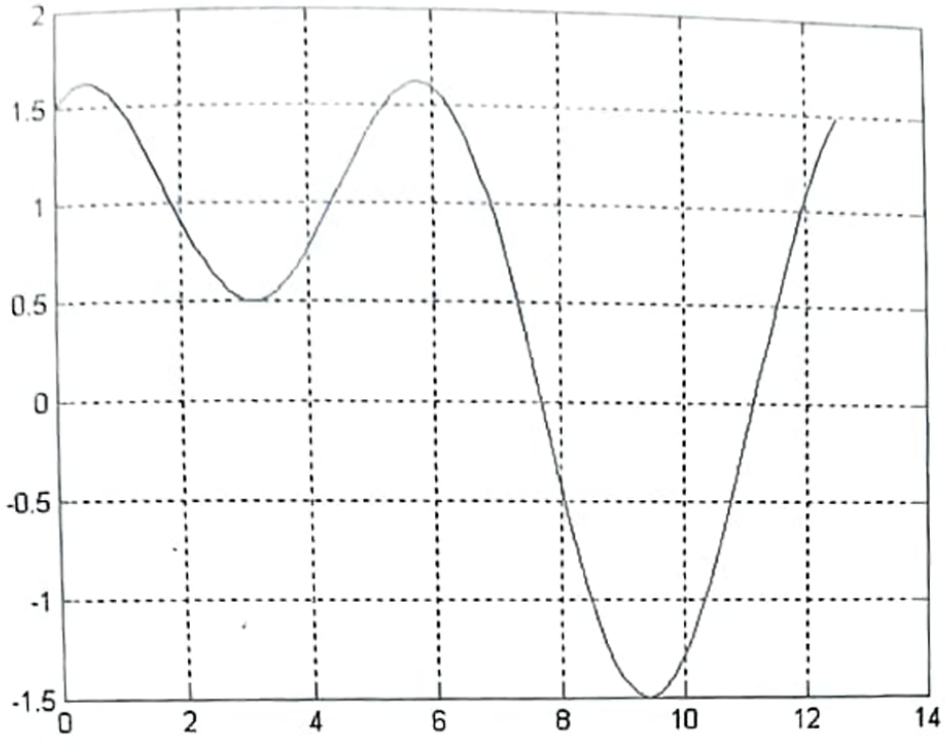
```
z = fminbnd('x.^3-4*x.^2',1,3)

z =
   2.6667
```

Alıştırmalar

1) y=cos(x)+sin(x/2)+0.5 fonksiyonunu 0 ile 4π arasında çizdiriniz ve y'nin sıfırdan geçiş değerlerini bulunuz.

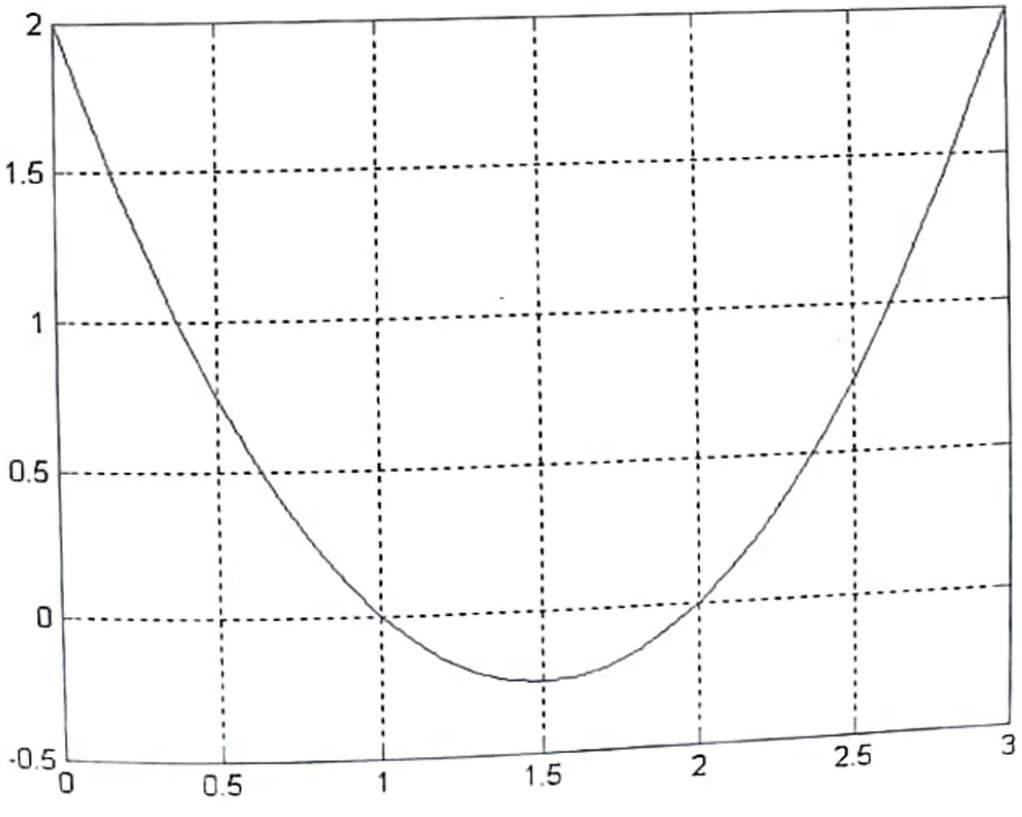
Çözüm:

```
t=0:pi/32:4*pi;
y=cos(t)+sin(t/2)+0.5;
plot(t,y)
grid
gecis1=fzero('cos(x)+sin(x/2)+0.5',8)
gecis2=fzero('cos(x)+sin(x/2)+0.5',12)

gecis1 =
   7.7020


gecis2 =
   11.1475
```



2) x2-3x+2 fonksiyonunun sıfırdan geçiş değerlerini bulunuz.

```
t=0:.1:3;
y=t.^2-3*t+2;
plot(t,y)
grid
gecis1=fzero('x^2-3*x+2',0)
gecis2=fzero('x^2-3*x+2',3)
```



1) $y=x^3-3x^2+2x$ fonksiyonunun x=0 ile 2.5 arasında değişirken y'nin sıfırdan geçiş değerini bulunuz.

2) y=tan(x) fonksiyonunu 100 derece ile 260 derece arasında çizdiriniz ve y'nin sıfırdan geçiş değerini bulunuz.

3) $y=x^2-3x+1$ fonksiyonunu 0 ile 4 arasında çizdiriniz ve y'nin sıfırdan geçiş değerini bulunuz.

4) y=sin(x)+cos(2x)+sin(x/2) fonksiyonunu 0 ile 2π arasında çizdiriniz ve y'nin sıfırdan geçiş değerini bulunuz.

5) y=sin(x)-cos(x) fonksiyonunu 0 ile 2π arasında çizdiriniz ve y'nin sıfırdan geçiş değerini bulunuz.

6) $y=x^4-9x^3+20x^2$ fonksiyonunu 0 ile 5 arasında çizdiriniz ve y'nin sıfırdan geçiş değerlerini bulunuz.

7) $y=x^3-4x$ fonksiyonunu -2 ile 2 arasında çizdiriniz.ve y'nin minimum değerini bulunuz.

8) $y=x^4-3*x^2-x+1$ fonksiyonunu 0 ile 2 arasında çizdiriniz ve y'nin minimum değerini bulunuz.

9) y=3-sin(x)/cos(x/2) fonksiyonunu π/2 ile 3π/2 arasında çizdiriniz ve y'nin minimum değerini bulunuz.

10) $y=x^5-3x^4+2x^2-x+5$ fonksiyonunu 0 ile 3 arasında çizdiriniz ve y'nin minimum değerini bulunuz.

İNTEGRAL ve TÜREV

İntegral alma, matematikle uğraşan bütün kişilerin karşı karşıya kaldığı bir işlemdir. Bilgisayarların gelişimi ile özellikle nümerik integral alma işlemi, yazılan bilgisayar programlarıyla kolayca yapılabilmektedir. Matlab'de nümerik olarak integral ve türev almak mümkündür. Ayrıca program yazılmasına gerek yoktur. Belli komutlar kullanılarak integral alımı mümkündür. Ayrıca sembolik integral hesaplamak da mümkündür.

A) trapz komutu:

Örnekler:

1)

```
clc
x=3:0.1:6;
y=x.^2
alan=trapz(x,y)
```

```
alan =63.0050
```

2)

```
x=-pi/2:pi/40:pi/2;
y=cos(x)+sin(x)
alan=trapz(x,y)
```

```
alan =  1.9990
```

3)

```
clc
x=1:0.01:5
y=x.^3+3*x.^2+2*x+1
alan=trapz(x,y)
```

```
alan =  308.0008
```

B) quad komutu:

Örnekler:

1)

```
alan=quad('x.^2',3,6)
```

```
alan=quad('x.^3+3*x.^2+2*x+1',1,5)

alan =   308
```

C) quad8 komutu:

Örnekler:

1)
```
alan=quad8('1./x',1,exp(2))

alan=2
```

2)
```
alan=quad8('-1./x.^2',2,10)

alan=-0.4
```

3)
```
alan=quad8('cos(x)',-pi/2,pi/2)

alan=2
```

Sembolik İntegral Alma

Matlab de integrallerin sembolik olarak da alınması mümkündür.

Örnek:

```
syms x;
int(2*x)

ans =
        x^2
```

Örnek:
```
syms x;
int((x+1)^2+4*x+2)

ans =

1/3*(x+1)^3+2*x^2+2*x
```

Örnek:

```
syms y;
int(1/(y^2+9))

ans =
1/3*atan(1/3*y)
```

Örnek:

```
syms z;
int(log(z))
ans =

z*log(z)-z
```

Örnek:

```
syms x;
int(1/sqrt(16-x^2))

ans =

asin(1/4*x)
```

Örnek:

```
syms x;
int(1/((x-1)*(x+2)*(x-3)))

ans =
 -1/6*log(x-1)+1/15*log(x+2)+1/10*log(x-3)
```

Sembolik integral hesaplamalarında da başlangıç ve bitiş değerleri verilebilir.

Örnek:

```
syms x;
int(3*x^2,2,4)

ans =
        56
```

Türev

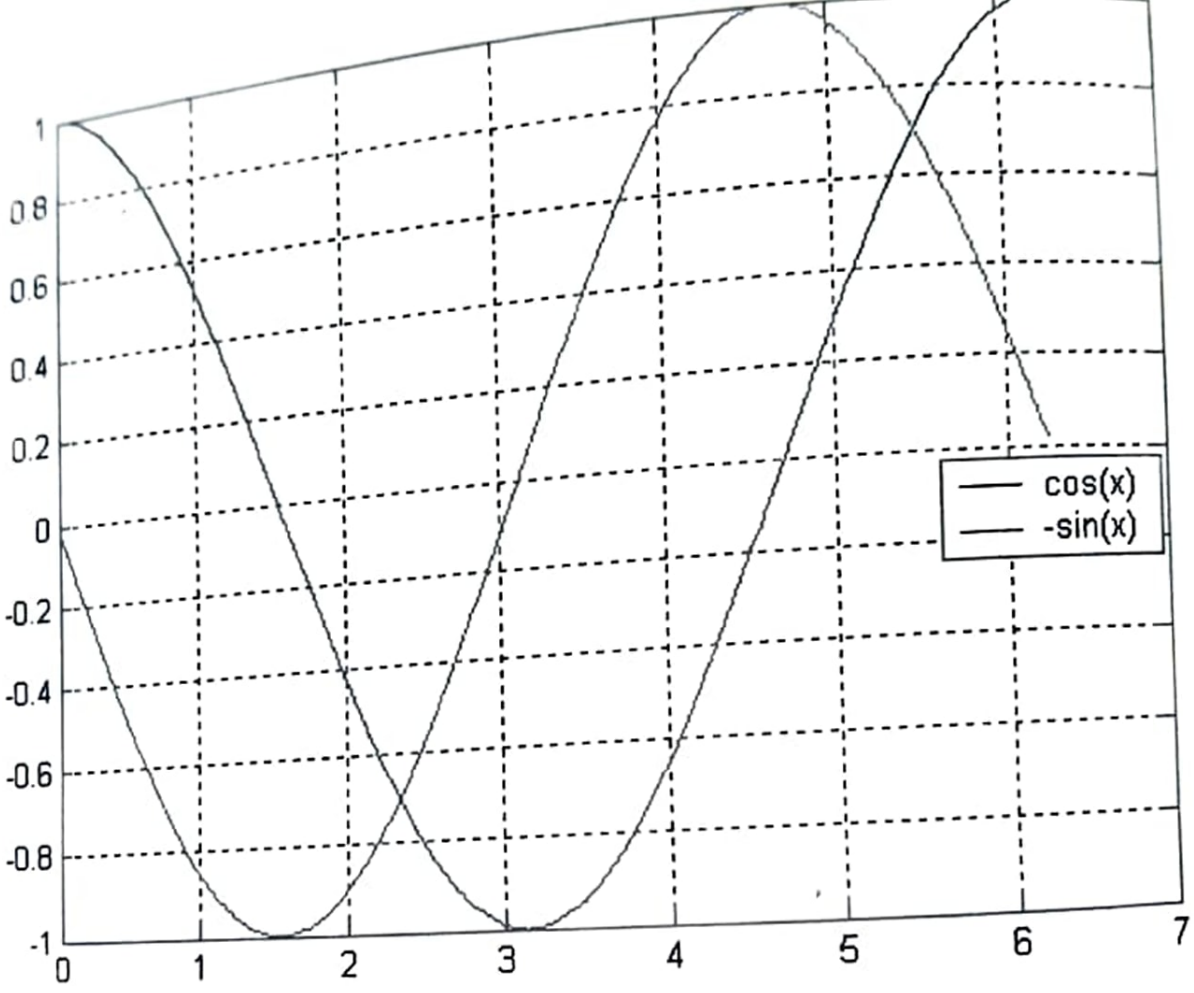
Matematik olarak türev alma şu şekilde ifade edilir.

$$f'(x_i) = \frac{f(x_i + 1) - f(x_i)}{x_{i+1} - x_i}$$

Matlab'de bu formulü diff komutunu kullanarak uygulayabiliriz. Böylece türev alma işlemini yapmış oluruz.

```
a=-2:.5:2;
b=a.^2;
c =diff(b) /diff(a)
```

grid



Alıştırmalar

1) $\int_{-2}^{6}(x-2)^6\,dx$ integralini hesaplayınız.

Cevap:

```
x=-2:0.05:6;
y=(x-2).^6;
trapz(x,y)
quad('(x-2).^6',-1,5)
```

2) $\int_1^5 \frac{x+2}{x^2+4x}dx$ integral

cevap:

```
x=1:.02:5;
y=(x+2)./(x.^2+4*x);
trapz(x,y)
```

3) $\int_1^{10} \ln(x)dx = ?$

cevap:

```
x=1:.01:10;
y=log(x);
plot(x,y)
trapz(x,y)
quad('log(x)',1,10)
```

4) $\int(x^4 e^x)dx$ integralini sembolik olarak hesaplayınız.

```
syms x;
int(x^4*exp(x))
```

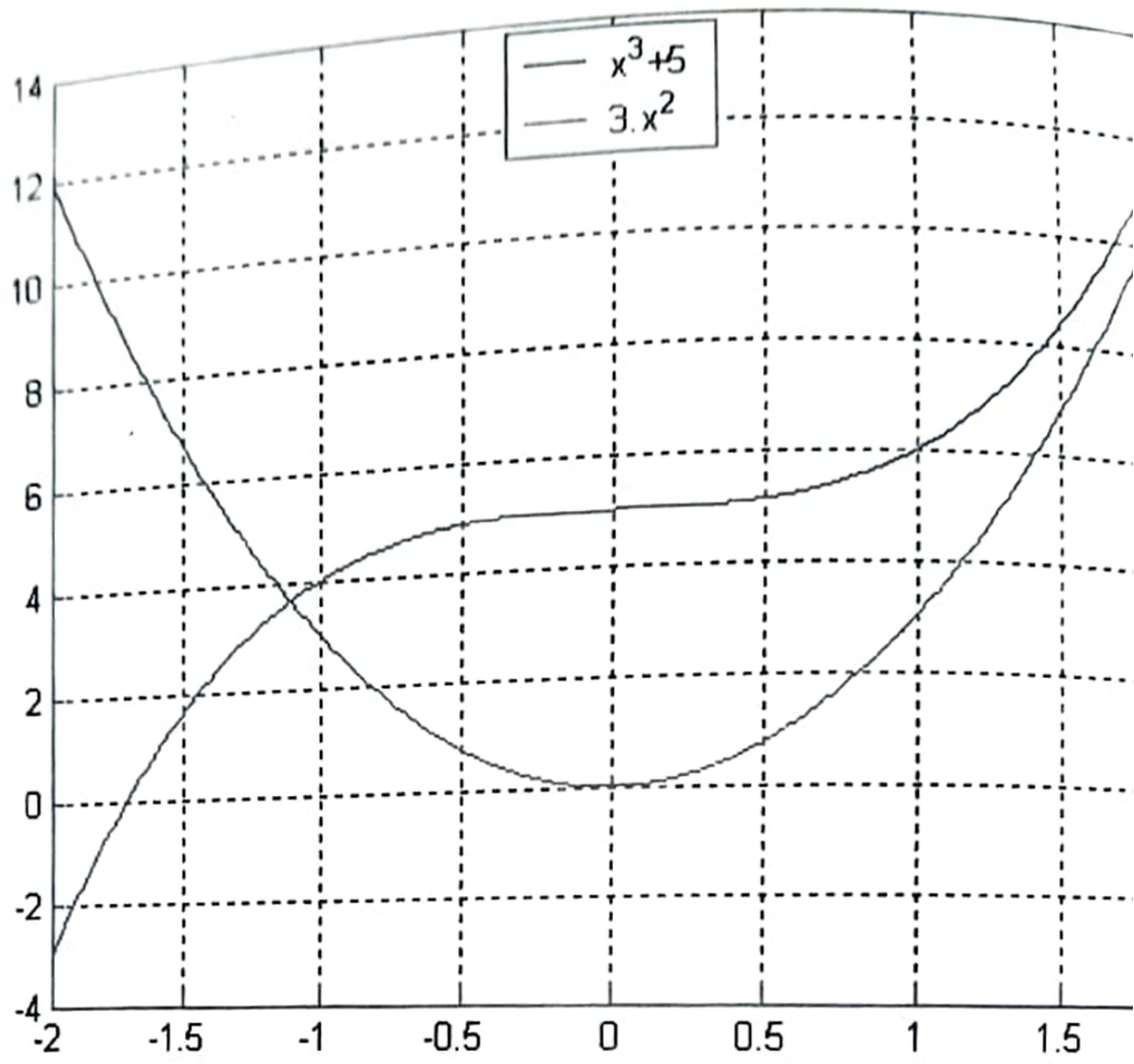
```
ans =
        x^4*exp(x)-4*x^3*exp(x)+12*x^2*exp(x)-24*x*exp(x)+24*exp
```

5) $\int x^2 . \sin x \; dx$ integralini sembolik olarak hesaplayınız.

Cevap:

```
syms x;
```

```
plot(x,y,x,z)
legend('x^3+5','3.x^2')
grid
```



Problemler

1) $\int_1^2 \frac{x^2+2x+1}{x+2} = ?$

2) $\int_2^5 \frac{dx}{(x-1)(x-7)} = ?$

3) $\int_0^{1.5} (x^4 - 3x^2 + 2x + 5)dx = ?$

Aşağıdaki integralleri sembolik olarak bulunuz

4) $\int \sin^2 x.\cos x \ dx = ?$

5) $\int \frac{dx}{x^2 - 9} = ?$

6) $\int 2x\sqrt{x^2 + 1} \ dx = ?$

7) $\int \sin x.\cos x.\tan x \ dx = ?$

8) $\int \cot x \ dx = ?$

9) $\int \frac{dx}{(x-1)(x+2)(x+5)} = ?$

10) $\int \frac{3x^5 + 10x^4 + 15x^3 + 22x^2 + 18x + 4}{x^3 + 2x^2 + 2x + 4} \ dx = ?$

11) $3x^2+5x$ fonksiyonunun türevini bulunuz ve 0 ile 5 arasında çizdiriniz.

12) tan(x) fonksiyonun türevini bulunuz ve istediğiniz aralıkta çizdiriniz.

"IF" Komutu:
if ifadesi karşılaştırma amacı için kullanılır. Eğer if ifadesi ile sorgulanan şart sağlanırsa işlem yapılır. Eğer sorgulanan şart sağlanmazsa işlem yapılmaz.
if ifadesi en basit halde şu şekilde kulanılır:

```
if gerekli şart
  yapılması istenilen ifade
end
```

ifade daha özel durumlar için şu şekilde kullanılır.

```
if gerekli şart1
  yapılması istenilen ifade
elseif gerekli şart 2
  yapılması istenilen ifade
else
  yapılması istenilen ifade
end
```

Karşılaştırmada şu semboller kullanılır:

< : küçük.
> : Büyük.
= = : Eşit.
~ = : Eşit değil.
< = : Küçük veya eşit değil.
> = : Büyük veya eşit değil.

örnek:

```
b=50;
if a<40 b=30
elseif a>60 b=40
else b=80
end
```

Yukarıda yazılmış program için b'nin hangi değer çıkmasını beklersiniz.
Cevap:
Çıkması gereken değer 80'dir. 1. ve 2. if şartları sağlanmadığı için else'den sonra gelen ifade gerçekleştirilir ve b'ye 80 atanır.

Örnek:
Bu örnekte bilgisayarda rasgele bir sayı üretilir. Kullanıcıdan bu sayıyı bulması istenilir. Eğer kullanıcının tahmin ettiği sayı bilgisayarın tutuğu sayıdan büyük ise tutulan sayının küçük olduğu uyarısı verilir. Eğer kullanıcının tahmin ettiği sayı bilgisayarın tutuğu sayıdan küçük ise tutulan sayının büyük olduğu uyarısı verilir. Eğer tahmin edilen sayı bilgisayar tarafından tutulan sayıya eşit ise doğru sayının bulunduğu söylenir. Programın sonunda bilgisayar tarafından tutulan sayı ekrana yazdırılır.

```
a=round(20*rand);
b=input('Bir sayi giriniz.: ');
if a>b fprintf('Küçük sayi girdiniz.')
elseif b>a fprintf('Buyuk sayi girdiniz')
else ('Dogru sayiyi buldunuz')
end
fprintf('\nTutulan sayi %d idi',a)
```

Örnek:
Programda kullanıcıdan
'siz genç bir insansınız', kullanıcının
kullanıcının yaşı 60'dan büyük ise 'siz yaşlı bir

```
a=input('Yasinizi giriniz : ');
if a<20 fprintf('Siz genç bir insansiniz.')
elseif a<60 fprintf('Siz orta yasli bir insansiniz')
else fprintf('Siz yasli bir insansiniz.')
end
```

"FOR" Komutu:

```
for değişken = değişim aralığı
      yapılması istenilen ifade

end
```

Örnekler:

1) a sayısına 1'den 10 kadar değerler verdirerek ekrana yazdırınız.

```
clc
for a=1:10
  a
end
```

2) Aşağıdaki döngülerin sonucunda x'in değerinin kaç çıkmasını beklersiniz?

```
clc
x=0;
for i= 2:2:10
  for j=1:4
  x=x+4;
  end
end
x
```

cevap: i döngüsü 5 defa tekrarlanacaktır. J döngüsü de her i döngüsü için 4 defa tekrarlanacaktır. Dolayısıyla, x, i ve j döngüleri içinde 4*5=20 defa toplanacaktır. Bu durumda x=20*4=80 değerini alacaktır.

3) Aşağıdaki döngülerin sonucunda toplam değerinin kaç çıkmasını beklersiniz?

```
toplam=0;
for k=1:5
  if k<3
    k=k*3
  else
    k=k*2
  end
  toplam=toplam+k;
end
toplam
```

k<3 için k 3 ile çarpılır ve 3 ile 6 sayılarını elde ederiz. diğer durumda k, 2 ile çarpılır ve 6,8,10 sayıları elde edilir. Bu sayıların toplanması sonucunda toplam 33 olarak bulunur.

Örnek:

```
a =
     1     2     3     4     5
     2     4     6     8    10
```

```
   3   6   9   12  15
   4   8   12  16  20
   5   10  15  20  25
```

matrisini iki tane içiçe for döngüsü kullanarak elde ediniz.

```
for i=1:5
  for j=1:5
    a(i,j)=i*j;
  end
end
a
```

örnek:

```
a =
   1   0   0   0   0
   1   1   0   0   0
   1   1   1   0   0
   1   1   1   1   0
   1   1   1   1   1
```

yukarıdaki matrisi iki tane içi içe for döngüsü kullanarak elde ediniz.

```
for i=1:5
  for j=1:i
    a(i,j)=1;
  end
end
a
```

"WHILE" Komutu:

```
while şart ifadesi
  yapılması istenilen ifade
end
```

while döngüsü şart ifadesi sağlandığı sürece işlem yapmaya devam eder. Şart ifadesi sağlanmadığı zaman döngü son bulur.

Örnekler:

1) 2 ile 8 arasındaki sayıları ekrana while-end döngüsü kullanarak yazdırınız.

```
x=2
while x<8
  x=x+1
end
```

2) Sizce aşağıdaki program ne yapmaktadır?

```
a=3
clc
while a~=5
  a=round(10*rand)
end
```

Bu program 1 ile 10 arasında sayılar üretip ekrana yazdırır. a=5 olduğu zaman döngü kesilir. Başlangıçta a=5 olabilme ihtimaline karşılık a'ya başka bir değer atanır.

3) while komutunu kullanarak 20 den 10 a geriye doğru saydırınız.

```
x=20
while x>10
  x=x-1
```

```
   3   6   9   12  15
   4   8   12  16  20
   5   10  15  20  25
```

matrisini iki tane içiçe for döngüsü kullanarak elde ediniz.

```
for i=1:5
  for j=1:5
    a(i,j)=i*j;
  end
end
a
```

örnek:

```
a =
   1   0   0   0   0
   1   1   0   0   0
   1   1   1   0   0
   1   1   1   1   0
   1   1   1   1   1
```

yukarıdaki matrisi iki tane içi içe for döngüsü kullanarak elde ediniz.

```
for i=1:5
  for j=1:i
    a(i,j)=1;
  end
end
a
```

"WHILE" Komutu:

```
while şart ifadesi
  yapılması istenilen ifade
end
```

while döngüsü şart ifadesi sağlandığı sürece işlem yapmaya devam eder. Şart ifadesi sağlanmadığı zaman döngü son bulur.

Örnekler:

1) 2 ile 8 arasındaki sayıları ekrana while-end döngüsü kullanarak yazdırınız.

```
x=2
while x<8
  x=x+1
end
```

2) Sizce aşağıdaki program ne yapmaktadır?

```
a=3
clc
while a~=5
  a=round(10*rand)
end
```

Bu program 1 ile 10 arasında sayılar üretip ekrana yazdırır. a=5 olduğu zaman döngü kesilir. Başlangıçta a=5 olabilme ihtimaline karşılık a'ya başka bir değer atanır.

3) while komutunu kullanarak 20 den 10 a geriye doğru saydırınız.

```
x=20
while x>10
  x=x-1
```

SWITCH Komutu:

Kullanımı:

```
switch değişken
 case 1. durum
   ifade
 case 2.durum
   ifade
...
 otherwise
   ifade
end
```

Örnek:

```
clc
x=4
switch x
case 1
   y=10
case 2
   y=20
case 3
   y=30
otherwise
   y=40
end
```

```
x=4
y=40
```

```
x=2
switch x
case 1
   y=3
case 2
   y=3^2
case 3
   y=3^2
otherwise
   y=100
end
```

```
x =
    2
y =
    9
```

Örnek:

Bu örnekte ekrandan iki sayı girilir. Daha sonra kullanıcıdan 4 matematik işleminden birini seçmesi istenir. Seçilen matematik işlemi yapılır ve ekrana yazdırılır.

```
clc
a=input('1. sayiyi giriniz.: ');
b=input('2. sayiyi giriniz.: ');
fprintf('\n1.Toplama\n2.Çikarma\n3.Çarpma\n4.Bölme\n\n')
c=input('Yapmak istediginiz matematik islemini giriniz.: ');
switch(c)
case 1
  sonuc=a+b
case 2
  sonuc=a-b
case 3
  sonuc=a*b
case 4
  sonuc=a/b
end
```

Alıştırmalar

1) Ekrandan 1 ile 20 arasında bir sayı girilsin. Bu sayı 1 ile 5 arasında ise ekrana sayının 1 ile 5 arasında olduğu yazdırılsın. Sayı 6 ile 10 arasında ise ekrana sayının 6 ile 10 arasında olduğu yazdırılsın. Sayı 11 ile 15 arasında ise sayının 11 ile 15 arasında olduğu yazdırılsın. Sayı 16-ile 20 arasında ise, sayının 1 ile 15 arasında olmadığı yazdırılsın. Bu programı if komutunu kullanarak yapınız.

```
clc
a=input('1 ile 20 arasinda bir sayiyi giriniz.: ');
if (a>0) & (a<=5)
   disp('sayi 1 ile 5 arasinda')
elseif (a>5) & (a<=10)
   disp('sayi 6 ile 10 arasinda')
elseif (a>10) & (a<=15)
   disp('sayi 11 ile 15 arasinda')
else disp('sayi 1 ile 15 arasinda degil')
end
```

2) Ekrandan 5 defa sayı girilmesi istensin ve bu sayıların ortalamaları hesaplatılsın. Bu programı for komutunu kullanarak yazınız.

```
clc
toplam=0;
for i=1:5
   a=input('Bir Sayi giriniz.: ');
   toplam=toplam+a;
end
ortalama=toplam/5
```

3)

```
x =   1   10   4   5   7   6   9
```

şeklinde tanımlı bir x dizisi için switch case komutunu kullanınız ve seçenek 1 ise dizinin maksimum sayısını, seçenek 2 ise dizinin minimum sayısını ve seçenek 3 ise ortalamayı bulunuz.

```
clc
x=[1 10 4 5 7 6 9]
```

```
se 1
 fprintf('Dizinin max. sayisi= %d',max(x))
se 2
 fprintf('Dizinin minimum sayisi= %d',min(x))
se 3
 fprintf('Dizinin ortalamasi=%d',mean(x))
nd
```

) Ekrandan bir sayı giriniz. 1'den bu sayıya kadar olan sayıları ve karelerini yazdırınız. Bu rogramı while komutunu kullanarak yazınız.

```
lc
=input('Bir sayi giriniz.: ');
=0;
hile x<a
  x=x+1;
  fprintf('Sayi=%d  Karesi=%d\n',x,x^2)
end
```

5)

```
a =
   1   3   7   5
   2   1   3   4
   8   9   6   5
   1   7   4   2

b =
   2   1
   8   9
   1   7
```

a matrisinden b matrisini iki tane for döngüsü ile elde ediniz.

Cevap:

```
clc
clear b
a=[1 3 7 5;2 1 3 4;8 9 6 5;1 7 4 2]
for i=2:4
   for j=1:2
      b(i-1,j)=a(i,j)
   end
end
```

6) Ekrandan 2 sayı giriniz. 1.sayı 2. sayıdan büyük ise ekrana 1.sayının 2.sayıdan büyük olduğunu yazınız, 2.sayı 1.sayıdan büyük ise ekrana 2.sayının 1.sayıdan büyük olduğunu yazsın. 1.sayı 2.sayıya eşit ise 1.sayının 2.sayıya eşit olduğunu yazsın. Bu programı if komutunu kullanarak yazınız.

```
clc
a=input('1. Sayiyi giriniz.: ');
b=input('2. Sayiyi giriniz.: ');
if a>b
   fprintf('1.sayi 2. sayidan büyüktür')
elseif b>a
```

```
lse
 fprintf('1. ve 2. sayi birbirine esittir.')
nd
```

```
)
atris =
   2   3   4   5   6
   3   4   5   6   7
   4   5   6   7   8
   5   6   7   8   9
   6   7   8   9  10
```

ukarıdaki matrisi 2 tane while döngüsü kullanarak elde ediniz. İpucu: matrisin her elemanı satır ve sütunların toplamına eşittir.

Cevap:

```
i=0;
while i<4
  i=i+1;j=0;
  while j<4
    j=j+1
    matris(i,j)=i+j
  end
end
```

8) Ekrandan bir sayı ve bir tercih girilsin. Eğer tercih 1 ise sayının karakökü alınsın, tercih 2 ise sayının karesi alınsın. Tercih 3 ise sayının küpü alınsın. Bu programı switch-case komutu kullanarak yazınız.

```
clc
fprintf('1.sayinin karekökünü hesaplar\n2.Sayinin karesini hesaplar\n')
fprintf('3.sayinin kübünü hesaplar\n\n')
a=input('Tercihinizi Giriniz.: ');
fprintf('\n')
b=input('islem yapilacak sayiyi giriniz.: ');
switch a
case 1
   sonuc=sqrt(b)
case 2
   sonuc=b^2
case 3
   sonuc=b^3
otherwise
   disp('yanlis tercih girisi yaptiniz.')
end
```

Ekrandan haftanın kaçıncı günü olduğunu sorunuz. Haftanın 1. günü ise ekrana
günü ise ekrana Salı, 3.günü ise ekrana Çarşamba, 4.günü ise ekrana Perşembe, 5.
rana Cuma, 6.günü ise ekrana cumartesi, 7. günü ise ekrana Pazar yazdırınız. Bu
itch-case komutunu kullanarak yazınız.

Aşağıdaki a matrisini iki tane for döngüsü kullanarak elde ediniz.

```
2 0 0 0 0
2 2 0 0 0
2 2 2 0 0
2 2 2 2 0
2 2 2 2 2
```

Aşağidaki matrisi iki tane while döngüsü kullanarak elde ediniz.

```
1 0 0 0 0
1 2 0 0 0
1 2 3 0 0
1 2 3 4 0
1 2 3 4 5
```

Ekrandan bir harf girilmesini isteyiniz. Girilen harf a ise ekrana haftanın 1.gün
ilen harf b ise ekrana haftanın 2. günü yazdırınız. Girilen harf c ise ekrana haf
dırınız. Girilen harf bu 3 harften farklı ise ekrana a,b,c harflerinden farklı bi
dırınız. Bu programı if komutu kullanarak yazınız.

Aşağıdaki matrisi iki tane for döngüsü kullanarak elde ediniz.

```
0 1 0 1 0
1 0 1 0 1
0 1 0 1 0
1 0 1 0 1
0 1 0
```

KOORDİNAT SİSTEMLERİNİN BİRBİRİNE ÇEVRİLMESİ

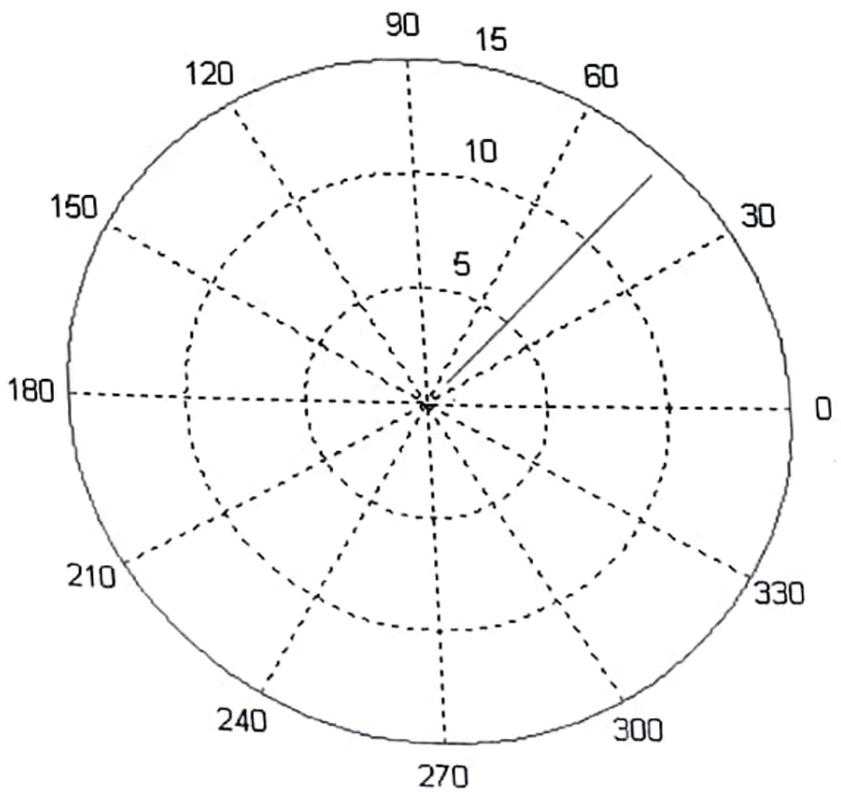
→ **[açı, yarıçap] = cart2pol(x,y):** Bu komut verilen kartezyen koordinatları polar koordinatlara çevirir.

```
x=1:10;
y=x;
[a,b]=cart2pol(x,y)
a=a*180/pi
```

```
a=
0.7854  0.7854  0.7854  0.7854  0.7854  0.7854  0.7854  0.7854  0.7854 0.7854

b =
1.4142  2.8284  4.2426  5.6569  7.0711  8.4853  9.8995 11.3137 12.7279 14.1421

a =
   45   45   45   45   45   45   45   45   45   45
```

Elde edilen sonucu polar koordinatlarda şu şekilde çizdirilebilir.

```
a=[0.7854  0.7854  0.7854  0.7854  0.7854  0.7854  0.7854  0.7854  0.7854 0.7854]
b=[1.4142  2.8284  4.2426  5.6569  7.0711  8.4853  9.8995 11.3137 12.7279 14.1421]
polar(a,b)
```



→ **[açı,yarıçap,yükseklik] = cart**, yarıçap silindirik koordinatlara çevirir. Burada açı, x ek noktanın x-y düzlemine izdüşümünün orijine ulan uzaklığını, noktanın x-y düzleminden yüksekliği de yükseklik olarak ifade edilmiştir.

```
x=[0 4];
y=[12 3];
z=[2 8];
[aci yaricap yukseklik]=cart2pol(x,y,z)
aci=180*aci/pi
```

```
aci =
   0.9273   0.6435
yaricap =
   15    5
yukseklik =
    2    8

aci =
   53.1301   36.8699
```

→ **[xa,za,r] = cart2sph(x,y,z)** : Bu komut verilen kartezyen koordinatları küresel koordinatlara çevirir. Burada xa, x ekseni ile yapılan açı; za, x-y düzlemi ile yapılan açı; r noktanın orijine olan uzaklığıdır.

```
[a b c]=cart2sph(0,3,4)
a=a*180/pi
b=b*180/pi
a =
   1.5708 (radyan)
b =
   0.9273 (radyan)
c =
    5
a =
   90 (derece)
b =
   53.1301 (derece)
```

→ **[x,y] = pol2cart(açı,uzaklık):** Polar koordinatları kartezyen koordinata çevirir.

```
a=60;
a=60*pi/180;
[x y]=pol2cart(a,10)

x =
    5.0000
y =
    8.6603
```

→ [x y z] [illegible]
boyutlu katezyan koordinatlara çevir

```
aci=53.1304*pi/180;
[x y z]=pol2cart(aci,5,12)

x =
   3.0000
y =
   4.0000
z =
    12
```

→ **[x,y,z] = sph2cart(xa,za,r):** Bu komut küresel koordinatları kartezyen koord
çevirir.

```
xa=60*pi/180;
za=30*pi/180;
[x y z]=sph2cart(xa,za,20)

x =
   8.6603

y =
    15

z =
   10.0000
```

Alıştırmalar

1) Aşağıdaki kartezyan koordinatları polar koordinatlara çeviriniz. (Açıyı derece
hesaplar.)

a) x=6 y=8
b) x=9 y=12
c) x=6 y=6

cevap:

```
x=[6 9 6];
```

3.1301

olarak verilen kartezyen koordinatları küresel koordinatlara çeviriniz.

rt2sph(3,4,12)
pi
/pi

3

0

301

801

Problemler

arıçap=20 ve açı=30 derece olarak verilen polar koordinatları, kartezyen koordin
iriniz.

cı=30 derece , r=6 ve z=10 olarak verilen silindirik koordinatları kartezyen koordin
iriniz.

açı1=30 derece, açı2=60 derece
ordinatlara çevir

→ **interp1(a,b,c,'metod'):** Bu komut, bize eldeki verilerden yararlanarak bilmediğimiz bir değeri hesaplamamıza yarar.

Kullanılan metdodlar:

Doğrusal Yaklaşımlar
- 'nearest'
- 'linear'

Kübik Yaklaşımlar
- 'spline'
- 'pchip'
- 'cubic

Eğer herhangi bir metot belirtilmez ise kullanılacak metodun doğrusal olduğu kabul edilir.

Örnekler:

```
a=0:5;
b=[0 10 15 35 40 45];
d1=interp1(a,b,2.5)
d2=interp1(a,b,3.5)

d1 =
   25
d2 =
   37.5000
------
a=-2:2;
b=a.^3;
d1=interp1(a,b,1.5,'spline')
d2=interp1(a,b,0.5,'spline')

d1 =
   3.3750

d2 =
   0.1250
```

Alıştırmalar

1) Aşağıda verilen a ve b dizileri için a=18.4 değerine karşılık gelen b değerini bulunu

```
a=
    2    6   10   14   18   22   26   30   34   38
b =
   20   28   36   44   52   60   68   76   84   92
```

lasyon

(a,b,18.4)

000

Problem

ğıdaki a ve b dizisi için a=10.7 değerine karşılık gelen b değerini 'spline' metodu ile

z.

000	8.4000	8.8000	9.2000	9.6000	10.0000	10.4000	10.8000	11.2000
592.7	681.5	778.7	884.7	1000	1124.9	1259.7	1404.9	

```
interp1(a,b,18.4)

ans =
   52.8000
```

Problem

1) Aşağıdaki a ve b dizisi için a=10.7 değerine karşılık gelen b değerini 'spline' metodu ile bulunuz.

```
a =
    8.0000    8.4000    8.8000    9.2000    9.6000   10.0000   10.4000   10.8000   11.2000

b =
   512   592.7   681.5   778.7   884.7   1000   1124.9   1259.7   1404.9
```

KARAKTER DİZİSİ KOMUTLARI

→strcmp
(s1,s2) : Eğer s1 ve s2 komutları birbirine eşit ise 1, eşit değilse 0 sonucu elde edilir.

Örnek 1:

```
>> a=strcmp('araba','bilgisayar')
a =
    0
>> b=strcmp('bilgisayar','bilgisayar')
b =
    1
```

Örnek 2:

```
>> a=strcmp('bilgisayar','disket')
a =
    0
>> b=strcmp('defter','defter')
b =
    1
```

Eğer bu komuta rakam eklenirse ve bu eklenen rakam kadar string'in karakterleri eşitse sonuç yine 1 olur.

→ strncmp(str1,str2,n)
Eğer str1 ve str2'nin ilk n harfi eşitse bu komut 1 değerini verir. Not: Bu karşılaştırma büyük-küçük harflere karşı duyarlıdır.

```
>> strncmp('abcde','abcfg',3)
ans =
    1
>> strncmp('abdefg','abhijk',2)
ans =
    1
>> strncmp('defter','ders',3)
ans =
    0
```

→ İki karakter dizisini şu operatörlerle karşılaştırmak mümkündür:(>, >=, <, <=, ==, ~=)

```
dizi1='edirne'
dizi2='efsane'
dizi1= =dizi2

ans =

    1    0    0    0    1    1
```

Birbirine denk gelen harflerden aynı olanları 1, aynı olmayanları 0 değerini alır.

) 0 1 0
sonra gelen harf önce gelen

1 ~=dizi2

1 1 1 0 0
karakterler aynı değil ise sonuç1, aynı ise sonuç 0'dır.

i1<=dizi2

1 1 0 1 1

letter('dizi'): Bu komut eğer karakter harf ise 1 sonucunu verir. Diğer durumlarda 0
erini verir.

nek 1:
isletter('saat 10.')
s =
1 1 1 1 0 0 0 0

rnek 2:
> isletter('3 tane kitap')
ns =

0 0 1 1 1 1 0 1 1 1 1 1

→isspace(dizi): Bu komut eğer karakter boşluk ise 1 değerini verir. Diğer durumlarda 0
değerini verir.

Örnek 1:
>> isspace('az cok')
ans =
0 0 1 0 0 0

Örnek 2:
>> isspace('matlab ve simulink')
ans =
0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0

→ strrep(dizi,aranın kısım,yeni kısım): Bu komut ile bir dizide bir bölümü aramak ve o
bölümü değiştirmek mümkündür.

Örnek 1;

```
>> dizi='Ders saat 10:00 da baslayacak'
dizi =
Ders saat 10:00 da baslayacak
>> strrep(dizi,'10:00','12:00')
ans =
Ders saat 12:00 da baslayacak
```

Örnek 2:

```
>> a='sinifin kapasitesi 15 kisidir';
>> b=strrep(a,'15','30')
b =
sinifin kapasitesi 30 kisidir
```

→ **findstr(dizi1,dizi2):** Bu komut, uzun bir dizi içerisinde yer alan kısa bir dizinin başlangıç konumunu belirtir.

Örnek 1:

```
>> dizi1='Bu gün hava güzel';
>> dizi2='gün';
>> findstr(dizi1,dizi2)
ans =
    4
```

Örnek 2:

```
dizi1='kitapçidan 4 adet kitap aldim.'
dizi2='kitap'
findstr(dizi1,dizi2)
ans =
    1   19
```

Örnek 3:

```
dizi1='Bana bu kitaptan dort tane lazim'
dizi2='t'
findstr(dizi1,dizi2)

ans =
   11   14   21   23
```

Not: dizilerdeki karakterlerin büyük veya küçük harf olması önemlidir.

→ **strvcat(dizi1,dizi2,dizi3,.....):** Bu komut verilen dizileri satırlar şeklinde düzenler.

```
>> strvcat('kitap','defter','kalem')
ans =
kitap
defter
kalem
```

karakterin kaç

Örnek 1:
```
b=strvcat('kalem','kitap','kasa')
c=strmatch('ka',b)

b =
kalem
kitap
kasa

c =
   1
   3
```

Eğer komut strmatch(dizi, dizi satırları,'exact') şeklinde kullanılıyorsa dizi karşılaştırılan diz ile birebir aynı olmalıdır.

Örnek 2:
```
b=strvcat('kalem','kitap','kasa','ka')
c=strmatch('ka',b,'exact')

b =
kalem
kitap
kasa
ka

c =
   4
```

→ **strcat(s1,s2,s3,...):** Bu komut verilen dizileri birbirine ekler.

Örnek 1:
```
s1='Dün'
s2='seni'
s3='gördüm'
c=strcat(s1,s2,s3)

c =
Dünsenigördüm
```

→ **mat2str(a):** Bu komut matematik değerleri yazı karakteri haline çevirir.
Örnek 1:
```
a=[1 2 3; 2 1 1]
b=mat2str(a)

a =
   1   2   3
   2   1   1
```

→ **strmatch(dizi, dizi satırları):** Bu komut verilen dizi satırları içinde aranan dizinin veya karakterin kaçıncı satırda olduğunu arar.

Örnek 1:
```
b=strvcat('kalem','kitap','kasa')
c=strmatch('ka',b)

b =
kalem
kitap
kasa

c =
     1
     3
```

Eğer komut strmatch(dizi, dizi satırları,'exact') şeklinde kullanılıyorsa dizi karşılaştırılan dizi ile birebir aynı olmalıdır.

Örnek 2:
```
b=strvcat('kalem','kitap','kasa','ka')
c=strmatch('ka',b,'exact')

b =
kalem
kitap
kasa
ka

c =
     4
```
→ **strcat(s1,s2,s3,...):** Bu komut verilen dizileri birbirine ekler.

Örnek 1:
```
s1='Dün'
s2='seni'
s3='gördüm'
c=strcat(s1,s2,s3)

c =
Dünsenigördüm
```

→ **mat2str(a):** Bu komut matematik değerleri yazı karakteri haline çevirir.
Örnek 1:
```
a=[1 2 3; 2 1 1]
b=mat2str(a)

a =
     1     2     3
     2     1     1
```

```
b =
[1 2 3;2 1 1]
```

Örnek 2:

```
a=eye(3)
b=mat2str(a)

a =
   1   0   0
   0   1   0
   0   0   1
b =
[1 0 0;0 1 0;0 0 1]
```

→ **eval(dizi):** matlab, karakter halinde verilen diziyi matlab komutu olarak yorumlar.

```
eval('2:5')

ans =
   2   3   4   5
```

→ **deblank(dizi):** Bu komut bir dizinin sonundaki boş karakterleri diziden atar.

```
>> A{1,1}='istanbul  '

A =
  'istanbul  '

>> deblank(A)

ans =
  'istanbul'
```

→ **lower('dizi'):** Bu komut dizideki büyük harfleri küçük harflere çevirir.

```
>> lower('PC')

ans =

pc

>> lower('KUTAHYA')
ans =
kutahya
```

→**strjust(dizi) ve strjust(dizi,'right'):** Bu komutlar diziyi sağa doğru kaydırırlar.

```
>> a='   kitap          '
a =
   kitap
>> b=strjust(a)
b =
          kitap
>> [a;b]
ans =
   kitap
          kitap
```

→ **strjust(dizi, 'left'):** Bu komut diziyi sola doğru kaydırır.

```
>> c=strjust(a,'left')
c =
kitap
>> [a;c]
ans =

   kitap
kitap
```

→**strjust(dizi,'center'):** Bu komut diziyi ortalar.

```
>> d=strjust(a,'center')

d =
      kitap
>> [a;d]

ans =
   kitap
     kitap
```

→ **[alınan kalan] = strtok('dizi',sınırlayıcı):** Bu komutta bir dizinin sınırlayıcıya kadar olan bölümü alınana atanır, geri kalan kısmı da kalana atanır.

Örnek 1:

```
>> d='Bugün trafik çok yogun'
d =
Bugün trafik çok yogun

>> [a k]=strtok(d,'f')
a =
Bugün tra

k =
fik çok yogun
```

Eğer herhangi bir sınırlama kullanılmazsa ilk kelimeden itibaren bölme gerçekleştirilir.

Örnek 2:
>> [a k]=strtok(d)

a =
Bugün

k =
trafik çok yogun

Örnek 3:

```
a='roots komutu polinomun koklerini bulur'
[b c]=strtok(a,'p')

a =
roots komutu polinomun koklerini bulur
b =
roots komutu
c =
polinomun koklerini bulur
```

→ **symvar(ifade):** Bu komut bir ifade içinde verilen değişkenleri bulur.

```
>>symvar 'cos(2*pi*f1 - teta1) + sin(2*pi*f2+teta2)'

ans =
    'f1'
    'f2'
    'teta1'
    'teta2'
```

→ **upper('dizi'):** Bu komut, küçük harfleri büyük harflere çevirir.

```
>> upper('araba')

ans =
ARABA
```

→ **char(a):** Bu komut verilen sayıyı , kendisine karşılık gelen karaktere çevirir.

```
>> char(65)
ans =
A
```

Örnek 2:

```
for i=65:3:90
   fprintf('%d %s %d %s %d %s\n',i,char(i),i+1,char(i+1),i+2,char(i+2))
end
```

```
5 A  66 B  67 C
8 D  69 E  70 F
1 G  72 H  73 I
4 J  75 K  76 L
7 M  78 N  79 O
0 P  81 Q  82 R
3 S  84 T  85 U
6 V  87 W  88 X
9 Y  90 Z  91 [
```

→ **str2mat(a1,a2,a3..)**: Bu komut verilen karakterleri sayıya çevirerek matris oluşturur.

```
clc
a = str2mat('432','653','764','987')

a =
432
653
764
987

>> a(2,2)
ans =
5
```

→ **x = str2num('dizi')** : Bu komut bir karakteri rakamı çevirir.

```
a=str2num(['12 ','54 ','78 ','24 '])

a =
   12   54   78   24
```

→ **bin2dec(binary sayı):** Verilen binary sayıyı decimale çevirir.

```
bin2dec('101101')

ans =
   45
```

→ **dec2bin(decimal sayı)**: Verilen decimal sayıyı binarye çevirir.

```
dec2bin(45)

ans =
101101
```

→ **dec2hex(decimal sayı):** Bu komut decimal sayıyı hex sayıya çevirir.

```
dec2hex(26)
ans =
1A
```

→ **hex2dec('hex sayı'):** Bu komut veilen hex sayıyı decimale çevirir.

```
hex2dec('1AB')

ans =

  427
```

Alıştırmalar

1) Araba ve arama kelimelerinin ilk 3 harfini karşılaştırınız.

```
>> strncmp('araba','arama',3)
ans =
   1
```

2) Akarsu ile badana kelimelerini karşılaştırınız.

a) Karşılıklı harfler birbiri ile aynı ise 1, değil ise 0 üretilsin.

```
k1='akarsu';
k2='badana'

>>k1==k2
ans =

   0   0   0   0   0   0
```

b) Karşılıklı harfler birbiri ile aynı değil ise 1, aynı ise 0 üretilsin.

```
>> k1~=k2

ans =

   1   1   1   1   1   1
```

c) 1. kelimenin harfleri 2. kelimenin harflerinden büyük ise1, değil ise 0 üretilsin.

```
>> k1>k2

ans =

   0   1   0   1   1   1
```

d) 2. kelimenin harfleri 1. kelimeden büyük ise 1, değil ise 0 üretilsin.

```
>> k1<k2

ans =

   1   0   1   0   0   0
```

3) Aşağıdaki cümle için karakterler harf ise 1, harf değil ise 0 üretilsin.

Tren saat 12 de kalkacak.

Cevap:

```
cumle='Tren saat 12 de kalkacak.'
isletter(cumle)

ans =
  Columns 1 through 14
    1    1    1    1    0    1    1    1    1    0    0    0    0    1
  Columns 15 through 25
    1    0    1    1    1    1    1    1    1    1    0
```

4) Aşağıdaki cümlede boşluklar için 1 diğer durumlar için sıfır üretilsin.

Cevap:

```
 cumle='2 defa geldi';
isspace(cumle)
ans =

    0    1    0    0    0    0    1    0    0    0    0    0
```

5) Dün sabah saat 8:00 de geldi cümlesi içinde yer alan sabah kelimesini akşam kelimesi ile yer değiştiriniz.

```
cumle='Dün sabah saat 8:00 de geldi.'
strrep(cumle,'sabah','aksam')

ans =
Dün aksam saat 8:00 de geldi.
```

6) Gebze Yüksek Teknoloji Enstitüsü ismi içerisinde yer alan 'e' harflerinin konumunu belirtiniz.

```
findstr('Gebze Yüksek Teknoloji Enstitüsü','e')

ans =
    2    5   11   15
```

7) Kütüphaneye yeni kitaplar gelmiş cümlesini kitaplar kelimesinden itibaren ikiye bölünüz.

```
[a,k]=strtok('Kutuphaneye yeni kitaplar gelmis.','k')

a =
Kutuphaneye yeni

k =
kitaplar gelmis.
```

8) BaHaRAT kelimesinin büyük harflerini küçük harflere çeviriniz.

```
lower('BaHaRAT')

ans =
baharat
```

9) fabrika kelimesinin harflerini büyük harflere çeviriniz.

```
upper('fabrika')

ans =
FABRIKA
```

10) İkilik tabanda verilen 0100110010011010 sayısını 16lık tabana çeviriniz.

```
sayi=bin2dec('0100110010011010');
sonuc=dec2hex(sayi)

sonuc =
        4C9A
```

Problemler

1) Uzunluğu aynı olan iki kelimeyi karşılaştırınız. 1 . kelimenin harfleri 2. kelimenin harflerinden büyük ise 1, küçük ise 0 üretilsin.

2) Aşağıdaki cümle için karakterler harf ise 1, değil ise sıfır üretilsin.
Kargoyu bugün teslim alacağım.

3) Aşağıdaki cümlede boşluklar için 1, diğer durumlar için 0 üretilsin.
Yeni bir bilgisayara ihtiyacım var.

4) Bütün gün boyunca çalıştım cümlesini iki saat boyunca çalıştım cümlesine çeviriniz.

5) Bugün 5 yeni kitap aldım cümlesi içindeki y harfinin konumu belirtiniz.

6)Dün tatile girdik cümlesini tatile kısmından itibaren ikiye ayırınız.

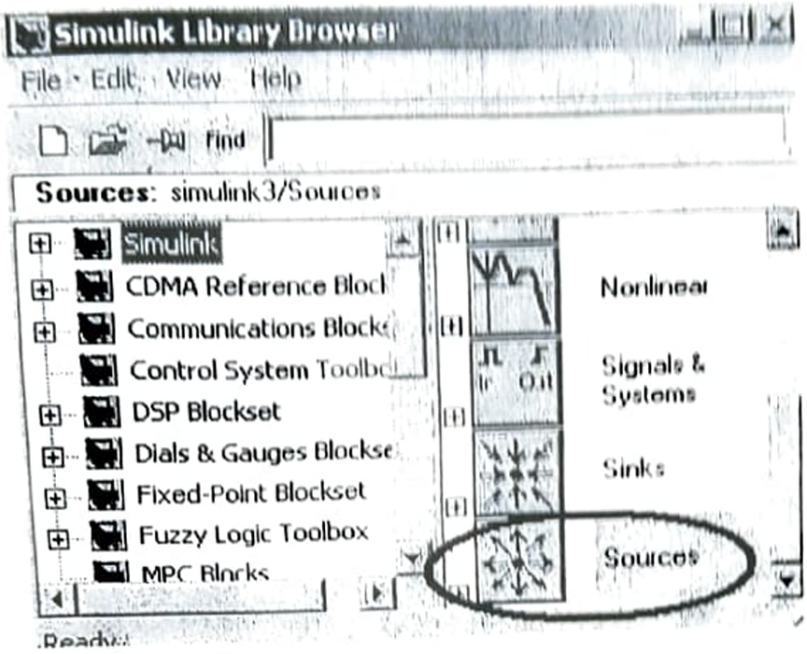
7) Bilgisayar kelimesini büyük harflere çeviriniz.

8) 16'lık tabanda verilen 6D8E sayısını ikilik tabana çeviriniz.

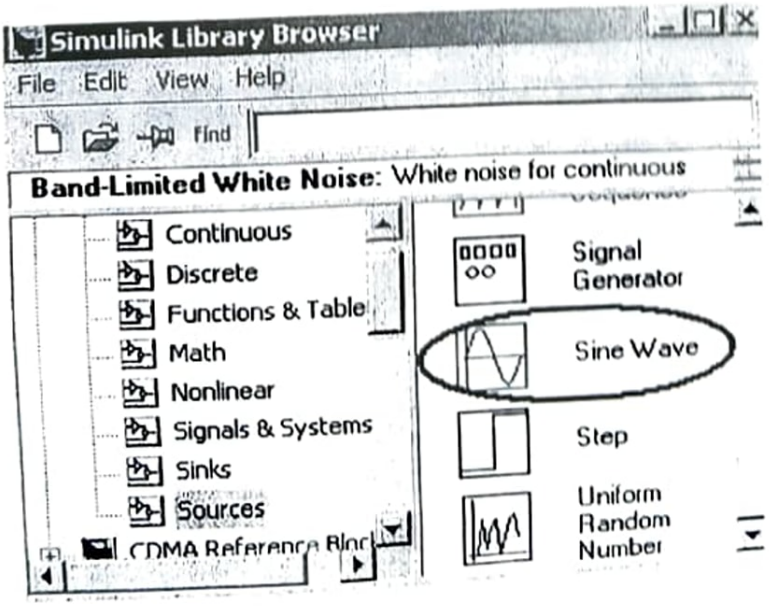
Simulink'i çalıştırmak için command window'da iken simulink yazılıp enter a basılır. Karşımıza simulink library browswe çıkar. Buradan file/new/model 'i seçersek karşımıza model penceresi çıkar. Simulinkte model oluşturmak için simulink library browser daki birimler mouse ile model penceresine taşınır ve aralarında bağlantılar yapılır.

Örnek: Sinüs dalganın osiloskop ile görüntülenmesi.

1) Simulink library browser'dan sources'ı seçiniz.

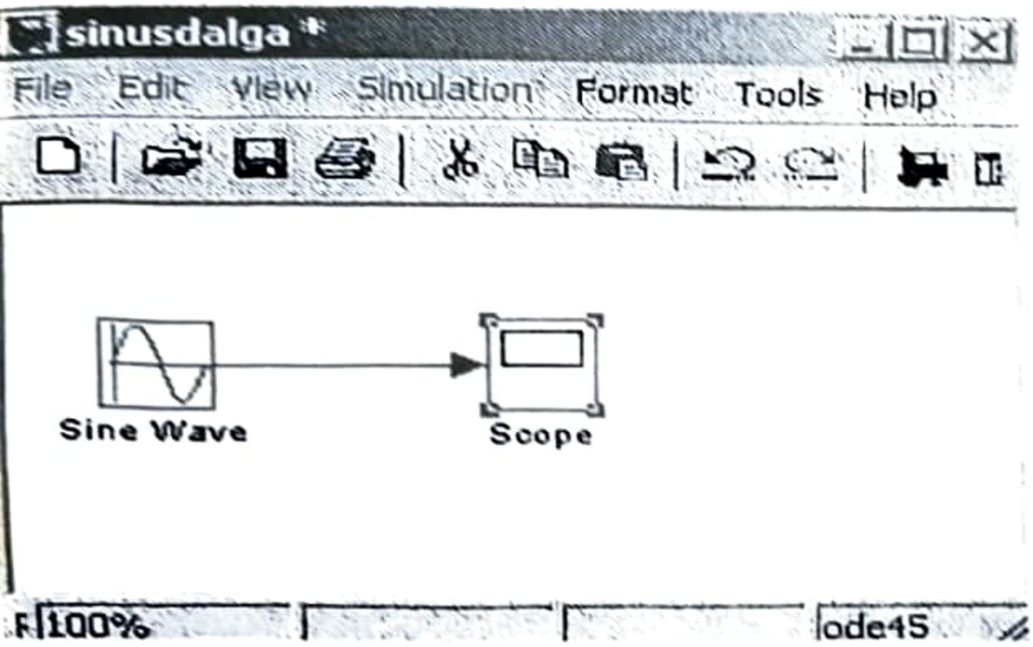


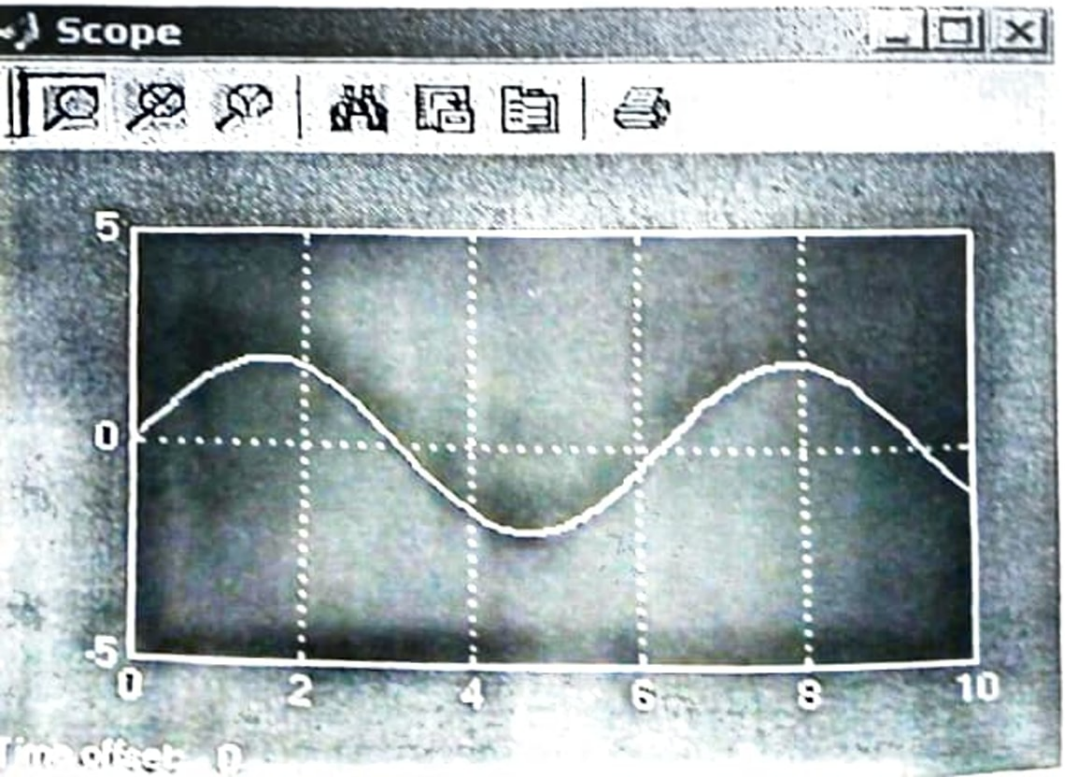
2) Karşınıza gelen ekrandan sine wave'i seçiniz.



3) Sonra bu birimi mouse ile model penceresine taşıyınız.

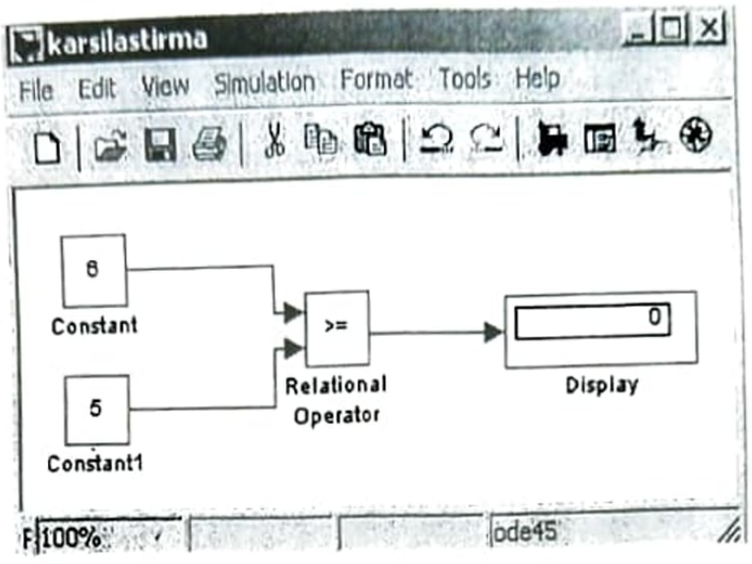
ı scope'u seçiniz ve model penceresine taşıyınız. Sinüs dalganın girişine mouse yardımıyla bir çizgi çekiniz. Sine wave'i çift rekansını değiştirebilirsiniz. Şu an model simule edilmeye hazırdır. ceresinden simulation kısmından start'ı seçiniz .Simule edilen cope üstüne çift tıklayınız.





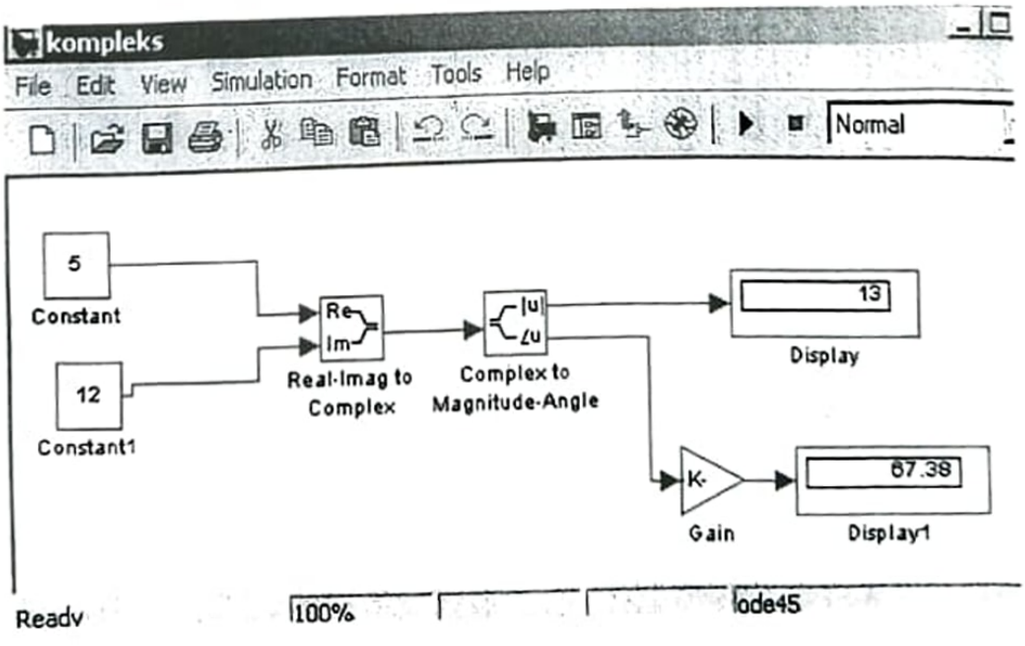
Örnekler:

1) Karşılaştırma örneği:



Önce sources kısmından iki tane constant alınır. Üstlerine çift tıklanarak değerlerine 5 ve 6 atanır. Daha sonra math kısmından relational operator seçilir. Bu birim karşılaştırma yapmak için kullanılır. Bu işlemden sonra sinks kısmından display seçilir. Bu model simule edildiğinde sonuç 1 çıkacaktır. Çünkü 6, 5 ten büyüktür. Relational operator'ün üstüne çift tıklanarak karşılaştırma yapılacak işaretin türü değiştirilebilir.

2) Verilen iki sayının önce kompleks sayıya çevrilmesi, sonra genlik ve açısının bulunması:

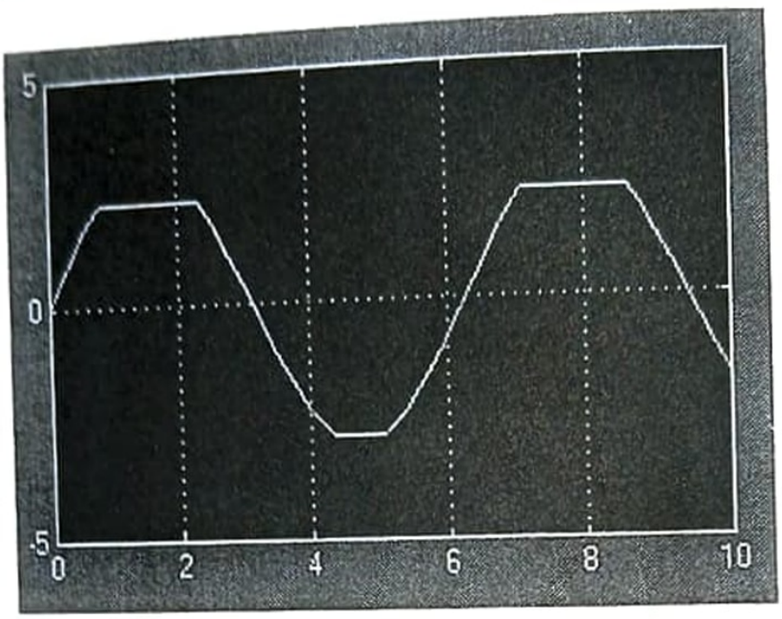


Sources kısmından iki tane constant, sinks kısmından iki tane display, math kısmından real-imag to complex ve complex to magnitude-angle ve gain seçilir. Burada açının radyandan komplekse çevrilmesi için Gain 180/pi seçilir. Sayı girişleri değiştirerek farklı sayılar için de büyüklük ve açı hesaplaması yapabiliriz.

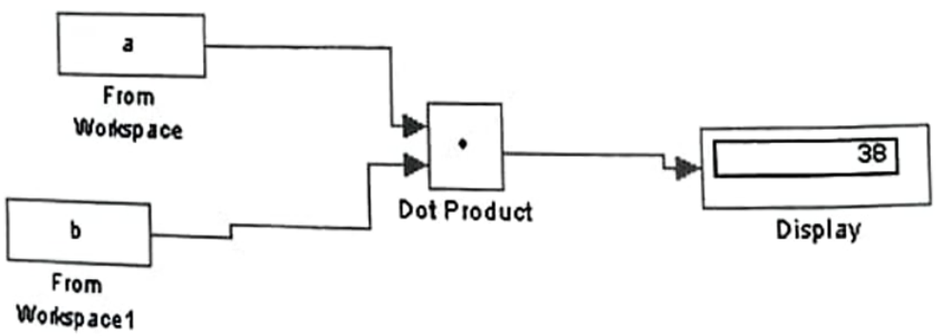
3) Girilen bir sinyalin alt ve üst sınırının belirlenmesi:



source kısmından sine wave, sinks kısmından scope ve nonlinear kısmından saturation seçilir. Bu örnek de sinüsün genliği +3 ile –3 arasında değişmektedir. Saturation biriminin üst limiti 2.2, alt limiti de –2.2 dir.

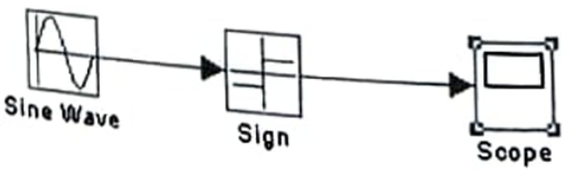


4) Nokta çarpımı:



source kısmından 2 tane workspace, sinks kısmından 1 tane display ve math kısmından 1 tane dot product seçilir. Burada a ve b vektörleri command penceresinden girilmiş olmalıdır. Simulink çarpım sırasında vektörlerin ilk elemanlarını dikkate almaz.

5) Sign işaretinin kullanılışı.



Source kısmından sine wave, sinks kısmından scope ve ve math kısmından sign seçilir. Burada sing işareti sinüsün 0'dan büyük olduğu yerlerde1, küçük olduğu yerlerde –1 üretir. Bu durumda aşağıdaki sonuç ortaya çıkar.

```
c =
   720
```

5)

```
a =
     1     2     3
     1     4     2
     1     5     3

b =
     1     2     3
     2     6     5
     3    11     8
```

Yukarıdaki a matrisinden b matrisini elde ediniz. (Not: iki matris arasındaki ilişkiye dikkat ediniz.)

Cevap:
B Matrisinin sütunlarına dikkat edilirse a matrisinin sütunlarının teker teker toplanıldığından oluştuğu gözükür. Bu durumda çözüm:

```
a=[ 1 2 3; 1 4 2; 1 5 3]
b=cumsum(a)
```

şeklinde olur.

6)

```
a =
     1     2     3     2     3     2     1     2

b =
     1     2     6    12    36    72    72   144
```

a dizisinin elemanlarından b dizisini elde ediniz.

Cevap:

B dizisinin elemanlarına dikkat edilirse a dizisinin elemanlarının teker teker çarpımından oluştuğu gözükür. Bu durumda çözüm:

```
a=[ 1 2 3  2 3 2  1 2 ]
b=cumprod(a)
```

şeklinde olur.

7) Aşağıdaki dizinin ortalamasını bulunuz.

```
a =
```

```
c =
   720
```

5)

```
a =
     1     2     3
     1     4     2
     1     5     3

b =
     1     2     3
     2     6     5
     3    11     8
```

Yukarıdaki a matrisinden b matrisini elde ediniz. (Not: iki matris arasındaki ilişkiye dikkat ediniz.)

Cevap:
B Matrisinin sütunlarına dikkat edilirse a matrisinin sütunlarının teker teker toplanıldığından oluştuğu gözükür. Bu durumda çözüm:

```
a=[ 1 2 3; 1 4 2; 1 5 3]
b=cumsum(a)
```

şeklinde olur.

6)

```
a =
     1     2     3     2     3     2     1     2

b =
     1     2     6    12    36    72    72   144
```

a dizisinin elemanlarından b dizisini elde ediniz.

Cevap:

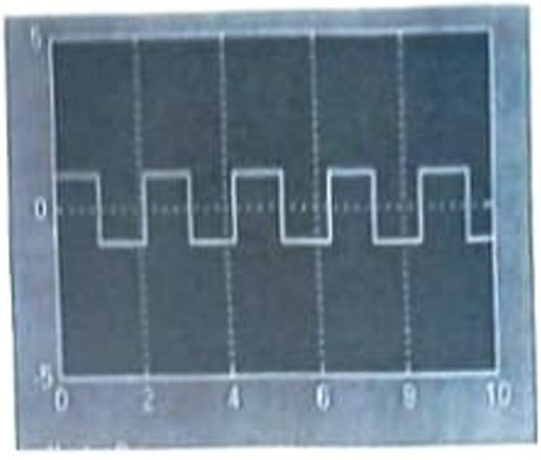
B dizisinin elemanlarına dikkat edilirse a dizisinin elemanlarının teker teker çarpımından oluştuğu gözükür. Bu durumda çözüm:

```
a=[ 1 2 3  2 3 2  1 2 ]
b=cumprod(a)
```

şeklinde olur.

7) Aşağıdaki dizinin ortalamasını bulunuz.

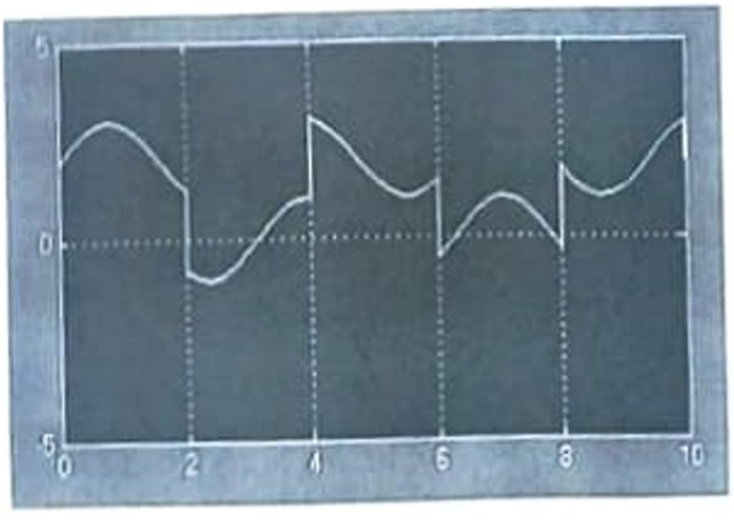
```
a =
```



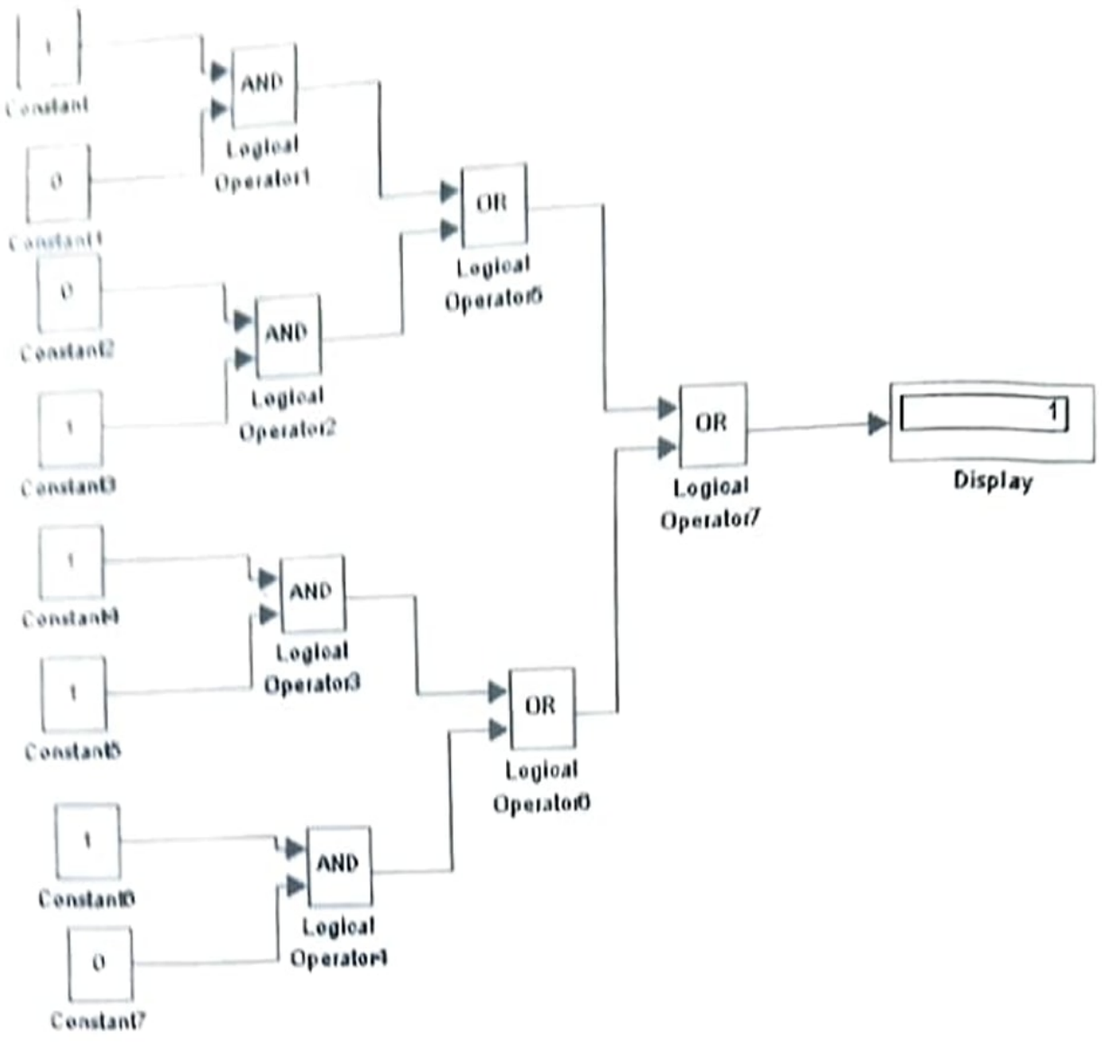
6) 2 sinyalın toplanması.



Sources kısmından pulse generator ve sine wave , sinks den scope ve math kısmından sum seçilir. Bu model de iki sinyal toplanır ve gösterilir. Sum'ın üstünde çift tıklarsanız + işaretlerinden birini – yapabilirsiniz. Bu durumda çıkartma işlemi yapılır.

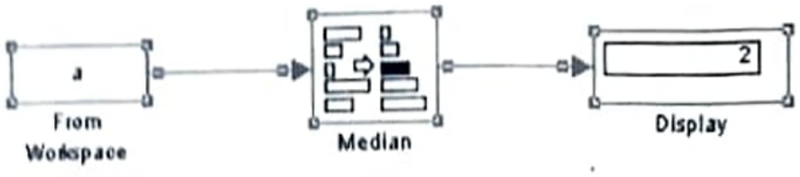


7)
{(1 ve 0) veya (0 ve 1) }veya {(1 ve 1) veya (1 ve 0)}
Yukarıdaki lojik işlemleri matlab simulink i kullanarak yapınız ve sonucu bulunuz.
Not: and işlemi math kısmından seçilir. And in üstüne çift tıklanarak or veya başka lojik operatör seçilebilir.



8) Bir dizinin ortasındaki sayının bulunması:

Bir dizinin ortasındaki sayı median komutu kullanarak elde edilir. Yalnız simulink'te dizinin ilk elemanı dikkate alınmaz.



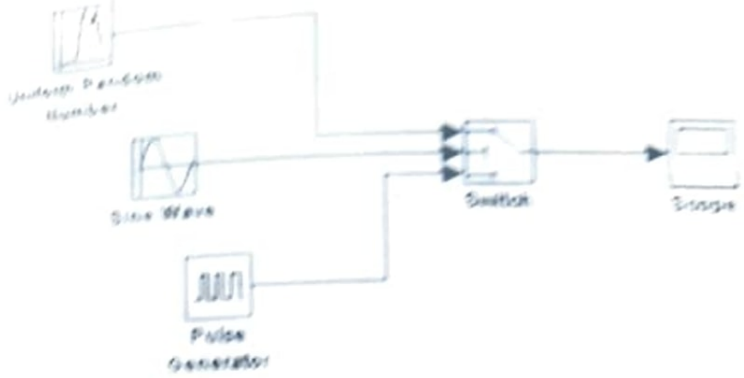
Alıştırmalar

1) a = [3 4 5 6 10 5 7 9 12 6 3] dizisinin maksimum elemanını bulunuz.

Cevap:
Matlab'da command window'da iken a dizisini şu şekilde giriniz.

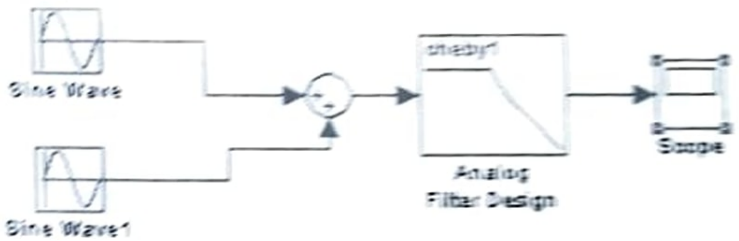
a=[0 3 4 5 6 10 5 7 9 12 6 3]

Daha sonra simulink'te aşağıdaki tasarım yapılır.

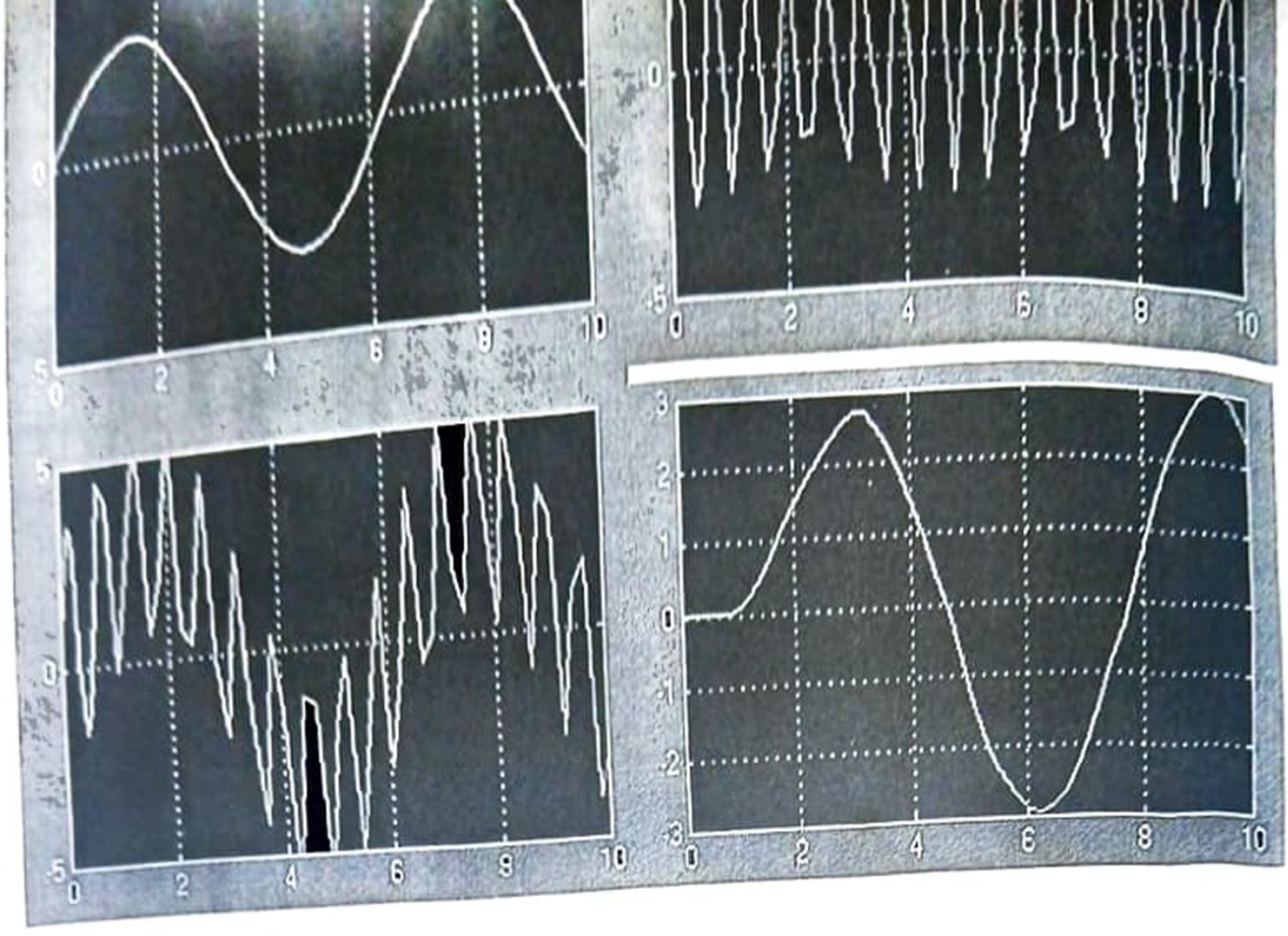


3) Frekansı 1 rad/sn olan bir sinüs ile frekansı 10 rad/sn. olan bir sinüsü toplayınız. Daha sonra elde edilen bu toplamı alçak geçiren bir filtreden geçirerek 1 rad/sn'lik sinüsü geri elde ediniz.

Cevap:



Filtre; DSP blockset /filtering/filter designs altından çağrılır.
Aşağıda 1 rad'sn sinüs, 10 rad/sn.'lik sinüs ikisinin toplamı ve filtre çıkışındaki sinyal verilmiştir.



Problemler

1) simulinkte a=[2 9 4 10 5 12 6 7 8 12] dizisinin minimumunu hesaplayan bir tasarım yapınız.

2) 2 rad/sn'lik ve 15 rad/sn lik sinüsleri toplayınız daha sonra yüksek geçiren bir filtre kullanarak 2. sinüsü elde ediniz.

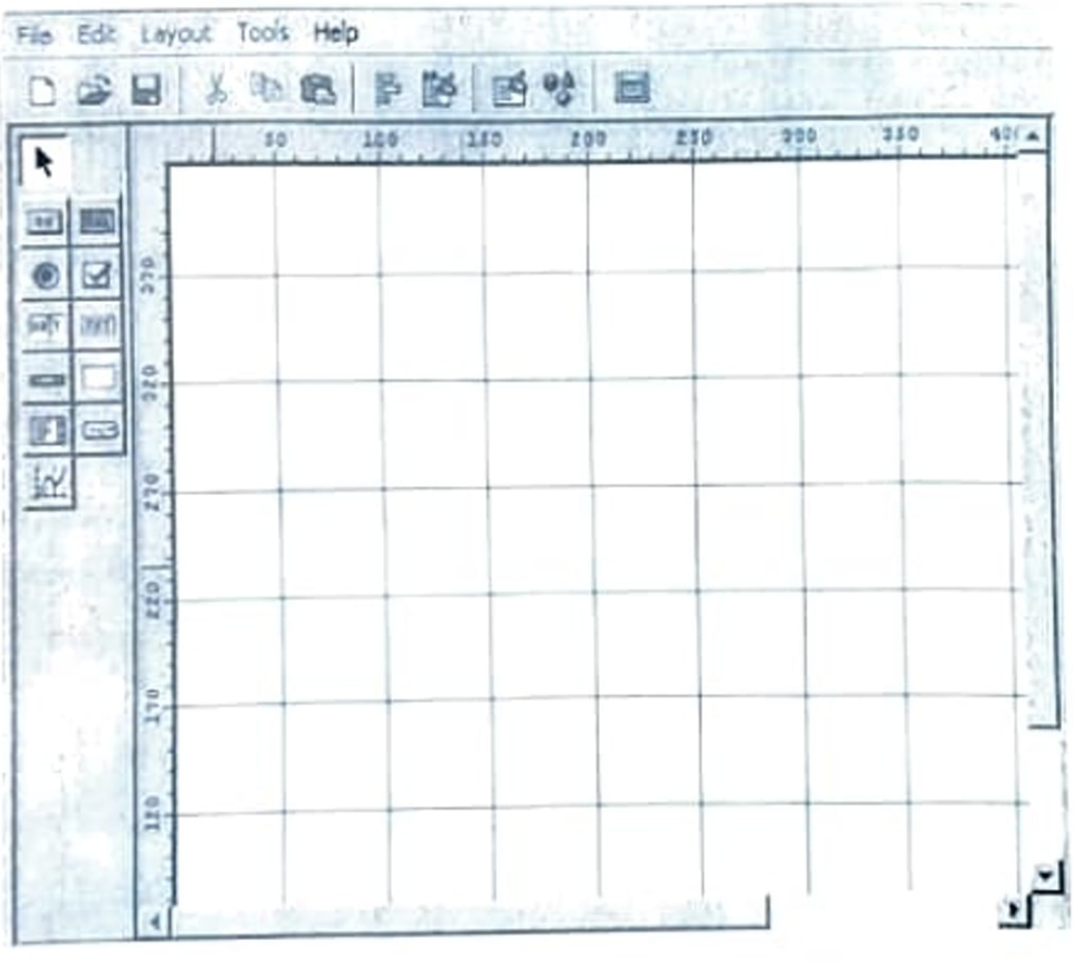
3) Dials&Gauges Blockset bölümünden seçeceğiniz 'lower left' göstergesinde ibreyi 60'a getiriniz.

GUİ (Grafiksel Kullanıcı Ara Yüzü)

Matlab de grafiksel kullanıcı ara yüzü kullanmak mümkündür. Eğer bir matematiksel ifadenin sadece belli birimlerini değiştiriyorsak bu arayüzü kullanmak büyük kolaylıklar getirecektir. Bu birimlerdeki değişikliklere karşı tüm ifadede meydana gelen değişimler, pratik bir şekilde izlenir. Değişimler bir grafik üzerinde de izlenebilir. Örneğin frekansı değiştirildiğinde bir sinüste meydana gelen değişiklik grafik üzerinde gözlemlenebilir.

Kullanıcı arayüzü çeşitli araçlar kullanılarak hazırlanır. Daha sonra bu araçların durumlarına göre Matlab programı yazılır . Hangi araç, hangi konumda ise bu konumla ilgili Matlab programı çalışır.

Kullanıcı ara birimini hazırlamak için Matlab'de command window'da guide komutu yazılır. Karşımıza aşağıda gösterilen, arayüzü hazırlamak için gerekli birim çıkar.



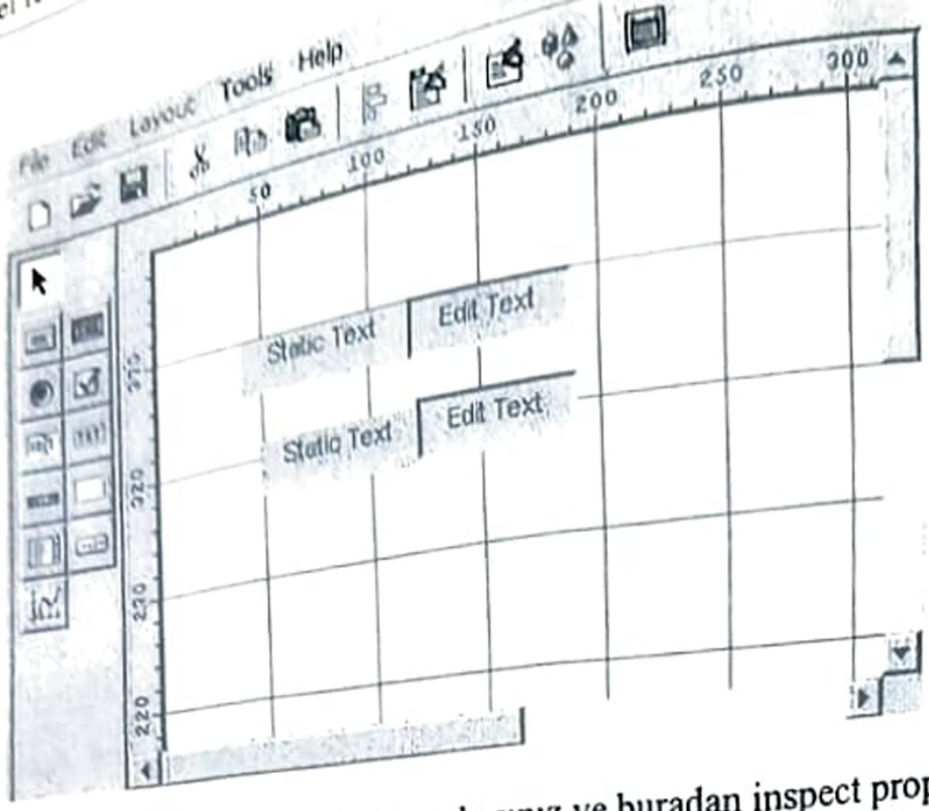
Sol tarafta görülen araçlar ile istenilen grafik ara birimi hazırlanır. Programı çalıştırmak için Tools'dan Activate Figure seçilir. Bu durumda bu birim ile aynı isimde bir .m uzantılı bir dosya Matlab tarafından oluşturulur. Bu dosya içine gerekli alt programlar yazılır.

Gui(kullanıcı ara birimi) ile ilgili açıklamalar örneklerle açıklanacaktır.

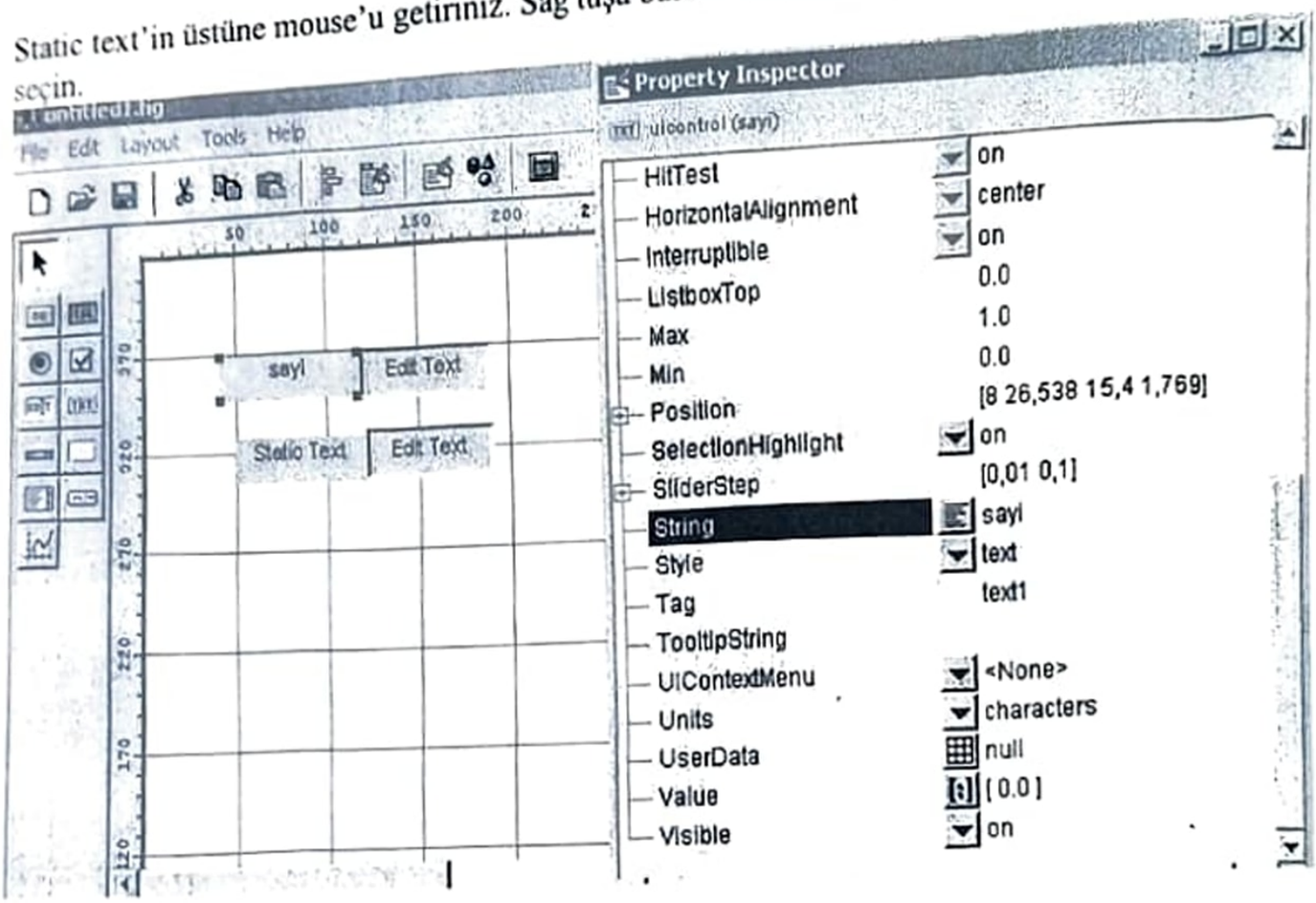
Örnek: Girilen bir sayının karesininin gösterilmesi.

guide komutu ile kullanıcı ara yüzünü hazırlamak için gerekli birimi çağırınız

sol kısımdan iki tane static text, iki tane de edit text'i bu birim üstüne taşıyınız.



Static text'in üstüne mouse'u getiriniz. Sağ tuşa basınız ve buradan inspect properties kısmını seçin.



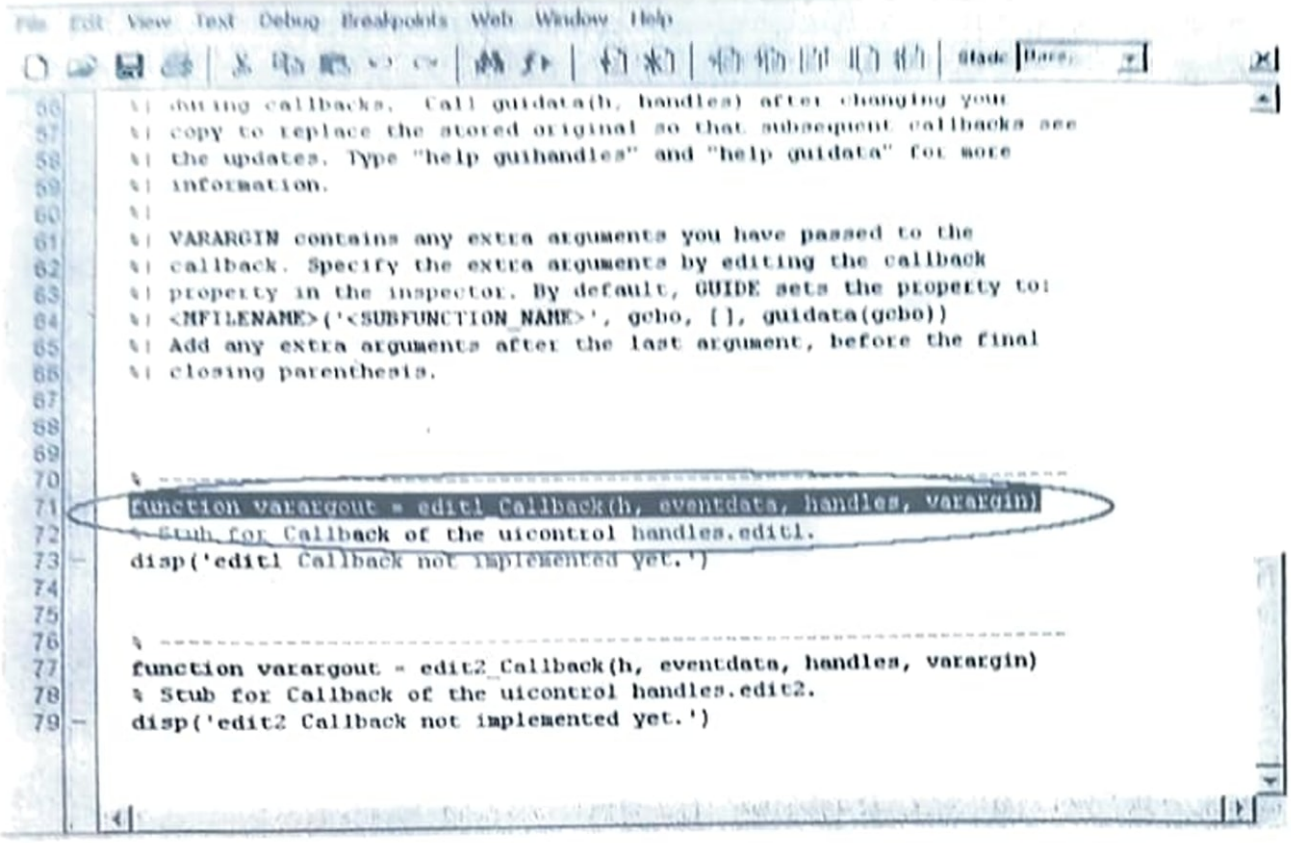
Karşınıza çıkan menüden string kısmını seçip oraya sayı yazınız.

Aynı şekilde diğer static text'i karesi olarak değiştiriniz.
Edit text'in ikisine de aynı şekilde değiştirerek boş metin haline getiriniz.

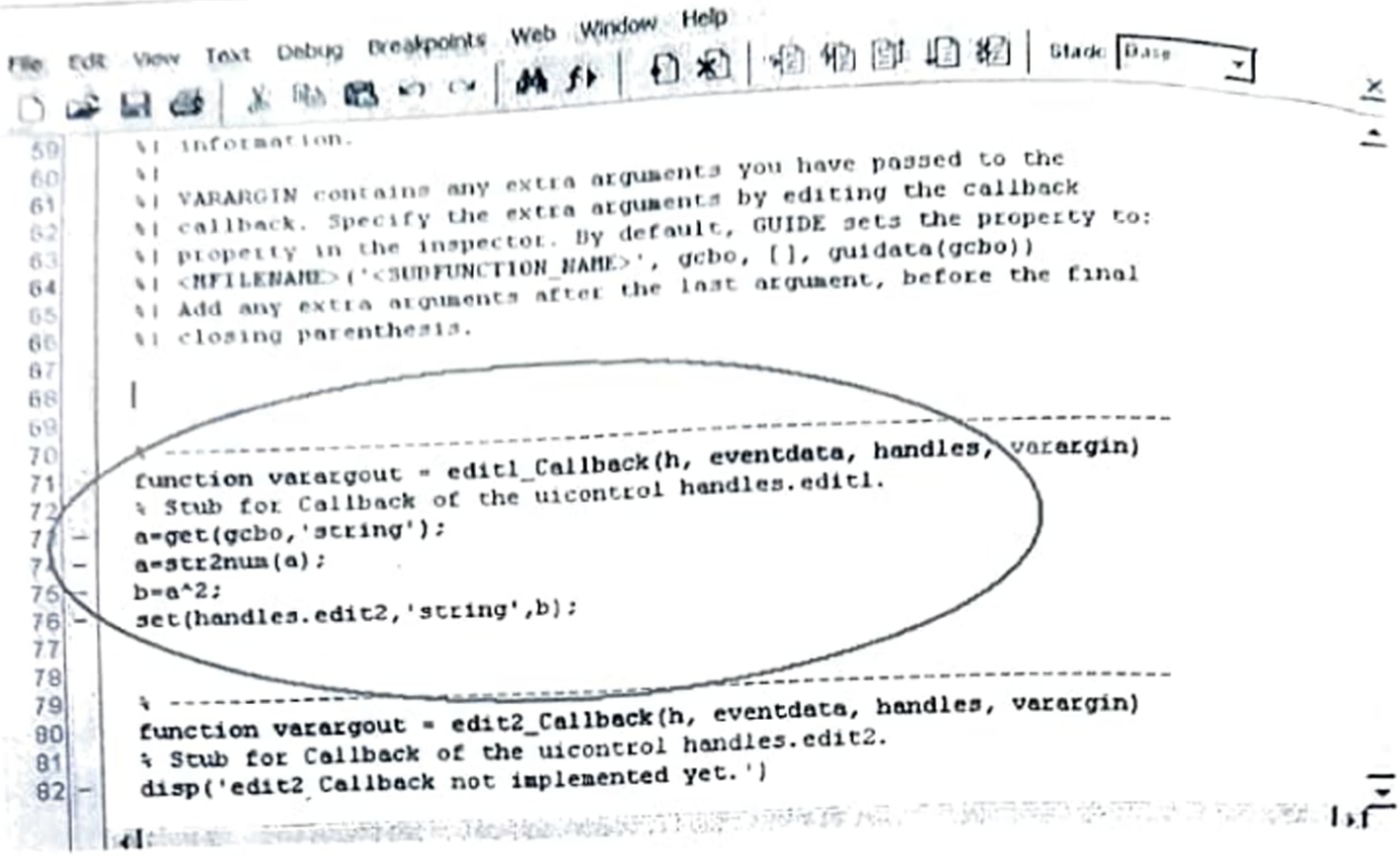
Bu programda amacımız girdiğimiz bir sayının karesini görmektir.

1. edit text'in üstüne mouse'u getirip sağ tuşa basın ve Edit Callback'i seçin. Bunu yapınca Matlab figure'ü kaydetmek isteyecek ve bir isim soracaktır. Programa isim olarak karealma

verin. Bu durumda Matlab karealma.m isimle bir Matlab dosyası oluşturacaktır. Şu anda edit text ile ilgili alt program yazılabilir. karealma.m isimli dosyada bu bölüm otomatik olarak gösterilmiştir.



Bu kısıma yazılacak program bu edittext kutusu üzerinde yapılan işlemler sırasında devreye girecektir. Birinci kutuya girilen bir sayının karesini alan program aşağıda gösterildiği gibi ilgili bölüme yazılır.



```
%  information.
%
%  VARARGIN contains any extra arguments you have passed to the
%  callback.  Specify the extra arguments by editing the callback
%  property in the inspector.  By default, GUIDE sets the property to:
%  <MFILENAME>('<SUBFUNCTION_NAME>', gcbo, [], guidata(gcbo))
%  Add any extra arguments after the last argument, before the final
%  closing parenthesis.

% --------------------------------------------------------------------
function varargout = edit1_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.edit1.
a=get(gcbo,'string');
a=str2num(a);
b=a^2;
set(handles.edit2,'string',b);

% --------------------------------------------------------------------
function varargout = edit2_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.edit2.
disp('edit2_Callback not implemented yet.')
```

Yukarıda programın açıklaması su şekildedir.

a=get(gcbo,'string');
%Kutu içindeki değeri oku ve a'ya ata.
a=str2num(a);
% a bir string olduğu için sayısal değere çevir.
b=a^2;
% a'nın karesini al ve b'ye ata.
set(handles.edit2,'string',b);
% ikinci kutuya b'nin karesini yaz..

Örneğin a'ya 12 yazılır ve enter tuşuna basılırsa b'de 144 gözükecektir.

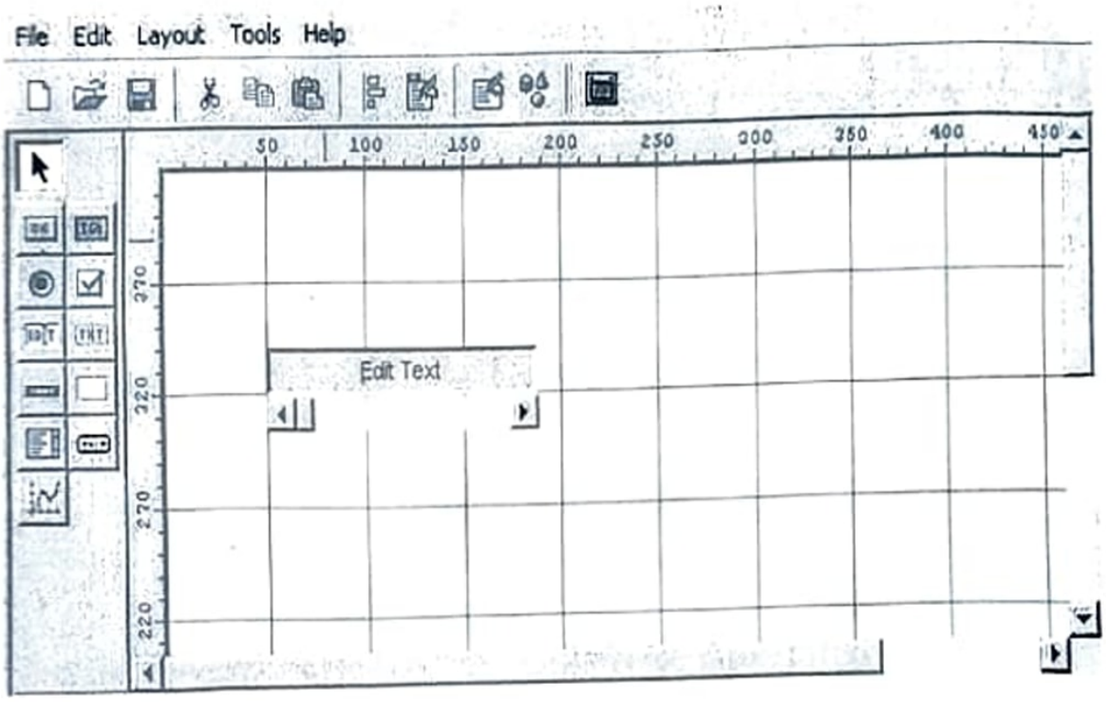


Örnek:

Kayan çubuğun hareket ettirilmesi sonucu oluşan değerlerinin bir yazı kutusu içinde okunması.

guide yazarak gerekli birimi açınız.

Bu birime slider ve edit text ekleyiniz.



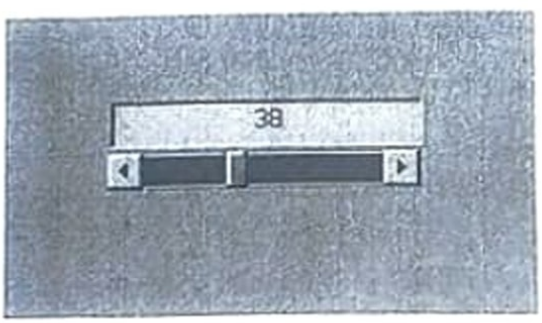
Edit text kutusunun içini boşaltınız. Slider'ın maksimum değerine 100, minimum değerine 0 atayın.
Edit text için background color' pembe, slider için mavi ve figure için kahverengi seçiniz.
Slider'in üstüne mouse'u getirin ve sağ tuşa basarak ve Edit Callback'i seçin. Bu durmda ".m" dosyasının aşağıda gösterilen kısmı gelir. Buraya yazacağız program kayan çubuk üzerine işlem yapıldığında aktif hale geçer. Buraya şu program yazılır.

```
deger = get(handles.slider1,'Value');
% Bu komut ile slider'ın o anki değeri okunur.
set(handles.edit1,'string',deger);
% Bu komut ile de slider'ın değeri edit1 kutusunda yer alan boşlukta gösterilir.
```

Bunun sonucunda slider ileri ve geri olarak hareket ettirildikçe edit1 kutusu içindeki sayı slider'a paralel olarak değişecektir.

```
% during callbacks.  Call guidata(h, handles) after changing your
% copy to replace the stored original so that subsequent callbacks see
% the updates. Type "help guihandles" and "help guidata" for more
% information.
%
% VARARGIN contains any extra arguments you have passed to the
% callback. Specify the extra arguments by editing the callback
% property in the inspector. By default, GUIDE sets the property to:
% <MFILENAME>('<SUBFUNCTION_NAME>', gcbo, [], guidata(gcbo))
% Add any extra arguments after the last argument, before the final
% closing parenthesis.



% --------------------------------------------------------------------
function varargout = edit1_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.edit1.


% --------------------------------------------------------------------
function varargout = slider1_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.slider1.
deger = get(handles.slider1,'Value');
set(handles.edit1,'string',deger);
```

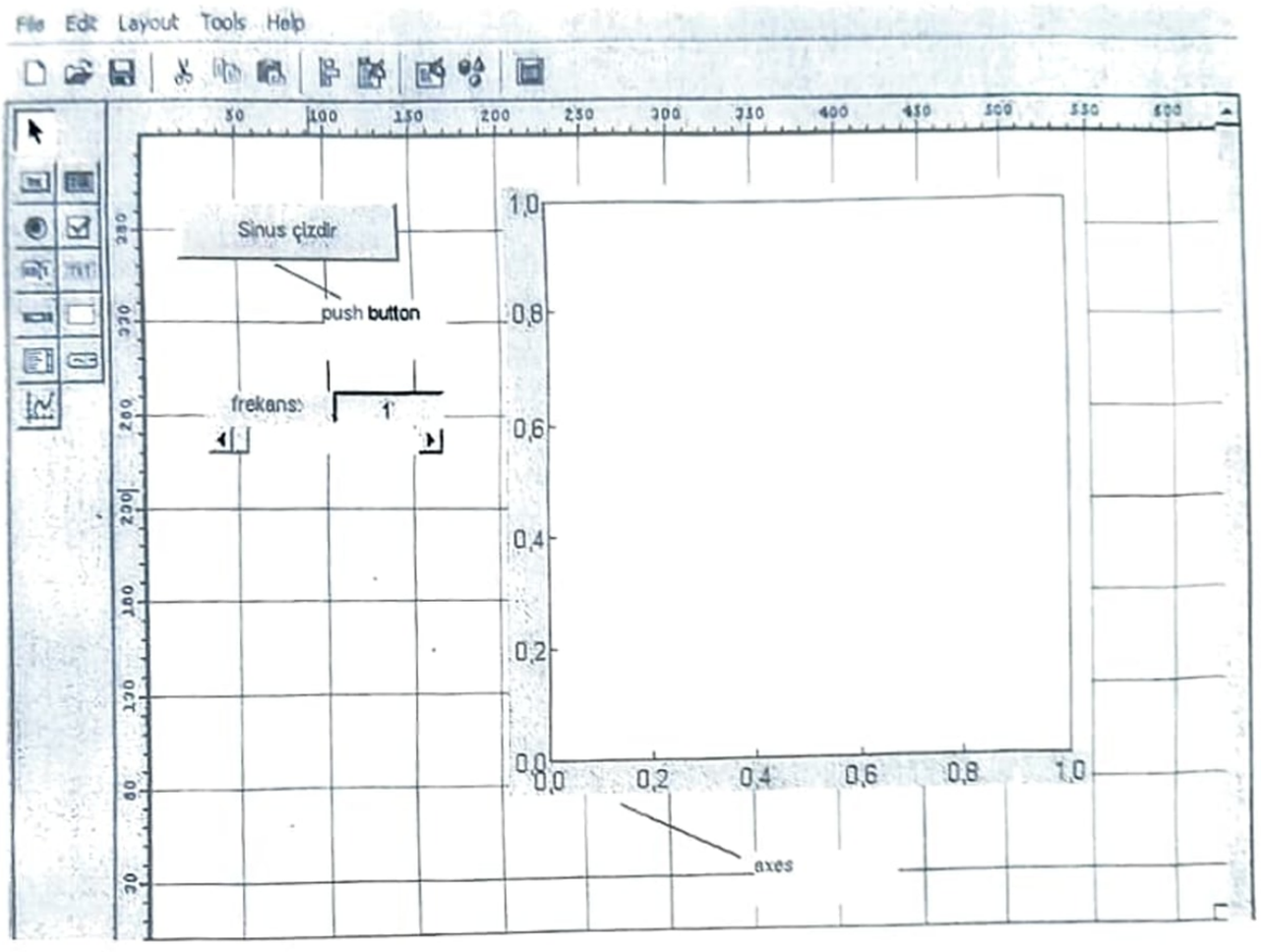
Sonuç olarak aşağıdaki görünümü elde ederiz.



Örnek:
Bir sinüs sinyalinin frekansının değiştirilmesi ve bu değişikliğin grafiksel olarak incelenmesi.

guide yazara gerekli birimi açınız.

Aşağıda gösterildiği gibi gerekli araçları ara yüze taşıyınız.



Sinus çizdir push button'ı için şu komutları yazınız:

```
f=1;
t=0:.05:2;
y=sin(2*pi*f*t);
plot(t,y)
```

Bu tuşa basıldığı durumda sinüs gösterilen grafik üzerine çizilecektir.

Slider için şu komutları yazınız:

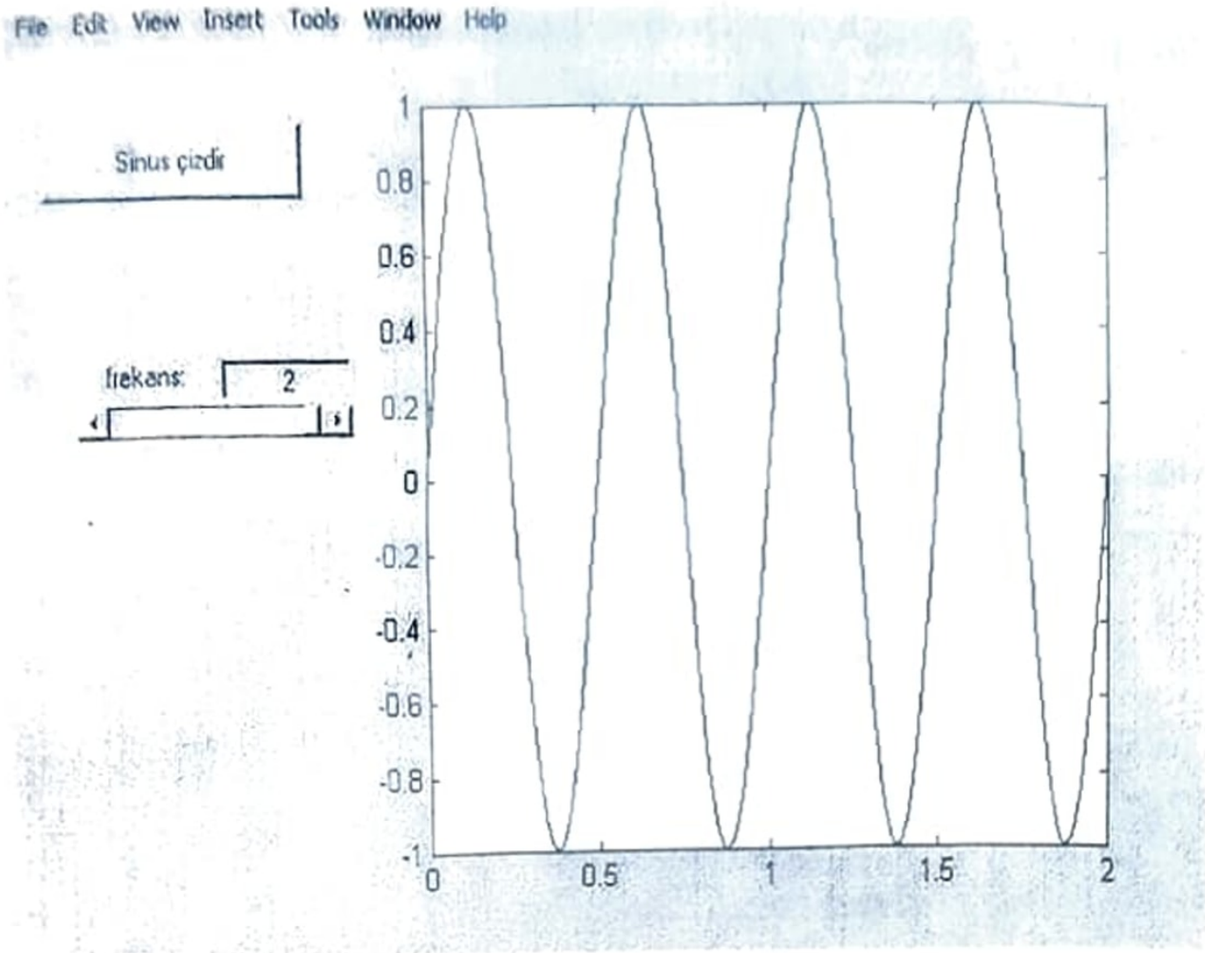
```
f = get(handles.slider2,'Value');
%slider'ın değerini oku
set(handles.edit3,'string',f);
% edit kutusu içine slider'in değerini yaz.
t=0:.001:2;
y=sin(2*pi*f*t);
plot(t,y)
% yeni frekans değeri ile sinüs grafiğini çiz.
axis([0 2 -1 1]);
```

Bu komutların tümünün yazıldığı ".m" uzantılı matlab dosyası aşağıda gösterilmiştir.

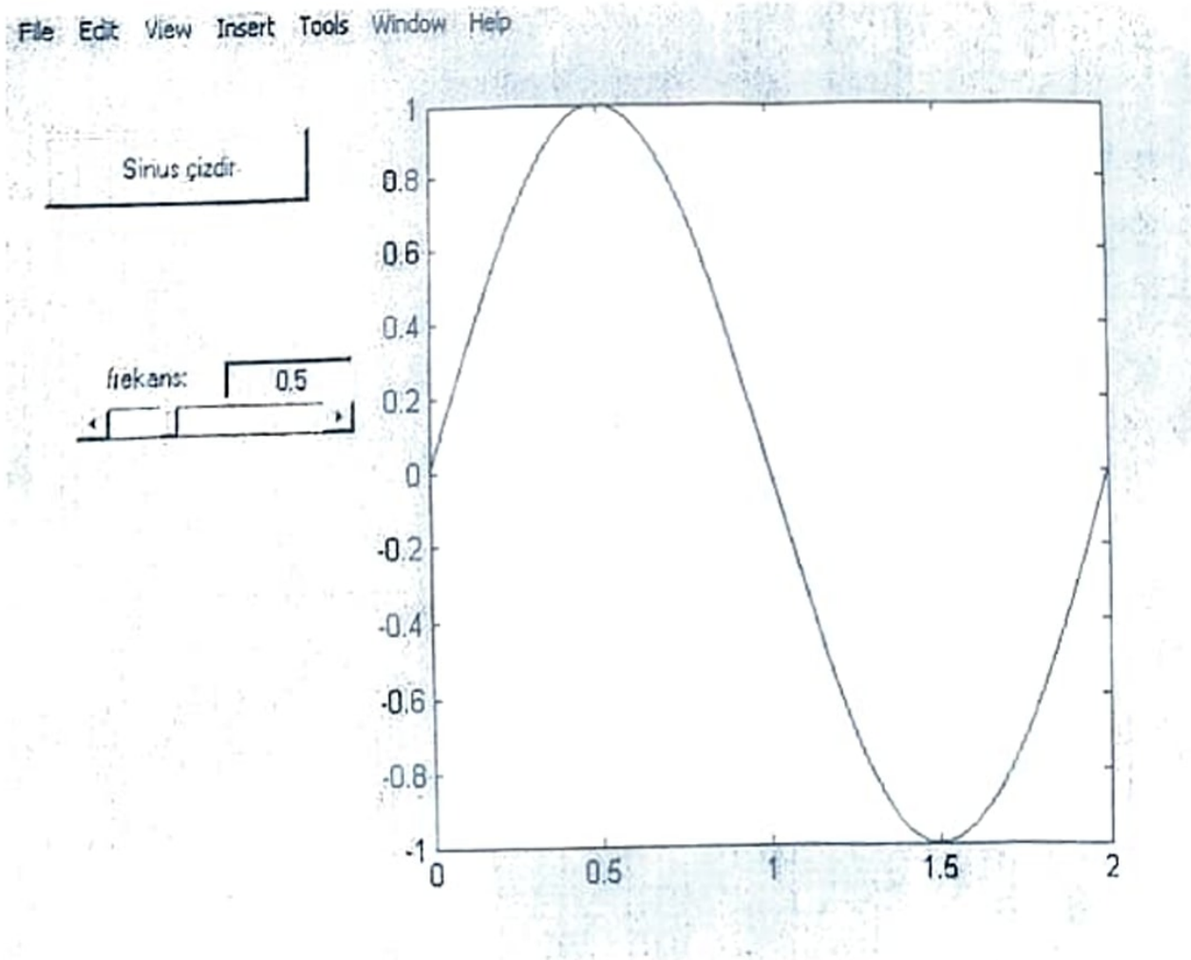
```
%  information.
%
%  VARARGIN contains any extra arguments you have passed to the
%  callback.  Specify the extra arguments by editing the callback
%  property in the inspector.  By default, GUIDE sets the property to:
%  <MFILENAME>('<SUBFUNCTION_NAME>', gcbo, [], guidata(gcbo))
%  Add any extra arguments after the last argument, before the final
%  closing parenthesis.


% --------------------------------------------------------------------
function varargout = pushbutton2_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton2.
f=1;
t=0:.05:2;
y=sin(2*pi*f*t);
plot(t,y)


% --------------------------------------------------------------------
function varargout = edit3_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.edit3.



% --------------------------------------------------------------------
function varargout = slider2_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.slider2.
f = get(handles.slider2,'Value');
set(handles.edit3,'string',f);
t=0:.001:2;
y=sin(2*pi*f*t);
plot(t,y)
axis([0 2 -1 1]);
```

programı çalıştırınca çıkan ortaya çıkan grafikte grafik çizdir düğmesine basılınca sinüs çizdirilir. Slider sağa sola kaydırılarak frekans artar ve azalır. Frekans değişimi hem değer olarak gözükür, hem de sinüsteki değişimden anlaşılır.

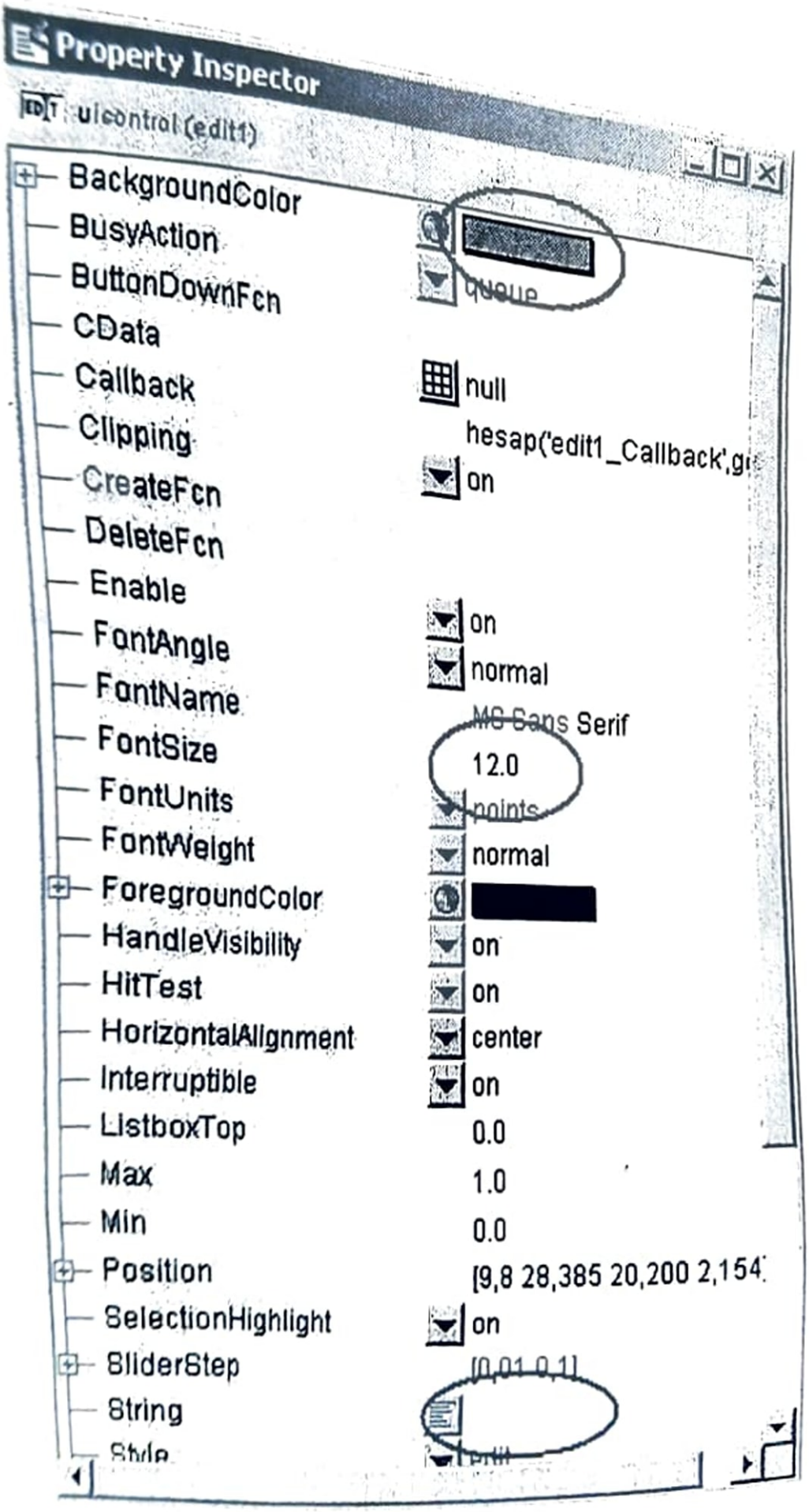
Frekansın 2 olduğu zamanki grafik aşağıda gösterilmiştir.



Aşağıdaki grafikte ise frekansın 0.5 olduğu zamanki durum gösterilmiştir.



yapınız:



un altına bir static text ekleyiniz ve Inspect properties'den şu seçimleri yapınız.

İkinci bir edit text ekleyiniz ve Inspect properties'den şu seçimleri yapınız.

Background color: Açık kırmızı
Fontsize:12
String: Boşluk

Bu edit text'in altına bir static text ekleyiniz ve Inspect properties'den şu seçimleri yapınız.

Background color: Açık sarı
Fontsize:10
String: Sayi 2

Bir tane popup menu ekleyiniz ve Inspect properties'den şu seçimleri yapınız.
Background color: Mor
Fontsize:10
String:

Seçiminizi yapın
Toplama
Çıkarma
Çarpma
Bölme

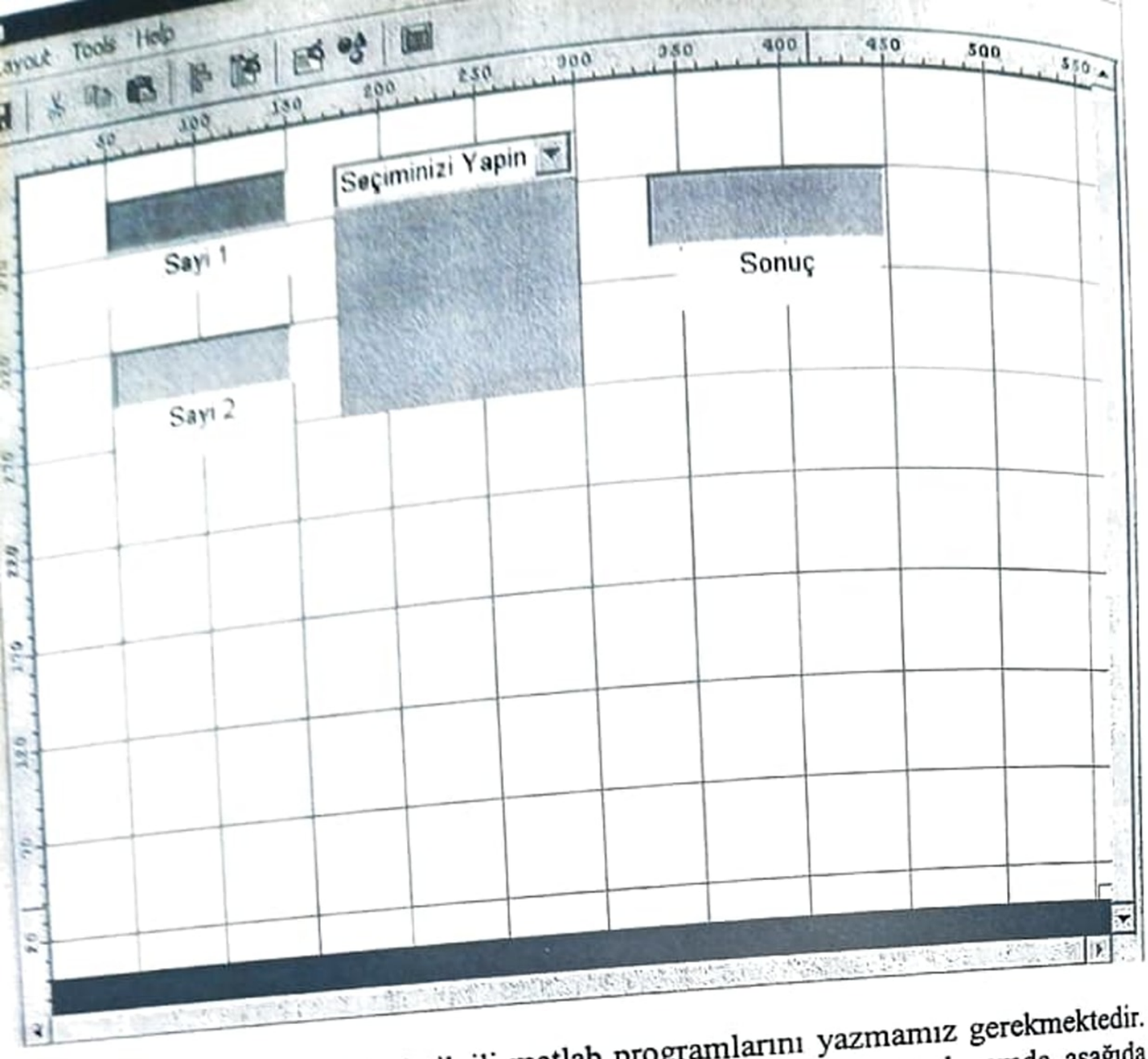
Sonuçları yazdırmak için üçüncü bir edit text ekleyiniz ve Inspect properties'den şu seçimleri yapınız.

Background color: Turuncu
Fontsize:14
String: Boşluk

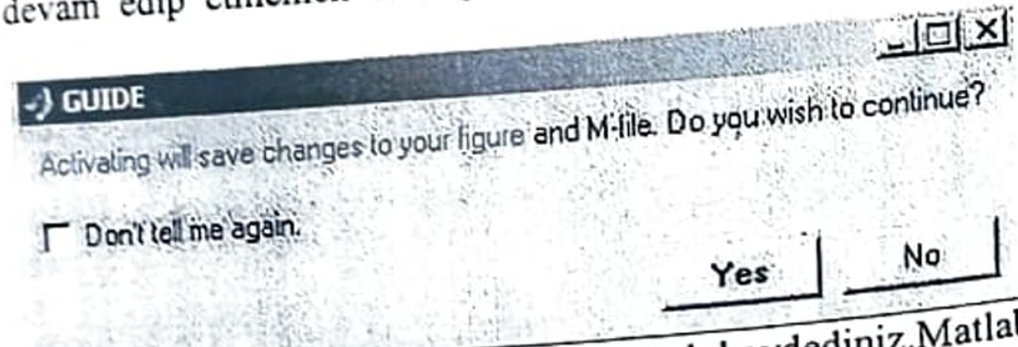
Bu edit text'in altına bir static text ekleyiniz ve Inspect properties'den şu seçimleri yapınız.

Background color: Açık sarı
Fontsize:10
String: Sonuç

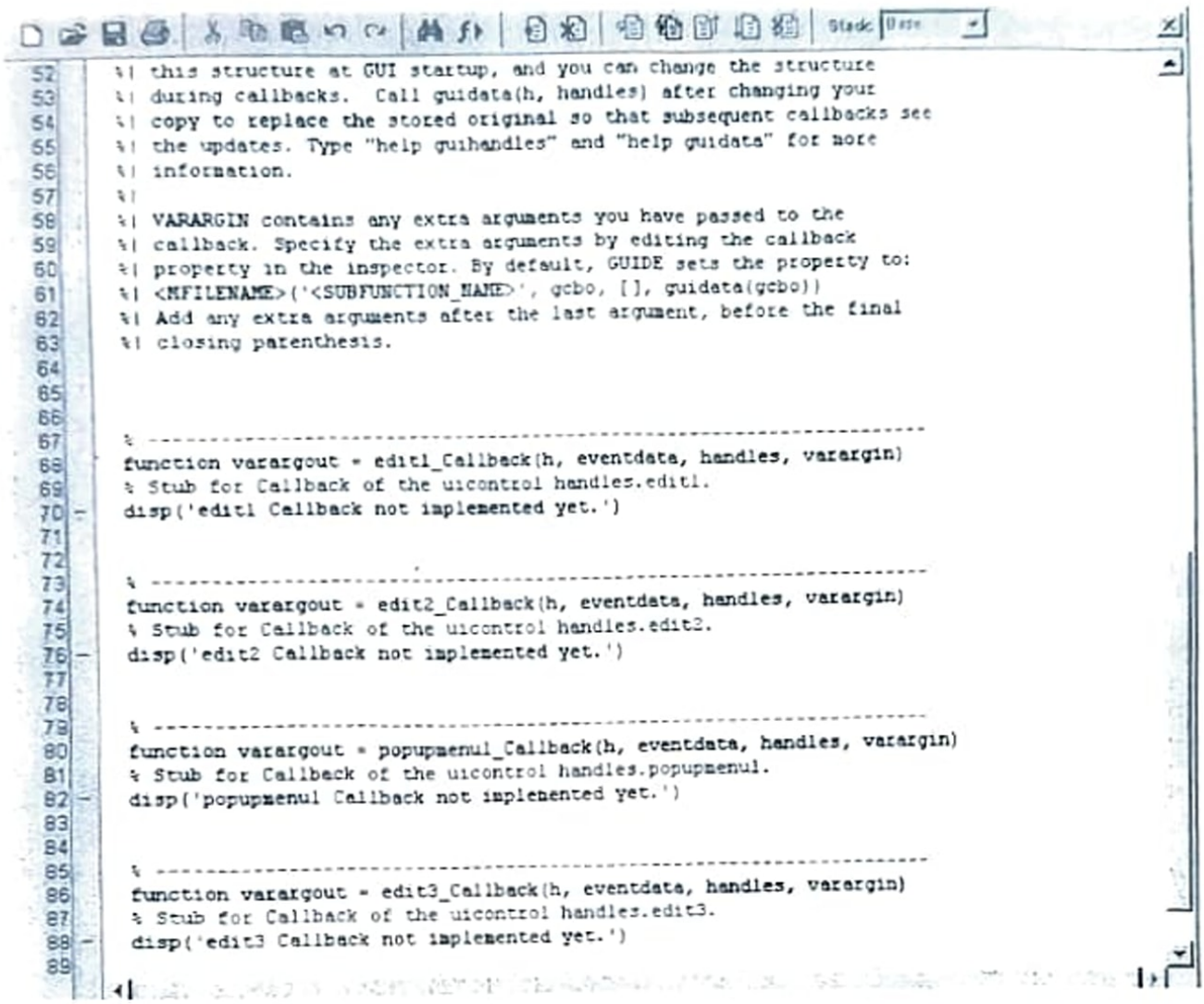
Şu anda ara yüzün taslağı bitmiştir. Ara yüzün bu hali aşağıda gösterilmiştir.



lan sonra kullanacağımız araçlarla ilgili matlab programlarını yazmamız gerekmektedir. ın için Tools menüsünden activate figure seçeneğine tıklayınız. Bu durumda aşağıda ken uyarı ekrana gelecek, değişikliklerin hem şekile, hem de M-file'a kaydedileceğini eyecek ve devam edip etmemek istediğimizi soracaktır. Yes seçeneğini seçip devam ız.

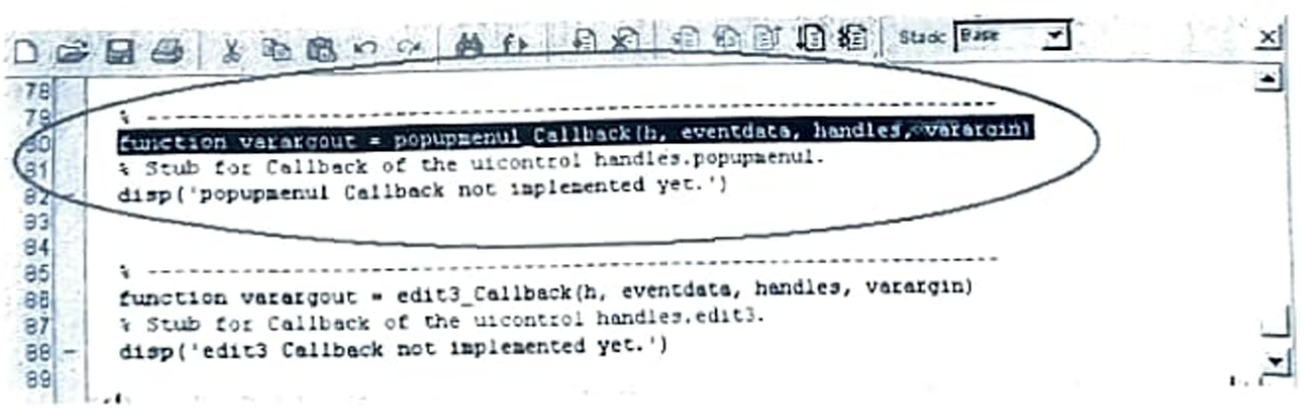


kli kaydetmenizi isteyecektir. Şekle hesap ismini vererek kaydediniz.Matlab hem şekli ydedecek, hem de hesap.m isimli bir dosya oluşturacaktır. Bu dosya aşağıda gösterilmiştir.



```
%| this structure at GUI startup, and you can change the structure
%| during callbacks.  Call guidata(h, handles) after changing your
%| copy to replace the stored original so that subsequent callbacks see
%| the updates. Type "help guihandles" and "help guidata" for more
%| information.
%|
%| VARARGIN contains any extra arguments you have passed to the
%| callback. Specify the extra arguments by editing the callback
%| property in the inspector. By default, GUIDE sets the property to:
%| <MFILENAME>('<SUBFUNCTION_NAME>', gcbo, [], guidata(gcbo))
%| Add any extra arguments after the last argument, before the final
%| closing parenthesis.



% --------------------------------------------------------------------
function varargout = edit1_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.edit1.
disp('edit1 Callback not implemented yet.')


% --------------------------------------------------------------------
function varargout = edit2_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.edit2.
disp('edit2 Callback not implemented yet.')


% --------------------------------------------------------------------
function varargout = popupmenu1_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.popupmenu1.
disp('popupmenu1 Callback not implemented yet.')


% --------------------------------------------------------------------
function varargout = edit3_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.edit3.
disp('edit3 Callback not implemented yet.')
```

taslak ekranına gidip popupmenu üstüne gidiniz ve mouse'un sağ tuşuna basıp Editcallback'e tıklayınız. Bu durumda karşımıza hesap.m dosyası gelecek ve ilgili bölümü işaretliyecektir. Bu durum aşağıda gösterilmiştir.



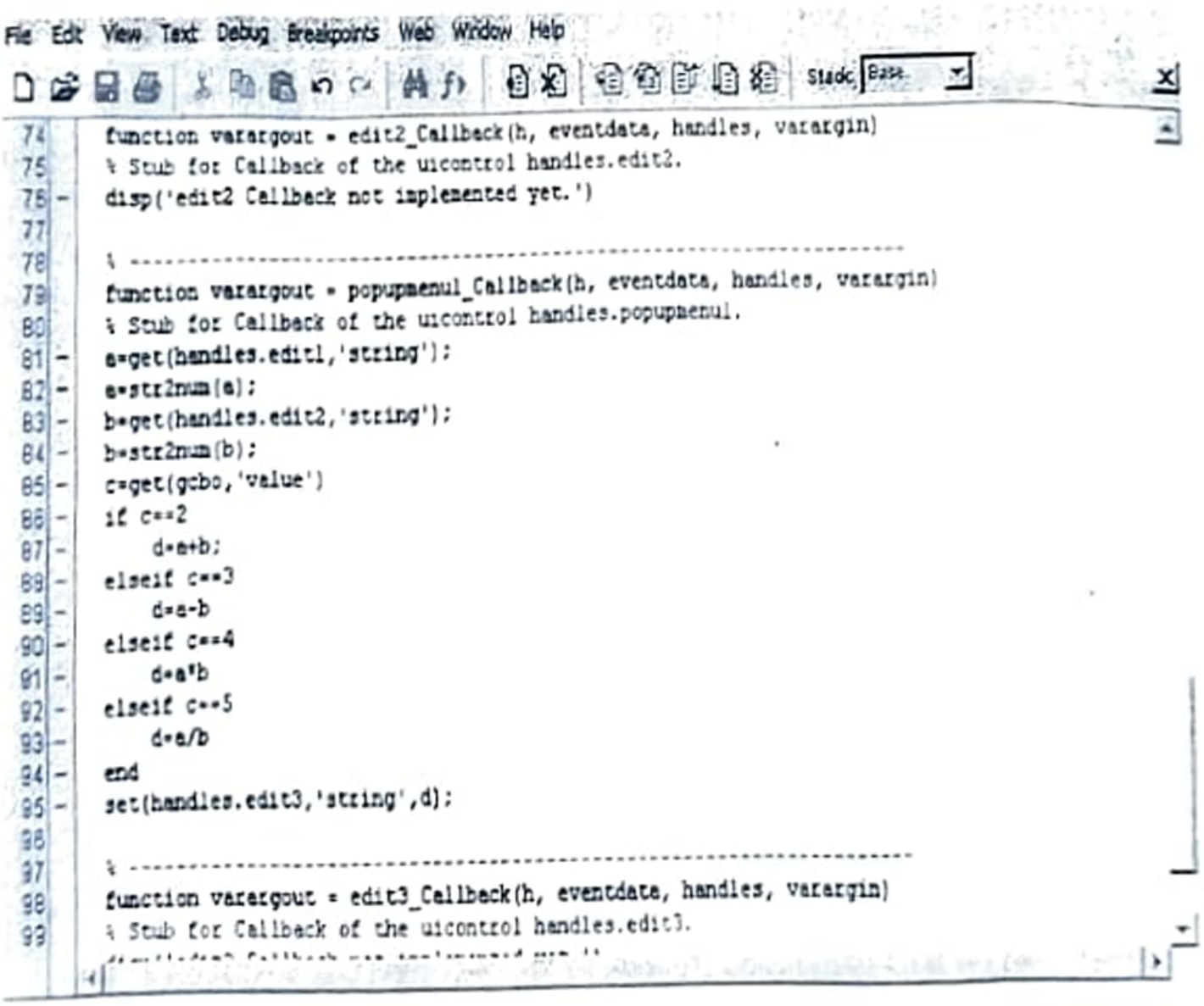
```
% --------------------------------------------------------------------
function varargout = popupmenu1_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.popupmenu1.
disp('popupmenu1 Callback not implemented yet.')


% --------------------------------------------------------------------
function varargout = edit3_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.edit3.
disp('edit3 Callback not implemented yet.')
```

Yukarıda gösterilen bölüme aşağıdaki programı yazınız ve disp(' popmenu1 Callback not implemented yet.') komutunu siliniz.

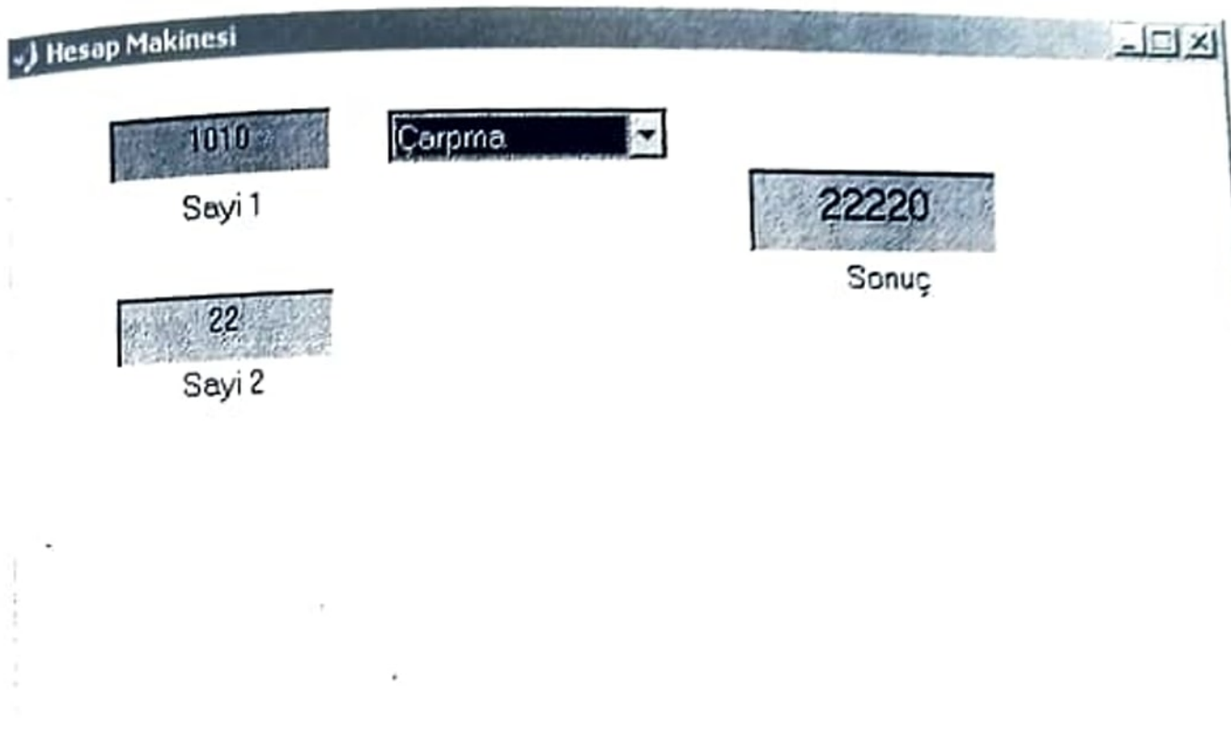
```
a=get(handles.edit1,'string');
```

```
%edit1 kutusu içindeki bilgiyi okuyun
a=str2num(a);
% bu bilgiyi sayıya çevirin
b=get(handles.edit2,'string');
% edit2 kutusu içindeki bilgiyi okuyun
b=str2num(b);
% Bu bilgiyi sayıya çevirin.
c=get(gcbo,'value')
% popup menünün değerini okuyun
if c==2
   d=a+b;
% Bu değer iki ise toplama işlemi seçilmiştir.
elseif c==3
   d=a-b
% Bu değer 3 ise çıkarma işlemi seçilmiştir.
elseif c==4
   d=a*b
% Bu değer 4 ise çarpma işlemi seçilmiştir.
elseif c==5
   d=a/b
Bu değer 5 ise bölme işlemi seçilmiştir.
end
set(handles.edit3,'string',d);
```

programı ilgili yere yazdığımızda aşağıdaki dosya elde edilir.

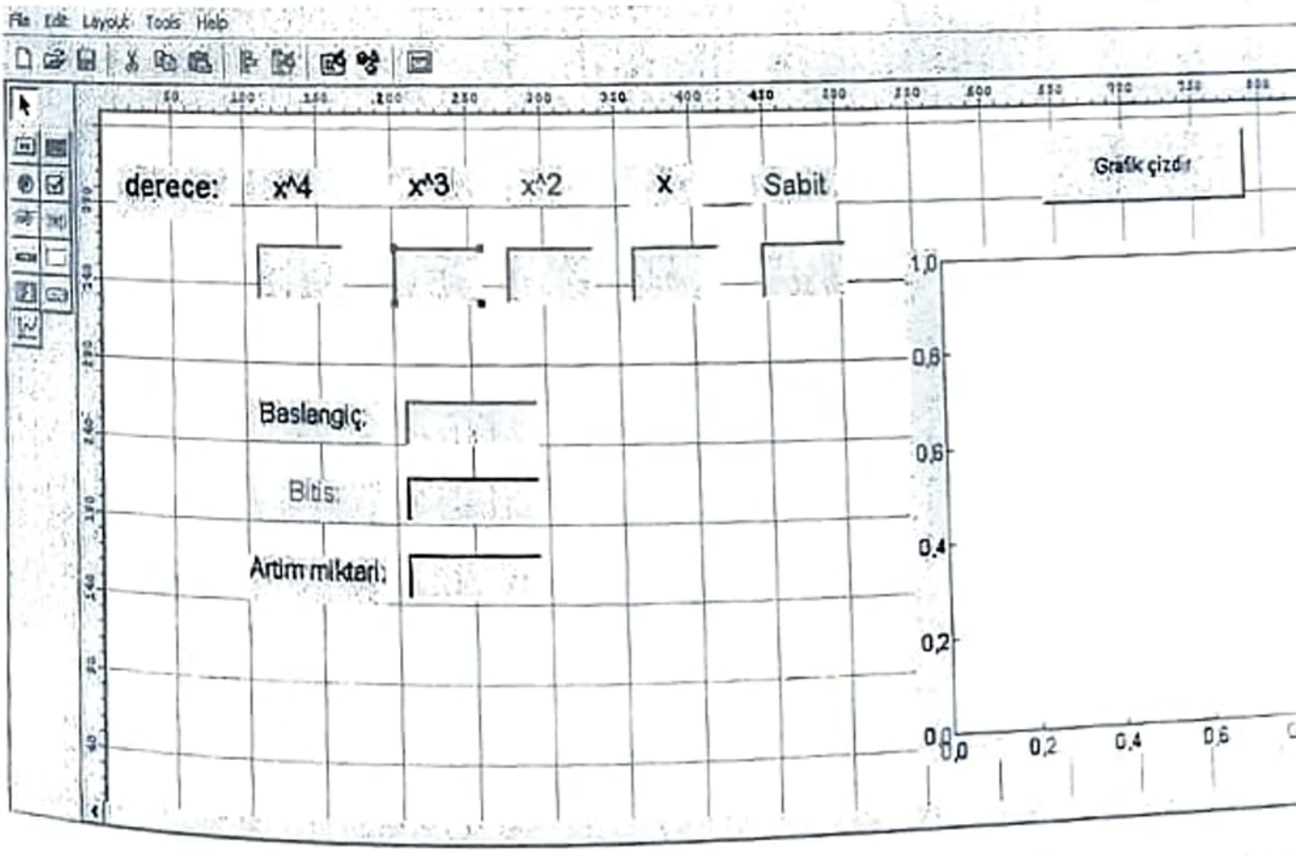


Bu durumda program bitmiştir ve çalıştırıldığında aşağıdaki arayüz elde edilir.



Örnek: Verilen katsayılardan fonksiyonun elde edilmesi ve çizilmesi.

Aşağıdaki taslağı hazırlayınız.

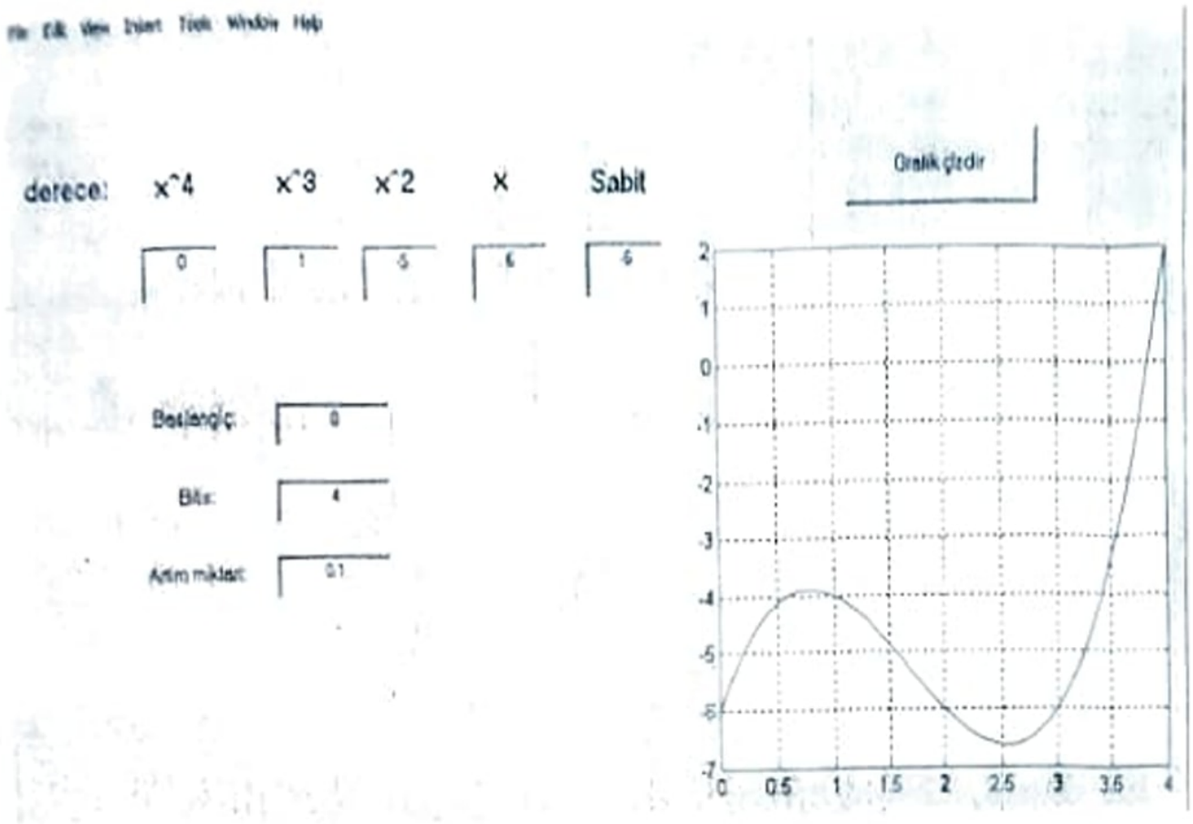


Bu programda bir fonksiyonun katsayıları yukarıda gösterilen kutulara girilir. Çiz
istenen grafiğin başlangıç, bitiş ve artım miktarı belirtilir. Grafik çizdir düğmesine bas
grafik çizdirilir. Bunun için grafik çizdir isimli push button için şu program yazılır.

```
a=get(handles.edit1,'string');
a=str2num(a);
% 4. derece için katsayıyı oku.
```

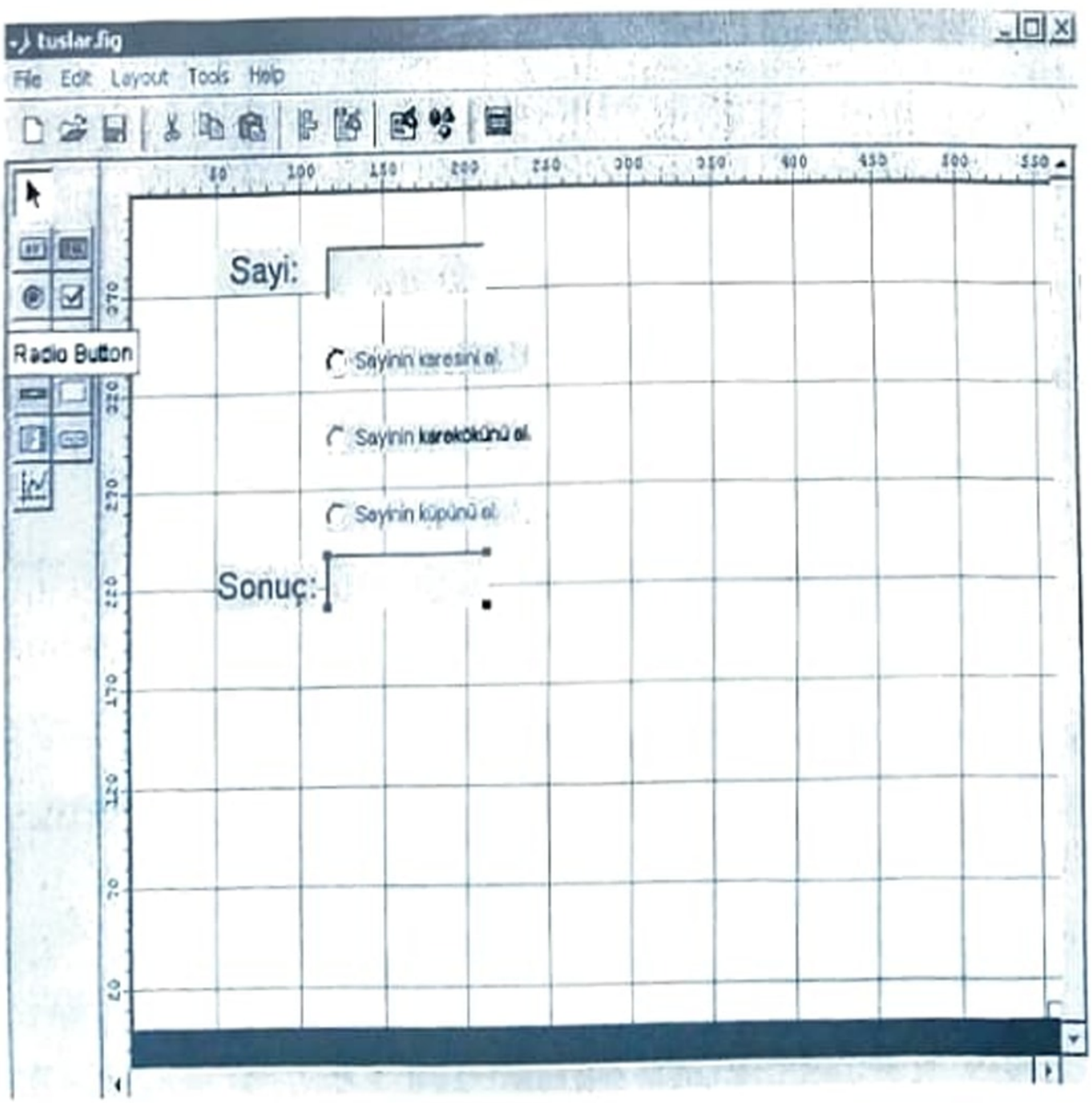
```
b=get(handles.edit2,'string');
b=str2num(b);
% 3. derece için katsayıyı oku.
c=get(handles.edit3,'string');
c=str2num(c);
% 2. derece için katsayıyı oku.
d=get(handles.edit4,'string');
d=str2num(d);
% 1. derece için katsayıyı oku.
e=get(handles.edit5,'string');
e=str2num(e);
% sabit katsayıyı oku.
bas=get(handles.edit6,'string');
bas=str2num(bas);
% grafik için başlangıç değerini oku.
bit=get(handles.edit7,'string');
bit=str2num(bit);
% grafik için bitiş değerini oku.
ara=get(handles.edit8,'string');
ara=str2num(ara);
% grafik için artış değerini oku.
t=bas:ara:bit;
z=[a b c d e]
% katsayıları alarak z'ye ata.
w=polyval(z,t);
% z fonksiyonunu t'ye göre hesaplatıp w'ye ata.
plot(t,w)
% grafiği çizdir.
grid
```

Aşağıda x^3-5x^2+6x-6 grafiğinin çizilmiş hali gösterilmiştir.



örnek: Bir sayının karesinin, karekökünün veya küpünün alınması.

İlk olarak aşağıdaki taslağı hazırlayınız

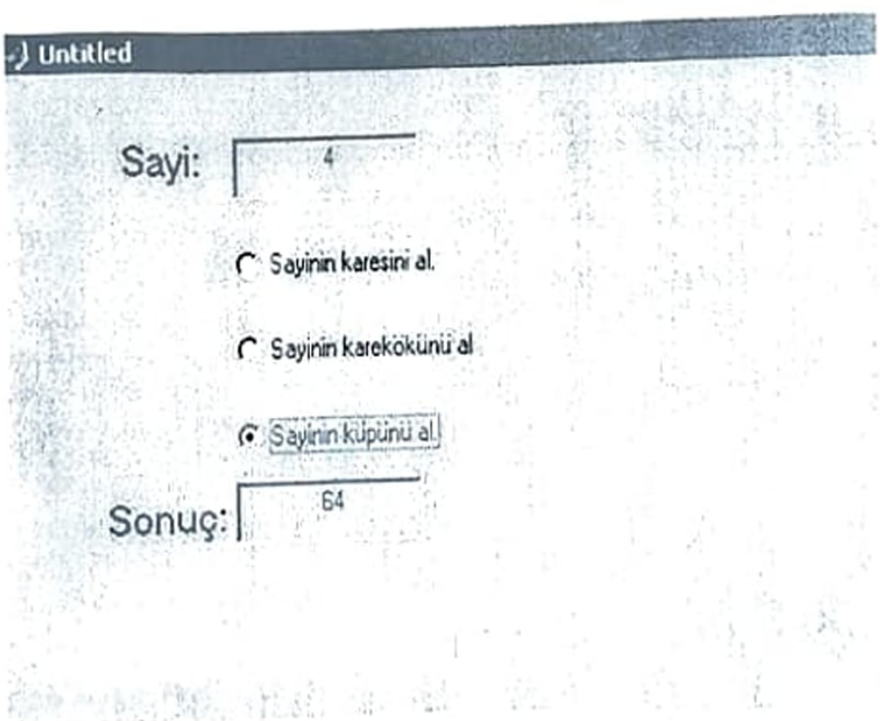


1. radio button üstüne basınca sayinin karesinin, 2. radio button üstüne basılınca sayının karekökünün alınması, 3. radio button üstüne basılınca sayının küpünün alınması isteniliyor. Bu yüzden her radio button için ayrı ayrı program yazılmalıdır. Yazılan programlar aşağıda açıklanmıştır.

```
function varargout = radiobutton1_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.radiobutton1.
a=get(gcbo,'value');
% radio button1 'in değerini oku
set(handles.radiobutton2,'value',0)
set(handles.radiobutton3,'value',0)
% Aynı anda diğer radio button'ların seçilmemesi için onların değerine sıfır atanır.
if a==1
% eğer radio button1 seçilmiş ise
b=get(handles.edit1,'string');
% edit1 kutusunun içeriğini oku.
b=str2num(b);
% Bu içeriğin sayıya çevir.
c=b^2
% Bu sayının karesini al.
set(handles.edit2,'string',c);
% sonuç kısmına karesi alınmış sayıyı yaz.
end
% --------------------------------------------------------------------
function varargout = radiobutton2_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.radiobutton2.
a=get(gcbo,'value');
% radio button2'in değerini oku
set(handles.radiobutton1,'value',0)
set(handles.radiobutton3,'value',0)
% Aynı anda diğer radio button'ların seçilmemesi için onların değerine sıfır atanır.
if a==1
% eğer radio button2 seçilmiş ise
b=get(handles.edit1,'string');
% edit1 kutusunun içeriğini oku.
b=str2num(b);
% Bu içeriği sayıya çevir.
c=sqrt(b);
% Bu sayının karekökünü al.
set(handles.edit2,'string',c);
% sonuç kısmına karekökü alınmış sayıyı yaz.
end
% --------------------------------------------------------------------
function varargout = radiobutton3_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.radiobutton3.
a=get(gcbo,'value');
% radio button3'ün değerini oku

set(handles.radiobutton1,'value',0)
```

```
set(handles.radiobutton2,'value',0)
% Aynı anda diğer radio button'ların seçilmemesi için onların değerine sıfır atanır.
if a==1
% eğer radio button3 seçilmiş ise

b=get(handles.edit1,'string');
% edit1 kutusunun içeriğini oku.
b=str2num(b);
% Bu içeriği sayıya çevir.
c=b^3;
% Bu sayının kübünü al.
set(handles.edit2,'string',c);
% sonuç kısmına kübü alınmış sayıyı yaz.
end
```

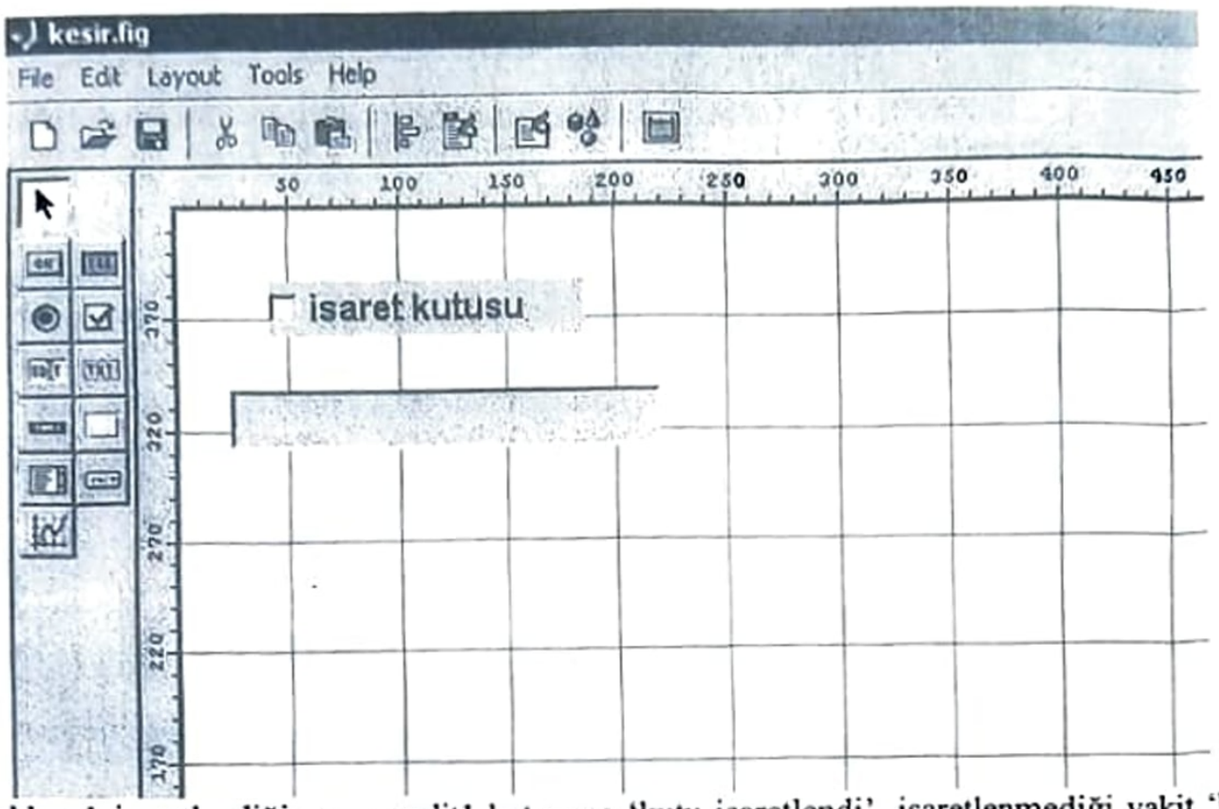
Aşağıda 4 sayısının küpünün alındığı grafik gösterilmiştir.



Örnek:

Bu programda bir işaretin konulup konulmadığı araştırılır. Eğer işaret konulmuş ise işaret konuldu, konulmamış ise işaret konulmadı yazılır.

İlkönce aşağıda görülen taslağı hazırlayınız.



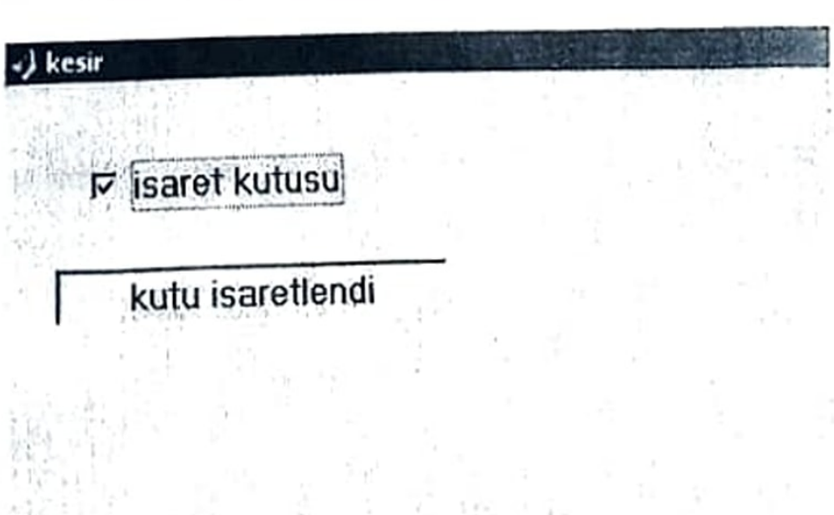
checkbox1 işaretlendiği zaman edit1 kutusuna 'kutu işaretlendi', işaretlenmediği vakit 'kutu işaretlenmedi' yazdırmak istiyoruz. Bu durumda chechbox1 için şu program yazılmalıdır.

```
a=get(gcbo,'value')
% checkbox1 kutusunun değerini oku. Eğer işaretlenmiş ise değeri 1, işaretlenmemiş ise
değeri 0'dır.
if a==1
    set(handles.edit1,'string','kutu isaretlendi')
% Eğer a'nın değeri 1 ise kutu işaretlenmiştir. Bu durumda edit1 kutusu içine 'kutu
işaretlendi' yazdırınız.
elseif a==0
    set(handles.edit1,'string','kutu isaretlenmedi')
% Eğer a'nın değeri 0 ise kutu işaretlenmemiştir. Bu durumda edit1 kutusu içine 'kutu
işaretlenmedi' yazdırınız.
  End
```

Programı yazdırınca aşağıdaki görüntü elde edilir.

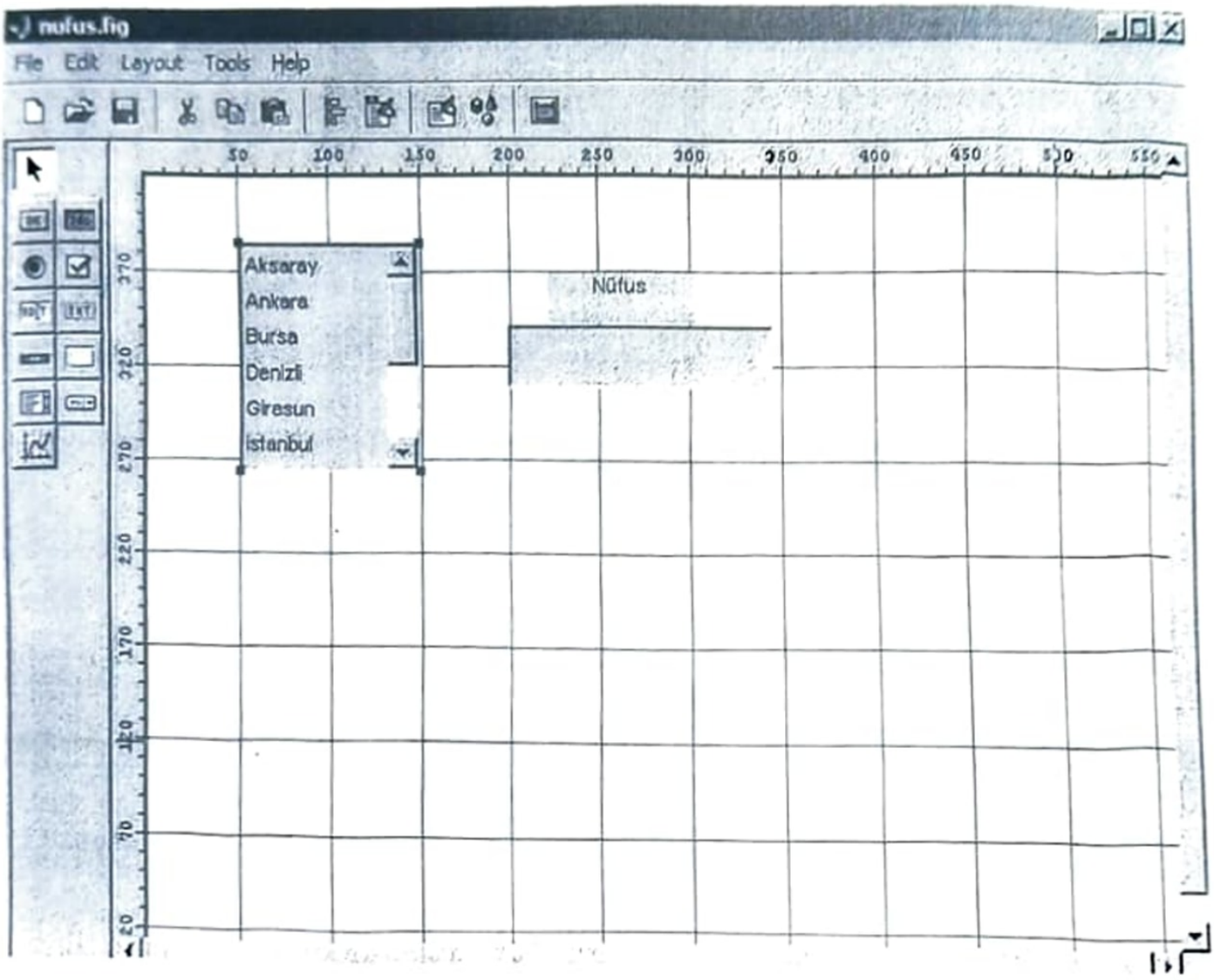
```
C:\matlabR12\work\kesir.m
File  Edit  View  Text  Debug  Breakpoints  Web  Window  Help
Stack: Base
59  %| information.
60  %|
61  %| VARARGIN contains any extra arguments you have passed to the
62  %| callback. Specify the extra arguments by editing the callback
63  %| property in the inspector. By default, GUIDE sets the property to:
64  %| <MFILENAME>('<SUBFUNCTION_NAME>', gcbo, [], guidata(gcbo))
65  %| Add any extra arguments after the last argument, before the final
66  %| closing parenthesis.
67
68
69
70  % --------------------------------------------------------------------
71  function varargout = checkbox1_Callback(h, eventdata, handles, varargin)
72  % Stub for Callback of the uicontrol handles.checkbox1.
73
74  a=get(gcbo,'value')
75  if a==1
76        set(handles.edit1,'string','kutu isaretlendi')
77  elseif a==0
78        set(handles.edit1,'string','kutu isaretlenmedi')
79     end
80
81  % --------------------------------------------------------------------
82  function varargout = edit1_Callback(h, eventdata, handles, varargin)
83  % Stub for Callback of the uicontrol handles.edit1.
84
Ready
```

programın çalıştırılmış hali aşağıda görüntülenmiştir.



Örnek: Bir listeden şehir isimlerinin seçilmesi ve bu şehirlerin nüfuslarının yazdırılması.

İlk olarak aşağıda gözüken taslağı hazırlayınız.

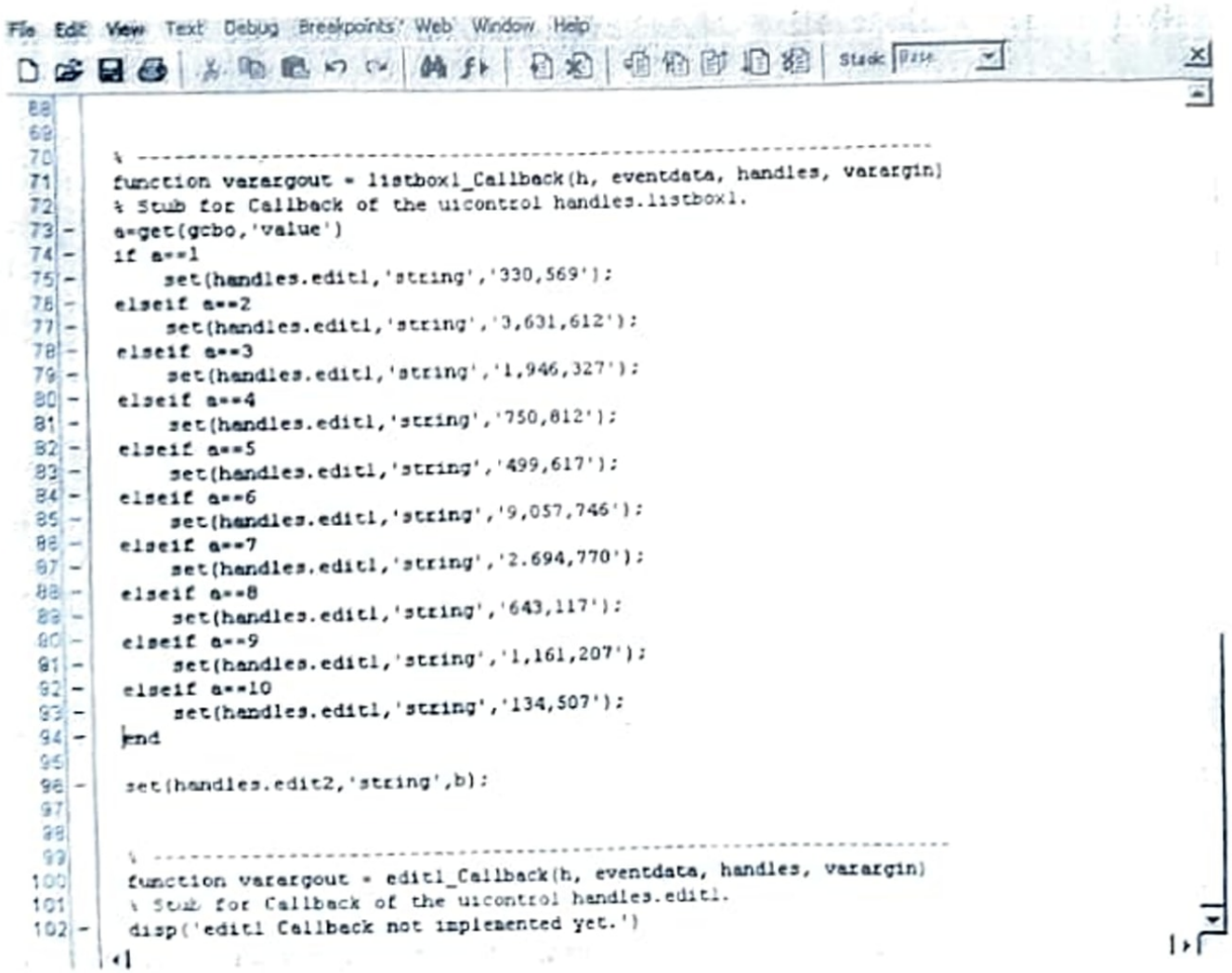


Şehir isimleri değiştikçe karşılık gelen nüfusların gözükmesi isteniyor. Bu yüzden list box1 altına şu program yazılmalıdır.

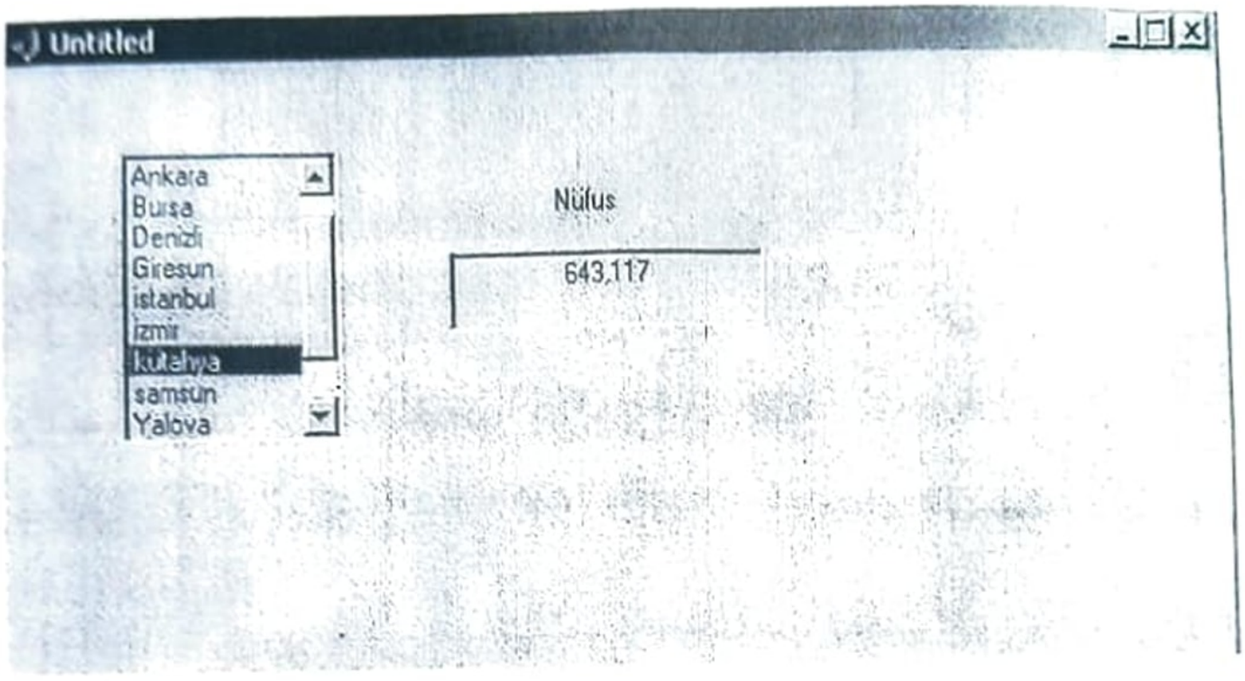
```
a=get(gcbo,'value')
listbox1'in değerini oku.
if a==1
   set(handles.edit1,'string','330,569');
% Eğer değer 1 ise Aksaray seçilmiştir. Bu şehrin nüfusunu edit1'in içine yazdır.
elseif a==2
   set(handles.edit1,'string','3,631,612');
% Eğer değer 2 ise Ankara seçilmiştir. Bu şehrin nüfusunu edit1'in içine yazdır.
elseif a==3
   set(handles.edit1,'string','1,946,327');
% Eğer değer 3 ise Bursa seçilmiştir. Bu şehrin nüfusunu edit1'in içine yazdır.
elseif a==4
   set(handles.edit1,'string','750,812');
% Eğer değer 4 ise Denizli seçilmiştir. Bu şehrin nüfusunu edit1'in içine yazdır.
elseif a==5
   set(handles.edit1,'string','499,617');
% Eğer değer 5 ise Giresun seçilmiştir. Bu şehrin nüfusunu edit1'in içine yazdır.
elseif a==6
   set(handles.edit1,'string','9,057,746');
% Eğer değer 6 ise İstanbul seçilmiştir. Bu şehrin nüfusunu edit1'in içine yazdır.
```

```
elseif a==7
   set(handles.edit1,'string','2.694,770');
% Eğer değer 7 ise İzmir seçilmiştir. Bu şehrin nüfusunu edit1'in içine yazdır.
elseif a==8
   set(handles.edit1,'string','643,117');
% Eğer değer 8 ise Kütahya seçilmiştir. Bu şehrin nüfusunu edit1'in içine yazdır.
elseif a==9
   set(handles.edit1,'string','1,161,207');
% Eğer değer 9 ise Samsun seçilmiştir. Bu şehrin nüfusunu edit1'in içine yazdır.
elseif a==10
   set(handles.edit1,'string','134,507');
% Eğer değer 10 ise Yalova seçilmiştir. Bu şehrin nüfusunu edit1'in içine yazdır.
End
```

Programın yazılmış hali aşağıda gösterilmiştir.



Program çalıştırıldığı zaman aşağıdaki grafiksel kullanıcı ara yüzü elde edilir.


Untitled
Ankara
Bursa
Denizli
Giresun
istanbul
izmir
kutahya
samsun
Yalova
Nüfus
643,117

SAYISAL İŞARET İŞLEME

Bazı Önemli Fonksiyonlar

→ Dürtü(Birim impuls) fonksiyonu:

Bu fonksiyon şu şekilde tanımlanır:

$$\delta(n)=\begin{cases}1 & n=0\\ 0 & n\neq 0\end{cases}$$

Fonksiyonun "a" kadar ötelenmiş hali şu şekilde gösterilir:

$$\delta(n-a)=\begin{cases}1 & n=a\\ 0 & n\neq a\end{cases}$$

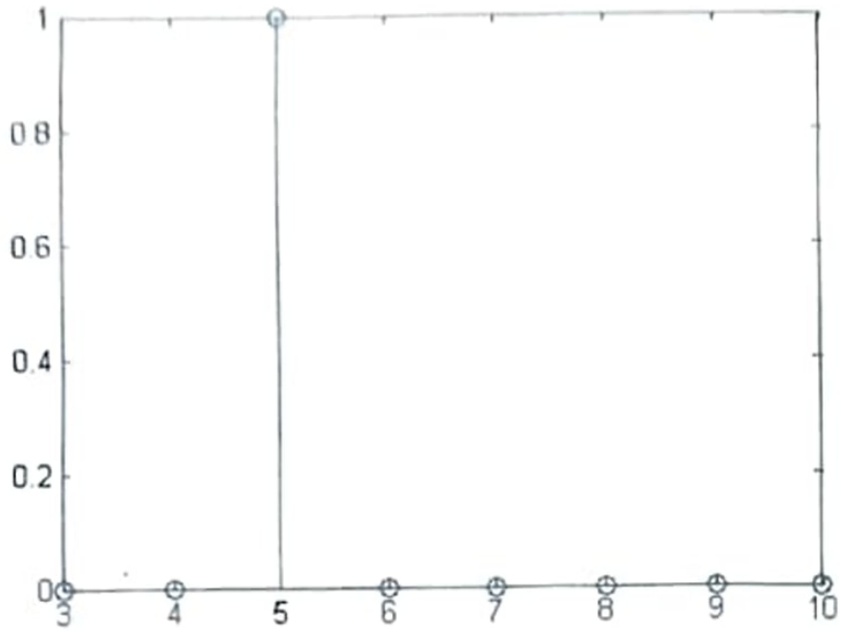
Bu fonksiyon Matlab'de bir alt program olarak şu şekilde tanımlanabilir.

```
function [y,t]=birim(a,b,c)
%a:fonksiyonun bire esit oldugu deger,
%b:fonksiyonun baslangic degeri
%c:fomksiyonun bitis degeri.
i=0;
for t=b:c;
i=i+1;
   %t, b ve c degerleri arasinda degissin.
   if (t==a)   %fonksiyonun bire esit oldugu yere gelinmistir
      y(i)=1;  %bunun için bir degerini ata
   else y(i)=0;
   end
end
```

Örnek:

0 0 0 0 0 1 0 0 0 0 dizisini, yani δ(n-6) fonksiyonunu, 3 ≤n ≤10 aralığında, yukarıda verilen alt programı kullanarak elde ediniz ve çizdiriniz.

```
t=3:10;
y=birim(5,3,10)
stem(t,y)

y =
   0    0    1    0    0    0    0    0
```



→ Birim basamak fonksiyonu:
Bu fonksiyon şu şekilde tanımlanır:

$$u(n)=\begin{cases}1 & n\geq 0\\ 0 & n<0\end{cases}$$

Fonksiyonun "a" kadar ötelenmiş hali şu şekilde gösterilir.

$$u(n-a)=\begin{cases}1 & n\geq a\\ 0 & n<a\end{cases}$$

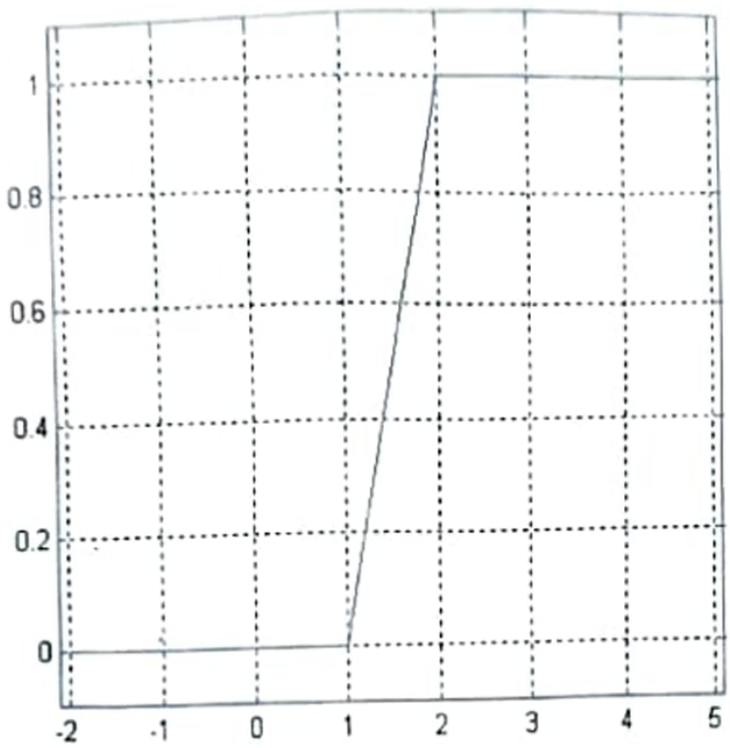
Bu fonksiyon Matlab'de bir alt program olarak şu şekilde tanımlanabilir.

```
function [y,t]=basamak(a,b,c)
%a:fonksiyon a'dan baslayarak 1 degerini alir.
%b:fonksiyonun baslangic degeri
%c:fomksiyonun bitis degeri.
i=0;
for t=b:c;
i=i+1;
   %t, b ve c degerleri arasinda degissin.
   if (t-a) >=0   %fonksiyonun a ve a'dan buyuk degerler icin 1 olur.
      y(i)=1;  %bunun için 1 degerini ata
   else y(i)=0;
   end
end
```

Örnek:
u(n-2) fonksiyonunu, $-2 \leq n \leq 5$ aralığında hesaplayınız ve çizdiriniz.

```
y=basamak(2,-2,5)
plot(t,y)
axis([-2.1 5.1 -0.1 1.1])
grid
```

```
y =
   0    0    0    0    1    1    1    1
```



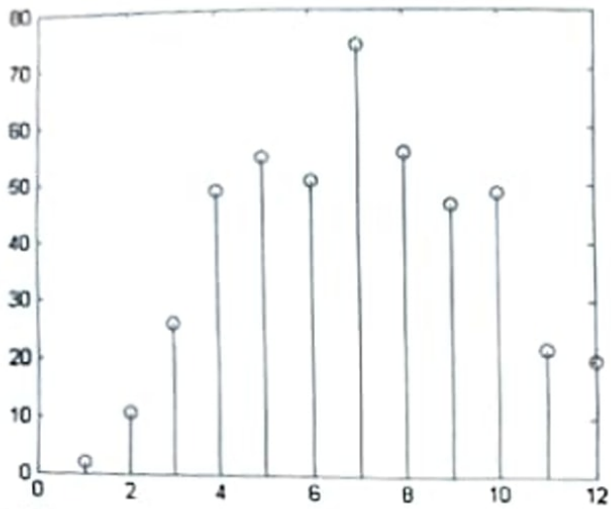
Konvolüsyon

Konvolüsyon, y(n)=x(n)*h(n) notasyonu ile gösterilir. y(n)'in ifadesi şu şekilde belirtilir:

$$y(n) = \sum_{k=-\infty}^{\infty} x(n-k).h(k)$$

→ conv komutu:
kullanımı: conv(a,b): Bu komut iki sinyalin konvolüsyonunu hesaplar.
Örnek:
x=[1 4 3 2 4] -2≤≤2
h=[2 3 8 4 1 7 3 5] -1≤≤
olduğuna göre y = x⊗h=?

```
x=[1 4 3 2 4]
h=[2 3 8 4 1 7 3 5]
y=conv(x,h)
stem(y)

y =
    2   11   26   49   55   51   74   56   47   49   22   20
```

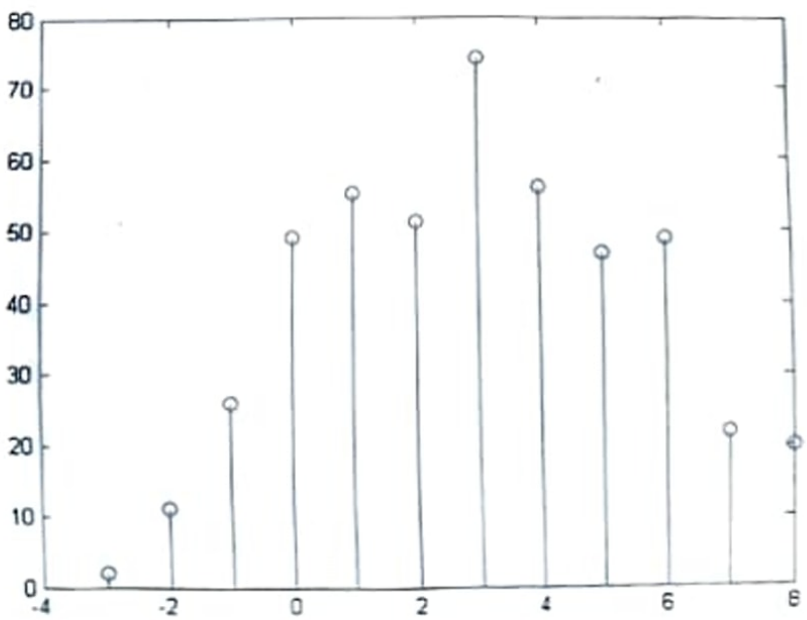


Grafikten görüldüğü gibi konvolüsyon doğru olarak hesaplanmıştır. Fakat grafik eksenleri yanlış olarak göstermektedir. Bunun için x'in ve h'nin değişin aralıkları göz önüne alınmalıdır. Bu değişkenler için iki vektör daha tanımlanmalıdır. Bu vektörlerin nasıl eklenildiği aşağıdaki programda gösterilmiştir.

```
x=[1 4 3 2 4]
h=[2 3 8 4 1 7 3 5]
xi=-2:2;
hi=-1:6;
t1=xi(1)+hi(1);t2=xi(end)+hi(end);
y=conv(x,h)
t=t1:t2
stem(t,y)
```

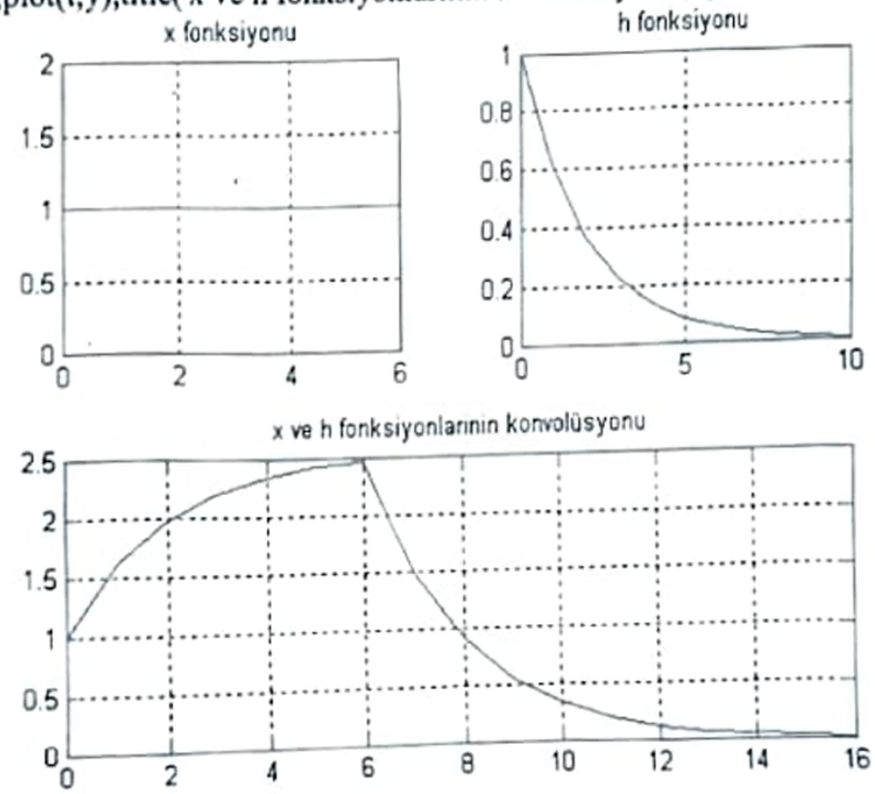
```
y =
   2   11   26   49   55   51   74   56   47   49   22   20
t =
  -3   -2   -1    0    1    2    3    4    5    6    7    8
```

Görüldüğü gibi y değeri için gerekli indislerde hesaba alınmıştır. Bu konvolüsyonun grafiği aşağıda verilmiştir.



Örnek 2:

```
x=[1 1 1 1 1 1 1];
xi=0:6;
h=[1 0.606 0.368 .223 .135 .0821 .0498 .0302 .0183 .0111 .0067] ;
hi=0:10;
y=conv(x,h)
t1=xi(1)+hi(1);
t2=xi(end)+hi(end);
t=t1:t2;
subplot(221),plot(xi,x),title('x fonksiyonu'),grid
subplot(222),plot(hi,h),title('h fonksiyonu'),grid
subplot(212),plot(t,y),title('x ve h fonksiyonlarinin konvolüsyonu'),grid
```

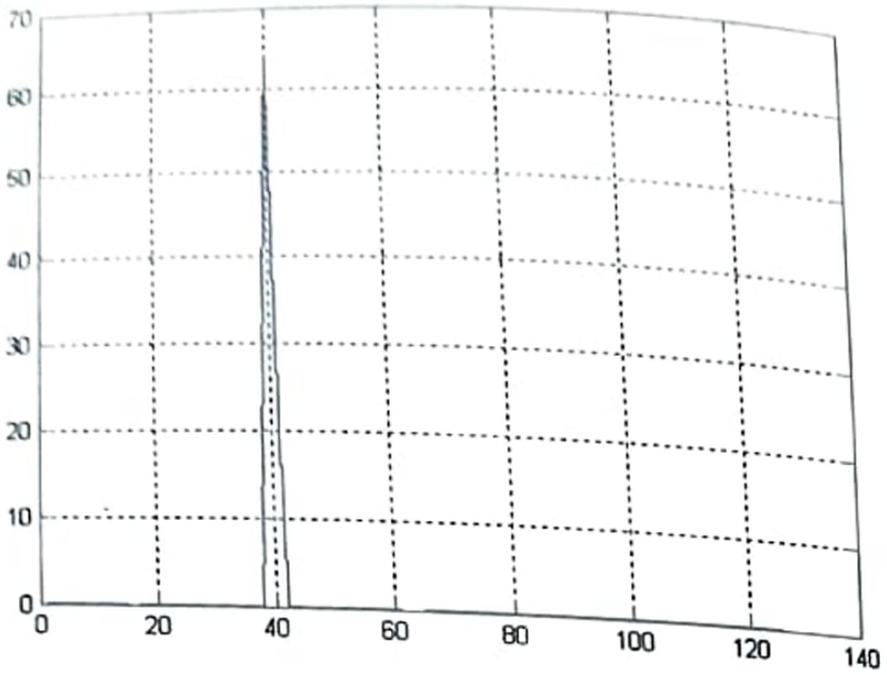


Fourier Dönüşümü:

Frekansı 20 Hz olan bir sinüs sinyalinin Matlab'de FFT'sini alalım.
FFT'si alınacak sinyal, grafiği aşağıda verilen bor cos sinyalidir. Asağıdaki Matlab programı ile bu cos'ün FFT'sini hesaplayabiliriz.

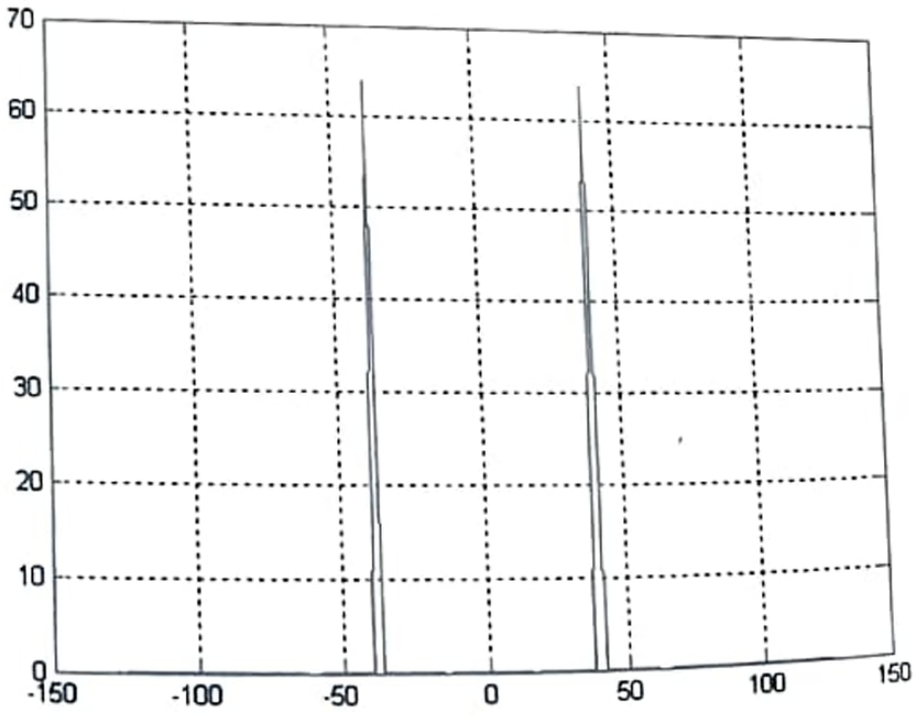
```
clc
N=128;T=1/256;f=40;
k=0:N-1;
x=cos(2*pi*f*k*T);
y=fft(x); %sinyalin FFT'sinin alinmasi
y_mag=abs(y);
f=k/(N*T);%frekans bölhesinde dogru degerlerin gözükmesi icin rakamlarin normalize
edilmesi
plot(f(1:N/2),y_mag(1:N/2)),grid
```

Bu durumda şu grafik elde edilir..

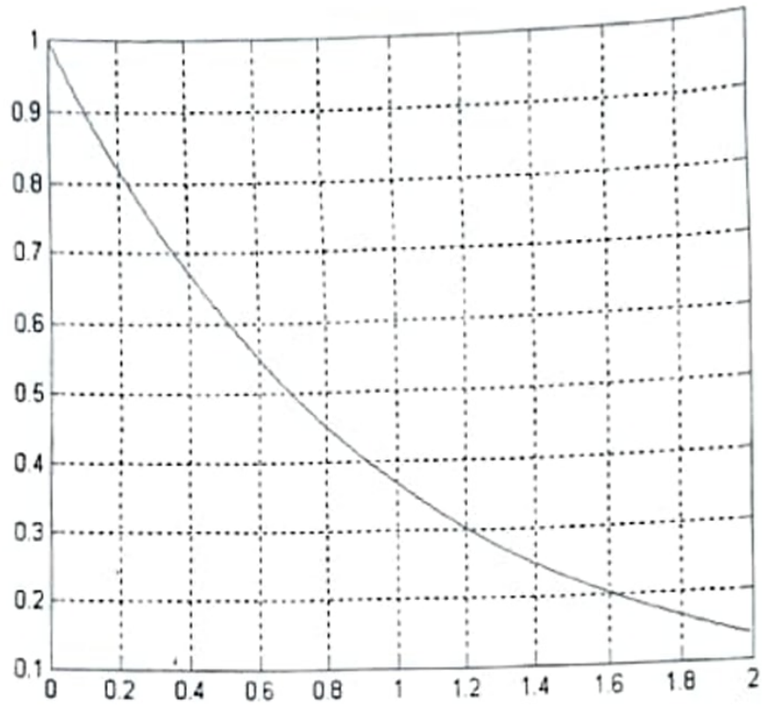


Eğer dönüşümde negatif değerli bileşenlerinde gösterilmesi istenilirse programa aşağıdaki bölüm eklenebilir. Bu durumda aşağıdaki grafik elde edilir.

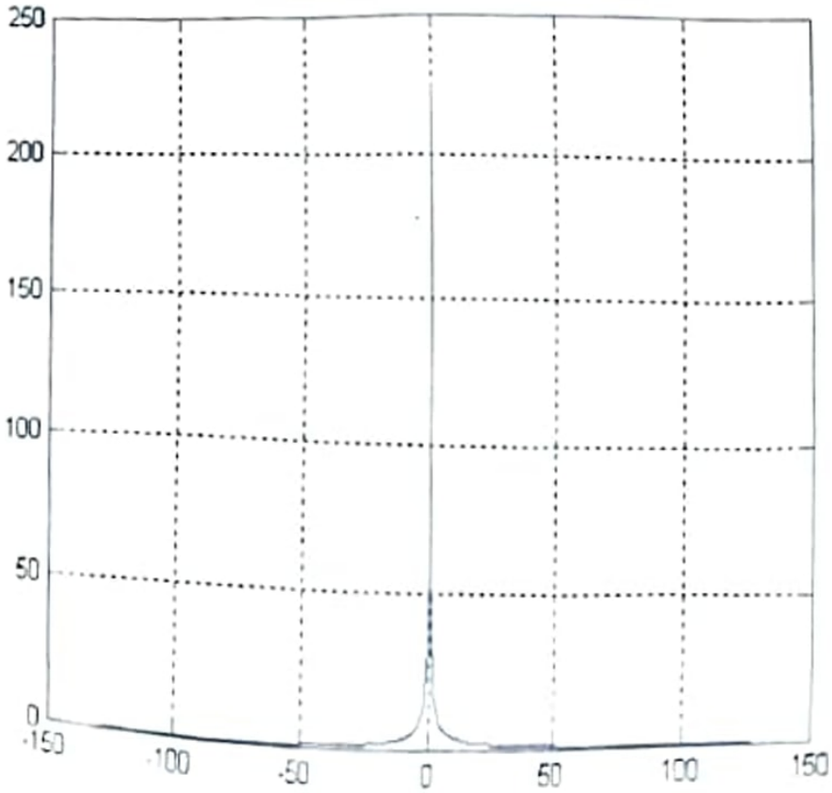
```
f2=f(N/2)-f(N/2:N);
y2=fliplr(y_mag(N/2:N));
plot(f2,y2,f(1:N/2),y_mag(1:N/2),'b'),grid;
```

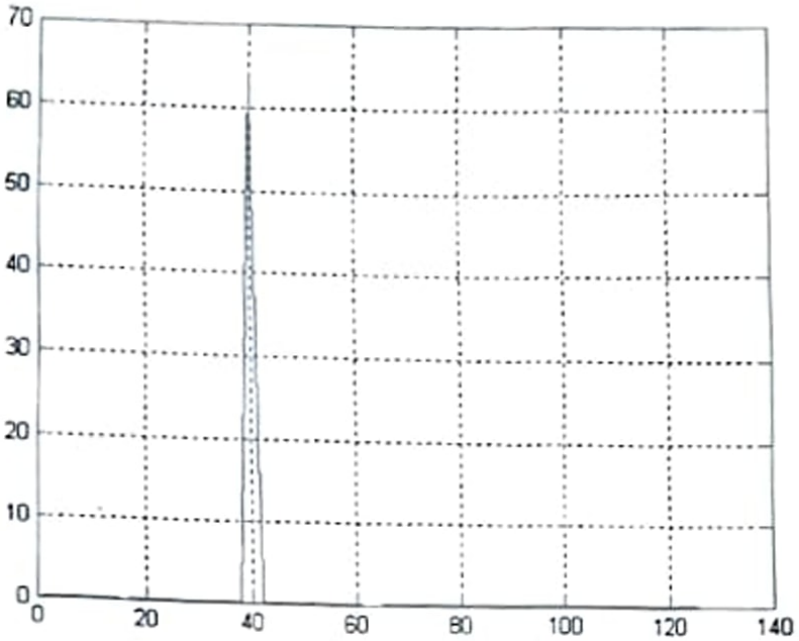


Örnek: Yandaki grafikte gösterilen $y=e^{-t}$ fonksiyonunun fourier dönüşümünü bulalım.



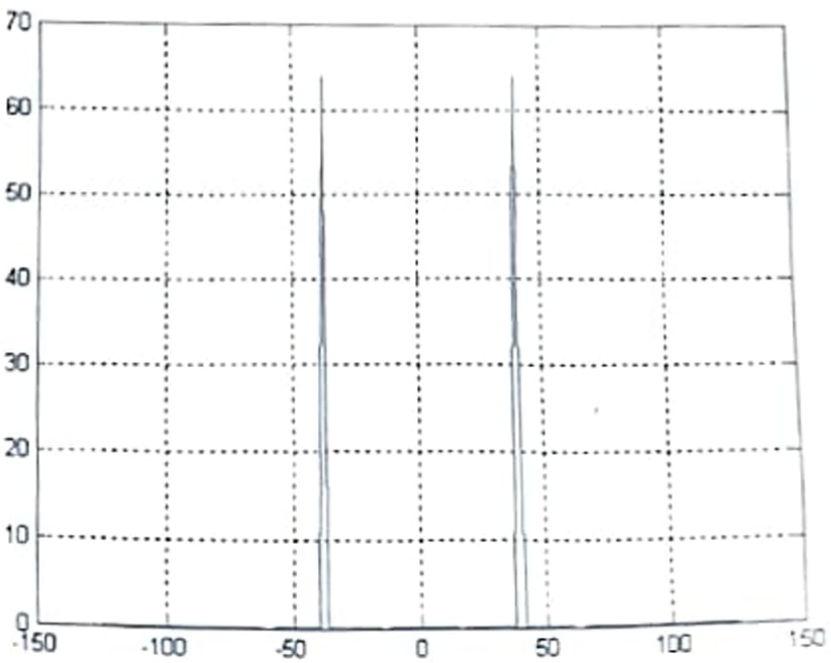

```
N=512;T=1/256;
k=0:N-1;
x=exp(-k*T);
y=fft(x); %sinyalin FFT'sinin alinmasi
y_mag=abs(y);
f=k/(N*T);%frekans bölgesinde dogru degerlerin gözükmesi icin rakamlarin normalize
edilmesi
f2=f(N/2)-f(N/2:N);
y2=fliplr(y_mag(N/2:N));
plot(f2,y2,f(1:N/2),y_mag(1:N/2),'b'),grid;
```



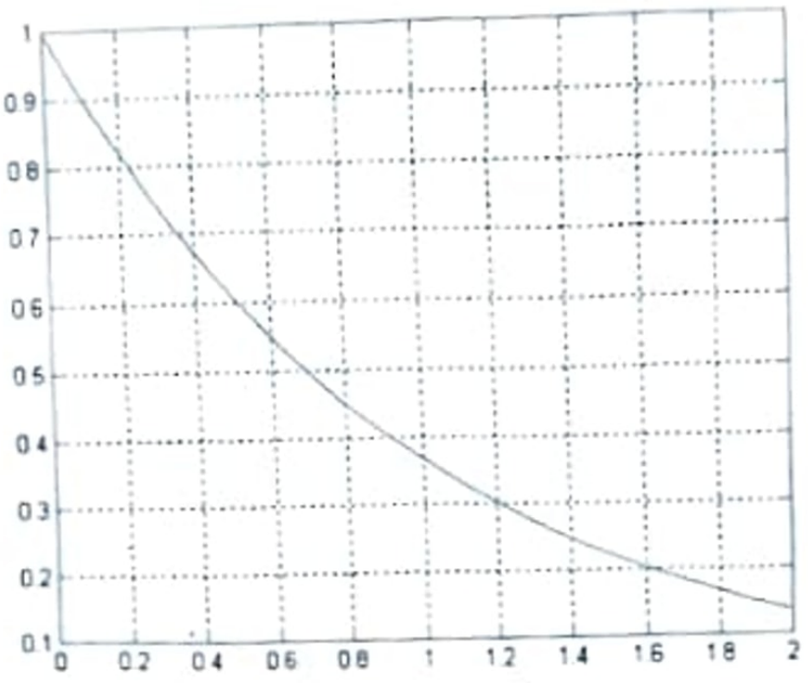


Eğer dönüşümde negatif değerli bileşenlerinde gösterilmesi istenilirse programa aşağıdaki bölüm eklenebilir. Bu durumda aşağıdaki grafik elde edilir.

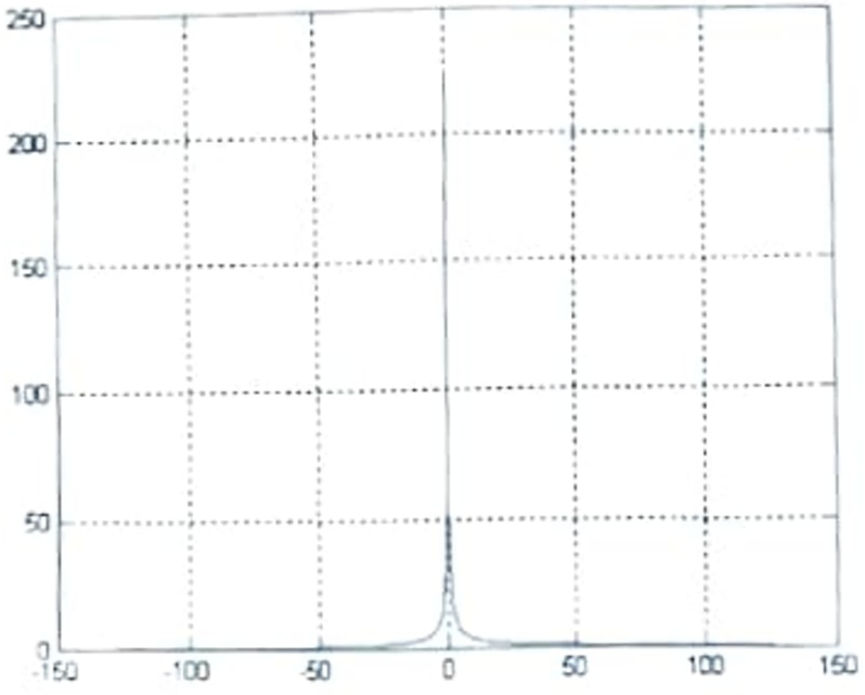
```
f2=f(N/2)-f(N/2:N);
y2=fliplr(y_mag(N/2:N));
plot(f2,y2,f(1:N/2),y_mag(1:N/2),'b'),grid;
```



Örnek: Yandaki grafikte gösterilen $y=e^{-t}$ fonksiyonunun fourier dönüşümünü bulalım.




```
N=512;T=1/256;
k=0:N-1;
x=exp(-k*T);
y=fft(x); %sinyalin FFT'sinin alinmasi
y_mag=abs(y);
f=k/(N*T);%frekans bölgesinde dogru degerlerin gözükmesi icin rakamlarin normalize
edilmesi
f2=f(N/2)-f(N/2:N);
y2=fliplr(y_mag(N/2:N));
plot(f2,y2,f(1:N/2),y_mag(1:N/2),'b'),grid;
```



IIR FİLTRELER

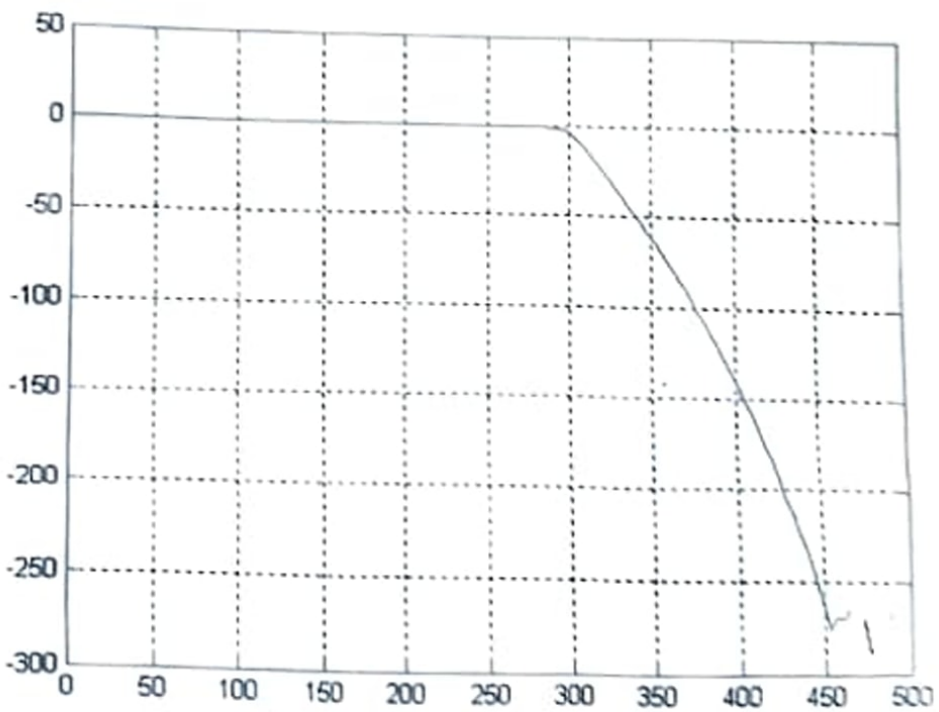
Matlab programı ile aşağıda verilen filtre türlerini tanımlamak mümkündür.

→ Butterworth Filtreler:

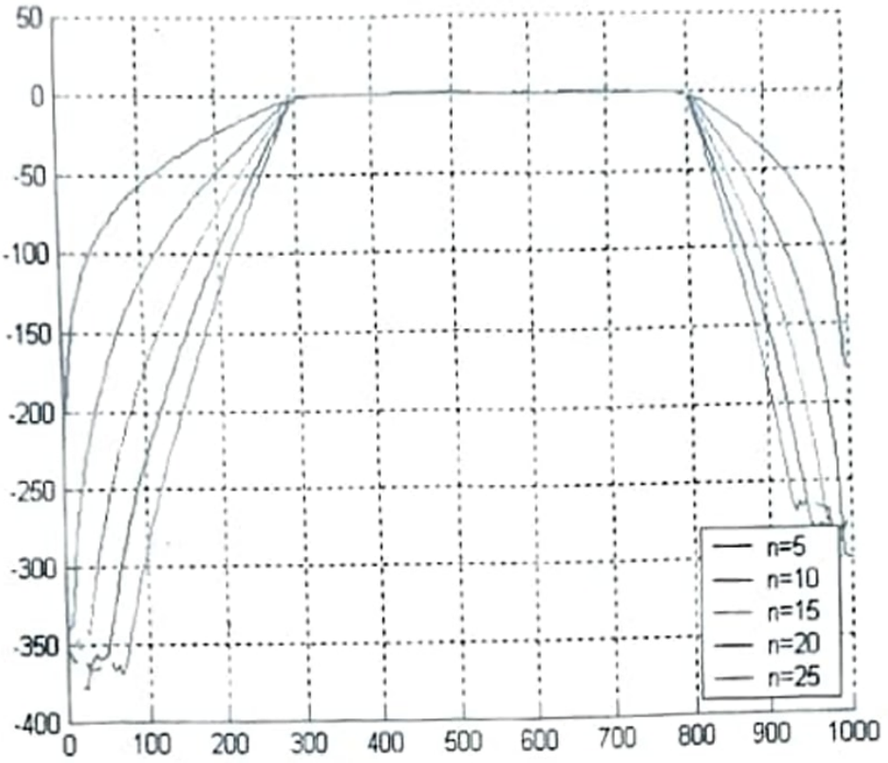
[b a]=butter(n,Wn,'tip') komutu ile sayısal Butterworth filtreleri tasarlamak mümkündür. Burada n, filtrenin derecesini, Wn'de kesim frekansını göstermektedir. Wn 0 ile 1 arasında bir değer almalıdır. Burada 1 nyquist frekansına denk gelmektedir. Tip olarak 'high' yüksek geçiren filtre için kullanılırken, 'stop' bant durduran filtre için kullanılır.

Örnek: Örnekleme frekansı 1000 Hz, derecesi 20, kesim frekansı 300 Hz olan bir alçak geçiren Butterworth filtre tasarlayınız.

```
fs=1000;n=20;fc=300;
[b a]=butter(n,fc/(fs/2));
[h w]=freqz(b,a,128);
f=w*fs/(2*pi);
h=20*log10(abs(h));
plot(f,h)
grid
```



Not: Derece arttıkça filtrenin doğruluk oranı artır. Aşağıdaki grafikte değişik dereceler için bant geçiren Butterworth filtreler gösterilmiştir



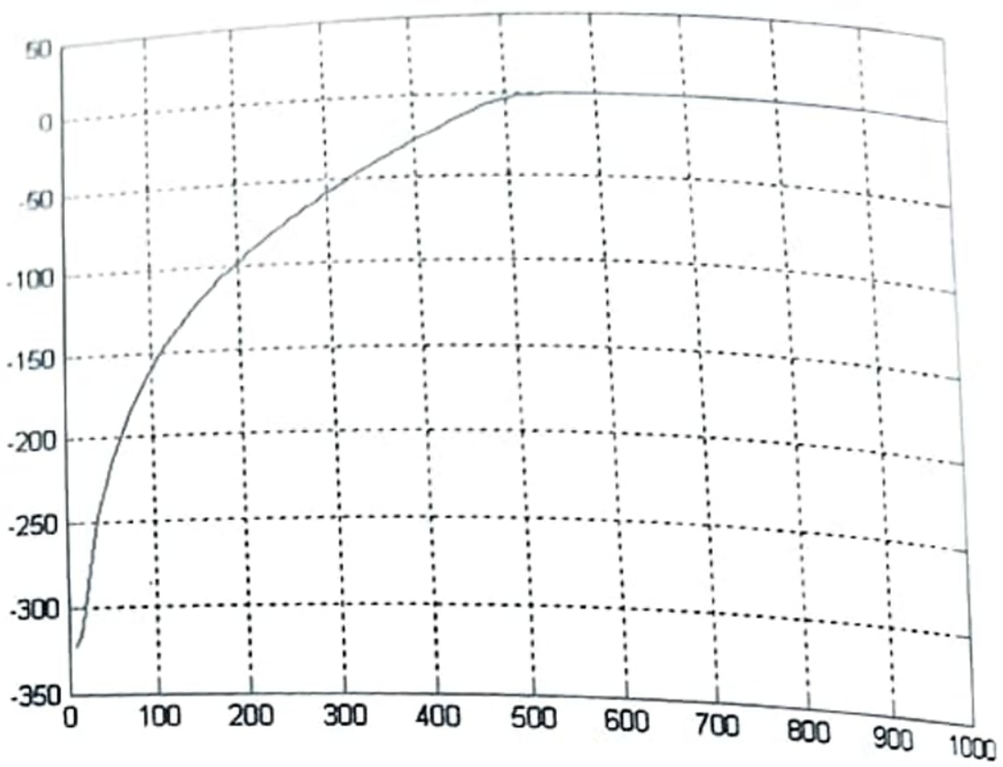
Görüldüğü gibi derece arttıkça filtrenin doğruluğu da artmaktadır. Yukarıdaki grafiği elde etmek için gerekli Matlab programı aşağıda verilmiştir.

```
clc
fs=2000;fc=[300 800];
r=['b','r','g','k','m']
i=0
for n=5:5:25
i=i+1
[b a]=butter(n,fc/(fs/2));
[h w]=freqz(b,a,256);
f=w*fs/(2*pi);
h=20*log10(abs(h));
hold on
plot(f,h,r(i))
end
legend('n=5','n=10','n=15','n=20','n=25',4)
grid
```

Örnek: Örnekleme frekansı 2000 Hz, derecesi 10, kesim frekansı 500 Hz olan bir yüklsek geçiren Butterworth filtre tasarlayınız.

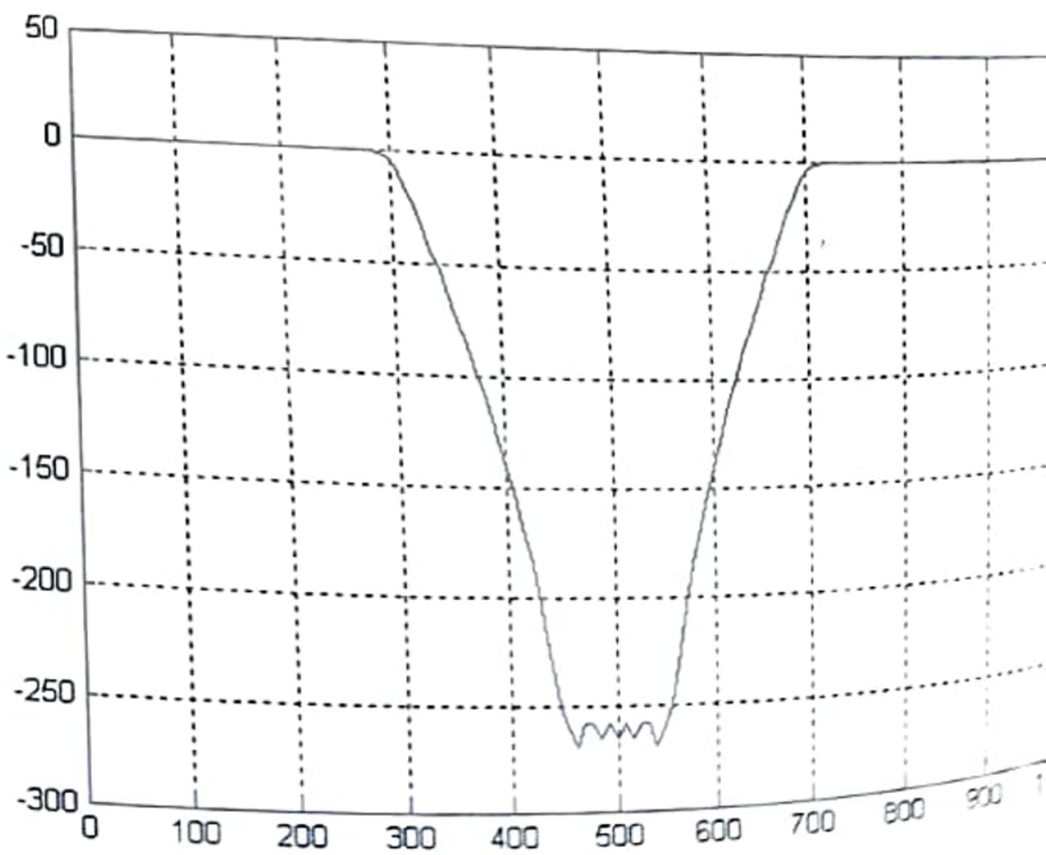
```
fs=2000;n=10;fc=500;
[b a]=butter(n,fc/(fs/2),'high');
[h w]=freqz(b,a,128);
f=w*fs/(2*pi);
h=20*log10(abs(h));
plot(f,h)
grid
```





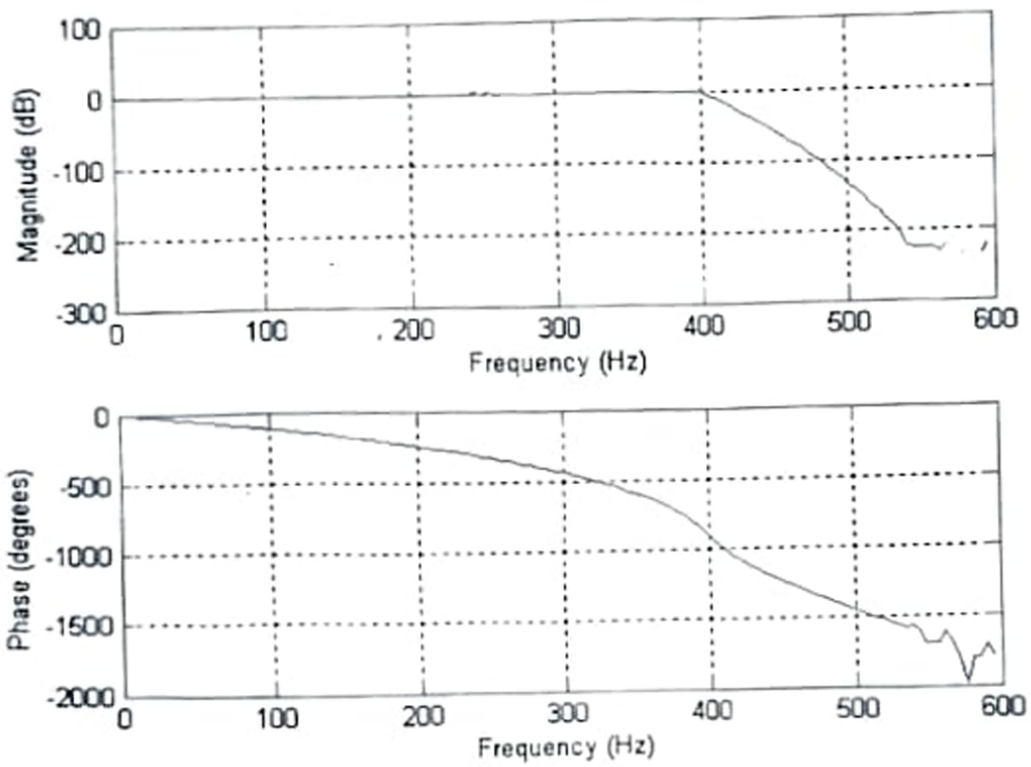
Örnek: Örnekleme frekansı 2000 Hz, derecesi 20, kesim frekansı 300 ve 700 band durduran Butterworth filtre tasarlayınız

```
fs=2000;n=20;fc=[300 700];
[b a]=butter(n,fc/(fs/2),'stop');
[h w]=freqz(b,a,128);
f=w*fs/(2*pi);
h=20*log10(abs(h));
plot(f,h)
grid
```



Açıklama: freqz komutu ile, bir filtrenin karakteristiğinin genlik ve açısını, hesaplama yapılmadan direkt olarak çizdirmek mümkündür.

Örnek: Örnekleme frekansı 1200 Hz, derecesi 20, kesim frekansı 400 Hz olan bir yüksek geçiren Butterworth filtre tasarlayınız.



→ Chebyshev Filtreler Tip 1:

Kullanımı: [b,a] = cheby1(n,Rp,Wn,'tip') komutu ile sayısal Chebyshev 1. tip filtreleri tasarlamak mümkündür. Burada n, filtrenin derecesini, Wn'de kesim frekansını göstermektedir. Wn 0 ile 1 arasında bir değer almalıdır. Burada 1 nyquist frekansına denk gelmektedir. Tip olarak 'high' yüksek geçiren filtre için kullanılılırken, 'stop' bant durduran filtre için kullanılır. Rp, band geçiren bölgede dB cinsinden titreşimi gösterir.

Örnek:

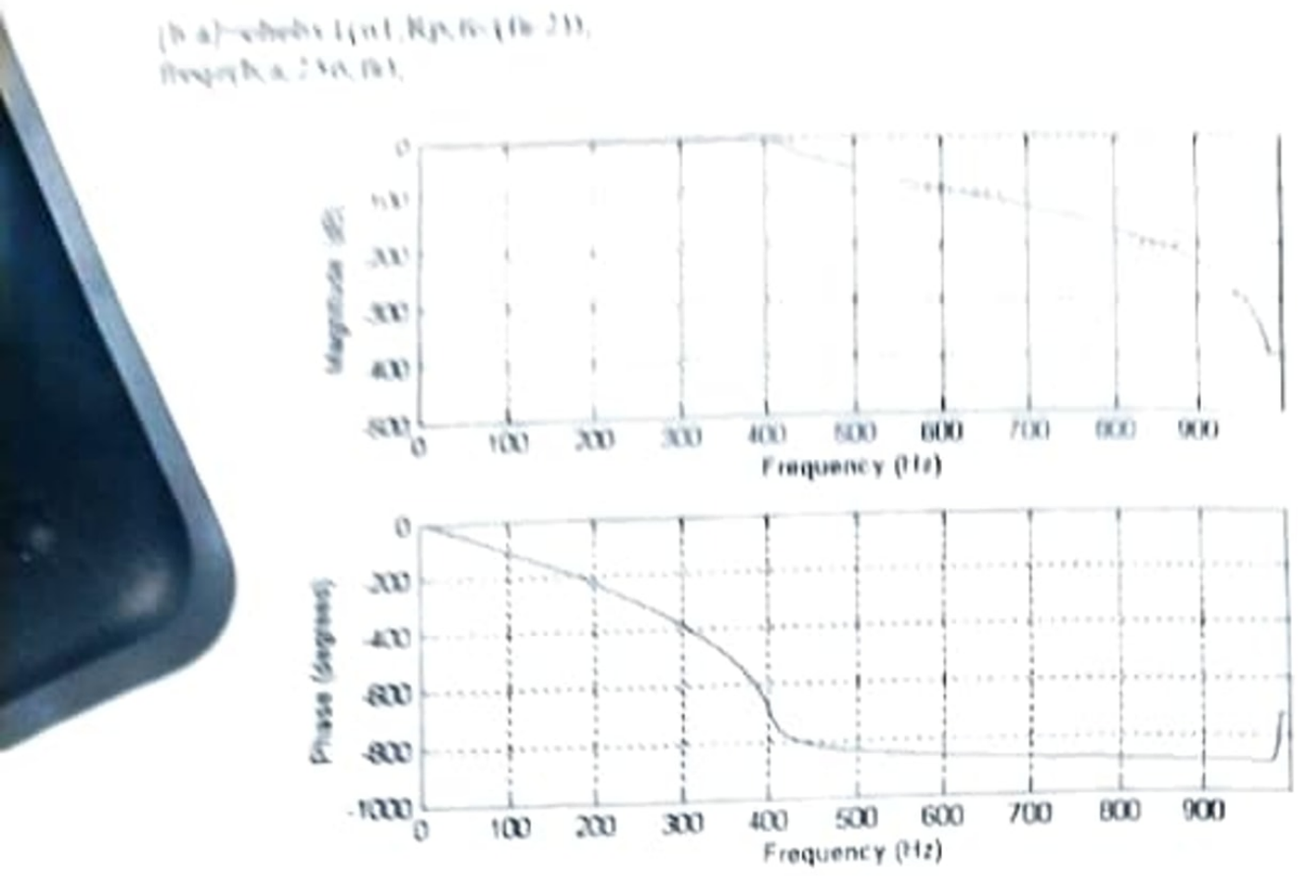
a) Örnekleme frekansı 2000 Hz, kesim frekansı 400 Hz, derecesi 10 ve bant geçiren bölge titreşimi 0.5 olan bir alçak geçiren Chebyshev filtre ,
b) Örnekleme frekansı 2000 Hz, kesim frekansı 600 Hz, derecesi 15 ve bant geçiren bölge titreşimi 0.5 olan bir yüksek geçiren Chebyshev filtre
c) Örnekleme frekansı 2000 Hz, kesim frekansı 300 ve 700 Hz, derecesi 10 ve bant geçiren bölge titreşimi 0.5 olan bir bant geçiren Chebyshev filtre
d) Örnekleme frekansı 2000 Hz, kesim frekansı 300 ve 700 Hz, derecesi 10 ve bant geçiren bölge titreşimi 0.5 olan bir bant durduran Chebyshev filtre

tasarlayıp çizdiriniz.

Cevap:

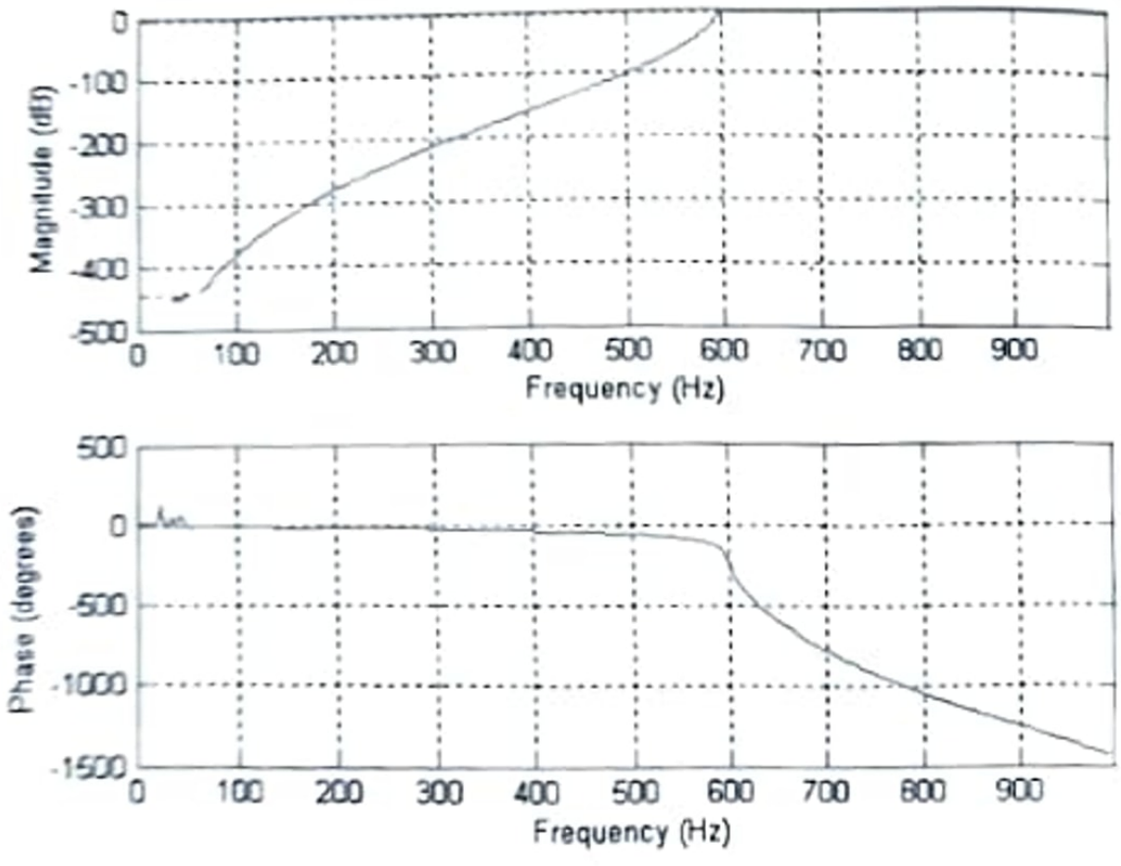
a)

```
fs=2000;n=10;fc=400,Rp=0.5;
```



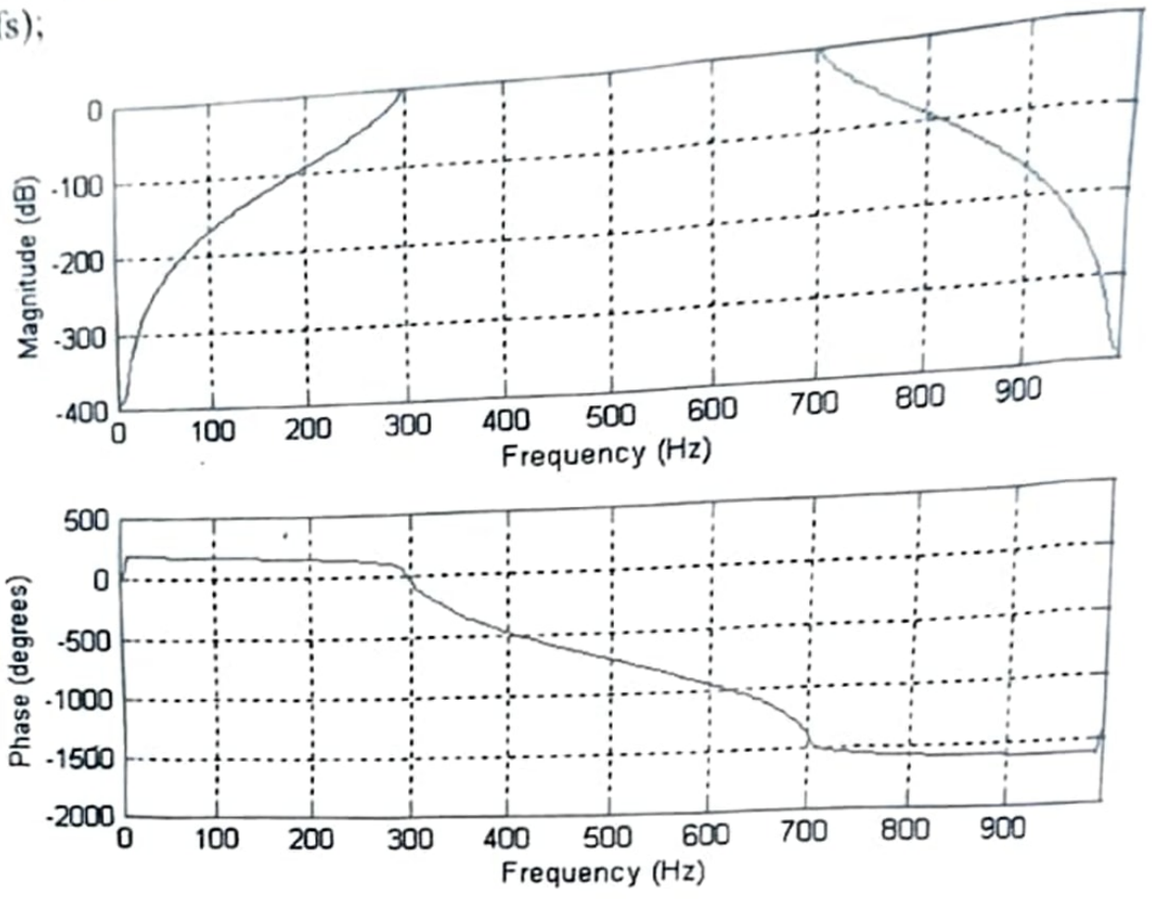
b)

```
fs=2000;n=16;fc=600,Rp=0.5;
[b a]=cheby1(n,Rp,fc/(fs/2),'high');
freqz(b,a,256,fs);
```



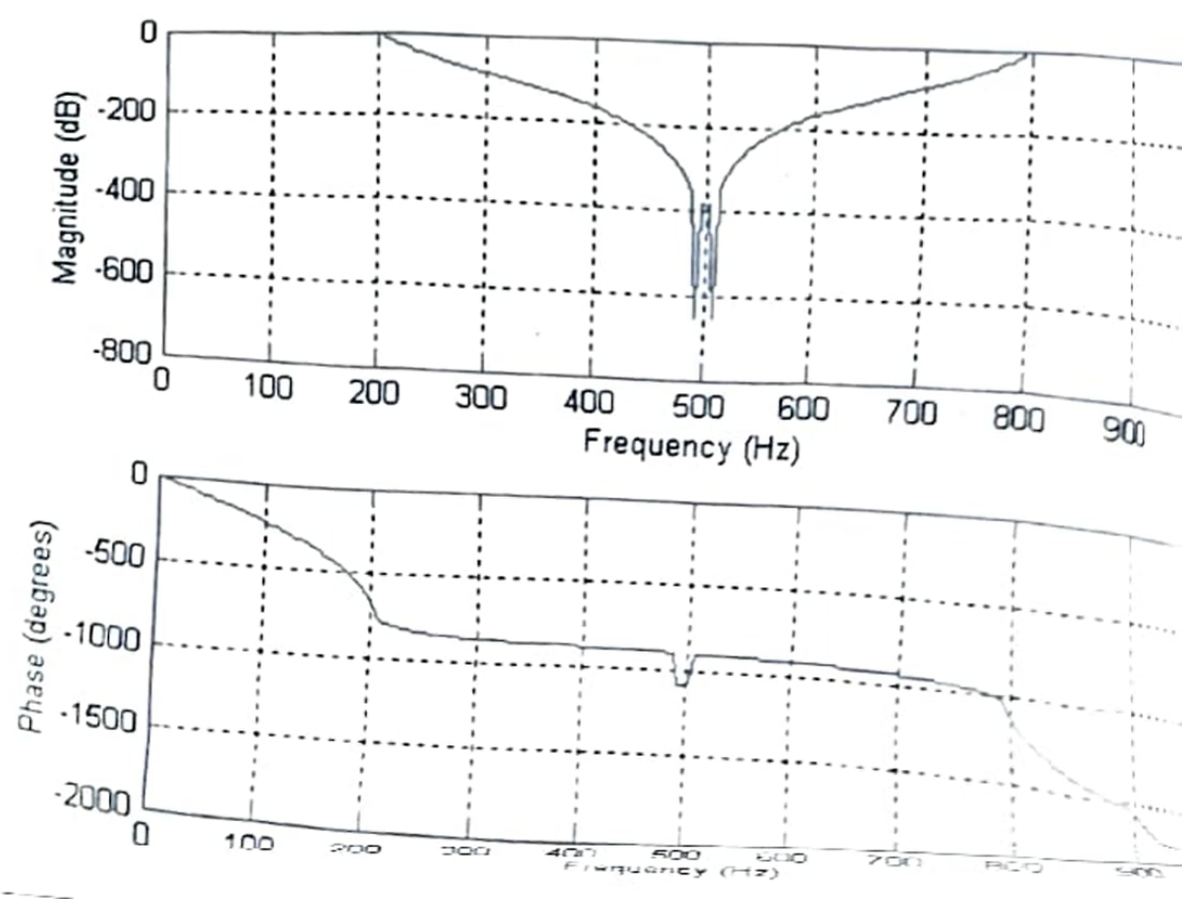
c)

```
fs=2000;n=10;fc=[300 700],Rp=0.5;
[b a]=cheby1(n,Rp,fc/(fs/2));
freqz(b,a,256,fs);
```



d)

```
fs=2000;n=10;fc=[200 800],Rp=0.5;
[b a]=cheby1(n,Rp,fc/(fs/2),'stop');
freqz(b,a,256,fs);
```



→ **Chebyshev Filtreler Tip 2:**

Kullanımı:
[b,a] = cheby2(n,Rp,Wn,'tip') komutu ile sayısal Chebyshev 2. tip filtreleri tasar
mümkündür. Burada n, filtrenin derecesini, Wn'de kesim frekansını göstermektedir. W
1 arasında bir değer almalıdır. Burada 1 nyquist frekansına denk gelmektedir. Tip
'high' yüksek geçiren filtre için kullanılırken, 'stop' bant durduran filtre için kullan
band durduran bölgede dB cinsinden titreşimi gösterir.

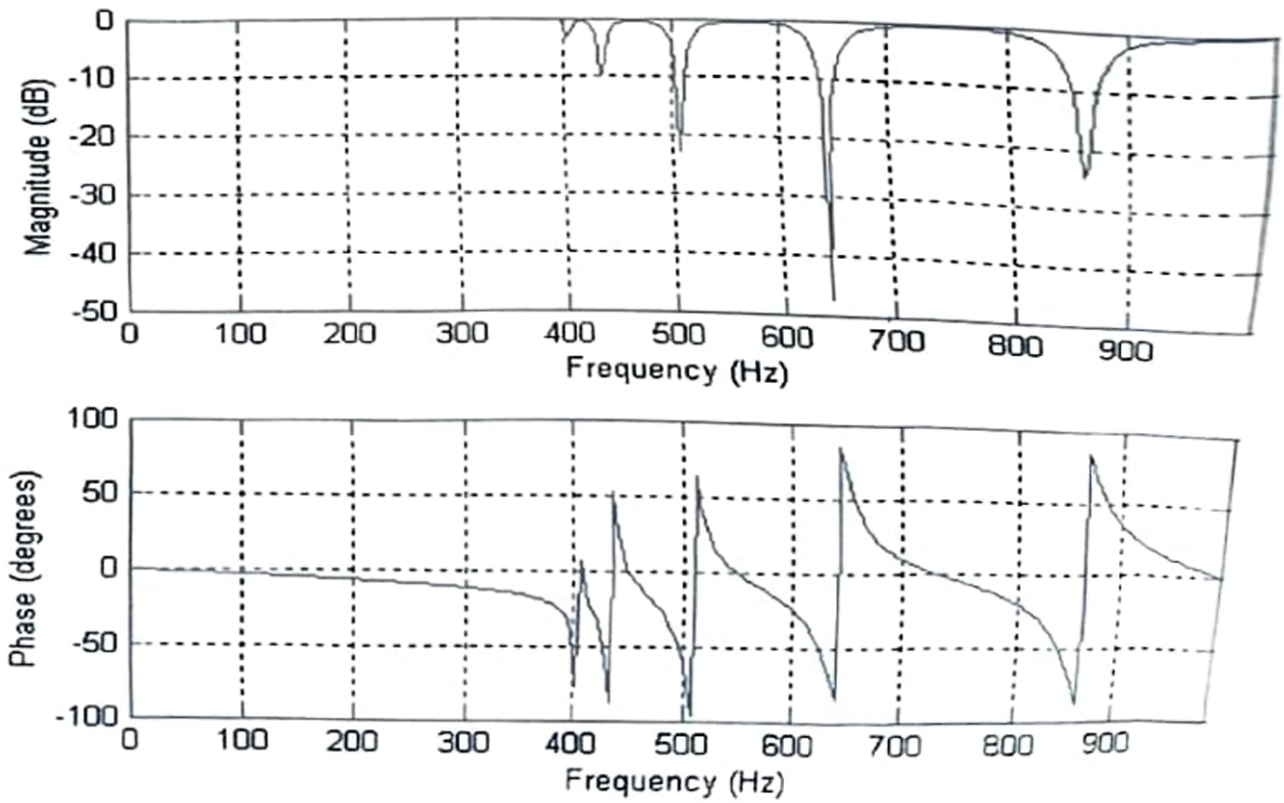
Örnek:

e) Örnekleme frekansı 2000 Hz, kesim frekansı 400 Hz, derecesi 10 ve bant
bölge titreşimi 0.5 olan bir alçak geçiren Chebyshev(II) filtre ,
f) Örnekleme frekansı 2000 Hz, kesim frekansı 600 Hz, derecesi 15 ve bant dur
bölge titreşimi 10 olan bir yüksek geçiren Chebyshev(II) filtre
g) Örnekleme frekansı 2000 Hz, kesim frekansı 300 ve 700 Hz, derecesi 10 ve
durduran bölge titreşimi 10 olan bir bant geçiren Chebyshev(II) filtre
h) Örnekleme frekansı 2000 Hz, kesim frekansı 300 ve 700 Hz, derecesi 20 ve
durduran bölge titreşimi 0.5 olan bir bant durduran Chebyshev(II) filtre

tasarlayıp çizdiriniz

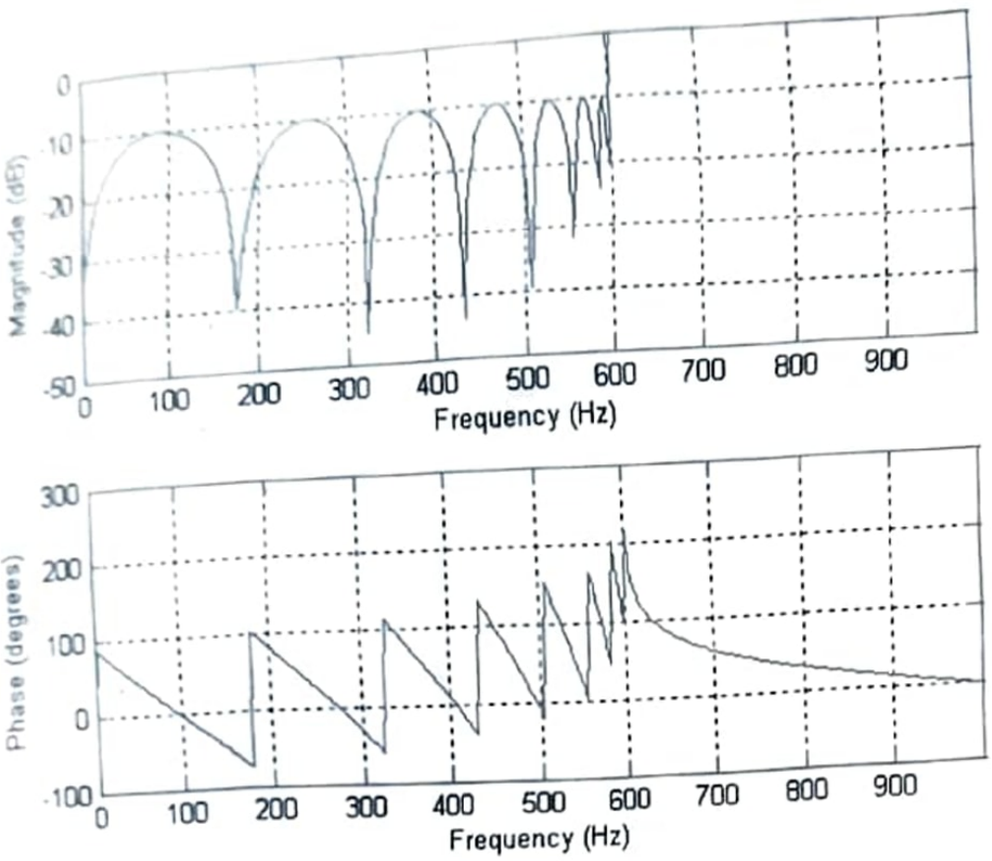
a)

```
fs=2000;n=10;fc=400,Rs=0.5;
[b a]=cheby2(n,Rs,fc/(fs/2));
freqz(b,a,256,fs);
```



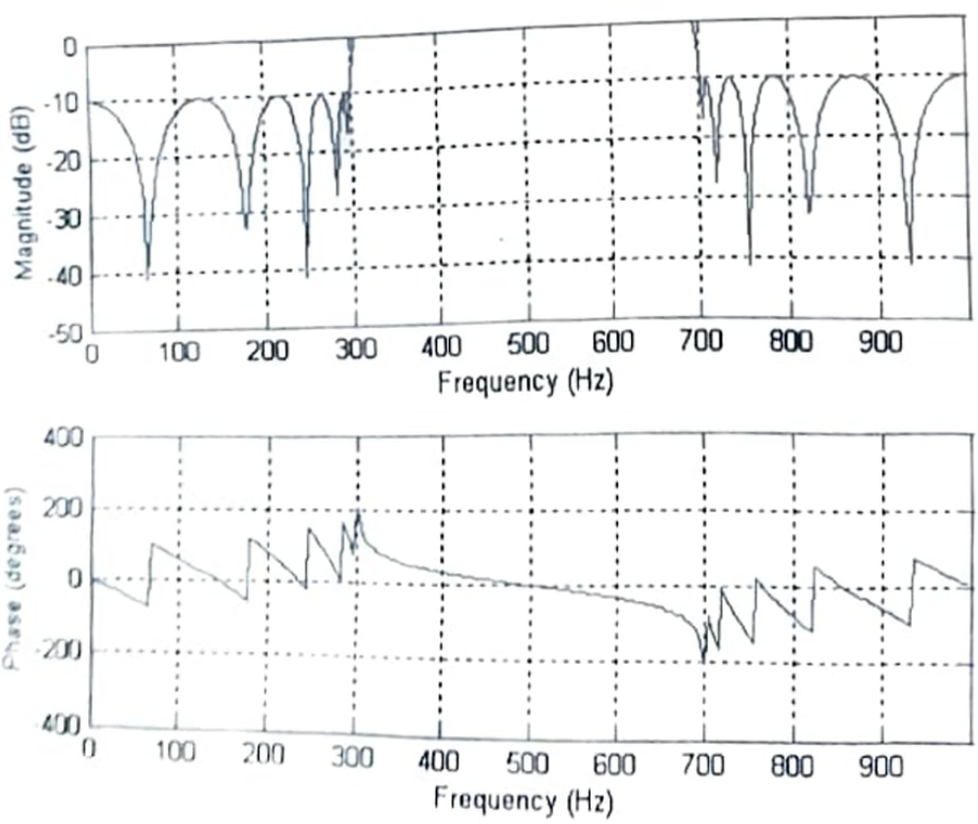
b)

...=2000;n=15;fc=600;Rs=10
...a]=cheby2(n,Rs,fc/(fs/2),'high');
freqz(b,a,256,fs);

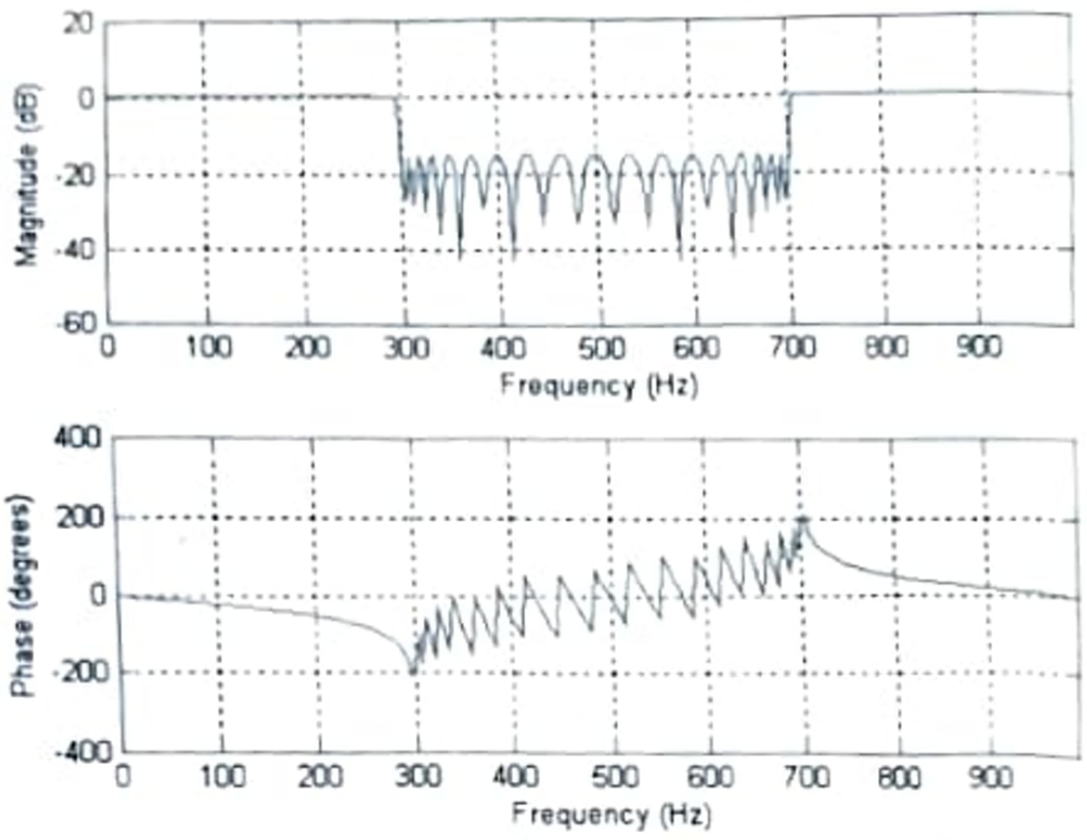


c)

```
fs=2000;n=10;fc=[300 700],Rs=10
[b a]=cheby2(n,Rs,fc/(fs/2));
freqz(b,a,256,fs);
```



```
d)
fs=2000,n=20,fc=[300 700],Rs=15
[b a]=cheby2(n,Rs,fc/(fs/2),'stop');
freqz(b,a,256,fs),
```



→Elliptic Filtreler:

Kullanımı:
[b,a] = ellip(n,Rp,Rs,Wn,'tip') komutu ile sayısal Elliptic filtreleri tasarlamak mümkündür. Burada n, filtrenin derecesini, Wn'de kesim frekansını göstermektedir. Wn 0 ile 1 arasında bir değer almalıdır. Burada 1 nyquist frekansına denk gelmektedir. Tip olarak 'high' yüksek geçiren filtre için kullanılırken, 'stop' bant durduran filtre için kullanılır. Rs band durduran bölgede, Rp band geçiren bölgede dB cinsinden titreşimi gösterir.

Örnek:

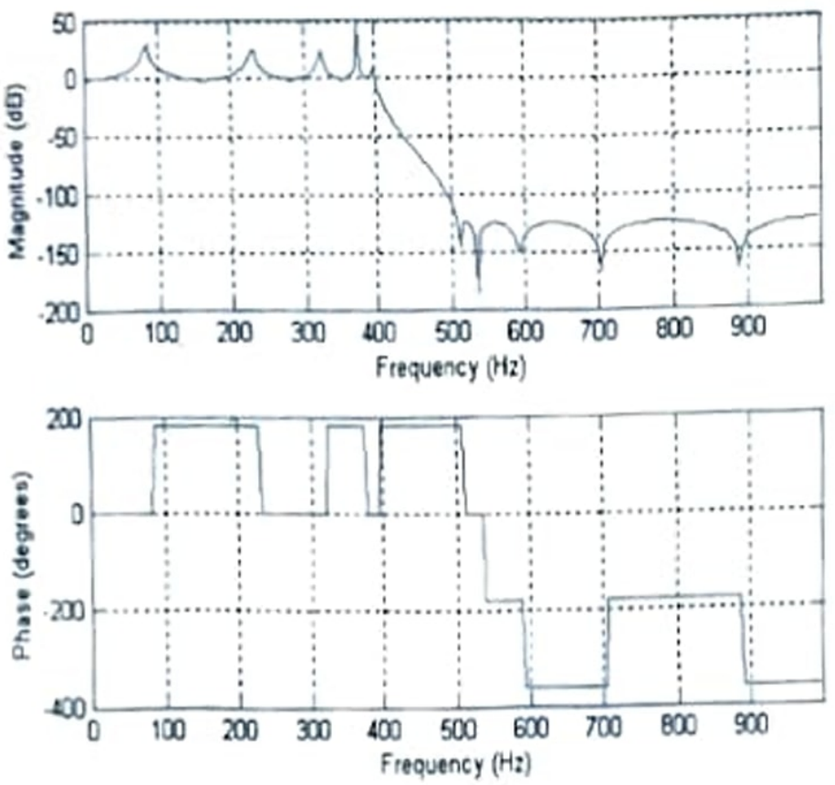
a) Örnekleme frekansı 2000 Hz, kesim frekansı 400 Hz, derecesi 10 , bant durduran ve bandt geçiren bölge titreşimleri 0.5 olan bir alçak geçiren Elliptic filtre ,
b) Örnekleme frekansı 2000 Hz, kesim frekansı 600 Hz, derecesi 15 ve bant durduran ve bant geçiren bölge titreşimi 0.5 olan bir yüksek geçiren Elliptic filtre
c) Örnekleme frekansı 2000 Hz, kesim frekansı 300 ve 700 Hz, derecesi 10 ,bant durduran ve bant geçiren bölge titreşimi 0.5 olan bir bant geçiren Elliptic filtre
d) Örnekleme frekansı 2000 Hz, kesim frekansı 300 ve 700 Hz, derecesi 20, bant durduran ve bant geçiren bölge titreşimi 0.5 olan bir bant durduran Elliptic filtre

tasarlayıp çizdiriniz

Cevap

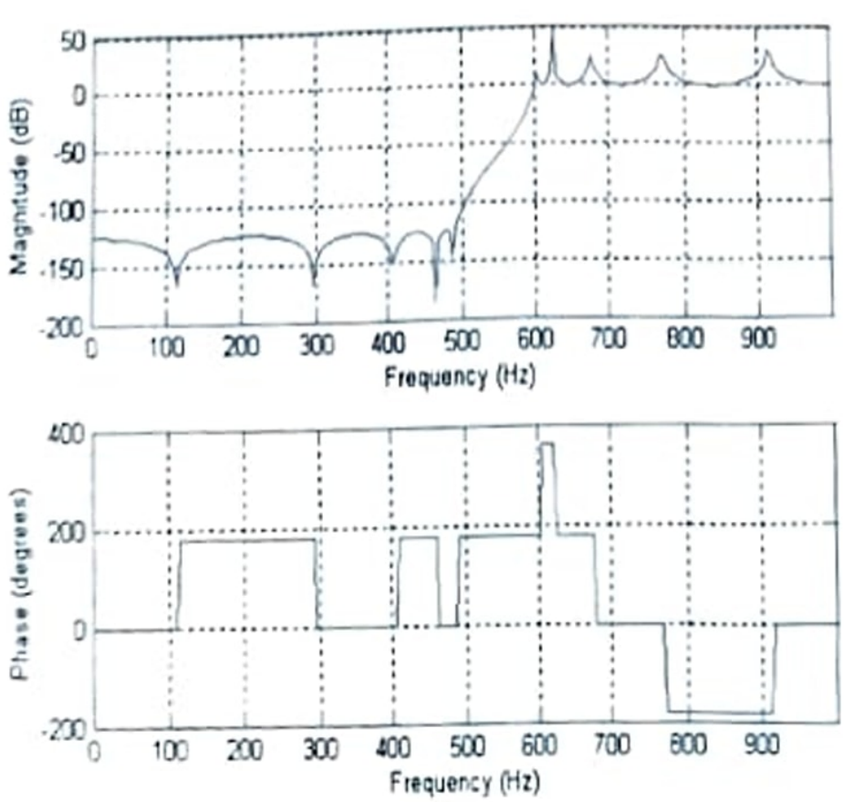
a)

```
fs=2000;n=10;fc=400,Rs=0.5,Rp=0.5
[b a]=ellip(n,Rp,Rs,fc/(fs/2));
freqz(b,a,256,fs);
```



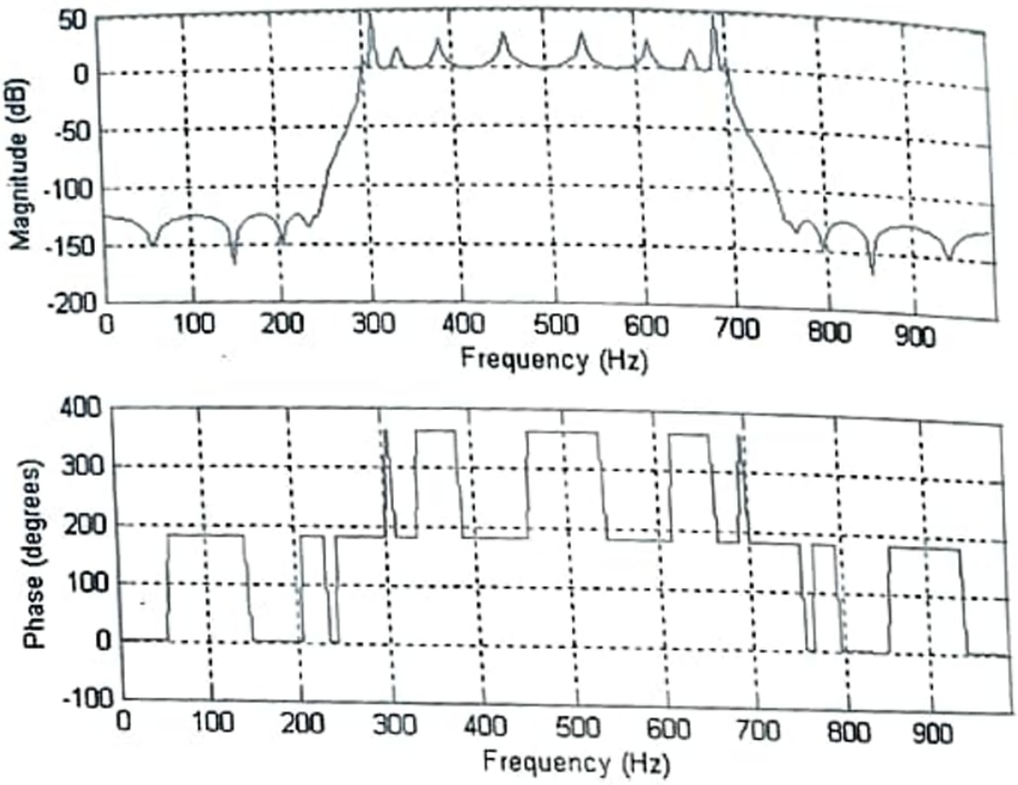
b)

```
fs=2000;n=10;fc=600,Rs=0.5,Rp=0.5
[b a]=ellip(n,Rp,Rs,fc/(fs/2),'high');
freqz(b,a,256,fs);
```



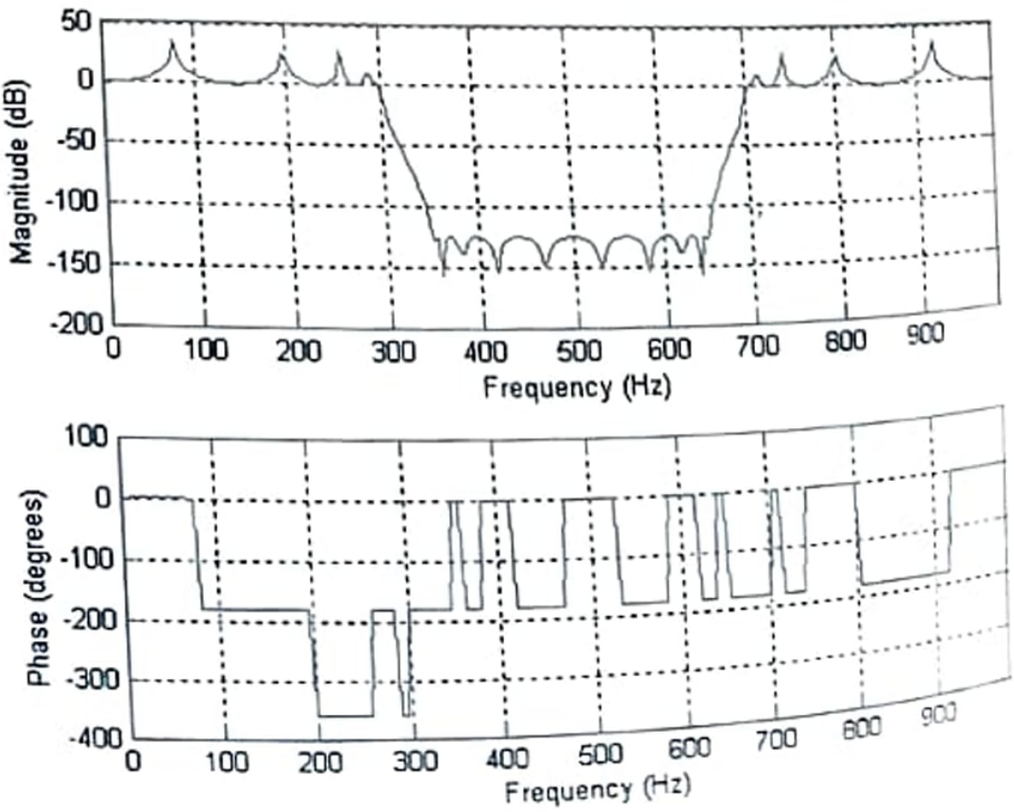
c)

```
fs=2000;n=10;fc=[300 700],Rs=0.5,Rp=0.5
[b a]=ellip(n,Rp,Rs,fc/(fs/2));
freqz(b,a,256,fs);
```



d)

```
fs=2000;n=10;fc=[300 700],Rs=0.5,Rp=0.5
[b a]=ellip(n,Rp,Rs,fc/(fs/2),'stop');
freqz(b,a,256,fs);
```



IIR Filtrelerde Derece Tanımlama

→ buttord komutu:
Kullanımı: [n,Wn] = buttord(Wp,Ws,Rp,Rs): Burada n, filtrenin derecesini, Wn, filtrenin kesim frekansını, Rp bant geçiren bölgede dB cinsinden titreşim, Rs bant durduran bölgede dB cinsinden zayıflama, Wp, bant geçiren bölge köşe frekansını, Ws, bant durduran bölge köşe frekansını göstermektedir.

Örnek: 2000 Hz'de örneklenen bir sinyal için, 0 Hz'den başlayıp 400 Hz'e kadar devam eden bant geçiren bölge için 2 dB titreşime, 700 Hz'den başlayıp Nyquist frekansına kadar devam eden bant geçiren bölge için 50 dB zayıflamaya sahip bir alçak geçiren filtre tasarlanmak istenıyor. Bu fitlerinin dereçesini ve kesim frekansını hesaplayınız.

Çözüm:
```
fs=2000;
Wp = 400/(fs/2); Ws = 700/(fs/2);Rp=2;Rs=50;
[n,Wn] = buttord(Wp,Ws,Rp,Rs);
Wn=Wn*fs/2;
clc
fprintf('Filtrenin minimum derecesi=%2d olmalidir\n',n);
fprintf('Filtrenin kesim frekansi=%5.2f olmalidir\n',Wn);
```

Filtrenin minimum derecesi= 7 olmalidir
Filtrenin kesim frekansi=453.04 olmalidir

→ cheb1ord komutu:
Kullanımı: [n,Wn] = cheb1ord(Wp,Ws,Rp,Rs). Burada n, filtrenin derecesini, Wn, filtrenin kesim frekansını, Rp bant geçiren bölgede, Rs bant durduran bölgede dB cinsinden zayıflamayı, Wp, bant geçiren bölge köşe frekansını, Ws, bant durduran bölge köşe frekansını göstermektedir.

Örnek: Bilginin 2000 Hz'de örneklendiği bir ortamda bant geçiren bölgenin köşe frekanslarının 300, 700 Hz, bant durduran bölgenin köşe frekanslarının 200 ile 800 Hz olduğu, bant geçiren bölge için 3 dB titreşime sahip, bant durduran bölge için 50 dB zayıflamaya sahip bir bant geçiren filtre tasarlayınız. Bu filtrenin derecesini ve kesim frekansını bulunuz.

```
fs=2000;
Wp = [300 700]/(fs/2); Ws = [200 800]/(fs/2);Rp=3;Rs=150;
[n,Wn] = cheb1ord(Wp,Ws,Rp,Rs);
Wn=Wn*fs/2;
clc
fprintf('Filtrenin minimum derecesi=%2d olmalidir\n',n);
fprintf('Filtrenin kesim frekansi=%5.2f olmalidir\n',Wn);
```

Filtrenin minimum derecesi=15 olmalidir
Filtrenin kesim frekansi=300.00 olmalidir
Filtrenin kesim frekansi=700.00 olmalidir

→ **cheb2ord komutu:**
Kullanımı: [n,Wn] = cheb2ord(Wp,Ws,Rp,Rs). Burada n, filtrenin derecesini, Wn, filtrenin kesim frekansını, Rp bant geçiren bölgede, Rs bant durduran bölgede dB cinsinden zayıflamayı, Wp, bant geçiren bölge köşe frekansını, Ws, bant durduran bölge köşe frekansını göstermektedir

Örnek: Bilginin 2000 Hz'de örneklendiği bir ortamda bant geçiren bölgenin köşe frekansı 600 Hz, bant durduran bölgenin köşe frekanslarının 500 Hz olduğu, bant geçiren bölge için 3 dB titreşime sahip, bant durduran bölge için 150 dB zayıflamaya sahip bir yüksek geçiren filtre tasarlayınız. Bu filtrenin derecesini ve kesim frekansını bulunuz.

```
fs=2000;
Ws = 500/(fs/2); Wp = 600/(fs/2);Rp=3;Rs=150;
[n,Wn] = cheb2ord(Wp,Ws,Rp,Rs);
Wn=Wn*fs/2;
clc
fprintf('Filtrenin minimum derecesi=%2d olmalidir\n',n);
fprintf('Filtrenin kesim frekansi=%5.2f olmalidir\n',Wn);
```

```
Filtrenin minimum derecesi=22 olmalidir
Filtrenin kesim frekansi=505.61 olmalidir
```

FIR Filtreler:

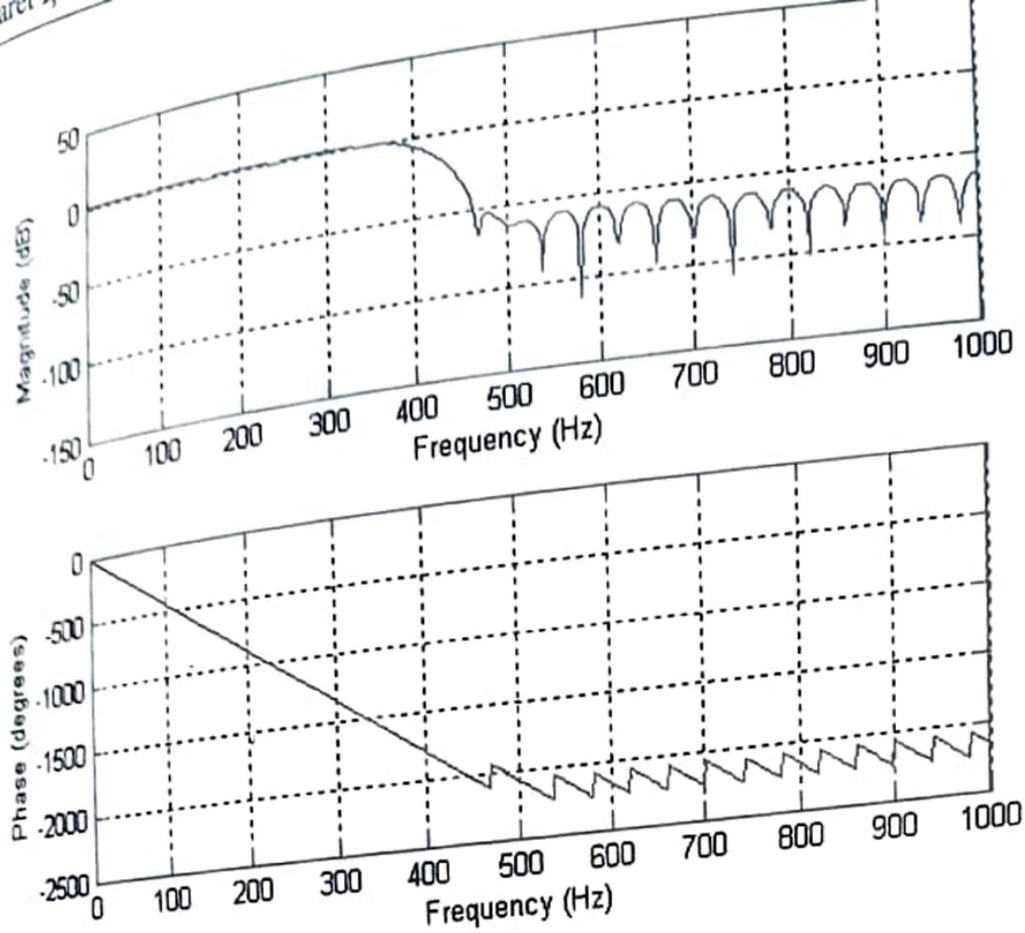
Avantajları: Tamamıyla doğrusal faza sahiptirler. Sürekli olarak dengededirler.
Dezavantajları: IIR filtreye göre daha yüksek derece gerektirir. Ayrıca gecikme miktarı IIR filtreye göre daha fazladır.

→ **Fir1 fonksiyonu:**

Kullanımı:
b = fir1(n,Wn,'tip') Wn normalize edilmiş kesim frekansıdır ve 0 ile 1 arası değişir. 1 nyquist frekansına denk gelir. N filtrenin derecesini gösterirken, tip ise filtrenin çeşidi ile ilgili bilgi verir. (Alçak geçiren, yüksek geçiren, bant geçiren veya bant durduran).

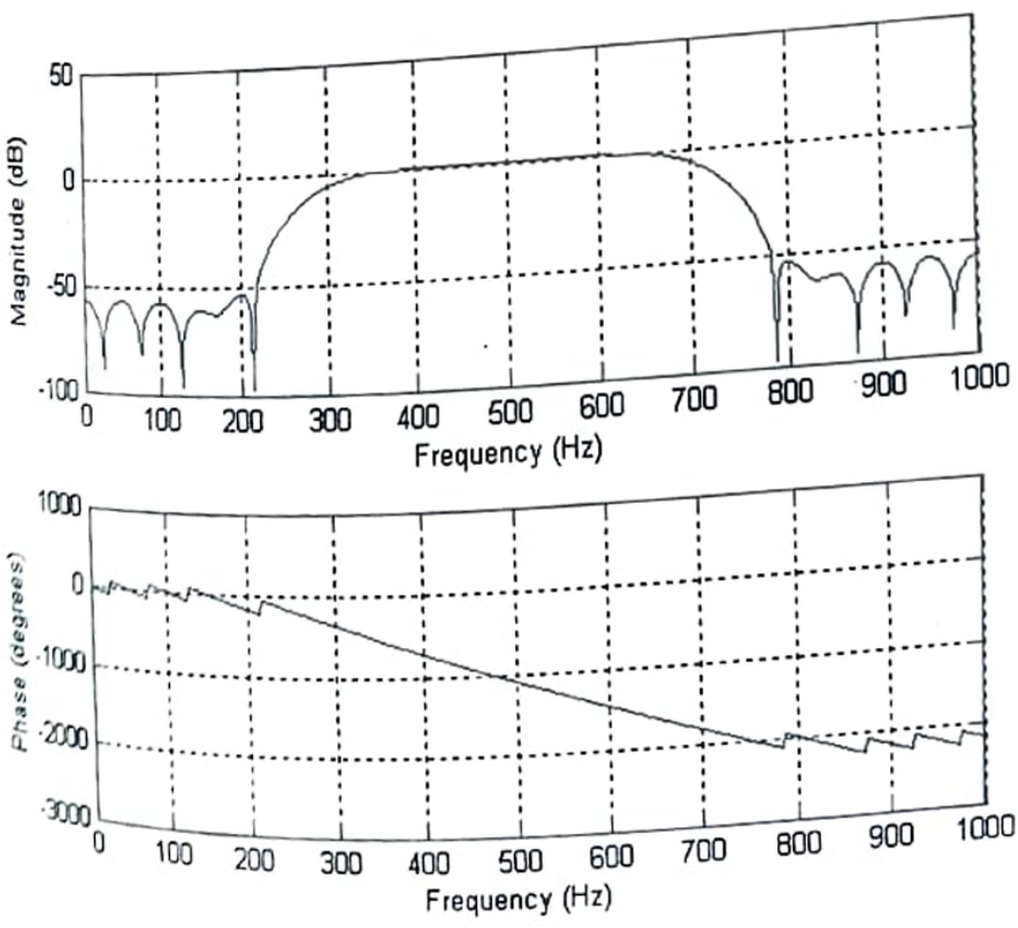
Örnek: Örnekleme frekansı 2 KHz, derecesi 50, kesim frekansı 400 olan bir fir1 alçak geçiren filtre tasarlayınız

```
fs=2000;n=50;fc=400;
b=fir1(n,fc/(fs/2));
freqz(b,1,512,fs);
```



Örnek: Örnekleme frekansı 2 KHz, derecesi 40, kesim frekansı 300 ve 700 olan bir fir1 bant geçiren filtre tasarlayınız

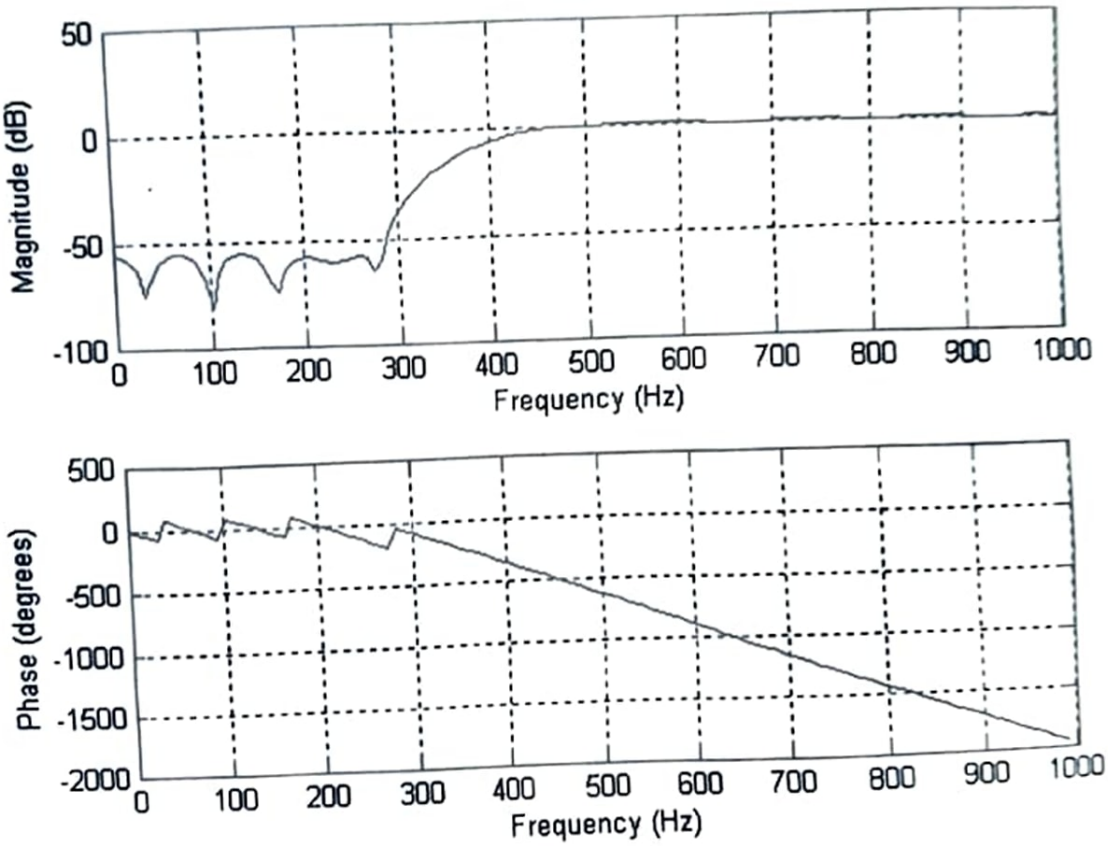
```
fs=2000;n=40;fc=[300 700];
b=fir1(n,fc/(fs/2));
freqz(b,1,512,fs);
```



b = fir2(n,f,m) . Burada n, [illegible] değerini gösterir ve 0 ile 1 arasında değişir. Burada 1 nyquist frekansına denk gelir. m ise belirtilen frekans noktalarına karşılık gelen genliktir.

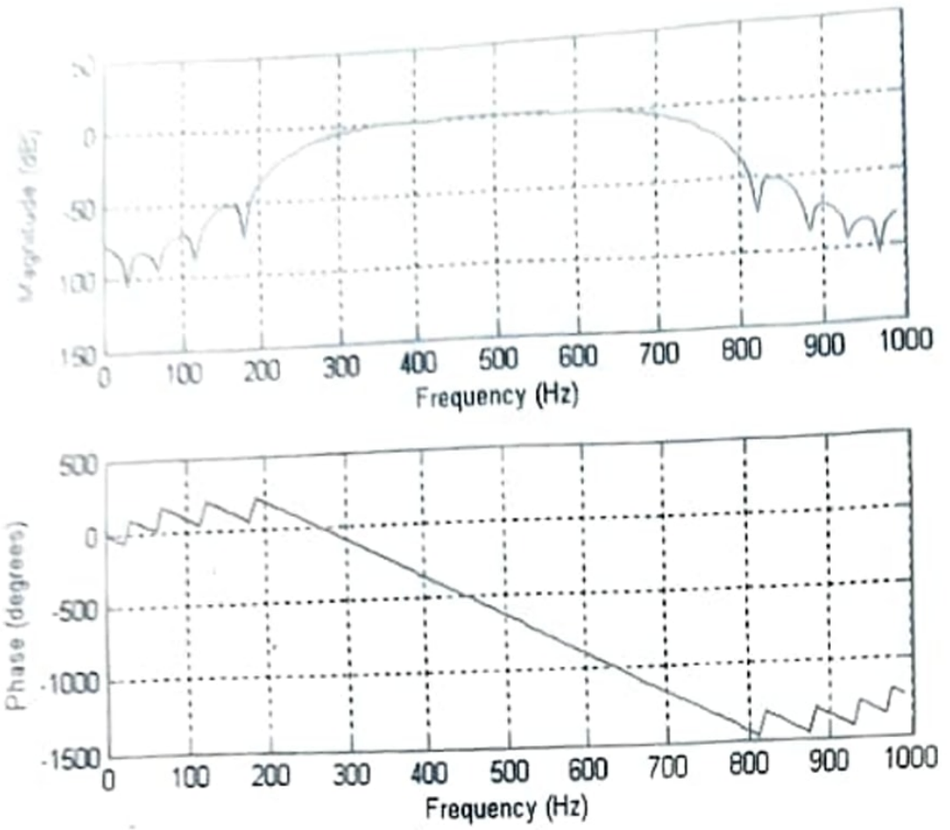
Örnek: Derecesi 30, örnekleme frekansı 2 KHz, kesim frekansı 400 Hz olan bir yüksek geçiren filtre tasarlayınız.

```
fs=2000,n=30,fc=400
f = [0 200 400 400   600 800 1000]; m = [0 0 0  1 1 1 1];
f=f/(fs/2);
b = fir2(n,f,m);
freqz(b,1,128,fs);
```



Örnek: Derecesi 30, örnekleme frekansı 2 KHz, kesim frekansı 300 ve 700 Hz olan bir bant geçiren filtre tasarlayınız.

```
fs=2000;n=30;
f = [0 100 200 300 300 400 500 600 700 700 800 900 1000];
m = [0  0   0  0  0   1   1   1   1   1   0   0   0   0];
f=f/(fs/2);
b = fir2(n,f,m);
freqz(b,1,128,fs);
```



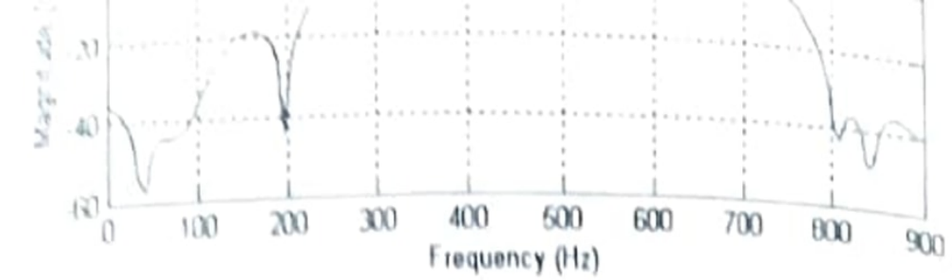
→Remez Fonksiyonu:

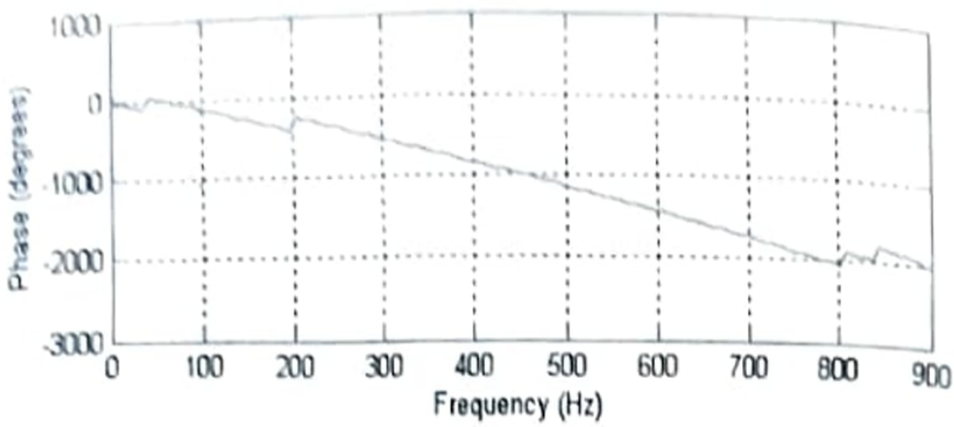
Kullanımı:

b = remez(n,f,a) . Burada n, filtrenin derecesini göstermektedir. f, frekans değerlerini gösterir ve 0 ile 1 arasında değişir. Burada 1 nyquist frekansına denk gelir. a ise belirtilen frekans noktalarına karşılık gelen genliktir. f ve a vektörlerinin uzunluğu çift olmalıdır.

Örnek: Örnekleme frekansı 1800 Hz, derecesi 30, kesim frekansı 300 ve 700 Hz olan bir bant geçiren filtreyi remez fonksiyonunu kullanarak tasarlayınız

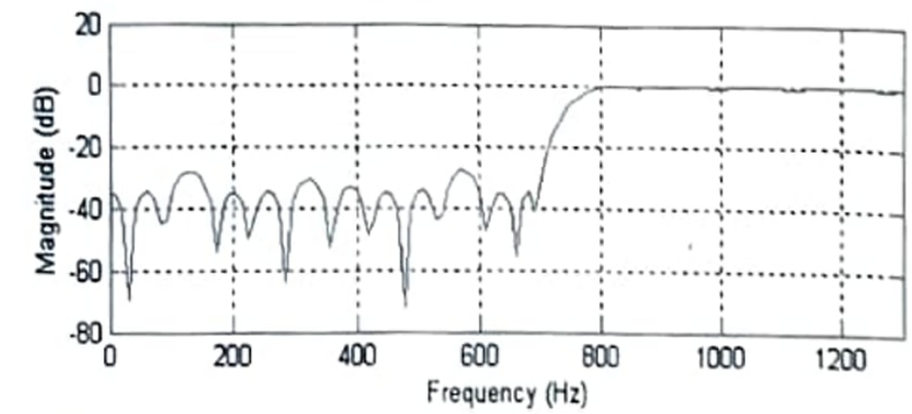
```
fs=1800;n=30;
f = [0 100 200 300 400 500 600 700 800 900];
a = [0  0 0   1 1  1  1  1 0  0  ];
f=f/(fs/2);
b = remez(n,f,a);
freqz(b,1,128,fs);
```

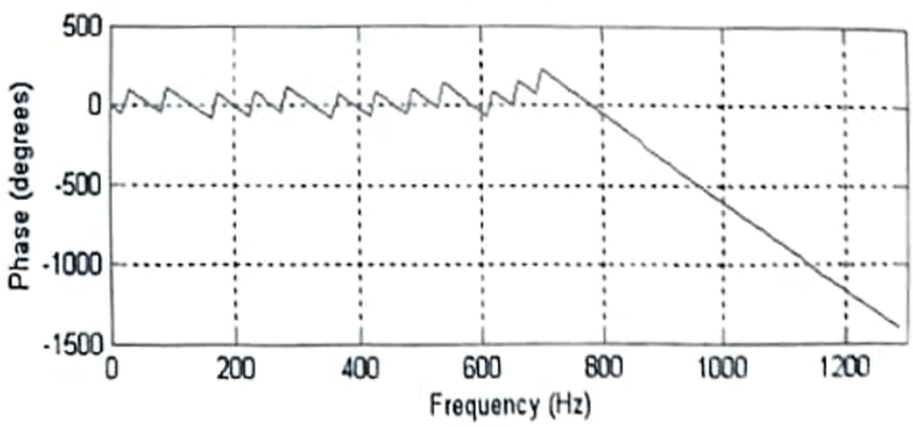




Örnek: Örnekleme frekansı 2600 Hz, derecesi 40, kesim frekansı 800 Hz olan bir yüksek geçiren filtreyi remez fonksiyonunu kullanarak tasarlayınız

```
fs=2600;n=40;
f = [0 100 200 300 400 500 600 700 800 900 1000 1100 1200 1300];
a = [0  0 0   0 0  0  0  0  0  1  1  1  1  1  1];
f=f/(fs/2);
b = remez(n,f,a);
freqz(b,1,128,fs);
```





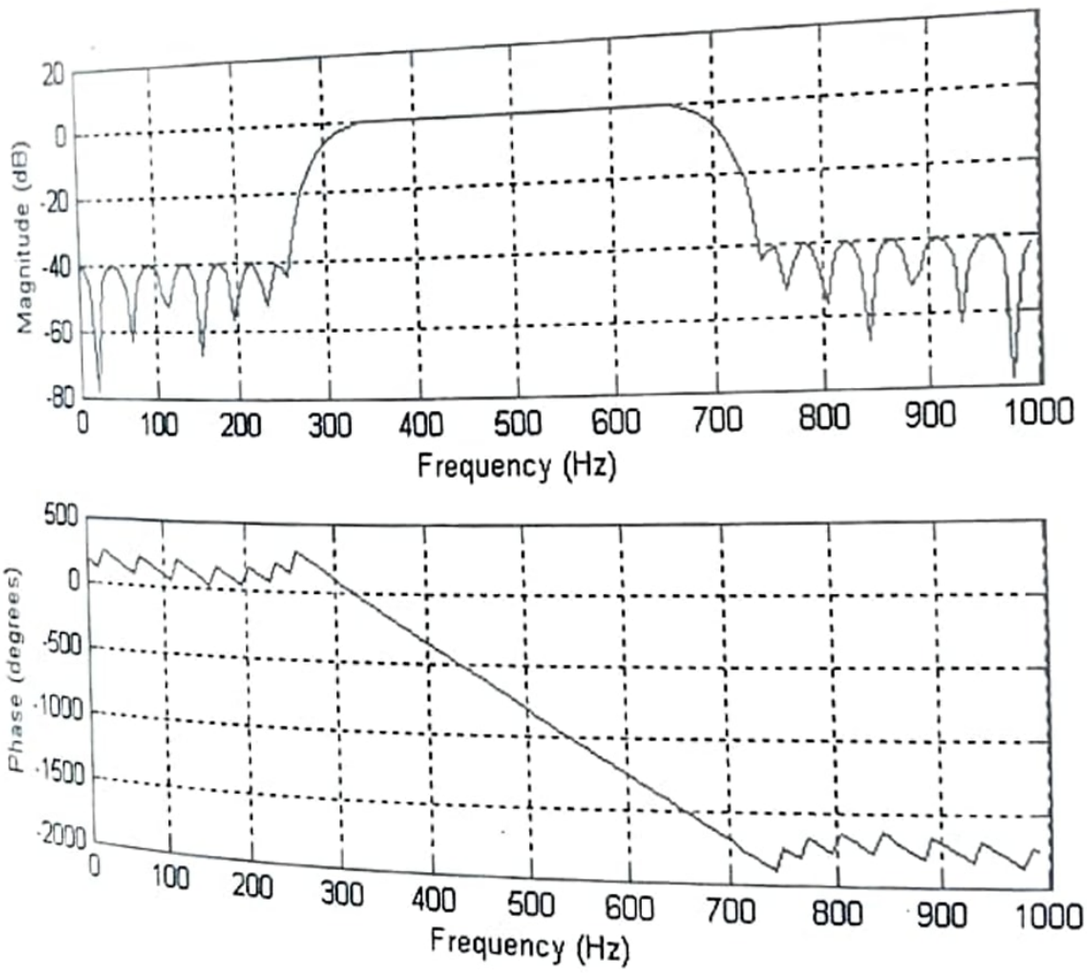
>>fircls fonksiyonu

Kullanımı: b = fircls(n,f,m,ust,alt)

Frekansları, m ise her frekans aralığına ait genlik değerlerini içeren vektörün uzunluğu, f vektörünün uzunluğundan 1 eksik olmalıdır. Her frekans aralığı için üst ve alt sınırları belirtir ve vektör uzunluğu, m vektör uzunluğuna eşit olmalıdır

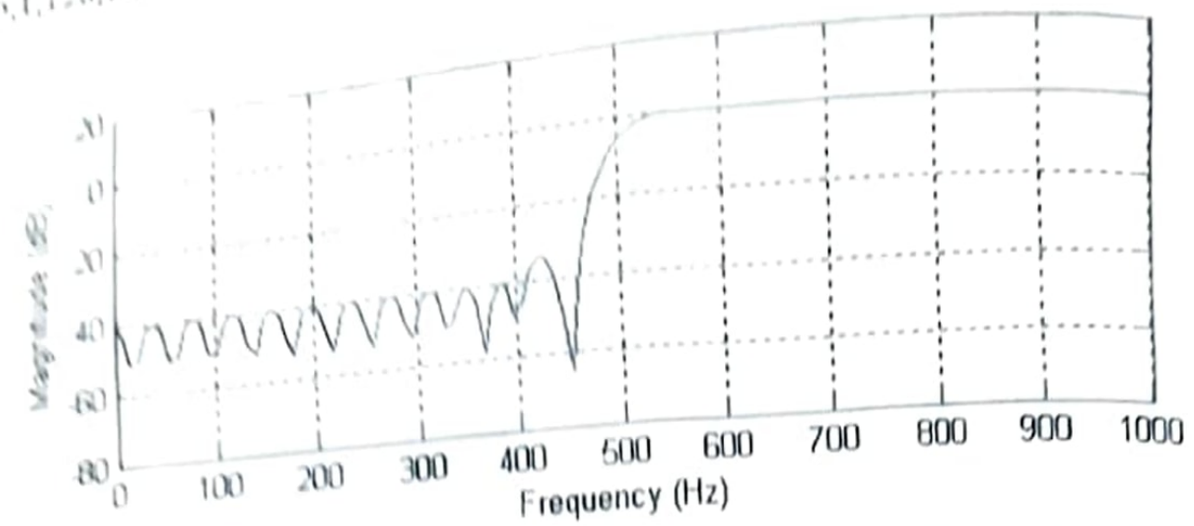
Örnek: Örnekleme frekansı 2000 Hz, derecesi 50, kesim frekansı 300 ve 700 Hz olan bir bant geçiren filtreyi fircls fonksiyonunu kullanarak tasarlayınız

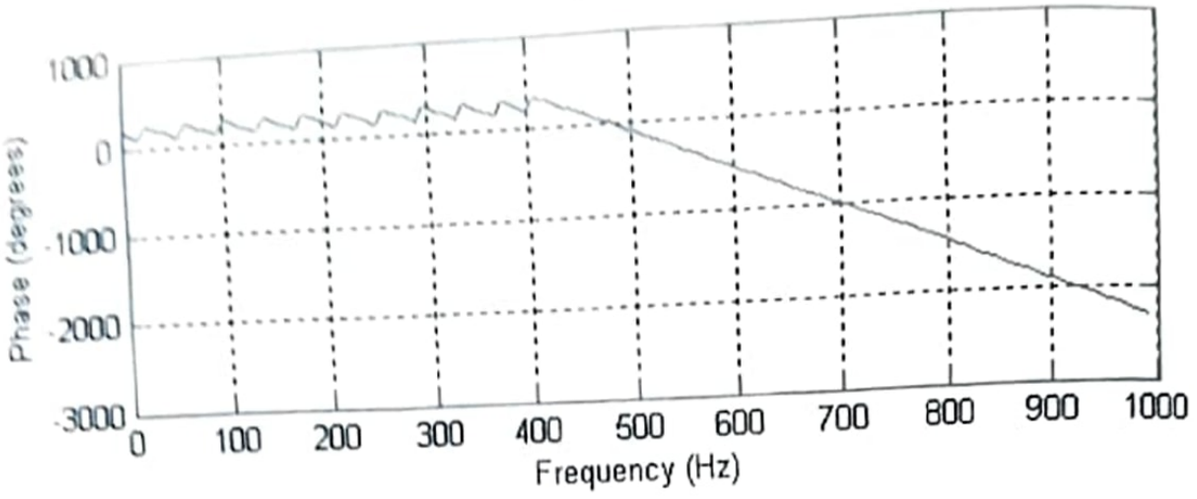
```
n=50;fs=2000;
f=[0 100 200 300 400 500 600 700 800 900 1000];
f=f/(fs/2);
m=[ 0  0  0  1  1  1  1  0  0  0];
ust=[0.01 0.01 0.01 1.02 1.02 1.02 1.02 0.01 0.01 0.01];
alt=ust-0.02;
b=fircls(n,f,m,ust,alt);
freqz(b,1,128,fs)
```



Örnek: Örnekleme frekansı 2000 Hz, derecesi 50, kesim frekansı 500 Hz olan bir yüksek geçiren filtreyi fircls fonksiyonunu kullanarak tasarlayınız

```
n=50;fs=2000;
f=[0 100 200 300 400 500 600 700 800 900 1000];
f=f/(fs/2);
m=[ 0  0  0  0  0  1  1  1  1  1];
ust=[0.01 0.01 0.01 0.01 0.02 1.01 1.01 1.01 1.01 1.01];
alt=ust-0.02;
```



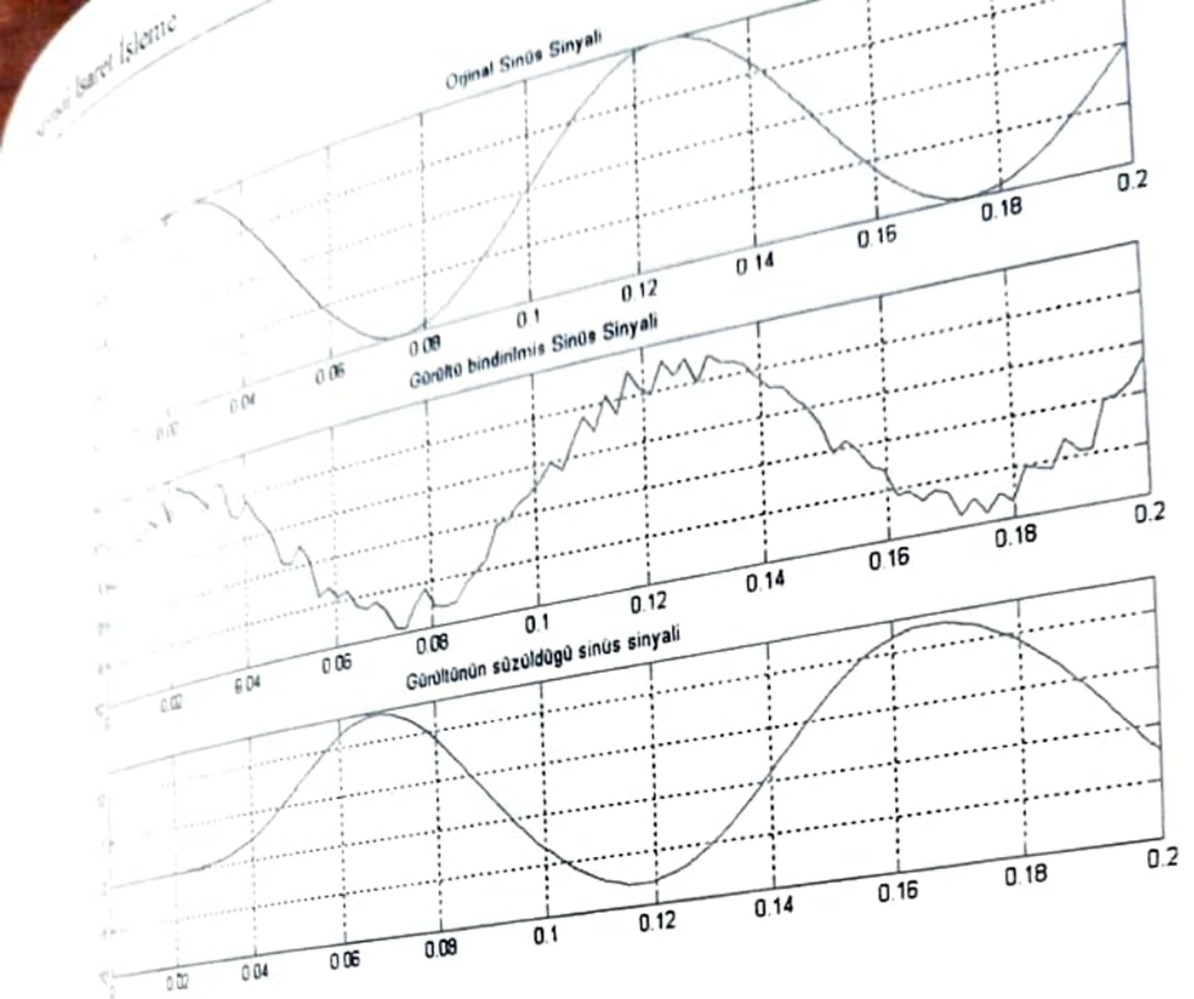


Filtre Uygulamaları

→**Filter komutu:** IIR veya FIR filtre kullanarak, sinyali filtreleme işlemine tabii tutar.

Örnek: 10 Hz frekansında bir sinüs sinyaline 4 genliğinde rasgele gürültü bindiriniz. Daha sonra fitler komutunu kullanarak bu gürültüyü süzünüz ve orijinal sinüs sinyalini elde edeniz.

```
f=10,fc=50;fs=1000;n=10;
t=linspace(0,1/5)
y1 = 10*sin(2*pi*f*t);
gurultu=4*rand(1,100)
y2=y1+gurultu
[b a]=butter(n,fc/(fs/2))
y3=filter(b,a,y2)
subplot(311),plot(t,y1);grid
title('Orjinal Sinüs Sinyali')
subplot(312),plot(t,y2);grid
title('Gürültü bindirilmis Sinüs Sinyali')
subplot(313),plot(t,y3);axis([0 0.2 -10 13]);grid
title('Gürültünün süzüldügü sinüs sinyali')
```

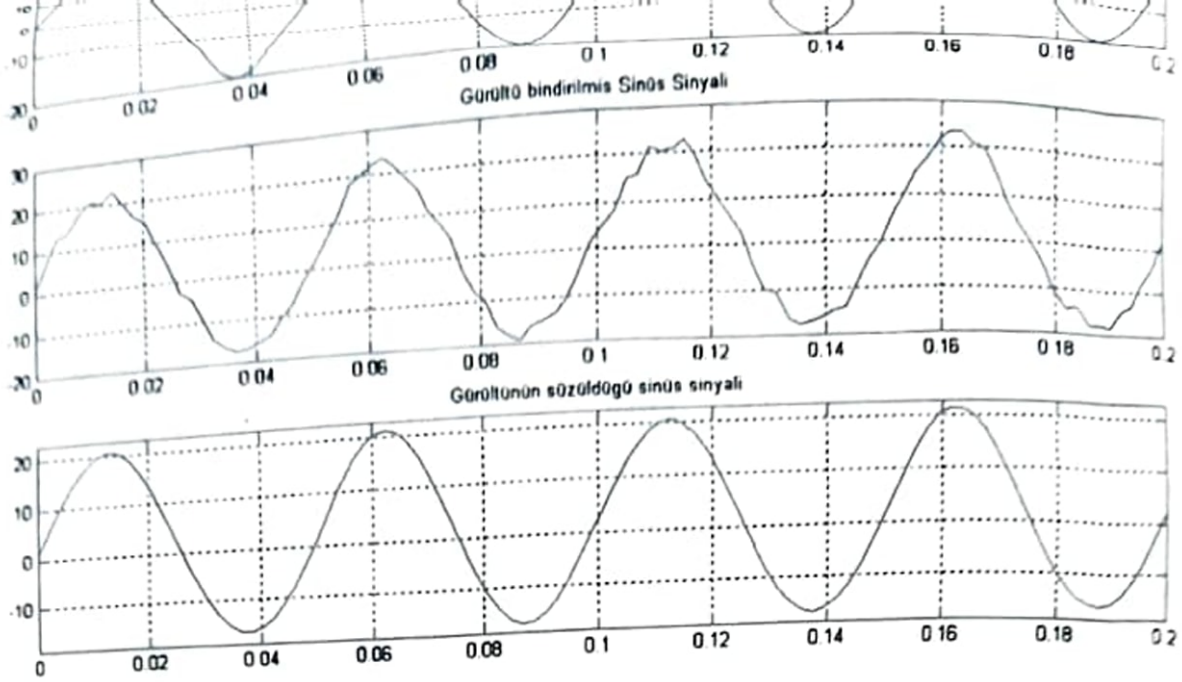


→filtfilt komutu:

kullanımı: y = filtfilt(b,a,x) komutu faz farkı olmaksızın filtreleme işlemi yapar

Örnek: 20 Hz frekansında bir sinüs sinyaline 4 genliğinde rasgele gürültü bindiriniz. Daha sonra filtfilt komutunu kullanarak bu gürültüyü süzünüz ve orijinal sinüs sinyalini faz farkı olmaksızın elde ediniz.

```
=20,fc=100;fs=1000;n=10;
=linspace(0,1/5)
 = 20*sin(2*pi*f*t);
gurultu=4*rand(1,100)
 =y1+gurultu
 ,a,=cheby1(n,0.5,fc/(fs/2))
 =filtfilt(b,a,y2)
subplot(311),plot(t,y1);grid
title('Orjinal Sinüs Sinyali')
subplot(312),plot(t,y2);grid
title('Gürültü bindirilmis Sinüs Sinyali')
subplot(313),plot(t,y3);axis([0 0.2 -19 23]);grid
title('Gürültünün süzüldügü sinüs sinyali')
```



→medfilt1 fonksiyonu:
kullanımı: y = medfilt1(x,n) komutu, x vektörüne n. dereceden median filtre uygular.

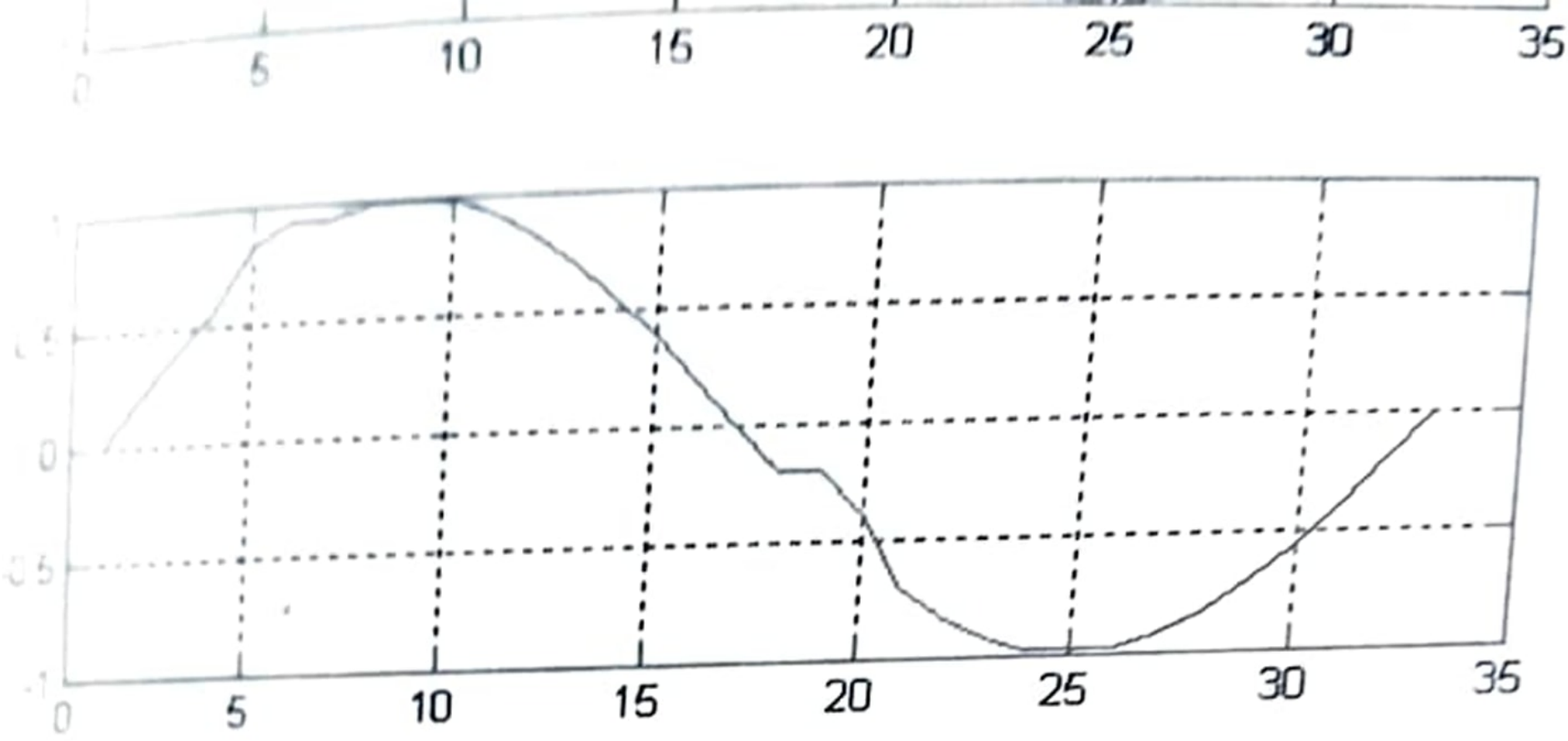
Örnek: Aşağıda verilen x vektörüne median filtre uygulayınız ve giriş ile çıkış grafiklerini karşılaştırınız.

```
x=[0 0.1951 0.3827 0.5556 1.1 0.8315 0.9239 0.9808 1 0.9808 0.9239 0.8315 0.7071...
0.5556 0.3827 0.1951 0 -0.1951 -0.3827 -0.1556 -0.7071 -0.8315 -0.9239 -0.9808 -1...
-0.9808 -0.9239 -0.8315 -0.7071 -0.5556 -0.3827 -0.1951 0]
```

Çözüm:

```
n=3;
x=[0   0.1951   0.3827   0.5556   1.1   0.8315   0.9239   0.9808...
  1.0000   0.9808   0.9239   0.8315   0.7071   0.5556   0.3827   0.1951...
  0.0000   -0.1951   -0.3827   -0.1556   -0.7071   -0.8315   -0.9239   -0.9808...
  -1.0000   -0.9808   -0.9239   -0.8315   -0.7071   -0.5556   -0.3827   -0.1951 0]
subplot(211),plot(x);
y=medfilt1(x,n)
subplot(212),plot(y);
```

Aşağıdaki grafikte görüldüğü gibi ilk grafikteki sinüs fonksiyonunda görülen farklılıklar, ikinci grafikte kısmen kaybolmuştur.



Dalga Biçimi Üretme

sawtooth komutu:

y=sawtooth(t,genislik)

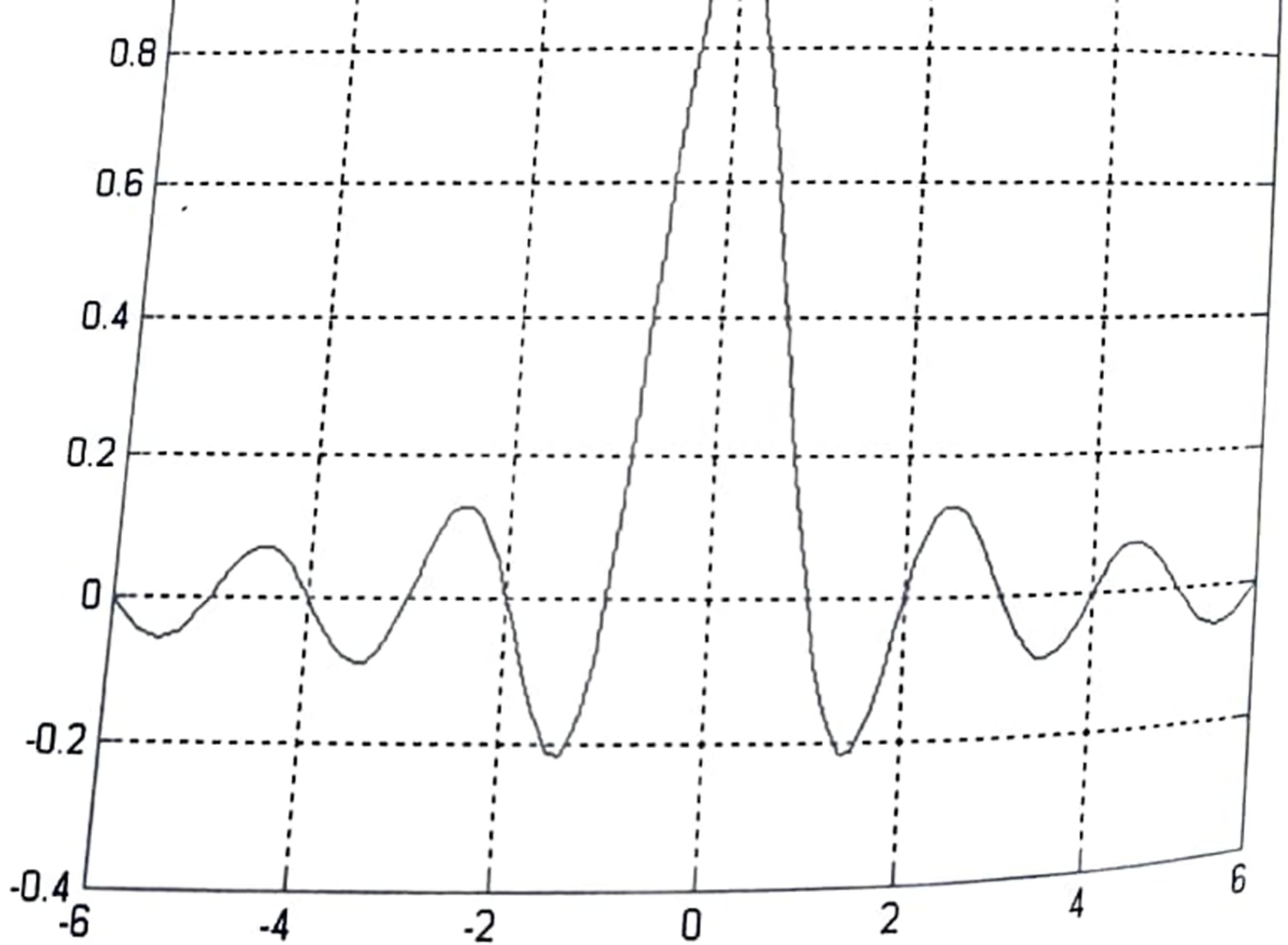
ile yükselme ve alçalma vaktini ayarlanabilen bir üçgen oluşturu

ile -1 arasında değişir ve üçgenin periyodu 2π'dir. Genişlik 0 ve 1 aras

-1'den +1'e 0 ile 2.genislik.. π süre aralığında yükselirken,

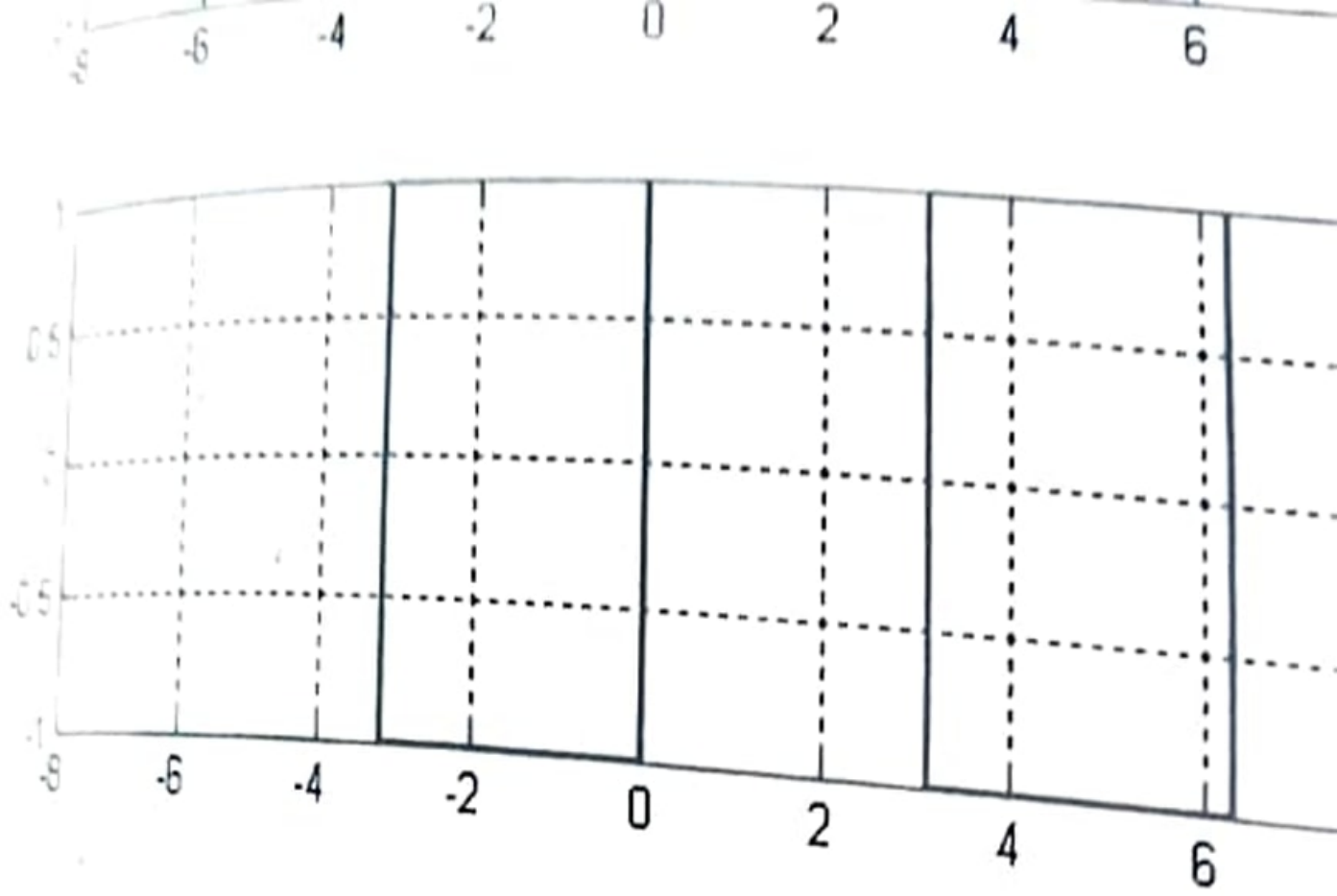
ile 2.π süresi arasında azalır.

```
*pi
(t,y1)
plot(t,y1),grid
(t,y2)
plot(t,y2),grid
```



→ square komutu:

Kullanımı: y=square(t). Bu komut periyodik olarak kare dalgalar üretir. Kare
periyodu 2π'dir. Bu komut sin fonksiyonuna benzer. Kare dalganın g



- tripuls komutu:

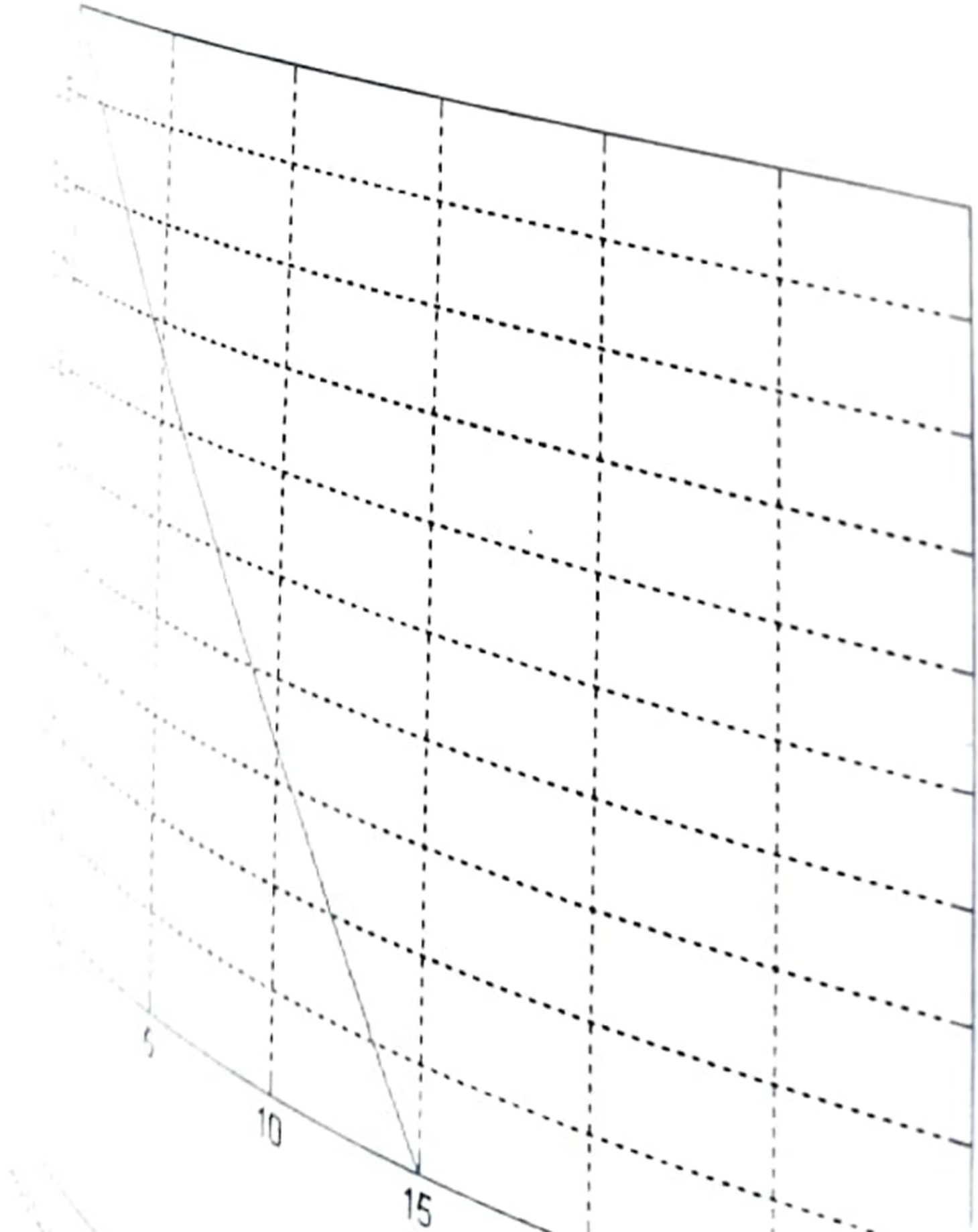
komut y = tripuls(t,g). Bu komut verilen bir dizi boyunca istenilen genişlik
yüksekliği 1'dir. Üçgen 0 merkezlidir ve periyodik değildir.

örnek:

t=-10:0.1:10;

y=tripuls(t,30);

plot(t,y)grid

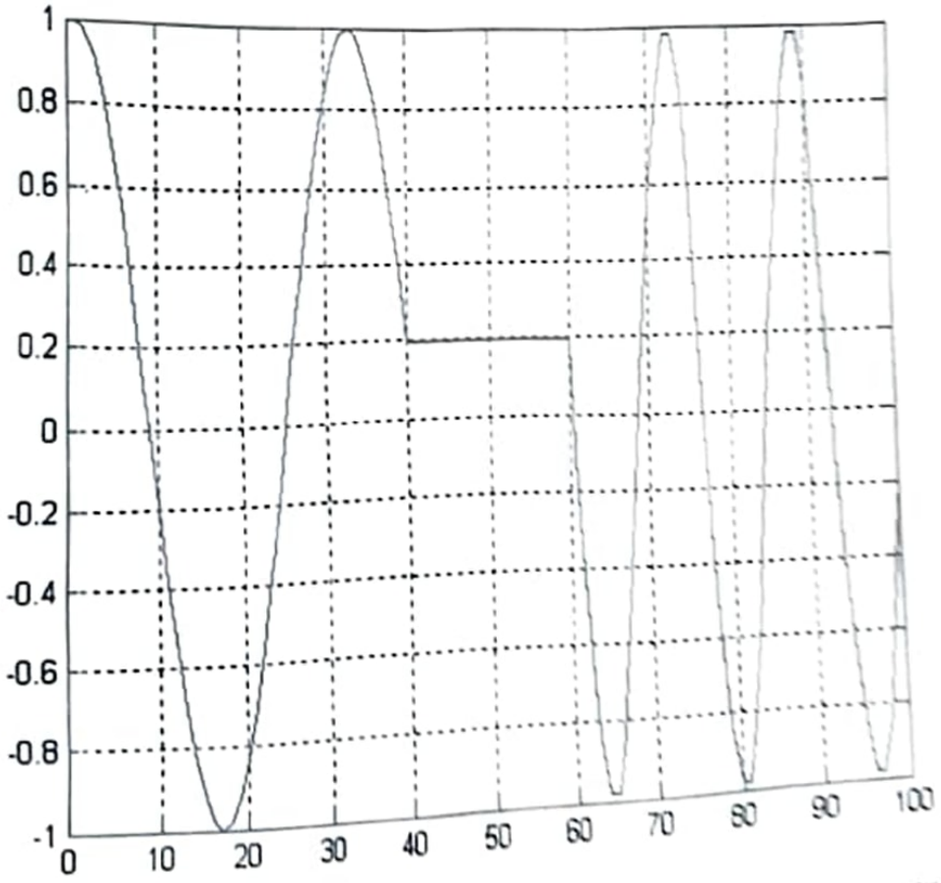


→ **vco komutu:**

Kullanımı: y = vco(x,f_s,f_c). Bu komut verilen x, f_s ve f_c değerlerine göre kosinüs sinyali üretir. x, -1, 0, +1 değerlerini alır. fs örnekleme frekansı ve fc referans frekansıdır. x sıfıra eşit iken, y, fs Hz ile örenklenmiş fc Hz'lik bir kosinüs sinyalidir. x=-1 ise çıkış 0, x=+1 ise çıkış 2fc'dir.

Örnek:

```
x1=zeros(1,40);
x2=-1*ones(1,20);
x3=ones(1,40);
x=[x1 x2 x3];
fs=1/4,fc=1/128;
y=vco(x,fc,fs);
plot(y),grid
```



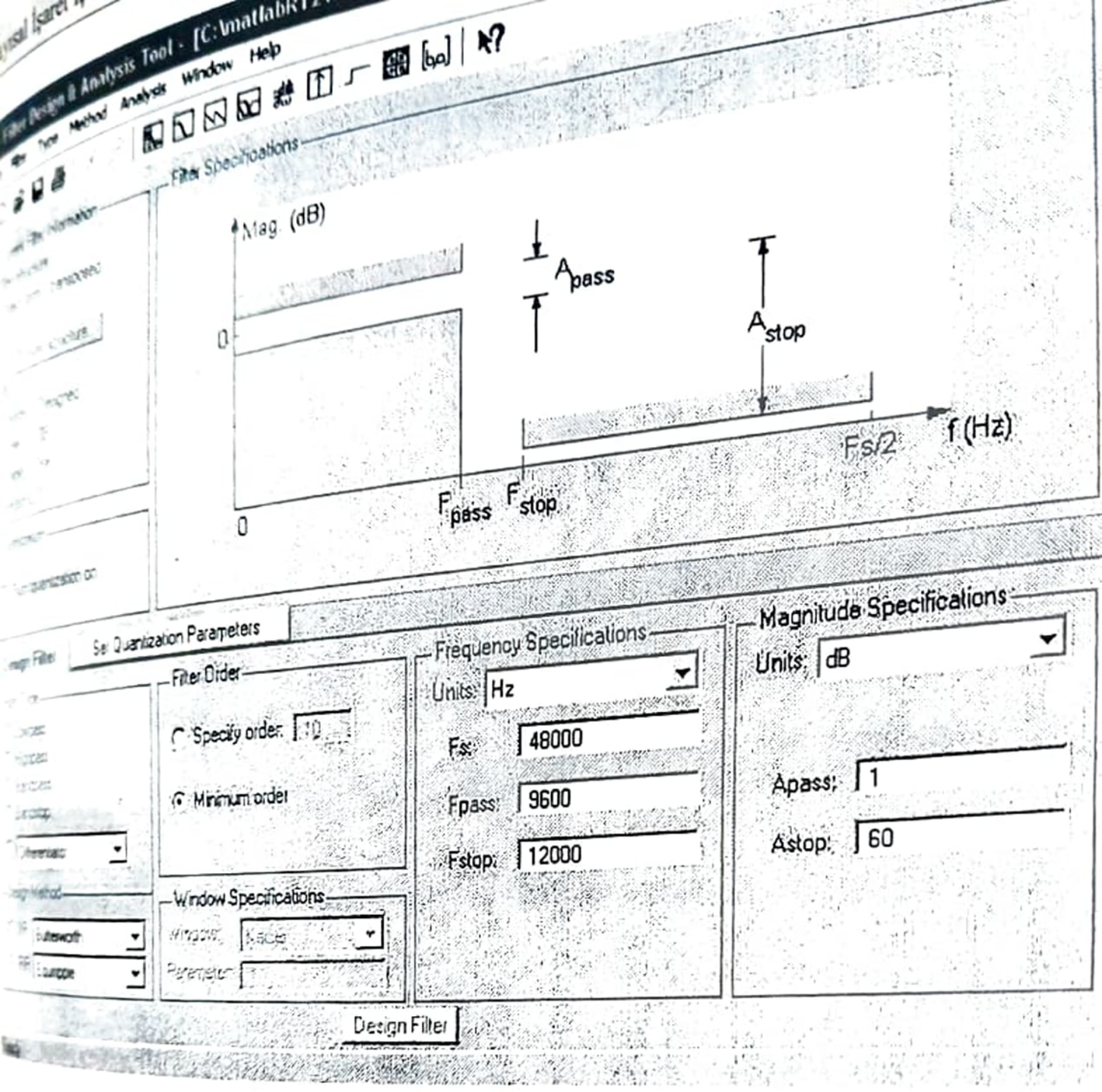
Sinyal İşleme Toolbox'ı İçinde Yer Alan Grafiksel Kullanıcı Ara Yüzleri

Matlab'de Sinyal İşleme Toolbox'ı içinde iki tana Grafiksel kulanı ara yüzü bulunur. Bunlar

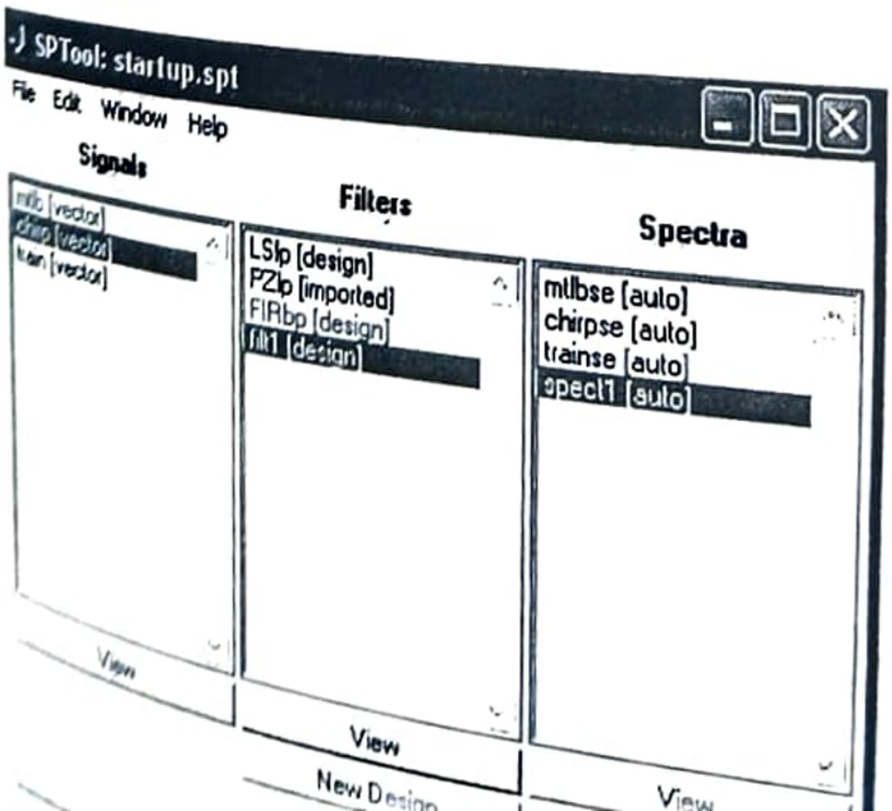
a)Filtre Tasarım ve Analiz Aleti. (fdatool)

b) İnteraktif sayısal işaret işleme aleti.(sptool)

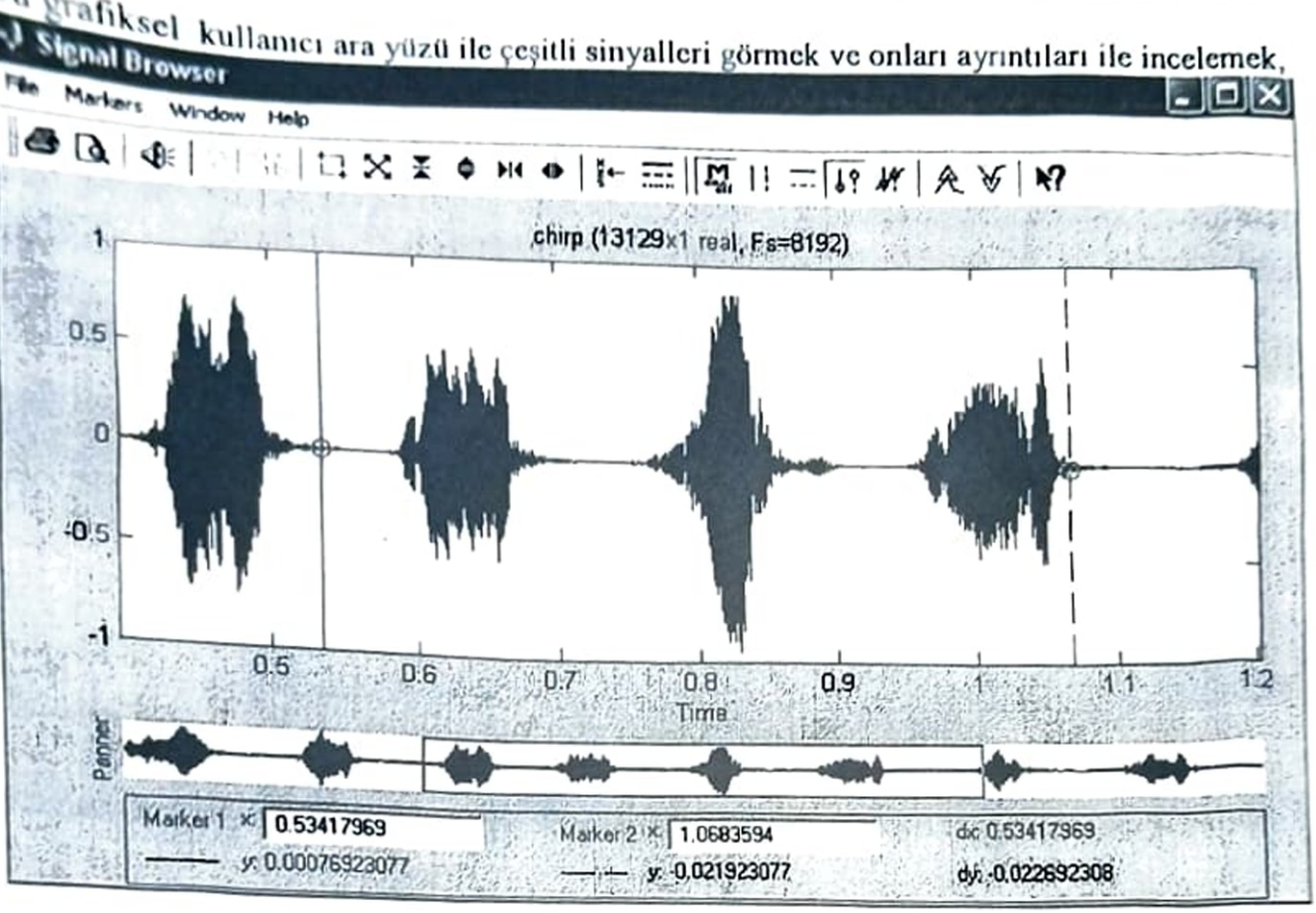
Filtre Tasarım ve Analiz Aleti: Bu aleti çağırmak için komut penceresine gidilerek fdatool yazılıp enter tuşuna basılır. Bu durumda aşağıdaki grafik ekrana gelir. Bu grafiksel kullanıcı ara yüzünde IIR ve FIR filtreler tasarlamak, Bu filtrelerin genliğini ,açısını ve genlik ve mümkündür. Ayrıca, dürtü ve birim basamak cevabını, kutup-sıfır katsayılarını bulmak mümkündür.



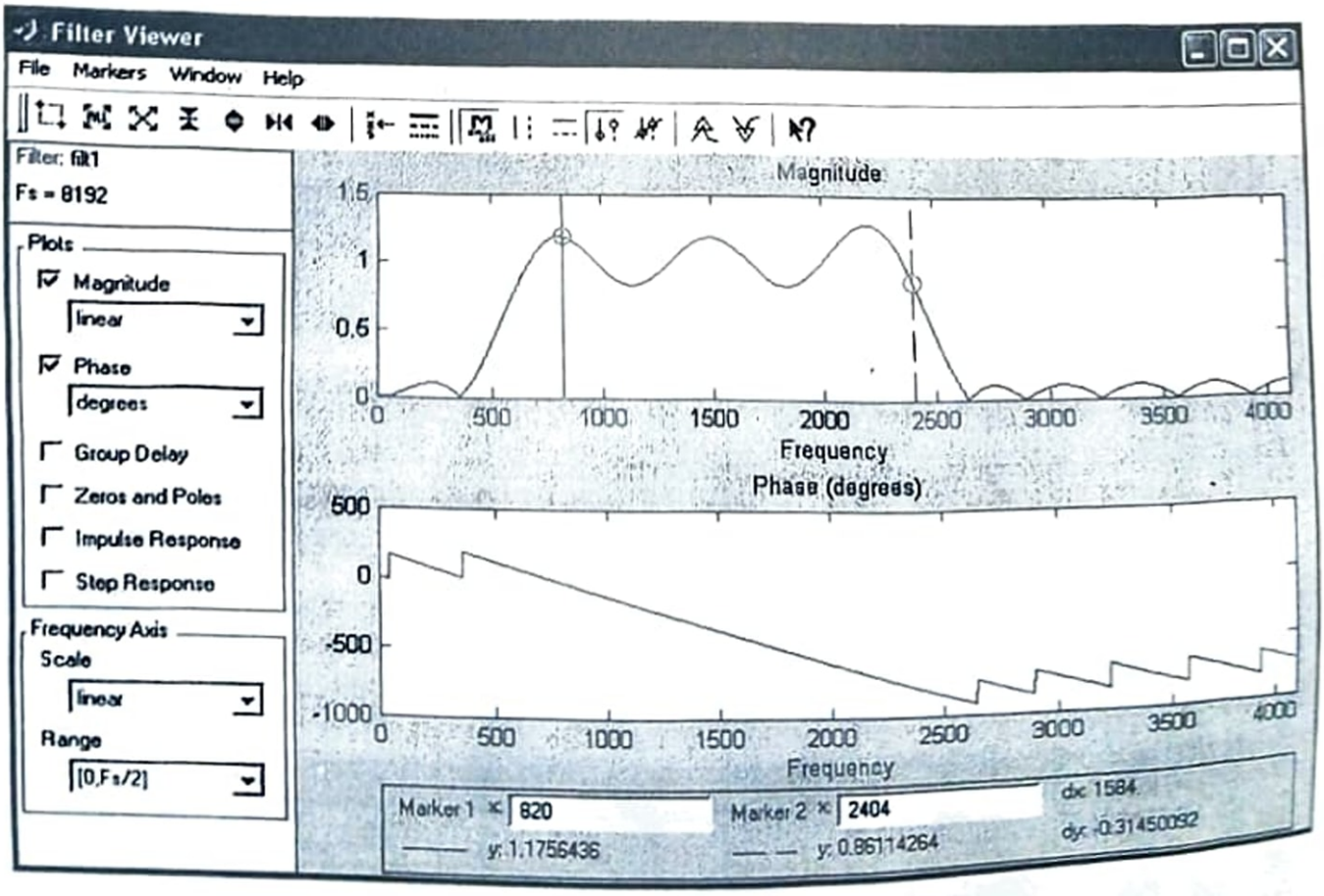
İnteraktif Sayısal İşaret işleme Aleti: Bu aleti çağırmak için komut penceresine gidilerek sptool yazılır ve enter tuşuna basılır. Bu durumda aşağıdaki grafik ekrana gelir.



Bu grafiksel kullanıcı ara yüzü ile çeşitli sinyalleri görmek ve onları ayrıntıları ile incelemek,

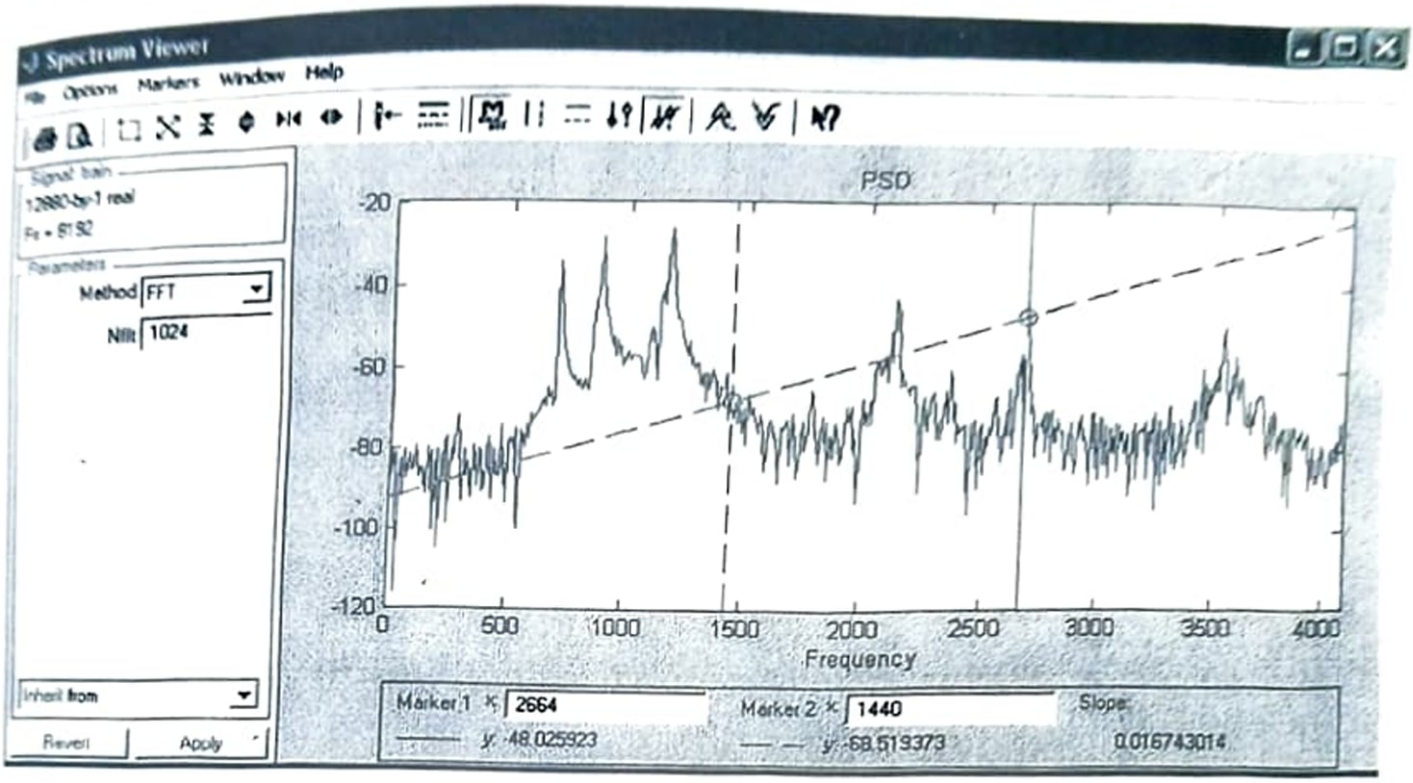


Hazır filtre yapılarına görmek ve yeni filtre tasarımı eklemek,



Sayısal İş

Sinyallerin spektrumlarını görmek mümkündür.



OTOMATİK KONTROL SİSTEMLERİ

SİSTEM MODELLEME

→Doğrusal Modeller:

- Transfer Fonksiyonu:

$$H(s) = \frac{s+2}{s^2+4s+3}$$

- Sıfır / Kutup / Kazanç:

$$H(z) = 2\frac{(s+1)(s-2)}{(s-1+j)(s-1-j)}$$

- Durum Uzayı :

$$\frac{dx}{dt} = Ax + Bu$$
$$y = Cx + Du$$

→ Modellerin oluşturulması:

- Transfer Fonksiyonu:

```
s = tf('s')
transfon= (s+2)/(s^2+10*s+20)
veya
transfon=tf([1 2],[1 10 20])

Transfer function:
    s + 2
-------------
s^2 + 10 s + 20
```

- Sıfır / Kutup / Kazanç:

```
skk=zpk(-1,[-2 -1 0.5], 2)

Zero/pole/gain:
    2 (s+1)
---------------
(s+2) (s+1) (s-0.5)
```

- Durum Uzayı:

```
R= 2.0
L1= 0.5
L2 =0.4
A = [-R/L1 -R/L1;-R/L2 -R/L2];
B = [-1; 0];
C = [R R;-R R];
D = [1;0];
states = {'akım1' 'akim2'};
inputs = {'besleme'};
outputs = {'gerilim1' 'gerilim2'};
ss(A,B,C,D,'statename',states,'inputname',inputs,'outputname',outputs)
```

```
a =
                akim1      akim2
   akim1           -4         -4
   akim2           -5         -5
b =
                besleme
   akim1              -1
   akim2               0
c =
                akim1      akim2
   gerilim1            2          2
   gerilim2           -2          2
d =
                besleme
   gerilim1            1
   gerilim2            0

Continuous-time model.
```

→ Modeller Arasında Çevirim Yapma:

- Durum uzayından transfer fonksiyonuna geçiş.

Örnek:

Yukarıda gösterilen durum uzayından transfer fonksiyonuna geçiş yapalım.

```
>> tf(q)

Transfer function from input "besleme" to output...

                s^2 + 13 s + 40
gerilim1:  ---------------
                  s^2 + 9 s

                    4 s
gerilim2:  ---------
                  s^2 + 9 s
```

- Durum uzayından sıfır/kutup/kazanç fonksiyonuna geçiş.

Örnek:

```
>> zpk(q)

Zero/pole/gain from input "besleme" to output...
           (s+8) (s+5)
gerilim1:  -----------
              s (s+9)

              4 s
gerilim2:  -------
           s (s+9)
```

→ Doğrusal Modeller Arasında Bağıntılar:

-Toplama:

Örnek: $\frac{1}{s} + \frac{s+1}{s^2+3s+1} + \frac{1}{s+4}$

Çözüm:

```
>> tf(1,[1 0]) + tf([1 1],[1 3 1])+tf(1,[1 4])

Transfer function:
3 s^3 + 15 s^2 + 18 s + 4
-------------------------
s^4 + 7 s^3 + 13 s^2 + 4 s
```

-Çarpma:

Örnek: $\frac{s+2}{s^2+3s+1} * \frac{s}{2+s} = ?$

Çözüm: tf([1 2],[1 3 1])*tf(1,[1 2])

```
>> tf([1 2],[1 3 1])*tf(1,[1 2])

Transfer function:
    s + 2
-----------------
s^3 + 5 s^2 + 7 s + 2
```

-Geri Besleme Bağlantısı:

Örnek:

```
>> feedback(tf(1,[1 0]),tf(1,[1 3]))

Transfer function:
  s + 3
-----------
s^2 + 3 s + 1
```

Kökün Geometrik Yeri (Root Locus):

Transfer fonksiyonu verilmiş olan bir sistemin köklerinin geometrik yeri şu komut ile çizilir:

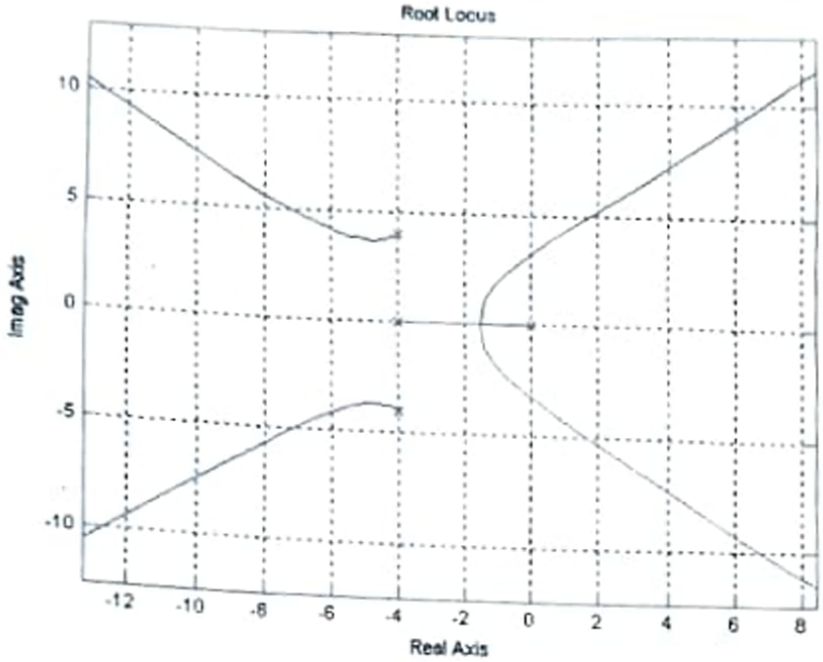
```
rlocus(sistem)
```

Örnek: Birim geri beslemeli sistemde, öne doğru olan transfer fonksiyonu:

$H(s) = \frac{K}{s(s+4)(s^2+8s+32)}$ şeklinde verildiğine göre $K \geq 0$ için karakteristik denklemin köklerinin geometrik yerini gösteriniz.

Çözüm

```
h=1/(tf([1 0],1)*tf([1 4],1)*tf([1 8 32],1))
rlocus(h),grid
```

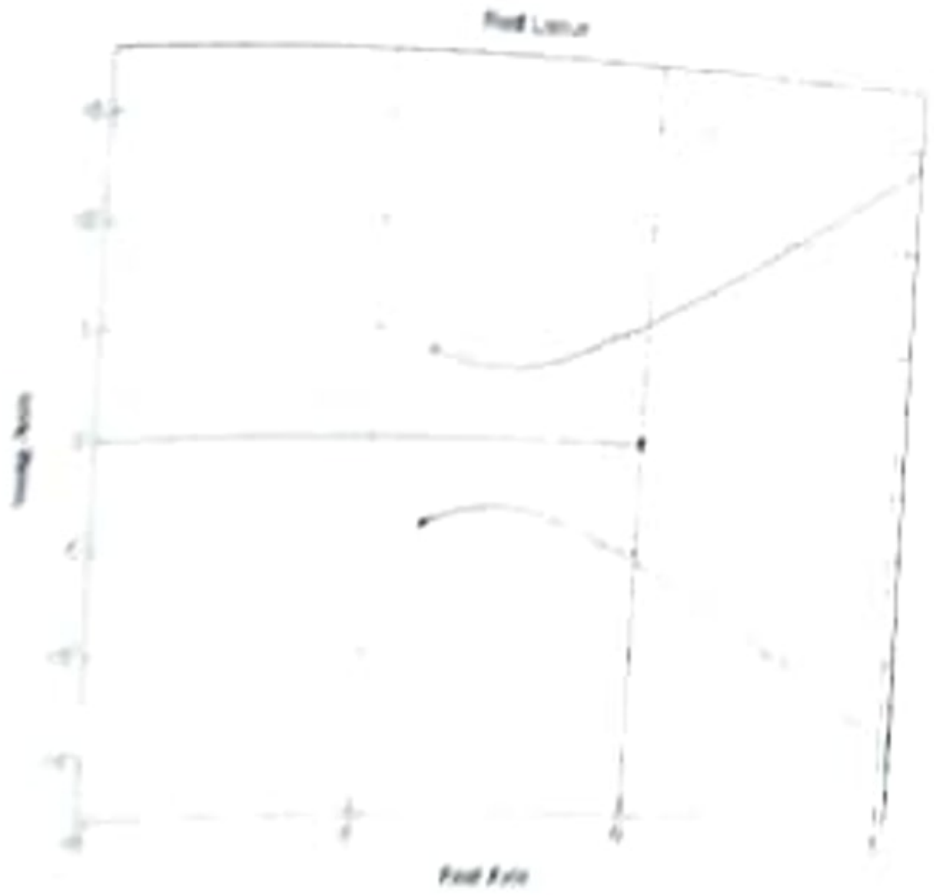


Örnek 2:

Açık çevrim transfer fonksiyonu $H(s) = \dfrac{K\left(\frac{s}{10}+1\right)}{s^2\left(\frac{s}{100}+1\right)\left(\frac{s}{500}+1\right)}$ olarak verilen sistem için köklerin geometrik yeri (K>0) için çiziniz.

Çözüm:

```
s=tf('s');
h=1/(s*((s+4)^2+16))
rlocus(h),grid
```



Bode Diyagramı

Bode diyagramında, [illegible] sinüzoidal frekansla [illegible] bir sistemin frekans cevabı, genlik ve faz olarak [illegible] çizilir. Genlik dB cinsinden, faz derece olarak çizilir. [illegible] diyagramda [illegible] bu frekans frekans parametresi olarak ve yüksek frekanslı [illegible] bu iki grafiğin [illegible]

Bode [illegible] diyagramları [illegible] genişletmek avantajlıdır. Çünkü pratikteki [illegible] bode [illegible] [illegible] bode [illegible]

[illegible] [illegible] [illegible] [illegible] bode [illegible]

[illegible] [illegible] [illegible] [illegible] [illegible] bode diyagramını çizin.

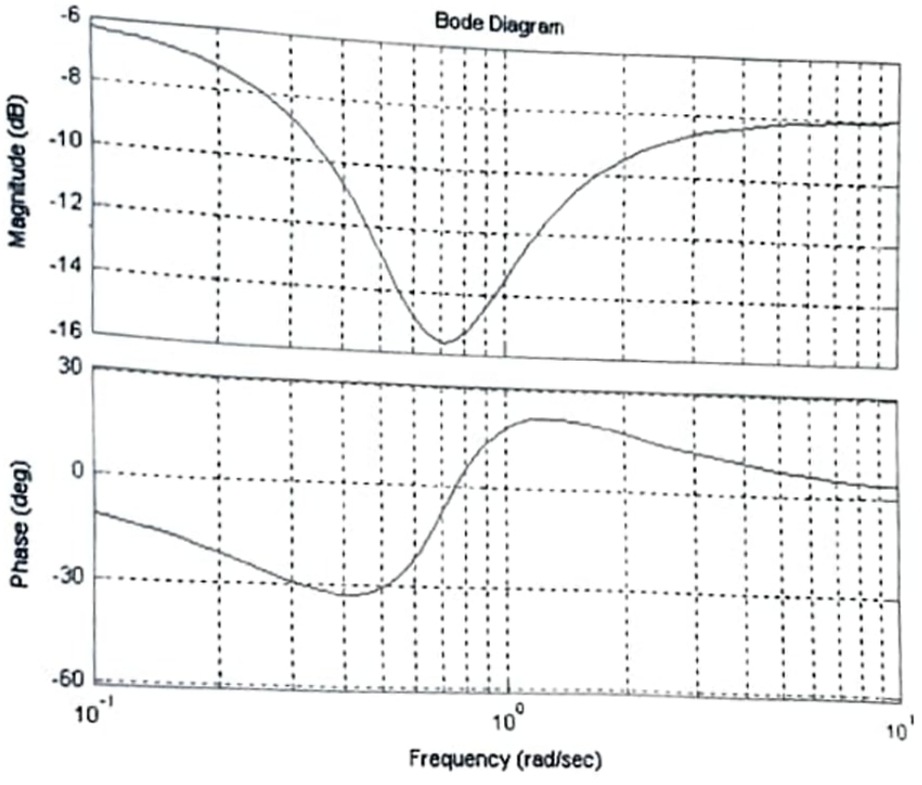
[illegible]



Örnek 2: $h(s) = \frac{2s^2 + s + 1}{5s^2 + 6s + 2}$ olduğuna göre bu sistemin bode diyagramını çiziniz.

Çözüm:

```
s=tf('s');
h=(2*s^2+s+1)/(5*s^2+6*s+2);
bode (h),grid
```

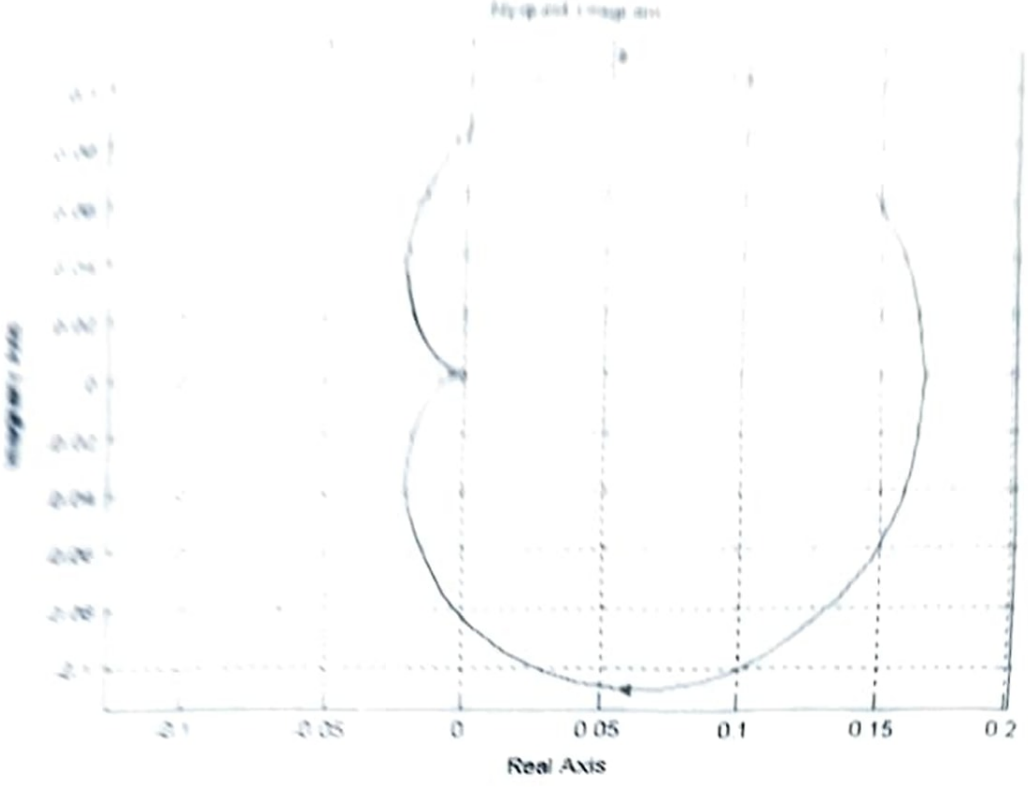


Nyquist Diyagramı

DZD (Doğrusal Zamanla Değişmeyen) sistemlerin frekans cevabını gösterir. Nyquist çizimleri; kazanç sınırı, faz sınırı ve denge gibi özellikleri analiz etmek için kullanılır.

Örnek1: Açık çevrim transfer fonksiyonu $H(s) = \frac{1}{(s+2)(s+3)}$ olan sistemin Nyquist diyagramını çiziniz.

Çözüm:



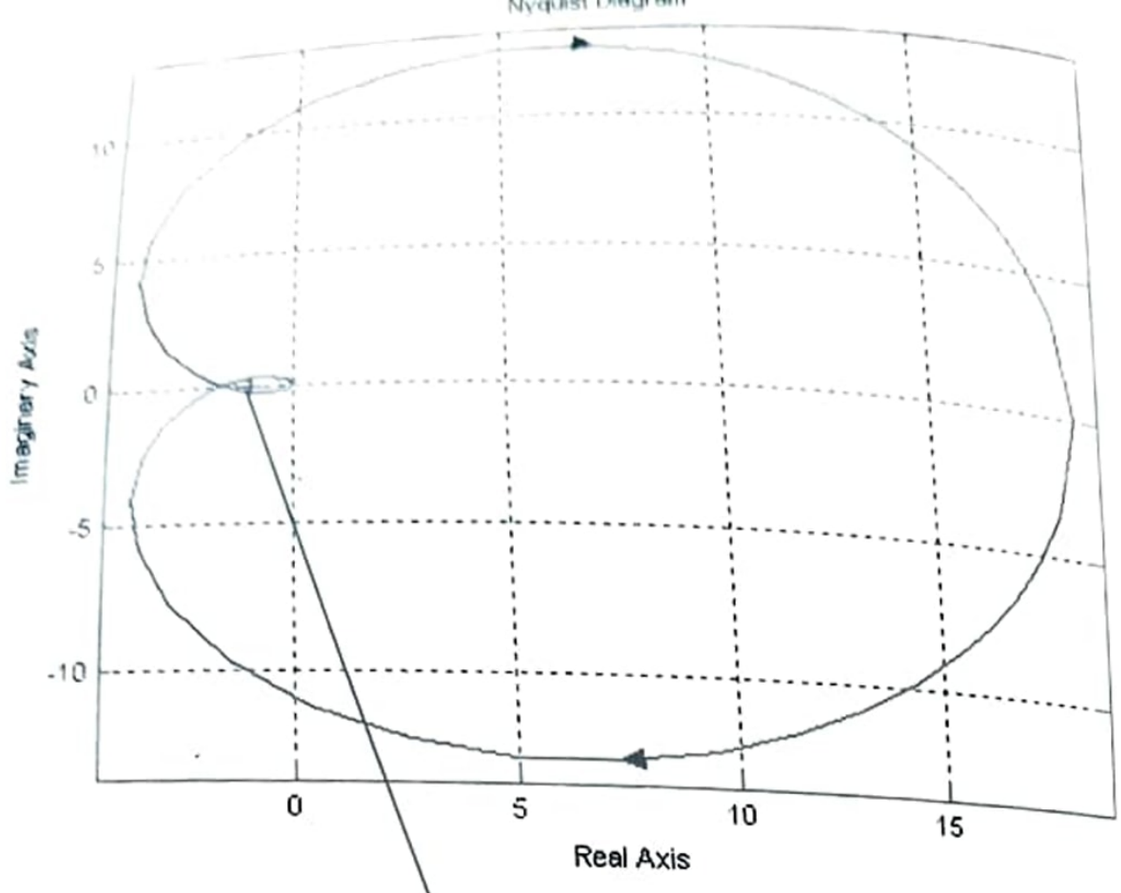
[illegible] gibi -1 noktası çevrilmediği için sistem dengededir.

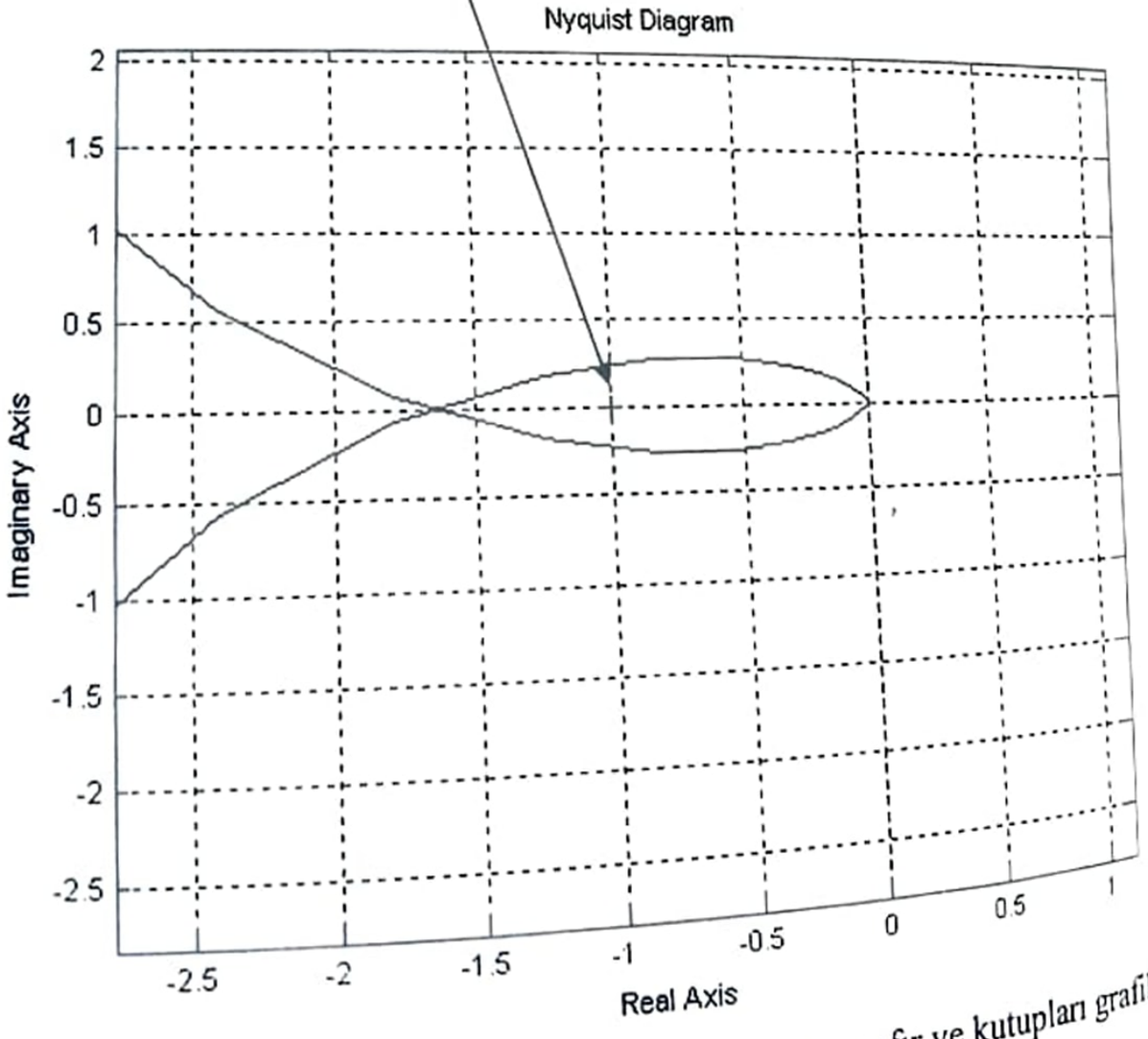
[illegible] Açık çevrim transfer fonksiyonu $H(s) = \dfrac{1}{(s+4)(s+2)(s+1)}$ olan sistemin Nyquist

[illegible]

```
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
```

[illegible] gibi -1 noktası çevrildiği için sistem denge değildir..



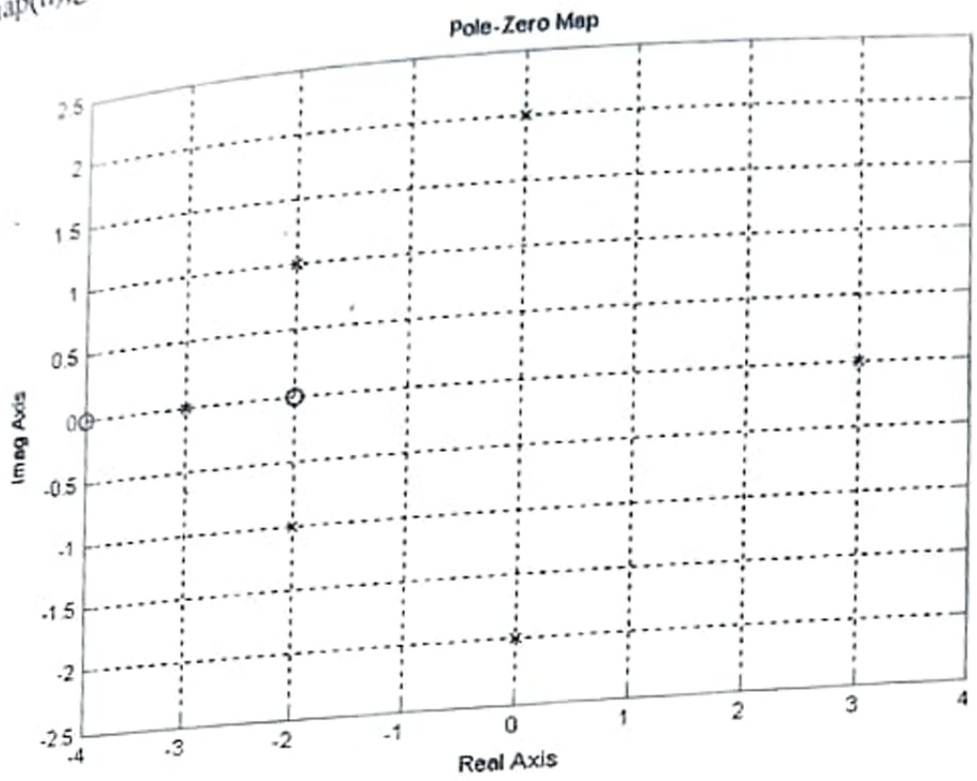


→pzmap komutu: Bu komut ile transfer fonksiyonunun sıfır ve kutupları grafik ekseninde gösterilir.

Örnek 1: $H(s) = \dfrac{(s+2)(s+4)}{(s^2+4)(s^2-9)(s^2+4s+5)}$ fonksiyonunun sıfır ve kutuplarını grafik ekranında gösteriniz.

Çözüm:

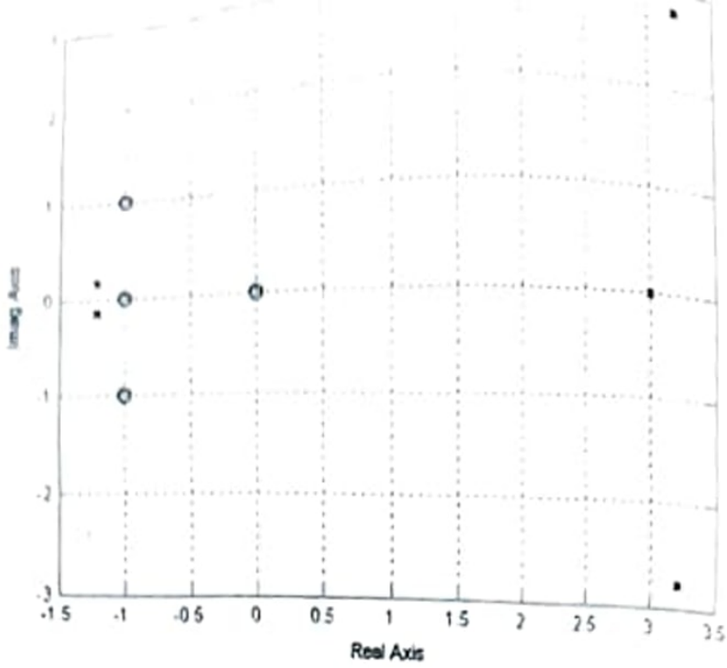
```
s=tf('s')
h=(s+2)*(s+4)/(s^2+4)/(s^2-9)/(s^2+4*s+5)
pzmap(h),grid
```



Örnek 2: $H(s) = \dfrac{s^4+3s^3+4s^2+2s}{s^5-7s^4+16s^3+22s^2-75s-80}$ fonksiyonunun sıfır ve kutuplarını grafik ekranında gösteriniz.

Çözüm:

```
clc
s=tf('s')
pay=[1 3 4 2 0]
payda=[1 -7 16 22 -75 -80]
h=tf(pay,payda)
pzmap(h),grid
```



→ **residue:** Bu komut verilen transfer fonksiyonunu kısmi kesirlere ayırır.

Örnek1:

$\frac{s+2}{s(s+3)(s+5)}$ kesrini kısmı kesirlere ayıralım.

```
pay=[1 2]
payda=[1 8 15 0]
[r,p,k]=residue(pay,payda)

r =
   -0.3000
    0.1667
    0.1333
p =
   -5
   -3
    0
k =
    []
```

Bu durumda yukarıdaki fonksiyonu kısmi kesirlere ayrılmış şekilde aşağıdaki gibi yazabiliriz.

$$\frac{s+2}{s(s+3)(s+5)} = \frac{-0.3}{(s+5)} + \frac{0.167}{(S+3)} + \frac{0.133}{S}$$

Örnek 2:

$\frac{7s^3 - 8s^2 - 2s}{s^3 - 2.5s^2 + 0.5s + 1}$ kesrini kısmı kesirlere ayıralım.

```
p =
   2.0000
   1.0000
   0.5000

k =
   7
```

Bu durumda:

$$\frac{7s^3 - 8s^2 - 2s}{s^3 - 2.5s^2 + 0.5s + 1} = 7 + \frac{8}{s-2} + \frac{2}{s-1} - \frac{0.5}{s+0.5}$$ sonucu ortaya çeker.

→ **tf2zp**: Bu komut verilen transfer fonksiyonunu sıfır ve kutupları ayrı ayrı gösterir.

Örnek1: $\frac{5}{s^3 + 3s^2 + 2s}$ transfer fonksiyonunu sıfır ve kutuplarını bulup tekrar yazınız.

```
pay=5;
payda=[1 3 2 0];
[z,p,k]=tf2zp(pay,payda)

z =
  Empty matrix: 0-by-1
p =
   0
  -2
  -1
k =
   5
```

Bu durumda

$$\frac{5}{s^3 + 3s^2 + 2s} = \frac{5}{s(s+1)(s+2)}$$ şeklinde yazılabilir.

Örnek 2:

$\frac{s^2+5s+6}{s^4-s^3-7s2+s+6}$ transfer fonksiyonunu sıfır ve kutuplarını bulup tekrar yazınız.

Çözüm:

```
pay=[1 5 6]
payda=[1 -1 -7 1 6];
[z,p,k]=tf2zp(pay,payda)

z =
  -3.0000
  -2.0000
p =
   3.0000
   1.0000
  -2.0000
  -1.0000
k =
   1
```

Bu durumda $\frac{s^2+5s+6}{s^4-s^3-7s2+s+6} = \frac{(s+2)(s+3)}{(s-3)(s-1)(s+2)(s+1)}$ şeklinde yazılır.

Ltiview Grafiksel Kullanıcı Ara Biriminin Kullanılması

Bu grafiksel ara yüzü kullanarak aşağıdaki çizim tiplerini görmek mümkündür.

-Birim basmak(step) ve dürtü(impulse) cevapları.
- Bode ve Nyquist çizimleri.
- Kutup/sıfır çizimleri.

→ltiview komutları:

- ltiview (sistem1,'r-*',sistem2,'b- -)

Bu durumda her sistem grafiği için ayrı renk ve çizgi stilini belirtmek mümkündür.

- ltiview(çizimtipi,sistem1,sistem2...sistemN) komutu ile hangi tür grafiği

görüntülemek istediğimizi belirtebiliriz. Görüntülenebilecek grafik türlerinden bazıları aşağıda verilmiştir.

a) 'step'
b) 'impulse'
c) 'bode'
d) 'nyquist'
e) 'bodemag'
f) 'pzmap'

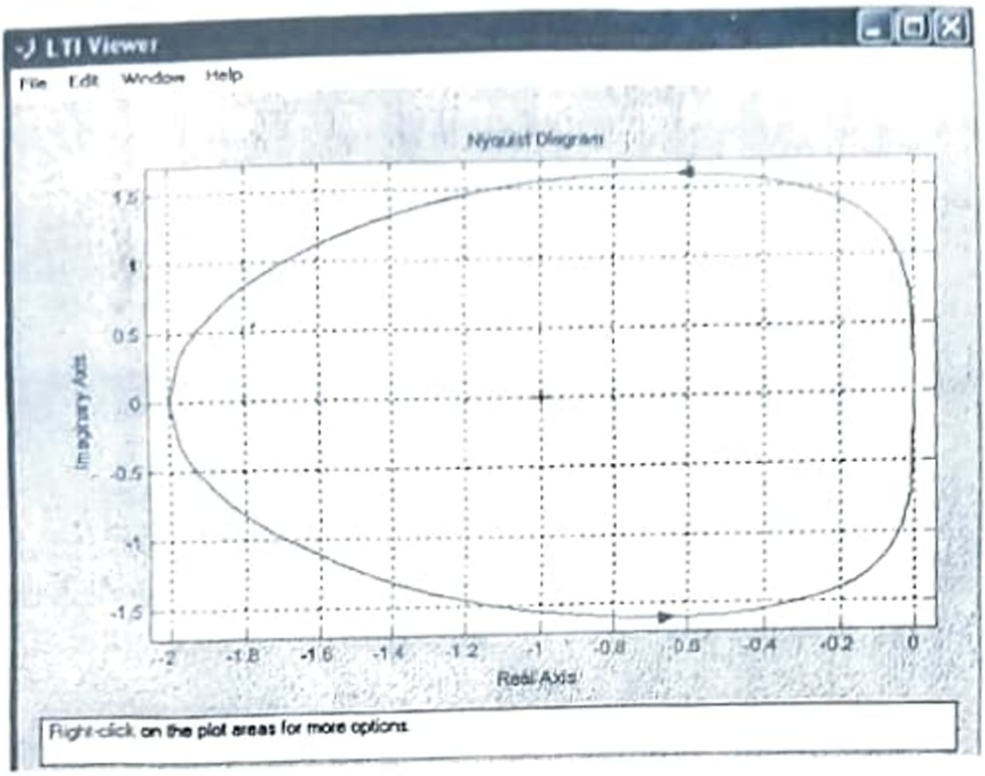
Ayrıca her sistem için aşağıda gösterildiği gibi ayrı grafik türü girmek mümkündür: ltiview({'step';'bode'},sistem1,sistem2). Bu durumda sistem1 için birim basamak(step) grafiği çizilirken , sistem2 için bode grafiği çizilir.

Örnek1:

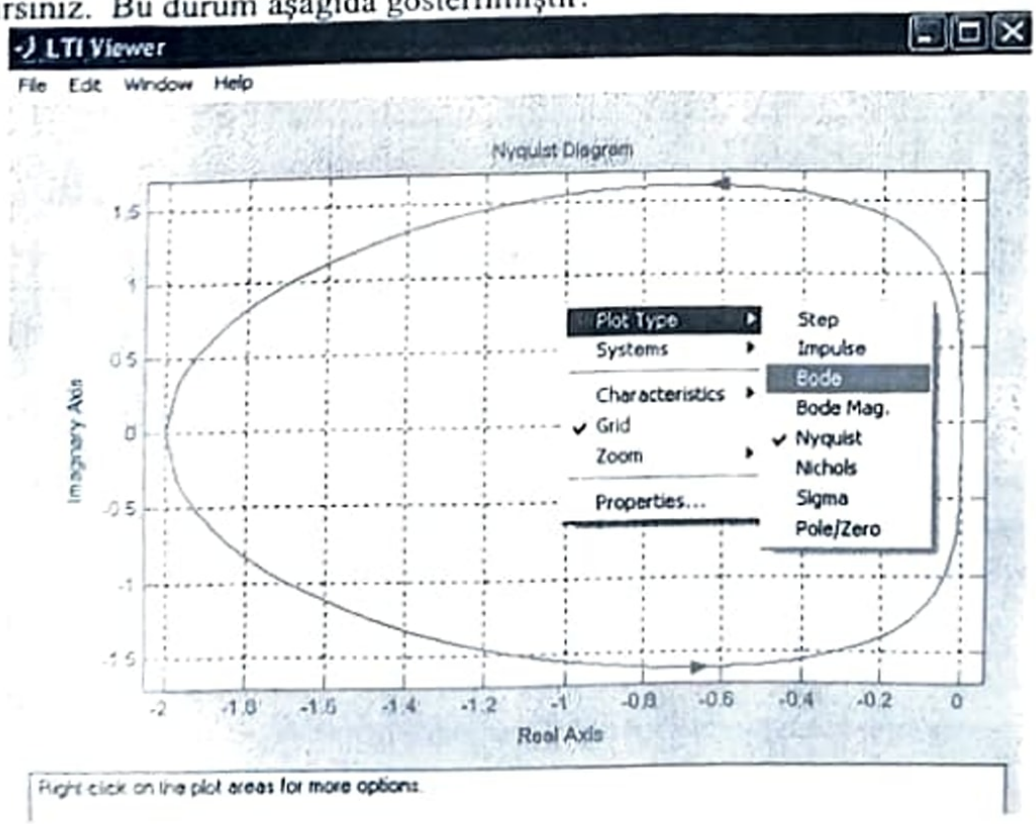
$H(s) = \frac{4(s+1)}{(s-1)(s+2)}$ fonksiyonunun nyquist diyagramını kırmızı renkli olarak, ltiview komutunu kullanarak çizdiriniz.

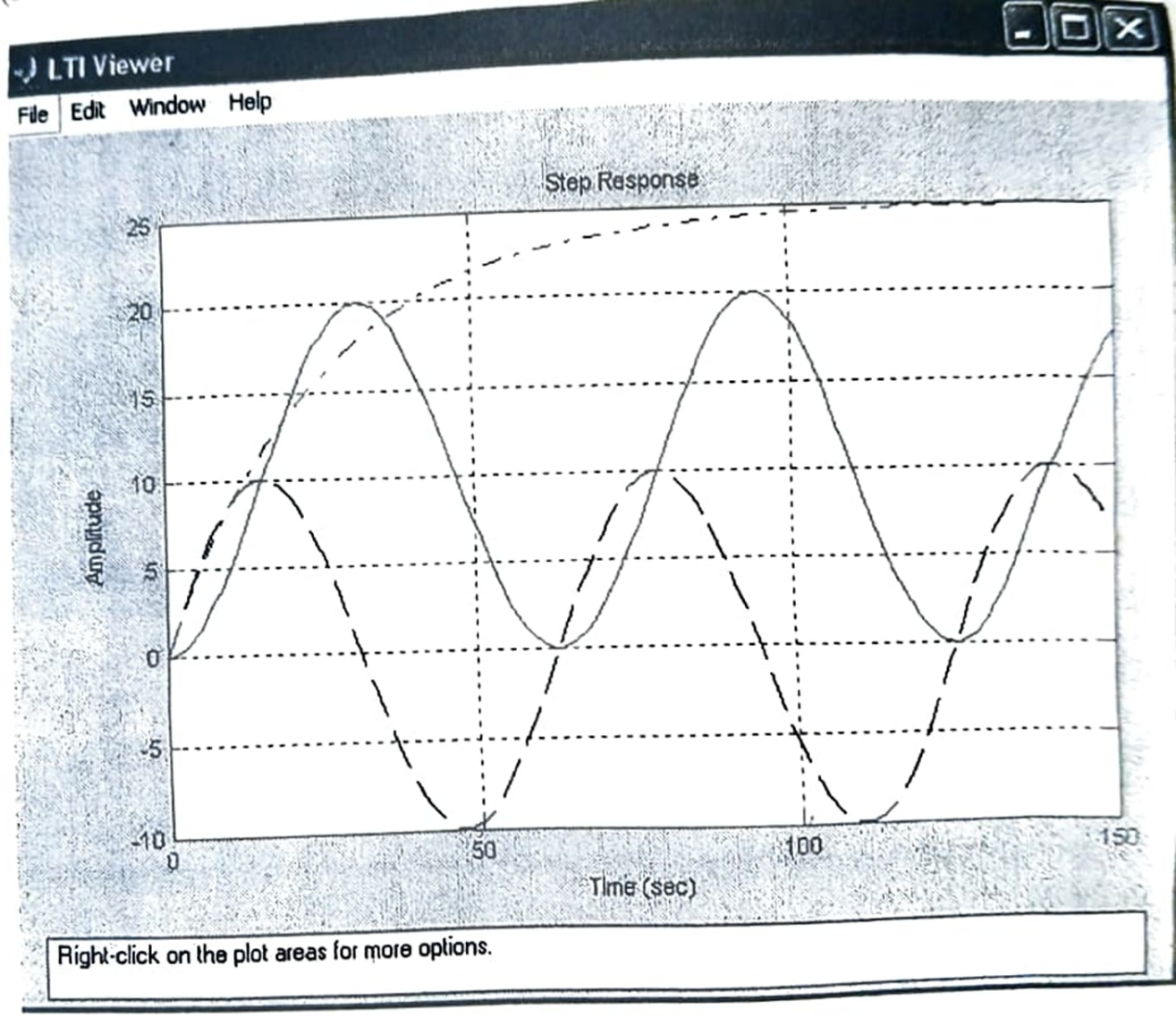
Çözüm

```
s=tf('s');
sistem1=4*(s+1)/(s-1)/(s+2)
ltiview('nyquist',sistem1,'r')
```

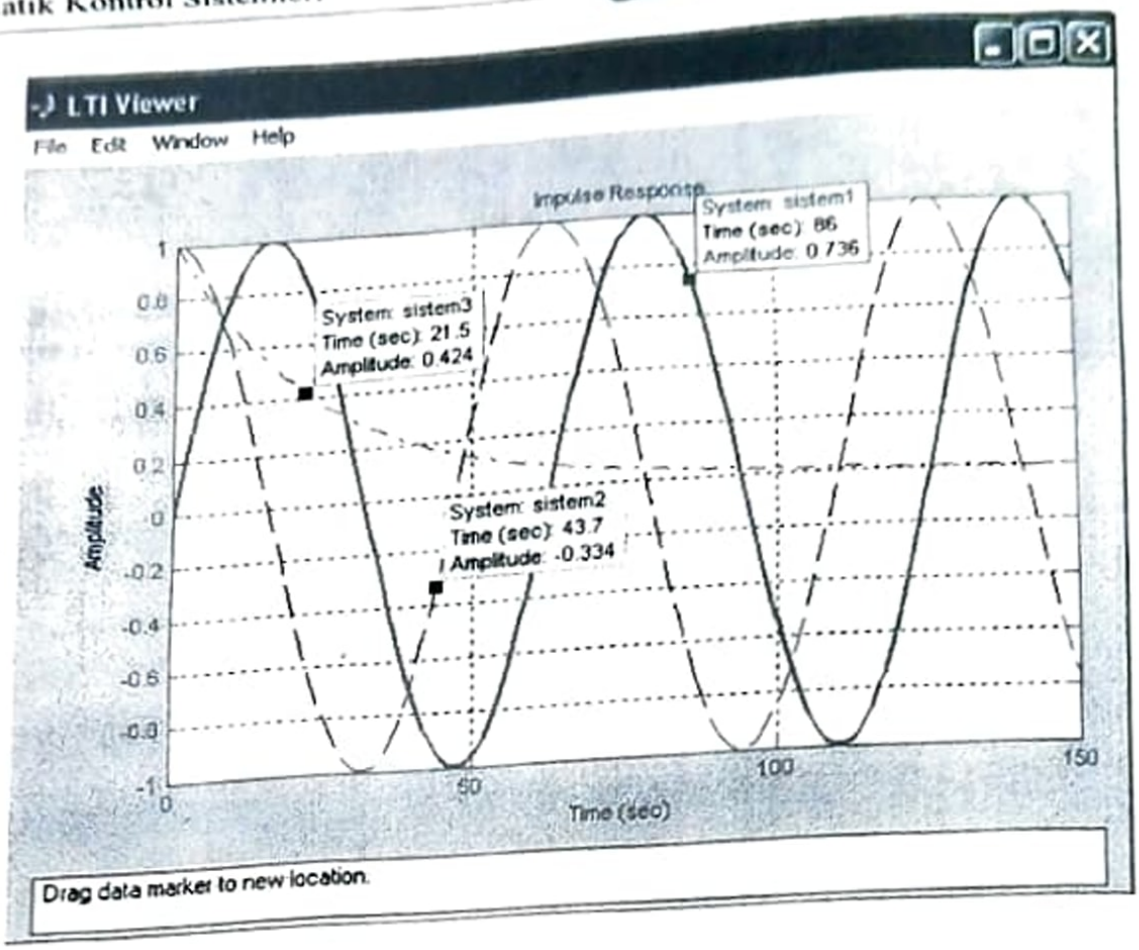


Not: ltivieew penceresinde çalışırken farenin sağ tuşuna basıp, çizim şeklini değiştirebilirsiniz. Bu durum aşağıda gösterilmiştir:





Not: ltivieew penceresinde çalışırken herhangi bir grafik üstüne gelip, farenin sol basıp bu grafik üstünde gezdirilirse, ilgili grafiğin x ve y değerlerinin okunması müm Bu durum aşağıdaki grafikte gösterilmiştir.

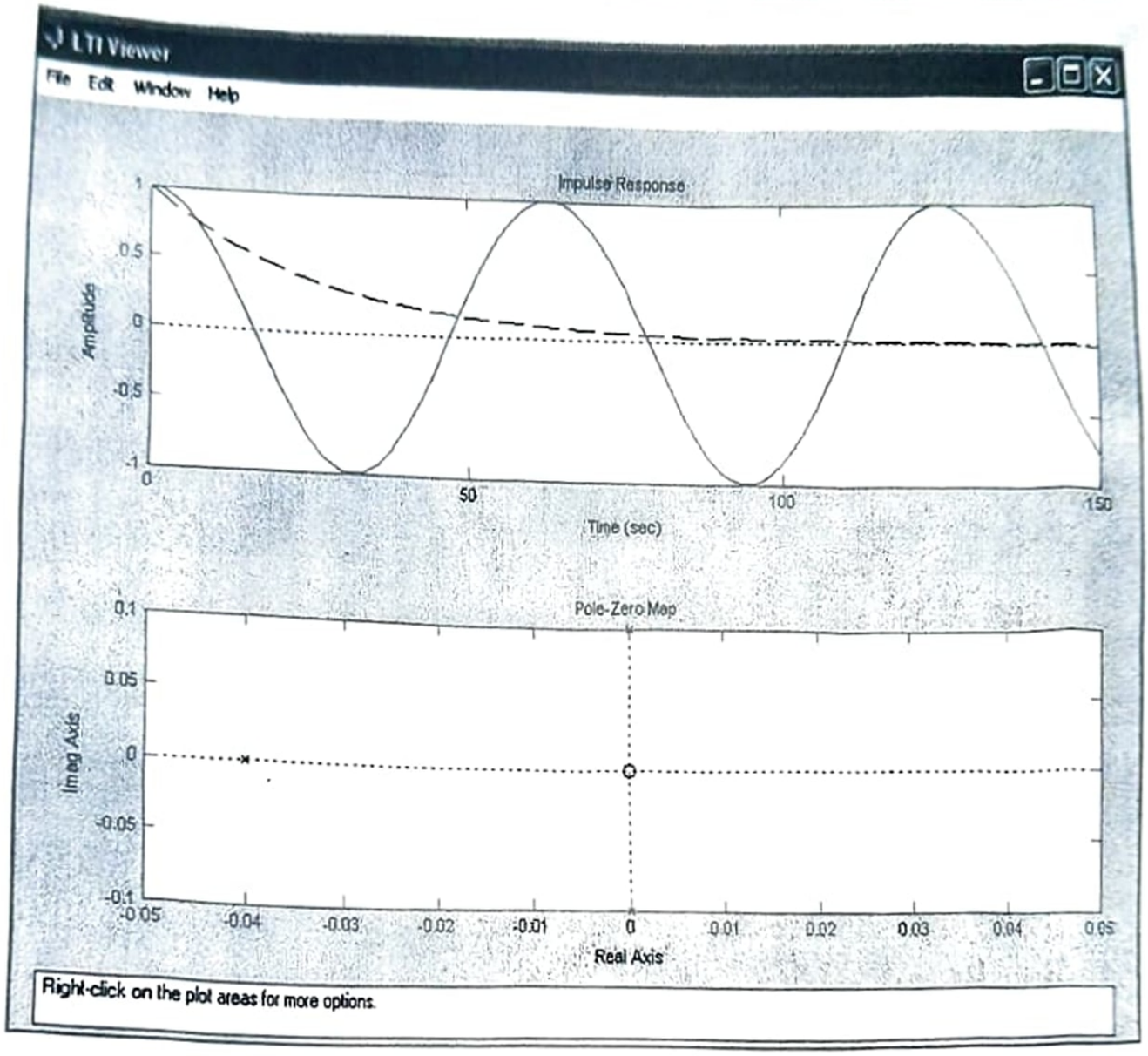


Örnek 3:

$\frac{s}{s^2+0.01}$ ve $\frac{1}{s+0.04}$ fonksiyonunun dürtü (impulse) cevaplarını ve sıfır-kutup grafiklerini ltiview grafiksel kullanıcı ara birimini kullanarak aynı pencerede çizdiriniz.

Çözüm:

```
s=tf('s');
sistem1=s/(s^2+0.01)
sistem2=1/(s+0.04)
ltiview({'impulse';'pzmap'},sistem1,'r',sistem2,'b--')
```

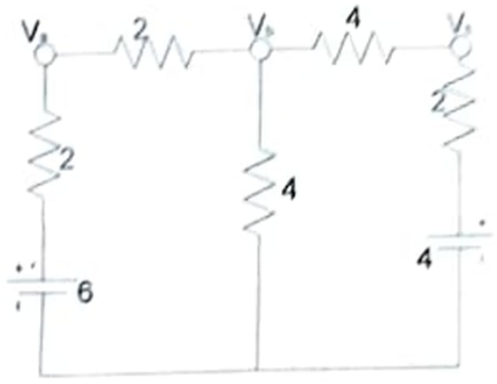

LTI Viewer
File Edit Window Help
Impulse Response
Amplitude
1
0.5
0
-0.5
-1
0
50
100
150
Time (sec)
Pole-Zero Map
Imag Axis
0.1
0.05
0
-0.05
-0.1
-0.05
-0.04
-0.03
-0.02
-0.01
0
0.01
0.02
0.03
0.04
0.05
Real Axis
Right-click on the plot areas for more options.

ELEKTRONİK MÜHENDİSLİĞİ PROBLEMLERİ

Bu bölümde elektronik mühendisliğindeki problemlerin Matlab ile çözülmesi üzerinde durulmuştur. Örnek olarak bir devre verilmiş, bunun matematiksel modellenmesi yapılmış ve Matlab'e aktarılmıştır.

Doğru Akım Devrelerinin Çözüm Yöntemleri

1) Düğüm Gerilimleri Yöntemi:



Yukarıdaki devre için Va,Vb,Vc değerlerini hesaplayınız.

Çözüm:

a noktası için $\frac{6-Va}{2}=\frac{Va-Vb}{2}$ $\quad 2Va-Vb=6$

b noktası için $\frac{Vb-Va}{2}+\frac{Vb}{4}+\frac{Vb-Vc}{4}$ $\quad -2Va+4Vb-Vc=0$

c noktası için $\frac{Vb-Vc}{4}=\frac{Vc-4}{2}$ $\quad -Vb+3Vc=8$

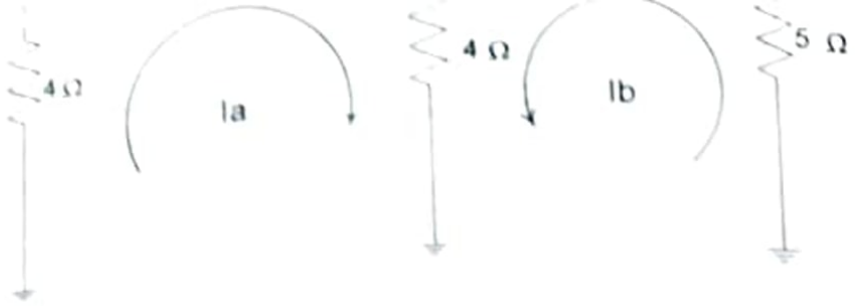
$$\begin{bmatrix} 2 & -1 & 0 \\ -2 & 4 & -1 \\ 0 & -1 & 3 \end{bmatrix} x \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = \begin{bmatrix} 6 \\ 0 \\ 8 \end{bmatrix}$$

matlab programı:

```
a=[2 -1 0;-2 4 -1;0 -1 3];
b=[6;0;8];
sonuc=inv(a)*b
```

```
sonuc =
   4.6250
   3.2500
   3.7500
```

Programdan görüldüğü gibi Va=4.625, Vb=3.25,Vc=3.75 volt olarak bulunur.



4 Ω'luk direnç üzerinden geçen akımı hesaplayınız.

$4I_a-10+2I_a+4(I_a+I_b)=0$
$5I_b-8+4(I_a+I_b)=0$

$10I_a+4I_b=10$
$4I_a+9I_b=8$

```
a=[10 4;4 9];
b=[10;8];
sonuc=inv(a)*b
```

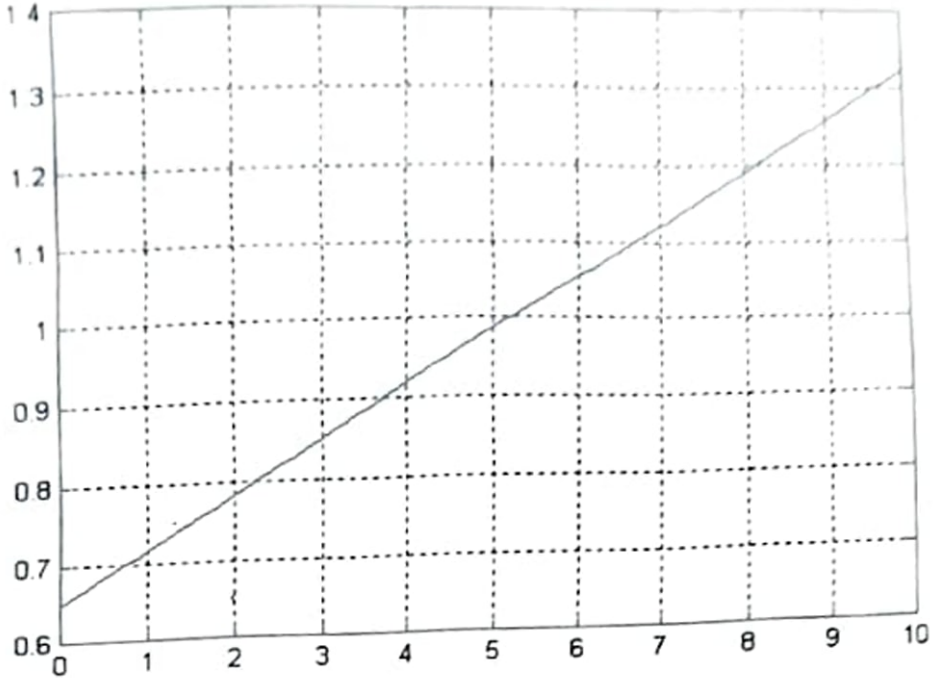
```
sonuc =
   0.7838
   0.5405
```

Dolayısıyla; I_a=0.7838 A ve I_b=0.5405 olur. 4 Ω'luk direnç üzerinden geçen akım bu iki akımın toplamı olacağından,
$I_c=I_a+I_b$=0.7838+0.5405= 1.3243 A olur.
Bu değeri matlab'de
>> sum(sonuc) komutunu kullanarak bulabiliriz.

Devremizde Vb gerilimini sabit tutarak Va gerilimini 0 ile 10 arasında değiştirelim ve Va akımının toplam akıma etkisini gözleyelim. Bu etkiyi görmek için aşağıdaki matlab programı yazılmıştır.

```
Va=0:.1:10;
Vb=8*ones(size(Va));
a=[10 4;4 9];
b=[Va;Vb];
sonuc=inv(a)*b;
It=sum(sonuc);
plot(Va,It)
grid
```

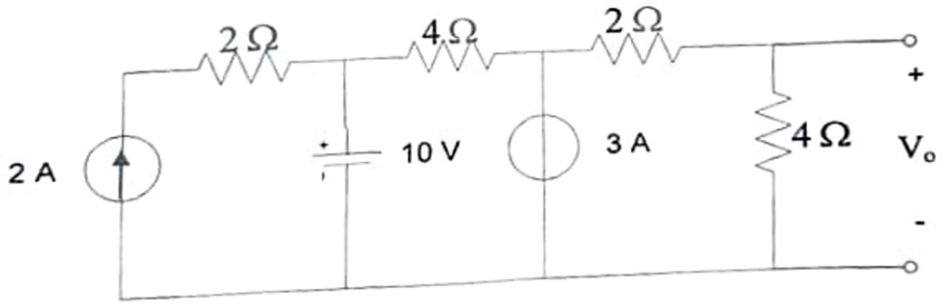
program sonrasında aşağıdaki grafik elde edilmiştir.



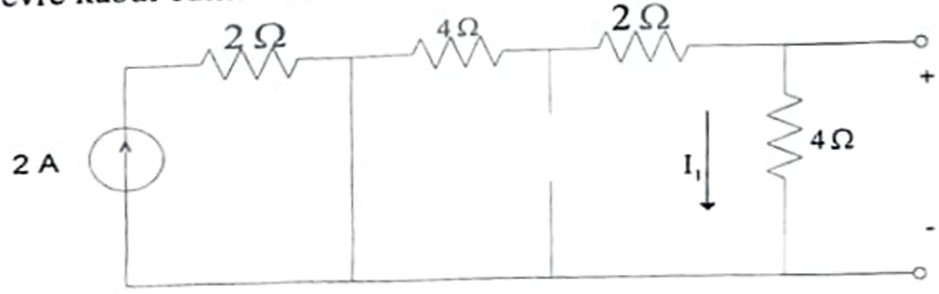
Görüldüğü gibi VA gerilimin It akımına etkisi doğrusal olmuştur. Çünkü bu devre sadece dirençlerden oluşan doğrusal bir devredir.

3) Süperpozisyon Yöntemi:

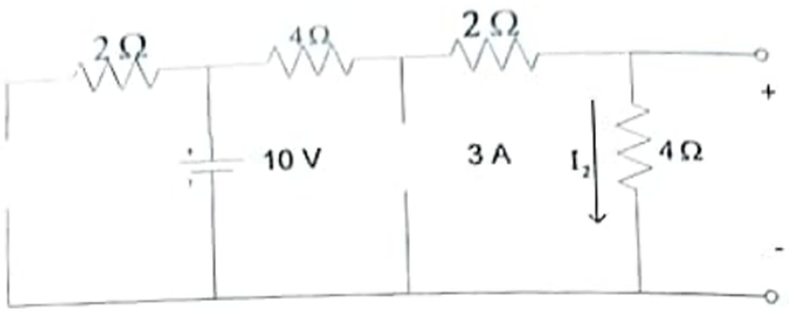
Örnek: V_o gerilimini hesaplayınız?



V_o gerimini hesaplamak için Io akımını bulalım.
2 A'lik akım kaynağının Io'ya katkısını hesaplamak için gerilim kaynağı kısa devre, akım kaynağı açık devre kabul edilir. Bu durumda aşağıdaki devre elde edilir.

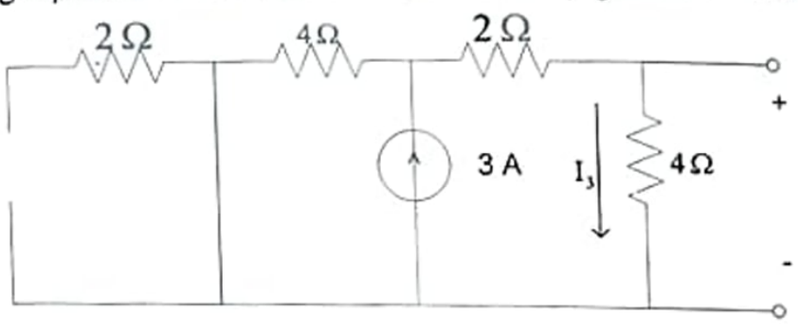


Devrede görüldüğü gibi kısa devreden dolayı I1=0 A bulunur.
10V'luk gerilim kaynağının Io'ya katkısını hesaplamak için 2A'lik ve 3A'lık akım kaynakları açık devre kabul edilir. Bu durumda aşağıdaki devre elde edilir.



Hesaplamalar yapılırsa $I_2 = \frac{10}{4+2+4} = 1\,A$ olarak bulunur.

3A'lik akım kaynağının I_o'ya katkısını bulmak için 10V'luk gerilim kaynağı kısa devre ve 3A'lik akım kaynağı açık devre kabul edilir. Bu durumda aşağıdaki devre elde edilir.



$I_3 = 3.\frac{4}{4+2+4} = 1.2A$ olarak bulunur.

Io akımı bu üç akımın toplamıdır. I0=I1+I2+I3=0+1+1.2=2.2A olarak bulunur.
Bu bilgileri matlab'e şu şekilde aktarabiliriz.

I_1=0
I_2=10/(4+2+4)
I_3=3*4/(4+2+4)
I_0=Io+I_1+I_2

Örnek: Paralel Bağlı Dirençlerin Eşdeğer direncinin Bulunması:

Paralel Bağlı dirençlerin eşdeğeri şu şekilde hesaplanır:

$$R_{eş} = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} + \ldots . \frac{1}{R_n}}$$

Matlab'de aşağıdaki program ile istenilen miktarda paralel direncin eşdeğeri hesaplanabilir.

```
clc
n=input('Toplam direnç sayisini giriniz: ');
for i=1:n
   fprintf('%d. nci direnç degerini giriniz: ',i);
   R(i)=input('');
end
Res=0;
for i=1:n
   Res=Res+1/R(i);
```

```
end
Res=1/Res;
fprintf('Verien %d tane paralel baglin direncin esdeger direnci = %f',n,Res)
```

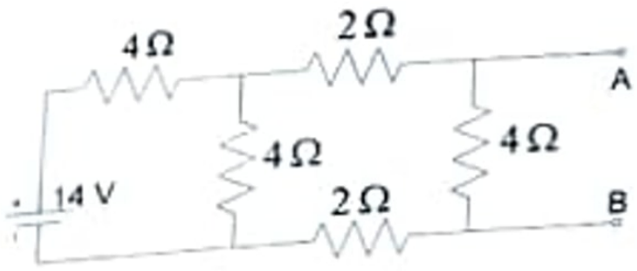
Bu program ile değerleri 6,4,12 Ω olan üç direncin eşdeğer direnci aşağıdaki gibi bulunur.

```
Toplam direnç sayisini giriniz: 3
1. nci direnç degerini giriniz: 6
2. nci direnç degerini giriniz: 4
3. nci direnç degerini giriniz: 12
Verien 3 tane paralel baglin direncin esdeger direnci = 2.000000
```

Thevenin Teoremi:

Bu teoremde devre bir gerilim kaynağı ve ona seri bağlı bir direnç şekline dönüştürülür.

Örnek:



A-B uçları arasındaki gerilim $= \frac{14}{\text{Reş}} \cdot \frac{6}{12} \cdot 4$

$\text{Reş} = 4 + \frac{4.8}{4+8} = 6.667\ \Omega$

Dolayısıla VAB = 4.2 V olarak bulunur.

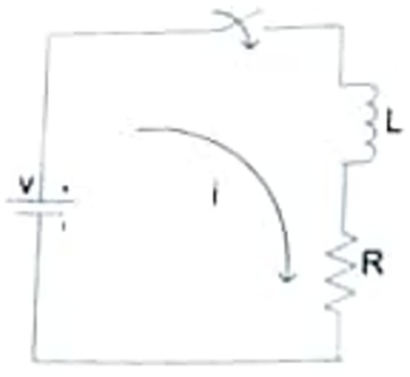
$\text{RAB} = \frac{4.6}{4+6} = 2.4\ \Omega$

Bu problemi Matlab ortamına şu şekilde aktarabiliriz.

```
Res=4+4*8/(4+8)
Vab=14/Res*6/12*4
```

Doğru Akımda Bobin ve Kondansitör

Bobinli Devrede Akımın Yükselişi:



Yukarıdaki denklemi $V - Ri = L\frac{di}{dt}$ ve $dt = \frac{L\,di}{V - Ri}$ her iki tarafında entegrali alınırsa:.

$t + k = -\frac{L}{R}\ln(V - Ri)$ $i(0^-)=i(0^+)=0$ olduğu için denklemlerde t ve i yerine sıfır konulursa

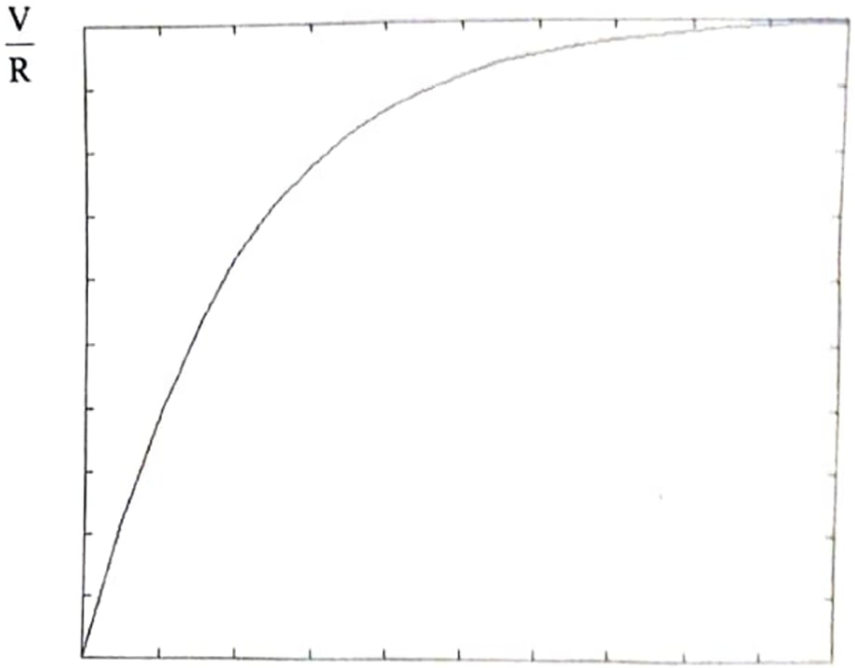
$k = -\frac{L}{R}\ln(V)$ olarak bulunur. k'yı yukarıdaki denklemde yerine koyarsak:

$t = -\frac{L}{R}\ln(V - Ri) + \frac{L}{R}\ln(V)$ bulunur.

$\frac{L}{R}\ln\left(\frac{V - Ri}{V}\right) = -t$ $\frac{V - Ri}{V} = e^{-\frac{R}{L}t}$ olur

ve $i = \frac{V}{R}\left(1 - e^{-\frac{R}{L}t}\right)$ (denklem 1.1)

olarak elde edilir.
Akımı zamana göre çizdirirsek aşağıdaki grafik elde edilir.

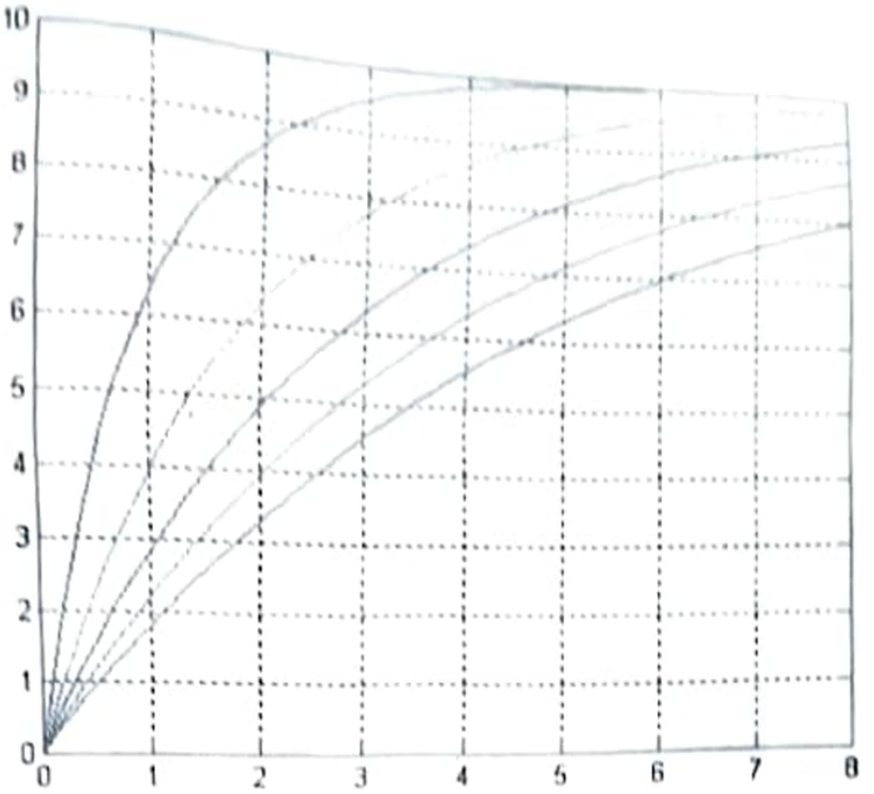


Şekil 1.1'de gösterilen devrede V=10 volt, R=5 Ω ve alınız. L=1 ile 5 arasında değişirken t'ye karşı i değerini çizdiriniz.

Örnek:

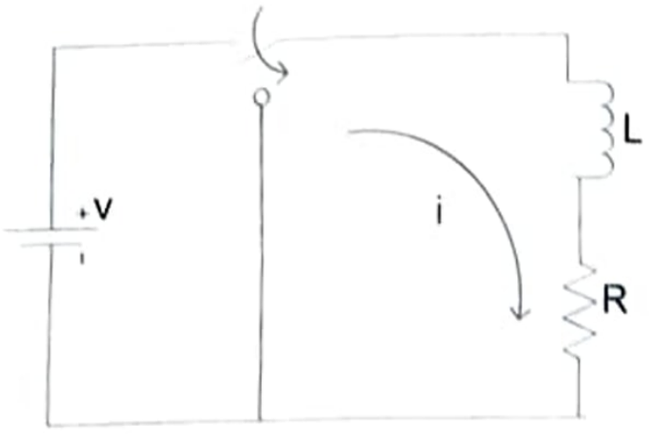
```
clf
V=10;R=1;
t=0:.1:8;
renk=['r','g','b','m','k']
for L=1:5
```

```
i=V/R*(1-exp(-R/L*t));
plot(t,i,renk(L))
hold on
end
grid
```



Görülüğü gibi, L=1 iken i akımı ,L=5 olduğu zamanki i akımından daha kısa sürede 10 değerine yakınsar. Bunun sebebi L yükseldikçe üssel değerin sıfıra daha geç gitmesidir.

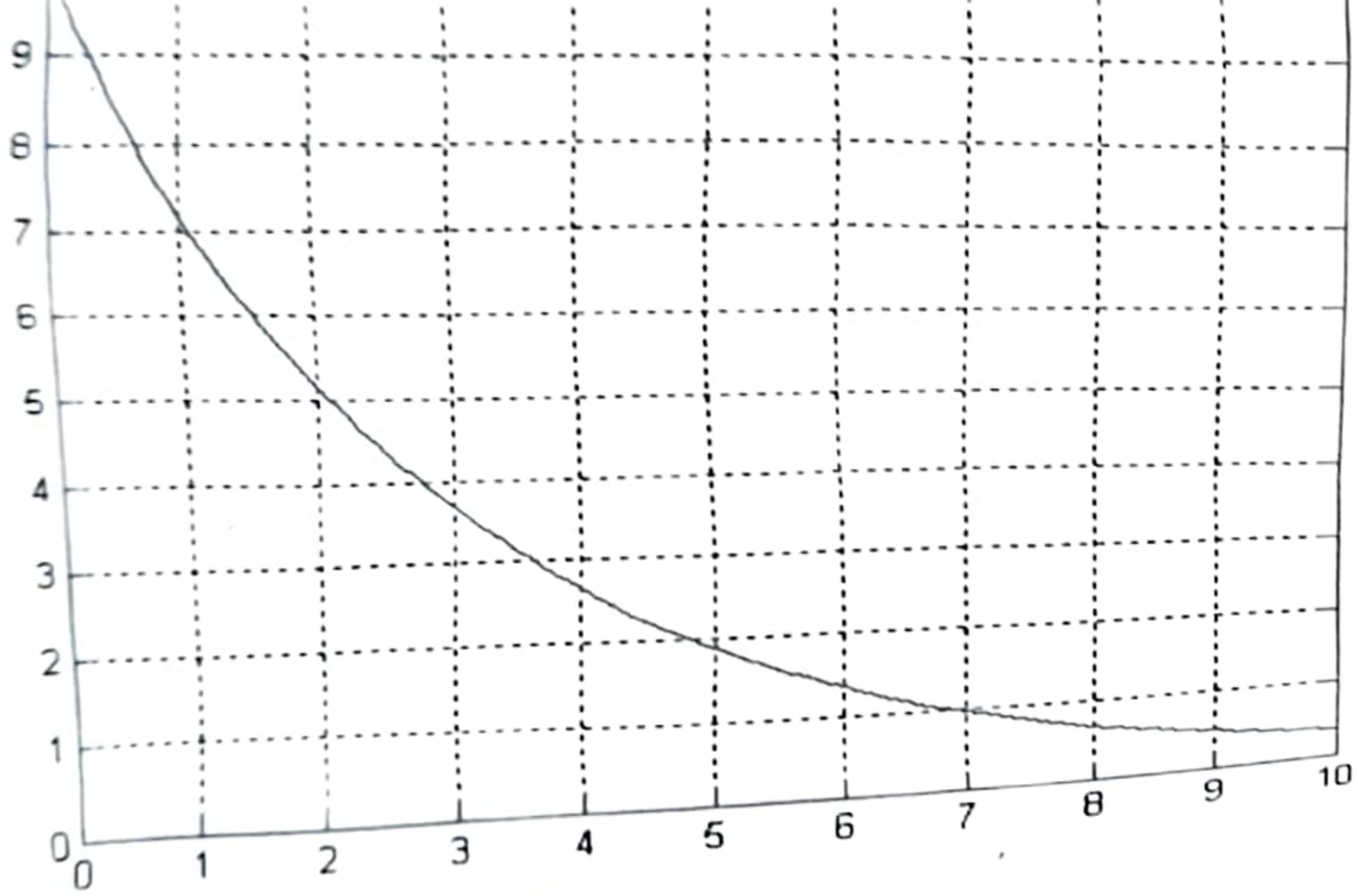
Bobinli devrede akımın azalışı



Şekil 1.2

Yukarıdaki devrede anahtar ok yönünde kapatılırsa akım bobinden dolayı aniden sıfıra düşmez. Anahtar kapatıldığı durumda i akımı için şu denklemi elde ederiz:

$Ri + L\frac{di}{dt} = 0$,Bu denklemi şu şekilde çözebiliriz:



satörün Dolması ve Boşalması:



Şekil 1.3'deki devrede anahtar kapatıldığında şu denklem yazılır:

$V=R.i+Vc$

$$V = R.C.\frac{dV_c}{dt} + V_c$$ (denklem 1.4)

$$\frac{V - V_c}{RC} = \frac{dV_c}{dt} \qquad \frac{dV_c}{V - V_c} = \frac{dt}{Rc} \qquad \frac{t}{RC} = -\ln(V - V_c) + k$$

$v(0^-)=v(0^+)=0$ değerlerini yerine koyarsak $0=-\ln(V)+k$, $k=\ln(V)$

$$\frac{t}{RC} = -\ln(V - V_c) + \ln(V) \quad \ln\left(\frac{V - Vc}{V}\right) = -\frac{t}{RC} \qquad \frac{V - Vc}{V} = e^{-\frac{t}{RC}} \text{ ve}$$

$$Vc = V\left(1 - e^{-\frac{t}{RC}}\right)$$ olarak bulunur. $i=C\frac{dv}{dt}$ olduğu için

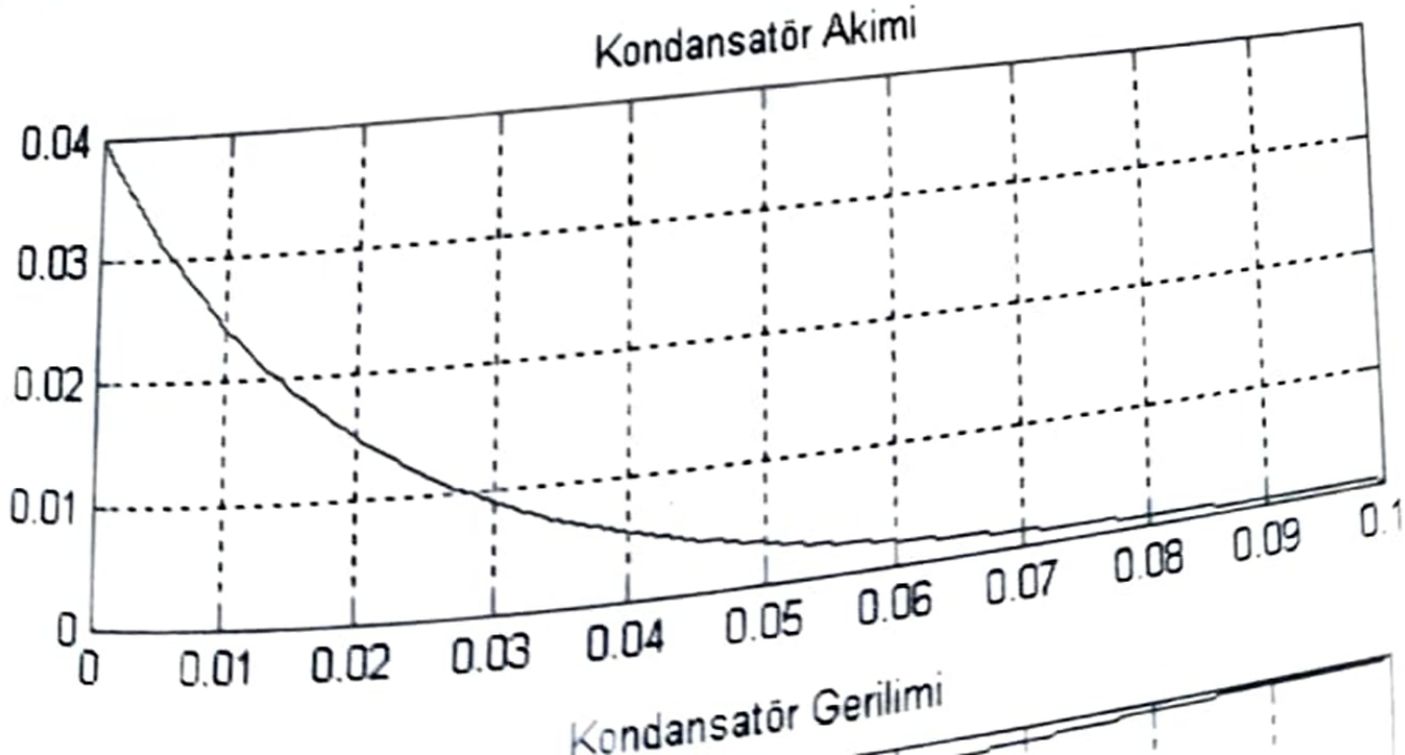
$$= \frac{V}{R} e^{-\frac{t}{RC}}$$ denklem(1.5)

arak elde edilir.

rnek :Şekil 3.teki devrede R= 2KΩ, C=10μF ve =80 volt dur. Kondansatörden geçen i
ımını ve kondansatör uçlarındaki gerilimi ayrı ayrı çizdiriniz.

özüm:

```
=80;R=2e3;C=10e-6;
inspace(0,0.1);
/R*exp(-t/R/C);
=V*(1-exp(-t/R/C));
plot(211),plot(t,i),grid,title('Kondansatör Akimi')
plot(212),plot(t,Vc),grid,title('Kondansatör Gerilimi')
```





Şekil 4

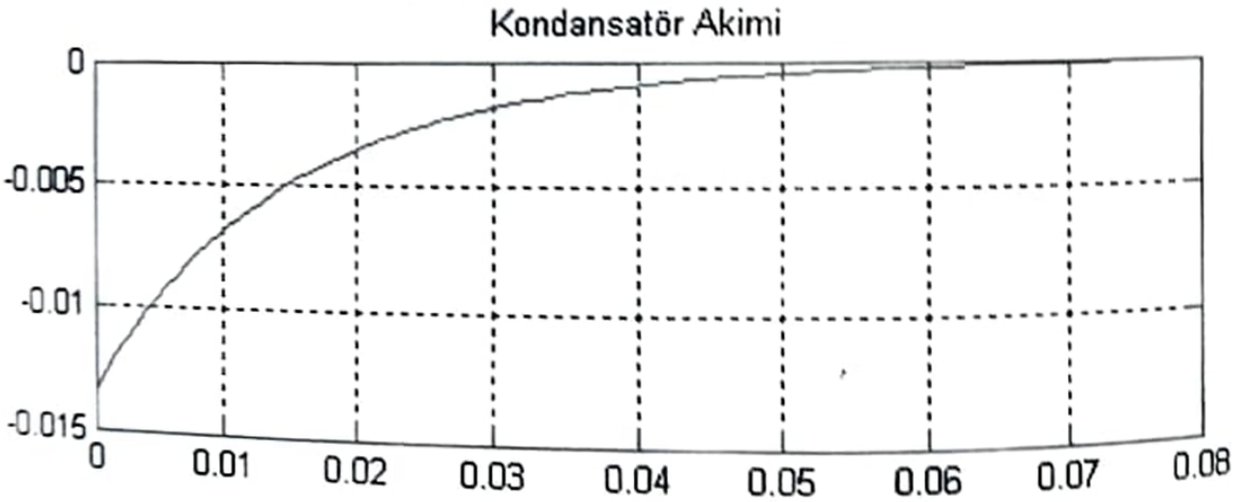
Yukarıdaki devre için anahtar ok yönünde kapatıldığı zaman şu denklemler elde edilir. (denklem 1.2'ye benzer şekilde çözülür)

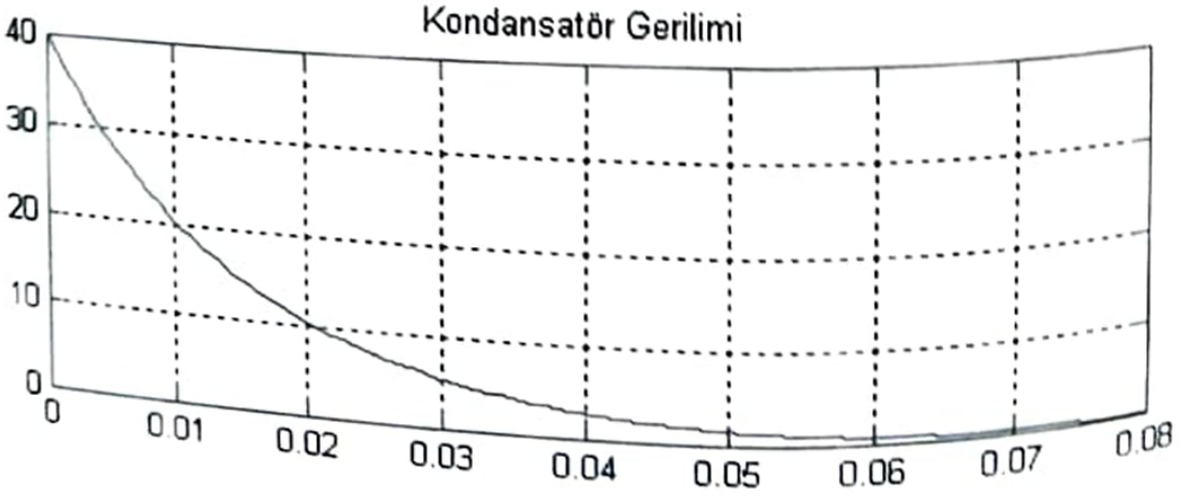
$V_c = Ve^{-\frac{t}{RC}}$ ve $i = -\frac{V}{R}e^{-\frac{t}{RC}}$

Şekil 4.teki devrede R= 3KΩ, C=2μF ve V=40 volt dur. Kondansatörden geçen i akımını ve kondansatör uçlarındaki gerilimi ayrı ayrı çizdiriniz.

Çözüm:

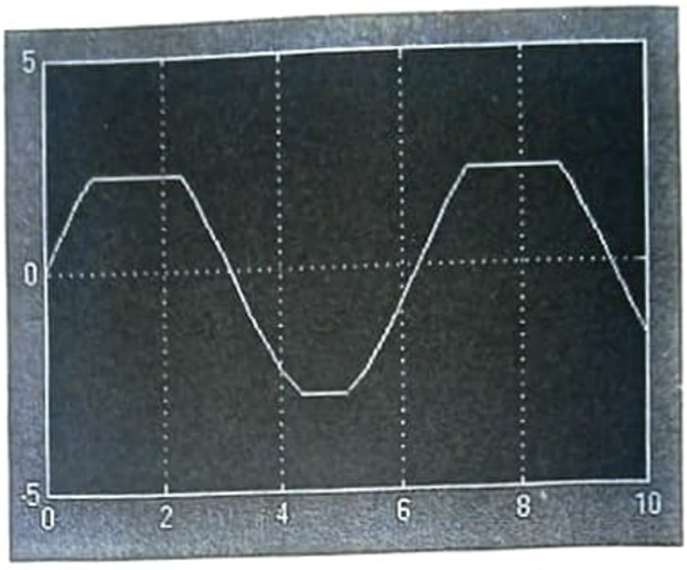
```
clf
V=40;R=3e3;C=5e-6;
t=linspace(0,0.08);
Vc=V*exp(-t/R/C);
i=-V/R*exp(-t/R/C);
subplot(211),plot(t,i),grid,title('Kondansatör Akimi')
subplot(212),plot(t,Vc),grid,title('Kondansatör Gerilimi')
```



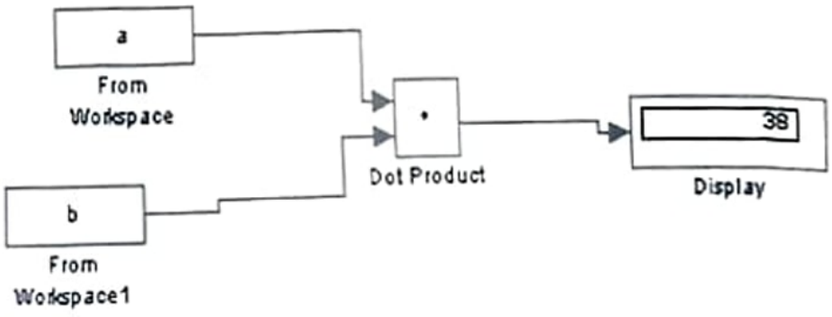




source kısmından sine wave, sinks kısmından scope ve nonlinear kısmından saturation se
Bu örnek de sinüsün genliği +3 ile –3 arasında değişmektedir. Saturation biriminin üst limiti
2.2, alt limiti de –2.2 dir.

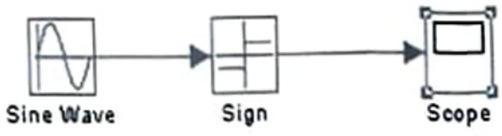


4) Nokta çarpımı:



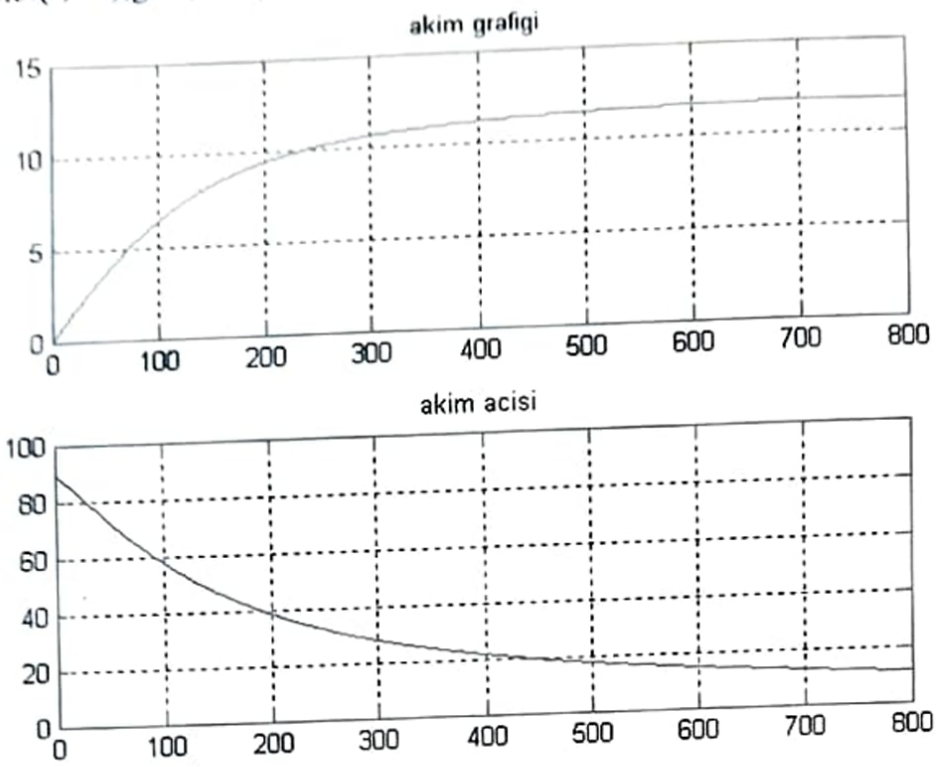
source kısmından 2 tane workspace, sinks kısmından 1 tane display ve math kısmından 1
dot product seçilir. Burada a ve b vektörleri command penceresinden girilmiş olmal
Simulink çarpım sırasında vektörlerin ilk elemanlarını dikkate almaz.

5) Sign işaretinin kullanılışı.

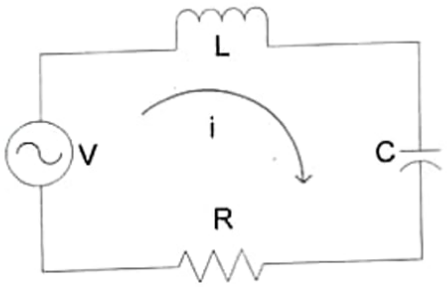


Source kısmından sine wave, sinks kısmından scope ve ve math kısmından sign se
Burada sing işareti sinüsün 0'dan büyük olduğu yerlerde1, küçük olduğu yerlerde –1
Bu durumda aşağıdaki sonuç ortaya çıkar.

```
genlik=abs(i);
aci=angle(i);
aci=rad2deg(aci);
subplot(211),plot(f,genlik),grid,title('akim grafigi')
subplot(212),plot(f,aci),grid,title('akim acisi')
```



Örnek:

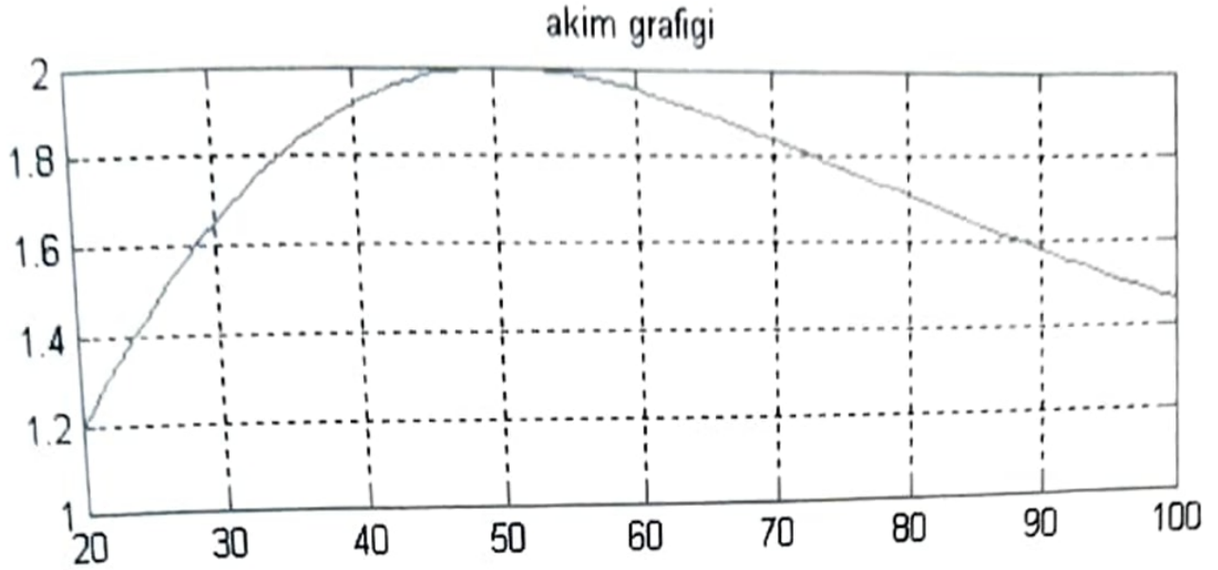


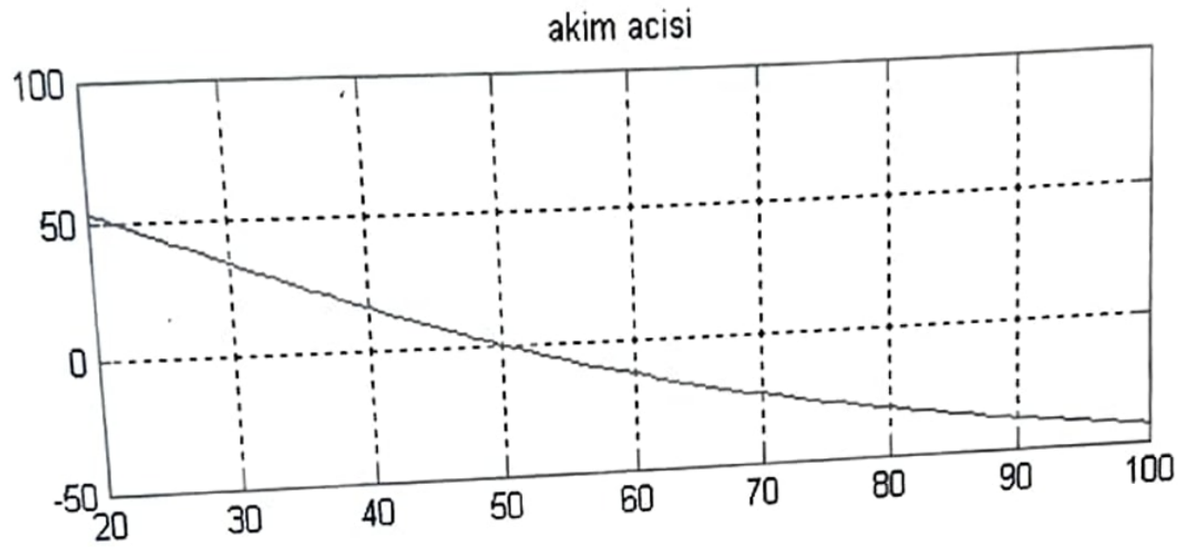
Yukarıdaki devre için V=100 volt, R=50Ω, L=100mH ve C=100μ'dır. Frekansın 20 ile 100 Hz arasında değiştiğini kabul ederek frekans-akım genliği ve frekans-akım açısı grafiklerini çiziniz.

Çözüm:

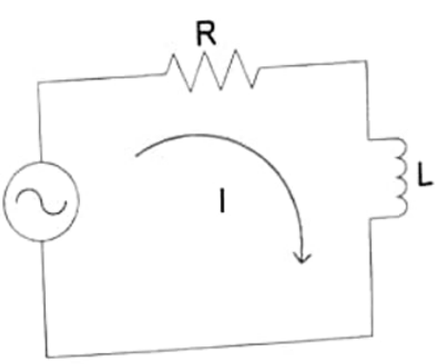
```
V=100;R=50;C=100e-6,L=100e-3;
f=20:100;
ZL=j*2*pi*f*L;
ZC=1./(j*2*pi*f*C);
Z=R+ZL+ZC;
i=V./Z;
genlik=abs(i);
aci=angle(i);
aci=rad2deg(aci);
```

(212),plot(f,aci),grid,title('akim acisi')





Örnek:

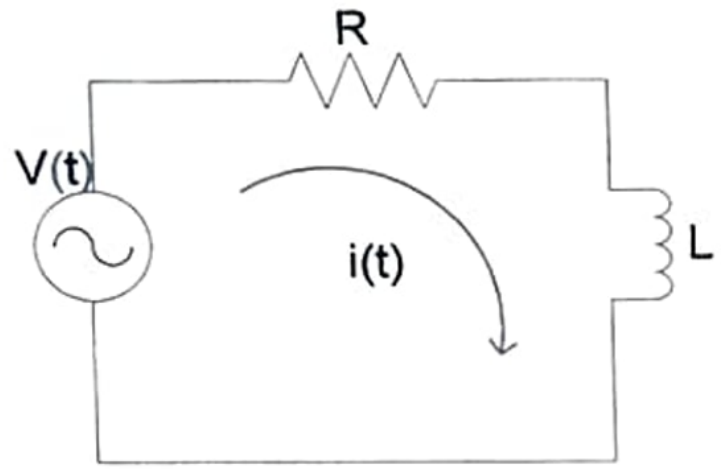


Yukaradaki devre için V=220 volt, 50Hz, R=20Ω ve L=100mH olduğuna göre
a) Empedansı, akımı ve güç çarpanını hesaplayınız.
b) Devrenin görünür gücünü, aktif gücünü ve reaktif gücünü hesaplayınız.

Çözüm:

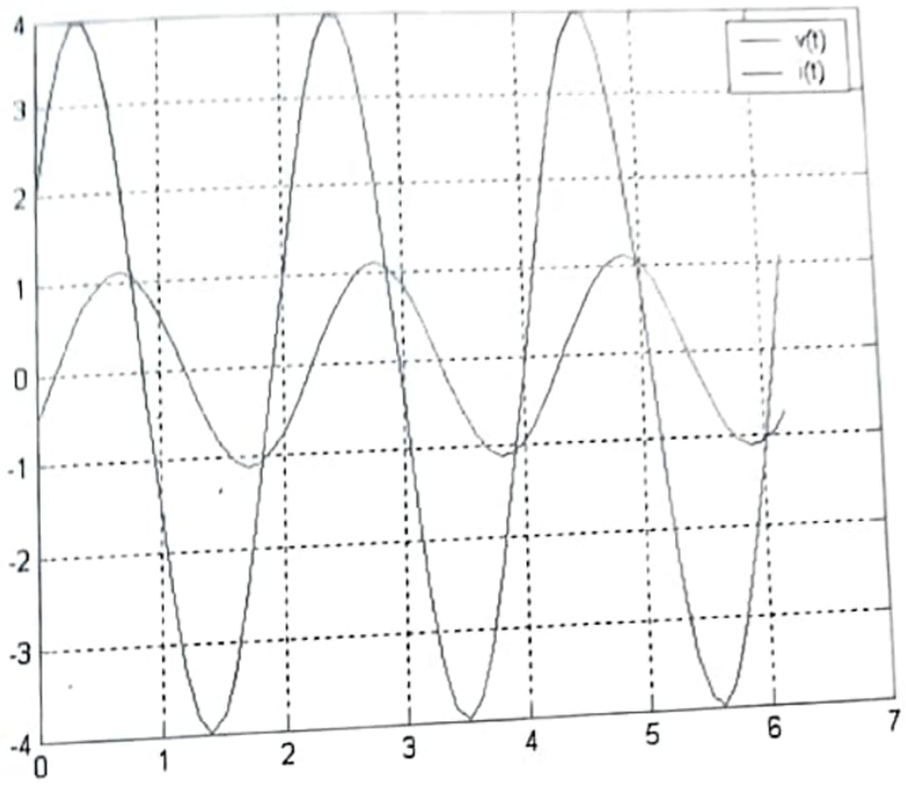
```
clc
V=220;R=20;L=100e-3;f=50;
ZL=j*2*pi*f*L;
Z=R+ZL
aci=angle(Z);
guc_carpani=cos(aci)
aci_derece=rad2deg(aci)
i=V/Z
genlik=abs(i)
```

Örnek:

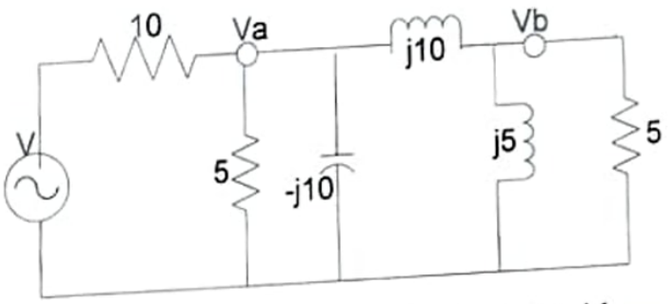


Yukarıdaki devrede $v(t) = 4\sin\left(3t + \frac{\pi}{6}\right)$, R=2Ω ve L=1H olduğuna göre v(t) ve i(t)'yi çizdiriniz. Voltaj ve akım arasındaki faz farkını derece cinsinden hesaplayınız.

```
clc
w=3;
v_g=4;v_a=pi/6;
Z=2+j*w*1;
i_g=v_g/abs(Z);
i_a=v_a-angle(Z);
t=0:.1:2*pi;
v=v_g*sin(w*t+v_a);
i=i_g*sin(w*t+i_a);
plot(t,v,t,i)
legend('v(t)','i(t)')
grid
faz=angle(Z);
faz=rad2deg(faz)

  faz = 56.3099
```



Örnek:



Yukarıdaki devrede V(t)=30cos(10t) olduğuna göre Va ile Vb'yi hesaplayınız ve çizdiriniz.

Va düğüm noktası için: $\frac{V_a - 30}{10} + \frac{V_a}{5} + \frac{V_a}{-j10} + \frac{V_a - V_b}{j10} = 0$

Vb düğün noktası için: $\frac{V_b - Va}{j10} + \frac{V_b}{j5} + \frac{V_b}{5} = 0$

jVa-30j+2jVa-Va+Va-Vb=0
Vb-Va+2Vb+2jVb=0

3jVa-Vb=30j
-Va+(3+2j)Vb=0

$$\begin{bmatrix} 3j & -1 \\ -1 & 3+2j \end{bmatrix} \cdot \begin{bmatrix} V_a \\ V_b \end{bmatrix} = \begin{bmatrix} 30j \\ 0 \end{bmatrix}$$

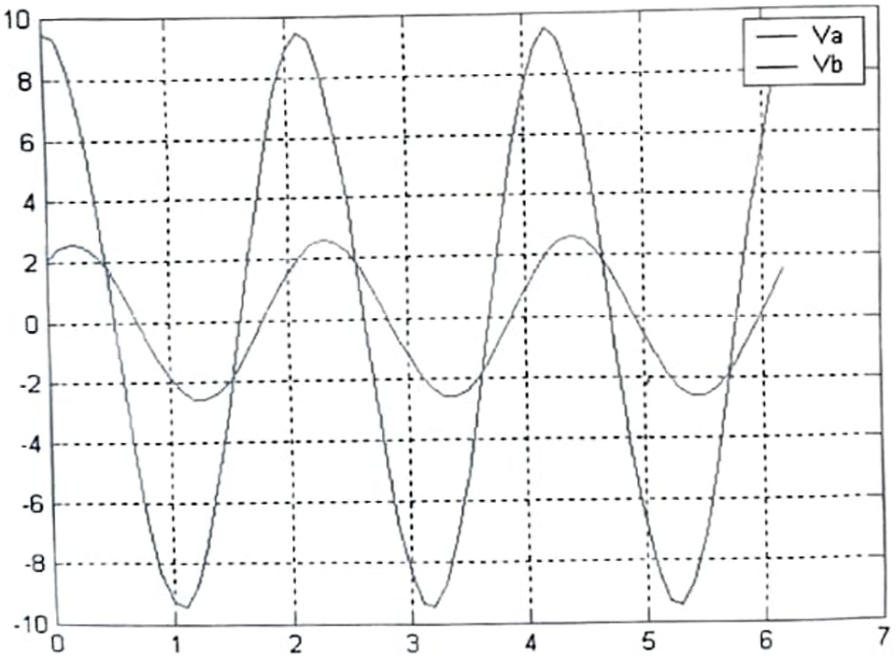
Va ve Vb'yi matlab'de şu şekilde bulabiliriz.

```
a=[3j -1;-1 3+2j];
b=[30j; 0];
sonuc=inv(a)*b
Va_gen=abs(sonuc(1))
Va_aci=angle(sonuc(1))
Vb_gen=abs(sonuc(2))
Vb_aci=angle(sonuc(2))
t=0:.1:2*pi;
Va=Va_gen*cos(3*t+Va_aci);
Vb=Vb_gen*cos(3*t+Vb_aci);
plot(t,Va,t,Vb),grid
legend('Va','Vb')
```

```
Va_gen =   9.4868
Va_aci =  -0.0730
Vb_gen =   2.6312
Vb_aci =  -0.6610
```

Bu durumda Va(t)=9.4868cos(3t-0.073) ve
Vb(t)=2.6312cos(3t-0.661) şeklinde bulunur. Burada açılar derece sinsindendir.

Bu gerilimlerin grafikleri aşağıda verilmiştir.



İNDEKS

A

abs 113
acos 113
acosh 114
acot 114
acoth 114
acsc 114
acsch 24
all 265
Alternatif Akım Devrelerinde Bobin ve Kondansatör 114
angle 23
any 78
area 115
asec 115
asech 115
asin 115
asinh 116
atan 116
atanh 72
axis

B

bar 80
bar3 99
bar3h 100
barh 81
bin2dec 166
Birim basamak 204
Birim impulse 203
Bobinli Devrede Akımın Yükselişi 259
bode 243
Bode diyagramı 243
box 105
brighten 111
Butterworth 210
buttord 221

C

camdolly 108
camlight 111
camorbit 109
camzoom 109
cart2pol 153
cart2sph 154
case 148
ceil 116
char 165
cheb1ord 221
cheb2ord 222
cheby1 213
cheby2 216
Chebysev Tip1 213
Chebysev Tip2 216
checkbox 199
color 75
comet3 101
compan 21
compass 82
complex 54
conj 116
contour3 99
conv 59,205
cos 117
cosh 117
counter3 99
csc 117
csch 117
cumprod 39
cumsum 38
cylinder 102

D

deblank 163
dec2bin 166
dec2hex 166
deconv 60
det 12
diag 14
diff 139
disp 33
Dizi Girişi 1
Doğrusal Modeller 239
doğrusal modeller 239
dot 3
durum uzayı 239
Düğüm Gerilimleri 255
Dürtü 203
DZD 244

E

edit 200
ellip 16
Elliptic 218
Eşdeğer Direnç 258
eval 162
exp 117
eye 5
ezmesh 105
ezplot3 106
ezsurf 106

F

factor 42
factoryel 129
fdatool 234
feedback 241
FIR filtreler 222
fill 78
fill3 98
filter 228
filtfilt 229
find 24
findstr 160
finite 25
fir1 222
fir2 224
firels 227

fix ... 118
fliplr ... 18
flipud ... 17
floor ... 118
fonksiyon yazma ... 129
fontsize ... 75
for ... 146
fourier ... 209
fprintf ... 33
function ... 129

G

gcbo ... 182
gcd ... 118
geomean ... 40
geri besleme bağıntısır ... 241
get ... 182
Grafiksel Kullanıcı Ara yüzü ... 179
grid ... 69
GUI ... 179
guide ... 179

H

hadamard ... 22
handles ... 182
hankel ... 22
hex2dec ... 167
hist ... 43

I

IIIR Filtreler ... 210

İ

if ... 145
imag ... 119
input ... 33
int ... 138
integral ... 137
interp1 ... 157
interpolasyon ... 157
intersect ... 5
inv ... 12
invilhb ... 23
isletter ... 160
ismember ... 5
isnan ... 24
isspace ... 160

K

Karakter Dizisi Komutları ... 159
kısmi kesirlere ayırma ... 248
kompleks sayılar ... 49
Kondsanitörün Boşalışı ... 264
Kontrol Komutları ... 145
Koordinat Sistemleri ... 153
kökün geometrik yeri ... 241
kron ... 19

L

lcm ... 119
legend ... 73
length ... 5
line ... 79
linspace ... 1
log ... 120
log10 ... 120
log2 ... 120
logspace ... 2
lower ... 163
ltiview ... 250

M

magic ... 15
mat2str ... 162
Matris Çarpımı ... 9
Matris Kuvveti ... 11
Matris Toplamı ... 9
max ... 37
mean ... 39
medfilt1 ... 230
median ... 40
mesh ... 94
min ... 37
mod ... 121
doğrusal modeller arasında bağıntılar ... 240
modeller arasında çevirim yapma ... 240
Modellerin Oluşturulması ... 239

N

numel ... 20
nyquist ... 245
Nyquist diyagramı ... 244

O

ones ... 5
Optimizasyon ... 133
Otomatik Kontrol Sistemleri ... 239

Ö

Özel Diziler ... 1

P

pascal ... 15
peaks ... 104
perms ... 42
pie ... 79
pie3 ... 101
plot ... 69
plot3 ... 93
pol2cart ... 154
polar ... 82
polinomlar ... 57
poly ... 58
polyder ... 61

fix 118
fliplr 18
flipud 17
floor 118
fonksiyon yazma 129
fontsize 75
for 146
fourier 209
fprintf 33
function 129

G

gcbo 182
gcd 118
geomean 40
geri besleme bağıntısır 241
get 182
Grafiksel Kullanıcı Ara yüzü 179
grid 69
GUI 179
guide 179

H

hadamard 22
handles 182
hankel 22
hex2dec 167
hist 43

I

IIR Filtreler 210

İ

if 145
imag 119
input 33
int 138
integral 137
interp1 157
interpolasyon 157
intersect 5
inv 12
invilhb 23
isletter 160
ismember 5
isnan 24
isspace 160

K

Karakter Dizisi Komutları 159
kısmi kesirlere ayırma 248
kompleks sayılar 49
Kondsanitörün Boşalışı 264
Kontrol Komutları 145
Koordinat Sistemleri 153
kökün geometrik yeri 241
kron 19

L

lcm 119
legend 73
length 5
line 79
linspace 1
log 120
log10 120
log2 120
logspace 2
lower 163
ltiview 250

M

magic 15
mat2str 162
Matris Çarpımı 9
Matris Kuvveti 11
Matris Toplamı 9
max 37
mean 39
medfilt1 230
median 40
mesh 94
min 37
mod 121
doğrusal modeller arasında bağıntılar 240
modeller arasında çevirim yapma 240
Modellerin Oluşturulması 239

N

numel 20
nyquist 245
Nyquist diyagramı 244

O

ones 5
Optimizasyon 133
Otomatik Kontrol Sistemleri 239

Ö

Özel Diziler 1

P

pascal 15
peaks 104
perms 42
pie 79
pie3 101
plot 69
plot3 93
pol2cart 154
polar 82
polinomlar 57
poly 58
polyder 61

polyfit 62
polyint 62
polyval 58
popupmenu 189
primes 42
prod 38
pzmap 243

Q

quad 137

R

radiobutton 194
rand 43
randperm 1
rank 25
real 121
rem 121
remez 223
repmat 17
reshape 16
residue 57,245
rlocus 239
root locus 238
roots 57
rose 82
rotate3d 108
rotation 76
round 122

S

sawtooth 229
Sayısal İşaret İşleme 201
scatter3 107
sec 122
sech 122
semilogx 77
semilogy 77
set 190
setdiff 5
shading 112
sıfır/kutup/kazanç 236
sign 175
Simulink 169
sin 122
sinc 230
sinh 122
sistem modelleme 236
size 19
slider 181
sort 37
sph2cart 154
sphere 103
sptool 232
sqrt 122
square 230
stairs 83
std 41
stem 81
str2mat 166
str2num 166
strcat 162
strcmp 159
strjust 163
strmatch 162
strncmp 159
strrep 159
strtok 164
strvcat 161
sum 38
surf 95
surfl 97
Süperpozisyon 257
switch 148
symvar 165

T

tan 123
tanh 123
text 74
tf 239
tf2zp 249
trace 25
transfer fonksiyonu 239
trapz 137
tril 19
tripuls 233
triu 18
türev 137

U

union 5
upper 165

Ü

Üç Boyutlu Grafik 93

V

var 41

W

waterfall 103
while 147

Z

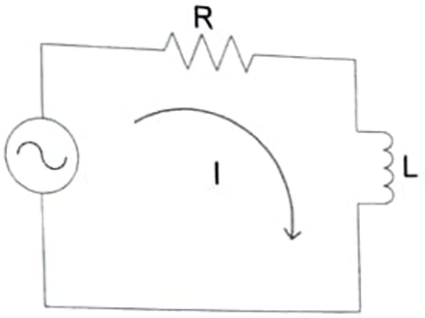
zeros 5
zpk 239

Alternatif Akım Devrelerinde Bobin ve Kondansatörün İncelenmesi

Bobin:

Bobin ve kondansatörün alternatif akım devrelerinde göstermiş olduğu direnç frekansla değişir. Bu direnç değerleri şu şekilde hesaplanır:

$$Z_L = j2\pi fL \qquad Z_c = \frac{1}{j2\pi fC}$$



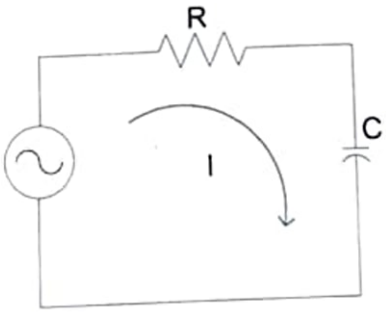
Yukarıdaki devrede gerilim kaynağı 220V ve 50Hz, R=100Ω ve L=100mH'dir. Bu devre için i akımını kompleks olarak hesaplayız ve daha sonra genliği ve açısını bulunuz.

Çözüm:

```
V=220;R=100;L=100e-3;f=50;
ZL=j*2*pi*f*L;
Z=R+ZL;
i=V/Z
genlik=abs(i)
aci=angle(i);
aci=rad2deg(aci)
```

```
i =   2.0024 - 0.6291i
genlik =    2.0989
aci =  -17.4406
```

Örnek:



Yukarıdaki devre için gerilim kaynağını 120 V, R=10Ω ve C=100μ olarak alınız. Frekansın 0 ile 800 Hz arasında değiştiğini kabul ederek, frekans-akım genliği ve frekans-akım açısı (derece cinsinden) grafiklerini çiziniz.

Çözüm:

```
V=120;R=10;C=100e-6;
f=0:800;
ZC=1./(j*2*pi*f*C);
Z=R+ZC;
i=V./Z;
```