VEKTÖRLER VE VEKTÖRLERLE İŞLEMLERİ

***Satır Vektörler**:* Bir parantez içinde vektörün elemanları yazılır ve bir değişkene atanır.

• a = [1 2 3 4 5 6 7 8 9]

Program sonucu aşağıdaki gibi döndürür

a =

1 2 3 4 5 6 7 8 9

• Örneğin zaman vektörü oluşturulurken çok sık kullandığımız gibi, 0 ile 20 arasında 2'şer

2'şer artan elemanlardan oluşan bir vektör oluşturalım.

t = 0:2:20

--> (Enter)

t =

0 2 4 6 8 10 12 14 16 18 20

• Şimdi de a kümesinin her elemanına 2 ekleyelim;

b = a+2

Sonuç;

b =

3 4 5 6 7 8 9 10 11

• İki vektörü toplayalım;

c=a+b

Sonuç;

c=

4 6 8 10 12 14 16 18 20

c(4:7) , 4- 7 arasını göster

10 12 14 16

Bir diğer gösterim yolu ise "*linspace*" komutunu kullanarak olmaktadır. Bu komutun kullanımı *"linspace(a,b,n)"* şeklindedir. a ile b arasını n elemana böl, eğer n verilmese 100 bölünür

» y=linspace(2, 4, 5)

y =

2.0000 2.5000 3.0000 3.5000 4.0000

Sütun (kolon) Vektörü:

Kolon vektörü n'e 1 boyutunda özel bir matristir.

» x=[1; 7; 18; 5]

Sembolik Matematik

$$\frac{(x^3 - 8)(x^2 + 7x)}{(2x^2 + 4x + 8)(x^2 - 2x)}$$

gibi bir ifadeyi sadeleştirebilmek için,önce ifadedeki değişken ya da değişkenleri sembolik olarak tanımlamalıyız.

Sembolik Matematik Fonksiyonları ile bazı harfli ifadeleri sadeleştirebilir, bazı limit problemlerini çözebilir, istediğimiz fonksiyonların türevlerini alabilir ya da istediğimiz fonksiyonların belirsiz integralini veya belirli integralini bulabiliriz...

Şayet sembolik değişken ve Sembolik Matematik Fonksiyonları hakkında daha geniş bilgi almak istersek komut satırına help symbolic yazabiliriz.

Örneğin;

*** x değişkenini sembolik değişken yapmak için; syms x yazabiliriz.**

*** Şayet a,b,c gibi birden çok değişkeni sembolik değişken tanımlamak istersek bunu tek tek komut satırına syms a; syms b; syms c; biçiminde yapabileceğimiz gibi tek satırda syms a b c; içiminde de yapabiliriz.**

*** Bir ifadenin tümünü sembolik olarak tanımlayıp bir değişkene de atayabiliriz.**

Örneğin; ifade yi d değişkenine sembolik olarak atama yapacaksak bunu;

Bunu d=sym('ifade') biçiminde yaparız.

Örneğin: x^2-2*x+13 ifadesini (fonksiyonunu) y değişkenine sembolik fonksiyon olarak atayalım.

>>y=sym('x^2-2*x-15')

Böylece hafızada y adında x değişkenine bağlı bir sembolik fonksiyon tanımlamış olduk.

İstersek bu fonksiyona bazı komutlar uygulayabiliriz.

Örneğin;
1) Fonksiyonu düzenli bir biçimde görüntülemek için **pretty(y)** komutunu uygulayabiliriz.

```
>>pretty(y)

     2
    x - 2 x - 15
```

2) y=0 denkleminin köklerini bulmak için **solve(y)** komutunu uygulayabiliriz.

```
>>solve(y)
ans =

[ -3]
[  5]
```

3) Fonksiyonun türevini bulmak için **diff(y)** komutunu uygulayabiliriz.

```
>>diff(y)
ans =

2*x-2
```

4) Fonksiyonun belirsiz integralini, ekranda düzenli bir biçimde görüntülemek için **int(y)** komutunu **pretty** komutuyla beraber uygalayabiliriz.

```
>>pretty(int(y))

                        3    2
                   1/3 x  - x  - 15 x
```

Harfli İfadeler ve İlgili İşlemler:

Örneğin;

$\frac{a^2 - 4b^2}{a - 2b}$ **veya** $\frac{2x-1}{x+3} + \frac{x+2}{x-1}$ **veya** $\frac{(x^3 - 8)(x^2 + 7x)}{(2x^2 + 4x + 8)(x^2 - 2x)}$ **veya**

$8 - 5\sin^2 x - 5\cos^2 x$

gibi harfli ifadeleri sadeleştirebilir ya da
$(2x^2 - \frac{3}{x})^5$ **veya** $(x^2-x+1)^3 - (x^2+x-1)^3$ **gibi bir ifadenin açılımını yapabiliriz.**

Bunun için öncelikle ifadeyi sembolik olarak tanımlamalıyız.

Harfli ifadelerde işlemlerle ilgili bazı komutlar aşağıda verilmiştir:

simplify Verilen harfli ifadeyi sadeleştirir.

pretty Harfli ifadenin sonucunu daha anlaşılır ve düzgün görünmesini sağlar.

expand Verilen üstel ifadenin açılımını yapar.

factor Açılmış olarak verilen harfli ifadeyi üstel ifade olarak verir.

Sadeleştirme İle İlgili Örnekler:

1)
$\frac{a^2 - 4b^2}{a - 2b}$ **ifadesini sadeleştirelim.**

Önce ifadeyi sembolik yapmalıyız.

>> x=sym('(a^2-4*b)/(a-2*b)')

Sonra simplify komutunu uygulayarak ifadeyi sadeleştirelim:

>>simplify(x)

>>ans =

a+2*b

Not: Daha düzgün bir görüntü elde etmek için simplify komutunu pretty komutuyla beraber kullannabiliriz.Buna göre komutu aşağıdaki gibi yazalım:

>>pretty(simplify(x))

```
        a + 2 b
```

2)
$\frac{2x-1}{x+3} + \frac{x+2}{x-1}$ **ifadesini sadeleştirelim.**

Önce ifadeyi sembolik yapmalıyız.

>> y=sym('(2*x-1)/(x+3)+(x+2)/(x-1)')

Sonra simplify komutunu pretty komutuyla beraber uygulayarak ifadeyi sadeleştirelim:

>>pretty(simplify(y))

```
            2
        3 x + 2 x + 7
        --------------
        (x + 3) (x - 1)
```

3) $8 - 5\sin^2 x - 5\cos^2 x$ **ifadesini sadeleştirelim:**

Önce ifadeyi sembolik yapalım.
>>y=sym('8-5*sin(x)^2-5*cos(x)^2')

Sonra simplify komutunu pretty komutuyla beraber uygulayarak ifadeyi sadeleştirelim:
>>pretty(simplify(y))

>>3

Üstel Bir İfadenin Açılımı:

1)

$(a-b)^5$ **binom açılımını yapalım.Önce ifadeyi sembolik yapalım.**

>>x=sym('(a-b)^5')

Sonra expand komutuyla açılımını yapalım.

>>expand(x)

ans =

a^5-5*a^4*b+10*a^3*b^2-10*a^2*b^3+5*a*b^4-b^5

Şayet komutu pretty komutuyla uygularsak;

>>pretty(expand(x))

```
 5     4         3 2        2 3      4     5
a - 5 a b + 10 a b - 10 a b + 5 a b - b
```

2)

$(2x^2 - \frac{3}{x})^5$ **binom açılımını yapalım.Önce ifadeyi sembolik yapalım.**

>>y=sym('(2*x^2-3/x)^5')

Sonra aşağıdaki komutu uygulayalım;

>>pretty(expand(y))

```
     10        7         4          8    10   243
32 x   - 240 x  + 720 x  - 1080 x + --- - ---
                                      2     5
                                     x     x
```

3)

$(x^2-x+1)^3 - (x^2+x-1)^3$ **açılımını yapalım.**

Önce ifadeyi sembolik yapalım.

y=sym('(x^2-x+1)^3-(x^2+x-1)^3')

Sonra aşağıdaki komutu uygulayalım;

>>pretty(expand(y))

```
        5      4      3      2
    -6 x  + 6 x  - 2 x  + 6 x  - 6 x + 2
```

Bir Açılımın Üstel Olarak İfadesi:

$x^3 - 3x^2 + 3x - 1$ **açılmının üstel olarak ifadesini bulalım. Önce ifadeyi sembolik yapalım.**

>>y=sym('x^3-3*x^2+3*x-1')

Sonra expand komutunu uygulayalım.

>>expand(y)

>>(x-1)^3

Not: Komutu pretty ile beraber uygulanırsak aşağıdaki görüntüyü elde ederiz.

>>pretty(expand(y))

```
           3
    (x - 1)
```

Dizi Toplamı İlgili Örnekler:

1)
$1+2+3+...+2006$ **toplamını bulduralım.**

Bu toplamı symsum(k,1,2006) biçiminde olduğunu görmeliyiz. Önce ifadedeki k değişkenini sembolik değişken olarak tanımlamalıyız.

>>syms k;

Sonra dizi-seri toplamı komutunu uygulayabiliriz.

>>symsum(k,1,2006)

ans =

2013021

2)
$7^2+8^2+9^2+...+77^2$ **toplamını bulduralım.**

Bu toplamı symsum(k^2,7,77) biçiminde ifade edebiliriz.
Önce ifadedeki k değişkenini sembolik değişken olarak tanımlamalıyız.

>>syms k;

Sonra dizi-seri toplamı komutunu uygulayabiliriz.

>>symsum(k^2,7,77)

ans =

155064

3)
$4.5.6+5.6.7+6.7.8+...+22.23.24$ **toplamını bulduralım.**

Bu toplamı symsum(k*(k+1)*(k+2),4,22) biçiminde ifade edebiliriz.
Önce ifadedeki k değişkenini sembolik değişken olarak tanımlamalıyız.

>>syms k;

Sonra dizi-seri toplamı komutunu uygulayabiliriz.

>>symsum(k*(k+1)*(k+2),4,22)

ans =

75810

4)
$(2/3)^3+(2/3)^4+(2/3)^5+...+(2/3)^{99}$ **toplamını bulalım.**

Bu toplamı symsum((2/3)^k,3,99) biçiminde ifade edebiliriz. Önce ifadedeki k değişkenini sembolik değişken olarak tanımlamalıyız.

>>syms k;

Sonra dizi-seri toplamı komutunu uygulayabiliriz.

>>symsum((2/3)^k,3,99)

ans =

**152704450587262615335745290072695420044661986328/
171792506910670443678820376588540424234035840667**

Bulnan sonuç, toplamın rasyonel sayı olarak ifadesidir.Bunu reel sayı olarak double komutundan faydalanarak yazalım.

>>double(ans)

0.8889

Dizi Toplam Formülleri:

1)
$1+2+3+...+n$ **toplam formülünü bulalım.**

Bu toplamı symsum(k,1,n) biçiminde ifade edebiliriz.

Önce ifadedeki k ve n değişkenlerini sembolik değişken olarak tanımlamalıyız.

>>syms k n;

>> symsum(k,1,n)

ans =

1/2*(n+1)^2-1/2*n-1/2

Şayet görüntünün daha dügün görünmesini istersek, komutu pretty komutuyla birlikte kullanmalıyız.

>> pretty(symsum(k,1,n))

$$1/2\ (n + 1)^2 - 1/2\ n - 1/2$$

2)

$1^2+2^2+3^2+\ldots+n^2$ **toplam formülünü bulalım.**

Bu toplamı symsum(k^2,1,n) biçiminde ifade edebiliriz.

Önce ifadedeki k ve n değişkenlerini sembolik değişken olarak tanımlamalıyız.

>>syms k n;

Sonra pretty komutu ile symsum komutunu beraber uygulayalım.

>> pretty(symsum(k^2,1,n))

$$1/3\ (n + 1)^3 - 1/2\ (n + 1)^2 + 1/6\ n + 1/6$$

3)

$1 + 3 + 3^2 + 3^3 + \ldots + 3^{n-1}$ **toplam formülünü bulalım.**

Bu toplamı symsum(3^k,0,n-1) biçiminde ifade edebiliriz.

Önce ifadedeki k ve n değişkenlerini sembolik değişken olarak tanımlamalıyız.

>>syms k n;

Sonra pretty komutu ile symsum komutunu beraber uygulayalım.

>> pretty(symsum(3^k,0,n-1))

$$1/2\ 3^n - 1/2$$

Seriler İle İlgili Örnekler:

1)

$1 + 2/3 + (2/3)^2 + (2/3)^3 + \ldots$ **sonsuz toplamının (serisinin) toplamını bulalım.**

Bu toplamı symsum((2/3)^k,0,Inf) biçiminde ifade edebiliriz. Buradaki Inf ifadesi sonsuz anlamına gelir.

Önce ifadedeki k değişkenlerini sembolik değişken olarak tanımlamalıyız.

>>syms k;

Sonra symsum((2/3)^k,0,Inf) komutunu uygulayalım.

>>symsum((2/3)^k,0,Inf)

ans =

3

3)

$$\frac{1}{1^2}+\frac{1}{2^2}+\frac{1}{3^2}+\cdots$$ **sonsuz toplamının (serisinin) toplamını bulalım.**

Bu toplamı symsum(1/k^2,1,Inf) biçiminde ifade edebiliriz.Buradaki Inf ifadesi sonsuz anlamına gelir.

Önce ifadedeki k değişkenlerini sembolik değişken olarak tanımlamalıyız.

>>syms k;

Sonra pretty komutu ile symsum komutunu beraber uygulayalım.

>>pretty(symsum(1/k^2,1,Inf))

```
          2
  1/6 pi
```

Polinomlar:

Çok terimli anlamına gelen polinomu tanımlarsak;

a0,a1,a2,…..an ve x belirsiz değişken olmak üzere

$P(x)=a_nx^n+a_{n-1}x^{n-1}+....+a_1x+a_0$ şeklinde gösterilir

Matlab'ta polinomlar bir vektörle temsil edilirler. Polinom oluşturmak için yüksekten düşük dereceliye doğru azalan sırada polinom katsayıları yazılır.

Polinomun Belli Noktadaki Değeri

• $x=s^4+3s^3-15s^2-2s+9$ polinomu programa aşağıdaki şekilde yazılır;
x=[1 3 -15 -2 9]
-->
x =
1 3 -15 -2 9
polinomun **s=2 için değeri**

>> polyval(x,2)
-15

• Benzer şekilde $y=s^4+1$'in gösterilimi y=[1 0 0 0 1] şeklindedir.
• Polinomun herhangi bir kök için değeri, örneğin s4+1'in s=2 için değeri;

z=polyval([1 0 0 0 1],2) veya doğrudan z=polyval(y,2)

z =
17

• Polinomun köklerinin bulunması,

örneğin $s^4+3s^3-15s^2-2s+9$ için;
roots([1 3 -15 -2 9])
ans=
-5.5745
2.5836
-0.7951
0.7860

• İki polinomun çarpılması,
a(x)= (x+2)
$b(x)=(x^2+4x+8)$
$a(x)*b(x)=x^3+6x^2+16x+16$
a=[1 2]
b=[1 4 8]
z=conv(a,b)
-->
z=
1 6 16 16
• İki polinomu bölelim
[xx,R]=deconv(z,y)
xx=

1 2 (bölüm=x+2)
R=
0 0 0 0 (kalan=0)

● İki polinomu toplamak ;
Aynı dereceden polinomlar toplanır.
$a(x)=-2x^2+10x-4$ $b(x)=5x^2-x+2$

a=[-2 10 -4];b=[5 -1 2]

t=a+b
t =
   3   9   -2

● Bir polinomdan diğerini çıkartmak ;

c=a-b
c =
  -7   11   -6

Denklemlerin Çözümü İle İlgili Örnekler:

1)
$15 - x = 3.(5 - x) + 14$ **denkleminin çözüm kümesini bulalım.**

Önce ifadeyi sembolik olarak tanımlayıp, solve komutunu uygulamalıyız.

>>y=sym('15-x=3*(5-x)+14')

>>solve(y)

7

Not: Aynı işlemi tek komutla;

>>solve('15-x=3*(5-x)+14')   **ile de yapabiliriz.**

2)
$x^3 + 3x^2 - x - 3 = 0$ **denkleminin çözüm kümesini bulalım.**

İfadeyi sembolik olarak tanımlayıp, solve komutunu uygulayalım.

>> solve('x^3+3*x^2-x-3=0')

ans =

[1]
[-3]
[-1]

3)
$2x^2+5x=3$ denkleminin çözüm kümesini bulalım.

İfadeyi sembolik olarak tanımlayıp, solve komutunu uygulayalım.

```
>> solve('2*x^2+5*x=3')

ans =

[ -3]
[ 1/2]
```

4)
$x^5 = 16x$ denkleminin çözüm kümesini bulalım.

İfadeyi sembolik olarak tanımlayıp, solve komutunu uygulayalım.

```
>>solve('x^5=16*x')

ans =

[     0]
[     2]
[    -2]
[  2*i]
[ -2*i]
```

5)
$x^2 - 6x - 3 = \sqrt{3x - 5}$ denkleminin çözüm kümesini bulalım.

İfadeyi sembolik olarak tanımlayıp, solve komutunu uygulayalım.

```
>>solve('x^2-6*x-3=(3*x-5)^(1/2)')

ans=
      7
```

6)
$ax^2+bx+c=0$ **denkleminin köklerini veren formülleri bulduralım.**

İfadeyi sembolik olarak tanımlayıp, solve komutunu uygulayalım. Sonuçların düzenli görünmesi için pretty komutuyla uygulayalım.

>>pretty(solve('a*x^2+b*x+c=0'))

```
[            2        1/2]
[     -b + (b  - 4 a c)    ]
[1/2 --------------------  ]
[            a            ]
[                         ]
[            2        1/2]
[     -b - (b  - 4 a c)    ]
[1/2 --------------------  ]
[            a            ]
```

7) sin 2x = cos x trigonometrik denkleminin çözüm kümesini bulalım.

İfadeyi sembolik olarak tanımlayıp, solve komutunu uygulayalım. Sonuçların düzenli görünmesi için pretty komutuyla uygulayalım.

>>pretty(solve('sin(2*x)=cos(x)'))

```
[ 1/2 pi ]
[        ]
[- 1/2 pi]
[        ]
[ 1/6 pi ]
[        ]
[ 5/6 pi ]
```

Fonksiyon Değeri İle İlgili Örnekler:

1)

$$f(x) = \frac{x^2 - 2x + 7}{3x+5}$$

fonksiyonu için f(2006) değerini hesaplatalım.

Önce x değişkenini sembolik değişken olarak tanımlayalım:

>>syms x

Sonra da fonksiyonu sembolik olarak tanımlayıp bir y gibi değişkene atayalım;

>>y=sym('(x^2-2*x+7)/(3*x+5)');

Sonra da tanımlı fonksiyonu x=2006 için hesaplatalım;

>>subs(y,x,2006)

ans=

667.4466

2) f(x)=2x+5 fonksiyonu ile g(x)=3x-4 fonksiyonları verilsin. g(f(x)) bileşke fonksiyonunu bulduralım.

Önce f(x) ve g(x) fonksiyonundaki x değişkenini sembolik değişken olarak tanımlayalım:

>>syms x;

f(x) fonksiyonunu sembolik fonksiyon olarak tanımlayıp y değişkenine atayalım;

y=sym('2*x+5');

g(x) fonksiyonunu sembolik fonksiyon olarak tanımlayıp z değişkenine atayalım;

z=sym('3*x-4');

Son olarak ta g(f(x)) i subs komutu kullanarak bulalım;

>>subs(z,x,y)

6*x+11

Şayet sonucu düzenli görmek istersek; pretty komutunu da uygulamalıyız.

>>pretty(subs(z,x,y))

6x+11

3) $f(x) = \dfrac{x^2 - 2x + 7}{3x+5}$ **ve** $g(x) = \dfrac{x^3 - 2}{x + 3}$ **fonksiyonları verildiğine göre; f(g(x)) bileşke fonksiyonunu bulduralım.**

Önce f(x) ve g(x) fonksiyonundaki x değişkenini sembolik değişken olarak tanımlayalım:

>>syms x;

f(x) fonksiyonunu sembolik fonksiyon olarak tanımlayıp y değişkenine atayalım;

y=sym('(x^2-2*x+7)/(3*x+5)');

g(x) fonksiyonunu sembolik fonksiyon olarak tanımlayıp z değişkenine atayalım;

z=sym('(x^3-2)/(x+3)');

Son olarak ta f(g(x)) i subs komutu kullanarak bulalım;

>>subs(y,x,z)

((x^3-2)^2/(x+3)^2-2*(x^3-2)/(x+3)+7)/(3*(x^3-2) /(x+3)+5)

Şayet sonucu daha düzgün görmek istersek; pretty komutunu da uygulamalıyız.

>>pretty(subs(y,x,z))

```
   3     2      3
 (x  - 2)      x  - 2
 --------- - 2 ------ + 7
         2     x + 3
 (x + 3)
 ------------------------
         3
        x  - 2
      3 ------ + 5
        x + 3
```

Şayet sonucun sadeleşmiş düzenli biçimini görmek istersek; bir de

simplify komutunu da kullanmalıyız.

>>pretty(simplify(subs(y,x,z)))

```
 6        3            4        2
x  - 10 x  + 79 - 2 x  + 46 x + 7 x
-----------------------------------
            3
    (3 x  + 9 + 5 x) (x + 3)
```

Denklem Sistemleri:

1)

$2x-3y=27$
$5x+2y=1$ **denklem sisteminin çözüm kümesini bulalım.**

Önce denklem sisteminde bulunan değişkenleri sembolik değişken yapalım.

>>syms x y;

Sonra denklem sisteminin çözümlerini solve komutuyla buldurup bir değişkene; örneğin sonuc değişkenine atayalım.

>>sonuc=solve('2*x-3*y=27','5*x+2*y=1');

Sonra x ve y değerlerini ekranda gösterelim.

>>sonuc.x
ans =
3

>> sonuc.y
ans =
-7

2)

$17x-3y+4z=7$
$15x-7y \quad =1$
$x + y - 9z =13$ **denklem sisteminin çözüm kümesini bulalım.**

Önce denklem sisteminde bulunan değişkenleri sembolik değişken yapalım.

```
>>syms x y z;
```

Sonra denklem sisteminin çözümlerini solve komutuyla buldurup bir değişkene; örneğin sonuc değişkenine atayalım.

```
>>sonuc=solve('17*x-3*y+4*z=7','15*x-7*y=1','x+y-9*z=13');
```

Sonra x ,y ve z değerlerini ekranda gösterelim.

```
>> sonuc.x

ans =

391/377

>> sonuc.y

ans =

784/377

>> sonuc.z

ans =

-414/377
```

3)

$$\begin{aligned} x^2-2xy+3y^2&=17 \\ xy-3x+5&=0 \end{aligned}$$

denklem sistemini çözelim.

Önce denklem sisteminde bulunan değişkenleri sembolik değişken yapalım.

```
>>syms x y;
```

Sonra denklem sisteminin çözümlerini solve komutuyla buldurup bir değişkene;
örneğin sonuc değişkenine atayalım.

```
>>sonuc=solve('x^2-2*x*y+3*y^2=17','x*y-3*x+5=0');
```

Sonra x ve y değerlerini ekranda gösterelim.

```
>>sonuc.x
ans =

[         1]
[         5]
[  i*15^(1/2)]
[ -i*15^(1/2)]

>> sonuc.y

ans =

[             -2]
[              2]
[  1/3*i*15^(1/2)+3]
[ -1/3*i*15^(1/2)+3]
```

Şayet sonuçları daha düzgün görünmesini istersek pretty komutunu da uygulamalıyız.

```
>> pretty(sonuc.x)

                    [  1      ]
                    [         ]
                    [  5      ]
                    [         ]
                    [     1/2 ]
                    [i 15     ]
                    [         ]
                    [      1/2]
                    [-i 15    ]
>> pretty(sonuc.y)

                  [      -2          ]
                  [                  ]
                  [       2          ]
                  [                  ]
                  [           1/2    ]
                  [ 1/3 i 15    + 3  ]
                  [                  ]
                  [            1/2   ]
                  [- 1/3 i 15     + 3]
```

• Polinomun türevini almak ;

1)

m=$12x^4+8x^3-3x^2+2x-1$

m=[12 8 -3 2 -1]

>>f=polyder(m) m polinomunun 1. türevi

```
f =

    48    24    -6     2
```

>> polyder(f) m polinomunun 2. türevi

ans =

144 48 -6

>> polyder(m,2) m polinomunun 1.türevinin 2 ile çarpımı

ans = 96 48 -12 4

diff fonksiyonuyla türev almak

```
>> syms x;
>> y=sym('12*x^4+8*x^3-3*x^2+2*x-1')

y =

12*x^4+8*x^3-3*x^2+2*x-1

>> diff(y)

ans =

48*x^3+24*x^2-6*x+2
```

>> pretty(diff(y)) Düzgün(güzel) göstermek için **pretty** fonksiyonu uygulanır.

```
                 3        2
             48 x  + 24 x  - 6 x + 2
>>

>> diff(y,2)
```

ans =

144*x^2+48*x-6

>> diff(y,3)

ans =

288*x+48

>> diff(y,4)

ans =

288

2)

$y = \frac{x^2-3x+7}{x^2+5x-1}$ **fonksiyonunun türevini bulalım.**

Önce x değişkenini sembolik yapalım.

>>syms x;

Sonra fonksiyonu sembolik olarak y değişkenine atayalım.

>>y=sym('(x^2-3*x+7)/(x^2+5*x-1)');

Sonra da pretty ve diff komutuyla y nin türevini alalım.

>>pretty(diff(y))

```
                      2
  2 x - 3       (x  - 3 x + 7) (2 x + 5)
------------ - -------------------------
  2                     2          2
 x  + 5 x - 1       (x  + 5 x - 1)
```

3)

$y=\sin^2 x.\cos x$ **fonksiyonunun türevini bulalım.**

Önce x değişkenini sembolik yapalım.

>>syms x;

Sonra fonksiyonu sembolik olarak y değişkenine atayalım.

>>y=sym('sin(x)^2*cos(x)');

Sonra da pretty ve diff komutuyla y nin türevini alalım.

>>pretty(diff(y))

```
              2         3
2 sin(x) cos(x)  - sin(x)
```

4)

$f(x) = x \ln^2 x$ **fonksiyonunun türevini bulalım.**

Önce x değişkenini sembolik yapalım.

>>syms x;

Sonra fonksiyonu sembolik olarak y değişkenine atayalım.

>>y=sym('x*log(x)^2');

Sonra da pretty ve diff komutuyla y nin türevini alalım.

>>pretty(diff(y))

```
      2
log(x)  + 2 log(x)
```

5)

$f(x)=(x^3 - 3x + 5)^7$ **fonksiyonunun türevini bulalım.**

Önce x değişkenini sembolik yapalım.

>>syms x;

Sonra fonksiyonu sembolik olarak y değişkenine atayalım.

>>y=sym('(x^3-3*x+5)^7');

Sonra da pretty ve diff komutuyla y nin türevini alalım.

>>pretty(diff(y))

```
           3           6      2
     7 (x  - 3 x + 5)  (3 x  - 3)
```

Polinom Köklerini Bulma ve Kökü verilen Polinomu Bulma

Roots fonksiyonu ;

m=$12x^4+8x^3-3x^2+2x-1$ fonksiyonu için;

fn=[12 8 -3 2 -1]

kokler=roots(fn)

kokler =

```
  -1.0965
   0.0263 + 0.4481i
   0.0263 - 0.4481i
   0.3772
```

Bulundu,sağlaması yapalım, köklerden polinom oluşturalım;

>> **poly**(kokler)

ans =

```
    1.0000    0.6667   -0.2500    0.1667   -0.0833
```

>> 12*ans

ans =

```
   12.0000    8.0000   -3.0000    2.0000   -1.0000
```

Başlangıçtaki denklem elde edildi.

Örnek-2 :

$d(x)=x^4-15x^2+10x+24$

>> d=[1 0 -15 10 24];
>> roots(d)

ans =

-4.0000
3.0000
2.0000
-1.0000

>> poly(ans)

ans =

1.0000 0.0000 -15.0000 10.0000 24.0000

Polinomun İntegralini Almak

$f(x)=3x^2+2x+1=0$ polinomunun integralini alalım,
f=[3 2 1]

>>polyint(f)
ans =

1 1 1 0 denklem katsayıları
Buna göre yukarıdaki polinom integrali,

$X^3+x^2+x=0$ olarak bulunur, bu polinomun türevini alalım

>>polyder(ans)

ans =

3 2 1 , yukarıdaki integralini aldığımız f polinoma ulaştık,

Örnekler;

1) int fonksiyonu ile integral alma

$\int (3x^2 - 2x + 5)\, dx$ **belirsiz integralini bulalım.**

Önce fonksiyonu sembolik olarak tanımlayıp y ye atayalım.

>>y=sym('3*x^2-2*x+5');

Sonra int komutuyla belirsiz integralini alalım.

>>int(y)

ans = x^3-x^2+5*x

Sonucu daha düzenli görmek istersek pretty komutunu da kullanmalıyız.

>>pretty(int(y))

```
  3   2
 x - x + 5 x
```

2)

$$\int \frac{2x+5}{x^2+1}\,dx$$ **belirsiz integralini bulalım.**

Önce fonksiyonu sembolik olarak tanımlayıp y ye atayalım.

>>y=sym('(2*x+5)/(x^2+1)');

Sonra pretty komutuyla beraber int komutunu kullanarak belirsiz integralini alalım.

>>pretty(int(y))

```
      2
log(x  + 1) + 5 atan(x)
```

3)

$$\int x^2 \sin x\, dx$$ **belirsiz integralini bulalım.**

Önce fonksiyonu sembolik olarak tanımlayıp y ye atayalım.

>>y=sym('x^2*sin(x)');

Sonra pretty komutuyla beraber int komutunu kullanarak belirsiz integralini alalım.

>>pretty(int(y))

```
    2
-x  cos(x) + 2 cos(x) + 2 x sin(x)
```

Belirli İntegral İle İlgili Örnekler:

1)

$$\int_0^1 3x^2 dx$$

belirli integralin değerini hesaplayalım.

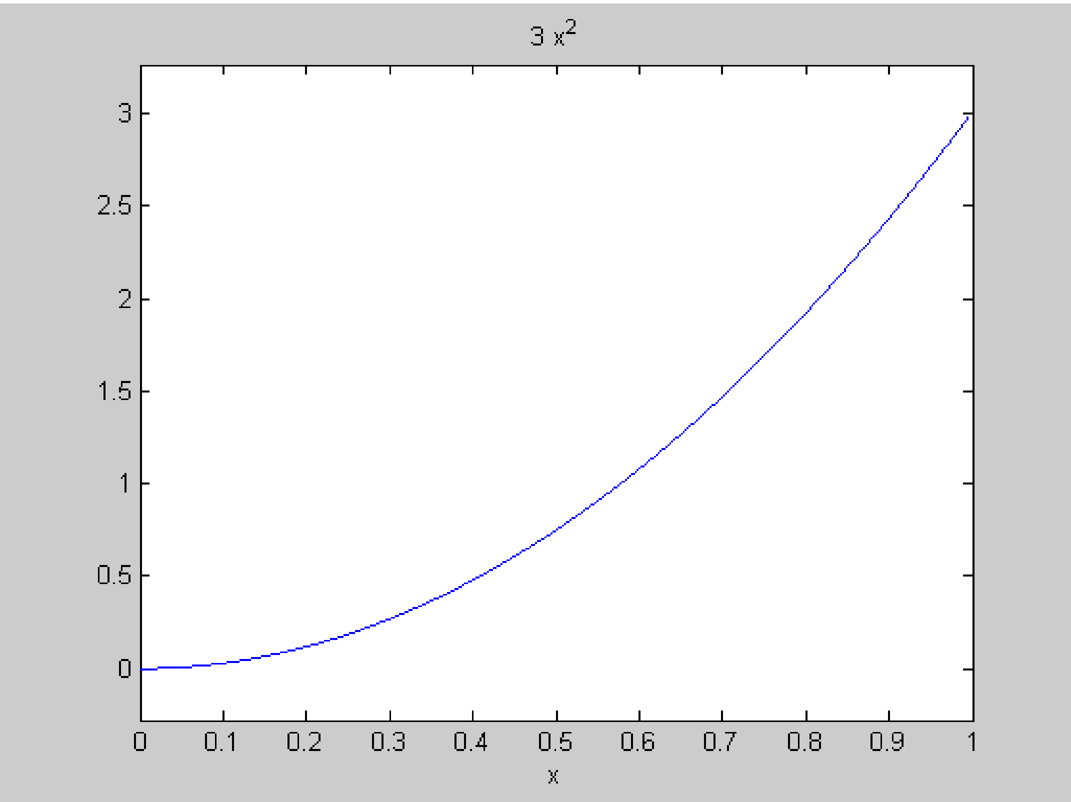
f='3*x^2'

>>int('3*x^2',0,1)

ans = 1

Grafiğini çizmek istersek

>> ezplot(f,[0,1])



2)

$$\int_1^{e^3} x.\ln x dx$$

belirli integralin değerini hesaplayalım.

>>int('x*log(x)',1,exp(3))

ans=
7990730503444001487616492393225/396140812571321687967 71975168*log(2826788018837635)-
3755643336618680699179751424 81575/396140812571321687967 71975168*log(2)-
7970923462815435403218106405641/792281625142643375935 43950336

Yukarıdaki sonucun gerçek değerini bulmak için double komutundan faydalanırız.

>> double(ans)

ans =

504.5360

Yamuk Kuralıyla integral Alma

trapez(x,y) Sayısal integralde trapez yöntemiyle integral hesaplanır

$$\int_0^1 e^{-x^2}\,dx$$

x=0:0.01:1;
y=3*x.^2
trapz(x,y)

ans= 0.7468

Çift Katlı Integral

Matlab da f(x,y) gibi iki değişkenli fonksiyonların integrali

dblquad(fonksiyon,xmin,xmax,ymin,ymax) şeklinde hesaplanır.

$$\int_0^1 \int_{-1}^2 3x^2 + y^3 \mathrm{dxdy}$$

>>dblquad('3*x.^2+y.^3',0,1,-1,2)

ans= 6.7500

Limit Hesaplama

limit fonksiyonu : limit(fonksiyon,limit sınırı)

$f=4x^3-8x^2+5x+17$ fonksiyonu x değeri 0 ve 3 yaklaşırken değeri,

f=sym('4*x^3-8*x^2+5*x+17')

```
>>limit(f)      0 daki değeri
ans=17

>>limit(f,3)   3 yaklaşırken değeri
ans=68
```

Verilere Göre Denklem ve Eğri Uydurma

Verilen sayı dizilerini temsil edecek eğrinin ,istenen dereceden polinom katsayıları hesaplanmak suretiyle eğri uydurma yapılabilir.

Polyfit(x,y,n),
x ve y değerler dizisi , n ise hesaplanacak polinomun derece katsayısı,

```
>> format long
```

Örnek 1:
x = [0 5 10 15 20 25 30];
y = [14.6 12.8 11.3 10.1 9.09 8.26 7.56];
olsun ve n=2 olsun

```
>> p = polyfit(x,y,2)

p =
0.00439523809524    -0.36335714285714    14.55190476190477
```

Böylece, en uygun parabol denklemi aşağıda bulunur;

$P(x) = 0.00439523809524x^2-0.36335714285714x+14.55190476190477$

fv=polyval(p,x)   fonksiyonun x noktasındaki değerlerini bulduk, fv =

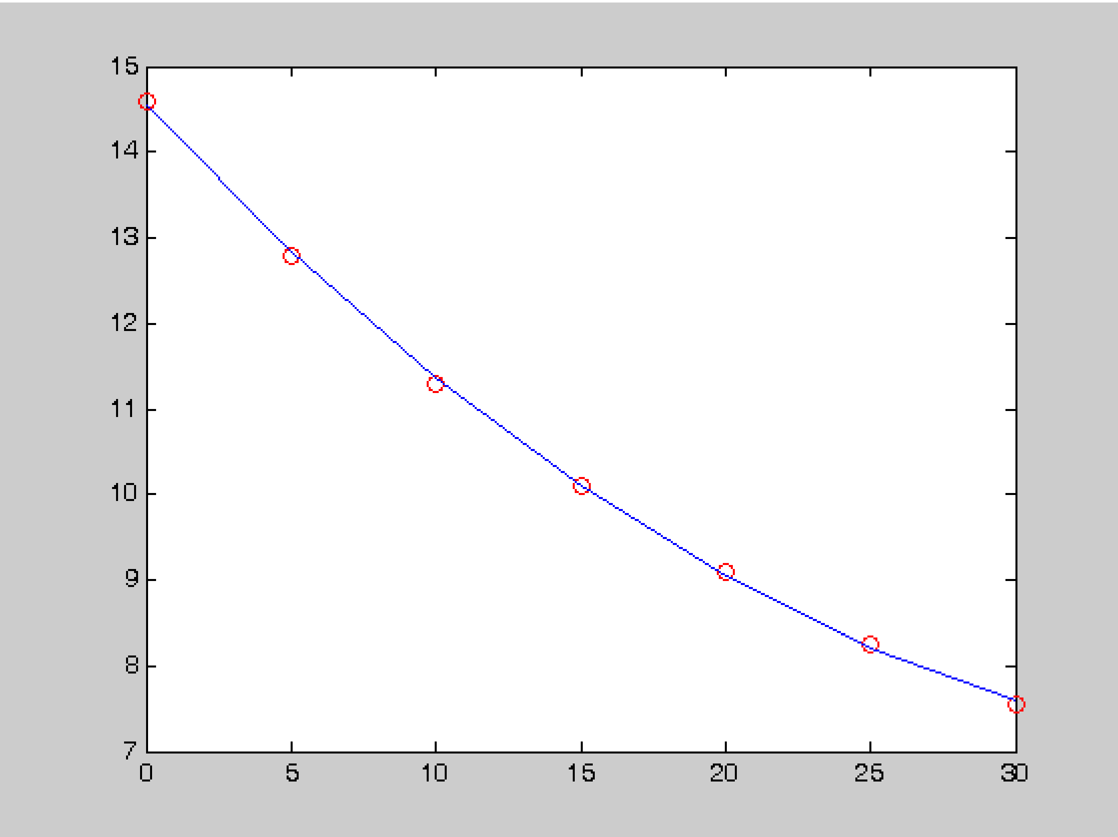
  14.55190476190477 12.84500000000000 11.35785714285715

  10.09047619047619   9.04285714285714   8.21500000000000

   7.60690476190477

Ve eğrimizi çiziyoruz.

```
>> plot(x,y,'or')
>> hold on            grafik silinmesin.
>> plot(x,fv)
```



Örnek 2:

x=1:12;
y=[15 13 14 12 10 13 15 18 19 20 23 20];

Olsun , bu verilere göre 2 faklı (2 ve 5. derece) dereceden polinoma göre eğri uyduralım.

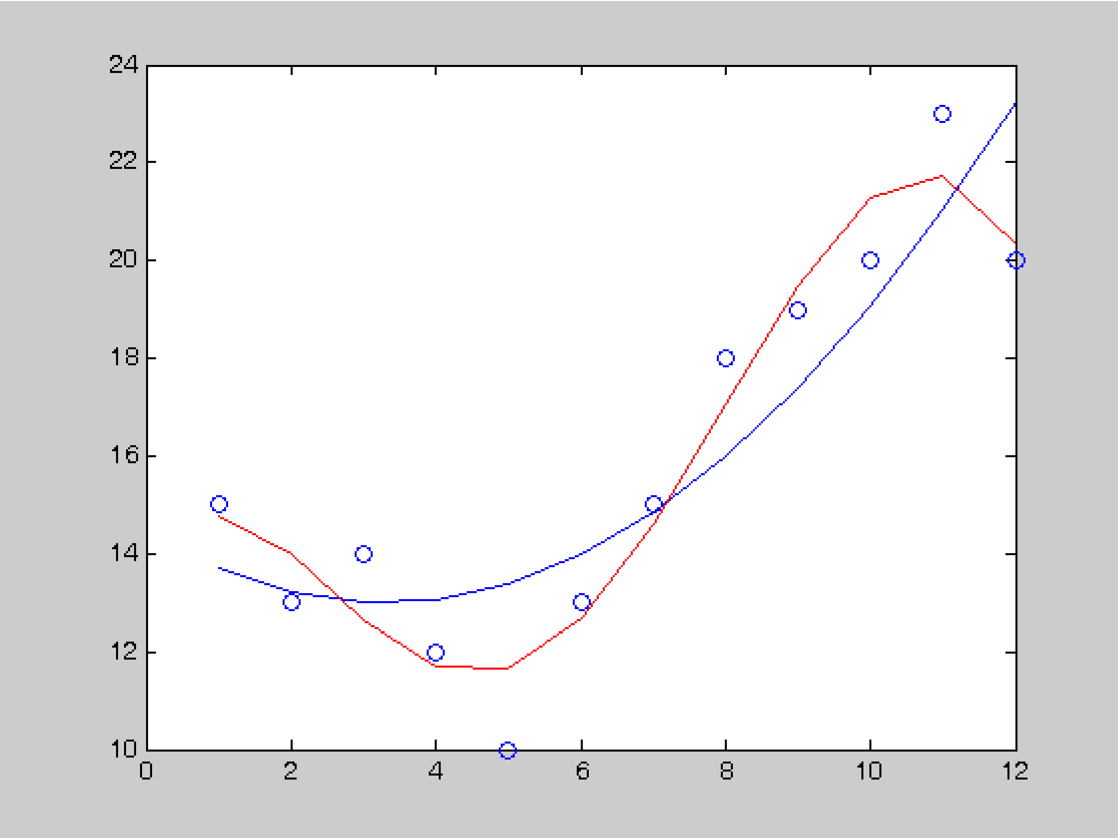
```
>> format short
>> fv2=polyfit(x,y,2)

fv2 =

   0.1349   -0.8861   14.4545
```

$fv2(x)=0.1349x^2-08861x+14.4545$

```
fv5=polyfit(x,y,5);

fv5 =

   0.0006   -0.0277    0.3992   -2.0436    2.9555   13.5000
```

$fv5(x)= 0.0006x^5-0.0277x^4+ 0.3992x^3-2.0436x^2+2.9555x+13.5000$

```
fv2val=polyval(fv2,x);
fv2val =

  13.7033   13.2218   13.0100   13.0679   13.3956   13.9930

  14.8601   15.9970   17.4036   19.0799   21.0260   23.2418

fv5val=polyval(fv5,x);
fv5val =

   14.7839   14.0061   12.6550   11.6994   11.6606   12.6849

   14.6158   17.0666   19.4929   21.2646   21.7387   20.3314

>> plot(x,y,'ob')
>> hold on
>> plot(x,fv2val)
>> hold on
>> plot(x,fv5val,'r')
```



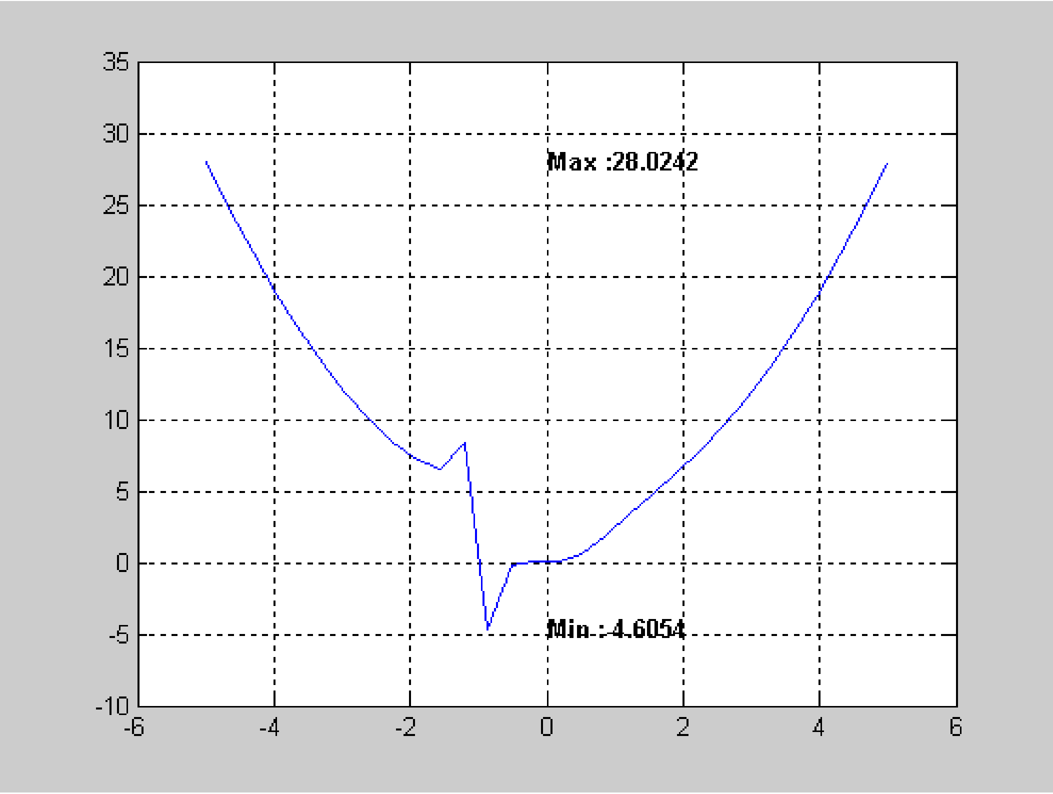
Bir Fonksiyon Grafiğinin Min ve Max Noktalarını Bulmak

$$y = x^2 + 3 - \frac{3}{x^3 + 1}$$

Yukarıda verilen fonksiyonun grafiği -5 ve 5 aralığında çizerek max. Ve min. Noktalarını grafik üzerinde belirtelim.

```
x=linspace(-5,5,30)

y=(x.^2+3)-3./(x.^3+1)

[a,k1]=min(y)

[b,k2]=max(y)

plot(x,y)

grid on

axis([-6 6 -10 35])

text(0,a,strcat('\bfMin :',num2str(a)))

text(0,b,strcat('\bfMax :',num2str(b)))
```

axis([x1 xn y1 yn]) → eksenlerdeki değerlerin başlangıç ve son değerlerini belirtir.



Özel (special) fonksiyonlar; specfun

Bunlardan bazıları ve anlamları aşağıda verilmiştir.

factor(n) **n sayısının asal çarpanlarını bulur.**

isprime(n) **n sayısının asal olup olmadığını denetler, asal ise 1 değilse 0 değerini döndürür.**

primes(n) **n sayısına kadar olan asal sayıları listeler**

gcd(a,b) **a ile b sayılarının OBEB ini bulur**

lcm(a,b) **a ile b sayılarının OKEK ini bulur**

rats(a) **a sayısını rasyonel sayıya çevirir.**

perms(a) **a stringinin permütasyonlarını bulur.**

factorial(n) **n faktöryeli (n!) bulur.**

nchoosek(n,r) **n nin r li kombinasyon-larının sayısını bulur.**

sqrt(x) x in karekökü

sin(x) Radyan cinsinden x in sinüsü

cos(x) Radyan cinsinden x in cosinüsü

tan(x) Radyan cinsinden x in tanjantý

cot(x) Radyan cinsinden x in cotanjantý

acos(x) arccosx

asin(x) arcsinx

atan(x) arctanx

acot(x) arccotx

exp(x) e^x

log(x) ln(x)

log10(x) Logx

abs(x) x (x in mutlak değeri)

fix(x) x in yukarıya yuvarlanmışı

ceil(x) x in aşağıya yuvarlanmışı

floor(x) x in tamdeğeri,

sign(x) x in işareti

round(x) x e en yakın tamsayıya yuvarlar.

mod(x,y) x in y modundaki değeri

rem(x,y) x in y ye bölümünden kalan

vpa(x, y) **variable-precision arithmetic. x işlemini y duyarlılıkta yazar vpa('sqrt(2)', 50)**

```
ans = 1.4142135623730950488016887242096980785696718753769
```

DOĞRUSAL OLMAYAN DENKLEMLERİN ÇÖZÜMÜ

Doğrusal olmayan denklemlerin doğrusal denklemlerde olduğu gibi tek bir standart biçimi yoktur. Gerek MATLAB içinde gerekse Otimatization Toolbox içinde, gerek tek değişkenli ve gerekse çok değişkenli denklemlerin çözümünde kullanılan çeşitli çözüm fonksiyonları vardır. Doğrusal olmayan denklemlerin çözümü, doğrusal denklem çözümü kadar basit olmayıp bunların çözümü için ayrıca bir fonksiyon dosyası hazırlanması gerekir.

Burada, MATLAB içinde yer alan fzero fonksiyon fonksiyonu ile Otimatization Toolbox içinde yer alan fsolve fonksiyon fonksiyonu ayrıntılı bir biçimde ele alınacaktır. Ayrıca diğer doğrusal olmayan fonksiyon fonksiyonlarının kısaca tanımları gözden geçirilecektir.

fzero: Fonksiyon fonksiyonu; tek değişkenli bir fonksiyonun sıfırını hesaplar. Genel kullanım biçimleri aşağıda olduğu gibidir.

```
z=fzero(‘function’,X0);
z=fzero(‘function’,X0,tol);
z=fzero(‘function’,X0,tol, trace);
```

fun(x) biçimindeki bir fonksiyonun, X0 ile tanımlanan değere yakın olan tek bir sıfırını hesaplar. Burada, fonksiyonu sıfır yapan, yani x eksenini kesen bir sıfır değeri hesaplanır.

İkinci bildirimde yer alan tol isimli, seçimli argüman bağıl hata toleransını belirler. Üçüncü bildirimde yer alan seçimli trace argümanı her bir hesap yineleme işlemindeki bilgileri görüntüler.

Fonksiyon fonksiyonu olan fzero fonksiyonunu kullanmak için ayrıca function ile başlayan bir fonksiyon dosyası hazırlanması gerekir.

Örnek: $f(x)=x^3-2x-5$ fonksiyonunun bir sıfırını bulunuz.

Çözüm: Önce bir fx.m adı ile fonksiyon dosyası hazırlanır.

```
Function y=fx(x)
y=x^3-2*x-5
```

Burada dosya adı ‘fx’ ile fonksiyon adı ‘fx’ aynı olması gerektiğine dikkat edilmelidir. Daha sonra MATLAB ortamında

z=fzero(‘fx’,2) bildirimi ile

z=2.0946

sonucu elde edilir. Burada X0=2 olarak tahmini bir başlangıç değer verilmiştir.

f (x) fonksiyonu gerçekten bir polinom olduğuna göre aşağıda verilen roots komutu ile

```
p=roots([1  0  -2  -5])
```

Aynı fonksiyonu sıfır yapan gerçek değer ile birlikte karmaşık eşlenik kökleri de;

```
p=
  2.0946
-1.0473   + 1.1359i
-1.0473   - 1.1359i
```

olarak elde edilmiş olur

Örnek : e^{2x} –x –2 biçiminde verilen doğrusal olmayan fonksiyonun bir adet sıfırını bulunuz.

Çözüm: Burada f(x) fonksiyonu;

$f(x)=e^{2x}-x-2$ biçimine sokulabildiğine göre fonksiyon dosyası;

```
function y=fex(x)
y=exp(2*x)-x-2;
```

biçiminde hazırlanır. Daha sonra MATLAB ortamında;

```
z=fzero('fex',1)
z0.4475
```

elde edilir.

Yukarıdaki örneklerden de görüldüğü gibi fzero fonksiyonu herhangi bir fonksiyonun tahmini bir sıfırının hesaplanmasında ve/veya doğrusal olmayan denklemlerin çözümünde daha kullanışlıdır. Doğrusal denklemlerin aynı anda tüm köklerini çözmek gerektiğinde roots fonksiyonunu kullanmak daha pratik olacaktır.

DOĞRUSAL OLMAYAN DENKLEM TAKIMLARININ ÇÖZÜMÜ

Doğrusal olmayan denklem takımlarının çözümünde, Optimization Toolbox içinde yer alan fsolve fonksiyon fonksiyonu kullanılır. fsolve fonksiyonu doğrusal olmayan denklem takımının çözümünü sağlar.

fsolve fonksiyonunun belli başlı kullanım biçimleri aşağıda olduğu gibidir.

```
x=fsolve('fun',x0)
x=fsolve('fun',x0,options)
x=fsolve('fun',x0,options,'grad')
x=fsolve('fun',x0,options,'grad',p1,p2, ...)
[x,options]=fsolve('fun',x0, ...)
```

fsolve doğrusal olmayan denklemlerin köklerini hesaplar. Çıkış argümanı olan X değerleri; F(x)=0 şeklinde hesaplanır. Burada F(x) ve X skalar, vektör veya matrislerden ibaret olabilir.

x=fsolve('dun',x0) bildirimi, fun.m isimli M-dosyasında tanımlanan denklemleri, X0 tahmini başlangıç değerlerinden başlayarak çözer ve sonucu X değişkenine atar. Burada X0 boyutu x değişken sayısı kadar olmalıdır.

İkinci bildirimde yer alan seçimli argüman options seçimli parametreler vektörünü tanımlar. options için pek çok seçenek mevcuttur. Bunlar ile ilgili bilgiler help folve yolu ile sağlanabilir.

Üçüncü bildirimde yer alan grad, X noktasında fonksiyonların kısmi türevlerini (Jacobianlarını) df/dx, df=grad(x) elde etmek için kullanılır. df'in i'inci sütunu f'deki fonksiyonun i'inci kısmı türevine karşılık gelir.

attgoal	Çoklu-amaçlı hedefe ulaşma problemi çözümü
constr	Kısıtlı minimizasyon çözümü
fmin	Skalar kısıtsız minimizasyon çözümü.
fminu, fmins	Kısıtsız minimizasyon çözümü.
Fsolve	Doğrusal olmayan denklem çözümü.
leastssq	Doğrusal olmayan en küçük kareler çözümü.
minimax	Minimum-maksimum çözümü.
seminf	Yarı mutlak minimizasyon
lp	Doğrusal programlama
nnls	Negatif olmayan en küçük kareler çözümü.
qp	Eğrisel programlama.

MATLAB'TA DİFERANSİYEL DENKLEM ÇÖZÜMLERİ

Birinci dereceden bir diferansiyel denklemin genelleştirilmiş biçimi

$$y' = \frac{dy}{dx} = g(x,y)$$

şeklinde ifade edilebilir. Burada; x bağımsız değişken ve y de bağımlı değişkendir. Birinci dereceden bir diferansiyel denklemin çözümü $f'(x) = g(x,y)$ fonksiyonundan elde edilen y=f(x) şeklinde bir fonksiyondur. Bir diferansiyel denklem çözümünün sayısal hesaplaması y' türevinden y nin elde edilmesi için gerekli integrasyon işlemini ($y = \int_{x=a}^{x=b} g(x,y)dx$) kapsar. Bir diferansiyel denklemin çözümü genellikle bir fonksiyonlar ailesi biçimindedir. Tek bir özel çözümün hesaplanabilmesi için sınır koşullarının (x=a, x=b) tanımlanması gerekir.

MATLAB'TA DİFERANSİYEL DENKLEMLERİN SAYISAL ÇÖZÜMÜ

MATLAB'ta diferansiyel denklemlerin sayısal çözümünde kullanılan; ode23 ve ede45 adında iki adet Ruge-Kutta fonksiyonu mevcuttur. Bunlardan; ode23 fonksiyonu ikinci ve üçüncü dereceden Runge-Kutta integrasyon denklemlerini, ode45 fonksiyonuda dördüncü ve beşinci dereceden Runge-Kutta integrasyon denklemlerini kullanır. ode23 ve ode45 fonksiyonlarında aynı türden giriş ve çıkış argümanları kullanılır.

Ode fonksiyonlarının genel kullanım şekli;

[y t] = ode23('yprime',t0,tf,yo,tol,trace);

biçimindedir. Giriş argümanlarının açıklaması ise aşağıda olduğu gibidir.

Yprime:
İntegre edilmesi gereken diferansiyel denklem takımını tanımlayan fonksiyon dosyası, m-uzantılı olarak saklanan fonksiyon dosyası ve fonksiyon adı aynı (örneğin yprime) olmalı ve ode fonksiyonları ile çağrıldığında " (tırnak) içinde gösterilmelidir. Kullanıcı tarafından hazırlanan fonksiyon dosyası bağımsız değişken x ve bağımlı değişken y olmak üzere iki giriş argümanına ve aşağıda gösterilen, durum türevlerinin bir sütun vektörü olan ydot(y') çıkışına sahiptir.

$$y_i' = \frac{dy}{dx} \quad \text{veya} \quad y_i' = \frac{dy}{dt}$$

Fonksiyon dosyası function ydot=dosya_adı(t,y) bildirimi ile başlar ve çözülecek DD takımı denklemlerinin MATLAB formatında yazılımı ile devam eder.

t0:
İntegrasyon için başlama zamanı veya y=f(x) şeklinde herhangi bir fonksiyonun integre edilmesi halinde integrasyon aralığının başlangıç değeri, a.

tf:
İntegrasyon zamanının nihai değeri veya y=f(x) şeklinde herhangi bir fonksiyonun integre edilmesi halinde integrasyon aralığının sona erme değeri, b.

y0:

Bağımsız değişkenin veya sınır değerleri. Bir diferansiyel denklem takımının çözümü halinde başlangıç koşulları bir satır vektörü biçiminde olur.

tol:
Seçime bağlı bir argüman olup çözümde integrasyon işleminin hata sınırlarını belirler. Eğer bu argüman çağırılan fonksiyonda yer almazsa ode23 fonksiyonunda hata sınır değeri 1.0e-3 ve ode45 ile hata sınır değeri 1.0e-6 olarak otomatik belirlenir.

trace:
İsteğe bağlı bir argüman olup hesaplama sırasında elde edilen ara değerlerin saklanıp saklanmayacağını belirleyen bir bayrak (flag) tır. 0 değeri saklanmayacağını ve 1 değeri saklanacağını gösterir. Kullanılmaması halinde otomatik olarak 0 kabul edilir.

ode fonksiyonları ile elde edilen çözümün sonuçları bir t ve y matrisleri biçiminde daha sonra kullanılmak üzere saklanır. Burada birinci argüman, t bir sütun vektörü şeklinde zaman değerlerini veya y=f(x) halinde bağımsız fonksiyon değerlerini temsil eder. İkinci argüman, y; diferansiyel denklemin derecesine bağlı olarak durum değişkenleri matrisini temsil eder. Bu matris durum değişkenlerinin sayısı kadar sütuna sahiptir. Satır sayısı ise çözümün durumunu belirler.

ode fonksiyonlarının kullanımı ve bunlar ile birlikte kullanılacak diferansiyel denklem fonksiyon dosyalarının yazılmasını açıklamak üzere basit bir birinci dereceden diferansiyel denklem çözüm örneği verelim.

Örnek: Aşağıda verilen birinci dereceden diferansiyel denklemin y(0)=1 başlangıç koşulu ve [1,10] sınır değerleri arasındaki çözümünü bulalım.

$$y' = \frac{dy}{dx} = g_1(x,y) = 4x^3\cos x^2 - x^2$$

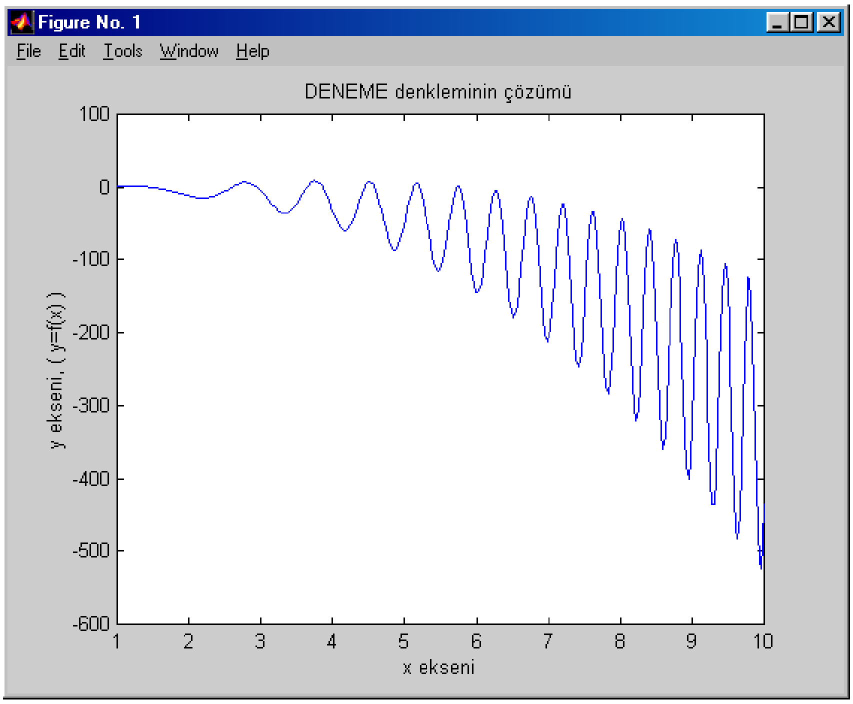
Çözüm: Problemin çözümü için, ilkönce aşağıdaki fonksiyon dosyası yazılır. Fonksiyonun yazılması için MATLAB'ın dahili editörü kullanılabilir. Bunun için MATLAB komut satırında edit yazılıp enter'a basılır. MATLAB'ta sayısal diferansiyel çözümleri için ode fonksiyonları kullanılır. ode fonksiyon dosyaları daha önceden de anlatıldığı gibi fonksiyon fonksiyonu dosyalardır. Yani çözümü üretilecek probleme ait fonksiyonu girdi olarak kabul ederler. Sonucu ise problemin bağımlı ve bağımsız değişkenlerine karşılık gelecek şekilde matrissel olarak üretir. Bundan dolayıdır ki, eleman sayısı birden fazla olan matrissel sonuçları gözlemlemenin en iyi yolu grafiğini çizdirmektir. MATLAB programında bu işi yapan fonksiyon ya da komut plot (çiz) komutudur. Plot komutunun kullanım şekli problemin çözümü sırasında anlatılacaktır

```
function dy =deneme(x,y)
dy=4*x^3*cos(x^2)- x^2;
```

function kelimesiyle başlayan dosyanın fonksiyon adı deneme olarak belirlendiğinden dosya adı da deneme.m olarak saklanmalıdır.

MATLAB ortamında çözüm sağlayan bildirimler ve grafik çıktısı aşağıdaki gibidir.

```
>> [a b]=ode23('deneme',1,10,1);
>> plot(a,b,'b-');
>> title(' DENEME denkleminin çözümü ');
>> xlabel('x ekseni')
>> ylabel('y ekseni, ( y=f(x) )');
```



Resim 6: *Çözümün grafik çıktısı*

MATLAB ortamında çalıştırılan yukarıdaki bildirimler ile aşağıdaki adımlar gerçekleştirilmiştir;

a) birinci satırda ode fonksiyonu ile çağrılan diferansiyel denklem fonksiyonunun sayısal çözümü gerçekleştirilmektedir; bundan sonra ode işleminin sonucunda a ve b 'nin içerikleri aşağıdaki gibi olmuştur.

```
a =

    1.0000
    1.0689
    1.1474
    1.2191
```

```
b =

    1.0000
    1.0720
    1.1178
    1.0967
```

b) ikinci satırda çözüm sonuçlarının plot komutu ile çizdirilmesi sağlanmaktadır.Burada Plot komutunun kullanım şekli aşağıdaki gibidir.

plot(x,y,seçenek)

x: Kordinat düzleminde x eksenine karşılık gelecek olan değerler topluluğu.

y: Kordinat düzleminde y eksenine karşılık gelecek olan değerler topluluğu.

seçenek: Seçenek parametresi zorunlu değildir. Grafik çıktısının görsel açıdan özelliklerini belirleyen bir değerdir. Örnekte kullanılan "b-" ifadesi grafiğin mavi (b:blue) renkte çizgisel (solid) olarak oluşturulacağını belirtir. Kullanılabilecek parametreler aşağıdaki gibidir

renkler		**nokta belirtecleri**		**çizim sitilleri**	
y	yellow	.	point	-	solid
m	magenta	o	circle	:	dotted
c	cyan	x	x-mark	-.	dashdot
r	red	+	plus	--	dashed
g	green	*	star		
b	blue	s	square		
w	white	d	diamond		
k	black	v	triangle (down)		
		^	triangle (up)		
		<	triangle (left)		
		>	triangle (right)		
		p	pentagram		
		h	hexagram		

c) üçüncü satırda ,açıklayıcı olması açısından grafiğe başlık verilmiştir.
d) dördüncü satırda, grafiğin x eksenine ait bir açıklama verilmiştir.
e) beşinci satırda, grafiğin y eksenine ait bir açıklama verilmiştir.

MATLAB'TA DİFERANSİYEL DENKLEMLERİN SEMBOLİK ÇÖZÜMÜ

MATLAB programı her nekadar sembolik çözümler gerçekleştirmek üzere geliştirilmiş bir paket program olmasa da, MATLAB'a 4.0 sürümünden sonra The Symbolic Math Toolbox adı altında bir paketin eklenmesiyle bu iş içinde kullanılmaya başlanmıştır. Sembolik çözümlemenin anlamı; bir sayısal çözüm elde etmek yerine, sayısal çözümün bir basamak gerisinde kalan, sayısal çözümü verebilecek bir fonksiyon elde etmektir. MATLAB'ta diferansiyel denklem çözümü için kullanılan fonksiyon fonksiyonu dsolve (differantial solve: diferansiyel çözümle)'dur.

dsolve **fonksiyon fonksiyonu :**

Dsolve fonksiyonu sıradan diferansiyel eşitliklerin sembolik çözümünü bulur. dsolve('eşitlik1','eşitlik2', ... , 'bsart1','bsart2',...) şeklinde başlangıç şartlarını ve sıradan diferansiyel eşitlikleri parametre olarak alır. Bu şekilde birçok başlangıç şartı ve eşitlik beraberce kullanılabilir. Dsolve fonksiyon fonksiyonu sunucu default olarak, bağımsız değişken "t" ye göre hesaplar. Bu "t" varsayılanı, fonksiyonun sonuna eklenecek bir başka bağımsız değişken ile değiştirilebilir.örneğin;

dsolve('eşitlik1','eşitlik2', ... , 'bsart1','bsart2',...,'x')

"D" harfi bağımsız değişken sırası ile diferansiyel bulmayı belirtir,genellikle "D", *d/dt* dir.Ayrıca *D2=d^2/dt^2* olarak ifade edilir. Hemen "D" diferansiyel bulma operatörünü izleyen herhangi bir karakter, bağımlı değişken olarak alınır.Örnek olarak, D3y y(t)'nin 3.türevini belirtir. Dikkat edilmesi gereken bir nokta vardır, buda "D" harfinin eşitliğin herhangi bir yerinde sembolik değişken olarak kullanılmamasıdır.

dsolve fonksiyonunda, y bağımlı değişkenlerden biri ve a ve b sabitler olduğunda y(a)=b veya Dy(a)=b şeklinde başlangıç şartları belirtilebilir. Eğer başlangıç şartlarının sayısı bağımlı değişkenlerin sayısından az verilirse, meydana gelen çözümler rasgele sabitler (C1,C2,C3,...gibi) içerir.

dsolve fonksiyonu eğer, belirgin bir çözüm bulduysa bunu ekrana basar, eğer belirgin bir çözüm bulunamazsa bir uyarı verilir ve boş bir sym(symbolic) değeri geri döndürülür. Ayrıca dsolve fonksiyonu bazı durumlarda diferansiyel eşitlik veya integral içeren ifadeler içeren çözümler bulabilir.

Birinci Mertebeden Adi Diferansiyel Denklemler

Örnek 1: $(3x^5y^5-2y)dx + (5x^6y^4+x)dy=o$ denklemini çözelim.

Çözüm:

Denklemi dsolve fonksiyon fonksiyonunu kullanarak çözebilmek için aşağıda olduğu gibi ifade etmek gerekir. Burada matematiksel operatörlere ve bunları çevreleyen parantezlerin konumlarına dikkat etmek gerekir. Burada fazla parantez kullanmak lehimize olacaktır, böylece yazabileceğimiz olası hatalı ifadelerden de kurtulmuş oluruz.

```
(3*x^5*y^5-2*y)*Dx+(5*x^6*y^4+x)*Dy=0
```

Denklemin çözümü için MATLAB komut satırına aşağıdaki komutu girmek yeterlidir.

```
>> dsolve('(3*x^5*y^5-2*y)*Dx+(5*x^6*y^4+x)*Dy=0')
```

bu komuttan sonra MATLAB komut penceresinde görüntülenen çözüm fonksiyonu aşağıdaki gibidir.

```
ans=

C1+x^3*y(t)^5+1/x^2*y(t)=0
```

Bu sonucu daha anlaşılır ifade etmek gerekirse;

$\Longrightarrow \quad x^3y^5 + x^{-2}y=c$ ‘dir.

Örnek 2: $(x^2+3y^2)\ dx + 2xy\ dy=0$ denkleminin çözümünü bulalım.

Çözüm:

Bu denklemin çözümü için gerekli MATLAB komutu aşağıdaki gibidir.

```
>> dsolve('(x^2+3*y^2)*Dx+(2*x*y)*Dy=0')
```

Bu komutun çıktısı şöyle olur.

```
ans =

C1+x^3*y(t)^2+1/5*x^5=0
```

Bu sonucu daha anlaşılır ifade etmek gerekirse;

$\Rightarrow$ $\frac{x^5}{5} + x^3 y^2 = c$'dir.

Örnek 3: dy/dt+y=5 dif.denk.çözümü

```
>> dsolve('Dy+y=5')
```

ans = 5+exp(-t)*C1 , sonuç: $5+C1e^{-t}$, C1 çözümle ilgili katsayıdır.

Örnek 4: $\frac{d^2y}{dt^2} + 5y = 25$ dif.denklemin y(0)=2 ve y(1)=3 için çözümü

```
cozum=dsolve('D2y+5*y=25','y(0)=2','y(1)=3')

cozum =

5+(-2+3*cos(5^(1/2)))/sin(5^(1/2))*sin(5^(1/2)*t)-3*cos(5^(1/2)*t)

>> pretty(cozum)

                      1/2        1/2
         (-2 + 3 cos(5   )) sin(5    t)           1/2
     5 + ------------------------------ - 3 cos(5    t)
                          1/2
                     sin(5   )
```

Homogen diferansiyel denklemler

Birinci mertebeden bir lineer adi diferansiyel denklemin

$$\frac{dy}{dx} = f(x, y)$$

şeklinde verildiğini biliyoruz. Eğer, y/x veya x/y nin bir g-fonksiyonu bulunabilirse, öyle ki,

$$\frac{dy}{dx} = f(x,y) = g(y/x)$$

olsun ozaman f(x,y) fonksiyonuna homogen fonksiyon ve yukarıdaki denkleme homogen diferansiyel denklem denir.

Örnek 1: $(y+\sqrt{x^2+y^2})dx - xdy = 0$ denklemini çözelim.
Çözüm:

Çözüm için gerekli MATLAB komutu aşağıdaki gibidir.

```
>> cozum=dsolve('(y+sqrt(x^2+y^2))*Dx-x*Dy=0')
```

Burada sqrt fonksiyonu karekök alma fonksiyonudur. Problemin çözümü olarak;

```
cozum =

[  (1-2*C1*y(t))^(1/2)/C1]
[ -(1-2*C1*y(t))^(1/2)/C1]
```

şeklinde bir sütun matris oluşmuştur. Burada matrisin 1. elemanı ile 2. elemanı arasındaki tek fark işaretlerinin farklı olmasıdır. Bu da problemin iki farklı çözümü olduğunu göstermektedir.

Burada sonuç ifadesinin okunabilirliği azdır. MATLAB’ta bunu normal, matematiksel tarzda ifade etmeye yarayan bir fonksiyon vardır. Bu fonksiyon aşağıdaki gibidir.

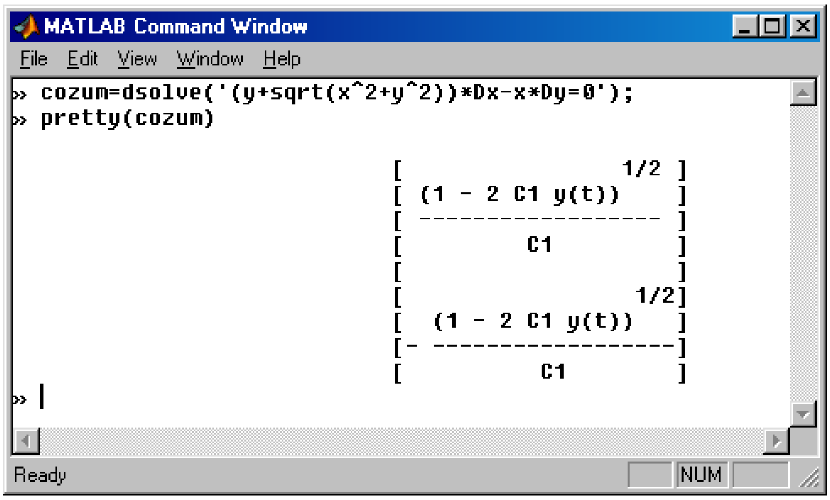
```
pretty(symbolic)
```

symbolic ifadesi normal matematiksel tarzda yazılması istenen fonksiyona karşılık gelmektedir. pretty fonksiyonun ile kullanılacak parametre mutlaka sembolik olmalıdır.Örnek olarak bulduğumuz cozum adındaki sütun matrisi sembolik biçimdedir. Bunu örneğimize uygularsak, yazacağımız komut satırı ve örnek ekran çıktısı aşağıdaki gibidir.

```
>> pretty(cozum)
```

veya

```
>> pretty(dsolve('(y+sqrt(x^2+y^2))*Dx-x*Dy=0'))
```



Resim 7: *pretty komutunun kullanımı*

Birinci mertebeden lineer diferansiyel denklemler

$\frac{dy}{dx} + P(x)y = Q(x)$ şeklindeki denklemlere lineer diferansiyel denklem denir.

Örnek 1: y′+(1/x)y = sin x denklemini çözelim

Çözüm:
Burada y′=dy / dx olduğundan bağımsız değişken x'tir.x dsolve fonksiyonun sonunda belirtilir. Çözüm için gerekli MATLAB komutu aşağıdaki gibidir.

```
>> pretty(dsolve('Dy+(1/x)*y=sin(x)','x'))
```

elde edilen sonuç;

$\frac{\sin(x) - x\cos(x) + C1}{x}$ 'gibidir.

Örnek 2: $\frac{dy}{dx} = e^{2x} + 3y$ denklemini çözelim.

Çözüm:
MATLAB'ta kullanılabilecek bir "e" sabiti yoktur. "e" üzerili ifadeleri belirtmek için exp(p) fonksiyonu kullanılır. Buradaki p "e" nin üssü olan

değerdir. Örneğin exp(1)= 2.7183 ‘dür. Çözüm için gerekli MATLAB komutu aşağıdaki gibidir.

```
>> pretty(dsolve('Dy=exp(2*x)+3*y','x'))
```

elde edilen sonuç;

-exp(2 x) + exp(3 x) C1 olur.

BERNOULLİ DENKLEMİ

Birinci mertebeden bir adi diferansiyel denklem,

$$\frac{dy}{dx} + P(x)y = y^n Q(x)$$

şeklinde ise bu diferansiyel denkleme Bernoulli denklemi denir.

Örnek: $\frac{dy}{dx} - \frac{y}{x} = -\frac{y^2}{x}$ denklemini çözelim

Çözüm:

Çözüm için gerekli MATLAB komutu aşağıdaki gibidir.

```
>> pretty(dsolve('Dy-y/x=-y^2/x','x'))
```

elde edilen sonuç;

$\frac{x}{x + C1}$ olur.

Interpolasyon

Intepolasyon için genel form yi = interp1(x, y, xi, method) seklindedir x ve y verilen degerler,yi ise interpolasyon fonksiyonundan hesaplanan degerler (yi = f(xi)).

YI = INTERP1(X,Y,XI,'method') specifies alternate methods.

Metotlar ise,

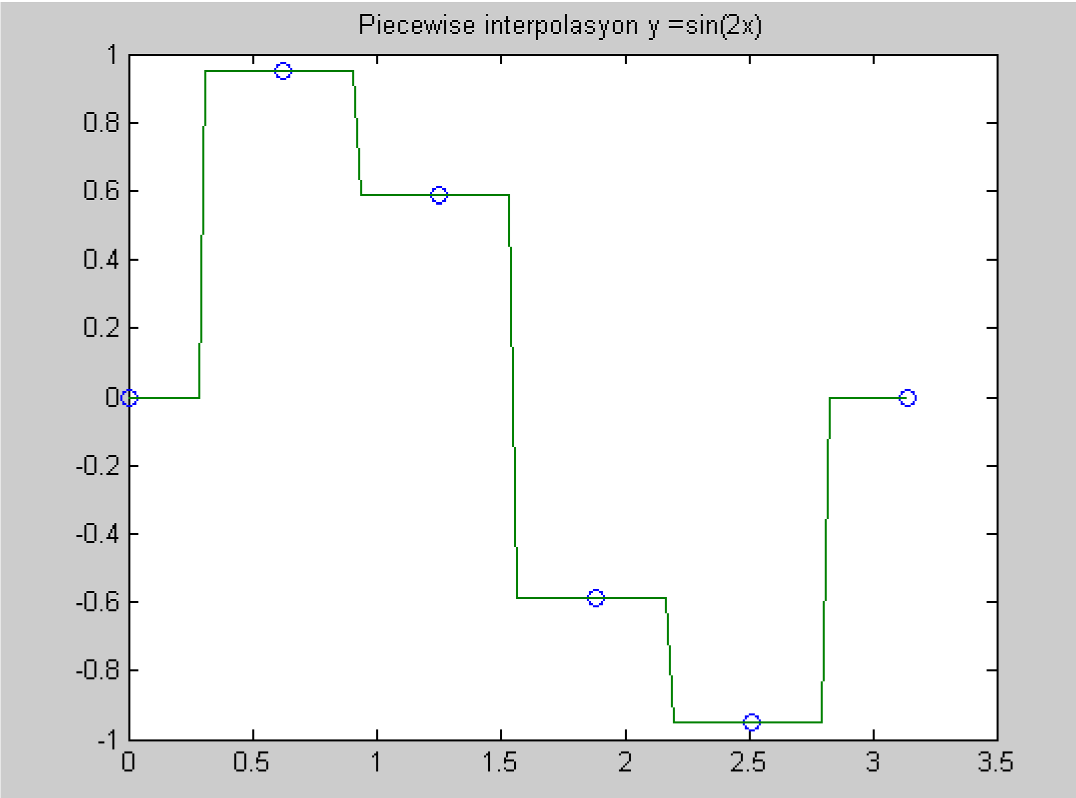
```
'nearest'  - nearest neighbor interpolation
'linear'   - linear interpolation
'spline'   - piecewise cubic spline interpolation (SPLINE)
'pchip'    - piecewise cubic Hermite interpolation (PCHIP)
'cubic'    - same as 'pchip'
```

örnek :
(xk, yk) = (k*pi/5, sin(2x*k)), k = 0, 1, … , 5, degerlerini üretelim,

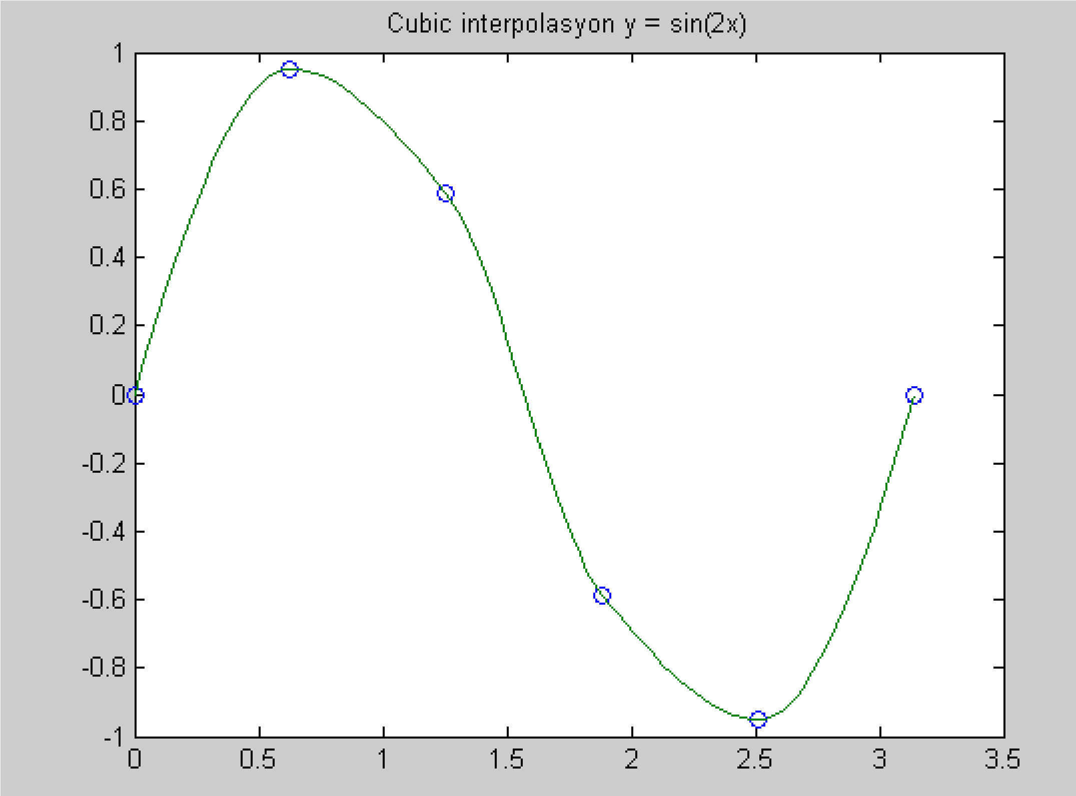
```
x = 0:pi/5:pi;
y = sin(2.*x);
```

bulunan degerleri kullanarak interpolasyon yaparsak ,

```
xi = 0:pi/100:pi;
yi = interp1(x, y, xi, 'nearest');
plot(x, y, 'o', xi, yi), title('Piecewise interpolasyon y =sin(2x)')
```

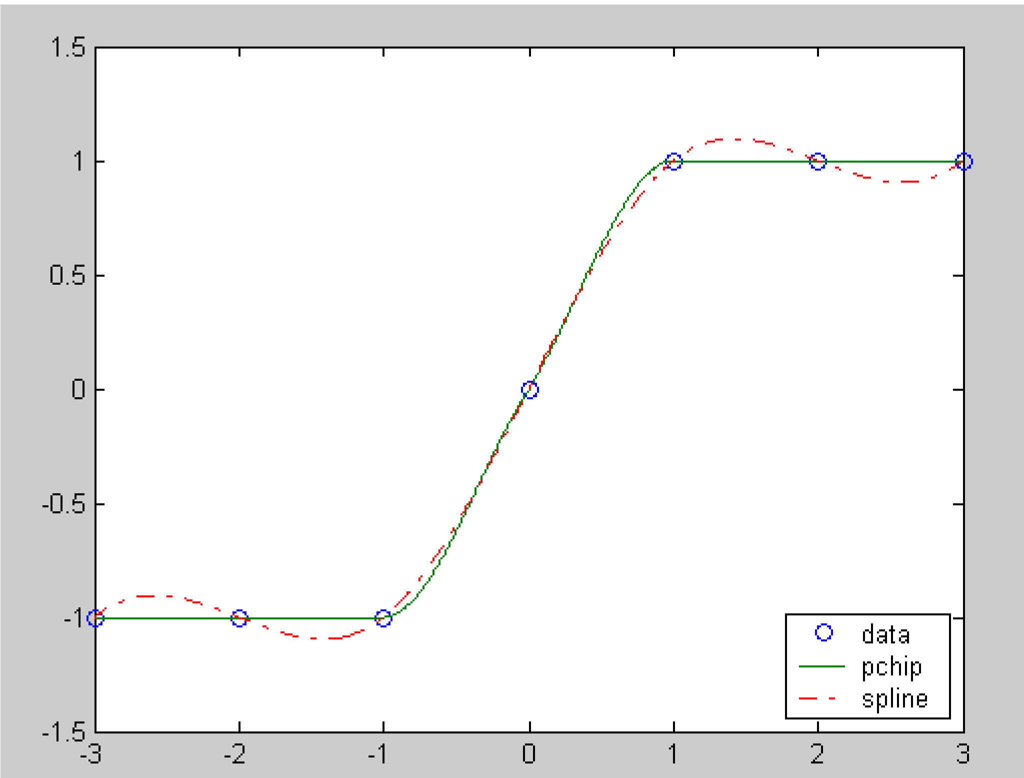



```
yi = interp1(x, y, xi, 'cubic');
plot(x, y, 'o', xi, yi), title('Cubic interpolant of y = sin(2x)')
```



spline fonksiyonu kullanılarakta interpolasyon yapılabilir,

```
x = -3:3;
y = [-1 -1 -1 0 1 1 1];
t = -3:.01:3;
p = pchip(x,y,t);
s = spline(x,y,t);
plot(x,y,'o',t,p,'-',t,s,'-.')
legend('data','pchip','spline',4)
```

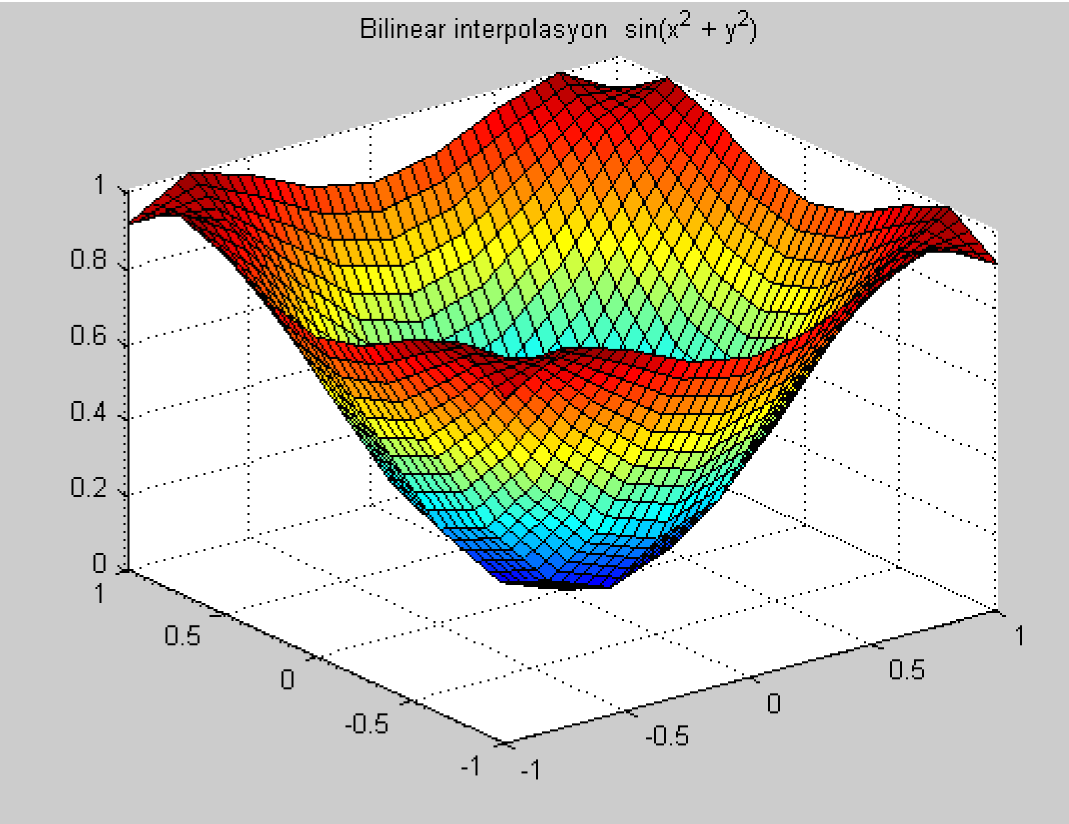


İki boyutlu interpolasyon

{xk, yl} noktaları için , 1 <=k<=m, 1 <= l <=n aralığında zkl, verildiginde
z = f(x, y) interpolasyon denklemi
zi = interp2(x, y, z, xi, yi, 'method') MATLAB fonksiyonu ile bulunabilir
metotlar :
_'nearest' - nearest neighbor interpolation
_'linear' - bilinear interpolation
_'cubic' - bicubic interpolation
_'spline' - spline interpolation

Örnek:
z = sin(x^2 + y^2) fonksiyonundan −1<=x <=1, -1 <=y <=1 aralığında data üreterek 'linear' ve the 'cubic' metotlaral interpolasyon yapalım,

```
[x, y] = meshgrid(-1:.25:1);
z = sin(x.^2 + y.^2);
[xi, yi] = meshgrid(-1:.05:1);
zi = interp2(x, y, z, xi, yi, 'linear');
surf(xi, yi, zi), title('Bilinear interpolasyon sin(x^2 + y^2)')
```



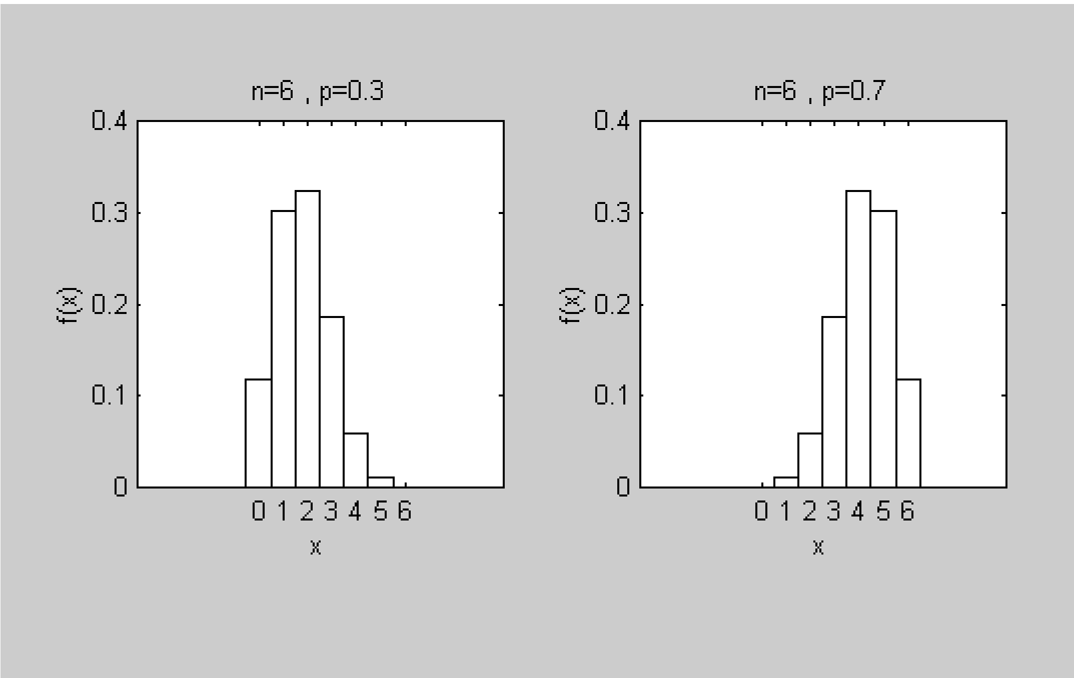
Istatistik uygulamalar

İstatistikte kullanılan dagılım fonksiyonları ve özellikleri üzerinde duralım,

Binomial dagılım:

Binom dagılımının olasılık fonksiyonu,
q=1-p
μ= np
□2= npq

örnek:
n=25,p=0.2,
47
p(x<=3) = ?
p1=binocdf(3,25,0.2)
baska bir biçimde ayni sonuç bulunabilir,
p2=sum(binopdf(0:3,25,0.2))
Örnek:
N=6,p=0.3 ve p=0.7 için olasılık yogunluk fonksiyonunu çizelim,
x=0:6
pdf1=binopdf(x,6,0.3)
pdf2=binopdf(x,6,0.7)
subplot(1,2,1)
bar(x,pdf1,1,'w')
title('n=6 , p=0.3')
xlabel('x')
ylabel('f(x)')
axis square
subplot(1,2,2)
bar(x,pdf2,1,'w')
title('n=6 , p=0.7')
xlabel('x')
ylabel('f(x)')
axis square



Input Output

Sisteme bilgi girisi için input fonksiyonu kullanılr,
A=input(‘bir sayı giriniz’)
Girilen deger degiskene atanır.
Herhangibir degeri görüntülemek için disp kullanılır,
disp('MATLAB');
disp(a)
disp(['Kültür ' 'Universitesi']);
ad = 'ali'; disp(['merhaba ' ad]);
d = [num2str(16) '-subat-' num2str(2004)];
8
disp(d);
16-subat-2004
x = 44.2;
disp(['sonuc = ' num2str(x)]);
cevap=44.2
formatlı çıkıs için, fprintf fonksiyonu kullanılır,
fprintf(format, data)
format için asagıdaki karekterler kullanılır,
_%d integer
_%f floating point format
_%e exponential format
_\n new line character
_\t tab character
fprintf('yarıçapı %d olan dairenin alanı %f', 3, pi*3^2);
yarıçapı 3 olan dairenin alan 28.274334
x = pi;
fprintf('x = %10.2f', x);
x = 3.14

Save , Load

Bir dosya veya degisken save komutu ile saklanır, load ile çalısma alanına getirilir.
save dosya-adı degisken1 degisken2 …
load dosya-adı degisken1 degisken2 …

M-Files
Program codlarını içeren file m-file adı verilir. Pull-down menu ile File-new seçenegi ile yaratılır. Script files
veya function files olmak üzere 2 biçimde yazılabilir.Script file input arguman ve output arguman içermez.
18
Örnek script file,
% Script file
x = pi/100:pi/100:10*pi;
y = sin(x)./x;
plot(x,y)
grid

ornek1 olarak save ediniz.
Yukarıdaki ornek programda % ile baslayan satır açıklama satırı.
Ikinci satırda x degiskeni [pi/100,10*pi] kapalı aralıgında pi/100 artımla yaratılıyor.
Üçüncü satırda x degerleri için hesaplanan sin(x)/x degerleri y degiskeninde saklanıyor.
Plot(x,y) ile grafik çiziliyor. Grid grafik ekranın grid olusturlması için kullanılmıstır.
Command Window da ornek1 yazılarak enter yapılırsa script file run edilmis olur.
Function file için bir vectoru büyükten küçüge sıralayan programı yazalım yazalım.

```
function [b, j] = sirala(a)
% Function a vectorunu büyükten küçüge sıralar
% b argumanı sıralanmıs vector
% j sıralama sonrası elemanların index degerlerini verir.
[b ,j] = sort(-a);
b = -b;
```

Command window da,

```
a = [pi –10 35 0.15];
[b, j] =sirala(a)
b =
35.0000 3.1416 0.1500 -10.0000
j =
3 1 4 2
```

Bazı zamanlarda çalısma anında kullanılmak üzere function tanımlamak gerekebilir.Bu durumda inline fonction
kullanılır.

```
f = inline('sqrt(x.^2+y.^2)','x','y')
f =
Inline function:
f(x,y) = sqrt(x.^2+y.^2)
```

herhangibir degeri hesaplamak için

```
f(3,4)
ans =
5
```

yukarıda yazılan inline function matrix içinde kullanılabilir.
Örnegin:

```
A = [1 2;3 4]
A =
1 2
3 4
```

and
19

```
B = ones(2)
B =
1 1
1 1
```

C = f(A, B)
C =
1.4142 2.2361
3.1623 4.1231
Karekterlerden olusan string bilgiler asagıdaki gibi tanımlanabilir
s = ' MATLAB'
s =
MATLAB
Döngüler,Kontrol
MATLAB programlamada asagıdaki düngü ve kontrollar kullanılabilir.
for loops
while loops
if-else-end
switch-case
for loops
for k = dizi
deyimler
end
for ve end arasındaki deyimler dizide belirtilen her eleman için icra edilir.Örnegin n = 0, 1, …, 10
için n/10 noktalarında sin degerlerini hesap edelim.
for n=0:10
x(n+1) = sin(pi*n/10);
end
x
x =
Columns 1 through 7
0 0.3090 0.5878 0.8090 0.9511 1.0000 0.9511
Columns 8 through 11
0.8090 0.5878 0.3090 0.0000
for döngüsü iç içe kullanılabilir,
H = zeros(5);
for k=1:5
for l=1:5
H(k,l) = 1/(k+l-1);
end
end
20
H
H =
1.0000 0.5000 0.3333 0.2500 0.2000
0.5000 0.3333 0.2500 0.2000 0.1667
0.3333 0.2500 0.2000 0.1667 0.1429
0.2500 0.2000 0.1667 0.1429 0.1250
0.2000 0.1667 0.1429 0.1250 0.1111
Bulunan matrix Hilbert matrix olarak adlandırılır.
10x10 boyutunda A = [akl], matrixini, akl = sin(k)cos(l) olacak sekilde elde etmek isteyelim,

```
A = zeros(10);
for k=1:10
for l=1:10
A(k,l) = sin(k)*cos(l);
end
end
```

yukarıdaki program daha kısa yazılabilir

```
k = 1:10;
A = sin(k)'*cos(k);
```

Bu islem MATLAB vectorization olarak adlandırılır.

While loops

Bu döngünün yapısı,

```
While kosul
Deyimler
End
```

Döngünün tekrar sayısını bilmiyorsa kullanılır.Bunun için asagıdaki örnege bakalım,

```
q = pi;
while q > 0.01
q = q/2;
end
q
q =
0.0061
```

if deyimi

Bir kosula baglı programın akısını degistirmek için kullanılır,

```
If kosul
21
deyimler
end
```

Bir baska kullanım biçimi ,

```
If kosul
Deyimler (Eger kosul dogruysa)
Else
Deyimler (Eger kosul yanlıssa)
End
```

_ç içe if lerde kullanılabilir,

```
if kosu l 1
Deyimler (Eger kosul 1 dogruysa)
elseif kosu l 2
Deyimler (Eger kosul 2 dogruysa)
elseif …
.
.
.
else
Deyimler (Eger tüm kosullar yanlıssa)
End
```

_f deyiminde asagıdaki operatörler kullanılabilir,
Operator
<
<=
>
>=
== Esit
~= Esit degil
Mantıksal operatörler**,**
| And
& Or
~ Not
Örnek : Sayısal analizde önemli yeri olan Chebyshev polinomlarını hesap edelim,
Tn(x), n = 0, 1, …
22
Tn(x) = 2xTn – 1(x) – Tn – 2(x), n = 2, 3, … , T0(x) = 1, T1(x) = x.

```
function T = ChebT(n)
% Coefficients T of the nth Chebyshev polynomial of the first kind.
% They are stored in the descending order of powers.
t0 = 1;
t1 = [1 0];
if n == 0
T = t0;
elseif n == 1;
T = t1;
else
for k=2:n
T = [2*t1 0] - [0 0 t0];
t0 = t1;
t1 = T;
end
end
```

Kübik Chebyshev polinomlarının katsayıları,

```
coeff = ChebT(3)
coeff =
4 0 -3 0
```

T3(x) = 4x3 – 3x.
Switch-case
Genel yapısı,

```
switch deyim (sayısal veya string)
case deger1 (Eger deyimin degeri deger1 ise)
komutlar
case deger2 (Eger deyimin degeri deger2 ise)
komutlar
.
.
.
```

otherwise
komutlar
end
Örnek:
Rastgele üretilen integer sayılar {1, 2, … , 10} kümesi içinde , eger x=1 veya x=2 ise %20 olasılık,x=3 veya
x=4,x=5 ise %30 olasılık, diger durumlarda %50 olasılık mesajı veren programı yazınız.
23
```
% Script M file (olasılık)
x = ceil(10*rand);
switch x
case {1,2}
disp('Olasılık = 20%');
case {3,4,5}
disp('Olasılık = 30%');
otherwise
disp('Olasılık = 50%');
end
```
Programda kullanılam MATLAB fonksiyonlar
rand – uniform dagılımdan rastgele sayı üretir (0, 1)
ceil – round towards plus infinity infinity (see Section 2.5 for more details)
disp – string/sayısal bilgileri ekranda görüntüler
Yukarıdaki programı Command windowda çalıstıralım
```
for k = 1:10
olasılık
end
olasılık = 50%
Probability = 30%
Probability = 50%
Probability = 50%
Probability = 50%
Probability = 30%
Probability = 20%
Probability = 50%
Probability = 30%
Probability = 50%
```
Örnek:
```
a = [1 1 3 4 1]
a =
1 1 3 4 1
i = (a == 1)
i =
1 1 0 0 1
```
Degeri 1 olan elemanlar baska bir diziye atanabilir
```
b = a(i)
b =
1 1 1
```

Baska bir biçimdede kullanılabilir
```
ind = find(a == 1)
ind =
1 2 5
b = a(ind)
b =
1 1 1
```
Örnek:
```
x = randn(1,7)
x =
-0.4326 -1.6656 0.1253 0.2877 -1.1465 1.1909 1.1892
ind = (x >= 1) | (x < -0.2)
ind =
1 1 0 0 1 1 1
y = x(ind)
y =
-0.4326 -1.6656 -1.1465 1.1909 1.1892
```
Dizide elemanların varlıgını sorgulayan isempty fonksiyonu için örnek verelim,
```
isempty(y)
ans =
0
```
y dizisi bos olmadıgı içi 0 döner.
```
isempty([ ])
ans =
1
```
Bos dizi içinse 1 döner.
Örnek:Bir polinomun türev katsayılarını veren fonksiyonu yazalım,
```
function dp = derp(p)
% dp türev polinomu
% p verilen polinom
n = length(p) - 1;
p = p(:)'; % p yi satır haline getirir
dp = p(1:n).*(n:-1:1); % katsayıları hesapla
k = find(dp ~= 0);
if ~isempty(k)
```

```
dp = dp(k(1):end); % sıfırları sil
else
dp = 0;
end
```
p(x) = x3 + 2x2 + 4 polinomu için yukarıdaki fonksiyon kullanılırsa
```
dp = derp([1 2 0 4])
dp =
3 4 0
```

MATLAB Grafik

Matlab birçok grafik fonksiyona sahiptir._ki ve üç boyutlu grafikler kolaylıkla çizilebilir.

F(x)= x/(1+x^2) fonksiyonunu çizelim.

```
% Script file
% Fonksiyon y = x/(1+x^2).
for n=1:2:5
n10 = 10*n;
x = linspace(-2,2,n10);
y = x./(1+x.^2);
plot(x,y,'r')
title(sprintf('Graph %g. Plot based upon n = %g points.' , (n+1)/2, n10))
axis([-2,2,-.8,.8])
xlabel('x')
ylabel('y')
grid
pause(3)
end
```

Yukarıdaki programı incelersek,

linspace(a, b, n) : [a,b] aralıgını n noktaya bölerek yaratır

'r' : renk kodu (red)

title : grafik üzerine açıklayıcı bilgilerin yazılmasını saglar.

sprintf : string ve sayısal bilgilerin birlikte olusturulmasını saglar.

26

%g : integer sayılar için format

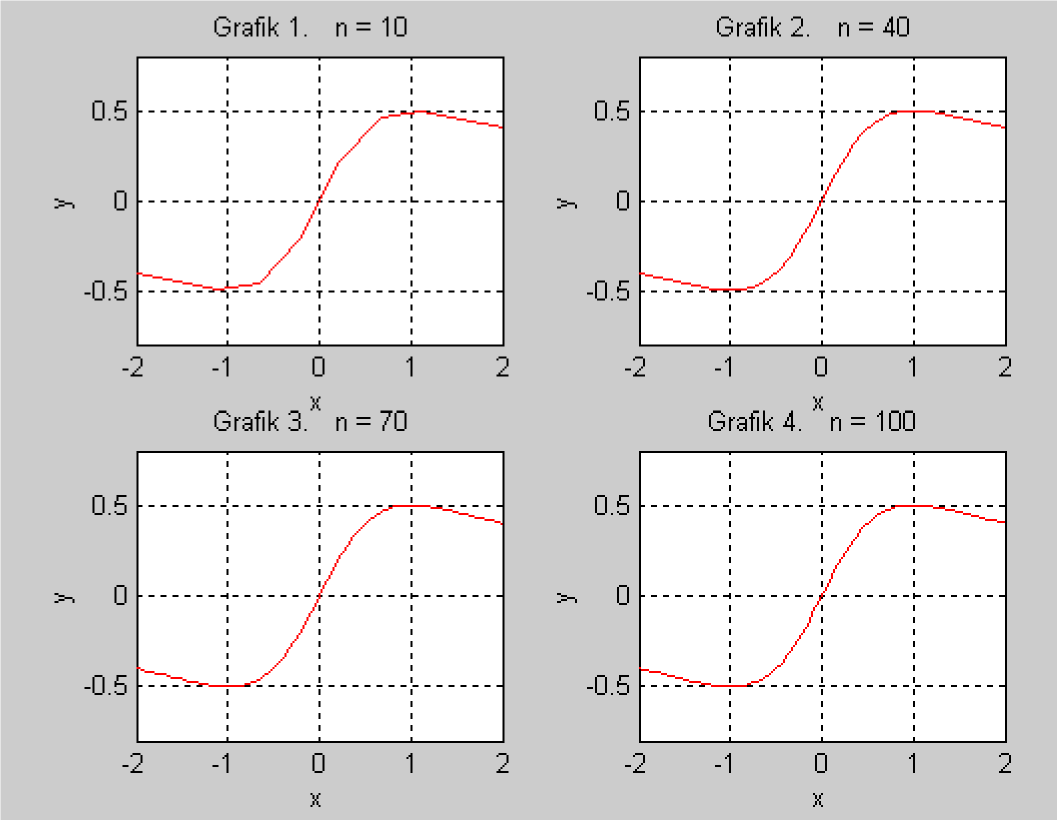
xlabel,ylabel : Exsenlerin etiketlerini yazmak için

axis : Eksenleri düzenler

pause : grafik ekranı bekletmek için kullanılır

Ayni grafigi tek bir grafik ekranda çizebiliriz.Bu amaçla subplot kullanalım.

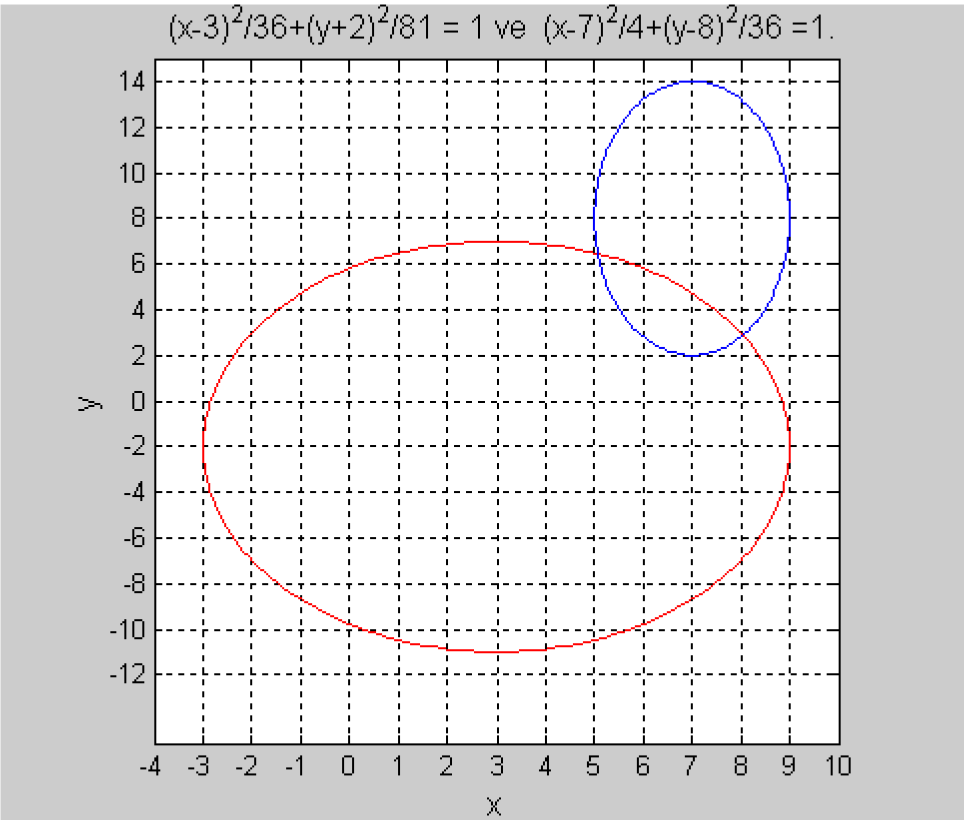
```
% Script file
% y = x/(1+x^2)
k = 0;
for n=1:3:10
n10 = 10*n;
x = linspace(-2,2,n10);
y = x./(1+x.^2);
k = k+1;
subplot(2,2,k)
% 2x2 4 grafik çizilecek, k=grafigin indexi
plot(x,y,'r')
title(sprintf('Grafik %g. n = %g ' , k, n10))
xlabel('x')
ylabel('y')
axis([-2,2,-.8,.8])
grid
pause(3);
end
```



Asagıdaki iki elipsi çizelim,

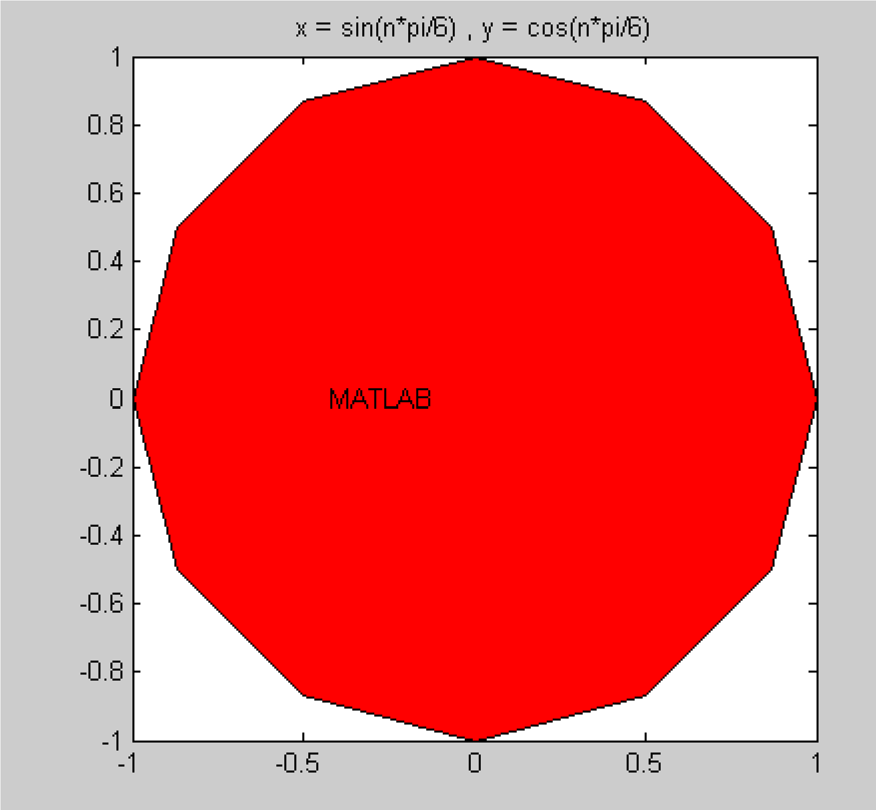
```
% Script file
% x(t) = 3 + 6cos(t), y(t) = -2 + 9sin(t)
% x(t) = 7 + 2cos(t), y(t) = 8 + 6sin(t).
t = 0:pi/100:2*pi;
x1 = 3 + 6*cos(t);
y1 = -2 + 9*sin(t);
x2 = 7 + 2*cos(t);
y2 = 8 + 6*sin(t);
h1 = plot(x1,y1,'r',x2,y2,'b');
set(h1,'LineWidth',1.25)
axis('square')
xlabel('x')
h = get(gca,'xlabel');
set(h,'FontSize',12)
set(gca,'XTick',-4:10)
ylabel('y')
h = get(gca,'ylabel');
set(h,'FontSize',12)
```

```
set(gca,'YTick',-12:2:14)
title(' (x-3)^2/36+(y+2)^2/81 = 1 ve (x-7)^2/4+(y-8)^2/36 =1.')
h = get(gca,'Title');
set(h,'FontSize',12)
grid
```



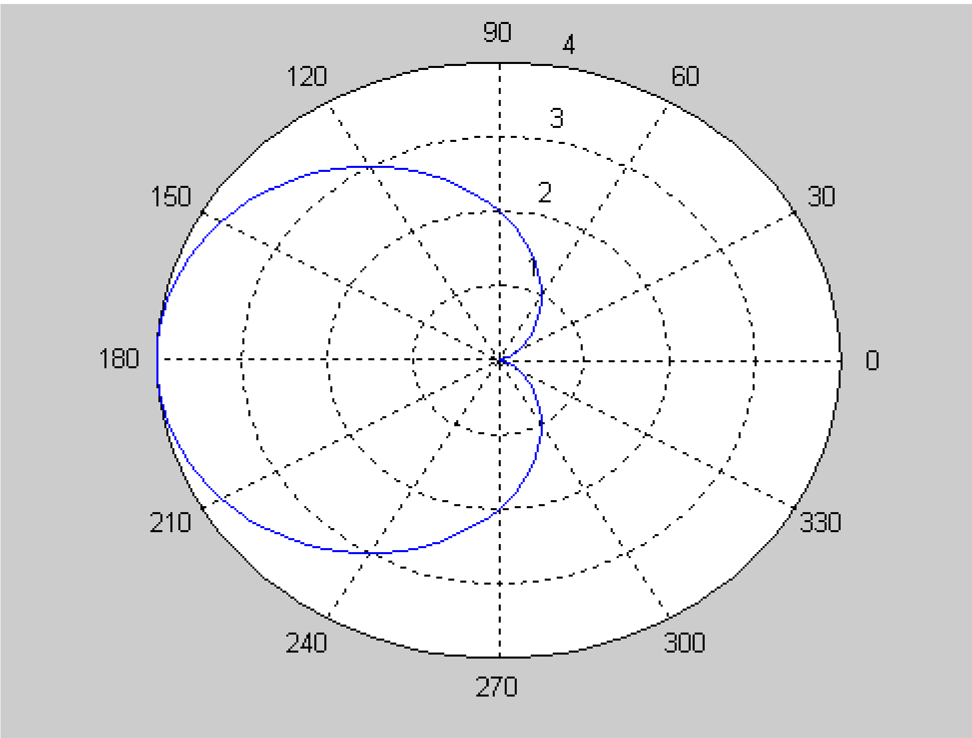
Grafik çiziminde kullanılacak renk kodları asagıdadır,
y yellow
m magenta
c cyan
r red
g green
b blue
w white
k black
Çizilen grafiklerin içleri doldurulabilir
Bir örnekle görelim,

```
n = -6:6;
x = sin(n*pi/6);
y = cos(n*pi/6);
fill(x, y, 'r')
axis('square')
title(' x = sin(n*pi/6) , y = cos(n*pi/6) ')
text(-0.45,0,' MATLAB')
```



Polar koordinatlarda çizim için bir örnek verelim,

theta=0:pi/60:2*pi;r=2*(1-cos(theta));polar(theta,r);axis square



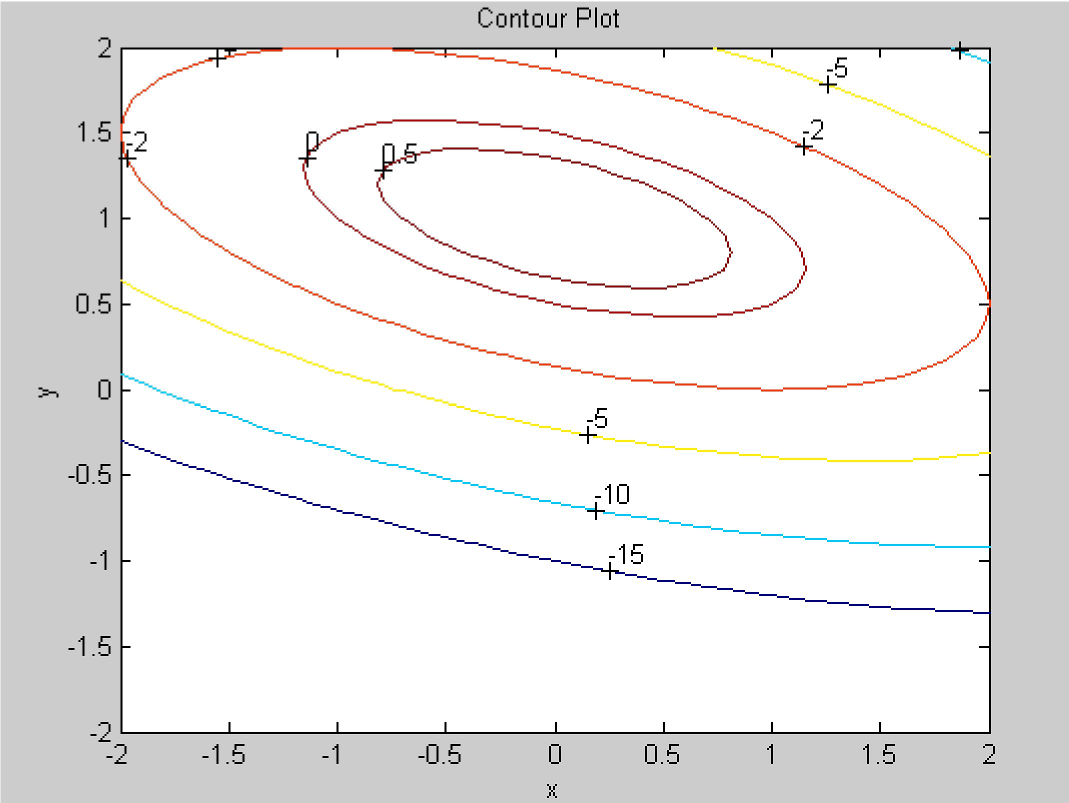
Contour' ların çizimi

Contour' ların çizimi sık karsılasılan bir problemdir. Örnek olarak,

$f(x,y) = 2 - [(x-1)_2 + 4(y-1)_2 + 2xy]$ fonksiyonunu alalım,

```
function m=pr(x,y)
m=zeros(0,length(x))
for k=length(y):-1:1
m=[m;2-((x-1).^2+4*(y(k)-1)^2+2*x*y(k))]
end
x=-2:.1:2;
```

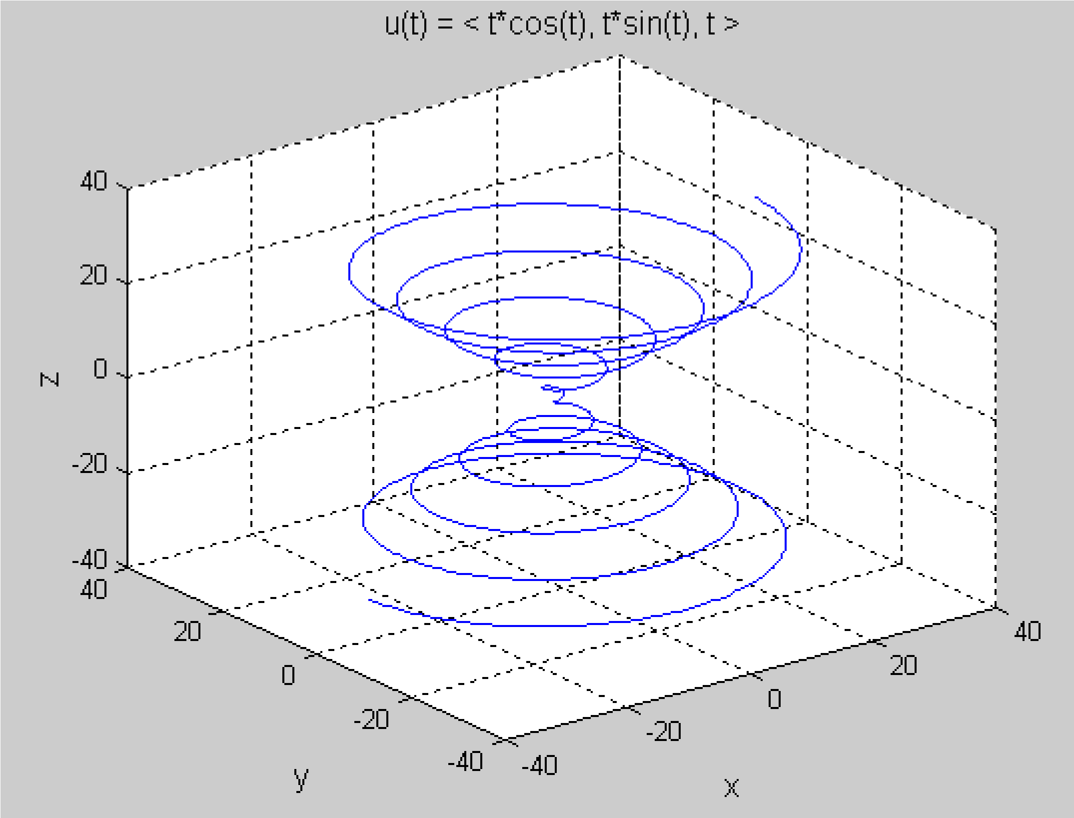
```
y=x;
m=pr(x,y);
cs=contour(x,y,flipud(m),[-15 -10 -5 -2 0 .5]);
clabel(cs)
xlabel('x')
ylabel('y')
title('Contour Plot ')
```



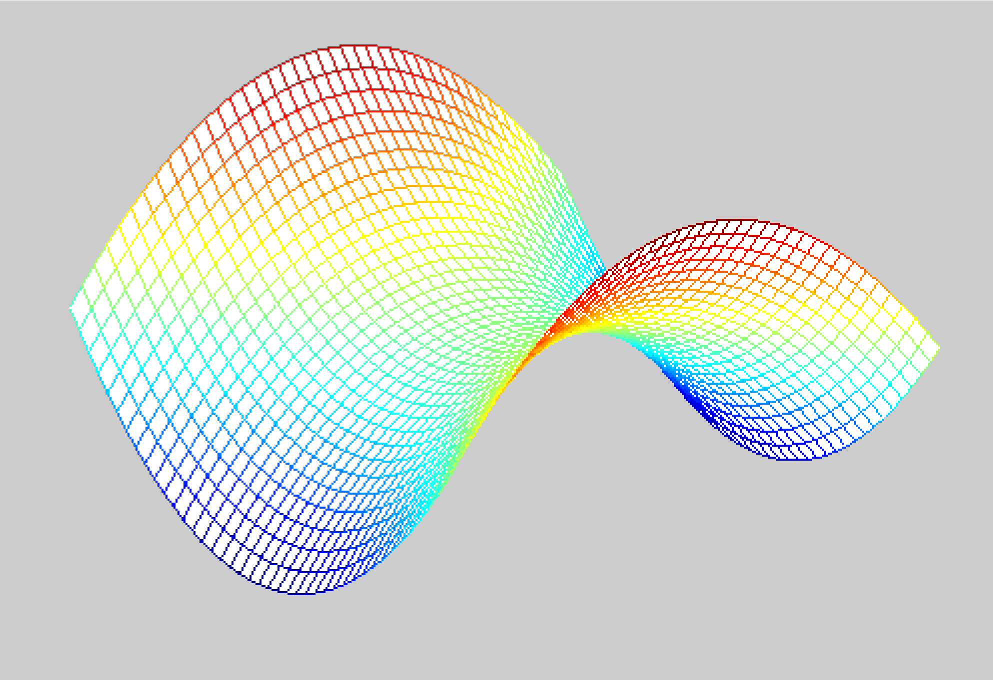
3 Boyutlu grafikler

plot3 fonksiyonu kullanarak

```
% Script file
% Curve r(t) = < t*cos(t), t*sin(t), t >.
t = -10*pi:pi/100:10*pi;
x = t.*cos(t);
y = t.*sin(t);
h = plot3(x,y,t);
set(h,'LineWidth',1.25)
title(' u(t) = < t*cos(t), t*sin(t), t >')
h = get(gca,'Title');
set(h,'FontSize',12)
xlabel('x')
h = get(gca,'xlabel');
set(h,'FontSize',12)
ylabel('y')
h = get(gca,'ylabel');
set(h,'FontSize',12)
zlabel('z')
h = get(gca,'zlabel');
set(h,'FontSize',12)
grid
```



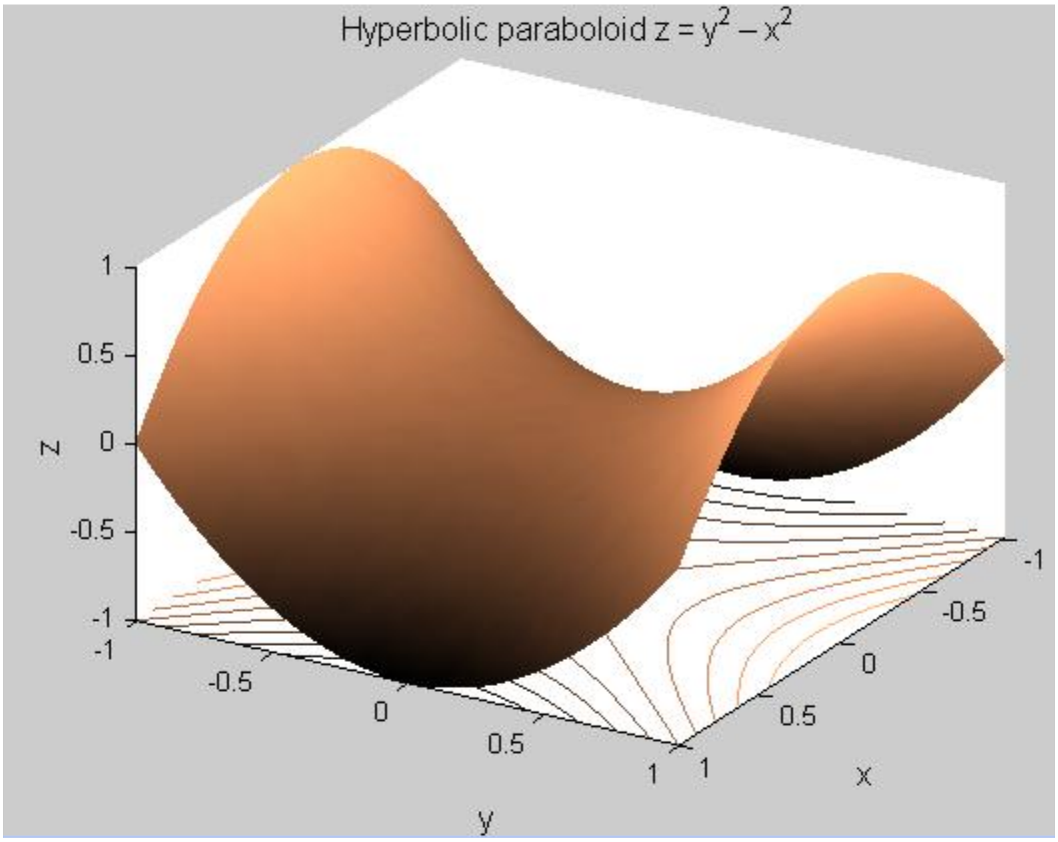
mesgrid fonksiyonu kullanılarak 3 boyutlu koordinatlar yaratılabilir.
x = [0 1 2];
y = [10 12 14];
[xi, yi] = meshgrid(x,y)
xi =
0 1 2
0 1 2
0 1 2
yi =
10 10 10
12 12 12
14 14 14
örnek:
x = -1:0.05:1;
y = x;
[xi, yi] = meshgrid(x,y);
zi = yi.^2 - xi.^2;
mesh(xi, yi, zi)
axis off

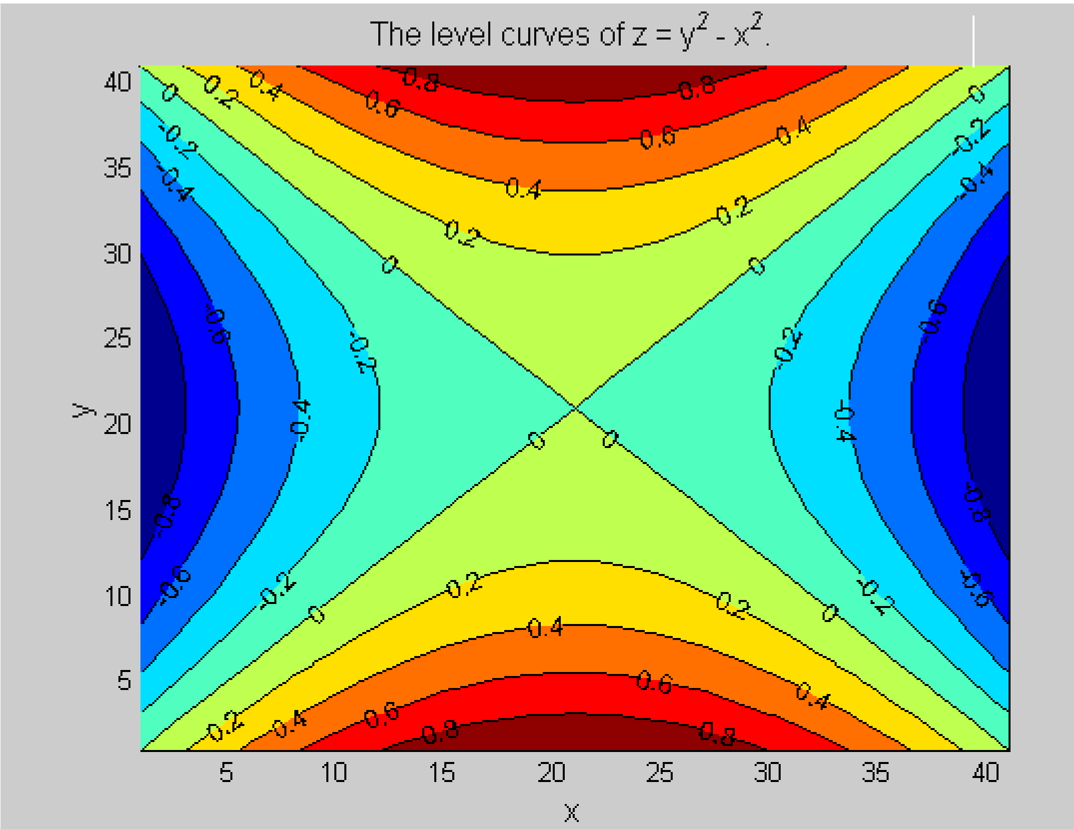


örnek:

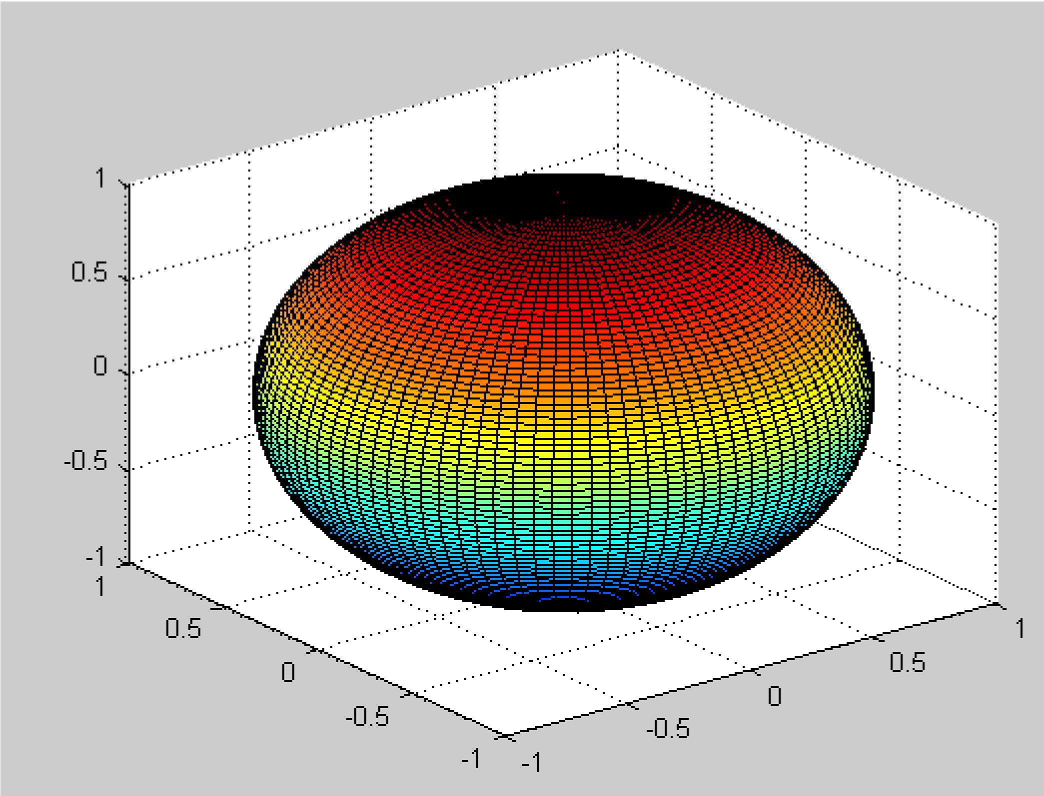
```
% Script file
% = y^2 - x^2
x = -1:.05:1;
y = x;
33
[xi,yi] = meshgrid(x,y);
zi = yi.^2 - xi.^2;
surfc(xi,yi,zi)
colormap copper
shading interp
view([25,15,20])
grid off
title('Hyperbolic paraboloid z = y^2 – x^2')
h = get(gca,'Title');
set(h,'FontSize',12)
xlabel('x')
h = get(gca,'xlabel');
set(h,'FontSize',12)
ylabel('y')
h = get(gca,'ylabel');
set(h,'FontSize',12)
zlabel('z')
h = get(gca,'zlabel');
set(h,'FontSize',12)
pause(5)
figure
contourf(zi), hold on, shading flat
[c,h] = contour(zi,'k-'); clabel(c,h)
title('The level curves of z = y^2 - x^2.')
h = get(gca,'Title');
set(h,'FontSize',12)
xlabel('x')
h = get(gca,'xlabel');
```

```
set(h,'FontSize',12)
ylabel('y')
h = get(gca,'ylabel');
set(h,'FontSize',12)
```

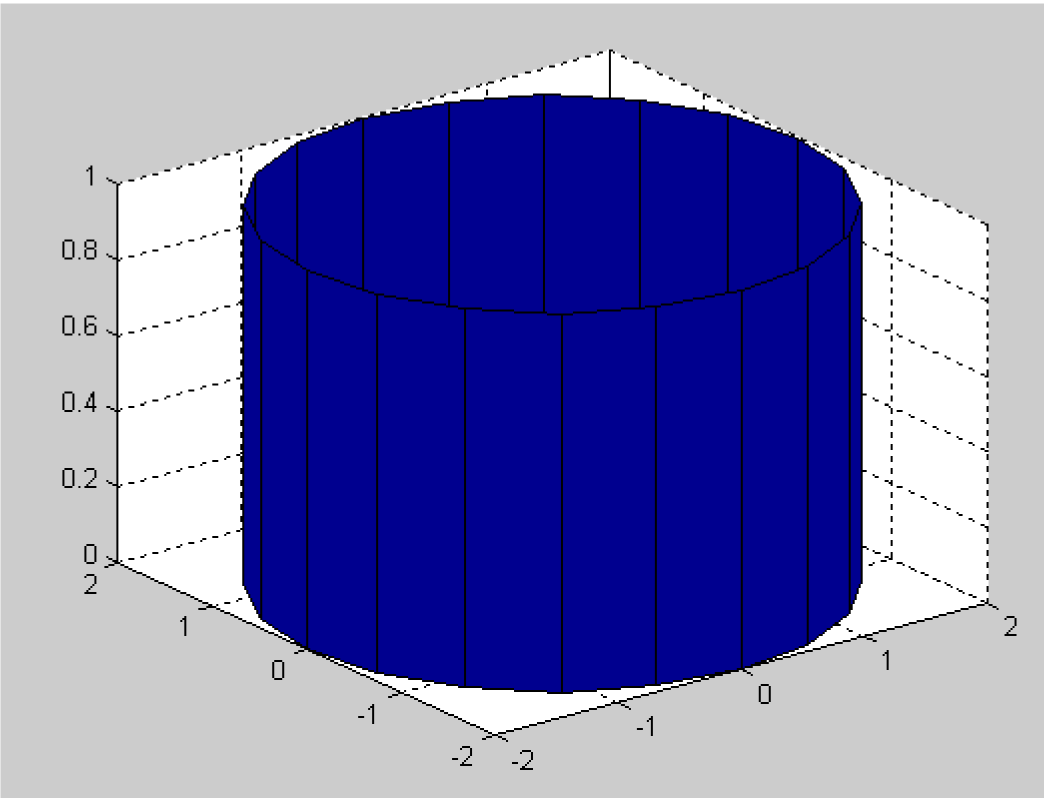




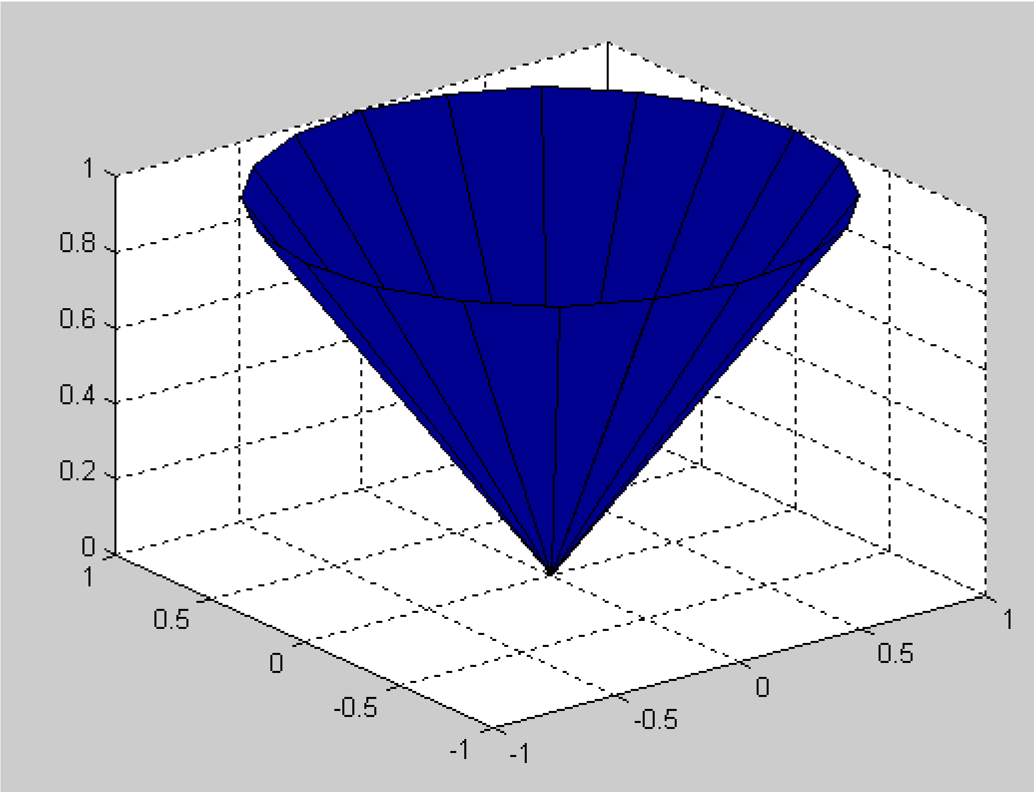
Bazı özel sekiller için hazır fonksiyonlar vardır,
Küre:
35
[x,y,z]=sphere(100) % 100 noktayla çiz
surf(x,y,z)



silindir:
[x,y,z] =cylinder(2) % r=2
surf(x,y,z)



koni:
[x,y,z] = cylinder([0 1])
surf(x,y,z)



2 MATRİSLER

Matris elemanları alelade rakamlardan ibaret olabileceği gibi MATLAB deyimlerinden ibaret de olabilir. Örneğin

x=[-1.3 sqrt(3) (1+2+3)*4/5]

bildirimi

x= -1.3000   1.7321   4.8000

sonucunu doğurur.
Bireysel matris elemanları parantez "(.)" içindeki indislerle ilişkilendirirler. Yukarıdaki örneğe devam edecek olursak
x(5)=abs(x(1))
bildirimi

x= -1.3000   1.7321   4.800   0.0000   1.3000

sonuçlandırır. Burada x, 5 elemanlı olarak tanımlandığından ilk örnekten atanan 3 elemanı, ilk üç eleman olarak yerine yerleştirmiş 4. elemanı otomatik olarak sıfır aldıktan sonra 5. elemanı x2in 1. elemanının mutlak değerini alarak yerleştirmiştir.

C=[1 2 3;4 5 6;7 8 9]

1	2	3
4	5	6
7	8	9

Küçük matrisleri eleman olarak kullanıp daha büyük matrisler oluşturmak mümkündür. Örneğin yukarıda verilen C matrisine aşağıda gösterildiği bir satır ekleyebiliriz.

C=[C; [10 11 12]]

Bu durumda

C=

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

olarak sonuçlanır. ":" işareti kullanmak sureti ile büyük bir matristen daha küçük bir matris oluşturulabilir. Örneğin

C=C(1:3, :)

Bildirimi ile yukarıda en son oluşturulan C matrisinin ilk 3 satırı ve tüm sütunları alınarak sol taraftaki C ye atanır ve böylece ilk orijinal C matrisi elde edilmiş olunur.

Verilerin değerlendirilmesinde sütun işlemcisi ":" çok kullanışlıdır. Örneğin

x=C(:,1)
y=C(:,2)

bildirimleri ile C adı altında matris biçiminde tanımlanan verilerin 1. sütun x değişkenine ve 2. sütunu da y değişkenine atanmış olur.

Benzer şekilde;

x=C(1,:)

data matrisindeki verilerin 1. satırını x değişkenine atar.

X=C(:,2:3)

C matrisindeki verilerin 2. ve 3. sütunlarını x değişkenine atar.

X=C(1:3,:)

C matrisindeki verilerin 1,2 ve 3 üncü satırlarını x değişkenine atar.

Bu şekilde mevcut bir veri tablosundan çok değişik matrisler oluşturmak mümkündür.

MATLAB'ta kullanıcı, verileri kendi matrisleri ile tanımlayabileceği gibi MATLAB'ın kendi özel matrislerinden de faydalanabilir. Bunların belli başlıları; ones, eye, zeros, magic, rand, randn şeklindedir.

Ones(n) :Tüm elemanları 1 olan nxn elemanlı kare matris oluşturur.
Ones(1,n):Tüm elemanları 1 olan n elemanlı bir satır matrisi oluşturur.
Ones(n,1):Tüm elemanları 1 olan n elemanlı bir sütun matris oluşturur.

triu(a) üst üçgen matris
tril(a) alt üçgen matris

Benzer şekilde eye ile köşegen elemanları 1 olan birim matris, zeros ile tüm elemanları sıfır olan matris, rand, randn ile elemanları rasgele sayılardan oluşan bir matris oluşturulabilir. help olanakları ile daha ayrıntılı bilgiler elde edilebilinir.

"*eye*" fonksiyonu ile nxn boyutunda identity(birim) matrisi oluşturulabilir.

```
» e=eye(3)

    e =

       1    0    0

       0    1    0

       0    0    1
```

Matlab, *size* ve *length* komutları yardımı ile size matrisinizin boyutlarını söyler.

```
» a=[ 2  3  4  5  6
      7  8  9 10 11];

» s=size(a)

    s =

       2    5
```

» b=[17 11 0 30 40 50];

» k=length(b)

k =

6

Rand(3) ,3x3 tipinde 0-1 arasında sayılardan oluşan random matris üretir
Randint(3,2,[1,10]) , 3*2 tipinde 1-10 arasında sayılardan oluşan bir random matris üretir

3 MATLAB'TA KULLANILAN SABİTLER

MATLAB programlarında kullanılabilen skalar değerler aşağıda tanımlanmıştır.Bu değişkenlerin içerikleri MATLAB komut satırında yazılıp, enter'a basılarak görüntülenebilir.

pi(π)

π değeri pi adı altında otomatik olarak saklanmıştır. Programlar içinde kullanılan pi kelimesi doğrudan π değerine karşılık gelir.

i,j($\sqrt{-1}$)

i, j harfleri doğrudan $\sqrt{-1}$değerine ayarlanmıştır.

İnf (∞)

Bu kelime MATLAB'ta sonsuz değeri için atanmış bir değişkendir ve sıfıra bölme işlemlerinde ortaya çıkar. Eğer sıfıra bölme işlemi görüntülenmek istenirse bir uyarı mesajı alınır ve sonuç ∞ işareti şeklinde görüntülenir veya basılır.

NaN

Bu değer Not-a-Number (rakam değil) anlamına gelir ve sıfır bölü sıfır bölümünde olduğu gibi tanımlanmamış deyimlerde ortaya çıkar.

eps

Bu değer fonksiyon, kullanılmakta olan bilgisayar için floatingpoint (virgüllü sayılar) tamlığını içerir. Bu epsilon tamlığı 1.0 ve bunu izleyen enbüyük decimal (onlu sayılar) arasındaki farktır.

ans

Bu değişken bir deyim tarafından hesaplanan fakat bir değişken ismi altında saklanmayan değerleri saklamak için kullanılır.

4 MATLAB’TA ELEMAN ELEMANA HESAPLAMA İŞLEMLERİ

Eleman elemana hesaplama işlemi eleman eleman icra edilir. Örneğin A ve B’nin 5’er elemanlı birer satır vektörü olduğunu varsayalım. Bu değerler ile yeni bir C satır vektörü oluşturmanın bir yolu, aşağıda gösterildiği gibi A ve B deki karşılık gelen değerlerin çarpımlarını almaktır.

```
C(1)  =  A(1) * B(1);
C(2)  =  A(2) * B(2);
C(3)  =  A(3) * B(3);
C(4)  =  A(4) * B(4);
C(5)  =  A(5) * B(5);
```

Bu komutlar esasında skalar komutlardır. Çünkü herbir komut tek bir skalar değeri diğer bir tek skalar değer ile çarparak çarpımı üçüncü bir değer olarak saklamaktadır. MATLAB’da aynı boyutlu iki matris arasında eleman eleman çarpma işleminin icrası aşağıdaki bildirimde gösterildiği gibi,”*” işlemcisi ile çok daha kısa yoldan yerine getirilir.

```
C = A.*B
```

Toplama ve çıkarma işleminde eleman elemana hesaplama ve matris işlemleri aynıdır. Dolayısıyla bunlarda farklı işlemci kullanmaya gerek yoktur. Buna karşılık eleman eleman işlemlerinde çarpma, bölme ve üst alma, matris işlemlerindeki çarpma, bölme ve üst alma işlemlerinden farklılık gösterir. Burada en önemli fark çarpma, bölme ve üst alma işlemcileri önünde nokta, “.” İşareti gelmesidir.

Eleman elemana hesaplama işlemleri yalnızca aynı boyutlu iki matris arasında uygulanmayıp aynı zamanda skalar ve skalar olmayan değerler arasında da uygulanır. Bununla beraber bir matrisin bir skalar ile çarpımı ve bölümünde işlemciler noktalı ve noktasız olarak kullanılabilir. Buna göre skalar olmayan bir A matrisi için aşağıda verilen bildirim takımları birbirine denktir.

```
B=3*A; veya B=3.*A;
C=A/5; veya C=A./5;
```

Sonuç matrisler B ve C her iki durumda da A ile aynı boyutta bir matris olur.

Eleman elemana çarpım: Vektörlerde kullanılan eleman elemana çarpma işlemini göstermek üzere aşağıda verilen iki satır vektörünü ele alalım.

```
A=[2  5  6];   B=[2  3  5];
C=A.*B
```

şeklinde verilen bir komut

C=[4 15 30]

eleman elemana çarpma sonucunu verecektir.

Eleman elemana bölme: MATLAB ileri veya sağdan bölme, ./ ve geri veya soldan bölme, .\ olmak üzere iki bölme işlemcisi kullanır. Sağdan eleman elemana bölme komutu

C=A./B

biçiminde olup bu da

C=[1 1.667 1.2]

A'nın her bir elemanın B tarafından bölümünü sonuçlandırır. Buna karşılık geri ve soldan bölme (ters bölme) işlemi

C=A.\B

komutu ile gerçeklenir ki bu B'nin A tarafından bölümünü sonuçlandırır.

C=[1 0.6 0.833]

Eleman elemana üst alma: Bu işlemde eleman elemana üst alınır. Yukarıda tanımlanan A ve B vektörlerini ele alacak olursak;

```
C=A.^2;
D=A.^B;
```

komutları yolu ile C ve D vektörleri elde edilir.

```
C=[4 25 36];
D=[4 125 7776];
```

Aynı zamanda bir skalar tabanın vektör üssünü almak mümkündür. Örneğin

C=3.0.^A;

bildirimi

C=[9 243 729];

Sonucunu doğuracaktır. Bu vektör aynı zamanda aşağıdaki bildirimle de hesaplanabilir.

C=3.^A;

Burada MATLAB nokta işaretini "." 3 sabitinin bir parçası sayacak ve ona göre işlem yapacaktır. Bu da matris işlemine karşılık aşağıda geleceğinden kastedilene göre yanlış olacaktır. Buna karşılık aşağıda gösterildiği gibi nokta ile sabit arasında bir boşluk konacak olursa yine doğru sonuç alınır.

C=3 .^A;

Bu örnekler eleman elemana hesaplama işlemi yapılırken çok dikkatli olunması gerektiğini göstermektedir. Yazılan ifadenin doğruluğundan tam olarak emin olunmadığı durumlarda basit örneklerle bazı testler yapılması yerinde olacaktır. Bunu aşağıdaki örneklerle gösterebiliriz.

Bundan önceki örneklerde eleman elemana hesaplama işlemlerinde, vektörler kullanılmıştır. Aşağıdaki örneklerde görüldüğü gibi eleman elemana hesaplama işlemlerinde matrisleri de kullanmak mümkündür.

```
d=[1:5; -1:-1;-5]
z=ones(d);
s=d-z;
P=d.*s;
Sq=d.^3;
```

Hesaplama sonuçları aşağıda gösterildiği gibidir.

```
D=
  1    2    3    4    5
 -1   -2   -3   -4   -5

Z=
 1    1    1    1    1
 1    1    1    1    1

S=
  0    1    2    3    4
 -2   -3   -4   -5   -6

Sq=
 1     8    27    64     125
-1    -8   -27   -64    -125
```

Transpose işlemi;

» c=b'

c =

7   10

8   11

9   12

Verilen bir x matrisi için:

» x=[2 -1; 5 8]

x =

2   -1

5 8

Determinant:

» deter=det(x)

deter =

21

Inverse işlemi:

» y=inv(x)

y =

8/21 1/21

-5/21 2/21

Bir z matrisi için:

» z=[1 2; -2 4]

z =

1 2

-2 4

Bölme işlemi aynı inverse işlemini yapıp çarpma işlemi gibi sonuç verir.

» res=z/x

res =

-2/21 5/21

-12/7 2/7

*» res=z*inv(x)*

res =

-2/21 5/21

-12/7 2/7

Eleman elemana bölme işlemi:

» res=z./x

res =

1/2 -2

-2/5 1/2

4.1 DENKLEM TAKIMLARININ ÇÖZÜMLERİ

GİRİŞ

Mühendislik problemlerinde doğrusal ve doğrusal olmayan çeşitli denklem ve denklem takımları ile karşılaşılır. Herhangi bir denklem en basitinden bir doğru veya bir eğriyi tanımlar. n. Dereceden tek değişkenli bir polinom biçimindeki bir denklemde, denklemin köklerini bulmak esastır. Birden fazla değişkeni bulunan denklem takımlarında ise denklemleri teşkil eden bilinmeyenlerin çözümü sağlanır.

Doğrusal denklemlerin çözümü doğrusal olmayan denklemlere göre daha kolaydır.

MATLAB'ta doğrusal denklemlerin çözümü matris işlemleri yolu ile çok basit şekilde çözülebilir. Polinom biçiminde doğrusal bir denklemin kökleri roots fonksiyonu yolu ile çözülebilir. Doğrusal denklem takımları ise sağ veya sol matris bölme işlemleri ve inv (matrisin tersini alma) fonksiyonu yardımı ile çözülür.

Doğrusal olmayan denklemlerin çözümünde; MATLAB içinde yer alan fmin, fmins, fplot, fzero fonksiyon fonksiyonları kullanılabilir. Doğrusal olmayan denklem takımlarının çeşitli sınır şartları altında çözümünde kullanılan çok değişik fonksiyonlar mevcuttur. Bunlar içinde en kullanışlısı fsolve fonksiyonudur.

4.2 DOĞRUSAL DENKLEM ÇÖZÜMLERİ

Tek değişkenli bir doğrusal sistemin denklemi, n. dereceden bir plinom biçiminde;

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_2 x^2 + a_1 x + a_0$$

tanımlanır. Burada f(x)=0 biçiminde denklemin köklerini bulmak için MATLAB içinde yer
alan roots fonksiyonu veya fzero fonksiyon fonksiyonu kullanılabilir. Fzero fonksiyon fonksiyonu daha çok doğrusal olmayan denklemlerin çözümünde kullanılır. Roots fonksiyonunun kullanılışı, aşağıda verildiği gibi çok basittir.

roots([an, an-1,..., a2, a1, a0])

Burada roots fonksiyonu için giriş argümanlarını, an, an-1, ...,a2, a1, a0 biçiminde polinomun en yüksek dereceden başlayarak azalan kuvvetlerinin katsayıları olarak tanımlamak yeterlidir. Sonuçta çıkış olarak denklemin kökleri bulunur.

Örnek :Aşağıda verilen 8. dereceden polinomun köklerini bulunuz.

$f(x)=3x^8+4x^7-9x^6+13x^5-x^4+1.5x^3-10.5x^2+15x-5$

Çözüm:

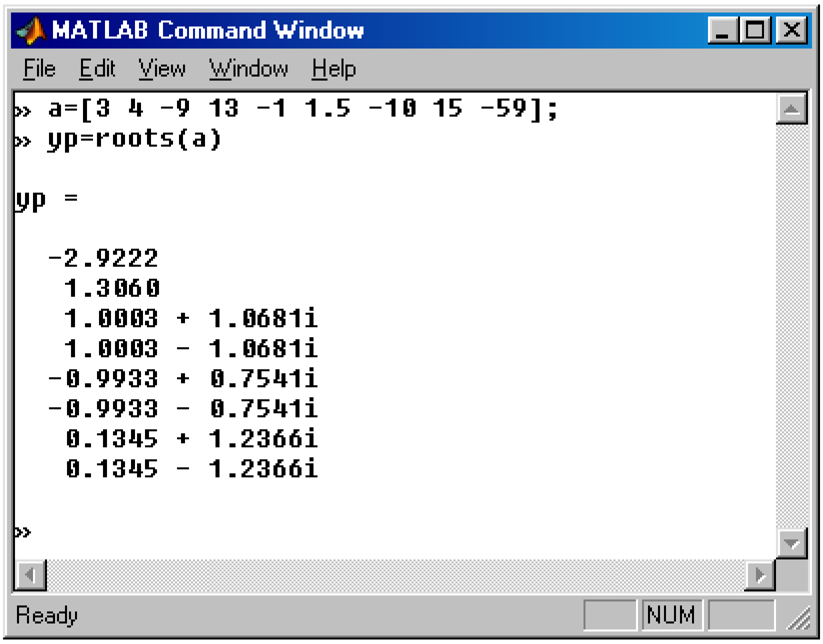
```
>> a=[3 4 –9 13 –1 1.5 –10.5 15 –5];
>>yp=roots(a)
```

veya

```
yp=roots( [3 4 –9 13 –1 1.5 –10.5 15 –59] )
```

bildirimi ile sonuç şu şekilde elde edilir.

X1= - 2.9170
X2= - 0.7475 + 0.7096i
X3= - 0.7475 – 0.7096i
X4= 0.6024 + 1.0396i
X5= 0.6024 – 1.0396i
X6= 0.7060 + 0.5552i
X7= 0.7060 – 0.5552i
X8= 0.4618



Resim 5: *Örneğin MATLAB ortamındaki çözümü ve çıktısı*

4.3 DOĞRUSAL DENKLEM TAKIMLARININ ÇÖZÜMLERİ

n. dereceden doğrusal denklem takımı

$$
\begin{aligned}
&a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\
&a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
&a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n
\end{aligned}
$$

biçiminde verilir. Bu denklem takımının matris biçiminde gösterimi

$$[A][X] = [B]$$

şeklinde tanımlanabilir. Burada;

$$A = \begin{bmatrix} a_{11} & a_{12} & .. & a_{1n} \\ a_{21} & a_{22} & .. & a_{2n} \\ .. & .. & .. & .. \\ a_{n1} & a_{n2} & .. & a_{nn} \end{bmatrix} \qquad B = \begin{bmatrix} b_1 & b_2 & .. & b_n \end{bmatrix}^T$$

katsayılar matrislerini ve

$$X = \begin{bmatrix} x_1 & x_2 & \ldots & x_3 \end{bmatrix}^T$$

çözümü istenen değişkenler matrisini (vektörünü) gösterir.

Bu tür istenen denklem takımlarının çözümü için, içinde yer alan özel fonksiyonlar yoktur. Bu denklemlerin çözümünde matris işlemlerinden yararlanılabilir. AX=B biçiminde verilen denklem takımının çözümünde A\B biçiminde soldan (bölen bölme işaretinin solunda yer almakta) matris bölme işlemi ile yerine getirilir. XA=B biçiminde tanımlanan matris denklemin çözümünde B/A şeklinde sağdan matris bölme işlemi kullanılır. Sağdan ve soldan matris bölme işleminde sayısal Gauss eliminasyon tekniği kullanılır.

Denklem Takımlarının Ters Matris İşlemi Yolu ile Çözümü

AX=B biçiminde verilen ve B'nin satır matrisi olarak tanımlandığı matris denkleminin her iki tarafını A^{-1} ile çarparsak

$$A^{-1}.AX=A^{-1}B$$

elde edilir. Burada A^{-1} A, I olarak tanımlanan birim matrise denktir. Buna göre

$$IX=A^{-1}B$$

veya

$$X=A^{-1}B$$

elde edilir.

MATLAB ortamında bu çözüm;

```
X=inv(A)*B
```

komutu ile elde edilebilir. Diğer taraftan B'nin sütun matrisi olarak tanımlandığı, XA=B biçiminde ifade edilen denklem takımının çözümü için, her iki taraf A^{-1} ile çarpılır ve gerekli düzenlemeler yapılırsa

$$X=BA^{-1}$$

elde edilir.MATLAB ortamında

```
X=B * inv(A)
```

bildirimi ile gerekli çözüm elde edilmiş olur.

Örnek : Aşağıda verilen denklem takımının çözümünü elde ediniz.

$$\begin{bmatrix} x_1+4x_2-x_3+x_4 & =2 \\ 2x_1+7x_2+x_3-2x_4 & =16 \\ x_1+4x_2-x_3+2x_4 & =1 \\ 3x_1-10x_2-2x_3+5x_4 & =-15 \end{bmatrix}$$

Çözüm : Çözüm ilk önce soldan ve sağdan matris bölme işlemlerine göre ele alınacak ve daha sonra da ters matris işlemine göre çözülecektir. AX=B biçiminde matris denklemi verildiğinde çözüm soldan bölme işlemine göre aşağıdaki bildirimlerle yerine getirilebilir.

```
a=[ 1  4  –1  1; 2  7  1  -2; 1  4  -1  2; 3  -10  -2   5 ]
b=[ 2  16  1  -15 ]
>>  x=a\b;
```

biçiminde tanımlandıktan sonra

```
x=inv(a)*b; ya da x=a\b
```

bildirimi ile çözüm elde edilir. Yukarıda verilen bildirimler yolu ile x çözümü için x= 2.0000 1.0000 3.0000 -1.0000

şeklinde elde edilmiş olur. Burada x_1= 2, x_2=1, x_3=3 x_4=-1 'dir.

6. GRAFİK İŞLEMLERİ

MATLAB grafik çizme konusunda geniş yetenekler sahip bir programdır. Bu bölümde anlatılacaklar basit çizimler içindir. En basit çizim iki değişkeni olan çizimlerdir. Bunlar için kullanılacak komut plot komutudur. Kullanımı aşağıdaki şekildedir.

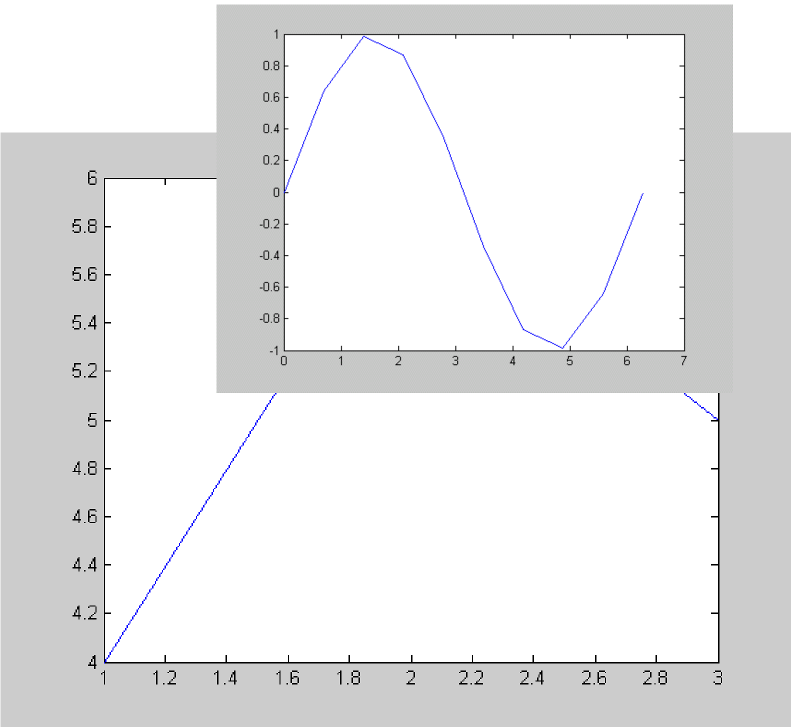
```
plot(x,y)
```

Bu komut x değişkenini ye değişkenine karşı gösteren bir grafik oluşturur. Eğer x ve ye matris şeklinde veriler ise bu durumda plot komutunun yapacağı iş önce ilk kolonları karşılaştırmalı çizmek ve sonar bu durumu her bir kolon için tekrar etmektir.

<u>Örnek :</u>

```
» x=linspace(0,2*pi,10);
» y=sin(x);
» plot(x,y)
```

BU kodun yaptığı iş, 0-2п aralığını 10 eşit parçada sinüs grafiği ile birleştirmektir.



<u>Örnek :</u>

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

>>plot(X, Y)

Ayrıca plot komutuna ekleyeceğimiz birkaç bilgi ile ne tür bir çizgi şekli ile çizeceğini ya da ne renk ile göstereceğini de ayarlayabiliriz. Bu tür bilgileri aşağıdaki tabloda bulabilirsiniz.

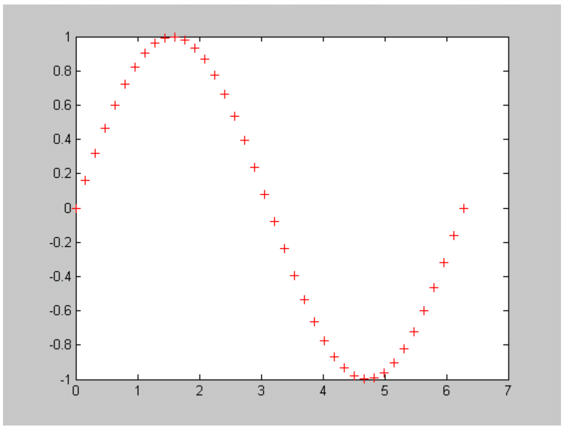
Renk	**Sembol**	**Çizgi Stili**	**Sembol**
Yellow	y	Point	.
Magenta	m	Circle	o
Cyan	c	X-mark	x
Red	r	Plus	+
Green	g	Star	*
Blue	b	Solid line	-
White	w	Dotted line	:
Black	k	Dash-dot line	-.
		Dashed line	--

Genel hali:

plot(x,y,'linetype')

şeklindedir. Örneğin;

» plot(x,y,'r+')



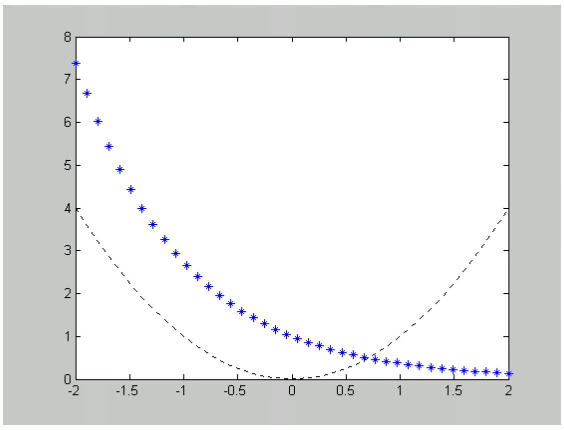
Eğer birden fazla grafik şeklini aynı kutuda görmek istiyorsak;

plot(x1, y1, 'linetype1', x2, y2, 'linetype2', …)

yapısını kullanabiliriz.

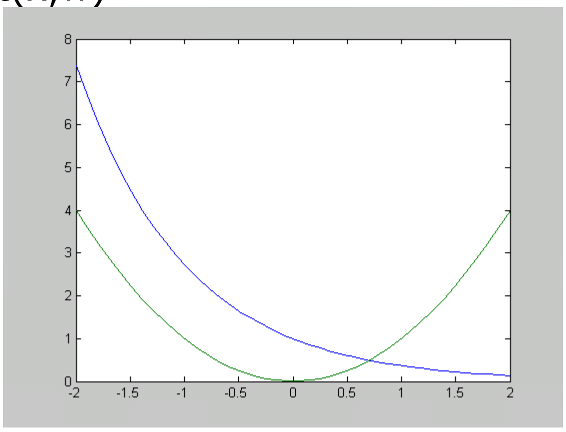
Örnek: $-2 \leq x \leq 2$ aralığında e^{-x} ve x^2 fonksiyonları için;

```
» x=linspace(-2,2,40);
» y=exp(-x);
» z=x.*x;
» plot(x,y,'b*',x,z,'k:')
```



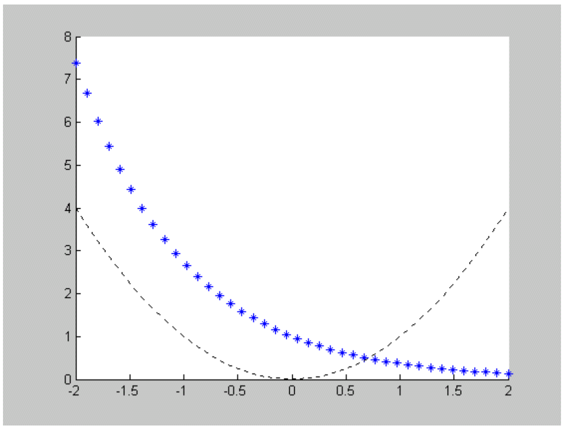
Ya da aşağıdaki ikinci yapıda aynı çıktıyı vermektedir.

```
» x=linspace(-2,2,40);
» w=[exp(-x);x.*x];
» plot(x,w)
```



Bu işlemi gerçekleştirmenin diğer bir yolu ise hold on komutunu kullanmaktır. Kullanım örneği;

```
» hold on
» x=linspace(-2,2,40);
» y=exp(-x);
» plot(x,y,'b*')
» z=x.*x;
» plot(x,z,'k:')
» hold off
```

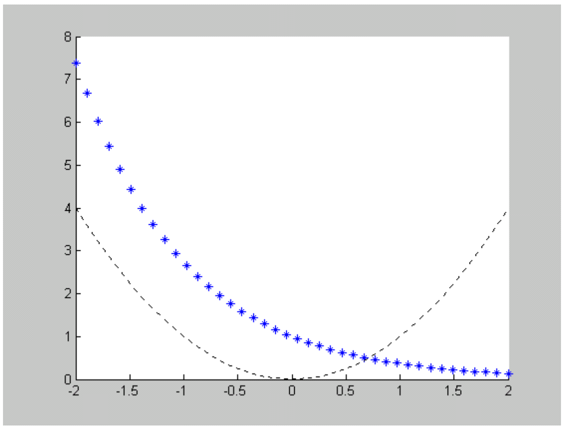


Bunu yanında çizdiğim grafikte çeşitli adlandırmalar yapabilmek için aşağıdaki notlar dikkate alınmalıdır.

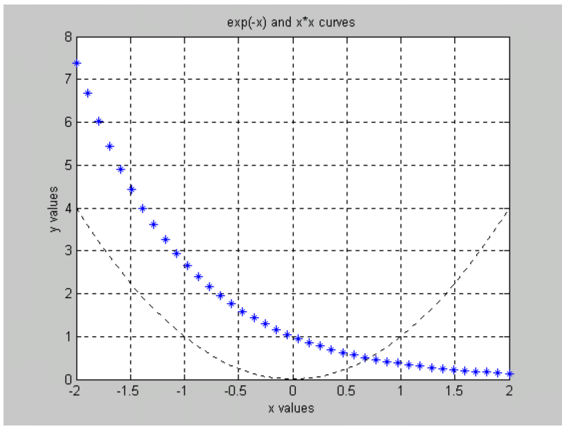
title('text')   Grafik üst tarafına başlık ekler
xlabel('text')   x-axisine isim verir
ylabel('text')   y-axisine isim verir
grid on   Grid çizgiler ekler
grid off   Grid çizgilerini kaldırır
text(x,y,'text')   (x, y) kesişim noktasını adlandırır
gtext('text')   Girdiğiniz yazı mouse ile tıkladığınız yerde belirir

Örnekler:

```
» x=linspace(-2,2,40);
» y=exp(-x);
» z=x.*x;
» plot(x,y,'b*',x,z,'k:')
```



```
» grid on
» xlabel('x values')
» ylabel('y values')
» title('exp(-x) ve x*x grafikleri')
```




```
» gtext('exp(-x)')
» gtext('x*x')
```