MATLAB’e Giriş

2016

MATLAB’e Giriş

MATLAB Nedir?

MATLAB,nümerik hesaplama,verileri ve/veya hesaplamaları görselleştirme,ve programlama için bir ortam ve yüksek seviyeli programlama dilidir.MATLAB'i kullanarak;data analizi yapılabilir,algoritma geliştirilebilir,modelleme ve uygulamalar oluşturulabilir.Ayrıca MATLAB uygulama alanı olarak,aşağıdaki paketlere (tool box) sahiptir :

signal processing and communications,
image and video processing,
control systems,
test and measurement,
computational finance,
computational biology

MATLAB tabanlı uygulamalar ile C,Java,.NET,ve Microsoft Excel gibi uygulamalar birlikte kullanılabilir.Örneğin,data analizi için Excel'den veri ihraç edip,uygulamanızı yapabilirsiniz.

MATLAB endüstride nasıl kullanıldı?

• Motorsport Teams Improve Vehicle Performance with MathWorks Tools (http://www.mathworks.com/products/simmechanics/userstories.html?file=11197)
• Bell Helicopter Develops the First Civilian Tiltrotor (http://www.mathworks.com/company/newsletters/news_notes/oct06/bellhelicopter.html)
• Greenhouse Designed with MATLAB and Simulink Revolutionizes Agriculture in Arid Coastal Regions (http://www.mathworks.com/company/user_stories/userstory2347.html?by=industry)
• Thames Water Aims to Reduce Leaks by More Than 25% Using a MATLAB-Based Leak-Location System (http://www.mathworks.com/company/user_stories/userstory2354.html?by=industry)
• Samsung UK Develops 4G Wireless Systems with Simulink (http://www.mathworks.com/company/user_stories/userstory10725.html?by=industry)
• Cambridge Consultants Develops WiMAX Test Bench for Aspex Semiconductor with MATLAB (http://www.mathworks.com/company/user_stories/userstory10996.html?by=industry)
• Halliburton Makes Oil Exploration Safer Using MATLAB and Neural Networks (http://www.mathworks.com/industrial-automation-machinery/userstories.html?file=2355&title=Halliburton%20Makes%20Oil%20Exploration%20Safer%20Using%20MATLAB%20and%20Neural%20Networks)

MATLAB’e Giriş 1

MATLAB’e Giriş 1

MATLAB ayarları

Transpoz

Matrisler ve Diziler (Matrices and Arrays)

Değişkenlerin boyutu

Matris ve Dizilerle İşlemler

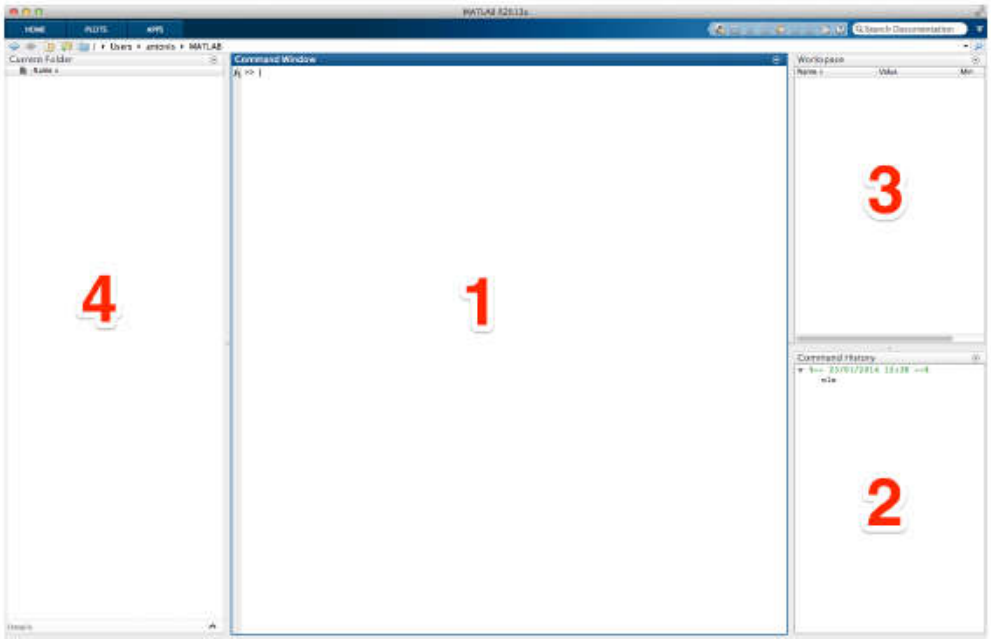
Matrislerde Çarpma

EKSTRALAR

MATLAB (**Mat**rix **Lab**oratory)

MATLAB'e Giriş 1

MATLAB nedir,ne yapar,kullanmalı mıyım gibi soruları cevaplandırdıktan sonra ,aşama MATLAB programını öğrenme aşamasına gelir.İlk böülmde MATLAB açıldığında önümüze gelen sayfa tanıtılacak ,matrisler hakkında genel bilgi ,basit olarak genel işlemlerden bahsedilecektir.



MATLAB'i açtığımızda 1,2,3 ve 4 numaralı yerlerin ne olduğu yazmaktadır.

1) Command Window
2) Command History
3) Workspace
4) Current Directory

İlk olarak >> ile gösterilen yerlere matematiksel ifadeler girersek MATLAB'i kolaylıkla bir hesap makinesi olarak kullanabiliriz.Ayrıca MATLAB 'de vektörel olarak işlem yapılabilmesi birçok açıdan işimizi kolaylaştırmaktadır.

Aşağıdaki işlemlede

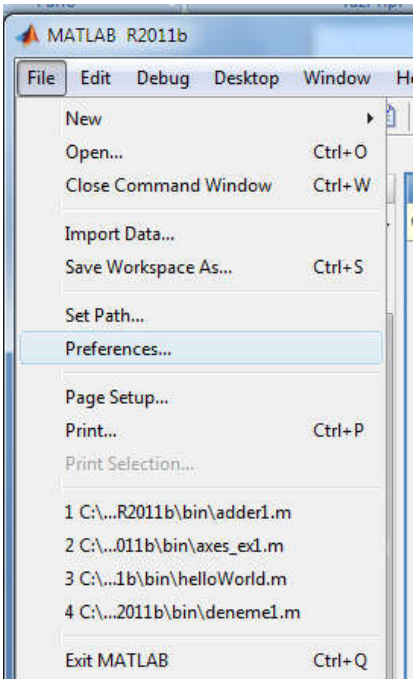
^ ➔ üs alma

exp ➔ exponential ($e^{(-2/3)}$)

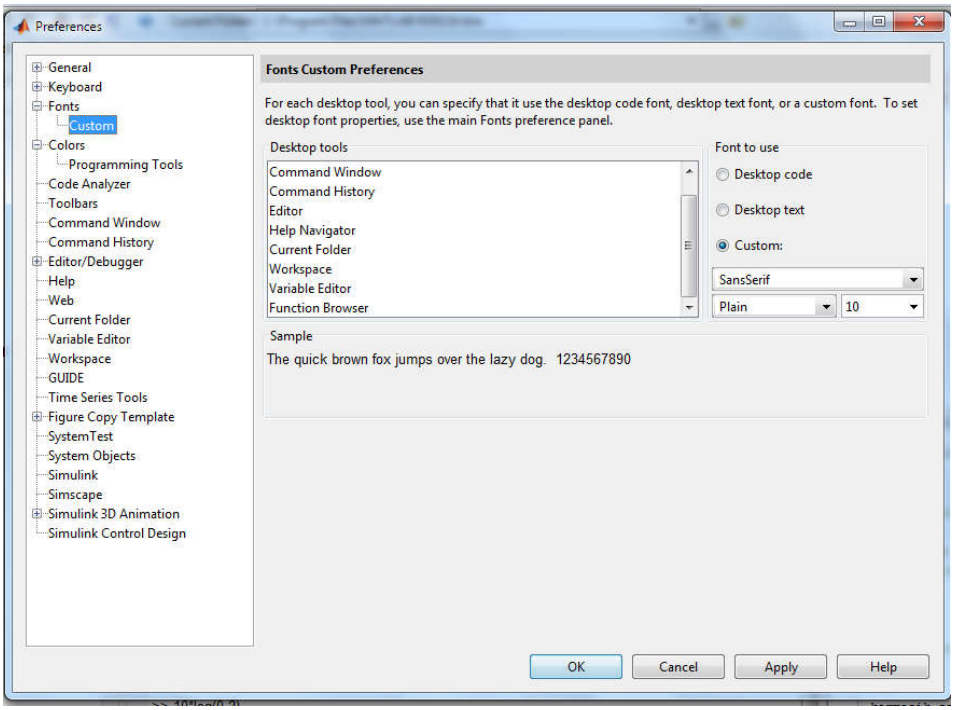
atan ➔ $\tan^{-1}$

```
>> 3 + 4

ans =

    7

>> 2^2

ans =

    4

>> karmasik_sayi = (3 + 4j) * (1 + 2j)

karmasik_sayi =

  -5.0000 +10.0000i

>> sonuc = 2*pi + exp(-2/3)

sonuc =

   6.7966

>> atan(5/5)

ans =

   0.7854

>> 10*log(0.2)

ans =

  -16.0944
```

MATLAB ayarları



Edit --> File --> Preferences seçeneğine tıklıyoruz.



Önünüze şekildeki gibi açılan ekrandan "Fonts" seçeneği ile yazı boyutunu ayarlarken,"Colors" seçeneği ile de yazınıza renk ayarı yapabilirsiniz.

ÖRNEK1 :

a = $\cos^2(30) + \sin^2(30)$

b = $2.5x10^{23}$

c = 2 + 3i (i ➔ karmaşık sayıdaki imajiner kısım)

d = $e^{j2\pi/3}$ + c - b (j karmaşık sayıda imajiner kısım burada **exp** ve **pi** yi kullanınız)

Değişkenlerin boyutu

>>whos

komutunu Command Window'a girersek değişkenler hakkında bilgi edinebiliriz.

```
>> whos
  Name           Size          Bytes  Class     Attributes

  ans            1x1               8  double
  karmasik_sayi  1x1              16  double    complex
  sonuc          1x1               8  double

fx >>
```

Size vektörün boyutunu gösterir ancak şu ana kadar sadece tek boyutlu değişkenler kullandık.

Matrisler ve Diziler (Matrices and Arrays)

Deteylı bilgi daha sonra incelenecek ancak genel olarak MATLAB'i kullanabilmemiz için az da olsa matris işlemlerini yapabiliyor olmamız gereklidir.

Şimdi vektörlere ve matrislere genel bir giriş yapabiliriz.

```
>> row_vector = [1 2 3 3 2 1]

row_vector =

    1    2    3    3    2    1
```

Eğer görseldeki gibi köşeli parantezin içine değerler arasında boşluk bırakarak veya aralarına virgül(,) koyarak değerleri girersek satır vektörü elde ederiz.

a = [1 2 3] ile a =[1,2,3] aynı şeylerdir.

```
>> column_vector = [1;2;3;9;8;7]

column_vector =

    1
    2
    3
    9
    8
    7
```

b = [1;2;3] gibi değerler arası noktalı virgül ise bir satır aşağıya geçmemizi sağlar.

Transpoz

Bu işlem (') şekildeki gibi girilir.

```
>> a_tarnspoz = [1 2 3 3 2 1]'

a_tarnspoz =

     1
     2
     3
     3
     2
     1

>> a_transpoz = row_vector'

a_transpoz =

     1
     2
     3
     3
     2
     1

fx >>
```

Dikkatli inceleyecek olursak değişkenlere de bu işlem uygulanabilir.Şimdi de aşağıdaki matrisi Command Window'a girelim.Daha önce dediğimiz üzere (;) noktalı virgül kullanımı bir aşağıki satıra geçiriyor yani ,Diğer satırı oluşturuyor.

```
>> m1 = [1 2 3;4 5 6;7 8 9]

m1 =

     1     2     3
     4     5     6
     7     8     9

fx >>
```

Matris oluşturmak için, bir diğer yöntem de fonksiyonları kullanmaktır.

*** ones,zeros,rand ***

>> Z = zeros(5,1) ➔ 5x1 lik bir 0 (sıfır) vektörü üretir.

ones,ve *rand* için de kullanım yöntemi aynıdır.

Matris ve Dizilerle İşlemler

```
>> m1 = [1 2 3;4 5 6;7 8 9]

m1 =

   1   2   3
   4   5   6
   7   8   9

fx >>
```

Şimdi m1 matrisi üzerinde işlemler yapalım.

>> m1 + 10 Tüm elemanlara 10 ekler.

>> sin(m1) Tüm elemanların sinüsünü alır.

>> m1' Transpoze

>> p = m1 * inv(m1) inv(m1) ,m1 matrisinin tersini alır ve bunu m1 ile çarparsak p matrisi birim matris olur.

```
format long
p = a*inv(a)

p =

   1.000000000000000                   0  -0.000000000000000
                   0   1.000000000000000                   0
                   0                   0   0.999999999999998
```

>> format long bu komut kullanıldığında uzun formatta yazdırır.

>> help format komutunu giriniz ve bu arada **MATLAB'da en önemli komut** *help* komutodur.

Matrislerde Çarpma

Örneğin iki tane matris var ve bu iki matrisi çarpacağız.

Eğer biz bu matrisi vektörel olarak çarpacaksak * işlemini yaparız.

Ama,birinci_matris(i,j) ile ikinci_matris(i,j) yi çarpacaksak(element-wise) .* işlemini yaparız.

```
>> birinci_matris = [2 3 ; 1 7]

birinci_matris =

   2   3
   1   7

>> ikinci_matris = [0 3 ; 1 1]

ikinci_matris =

   0   3
   1   1
```

```
>> carpim1 = birinci_matris * ikinci_matris

carpim1 =

   3   9
   7  10

>> carpim2 = birinci_matris .* ikinci_matris

carpim2 =

   0   9
   1   7

>>
```

EKSTRALAR

```
A = [1 3 5];
max(A)

ans =

     5
```

>>max(A) ile A matrisinin maksimum değerini bulabiliriz.

Eğer Command Window'da girdiğimiz değerlerin sonuna (;) noktalı virgül koyarsak ,MATLAB onu oluşturur ancak,ekrana basmaz.Bu çok kullanışlıdır.Keşke daha önce yazsaydım :)))) Ya da zaten bunu biliyordunuz.

Aşağıdaki gibi iki matris tanımlayıp bunların maximumlarının bulunması için ;

>>A = [1 2 3];

>>B = [-2 10 1];

>>max(A,B)

Hepsini teker teker kıyaslar ,A ve B matrisi ile aynı boyutta yeni bir matris oluşturur.

```
>> A = [1 2 3];
>> B = [-2 10 1];
>> max(A,B)

ans =

   1   10    3
```

Eğer matrislerde 1'den 10'a kadar 0.5'er 0.5'er artan vektörüretmek istiyorsanız!!!

```
>> a = 1:0.5:10

a =

 Columns 1 through 8

   1.0000    1.5000    2.0000    2.5000    3.0000    3.5000    4.0000    4.5000

 Columns 9 through 16

   5.0000    5.5000    6.0000    6.5000    7.0000    7.5000    8.0000    8.5000

 Columns 17 through 19

   9.0000    9.5000   10.0000
```

A = başlangıç : artma veya azalma(azalma ise başına eksi(-)) : sonlanma noktası

Veya

```
>> a = linspace(1,10,19)

a =

 Columns 1 through 8

   1.0000    1.5000    2.0000    2.5000    3.0000    3.5000    4.0000    4.5000

 Columns 9 through 16

   5.0000    5.5000    6.0000    6.5000    7.0000    7.5000    8.0000    8.5000

 Columns 17 through 19

   9.0000    9.5000   10.0000
```

A = linspace(1,10,19) ➔ 1'den 10'a kadar 19 tane(yani 0.5 artarak)

disp ile aşağıdaki gibi string yazdırabilirsiniz.

```
disp('hello world')
hello world
```

***help çok önemli bir fonksiyondur.

>> help length

>> help clear

>> help clc

>>help plot

PROBLEM 1 : Aşağıdaki vektörleri oluşturunuz.

a) aVec = [3.14 15 9 26]

b) bVec = $\begin{pmatrix} 2.71 \\ 8 \\ 28 \\ 182 \end{pmatrix}$

c) cVec = [5 4.8 … -4.8 5] (numaralar -5 ile 5 arasında,ve 0.2 artıyor)

d) dVec = [10^0 $10^{0.001}$ … $10^{0.99}$ 10^1] (1'den 10'a kadar logaritmik artış,bunun için **logspace** kullan)

PROBLEM 2 : Aşağıdaki vektörleri oluşturunuz.İşlemleri yaparken (***.**** , ***./*** , ***.^)*** gibi işlemleri (elementwise-operators) kullanınız.

a. $xVec = \frac{1}{\sqrt{2\pi 2.5^2}} e^{-cVec^2 / \left(2 \bullet 2.5^2\right)}$

b. $yVec = \sqrt{\left(aVec^T\right)^2 + bVec^2}$, aVec^T ,aVec vektörünün transpozudur.

c. $zVec = \log_{10}\left(1 / dVec\right)$ Log_{10} için **log10** kullan

Problemleri çözerken **help** fonksiyonu ile kullanılması gereken komutları nasıl kullanmanız gerektiğini öğrenebilirsiniz. (örnek: help log10)

MATLAB’e Giriş 2

2D-Plot

Grafik Çiziminde Ek Bilgiler

Aynı Figürde Çoklu Grafik (Multiple Plots)

Programlama ve Script Dosyaları(Programming and Scripts)

PROGRAMLAMA

İf-else yapısı

FOR DÖNGÜSÜ

while Döngüsü

MATLAB'a Giriş 2

İkinci bölümde, basit olarak 2D grafik çizimi (2D plotting) ,Script dosyası ve basit olarak programlamaya(programming) giriş yapılacaktır.Daha sonraki derste ise bu işlemleri daha da detaylandıracağız.

Bu arada internet üzerinden "MATLAB Primer pdf" şeklinde arama yaparsanız MathWorks'un kaynağına ulaşabilirsiniz.

2D-Plot

>> help plot

İlk olarak bu komutu Command Window'a girerek nasıl kullanıldığına bakıyoruz.

Şimdi, sinüs grafiği çizdirelim.İşlemlerimizi yaparken küçükte olsa algoritmalar kurmak daha düzgün,amaca ulaşabilen ve daha karmaşık kodları daha kolay yazmak için çok önemlidir.

1.Amaç : sinüs grafiği çizdirmek

2.Girdiler(Inputs) : x (0 ile 2π arasında olsun)

3.Matematiksel denklem : y = sin(x)

4.Çıktı(Output) : y

```
>> x = 0:pi/100:2*pi;
```
Burada x, 0'dan başlayıp , pi/100'er pi/100'er artıp, 2 x pi 'de bitiyor.

```
>> y = sin(x)
```

```
>> plot(x,y)
```
x(yatay) ekseni x,y(düşey) ekseni y olan grafik çiziyor.

```
>>xlabel('x')
>>ylabel('sin(x)')
>>title('sinüs fonksiyonu grafiği')
```
label ve başlık ekleyebilirsiniz…

*** plot fonksiyonu ilk olarak x vektörünün her bir değeri için y vektörünü oluşturur ve sırasıyla her bir noktayı birleştirir.Yani x ile y vektörü aynı boyuttadırlar. length(x),length(y) komutlarını kullanarak boyutlarını bulabilirsiniz.

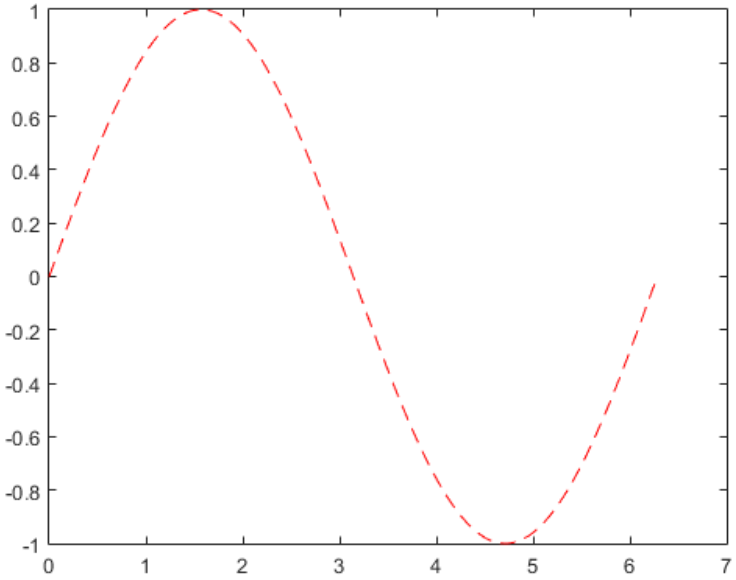
*** Ayrıca x vektörünün artışını(increment) pi/100 değil de, pi/2 'şer arttırırsanız x vektörü 5 elemana sahip olacaktır.Bu noktaları birleştirdiğinde daha sağlıksız sonuç alırız.Yani data fazlaysa,çizim iyileşir ancak bellekte daha fazla yer depolar.

Grafik Çiziminde Ek Bilgiler

```
>>plot(x,y,'r--')
```

Burada,r → red (kırmızı) ve -- → grafiğin çizgi çizgi basılmasını sağlar.

```
>>plot(x,y,'g+','LineWidth',2)
```
burada "LineWidth" çizgi kalınlığını 2 olarak ayarlar.



Örnek 1 : $y = \cos(t)e^{-|t|}$

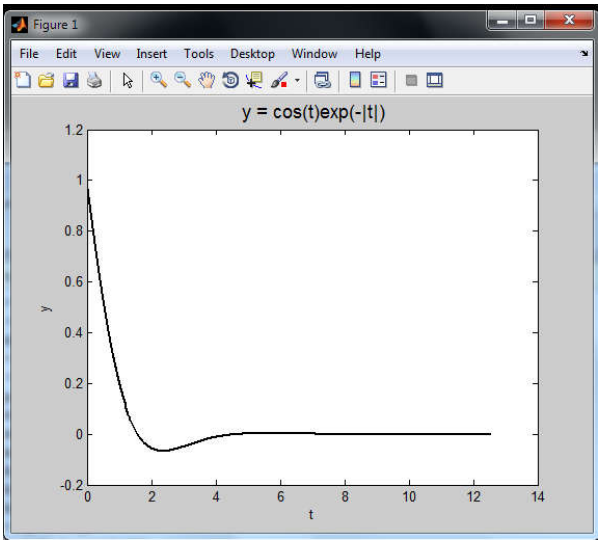
Grafiğini çizdiriniz.

t 'nin aralığı 0'dan +4π'ye olabilir.

Grafik siyah renk olsun.

Çizgi kalınlığı 2

```
>> t = 0:pi/100:4*pi;
>> y = cos(t) .* exp(-(abs(t)));
>> figure (1)
>> plot(t,y,'k','LineWidth',2)
>> xlabel('t')
>> ylabel('y')
>> title(' y = cos(t)exp(-|t|)','fontSize',13)
>>
```



help komutu kullanarak

semilogx, semilogy, loglog, grid, hold,axis,legend,text,abs fonksiyonlarına bakınız.

Bunlar,ileride detaylı incelenecek…

ÖRNEK 2 : w = 4000 rad/s ve t'nin aralığı -2π'den +2π'ye olacak şekilde

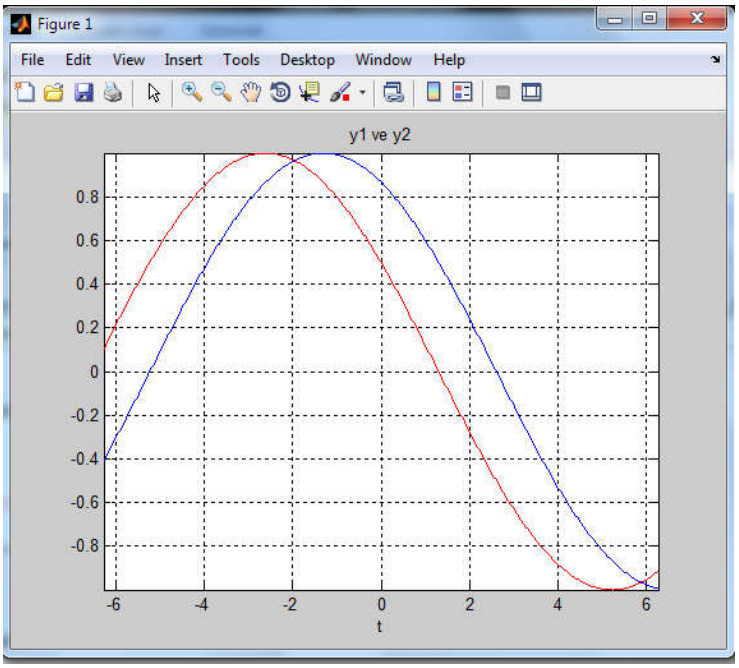
y1 = cos(wt + π/6)

y2 = cos(wt + π/3) grafiklerini çizdiriniz.Bunu yapabilmek için w,t,y1 ve y2 ifadelerini tanımladıktan sonra

>>figure (1) Grafiğin çizdirileceği figürü açınız.

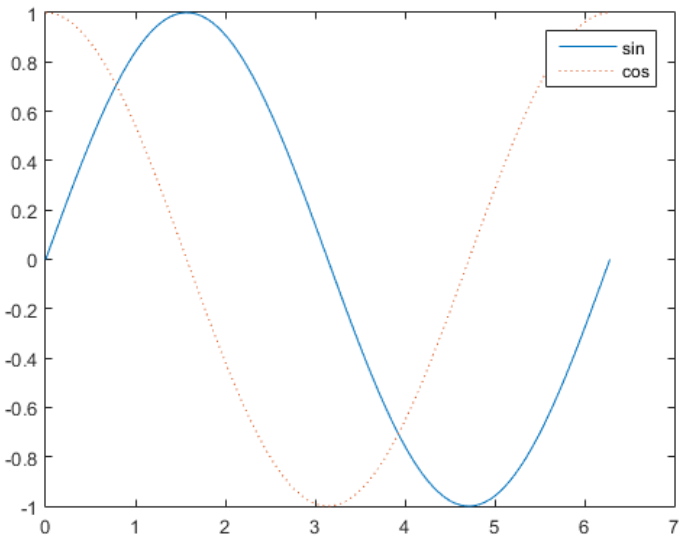
>>hold on Sonra bunu yazınız,bu komut aynı figürde çoklu grafik çizimine ve **hold off** yazana kadar her komutu açtığınız figür için yapmanızı sağlar.Bunun için label ve title tanımlamalarınızı da bunu kullanarak yapabilirsiniz.

>>help grid ve **>>help axis** komutlarını girmeyi unutmayınız.



Aynı Figürde Çoklu Grafik (Multiple Plots)

```
x = 0:pi/100:2*pi;
y = sin(x);
plot(x,y)

hold on
y2 = cos(x);
plot(x,y2,':')
legend('sin','cos')
```



Fazladan grafik eklemek için **hold** fonksiyonun kullanırız.Ve **hold off** kullanana kadar istediğiniz kadar çizdirebilirsiniz.

3D plot ve subplot daha sonraincelenecektir.Şimdi çok fazla detaya girmedik.

Programlama ve Script Dosyaları(Programming and Scripts)

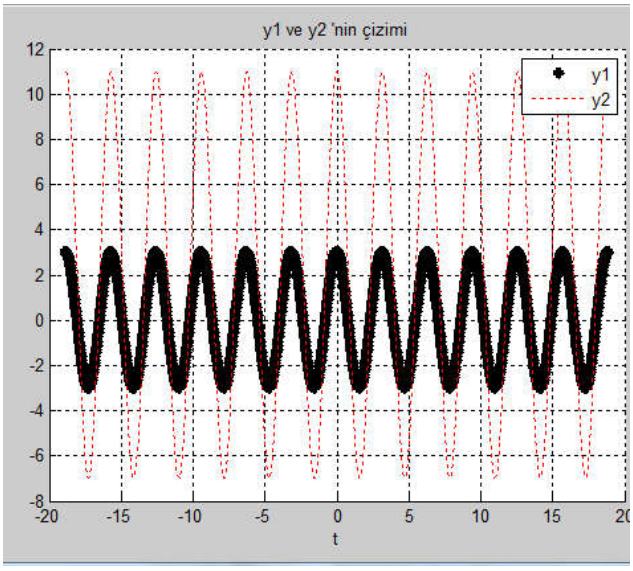
Command Window'a **edit plotting** girerseniz,plotting.m olarak kayıtlı script dosyası açılacaktır.

Daha önceden Command Window'da yaptığımız işlemleri daha düzgün yapacağız.Eğer işlemlerin sonuna noktalı virgül(;) koyarsanız sonuç Command Window'da basılmaz,ancak hiçbir şey koymazsanız ekrana basacaktır.

>>edit plotting_ex1 YES,diye yanıtlıyoruz :)

Aşağıdaki kodları inceleyelim :

```
% bu dosya plotting_ex.m olarak kaydedildi
% ilk olarak t aralığı tanımlayacağız
% sonra y1 = 3*cos(2.*t)
% y2 = 3*y1 + 2

t = -6*pi:pi/100:6*pi;
y1 = 3*cos(2.*t);
y2 = 3*y1 + 2;

%plot çizimi
figure (1)                          %figür açtık
hold on
plot(t,y1,'k+','LineWidth',3)
plot(t,y2,'r:')
xlabel('t')
title('y1 ve y2 "nin çizimi')       %başlık
legend('y1','y2')                   %önce ilk çizilen plot,sonra da ikinci çizilen plot'u
                                    %kutu içinde hangi plotun nasıl
                                    %çizildiğini gösterir

grid on                             %figürü karelere böldük
hold off                            % işlemi tamamladık

%şimdi de y1 ve y2 'nin maximum değerlerini bulup yazdıralım
y1_max = max(y1);
y2_max = max(y2);

disp('y1 için en büyük değer ')
disp(num2str(y1_max))       %burada num2str(y1_max) nümerik ifadeyi string'e çevirdi

disp('y2 için en büyük değer ')
disp(num2str(y2_max))
```



1. % operatörünü kullanarak yorum yazabilirsiniz
2. Command Window'da tek tek komutları girmek yerine çok daha kullanışlıdır.
3. num2str string'e çevirdi.
4. 2 plot'u da aynı figürde çizdiğimizden dolayı x ekseni ortak kullandık,yoksa matematiksel yorumunda zorluk çekebilirdik.
5. Son olarak da ,help fonksiyonu nasıl önemliyse ,scriptler'de de yorum yazmak (%....) çok önemlidir.

ÖRNEK 3 : Aşağıdaki aşamaları uygula ve yazacağın script dosyasına kendi yorumlarını da ekle.

a) Yeni script dosyası aç ve **throwBall.m** olarak kaydet.
b) İlk olarak dosyanın üst kısımlarına bazı sabitler(constants) tanımlaman gerekiyor.
   i. Topun atıldığı andaki, yani ilk yüksekliği 1.5 m
   ii. Yer çekim ivmesi 9.8 m/s^2
   iii. Topun atılma hızı 4 m/s
   iv. Topun fırlatılma açısı yere göre 45 derece'dir.

c) Şimdi zaman(time vector) oluşturalım ve,0 ile 1 arasında 1000 elemanlı olsun.

d)**x** mesafe ve **y** ise yükseklik,buna göre aşağıdaki denklemleri zamanın fonksiyonu olarak tanımla.Diğer parametreler ise,(ilk yükseklik **h**,yerçekimi ivmesi **g**,topun ilk hızı **v**,hız vektörünün açısı **theta**) ve denklemleri çöz.

i. $x(t) = v\cos\left(\theta\frac{\pi}{180}\right)t$.

ii. $y(t) = h + v\sin\left(\theta\frac{\pi}{180}\right)t - \frac{1}{2}gt^2$

NOT : theta'yı (θ) ,π/180 ile çarparsak dereceden radyana çevirmiş oluruz.

e) i. Yüksekliğin ilk defa negatif olduğu indeksi bulunuz (**find** kullanınız.)

ii. topun yere çarptığı andaki x mesafesini bulunuz ve indeksini bulunuz

iii. Şunu yazdırınız : "top x mesafe sonra yere çarptı." ii'de bulduğumuz değer x'tir.

f) Şimdi sıra,plot olarak çizdirmekte

i. yeni figür aç (**figure**)

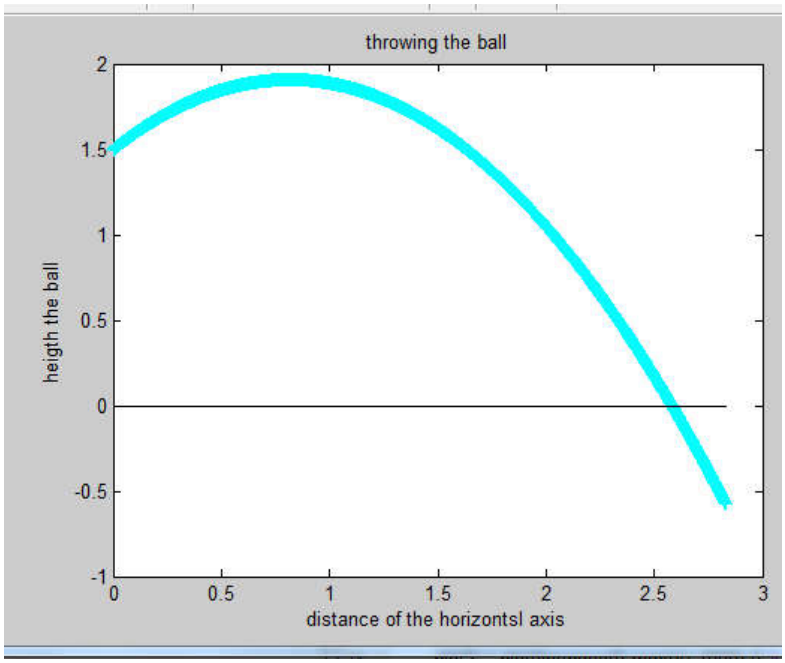
ii. mesafe,yükseklik grafiği çizdir (**plot**)

iii. label ve başlık ekle.

iv. başka bir grafik çizimi için(**hold**)

v. yer seviyesini çizdir.Bunun için yatay bir çizgi bas ve 0'dan x'in maximum değerine kadar değerleri ata (**max**)

g) Kodu çalıştır,ve yaklaşık grafik şunun gibi görünecek,Command Window'a yazdırılan değer yaklaşık 2.5821 olacak.



Bu problem MIT OpenCourseWare 6.094 Introduction to MATLAB® January Homework 1'den alınmıştır.

Şimdi bazı fonksiyonların kullanımına bakalım :

input() komutu : Kullanıcıdan klavye aracılığıyla programcı tarafından girilmesi istenen değişken istenir ve ilgili değişkene atanır.

```
>> yas = input('Yasinizi Giriniz : ')
Yasinizi Giriniz : 21
yas =
21
```

fprintf() komutu : Bir açıklama ifadesiyle birlikte bir veya birden fazla değerin görüntülenebilmesini sağlar.

```
>> a=231565465;
>> fprintf('Hesap = %d ',a)
Hesap = 231565465.000000
```

PROGRAMLAMA

İf-else yapısı

İf-elseif-else yapısı seçim mekanizması için kullanılır. Bir mantıksal ifadeyi kontrol ederek bunun sonucuna göre mümkün seçeneklerden birini icra edebilen bir komuttur.

-- Mantik ve Iliski Operatörleri :

==	Esittir	&	and	Ve
~=	Esit degil	&	and	Ve
<	Küçük	~	not	Degil
<=	Küçük esit			
>	Büyük			
>=	Büyük esit			

ÖRNEK 4 : Bir Script dosyasında,ilk olarak kullanıcıdan 0-100 arasında bir integer sayı isteyin.Eğer girdiği

Sayı 50'den büyük ise ,Sayı 50'den büyük

Sayı 50 ise,sayı 50

Sayı 50'den küçük ise ,50'den küçük sayı girdiniz. Yazdıran bir program yazınız.

FOR DÖNGÜSÜ

Bir çok uygulamada belirli işlemlerin tekrar tekrar gerçekleştirilmesi gerekir. Programlamada bu işlemler grubunu çok sayıda tekrar etmek imkanı sağlayan yapılara **ÇEVRİM, DÖNGÜ** veya **LOOP** denir.

```
for döngü değişkeni = başlangıç : bitiş
        komutlar
        ....
end
```

Örnek 5 : 1'den 100'ye kadar sayıların toplamını bulan program

```
toplam = 0 ;
 for x = 1:100
toplam = toplam + x ;
end toplam
```

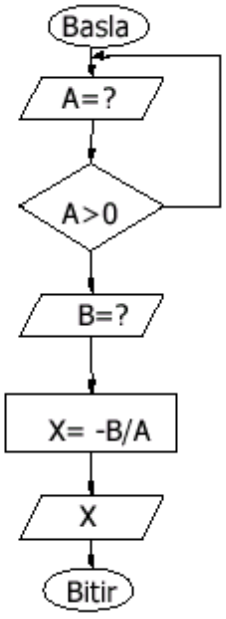
while Döngüsü

while'da belirtilen durum gerçekleşmeyinceye kadar içindeki komutları uygular.Yani,while ifadesi 1 (doğru)ise,while döngüsünün içine girer,0(yanlış) olunca döngü sonlanır.

```
while durum
        1.ifade
        2.ifade

        n.ifade
end
```

Örnek 6 : Ax+b=0 şeklinde verilen 1. derece denklemin çözümünü veren programı aşağıda verilen akış diyagramından yararlanarak MATLAB'de programlayınız.

```
A=input('A katsayısını giriniz..: ');
while A=0
A=input('A katsayısını giriniz..: ');
end
B=input('B katsayısını giriniz..: ');
x=-B/A;
fprintf('%d \n',x)
```



MATLAB Dilinin Temelleri

MATRİSLER

Matris Oluşturmak

sum, transpose*, ve *diag

Bazı Fonksiyonlar ile Matris Üretmek

MATLAB'da Sayılar

MATRİS İŞLEMLERİ(OPERATORLERİ)

***format* fonksiyonu**

İndeksleme (Indexing)

Colon (:) Operatörü

SATIR veya SÜTUN Silme :

Polinomlar

Polinomun kökleri

Kökleri bilinen bir polinom

Polinomun belli bir noktada değeri

MATLAB Dilinin Temelleri

Bu bölümden itibaren, yüzeysel değinmeleri bırakıp detaya ineceğiz.

Bu derste ise, matrisler, arrayler, değişkenler ve Command Window kullanımındaki detaylara gireceğiz. Matrisler ve vektörler denklem çözme,plot çizdirme işlemlerimizi daha kolay yapabilmemiz için önemi fazladır.Ve sonunda kısaca polinomları tanımlayacağız.

MATRİSLER

Bizim MATLAB'de girdiğimiz skalar değişkenler aslında 1x1 matrislerdir.

```
>> degisken = 13

degisken =

   13

>> whos
  Name        Size            Bytes  Class     Attributes

  degisken    1x1                 8  double

>>
```

Buradan da size'dan anlaşılıyor.

>>degisken = 13;

>>size(degisken)

Matris Oluşturmak

- Satır (row) vektörü girmek için, elemanları boşlukla veya virgül ile ayır.
- Her satır vektörünün sonuna gelindiğinde noktalı virgül (;) kullanarak diğer satıra geç.
- Elemanların çevresine kare parantez(köşeli parantez) gir : []

>> A = [16 3 4 13; 5 10 7 8; 9 6 7 2; 0 5 14 1]

```
A =

    16     3     4    13
     5    10     7     8
     9     6     7     2
     0     5    14     1

>> size(A)

ans =

     4     4
```

sum, transpose, ve *diag*

sum : her satırı(column) toplar ve yeni bir vektör üretir.

Eğer biz tek boyutlu matris(yani vektörler)'de sum fonksiyonunu kullanırsak o vektörün tüm elemanlarını toplar.

```
>>sum(A)

ans =

  30   24   32   24
```

Transpoz(') : " ' " sembolünü kullanarak bir matrisin transpozunu elde edebilirsiniz.Satırlar ve sütunlar yer değiştiriyor.

```
>>A'

ans =

  16    5    9    0

   3   10    6    5

   4    7    7   14

  13    8    2    1
```

```
>> sum(A')'

ans =

   36
   30
   24
   20
```

>>sum(A')'

: Böyle bir işlem tanımladığımızda öncelikle A matrisinin transpozunu aldı ve her bir sütun(column)'u topladı ve bir vektör üretti.Daha sonra bu vektöründe transpozunu aldı.

- Fakat biz iki tane transpoz işleminden kaçınmak istersek ,
  >>sum(A,2) ile aynı işlemi yapabiliriz.

diag : diagonal vektörü bulmak içindir.

```
>>diag(A)

ans =

   16
   10
    7
    1
```

Bazı Fonksiyonlar ile Matris Üretmek

zeros : bütün elemanları 0 olan matris üretir.

ones : bütün elemanları 1 olan matris üretir.

rand : random(rastgele) sayılar üretir.

>> help rand >>help randn >>help fix (fix fonksiyonu sayıları 0'a yakın olan integer'a yuvarlar.)

```
Z = zeros(2,4)
Z =
     0     0     0     0
     0     0     0     0

F = 5*ones(3,3)
F =
     5     5     5
     5     5     5
     5     5     5
```

```
>> fix(100.*rand(1,10))

ans =

   42   91   79   95   65    3   84   93   67   75
```

MATLAB'da Sayılar

Normal girdiğimiz sayılar dışında MATLAB'de bilimsel gösterim " e " ile yapılıyor.10'un kuvveti şeklinde yazmamızı sağlıyor.

İ ve j ise karmaşık sayıların imajiner kısmı için kullanılır.

```
>> 147e5

ans =

   14700000

>> 3e5i

ans =

      0 +3.0000e+005i

>> 6.022e23

ans =

  6.0220e+023
```

ÖRNEK 1 : A = [3 4 7; 1+3j 6 -2j; 1+9j -2 -7j] 3x3 matris

İlk olarak bir dosyada A matrisini oluşturun.Daha sonra for ve if-else döngülerini kullanarak a matrisinin karmaşık sayılarını 0'a atayarak yeni bir matris oluşturunuz.(**help isreal**)

MATRİS İŞLEMLERİ(OPERATORLERİ)

+	toplama	Bu işlemler element-element değil,normal matris işlemleridir.
-	çıkarma	
*	çarpma	
/	bölme	
^	kuvvet	
'	transpoz	

$$\begin{bmatrix}1 & 2 & 3\end{bmatrix} * \begin{bmatrix}4\\2\\1\end{bmatrix} = 11$$

$$1\times 3 * 3\times 1 = 1\times 1$$

$$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix} \text{^} 2 = \begin{bmatrix}1 & 2\\3 & 4\end{bmatrix} * \begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}$$

Must be square to do powers

$$\begin{bmatrix}1 & 1 & 1\\2 & 2 & 2\\3 & 3 & 3\end{bmatrix} * \begin{bmatrix}1 & 2 & 3\\1 & 2 & 3\\1 & 2 & 3\end{bmatrix} = \begin{bmatrix}3 & 6 & 9\\6 & 12 & 18\\9 & 18 & 27\end{bmatrix}$$

$$3\times 3 * 3\times 3 = 3\times 3$$

+	toplama	Bu işlemler element-by-element matris işlemleridir.
-	çıkarma	
.*	Element -element çarpma	
./	Element -element bölme	
.^	Element -element kuvvet(üs)	

$$\begin{bmatrix}1 & 2 & 3\end{bmatrix} .* \begin{bmatrix}4\\2\\1\end{bmatrix} = ERROR$$

$$\begin{bmatrix}1\\2\\3\end{bmatrix} .* \begin{bmatrix}4\\2\\1\end{bmatrix} = \begin{bmatrix}4\\4\\3\end{bmatrix}$$

$$3\times 1 .* 3\times 1 = 3\times 1$$

$$\begin{bmatrix}1 & 1 & 1\\2 & 2 & 2\\3 & 3 & 3\end{bmatrix} .* \begin{bmatrix}1 & 2 & 3\\1 & 2 & 3\\1 & 2 & 3\end{bmatrix} = \begin{bmatrix}1 & 2 & 3\\2 & 4 & 6\\3 & 6 & 9\end{bmatrix}$$

$$3\times 3 .* 3\times 3 = 3\times 3$$

$$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix} .\text{^} 2 = \begin{bmatrix}1^2 & 2^2\\3^2 & 4^2\end{bmatrix}$$

Can be any dimension

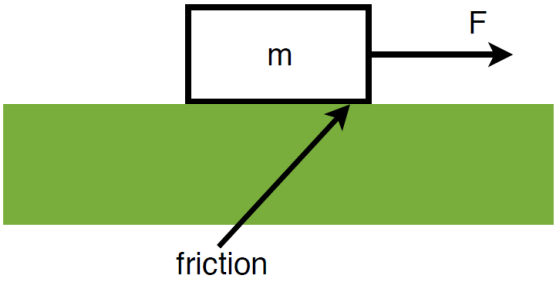
ÖRNEK 2 : **rand** ve **flor, rand** ve **ceil**, kullanarak A : 5x3'lük ve B : 3x5'lik iki matris oluşturunuz.

Ardından şu işlemleri yapınız :

1. A*B
2. A.*B
3. max(sum(A + B))
4. C = A*(5*rand(3,5))
5. C / B
   Hata mesajı aldınız mı???

ÖRNEK 3 : Sürtünme Katsayısı Hesaplama

Aşağıdaki deneyde sürtünme katsayısını bulmak amacıyla deneyler yapılmış ve tablodaki gibi veriler elde edilmiştir.



F (Force) ,kuvvet(N)

m (MASS) , kütle(kg)

g = 9:81 m/s2 yer çekimi ivmesi

$$\mu = \frac{F}{mg}$$

TEST NO.	1	2	3	4	5	6
MASS (KG)	2	4	5	10	20	50
FORCE (N)	12.5	23.5	30	61	117	294

Buna göre formülü de kullanarak sürtünme katsayısını hesaplayıp yazdırınız.

(The University of Edinburgh, School of Engineering , 2010-2014, an interactive introduction to MATLAB)

ÖRNEK 4 : Değişkenler (Şu anki zaman ve tarih)

1. clock fonksiyonunu kullanarak start adında değişken oluştur.
2. start değişkeninin boyutu kaç? Satır vektörü mü yoksa sütun vektörü mü?
3. start değişkenini string'e çevir. Bunun için datestr fonksiyonunu kullan ve startString değişkenine ata.
4. start ve startString değişkenlerini startTime dosyasına kaydet.

```
»help clock
»start=clock;
»size(start)
»help datestr
»startString=datestr(start);
»save startTime start startString
```

Soru "MIT OpenCourseWare Introduction to MATLAB Lecture 1 " den alındı.

= = > help elfun (birçok fonksiyonu görebilirsiniz.)

format fonksiyonu

Nümerik ifadelerin nasıl yazdırılacağını kontrol eder.

```
>> A = [4/3 4.241e-5]

A =

   1.3333    0.0000

>> format short
>> A

A =

   1.3333    0.0000

>> format short e
>> A

A =

  1.3333e+000  4.2410e-005

>> format short g
>> A

A =

     1.3333   4.241e-005
```

```
>> format long
>> A

A =

  1.333333333333333   0.000042410000000

>> format bank
>> A

A =

        1.33         0.00

>> format rat
>> A

A =

      4/3          3/70738

>> format hex
>> A

A =

  3ff5555555555555   3f063c2c81126431
```

format fonksiyonu ile kontrol edebileceğin gibi ,ayrıca **fprintf** ve **sprintf** ile nasıl yazdırılacağını kontrol edebilirsin.

Eğer uzun ifade gireceksen :

Örneğin;

>> s = 1 -1/2 + 1/4 -1/6 + 1/8 - 1/10 + 1/12- 1/8 + 1/14 - 1/16 + 1/18 - 1/20 + 1/22 - 1/24 + 1/26;

Bu ifadeyi tek satırda değil de ikinci satıra da yazdırsaydık;

>> s = 1 -1/2 + 1/4 -1/6 + 1/8 - 1/10 ...

+ 1/12- 1/8 + 1/14 - 1/16 + 1/18 - 1/20 + 1/22 - 1/24 + 1/26;

>>

Aşağıya inmek istediğimiz yerde sonuncu ifadeden sonra boşluk bırakıp ,üç nokta (…) koyduktan sonra, alt satıra enter komutu ile geçebiliriz.

NOT : Örneğin Command Window'a

>>rho = (1 + sqt(5))/2 ifadesini girdik ve sqrt yerine sqt yazdık.

Ve bir hata mesajı aldık.Sonra sqrt girmemiz gerektiğini anladık.

İlk olarak klavyeden ↑ ile ,girdiğimiz ifadeyi tekrar yazıyoruz.Veya Command History'den ifadeyi kopyalıyoruz.Sonra ← ile sola kayıp,eksik olan yere " r " harfini ekliyoruz.

İndeksleme (Indexing)

A(i,j) matrisinde

i , matrisin satırını(row); j ise sütununu(column) gösterir.

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

A(2,3) = 8 olur.

Colon (:) Operatörü

Colon(iki nokta üst üste),MATLAB'in en önemli opratörlerinden biridir.Değişik çeşitlerde kullanılabilir.

>>1:10

1'den 10'a kadar integer satır vektörü üretir.

1 2 3 4 5 6 7 8 9 10

>>100:-9:36

100'den başlayarak her seferinde 7 azalan satır vektörü :

100 91 82 73 64 55 46 37

>>-pi:pi/2:pi

-3.1416 -1.5708 0 1.5708 3.1416

Aşağıda ise,1'den k'ya kadar olan satırlar ve j. Sütun belirtilmiştir :

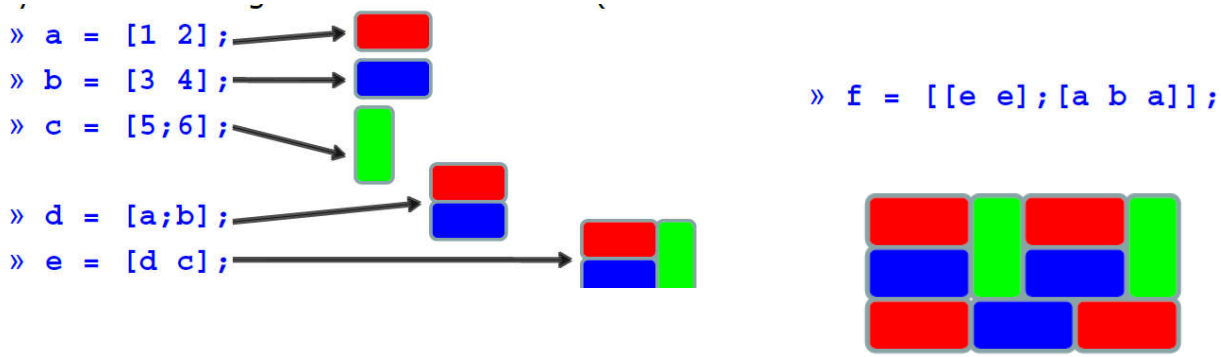
A(1:k,j)

>> A = ceil(10*rand(6,4))

```
A =

    7    8    7    8
    8    1    4    8
    8    3   10    2
    4    1    1    5
    7    1    5    5
    2    9    4    7
```

1) A(2:4 , 1:2)
   Burada satır olarak 2,3 ve4 ; sütun olarak ise 1ve2. 'yi seçer.

```
     8     1
     8     3
     4     1
```



ÖRNEK 5 : Aşağıdaki ifadeleri hesaplayınız.

$$M = \begin{bmatrix} 6 & 9 & 12 & 15 & 18 & 21 \\ 4 & 4 & 4 & 4 & 4 & 4 \\ 2 & 1 & 0 & -1 & -2 & -3 \\ -6 & -4 & -2 & 0 & 2 & 4 \end{bmatrix}$$

a) A = M([1,3], [2,4])
b) B = M(:, [1,4:6])
c) C = M([2,3], :)

Soru şuradan alınmıştır : *Gilat, A. (2008). MATLAB: An Introduction With Applications. John Wiley & Sons, Inc., 3rd edition. Copyright ©2008 John Wiley & Sons, Inc. and reprinted with permission of John Wiley & Sons, Inc.*

SATIR veya SÜTUN Silme :

>> A = [16 3 4 13; 5 10 7 8; 9 6 7 2; 0 5 14 1]

Eğer,3.sütunu silmek istersek,

A(: , 3) = [] burada 3.sütun ve bütün satırların kesişimini seçmiş oluyoruz.Bu da 3.sütun demektir.

```
>> A(2:3 , 1:3) = 0

A =

    16     3    13
     0     0     0
     0     0     0
     0     5     1
```

Yanda ise A matrisinin 2,3. Satırları ve 1,2,3. Sütunlarının kesişimleri seçiliyor ve 0(sıfıra) eşitleniyor.

ÖRNEK 6 : Bir direncin üzerindeki volt değeri (Ohm kanunundan) v = iR verilmiştir.

i,akım(A) v,potansiyel fark(V) R,direnç(Ω)

Bu direncin üzerindeki güç :

$P = Ri^2$

Eğer R=10 Ω ve akım 0'dan 10 A değerine kadar 2'şer 2'şer artıyorsa,akım,volt,güç değerlerini veren MATLAB programı yazınız.(*JOHN O. ATTIA, ELECTRONICS and CIRCUIT ANALYSIS using MATLAB*)

```
% Voltage and power calculation
R=10; % Resistance value
i=(0:2:10); % Generate current values
v=i.*R; % array multiplication to obtain voltage
p=(i.^2)*R; % power calculation
sol=[i v p] % current, voltage and power values are printed
% the last diary command turns off the diary state
```

ÖRNEK 7 : Bir kapasitörün boşalma halinde voltaj değeri aşağıdaki gibi verilmiştir.

$$v(t) = 10(1 - e^{0.2t})$$

Volt değerlerini gösteren tablo(matris, vektör) oluşturur. $v(t)$

t vektörü 0'dan 50s ye kadar ve artış miktarı 5s.

t1 vektörü oluştur.(0'dan 50s ye kadar,artış miktarı 0.5) ve $v(t1)$ grafiğini çiziniz.

ÖRNEK 8 : A = [2 7 9 7; 3 1 5 6; 8 1 2 5] verilen matris ile aşağıdaki işlemleri uygulayınız.

(a) A
(b) A(:,[1 4])
(c) A([2 3], [3,1])
(d) A(:)
(e) [A A]
(f) [A;A(1:2,:)]

Bu soru "*An introductory course in MATLAB: MATLAB for beginners, Alba M. Franco-Pereira, September 2010*"' kaynağından alınmıştır.

Polinomları matris olarak tanımlarız,bu yüzden kısaca polinom işlemlerine de bakalım.

Polinomlar

Matlab'de polinomlar katsayılarının vektörü ile tanımlanır.

Örnek: $P(x) = -2x^5 + 3x^3 – x + 7$ polinomunu yazınız.
P = [-2 0 3 0 -1 7]

x4 ve x2 dereceli terimlerin katsayılarının 0 olarak girildi.

Polinomun kökleri

P polinomunun kökleri **roots** komutu ile bulunabilir.

```
>> r = roots(P)

r =

  -1.2297 + 0.6225i

  -1.2297 - 0.6225i

   1.5136

   0.4729 + 0.9969i

   0.4729 - 0.9969i
```

Kökleri bilinen bir polinom

Kökleri [-2 3] olan polinomu için **poly**

```
ans =

     1    -1    -6 (x^2 - x -6)
```

Polinomun belli bir noktada değeri

Bu amaçla **polyval** fonksiyonu kullanılır.

```
>> polyval(P,2)

ans =
   -35
```

Polinom türevi için : **help polyder**

Polinom integrali için : **help polyint**

LİNEER CEBİR

Lineer Denklemlerin Çözülmesi

Kare Matris(Square Matrix)

Ax = b Least Square? Exact(Kesin çözüm)?

***Over-determined* system :**

Under-determined* systems *:

MATRİSLERİN TERSİ ve DETERMİNANTI :

LİNEER CEBİR

MATLAB'da bu bölümde Lineer Cebir incelenecek.Bunu incelemek bize MATLAB'in matematiksel özelliklerini kullanmada ve lineer denklemleri çözmede yardım edecektir.

Lineer Denklemlerin Çözülmesi

İlk olarak lineer denklemlerin çözülmesini genel olarak inceleyelim.

$$4x_1 + 5x_2 = 6$$

$$3x_1 - 2x_2 = 14$$

Buradaki amacımız x_1 ve x_2 yi bulmak.

>> A = [4 5; 3 -2]

A matrisi eşitliğin sol taraftaki katsayıların yazılması ile oluşturulmuştur.

>> b = [6; 14]

b matrisi ise,çözümleri ifade ediyor.Ancak,bunun sütun vektörü olduğuna dikkat etmeliyiz.

>> x = A\b

A*x = b olduğundan dolayı, $x = A^{-1} * b$ sonucunu alırız.Bu işlemi de yukarıdaki ters bölme(back-slash) ile yapıyoruz.

x =

3.5652
-1.6522

**Eğer sonucun doğruluğunu kontrol etmek istiyorsak A*x işlemini yaptığımızda b vektörünü bulmamız gerekiyor.

>> A*x

ans =

6
14

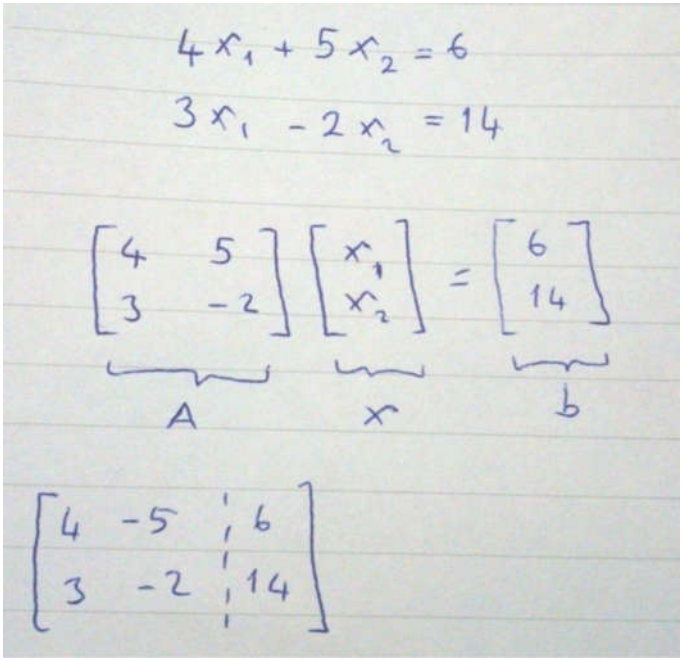
**Başka bir kontrol etme yöntemi de

>> A*x – b

ans =

0
0

*** >> help mldivide ***



ÖRNEK 1 : İki açı birbirinin tümleyenidir. Bir açı diğerinin 2 katından 81 derece eksiktir.Buna göre bu iki açıyı MATLAB ile bulunuz.

Toplamları 90 : A + B = 90

Bir açı diğerinin 2 katından 81 derece : A = 2B -81

Buradan;

A + B = 90

A – 2B = -81

……..

MATLAB ÇÖZÜMÜ???

……..

Sorun Nerede??

Ama biz her zaman bu yöntemle çözümlerimizi yaparsak hata yapabilir ve uyarı mesajları alabiliriz!

İlk uygulamada A matrisi kare matris olduğundan ve b matrisinin satır sayısı ile A matrisinin satır sayısı aynı olduğundan dolayı işlemlerimizde sorun olmadı.

Lakin,A matrisi her zaman kare olmayabilir.Eğer A matrisi mxn boyutunda olsaydı;

m = n Kare matris(Square system)

m > n Overdetermined system olarak adlandırılır. "least-squares solution" bulmamız gerekli

m < n Underdetermined system olarak adlandırılır.

Kare Matris(Square Matrix)

A*x = b şeklinde ,A matrisi nxn matris

Non-singular(Singular olmayan) Matris ise :

```
A = pascal(3);
u = [3; 1; 4];
x = A\u
```

```
x =
10
-12
5
```

A*x bize tam olarak u vektörünü verir ve bu yöntemi kullanırız.

Singular ise :

P = pinv(A)*b

Eğer Ax = b ,tam sonuç vermiyorsa ,pinv(A) kullanılabilir.

pinv = = > pseudoinverse (pseudo inverse)

pseude : yalancı,yapay anlamında

inverse : tersi anlamında

Dikdörtgen(rectangular) matrislerin tersi olmadığından dolayı ,pinv fonksiyonuna ihtiyaç vardır.

Ax = b Least Square? Exact(Kesin çözüm)?

Ax = *b* işlemini yapmak için;

1)

```
A = [      1 3 7
          -1 4 4
           1 10 18 ]

b =[5;2;12]

pinv(A)*b
ans =
0.3850
-0.1103
0.7066

A*pinv(A)*b
ans =
5.0000
2.0000
12.0000
```

exact(kesin çözüm)

Yukarıda ilk olarak A martinin tersi ile b matrisi çarpılarak, sonuç bulundu. Daha sonra doğru olup olmadığını kontrol etmek için A*pinv(A)*b işlemi yapıldı.
Ve $A*A^{-1}*b$ işlemi b vektörünü verdi.

2)

Ax = b ,bu işlemi yapmak için,aşağıdaki çözümü yaptığımızda ve sonucu kontrol etmek için A*pinv(A)*b işlemi yapıldığında b vektörü ,girilen b vektöründen farklı çıkıyor.

```
A = [      1 3 7
          -1 4 4
           1 10 18 ]

b = [3;6;0],

A*pinv(A)*b
ans =
-1.0000
4.0000
2.0000
```

Bu yüzden satırları indirgeme işlemlerini uygulayacağız.
"row reduced echelon form" (rref)

```
rref([A b])
ans =
1.0000  0       2.2857  0
0       1.0000  1.5714  0
0       0       0       1.0000
```

Echelon forma bakıldığında, son satır 0 katsayılara karşılık, 1 sonucu olduğundan dolayı bu eşitliğin çözümü yoktur.Bu yüzden pinv(A) "least-square" sonucunu döndürdü.

>>help eye (birim matris)

Eğer aşağıdaki yöntemler ile ilgileniyorsanız bu fonksiyonları kullanabilirsiniz.

pcg
Preconditioned conjugate gradients method.
bicg
BiConjugate Gradients Method
bicgstab
BiConjugate Gradients Stabilized Method
bicgstabl
BiCGStab(l) Method
cgs
Conjugate Gradients Squared Method
gmres
Generalized Minimum Residual Method
lsqr
LSQR Method
minres
Minimum Residual Method.
qmr
Quasi-Minimal Residual Method
symmlq
Symmetric LQ Method
tfqmr
Transpose-Free QMR Method

***Over-determined* system :**

Over-determined olma durumu, deneysel verilerin, grafikleri için güzel bir kullanım alanıdır.

A matrisi mxn boyutunda bir matris olsun. m>n durumunda ,over-determined olarak sistem adlandırılır.Yani,satır sayısı sütun sayısından fazla,kare matrisi aşmış(over).
Böyle sistemlerde bütün değerleri aynı anda bulamıyoruz.

Örneğin, iki bilinmeyenli lineer bir denklemi çözmek için,iki tane denklem yeter.

$2x_1 + 3x_2 = -4$
$3x_1 + 7x_2 = 2$ ise bu denklemin çözümünden $x_1 = -6.8$ ve $x_2 = 3.2$ bulunur.

Diyelim ki aşağıdaki gibi bir durum yazmak istedik.

$3x_1 + 4x_2 = 1$
$x_1 + 2x_2 = 2$
$3x_1 + 7x_2 = 3$
$4x_1 + 3x_2 = 4$
$x_1 + 4x_2 = 5$

```
>> A = [3 4;1 2;3 7;4 3;1 4]

A =

    3    4
    1    2
    3    7
    4    3
    1    4
```

```
>> b = [1 2 3 4 5]'

b =

    1
    2
    3
    4
    5
```

```
>> x = A\b

x =

    0.2286
    0.5249
```

Ax = b denkleminde,x = A\b ile çözmeye çalışırsak,bize bir çözüm döndürür(least-square solution).Ama,bu sonucun yanlış olduğu kolayca anlaşılıyor.Çünkü denklemlerde yerine koyduğumuzda sağlamıyor.
Ya da kontrol etmek için A*x = b yöntemini kullanırsak,b vektöründen farklı bir vektör buluruz.

```
>> b = A*x

b =

    2.7854
    1.2784
    4.3602
    2.4891
    2.3282
```

```
>> b = [1 2 3 4 5]'

b =

    1
    2
    3
    4
    5
```

Aslında aynı olması gereken b vektörleri aynı değil!!!

Burada hata payını bulmak için:

```
error = A*x – b
norm(error)
```

norm(error) ne kadar küçükse, least-square çözüm için o kadar iyi sonuç elde edilmiş olur.

ÖRNEK 2 : Yapılan bir deneyde değişik t zamanlarında,y nin değeri aşağıdaki gibidir.Aşağıdaki komutları Command Window'a girerek değerleri görebilirsiniz.

```
t = [0 .3 .8 1.1 1.6 2.3]';
y = [.82 .72 .63 .60 .55 .50]';
B = table(t,y)
```

Ve bu deneyde öngörülen denklem

$y(t) = c_1 + c_2e^{-t}$ şeklindedir.

\ ters bölme(back slash) kullanarak least square çözümünden c_1 ve c_2 yi bulunuz.

Bir de T = 0:0.1:2.5 ve Y(T) = $c_1 + c_2e^{-T}$ 'yi hesaplayınız.

plot(T,Y,'-',t,y,'o') plot'u çizdiriniz.

Under-determined* systems *:

A matrisi mxn boyutlarında ve m<n boyutunda ise,bu sistemler *under-determined* ya da *undetermined* oluyor.Bunun anlamı ;denklemi çözmek için yeterli bilginin olmamasıdır. Örneğin,iki bilinmeyenli denklemi çözmek için ,iki tane lineer denkleme ihtiyacımız vardır.

$$4x_1 + 5x_2 = 6$$

Burada A matrisi dediğimiz A = [4 5] yani 1x2 boyutunda bir matristir.

Bu eşitliği $x_2 = -(4x_1 - 6) / 5$ olarak çözeriz.

x_1 'in değeri -∞ ile +∞ arasında herhangi bir değer seçilerek x_2 değeri bulunabilir.

Eğer biz bu işlemi MATLAB üzerinden yapmak istersek, sonsuz çözümün sadece birini döndürecektir.

```
>> A = [4 5]

A =

     4     5

>> b = 6;
>> x = A\b

x =

         0
    1.2000
```

MATRİSLERİN TERSİ ve DETERMİNANTI :

Eğer bir matris kare ve non-singular(singular olmayan) ise AX = I ve XA = I aynı çözümü verir.Burada X matrisi A^{-1} ya da A matrisinin tersidir.

>>help det
>>help inv

>>A=pascal(5) 5x5 boyutunda pascal üçgeni oluşturan matris. >>help pascal
>>d = det(A)
>>b = inv(A)

```
>> format rat
>> b

b =

     5    -10     10     -5      1
   -10     30    -35     19     -4
    10    -35     46    -27      6
    -5     19    -27     17     -4
     1     -4      6     -4      1

>> d

d =

     1
```

>>help magic
>>B = magic(4)
>>d = det(B)
>>X =inv(B)

NOT : Eğer A ve B kare matris ve singular olmayan ise, X = inv(A)*B ile X = A\B;
ve Y = B*inv(A) ile X = A\B teorik olarak aynıdır.
Ama,bölme(slash) ve ters slash(back slash) ile hesaplamak daha hızlı ve bellekte daha az yer kaplıyor.

>>help rank
>>help sqrtm
>>help expm
>>help eig
>>help schur