MATLAB ile Meslek Matematiği Kullanım Kılavuzu

Tez Danışmanı: Yrd. Doç. Dr. MEHMET TEKTAŞ

HAZIRLAYANLAR

Ali Süleyman TOPUZ Bilgisayar Teknolojileri ve Programlama BS2 2722008

Bekir Hakan AYDOGAN Bilgisayar Teknolojileri ve Programlama BS2 2722007

Ön Lisans Bitirme Tezi

GÖZTEPE-2003

BÖLÜM 1: MATLAB KULLANIMI ve MATRİS İŞLEMLERİ

BÖLÜM 2: GRAFİK ve EĞRİ ÇİZİMLERİ

BÖLÜM 3: MATLAB ile MESLEKİ MATEMATİK

İÇİNDEKİLER

BÖLÜM 1

BÖLÜM 2

BÖLÜM 3

BÖLÜM 1

MATLAB Kullanımı ve Matris İşlemleri

1.1 Giriş

MATLAB programı (MATrix LABorartory ‘ nin ilk üç harfi alınarak isimlendirilmiştir.) mühendislik uygulamalarının, hesaplamalarının ve simülasyonlarının çoğunun gerçekleştirildiği matris ve matematik tabanlı karmaşık bir programdır. Her türlü grafiksel sonuçlar istenilen tarzda alınabildiği için kullanım alanı çok geniştir. Ayrıca MATLAB versiyonlarından en az 6.0 ve üzeri olanlarının kullanılması güncellik açısından daha yararlı olacaktır.

1.2 MATLAB KULLANIMI ve MATRİS İŞLEMLERİ

Bu bölümde programı kullanmaya başlamak için giriş komutları, matematiksel fonksiyonlar ve matris operatörleri anlatılacaktır. Ayrıca kılavuzun en son kısmında da en çok kullanılan matris komutları ve fonksiyonları tablo halinde verilmiştir.

Help ‘fonksiyon ismi’

Komutu yazıldığında yardım istenilen fonksiyon hakkında detaylı bilgiye ulaşılabilmektedir.

Help help

Yazıldığında online olarak yardım hizmetinin nasıl kullanılacağı hakkında bilgilere ulaşılabilmektedir.

1.2.1 Matris Operatörleri

+	Toplamak
-	Çıkarmak
*	Çarpmak
^	Kuvvet Almak
‘	Konjüge Transpozunu Almak

1.2.1 Mantık ve İlişki Operatörleri

<	Küçük	~=	Eşit Değil
<=	Küçük Eşit	&	Ve
>	Büyük	|	Veya
~	Değil		

Başlangıç Olarak Komut Satırına:

date

yazılırsa program tarafından geçerli olan tarih alınacaktır. Yani:

ans=

03-May-2009

Matlab bir işlemin (yazılan komutun) sonucunu ans=answer=cevap olarak gösterir. Matlab programından çıkmak için ise exit veya quit yazmak yeterli olacaktır. En son yazılan komutların hepsine üst ve alt yön tuşları kullanılarak kolay bir şekilde ulaşılabilir. En son tanımlanan herhangi bir 'x' değeri için yapılan işlemlerden sonra bu 'x değeri komut satırına yazılıp enter tuşuna basılırsa daha önce neye karşılık olarak tanımlandığı ekrana yazılacaktır.

n x 1 veya *1 x n* boyutunda vektör tanımlamak için:

X=[1 2 3 -4 -5]

Veya

X=[1,2,3,-4,-5] yazılmalıdır…

Yukarıdaki iki yazım biçiminden okuma kolaylığı olması için ilk yazılan tip kullanılacaktır…

Tanımlanan bu satır vektörünü sütun vektörüne dönüştürmek için :

Y=X' yazılırsa ekranda görülen değer aşağıdaki gibi olacaktır.

```
>> Y=X'

Y =

     1
     2
     3
    -4
    -5
```

Örnek:

Matris tanımlamak için A matrisi verilmiş olsun:

$$A = \begin{matrix} 1.2 & 10 & 15 \\ 3 & 5.5 & 2 \\ 4 & 6.8 & 7 \end{matrix}$$

Bu matrisi Matlab ile tanımlamak için komut satırına komut, şu şekilde yazılır:

```
>> A = [1.2 10 15 ; 3 5.5 2 ; 4 6.8 7]
```

Yani her satırın sonunun neresi olduğu, yukarıda da görüldüğü gibi ';' ile belirtilmiştir. Çıktısı şu şekilde olacaktır:

```
A =

    1.2000   10.0000   15.0000
    3.0000    5.5000    2.0000
    4.0000    6.8000    7.0000
```

Örnek:

Aşağıdaki B matrisini tanımlayacağız:

$$B = \begin{matrix} 1 & e^{-0.02} \\ \sqrt{2} & 3 \end{matrix}$$

Bu matrisi komut satırında bize verecek komut şu şekildedir:

```
>> B = [1 exp(-0.02); sqrt(2) 3]
```

Bu komut bize yukarıdaki matrisi Matlab de şu şekilde döndürür:

```
B =

    1.0000    0.9802
    1.4142    3.0000
```

İstenen matriste olduğu gibi fonksiyonlar matematiksel görünümleriyle değil, matematiksel değerleriyle geri dönmektedir.

Apostrof işareti (‘) matrisin Konjüge transpozesini almaya yaramaktadır… Eğer matris reel bir matris ise basit olarak transpoze alım işlemi olarak ta tanımlanabilir.

Örnek:

Yeni bir A matrisi tanımlayalım:

```
>> A = [1 2 3 ; 4 5 6 ; 7 8 9]

A =

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

Bu matrisin transpozesini almak için:

```
>> C=A'

C =

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

Görüldüğü üzere yeni bir C değerine eşitlenen A’ değeri, yandaki şekilde gözükmektedir.

1.2.2 MATLAB ile Komplex Sayıların kullanımı:

Komplex sayıların girilmesi için ise i^2=-1denkleminin kökü i veya j olarak tanımlanır.

Örnek:

Yandaki Komplex tanımı tanımlamak için: $1+j\sqrt{3}$

Komut satırımıza yazılacak komut şu şekilde olmalıdır:

```
>> X = 1+sqrt(3)*i
```

```
X =

   1.0000 + 1.7321i
```

Karşımıza sonuç olarak X tanımlaması yandaki şekildeki gibidir.

i ve j daha önceden değişken olarak kullanılmışsa tanımlama için ii ve jj kullanılacaktır. Yani :

ii = sqrt(-1)   ya da   jj = sqrt(-1)

Dolayısıyla aşağıdaki yazımda mümkün olmaktadır:

X=1+sqrt(3)*ii   ya da   X=1+sqrt(3)*jj

Örnek:

Komplex matris tanımlamak için aşağıdaki X matrisi verilmiş olsun:

$$X = \begin{matrix} 1 & j \\ -j5 & 2 \end{matrix}$$

Komut satırına şu şekilde girilecektir:

```
>> X = [1 j ; -j*5 2]
```

Bu durumda ekranda gösterilecek değer:

```
X =

   1.0000                  0 + 1.0000i
        0 - 5.0000i   2.0000
```

Transpozesi için komut satırı ve çıktısı şu şekilde olur:

```
>> Y=X'

Y =

   1.0000                    0 + 5.0000i
        0 - 1.0000i    2.0000
```

Daha önce de belirtildiği gibi yukarıdaki işlem Konjüge transpoze olarak algılanmaktadır. Eğer sadece transpoze alınacaksa (Konjügesiz) komut şu şekilde yazılmalıdır:

Y.’   veya   conj(Y’)

Bu durumda ekranda gözükecek değerler:

```
>> Y.'

ans =

   1.0000                    0 - 1.0000i
        0 + 5.0000i    2.0000
```

Örnek:

Toplama ve çıkarma işlemi yapmak için M ve N matrisleri verilmiş olsun.

$$M = \begin{matrix} 2 & 3 \\ 4 & 5 \\ 6 & 7 \end{matrix} \qquad N = \begin{matrix} 1 & 0 \\ 2 & 3 \\ 0 & 4 \end{matrix}$$

Komut satırındaki kodlar şu şekildedir:

```
>> M = [2 3 ; 4 5 ; 6 7];
>> N = [1 0 ; 2 3 ; 0 4];
>> M

M =

     2     3
     4     5
     6     7

>> N

N =

     1     0
     2     3
     0     4
```

Toplama işlemi için komut satırındaki kod ve çıktısı:

```
>> C=M+N

C =

     3     3
     6     8
     6    11
```

Not:

Eğer X vektörü aşağıdaki gibi verilirse:

$$X = \begin{matrix} 5 \\ 4 \\ 6 \end{matrix}$$

Komut satırı aşağıdaki gibi uygulanmalıdır. Çıktısı da görüntüdedir:

```
>> X = [5 ; 4 ; 6]

X =

     5
     4
     6
```

Örnek:

T=X-1 gibi işlemi gerçekleştirmek için:

```
>> X=[2 3 4 5 6 ]

X =

     2     3     4     5     6

>> X=X'

X =

     2
     3
     4
     5
     6

>> T=X-1;
>> T

T =

     1
     2
     3
     4
     5
```

Yandaki resimde ilk olarak bir X matrisi tanımlanmıştır. Aynı X'e, X'in transpozesi ikinci komutta atanmıştır. Üçüncü komutta T tanımlamasında X in her bir elemanının 1 eksiği matris olarak atanmıştır.

Son olarak T matrisi gösterilmiştir…

Matris çarpımı daha önce de belirtildiği gibi * çarpma operatörüyle yapılmaktadır. Aşağıdaki örnek incelenirse çarpmanın da tanımı gereği çarpılan matrislerin boyutlarının uyuşması gerekmektedir. Aksi takdirde çarpma işlemi yapılmayacak ve hata mesajı verilecektir.

```
>> x = [1 ; 2 ; 3];
>> y = [4 ; 5 ; 6];
>> A = [1 1 2 ; 3 4 0 ; 1 2 5];
>> x.*y

ans =

     4
    10
    18

>> b=A*x

b =

     9
    11
    20
```

Bunların dışında matris bir scaler değerle çarpılabilir:

```
>> 5*A

ans =

     5     5    10
    15    20     0
     5    10    25
```

Matris üssü (expm(A)) *nxn* matrise uygulanır.Matematiksel tanımı ise şu şekildedir:

$$\mathrm{expm}(A) = I + A + A^2/2! + A^3/3! + \ldots$$

Eğer A Kompleks bir matris ise abs(A) değeri de Kompleks modül değerler üzerinden hesaplanacaktır. Yine matematiksel ifadesine bakacak olursak:

$$\mathrm{abs}(A) = \mathrm{sqrt}(\mathrm{real}(A).\hat{}2 + \mathrm{imag}(A).\hat{}2)$$

angle(A) ise faz açılarını radyan cinsinden A Kompleks matrisi için hesaplamaktadır. Burada tanım değerleri –Π ve + Π arasında kabul edilmektedir.

Örnek:

Sonuç olarak verilen bir K matrisi için aşağıdaki uygulama incelenebilir:

```
>> A = [2+2*i 1+3*i ; 4+5*i 6-i];
>> abs(A)

ans =

    2.8284    3.1623
    6.4031    6.0828

>> angle(A)

ans =

    0.7854    1.2490
    0.8961   -0.1651
```

Bir vektörün elemanlarının teker teker karesinin alınması işlemi şu şekilde yapılmaktadır:

```
>> x = [1 2 3];
>> x.^2

ans =

     1     4     9
```

Eğer Komplex sayı mevcut ise:

```
>> y = [2+5*i 3+4*i 1-i];
>> y.^2

ans =

 -21.0000 +20.0000i  -7.0000 +24.0000i        0 - 2.0000i
```

2x2 bir kare matris olursa yine aynı şekilde:

```
>> A = [1 2 ; 3 4];
>> A.^2

ans =

     1     4
     9    16
```

Eleman elemana çarpma işlemi için çarpma operatörünün önüne bir nokta işareti (.) konmaktadır:

$$x = [1 \;\; 2 \;\; 3], \;\; y = [4 \;\; 5 \;\; 6]$$
$$z = [x.y]$$
$$z = [4 \;\; 10 \;\; 18]$$

Bir örnek daha verilecek olunursa:

$$A = \begin{matrix} 1 & 2 & 3 \\ 1 & 9 & 8 \end{matrix} \qquad B = \begin{matrix} 4 & 5 & 6 \\ 7 & 6 & 5 \end{matrix}$$

Gerekli komut satırı ve çıktısı:

```
>> a=[1 2 3;1 9 8];
>> b=[4 5 6;7 6 5];
>> c=a.*b

c =

     4    10    18
     7    54    40
```

Bir matrisin tüm elemanlarının tek tek karesini almak için:

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

A =

     1     2     3     4

>> A.^2

ans =

     1     4     9    16
```

Eleman elemana bölmek için ise çarpma işareti (*) yerine bölme işareti (/) kullanılır.

```
>> a=[1 2 3;1 9 8];
>> b=[4 5 6;7 6 5];
>> a./b

ans =

    0.2500    0.4000    0.5000
    0.1429    1.5000    1.6000
```

1.2.3 Bir Matrisin Tersini Bulmak:

Bir matrisin tersini bulmak için inv(A) komutu kullanılır:

Örnek:

```
>> A = [0 1 0;0 0 1;-6 -11 -6];
>> inv(A)

ans =

   -1.8333   -1.0000   -0.1667
    1.0000         0         0
         0    1.0000         0

>>
```

Yukarıdaki Matlab işleminde ilk önce A matrisi tanımlanmış olup, ondan sonra inv(A) komutu ile tersi alınmıştır. Sonuç görüldüğü gibidir.

Çeşitli komutlar ve durumlar tek bir sırada virgül (,) veya noktalı virgül ile (;) ayrılarak yazılabilir.

Çıkış formatını istediğimiz uzunlukta elde edebiliriz. Eğer matris elemanları tamsayı ise bu durum sonuçta bir değişiklik yapmayacaktır. Bunun için aşağıdaki komutları kullanmak gerekmektedir:

```
>> format short
>> inv(A)

ans =

   -1.8333   -1.0000   -0.1667
    1.0000         0         0
         0    1.0000         0
```

```
>> format long
>> inv(A)

ans =

  -1.833333333333333  -1.000000000000000  -0.166666666666667
   1.000000000000000                   0                   0
                   0   1.000000000000000                   0
```

Kısa ile uzun formattaki görünüm farkı yukarıdaki uygulama görüntüsünde belirtilmiş olup; genelde kullanılan formatın kısadan ziyade, küsuratın önemli olduğu uygulamalarda uzun format cazip gelmekte ve kullanılmaktadır.

1'den 5'e kadar sayıları 0,5'lik aralıklarla yazdırmak istersek iki nokta'yı (:) kullanmak yeterli olacaktır:

```
t =

     1     2     3     4     5

>> t=1:0.5:3

t =

    1.0000    1.5000    2.0000    2.5000    3.0000
```

Düzgün azalan biçimde yazdırırsak:

```
>> t = 5:-1:2

t =

     5     4     3     2
```

Bir matrisin i. satırını veya j. sütununu görüntülemek için aşağıda tanımlanan A matrisini komutlarıyla inceleyelim:

Aşağıdaki A matrisinin 2. satırı görüntülemek için:

```
>> A = [0 1 0;0 0 1;-6 -11 -6]

A =

     0     1     0
     0     0     1
    -6   -11    -6

>> A(2 , :)

ans =

     0     0     1
```

A matrisinin 3. sütununu görüntülemek için:

```
>> A(: , 3)

ans =

     0
     1
    -6
```

Bir matrisin (i,j) ninci elemanını bulmak için:

```
>> k = A(3,3)

k =

    -6
```

Bir matrisin boyutlarını öğrenmek için size(A) komutu, rankını bulmak için rank(A) kullanılır.

```
>> A=[2 3 2;5 4 1;2 6 8]

A =

     2     3     2
     5     4     1
     2     6     8

>> size(A)

ans =

     3     3

>> rank(A)

ans =

     3
```

Bir matrisin determinantını bulmak için det(A) komutu kullanılır.

```
>> det(A)

ans =

   -18
```

1.2.4 Matrisin Özdeğerleri:

Matrisin özdeğerlerini bulmak için eig(A) komutu kullanılır:

```
>> eig(A)

ans =

        0 + 1.0000i
        0 - 1.0000i
```

Öz vektörleri bulmak da tek satırlık bir işlem gerektirmektedir. Aslında özvektörleri bulmak için verilen [X,D] = eig(A) komutu aynı zamanda öz değerleri de bulduğu için her iki bilgiye aynı anda ulaşma imkanı olmaktadır .

Örnek:

```
>> A = [0 1 0 ; 0 0 1 ; -6 -11 -6]

A =

     0     1     0
     0     0     1
    -6   -11    -6

>> [X,D]=eig(A)

X =

   -0.5774    0.2182   -0.1048
    0.5774   -0.4364    0.3145
   -0.5774    0.8729   -0.9435


D =

   -1.0000         0         0
         0   -2.0000         0
         0         0   -3.0000
```

Burada X sonuç matrisinin her bir sütunu verilen A matrisinin bir öz değerini göstermektedir.

D sonuç matrisinin diyagonalindeki (köşegenindeki) elemanların her biri de verilen A matrisinin özdeğerlerini göstermektedir.

Verilen eş boyutlu farklı iki A ve B gibi matrisin genelleştirilmiş öz değerlerini ve öz vektörlerini bulmak için ise [X,D] = eig(A,B) komutu yazılmalıdır.

Karakteristik denklemi bulmak için poly(A) komutu kullanılır.

```
>> A = [0 1 0 ; 0 0 1 ; -6 -11 -6];
>> p = poly(A)

p =

    1.0000    6.0000   11.0000    6.0000
```

Burada görülen sonuç katsayıları karakteristik denklemin katsayılarıdır. Yani :

$$s^3 + 6s^2 + 11s + 6 = 0$$

Bir polinomun köklerini bulmak için roots(a) komutu yazılmalıdır. Yukarıdakikarakteristik denklemin köklerini bulmak istersek :

```
>> r=roots(p)

r =

   -3.0000
   -2.0000
   -1.0000
```

Örnek:

Polinomların çarpımı için conv(a,b) komutu kullanılır.

$$a(s) = s^2 - 20.6$$
$$b(s) = s^2 + 19.6s + 151.2$$

a(s) ve b(s) polinomlarını çarpmak için :

```
>> a=[1 0 -20.6];
>> b=[1 19.6 151.2];
>> conv(a,b)

ans =

  1.0e+003 *

    0.0010    0.0196    0.1306   -0.4038   -3.1147
```

Dolayısıyla çarpım sonucu şu şekilde yazılır:

$$c(s) = s^4 + 19.6s^3 + 130.6s^2 - 403.8s - 3114.7$$

Bir polinomda herhangi bir tamsayı değerini hesaplatmak için polyval(c) komutu kullanılır :

Örnek:

Aşağıdaki polinomda s=5 için değer ne döner? $p(s) = 3s^2 + 2s + 1$

```
>> p=[3 2 1];
>> polyval(p,5)

ans =

    86
```

1 ve 0 sayılarının istenilen matrisel boyutta çabuk olarak üretilebilmesi için ones(m,n) ve zeros(m,n) komutları kullanılabilir ayrıca Birim matris de eye(n) komutuyla istenilen boyutta oluşturulabilir :

```
>> ones(2,2)

ans =

     1     1
     1     1

>> zeros(3,3)

ans =

     0     0     0
     0     0     0
     0     0     0

>> eye(5)

ans =

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

Bir matrisin köşegenindeki elemanları listelemek için diag(A) komutu kullanılır :

```
>> A = [1 2 3 ; 4 5 6 ; 7 8 9] ;
>> diag(A)

ans =

     1
     5
     9
```

Köşegenin elemanları haricindeki matris bileşenleri 0 olarak göstermek istersek :

```
>> diag(diag(A))

ans =

     1     0     0
     0     5     0
     0     0     9
```

Köşegen matrisi oluşturmayla alakalı aşağıdaki diğer örnekler de incelenebilir :

```
>> diag(1:5)

ans =

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

>> diag(0:4)

ans =

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

Bir matrisi rastgele olarak oluşturmak için rand(n) komutu kullanılır.0 ile 1 arasındaki sayıları alır.

```
>> rand(4)

ans =

    0.6948    0.4387    0.1869    0.7094
    0.3171    0.3816    0.4898    0.7547
    0.9502    0.7655    0.4456    0.2760
    0.0344    0.7952    0.6463    0.6797

>> rand(5)

ans =

    0.6551    0.3404    0.5060    0.1386    0.8143
    0.1626    0.5853    0.6991    0.1493    0.2435
    0.1190    0.2238    0.8909    0.2575    0.9293
    0.4984    0.7513    0.9593    0.8407    0.3500
    0.9597    0.2551    0.5472    0.2543    0.1966

>> rand(6)

ans =

    0.2511    0.5497    0.5678    0.1299    0.7943    0.6541
    0.6160    0.9172    0.0759    0.5688    0.3112    0.6892
    0.4733    0.2858    0.0540    0.4694    0.5285    0.7482
    0.3517    0.7572    0.5308    0.0119    0.1656    0.4505
    0.8308    0.7537    0.7792    0.3371    0.6020    0.0838
    0.5853    0.3804    0.9340    0.1622    0.2630    0.2290
```

BÖLÜM 2

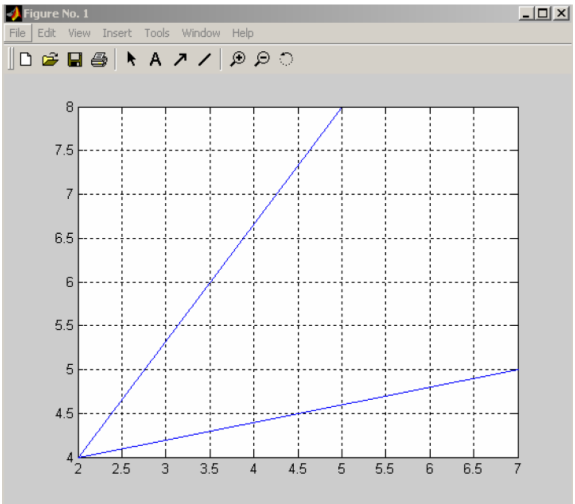
GRAFİK ve EĞRİ ÇİZİMLERİ

Örnek olarak X ve Y vektörleri olsun. Bu vektörler aynı boyutta ise, bu vektörleri ekrana çizdirmek için plot(x,y) komutu kullanılır.

```
A =[ 7  2  5];
B =[ 5  4  8 ];
plot(A,B);
grid
```

Bu durumda grafik ekrana aşağıda gösterildiği gibi otomatik olarak çizdirilecektir:

Ayrıca plot(X,Y,’x’) komutu çizilen eğriyi ‘x’ karakterini kullanarak çizmektedir.



2.1 GRAFİK Çizimiyle ilgili bazı özellikler:

2.1.1 X=3:0.5:10:

Seçilen bir parametreye göre (burada x parametresi seçilmiş) çizdirilmesi planlanan eğrinin sınırları yukarıdaki gibi yazılır. 3 ve 10 değerleri çizdirilmek istenen aralığı, ortadaki 0.5 değeri artış miktarını göstermektedir.

2.1.2 Grid:

Grafik arka yüzünün ölçekli olarak gösterilmesini sağlar. Izgara manasına gelir. Yukarıdaki şekilde mavi çizgilerin (grafiğin sonucunun) arka kısmındaki noktalı işaretlemelerle oluşmuş şekle denir.

2.1.3 Title('…')

Çizilen grafiğe başlık yazmak için kullanılır.

2.1.4 Xlabel('…')

Çizilen grafiğin x-eksenine istenilen açıklamayı yazmak için kullanılır.

2.1.5 Ylabel('…')

Çizilen grafiğin y-eksenine istenilen açıklamayı yazmak için kullanılır.

2.1.6 text('X,Y,'text')

Grafik ekranı üzerine istenilen koordinatlar dahilinde herhangi bir açıklama yazmak için kullanılır.

2.1.7 + * o x

İstenildiği takdirde çizilen eğrinin düz çizgi olarak değil de farklı karakterlerle çizdirilebilir. Bunlar için ise yukarıda gösterilen nokta, artı, yıldız, yuvarlak ve x karakterleri kullanılır. Bu karakterleri plot() komutu içerisinde '+' şeklinde yazmak yeterli olacaktır.

2.1.8 r g b w i

Çizilen eğrinin rengi de yukarıda gösterilen kısaltmalarla değiştirilebilir. Burada 'r' kırmızı renk (red), 'g' yeşil renk (green), 'b' mavi renk (blue), 'w' beyaz renk (white) ve 'i' ise (invisible) olarak kısaltımıştır.

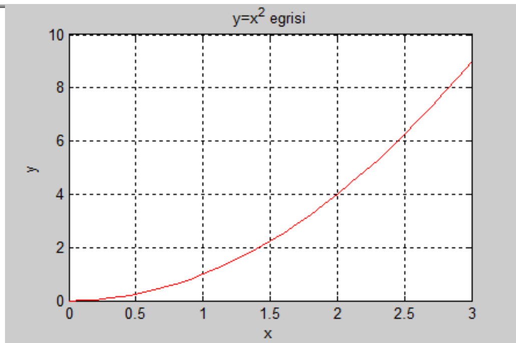
Not : Bu özellikler ve daha farklı görüntü özellikleri grafik ekranı üzerindeki "Insert" ve "Tools" menüleri aracılığıyla komut satırını kullanmadan da yapılabilmektedir.

Örnek:

Aşağıdaki örnekte ise y = x^2 eğrisini 0 ve 3 aralığında çizdirelim :

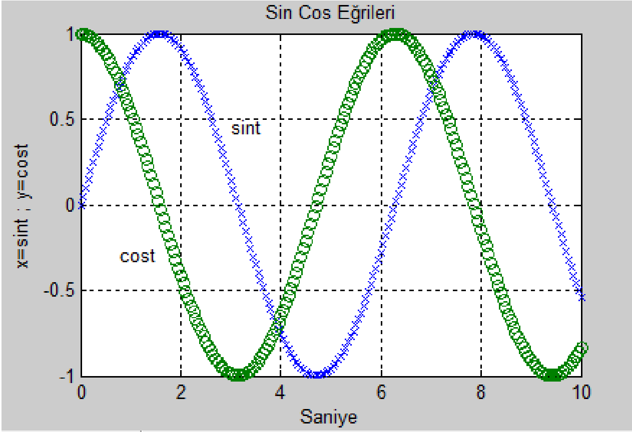
Kodumunuzun doğrultusunda karşılaşılan grafik şu şekildedir:

```
>> x = 0:0.1:3;
>> y = x.^2;
>> plot(x,y,'r');
>> title('y=x^2 egrisi');
>> xlabel('x');
>> grid;
>> ylabel('y');
>> xlabel('x');
>>
```



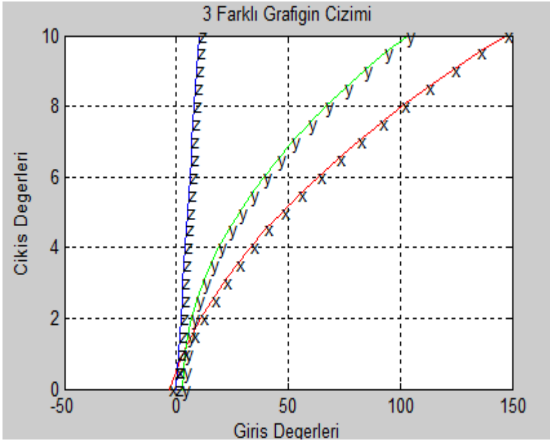
Birden fazla eğriyi tek bir grafik ekranı üzerinde görmek için çizdirilmesi istenen eğriler aynı plot(...) komutu içinde yazılmalıdır. Birden fazla eğriyi üst üste çizme uygulaması olarak aşağıdaki örnekte sin(x) ve cos(x) eğrileri tek bir grafik ekranı üzerinde çizdirilmiştir:

```
>> t = 0:0.05:10;
>> x = sin(t);
>> y = cos(t);
>> plot(t,x,'x',t,y,'o');
>> grid;
>> title('Sin Cos Eğrileri');
>> xlabel('Saniye');
>> ylabel('x=sint ; y=cost');
>>
>> text(3,0.45,'sint');
>> text(0.8,-0.3,'cost');
>>
```



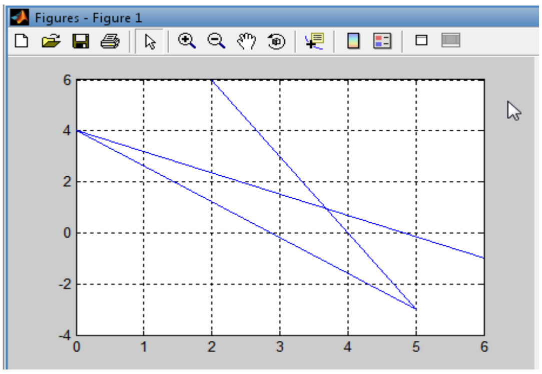
Örnek:

```
>> t=0:0.5:10;
>> x=(t.^2)+(5.*t)-3;
>> y=t.^2+3;
>> z=t;
>> plot(x,t,'r',y,t,'g',z,t,'b');
>> grid;
>> title('3 Farklı Grafigin Cizimi');
>> xlabel('Giris Degerleri');
>> ylabel('Cikis Degerleri');
>> text(x,t,'x');
>> text(y,t,'y');
>> text(z,t,'z')
>>
```



Kompleks vektörlerin çiziminde plot(z) ifadesi kullanılır. Çizim işleminde ise reel ve imajiner kısımlar ayrı ayrı ikili noktalar olarak kabul edilir:

```
>> C=[2+6i 5-3i 4i 6-i ];
>> plot(C)
>> grid
>>
```

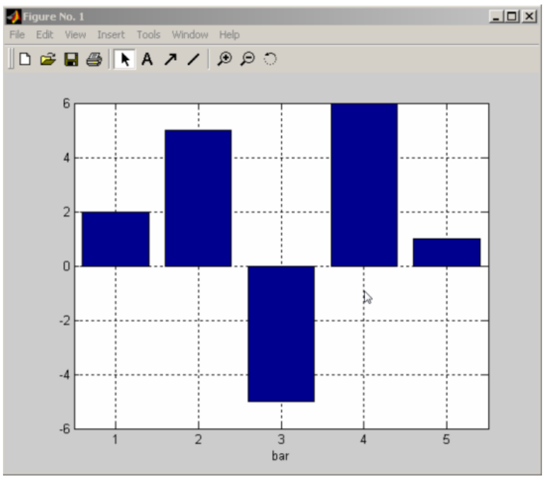


Not: loglog(X) komutu hem x eksenini hem de y eksenini logaritmik ölçeklendirmeyi kullanarak X'in grafiğini çizdirir.

Bir A vektörünü " bar grafiklerini " kullanarak çizdirmek için bar(A) komutu kullanılır. " Basamak " fonksiyonu şeklinde çizilecek ise stairs(A) komutu kullanılır. Her iki çizime ait örnek grafikler aşağıda ayrı ayrı verilmiştir :

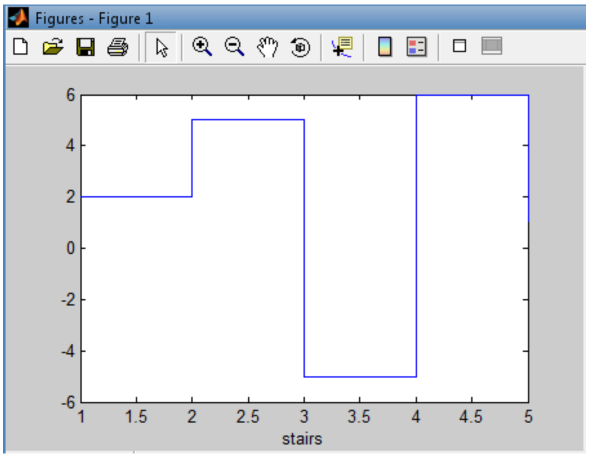
Örnek:

```
>> A = [ 2 5 -5 6 1 ];
>> bar(A);
>> grid;
>> xlabel('bar');
```



Kodlamaya aşağıdaki satırlar eklenince görünüm yanındaki gibi olacaktır.

```
>> stairs(A);
>> xlabel('stairs');
>>
```

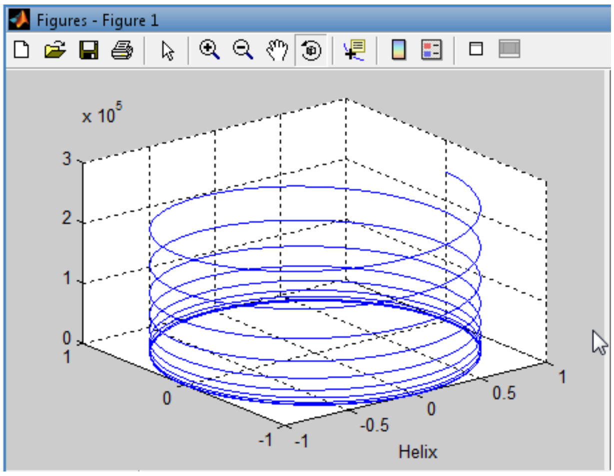


Ayrıca grafik ekranındaki menülerden yararlanarak çeşitli görüntü değişiklikleri yapılabilir. Örnek olarak “ Tools ” menüsünde “ Rotate-3D ” seçeneği kullanılarak mouse yardımıyla iki üstteki “ bar ” grafiğinin görüntüsü aşağıdaki gibi elde edilebilir.
Grafik çiziminde grafik çizgi tipleri, işaretler ve renkler aşağıdaki tabloda sıralanmıştır:

Sembol	Renk(RGB)	Çizgi stili	Sembol	Nokta stili	
Y	Sarı(110)	.	Nokta	-	Çizgi
M	Magenta(101)	O	Yuvarlak	:	Noktalı
C	Cyan(011)	X	Çarpı işareti	-.	Çizgili, Noktalı
R	Kırmızı(100)	+	Artı işareti	- -	Kesik Çizgili
G	Yeşil(010)	*	Yıldız		
B	Mavi(001)	S	Karekök		
W	Beyaz(111)	D	Baklava		

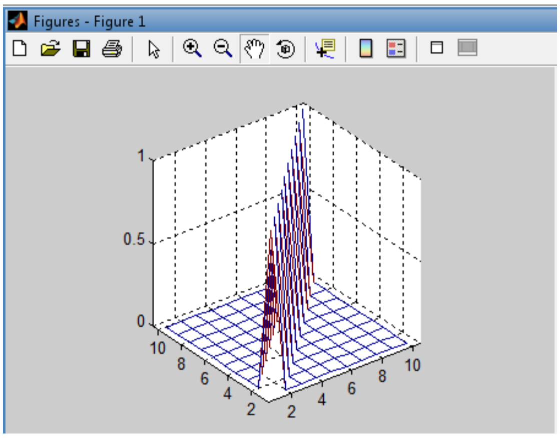
‘3-D Line’ (3 Boyutlu düz çizgi) çizimi için plot3(...) komutu kullanılır .Aşağıda heliks çizimi programı verilmiştir :

```
>> t=0.01:0.01:20*pi;
>> x=cos(t);
>> y=sin(t);
>> z=t.^3;
>> plot3(x,y,z);
>> xlabel('Helix');
>> grid
```

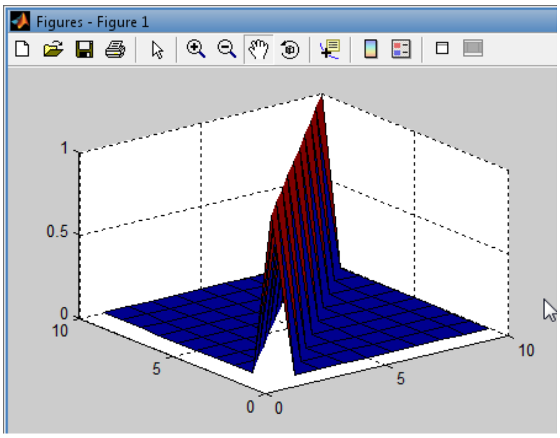


3 boyutlu ağ ve yüzey çizimlerinde kullanılan komutlardan biri mesh(...) komutudur. Bu komut verilen girişi z bileşeni olarak algılar ve dikdörtgen x-y düzlemi üzerinde z ekseni boyunca çizim yapar. surf(...) komutu ise aynı işi yüzey olarak yapar.Aşağıdaki komut satırlarının çizim görüntüleri yine alt tarafında verilmiştir.

```
>> mesh(eye(10));
>> grid
>> grid
```



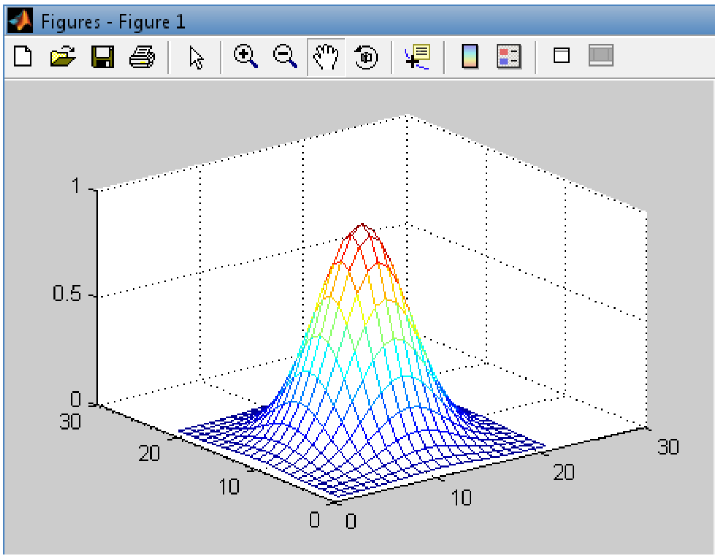

```
>> surf(eye(10));
>> |
```



Örnek:

z=exp(-x2-y2) fonksiyon yüzeyini [-2,2]x[-2,2] tanım aralığında 3 boyutlu olarak çizdirelim :

```
>> x=-2:0.2:2;
>> y=x;
>> [x,y]=meshgrid(x,y);
>> z=exp(-x.^2-y.^2);
>> mesh(z)
```



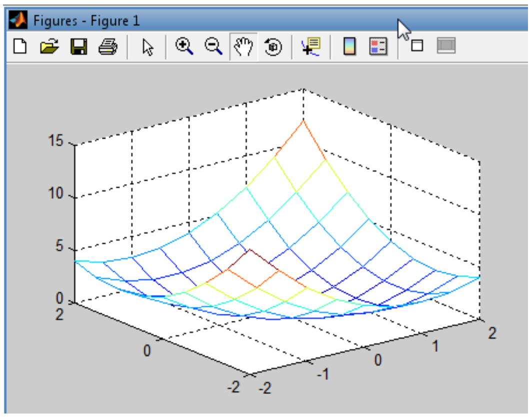
Ayrıca view komutu yardımıyla da küresel ve kartezyen koordinatlar ekranda görüntülenebilir.

```
>> view

ans =

    0.7934   -0.6088         0   -0.0923
    0.3044    0.3967    0.8660   -0.7835
    0.5272    0.6871   -0.5000    8.3031
         0         0         0    1.0000
```

Örnek:

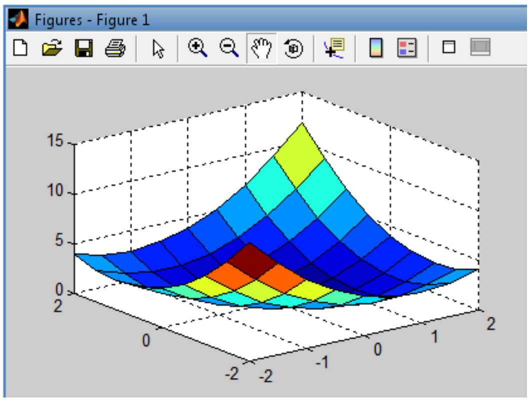
Örnek olarak $z=x^2+y^2+xy$ yüzeyini -2<x<2 ve -2<y<2 aralığında çizdirelim :

```
>> [X,Y]=meshgrid(-2:0.5:2,-2:0.5:2);
>> Z=X.^2+Y.^2+X.*Y;
>> mesh(X,Y,Z)
```



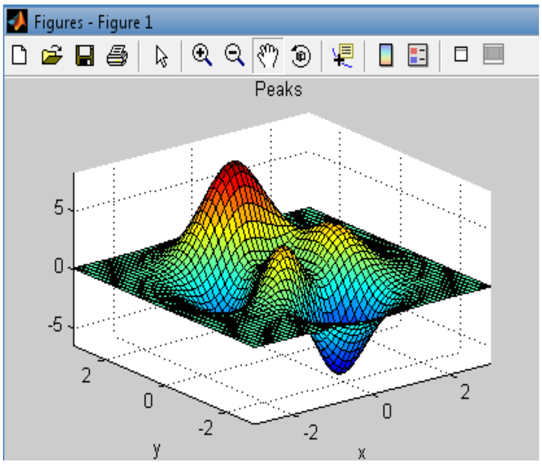
Yukarıdaki örnekte çizim fonksiyonu olarak mesh(X,Y,Z) yerine surf(X,Y,Z) çizim fonksiyonu kullanılırsa grafik yüzeyi aynı fakat her bir karesi farklı renklere boyanmış şekilde çizilecektir:

```
>> [X,Y]=meshgrid(-2:0.5:2,-2:0.5:2);
>> Z=X.^2+Y.^2+X.*Y;
>> mesh(X,Y,Z)
>> surf(X,Y,Z)
```



Herhangi bir yüzey grafiğinde tepe ve alt tepe (minimum ve maximum) değerlerini göstererek yapılan çizimlerde peaks(...) komutu kullanılır :

```
>> [X,Y]=meshgrid(-3:0.125:3);
>> peaks(X,Y)

z =  3*(1-x).^2.*exp(-(x.^2) - (y+1).^2) ...
   - 10*(x/5 - x.^3 - y.^5).*exp(-x.^2-y.^2) ...
   - 1/3*exp(-(x+1).^2 - y.^2)
```



BÖLÜM 3

MATLAB ile MESLEKİ MATEMATİK

3.1 Polinomlar:

Polinomlar genellikle tek değişkenli ve sabit katsayılı fonksiyonlardır. Bu fonksiyonların en genel hali aşağıda verilmiştir.

$$a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0 = 0$$

Denklemde verilen eşitliğin kökleri gerçek veya karmaşık sayı olabilirler.

3.2 Polinom Kökleri:

MATLAB'da polinom köklerini bulmak için ilk önce katsayılar dizini Denklem' deki gibi oluşturulur daha sonra *roots*(k) yazılarak sonuçlar elde edilir.

$$k = \left[a_n, a_{n-1}, \ldots, a_1, a_0\right]$$

Örnek:

$4x^5 - x^3 + 2x^2 - x - 20 = 0$ denkleminin köklerini MATLAB yardımı ile hesaplayınız.

```
>> k=[4, 0, -1, 2, -1, -20]

k =

     4     0    -1     2    -1   -20

>> roots(k)

ans =

   1.3825
   0.4639 + 1.3186i
   0.4639 - 1.3186i
  -1.1552 + 0.7187i
  -1.1552 - 0.7187i
```

3.3 Kökleri Bilinen Polinomun Katsayılarının Bulunması:

Bir polinomun çözüm kümesi biliniyor ise MATLAB onu bir polinom halinden de istenirse yazabilir. *poly*([kök1, kök2, kök3]) yazılarak türetilen polinomun katsayıları elde edilir.

Örnek:

Kökleri $x1 = 4$, $x2 = -1$ve $x3 = 1$ olan bir polinom türetiniz.

```
>> poly([-1,1,4])

ans =

     1    -4    -1     4
```

Bunun anlamı $x^3 - 4x^2 - x - 4 = 0$ yani polinom elde edilmiş olur.

3.4 Polinom Değerini Bulmak:

Eğer bir polinomun verilen herhangi bir değişken değerine karşı gelen polinom değerini bulmak istersek *polyval*( ) komutu kullanılır.

Örnek:

$f(x) = 4x^5 - x^3 + 2x^2 - x - 20$ fonksiyonunun $x = 3$ iken değerini hesaplayınız.

```
>> k=[4, 0, -1, 2, -1, 20]

k =

     4     0    -1     2    -1    20

>> polyval(k,3)

ans =

   980
```

3.5 Polinom Çarpımı ve Bölme:

Polinom çarpımı, bölme ve çıkarma işlemlerinden daha zordur. Bu işlemlerin daha kolay yapılabilmesi adına çarpma için MATLAB *conv*( ) komutunu ve bölme işlemi için de *deconv*() komutunu kullanılır. Burada *deconv* kullanılırken biraz dikkat etmek edilmesi gereken nokta tam bölünememe durumudur. Eğer verilen polinomlar tam bölünmüyor ve fonksiyonun kalanının gösterilmesi isteniyor ise bu durum da [a,b]=*deconv*(f,g) şeklinde bir komut kullanılmadır. Burada *a*, bölünen polinomun katsayılarını, *b* ise bölünmeyen kısmın katsayılarını verir.

Örnek:

$f(x) = 2x^3 - 5x^2 + 3^x - x$ ile $g(x) = x - 1$ olarak verilirse $f(x)\, g(x)$ve $f(x) / g(x)$ fonksiyonlarını MATLAB ile hesaplayınız.

```
>> fk=[2 -5 3 -3];
>> gk=[ 1 -1];
>> capma=conv(fk,gk)

capma =

     2    -7     8    -6     3

>> [bolme,r]=deconv(fk,gk)

bolme =

     2    -3     0


r =

     0     0     0    -3
```

3.6 SAYISAL TÜREV(Numeric Derivative):

Türev matematiksel olarak bir *f* (*x*) fonksiyonunun *x*' e göre değişim oranı olarak tanımlanır ve Denklem' deki gibi gösterilir.

$$f'(x) = \lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x} = \frac{df(x)}{dx}$$

MATLAB' da türev işlemi *polyder*() ve *diff*(y) fonksiyonlarıyla yapılır. Polinomların türevi alınacak ise *polyder*(katsayılar) komutu ile kullanılır. Eğer bir fonksiyonu türevi alınacak ise *diff*() komutu kullanılır. n. dereceden türev *diff*(f,n) olarak verilir.

Bir fonksiyonun bilinmeyen parametrelerine göre türevinin alınması için Jacobian matrisin oluşturulması gerekir. Bunun için '*jacobian'* komutu kullanılır.
Adı diferansiyel denklemlerin çözümü için '*dsolve'* komutu kullanılmaktadır.

Örnek:

Soru 1: f = sin(5*x) fonksiyonunu Matlab ile çözümlenmesi;

```
>> syms x;
>> f=sin(5*x);
>> diff(f)

ans =

5*cos(5*x)
```

Soru 2: g = exp(x)*cos(x) fonksiyonunu Matlab ile çözümlenmesi;

```
>> g = exp(x)*cos(x);
>> diff(g)

ans =

exp(x)*cos(x)-exp(x)*sin(x)
```

Soru 3 (Bir sabitin diferansiyel sonucu): c=5 sabitinin Matlab ile çözümlenmesi;

```
>> c = sym('5');
>> diff(c)

ans =

0
```

3.7 Çok Değişkenli Türemiş ifadelerin işlenmesi (Derivatives of Expressions with Several Variables):

Birden çok değişkenli(parametreli) türemiş ifadelerin hesaplanması yine diff fonksiyonuyla olmaktadır. Parametreleri tanımlarken kullandığımız 'syms' ile arasında boşluklar bırakarak parametrelerimizi tanımlarız.

Örnek:

f = sin(s*t) fonksiyonunun Matlab ile çözümlenmesi;

```
>> syms s t
>> f = sin(s*t);
>> diff(f,t)

ans =

cos(s*t)*s
```

Diferasiyel (differantial) Tablosu:

	f	Diff(f)
1	**x^n**	x^n*n/x
2	**sin(a*t+b)**	cos(a*t+b)*a
3	**exp(i*theta)**	i*exp(i*theta)

Birinci örnekte Matlab otomatik olarak cevabı yönlendirmez. Cevabı doğru bir şekilde görüntüleyebilmek için şu şekilde yazılır;

```
>> simplify(diff(x^n))

ans =

x^(n-1)*n
```

	f	Diff(f)
1	df/dx	diff(f) , diff(f,x)
2	df/da	diff(f,a)
3	d^2f/db^2	diff(f,b,2)

3.8 LİMİTLER(Limits):

1.)

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Matlab ile tanımlaması şu şekildedir:

```
>> syms h n x
>> limit( (cos(x+h) - cos(x))/h,h,0 )

ans =

-sin(x)
```

2.)

$$\lim_{x \to 0^-} \frac{x}{|x|}$$

Matlab ile tanımlaması şu şekildedir:

```
>> limit(x/abs(x),x,0,'left')

ans =

-1
```

3.)

$$\lim_{x \to 0^+} \frac{x}{|x|} = 1$$

Matlab ile tanımlanması şu şekildedir:

```
>> limit(x/abs(x),x,0,'right')

ans =

1
```

Limit(Limit) Tablosu:

Mathematical Operation	MATLAB Command
$\lim_{x \to 0} f(x)$	limit(f)
$\lim_{x \to a} f(x)$	limit(f,x,a) ya da limit(f,a)
$\lim_{x \to a^-} f(x)$	limit(f,x,a,'left')
$\lim_{x \to a^+} f(x)$	limit(f,x,a,'right')

3.9 İNTEGRAL(Integration):

Simgesel integral alma fonksiyonu '*int*' genel kullanım şekli:
***int*(s) :** *findsym* ile belirlenen simgesel değişkene göre S'nin belirsiz integralini alır
***int*(s,v) :** S'nin v'ye göre integralini alır.
***int*(S,a,b) :** S'nin varsayılan değişkene göre a' dan b' ye kadar belirli integralini alır.
***int*(S,v,a,b) :** S'nin tanımlı a' dan b' ye kadar belirli integralini alır.

Mathematical Operation	MATLAB Command
$\int g(t)dt = \sin(at+b)/a$	int(x^n) ya da int(x^n,x)
$\int_0^{\pi/2} \sin(2x)dx = 1$	int(sin(2*x),0,pi/2) ya da int(sin(2*x),x,0,pi/2)
$\int x^n\,dx = \frac{x^{n+1}}{n+1}$	g = cos(a*t + b) ya da int(g) ya da int(g,t)
$\int J_1(z)dz = -J_0(z)$	int(besselj(1,z)) ya da int(besselj(1,z),z)

Matlab de parametreleri tanımladıktan sonra:

```
>> syms a b theta x y n u z
>>
```

f	int(f)
x^n	x^(n+1)/(n+1)
y^(-1)	log(y)
n^x	1/log(n)*n^x
sin(a*theta+b)	-1/a*cos(a*theta+b)
1/(1+u^2)	atan(u)
exp(-x^2)	1/2*pi^(1/2)*erf(x)

Örnek:

∫(−2x5−4x+20)dx integralinin hesaplanması;

```
>> syms x
>> int(-2*x^5-4*x+20)

ans =

-1/3*x^6-2*x^2+20*x


>> pretty(int(-2*x^5-4*x+20))

                                              6       2
                                      - 1/3 x  - 2 x  + 20 x

>> |
```

Pretty fonksiyonu içerisinde tanımlanan int(integral) fonksiyonu hiyerarşik bir görünüm sağlar.

3.11 LAPLACE Dönüşümü(Transformation):

Matematikte sınır değer problemleri dahil diferansiyel denklemleri çözmekte ve olasılık teorisinde mühendislik alanında zamandan bağımsız doğrusal sistemleri modellemekte kullanılan bir dönüşümdür. Genel anlamda bir fonksiyonun tanım kümesini zamandan frekansa çevirir. Zaman tanım kümesinde çözmesi zor olan differensiyal denklemler frekans tanım kümesinde daha basit cebirsel denklemlere dönüştüğünden diferansiyel denklemleri çözmekte kullanılırlar. Söz konusu metot, kolay çözüm avantajına karşın ters Laplace dönüşümünün zorluğu ile dengelenir. Laplace dönüşümünün frekans karakterlerini net bir şekilde göstermesinden dolayı sinyal işlemede kullanılır.

3.11.1 Matlab ile Laplace Dönüşümü:

Fonksiyon f(t)'nin Matlab ile Laplace F(s) dönüşümü oldukça basittir. Öncelikle belirtmek gerekir ki değişken t, sembolik kullanılır. Bu komut ile yapılır:

```
>> syms t s
```

Daha sonra bu fonksiyonu tanımlanır: f (t). Gerçek komut dönüşümü hesaplamak için:

```
>> F=laplace(f,t,s)
```

İfadelerin okunabilirliğini ve anlaşılabilirliğini artırabilmek için simplify ve Pretty fonksiyonlarını kullanabiliriz.

Örnek:

$f(t) = -1.25 + 3.5te^{-2t} + 1.25e^{-2t}$ fonksiyonun dönüşümünü Matlab transformasyon ile gerçekleştiriniz.

```
>> syms t s
>> f=-1.25+3.5*t*exp(-2*t)+1.25*exp(-2*t);
>> F=laplace(f,t,s)

F =

(s-5)/s/(s+2)^2


>> simplify(F)

ans =

(s-5)/s/(s+2)^2


>> pretty(ans)

                                                  s - 5
                                                ----------
                                                         2
                                                s (s + 2)
>> |
```

Yukarıdaki sonuç alttaki F(s) fonksiyonu karşılamaktadır.

$$F(s)=\frac{(s-5)}{s(s+2)^2}$$

Alternatif olarak aşağıdaki Matlab komut satırıyla da aynı işlemi yapabiliriz.

```
>> F2=laplace(-1.25+3.5*t*exp(-2*t)+1.25*exp(-2*t))
```

Yine aynı şekilde devam ederek sonuca ulaşabiliriz.

```
F2 =

(s-5)/s/(s+2)^2


>> simplify(F2)

ans =

(s-5)/s/(s+2)^2


>> pretty(ans)

                                  s - 5
                                ----------
                                         2
                                s (s + 2)
```

3.11.2 Ters Laplace Dönüşümü(Inverse Laplace Transform):

Ters Laplace için ‘İlaplace’ fonksiyonu kullanılır. Ayrıca ‘t’ ve ‘z’ sinonimleri tanımlamak gerekmektedir. Bir önceki F(s) fonksiyonunun Inverse Laplace fonksiyonunu hesaplayalım:

$$F(s) = \frac{(s-5)}{s(s+2)^2}$$

İlk olarak sinonimlerimizi tanımlayarak işleme başlıyoruz. Daha sonra f fonksiyonumuzu yani f(s) tanımlıyoruz…

İlaplace fonksiyonuyla yazdığımız f fonksiyonunu çalıştırıyoruz.

Kodlamaların sıralaması aşağıdaki gibidir.

```
>> syms t s
>> F=(s-5)/(s*(s+2)^2);
>> ilaplace(F)

ans =

-5/4+1/4*exp(-2*t)*(5+14*t)


>> simplify(ans)

ans =

-5/4+7/2*t*exp(-2*t)+5/4*exp(-2*t)


>> pretty(ans)

                    - 5/4 + 7/2 t exp(-2 t) + 5/4 exp(-2 t)
>> 
```

Yukarıda görülen cevap aşağıdaki fonksiyonu karşılamaktadır:

$$f(t) = -1.25 + 3.5te^{-2t} + 1.25e^{-2t}$$

Alttaki komut satırı da aynı işlevi görmektedir:

```
>> ilaplace((s-5)/(s*(s+2)^2))
```

Örnek:

$$F(s) = \frac{10(s+2)}{s(s^2+4s+5)}$$

Fonksiyonunu Matlab ile ters Laplace işleyelim.

```
>> syms t s
>> F=10*(s+2)/(s*(s^2+4*s+5));
>> ilaplace(F)

ans =

4+2*(-2*cos(t)+sin(t))*exp(-2*t)


>> 
```

Yukarıdaki komut satırlarının karşılığı:

$$f(t) = [4 - 4e^{-2t}\cos(t) + 2e^{-2t}\sin(t)]u(t)$$

Sonuca göre trigonometrik ilişki kullanmak:

$$x\sin(\alpha) + y\cos(\alpha) = R\sin(\alpha + \beta)$$

Ve

$$x\cos(\alpha) - y\sin(\alpha) = R\cos(\alpha + \beta)$$

Ayrıca

$$R = \sqrt{x^2 + y^2}$$
$$\beta = \tan^{-1}(y / x)$$

Fonksiyonu ile birlikte

$$\mathrm{f(t)} = [4 + 4.47e^{-2t}\cos(t - 153.4^{o})]u(t)$$

Sonucumuzu yazabiliriz.

3.12 Eğri Uydurma (Curve-Fitting):

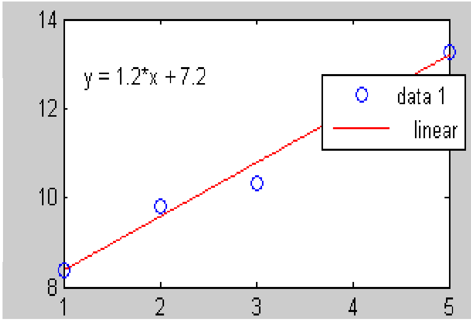
Eğri uydurma veri noktaları bir dizi için en uygun ve büyük olasılıkla diğer kısıtlamalar bir eğri bulgudur. Bu bölümde her iki interpolasyon bir giriştir (nerede kısıtlamalar bekleniyor için tam uyum) ve regresyon analizi. Her iki bazen extrapolation için kullanılır. Regresyon analizi bir yaklaşık uyum için veri noktaları ve eğri arasındaki farkı minimize ederek sağlar.

Örnek:

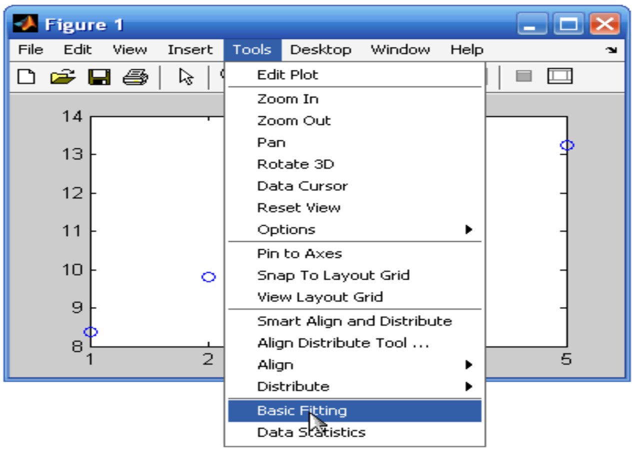
```
>> tdata=[1 2 3 4 5];
>> ydata=[8.3800 9.8200 10.3300 12.1400 13.2500];
```

Eğri uydurmada yukarıdaki verileri kullanacağım.

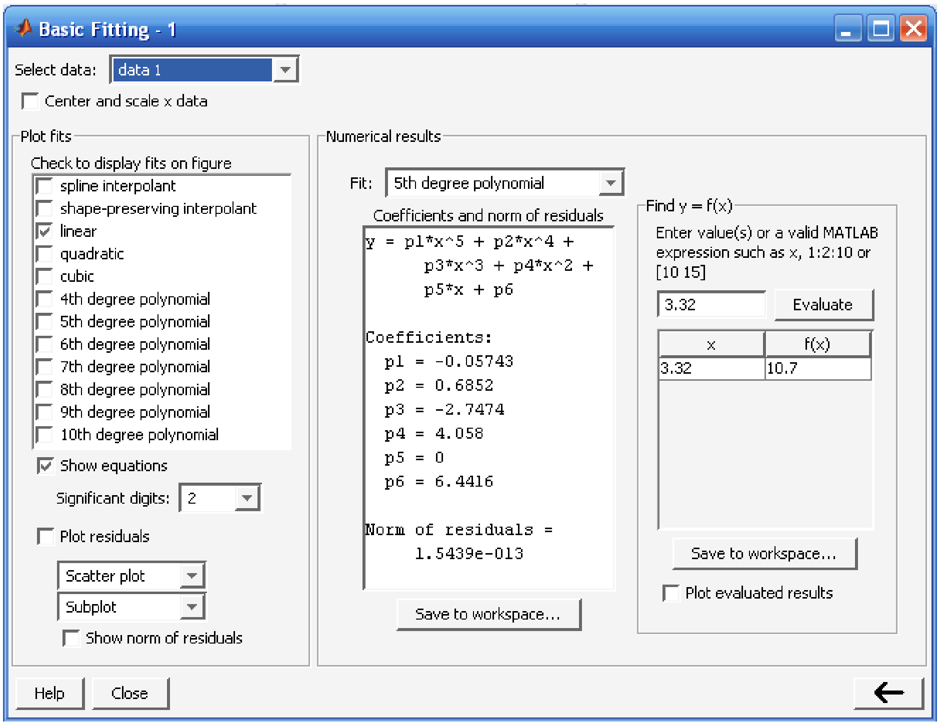
```
>> tdata=[1 2 3 4 5];
>> ydata=[8.3800 9.8200 10.3300 12.1400 13.2500];
>> plot(tdata,ydata,'o')
>>
```

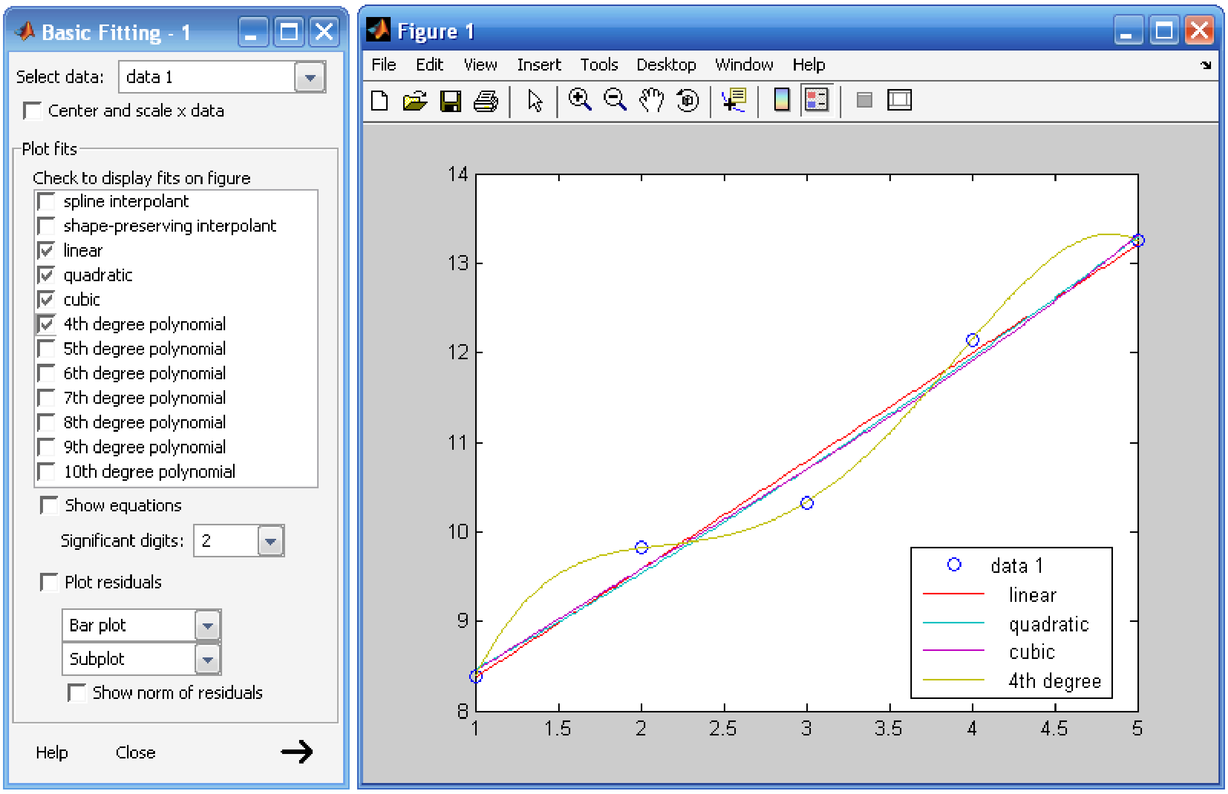


Daha sonra Figüre ekranındaki Tools menüsünden, Basic Fitting seçilir.

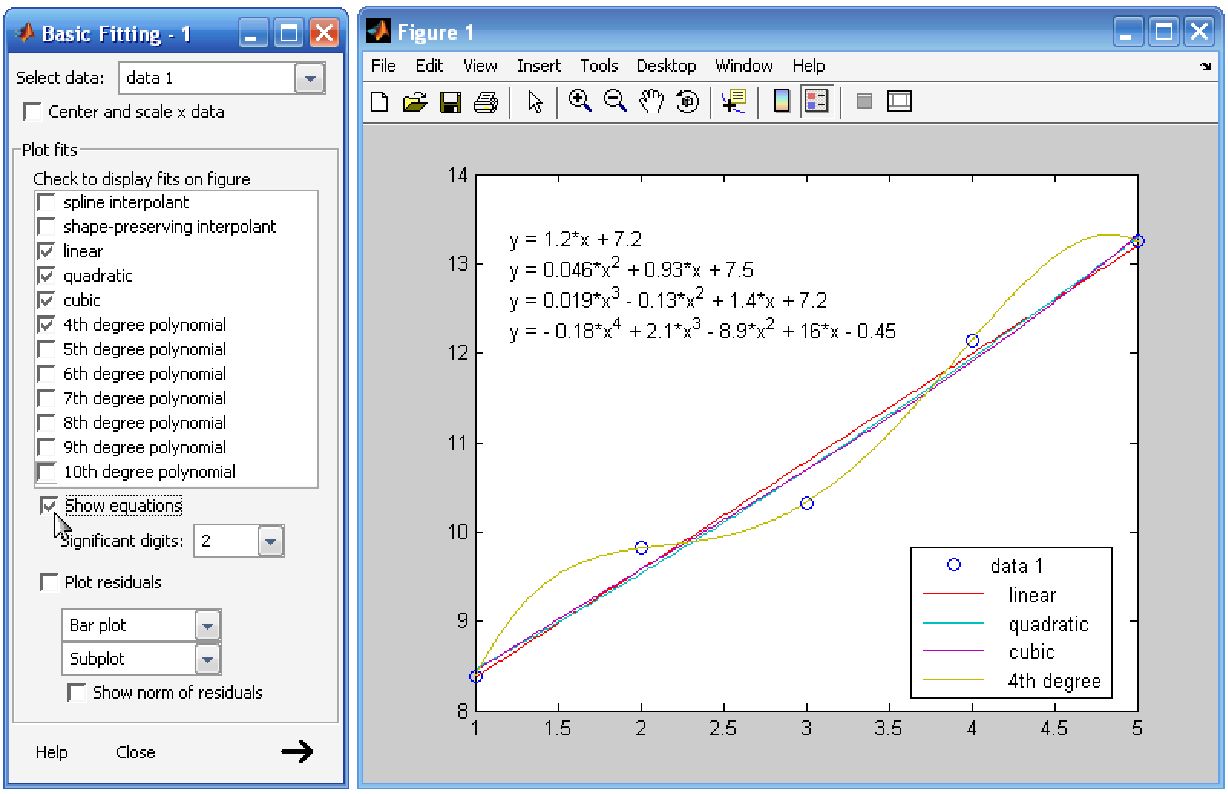


Uygulamamızı gerçekleştirdiğimiz için soldaki sütunda linear seçiminde ortadaki kırmızı hat çıkmakta…

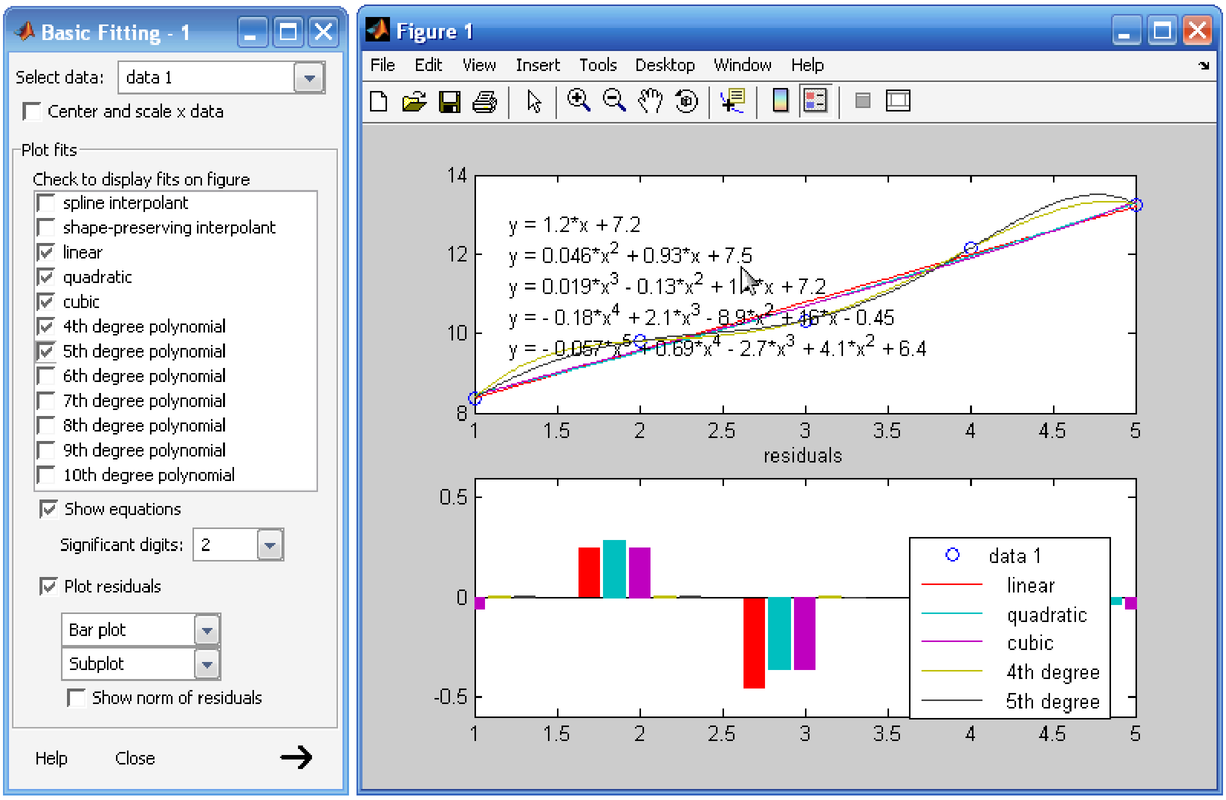




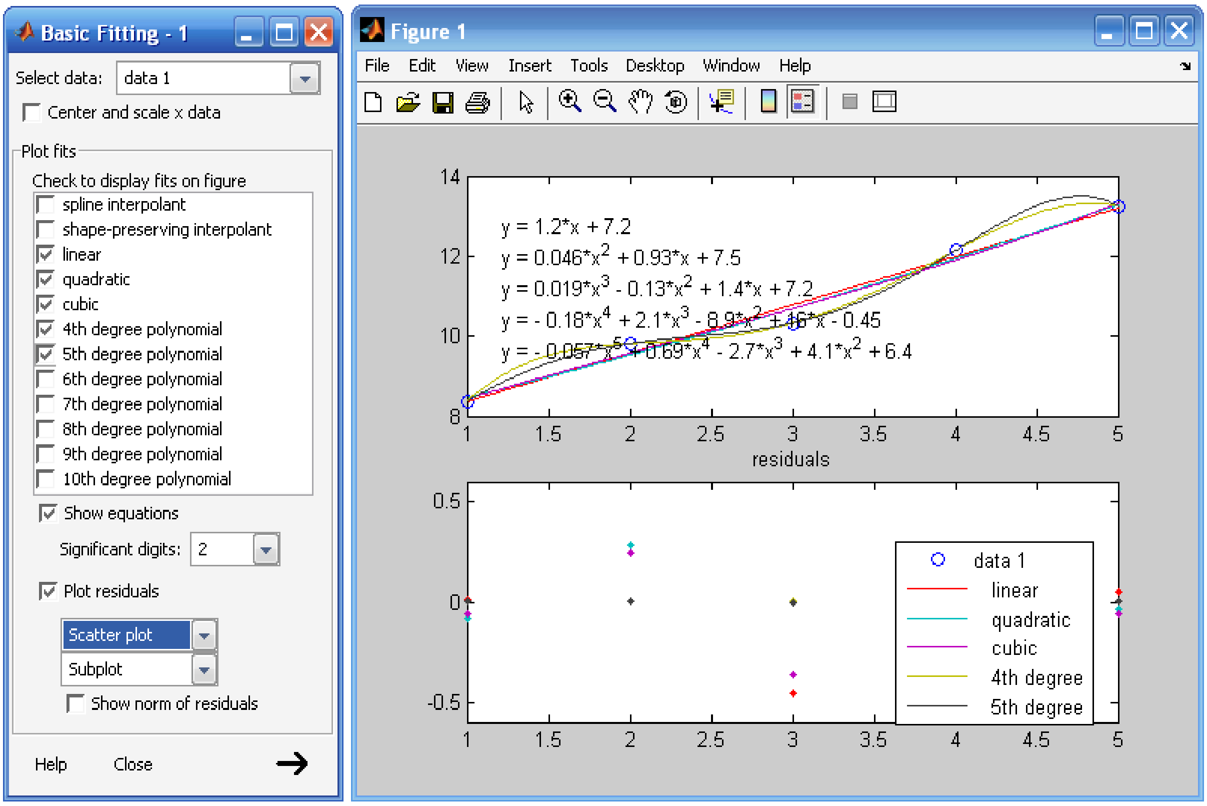
Soldaki seçimlere göre eğrileri Figure’de görmekteyiz. Kare halinde hangi rengin hangi eğriyi temsil ettiği görünmektedir.



Show Equations bölümünü tıkladığımızda Eğrilerin Polinom halleri işaretlenme sırasıyla görünmekte…



Plot Residuals Seçenekleriyle değerleri barlar halinde görüntüleyebiliyoruz.



3.13 Genel Örnekler ve Çözümleri:

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)

yp =

  -2.9112
   0.6613 + 1.1551i
   0.6613 - 1.1551i
  -0.5512 + 0.8874i
  -0.5512 - 0.8874i
  -0.8320
   0.9351 + 0.4894i
   0.9351 - 0.4894i
   0.3195
```

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:

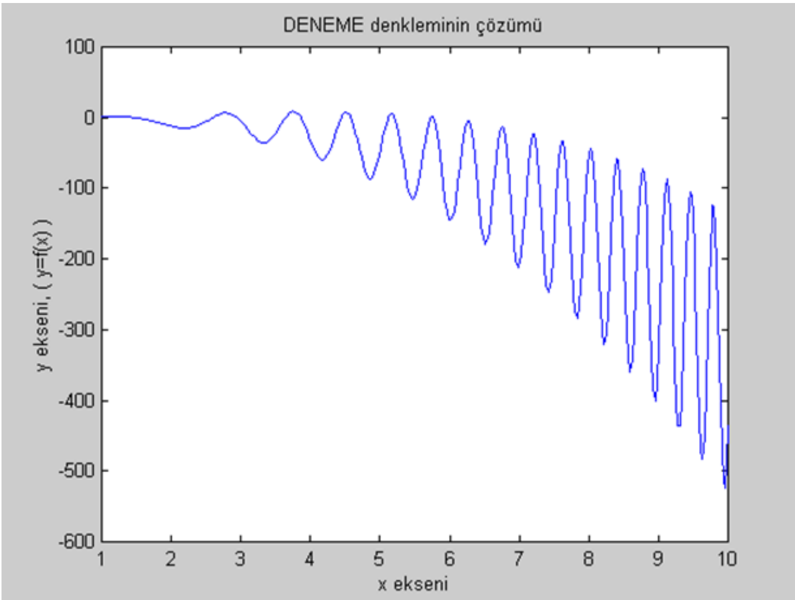
Problemin çözümü için, ilkönce aşağıdaki fonksiyon dosyası yazılır. Fonksiyonun yazılması için MATLAB'ın dahili editörü kullanılabilir. Bunun için MATLAB komut satırında edit yazılıp enter'a basılır. MATLAB'ta sayısal diferansiyel çözümleri için ode fonksiyonları kullanılır. ode fonksiyon dosyaları daha 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.

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

Deneme Denkleminin grafik çıktısı:



Örnek:

$(3x^5y^5-2y)dx + (5x^6y^4+x)dy=o$ 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.

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

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')
```

```
ans =

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


>> pretty(ans)

                                 3      5    y(t)
                                x   y(t)   + ---- + C1 = 0
                                               2
                                              x
```

Örnek:

(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')
```

```
ans =

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


>> pretty(ans)

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

Örnek:

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

Çözüm:

Çözüm için gerekli MATLAB komutu aşağıdaki gibidir.

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


cozum =

                  x^2+y(t)^2 = 0
 (x^2+y(t)^2)^(1/2)-y(t)-C1 = 0


>> pretty(cozum)

                        [          2        2                ]
                        [         x   + y(t)   = 0           ]
                        [                                    ]
                        [   2        2  1/2                  ]
                        [(x   + y(t) )      - y(t)  - C1 = 0]
```

Örnek:

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'))

                              -sin(x) + cos(x) x - C1
                            - -----------------------
                                         x
```

Örnek:

$\frac{dy}{dx} = e^{2x} + 3y$ 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'))

                              -exp(2 x) + exp(3 x) C1
```

Örnek:

X(t)=t ile Laplace transformasyonunu uygulayınız.

Çözüm:

Çözüm için gerekli MATLAB komutu aşağıdaki gibidir.

```
>> syms f t
>> f=t;
>> laplace(f)

ans =

1/s^2


>> pretty(ans)

                                        1
                                       ----
                                         2
                                        s
```

Örnek:

$\frac{dy}{dx} - \frac{y}{x} = -\frac{y^2}{x}$ Bernoulli denklemini çözelim.

Çözüm:

Çözüm için gerekli MATLAB komutu aşağıdaki gibidir.

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

                                                   x
                                                ------
                                                x + C1
```

Örnek:

Aşağıdaki fonksiyonu Laplace transformasyonla dönüştürünüz.

$$F(s) = \frac{24}{s(s+8)}$$

Çözüm:

Çözüm için gerekli MATLAB komutu aşağıdaki gibidir.

```
>> syms f s
>> f=24/(s*(s+8));
>> ilaplace(f)

ans =

3-3*exp(-8*t)


>> pretty(ans)

                                            3 - 3 exp(-8 t)
```

Örnek:

Aşağıdaki fonksiyona Laplace transformasyon uygulayınız.

$$Y(s) = \frac{4s^2 + 4s + 4}{s^2(s^2 + 3s + 2)}$$

Çözüm:

Çözüm için gerekli MATLAB komutu aşağıdaki gibidir.

```
>> n=[0 0 4 4 4];
>> d=[1 3 2 0 0];
>> [r,p]=residue(n,d)

r =

    -3
     4
    -1
     2


p =

    -2
    -1
     0
     0
```

Fonksiyon cevap görünümü ise:

$$\frac{-3}{s+2}+\frac{4}{s+1}-\frac{1}{s}+\frac{2}{s^2}$$

KAYNAKÇA

- "MATLAB Help"

- Gündoğdu, Ömer ve Kopmaz, Osman ve Ceviz, M.Akif, "Mühendislik ve Fen Uygulamalarıyla MATLAB", Paradigma Akademi, Bursa-2003

- Yüksel, İbrahim, "MATLAB ile Mühendislik Sistemlerinin Analizi ve Çözümü, Genişletilmiş II. Baskı", Vipaş A.Ş., Bursa-2000

- C. Kuo, Benjamin, Çeviren ve uyarlayan: Bir, Atilla, "Otomatik Kontrol Sistemleri, Yedinci Baskı", Literatür, 1999

- www.mathworks.com internet adresi, 01..25/04/2003

- http://www.mame.mu.oz.au/control/mcg/ctrl301/matlab/ctm/index.html

- http://www-personal.engin.umich.edu/~tilbury/tutorials/me461.html

- http://teal.gmu.edu/~gbeale/examples_421.html

- http://www.math.mtu.edu/math/Content.html

- F. Coughlin, Robert ve F. Driscoll, Frederick, Operational Amplifiers & Linear Integrated Circuits, Fourth Edition, Prentice Hall, Englewood Cliffs, NJ

- Matlab Kontrol Sistemleri Kullanım Kılavuzu İstanbul Üniversitesi Bilgisayar Mühendisliği Bölümü 2002-2003 Güz Yarıyılı, Ekim 2002