SEMBOLİK MATEMATİK

GİRİŞ

syms komutu

Sayısal ve Sembolik Denklemlerin Farkı

Sembolik Denklem Köklerini Bulma

Sembolik Türev Alma

Sembolik İntegral Alma

Çok Bilinmeyenli Denklemleri Çözme

GİRİŞ

- Bu derste bugüne kadar sayısal işlemler yaptık.
- Matlab ortamının en güçlü taraflarından bir tanesi de sembolik değişkenlerle işlem yapabilme yeteneğidir.
- Örneğin

```
x=a;
y=b;
z=x+y
```

olarak verildiğinde, Matlab z'nin değerini a+b olarak bulabilmektedir.

- Matlab, yukarıdaki örneğe benzer şekilde, x cinsinden verilmiş bir denklemin köklerini, sembolik işlemler yardımıyla bulabilmektedir.
- Dolayısıyla bu derste sembolik işlemler kullanarak, bilinmeyen bulma, türev ve integral alma gibi konuları Matlab ortamında nasıl yapabildiğimizi konuşacağız.

syms KOMUTU

- Herhangi bir değişkeni sembolik olarak kullanabilmek için, **syms** komutuyla kullanmazdan önce tanımlamamız gerekir.
- Genel kullanımı;

```
syms x
```

şeklindedir.

- Bu tanımlama yapıldıktan sonra herhangi bir denklem **x** cinsinden yazılabilir.
- Aşağıdaki örnek bu durumu açıklamaktadır:

```
syms x
f(x)=x^2+5;
fprintf('f=%s \n',f(x))
----------------------
f=x^2 + 5
```

x değişkenine değer atamak için yanda verilen koda, aşağıdaki eklemeler yapılabilir.

```
fprintf('f=%f \n',f(2))
--------------------
f=9
```

Sayısal ve Sembolik Denklemlerin Farkı

- Aşağıdaki iki örnek, aslında aynı hesaplamayı yapmaktadır.
- Ancak soldaki örnekte, önce x ve y'nin değerleri veriliyor ve toplama işlemi daha sonra yapılıyor.
- Soldaki örnek ise, önce semboller toplanıyor ve daha sonra sayısal değerler yerine konularak işlem tamamlanıyor.

Sayısal İşlem

```
x=5.2;
y=8.3;
f=x+y
----------------------
f =
    13
```

Sembolik İşlem

```
syms x y;
f(x,y)=x+y;
f(5,8)
----------------------
f=
    13
```

! syms komutuyla birden fazla değişken sembolik işlem tanımlanacaksa aray virgül konulmaz.

Sembolik Denklem Köklerini Bulma

Verilen n dereceli bir polinomun köklerini bulmak için **solve** komutu kullanılabilir.

Genel kullanımı:

```
solve(eşitlik, degişken)
```

şeklindedir. Burada **eşitlik** n dereceli polinomu, **değişken** ise bu fonksiyonun bu değişken için çözüleceğini gösterir. Eğer değişken verilmez ise, komut

```
solve(eşitlik)
```

şeklinde de yazılabilir.

Örnek:

```
solve(x^2 - 4*x + 2==0,x)
---------------------
ans =
     0.5858
     3.4142
```

Örnek:

```
solve(x^2 - 4*x + 2==0,x)
---------------------
ans =
     0.5858
     3.4142
```

Önemli Nokta:

Örnek:

```
syms x
f=x^3-3*x^2-7*x-3;
c=solve(f==0)
----------------------
c =
            -1
 2 - 7^(1/2)
 7^(1/2) + 2
```

Sol yanda sonuçların kesirli çıktığını görüyoruz. Sonuçları ondalıklı olarak göstermek için **double** komutunu kullanırız.

Örnek:

```
syms x
f=x^3-3*x^2-7*x-3;
c=solve(f==0)
double(c)
----------------------
ans =
   -1.0000
   -0.6458
    4.6458
```

Sembolik Türev Alma

Sembolik olarak verilmiş bir denklemin türevini `diff` komutuyla alabiliriz. Genel kullanımı:

```
diff(denklem, değişken)
```

Burada `denklem` sembolik olarak tanımlanmış bir denklemi gösterir iken, `değişken` türevin hangi değişkene göre alınacağını belirtir.

Örnek:

```
syms x
f=x^2-3*x+4;
diff(f,x)
----------------------
ans =
    2*x - 3
```

Sembolik İntegral Alma

Sembolik olarak verilmiş bir denklemin türevini `int` komutuyla alabiliriz. Genel kullanımı:

```
int(denklem, değişken)
```

Burada `denklem` sembolik olarak tanımlanmış bir denklemi gösterir iken, `değişken` integralin hangi değişkene göre alınacağını belirtir.

Örnek:

```
syms x
f=x^2-3*x+4;
int(f,x)
----------------------
ans =
   (x*(2*x^2 - 9*x + 24))/6
```

Sol yanda sonucun çarpılmamış halde verildiğini görüyoruz. Sonuçları çarpılmış halde alabilmek için sağ yanda gösterildiği gibi **expand** komutunu kullanırız.

Örnek:

```
syms x
f=x^2-3*x+4;
expand(int(f,x))
----------------------
ans =
   x^3/3 - (3*x^2)/2 + 4*x
```

Çok Bilinmeyenli Denklemleri Çözme

N sayıda bilinmeyeni olan n sayıda denklemin bilinmeyenlerini yine `solve` komutu kullanarak bulabiliriz. Genel kullanımı:

```
solve(denklemler, değişkenler)
```

Burada `denklemler` sembolik olarak tanımlanmış bir denklemleri gösterir iken, `değişkenler` hangi bilinmeyenlerin bulunacağını belirtir.

Örnek:

```
syms x y z
f1 = 3*x + 2*y -z - 10;
f2 = -x + 3*y + 2*z -5;
f3 = x - y - z +1;
solve(f1,f2,f3)
---------------------
ans =
  struct with fields:
    x: [1×1 sym]
    y: [1×1 sym]
    z: [1×1 sym]
```

Sol yanda sonucun sembolik olarak verildiğini görüyoruz. Sonuçları sayısal değerlerini alabilmek için sağ yanda gösterildiği gibi ilgili değişkenden önce . kullanırız.

Örnek:

```
syms x y z
f1 = 3*x + 2*y -z - 10;
f2 = -x + 3*y + 2*z -5;
f3 = x - y - z +1;
cevap=solve(f1,f2,f3);
cevap.x
cevap.y
cevap.z
---------------------
ans =
  -2
ans =
  5
ans =
  -6
```