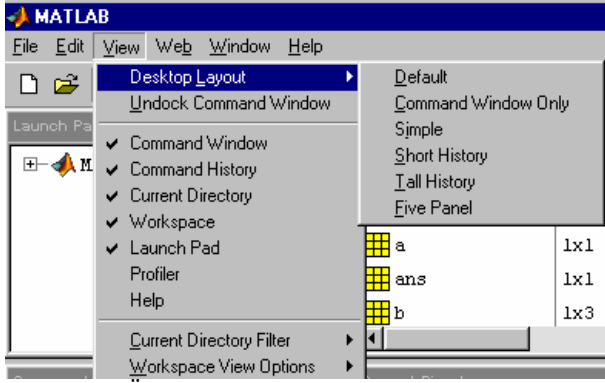


MATLAB

Matlab, MATrix LABoratory kelimelerinden türetilmiş, daha çok matematiksel işlemler yaptırmaya yönelik tasarlanmış bir bilgisayar programıdır.

Matlab'ın Genel Yapısı :

Matlabı çalıştırdığımızda, karşımıza ana pencere gelir.Bu pencerede **File, Edit, View, Web, Window** ve **Help** ana başlıkları vardır.Bu başlıklar altından, diğer Windows programlarında alışlagelmiş benzer işlemler



yapılabilir.Örneğin; File ile klasik dosyalama işlemleri, Edit ile çalışılan dosyadaki düzenleme işlemleri, View ile görünüm ayarlamaları, Web ile, ilgili İnternet bağlantıları, Window ile, Matlab dışında açılan pencerelerin, uygulamaların ve figürlerin kapatılmasını, Help ile de program ya da işlemler ile ilgili yardım almayı sağlar.

Ortalama bir bilgisayar kullanıcısı, yukarıda sayılan bölümlerin, kabaca ne anlama geldiğini bilir.Ancak View (Görünüm) ile ilgili bilinmesi gereken birkaç maddeyi açıklamakta fayda var.Bu bölüm ve alt seçeneklerinin görünümü yandaki gibidir.Burada Desktop Layout ile Matlabın masaüstü yerleşimini düzenleyebilirsiniz.Örneğin; **Default** ile varsayılan görünümünü, **Command Window Only** ile sadece klasik komut penceresini, **Five Panel** ile çok kullanışlı ve çok amaçlı olan 5 pencereli görünümünü seçebiliriz.Genellikle Five Panel görünümünde çalışmak daha uygundur.Bu görünüm seçildiğinde karşımıza, adından da anlaşılacağı gibi Matlab 5 pencereden izlenebilir ve çalışılabilir.Bu pencereler ve kısaca yapılan işlemler şunlardır:

Launch Pad: Matlab kısayollarının bulunduğu penceredir.Bu pencereden Matlab uygulamalarına, simulink penceresine, araç kutularına ve blok setlerine ulaşılabilir.Örneğin Matlab ile ilgili yapılabilecek olan işlemler hakkında bilgi sahibi olmak için bu pencereden yararlanabiliriz.Örneğin Matlab ile ilgili yapılabilenleri, demo olarak izlemek istersek; MATLAB-Demos sekmesine tıklamalıyız.Karşımıza Desktop Environment, Matrices, Numerics, Graphics, Language ... gibi alt bölümler çıkar.Örneğin Grafik ile ilgili bilgilenmek ve bazı grafiklerin demolarını görmek istersek Graphics bölümünü tıklamalıyız.Bu bölümü (veya yanındaki + işaretini) tıkladığımızda, ... 2-D Plots, 3-D Plots, ... gibi bölümler görünür.Örneğin 2-D Plots tıklanırsa iki boyutlu grafiklerle ilgili, 3-D Plots tıklanırsa üçboyutlu grafiklerle ilgili demoları görebilir ve inceleyebiliriz.

Command Window: Adından da anlaşılacağı gibi bu pencere **komut penceresi** olup Matlabın en önemli penceresidir.Bu pencereden Matlab ile ilgili komutları klavyeden girer, komutun işlenmesini sağlamak için de Enter tuşuna basarız.Komutları girdiğimiz satır **>>** ile başlar ki bu satıra **komut satırı** denir.Tabii dir ki komut satırına, Matlab için anlamlı komutlar yazmalıyız.

Örneğin naber yazıp enter tuuna basarsak **??? Undefined function or variable 'naber'.** gibi bir karşılık alırız.Bu da naber adlı ne bir fonksiyon ne de bir değişkenin tanımlanmamış olduğu anlamına gelir.

Yine komut satırına **naber='iyidir'** yazıp enter tuşuna basarsak; ekranda;
naber =

iyidir görünür.

Örneğin a=3 (enter), b=-7 (enter) işlemlerini yapıp a*b (enter) yaptığımızda ekranda;
ans =

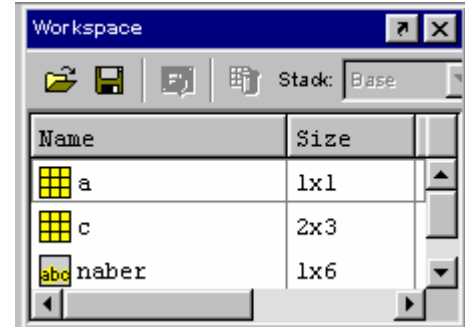
-21 görülür.

Örneğin; komut satırına **clc** yazıp enter tuşuna basarsak, komut penceresine yazılan komutların tümü silinir ve kursor (imleç) pencerenin en üst ve sol köşesine konumlanır.

Command History: Bu pencere o ana kadar komut satırından girilen komutları gösterir.İstersek bunların birini fareyle seçer, ya da bir kaçını veya tümünü fare ve aşağı-yukarı yön tuşları yardımıyla seçer ve **delete** tuşuna basarak silebiliriz

Workspace:

Komut satırından ya da çalıştırılan bir dosya ya da fonksiyon ile hafızada oluşturulan değişkenlerin adlarının, tiplerinin ve özelliklerinin görüntülediği penceredir.Bu alana çalışma alanı denir.



Örneğin bu pencerenin görüntüsü yandaki gibiyse; **a** değişkeninin 1x1 boyutunda bir **matris** yani sayı, **c** değişkeninin ise 2x3 boyutunda bir matris, yani iki satır ve 3 sütundan oluşan bir matris, **naber** adlı değişken de 6 karakterden oluşan bir **karakter zinciri (string)** olduğu görülür.

Current Directory: Matlab dosyalarının kaydedildiği, yüklendiği dosyaların bulunduğu klasörü (dizin), varsayılan klasör olarak belirlemeye yarar..Aksi belirtilmedikçe bu klasör **C:\Matlab6p5\work** gibi bir klasördür.

Matlab'da Matematiksel İşlemler:

Matlab'da bir çok eylem, dört işlem ve matematiksel bazı işlemler yaptırabiliriz.Bunun için, ya ilgili komutları komut penceresinden teker teker girerek veya ilgili komutları bir dosyaya yazıp, o dosyayı çağırarak

Matlab Ders Notları

çalıştırabiliriz.İşlemleri yaptırırken, sayıları reel sayı ya da karmaşık sayı olarak alabiliriz.Bunu aşağıdaki örneklerde inceleyelim:

- 1) Komut satırına a) $2+3$ b) $24-3*(4-2)$ c) $12-12/6*8$ d) 2^3
e) $(2-3i)*(4+i)$ f) $\sin(30)$ g) $\sin(30*\pi/180)$ yazıp enter tuşuna bastığımızda ne olur?
Çözüm:
a) 2 ile 3 ün toplamı 5 görülür.
b) Önce parantez içindeki işlem yapılır (2), sonra 3 ile 2 çarpılır (6), son olarak ta 24 ten 6 çıkarılarak 18 soncu elde edilir.
c) 12 6 ya bölünür (2), 8 ile çarpılır (16), 12 den 16 çıkarılarak -4 sonucu bulunur.
d) 2 nin 3 üncü kuvveti alınarak 8 elde edilir.
e) $2-3i$ karmaşık sayısı ile $4+i$ karmaşık sayısının çarpımı olan $11.0000 -10.0000i$ sonucu görülür.
f) -0.9880 sonucu görülür ki bu 30° nin sinüsünde n farklıdır.Çünkü bu 30° derece de ğil 30 radyanın sinüsüdür.
g) 0.5000 sonucu görülür ki bu da 30° nin sinüsüdür.O halde bir sayının trigonometrik değerini buldurmak için, önce pi ile çarpıp 180 e bölerek açıyı radyan çevirip sonra trigonometrik değerini hesaplayabiliriz.

O halde örnekte görüldüğü gibi Matlab'da; matematiksel işlemleri, komut satırından girip enter tuşuna basarak sonuçlarını görebiliriz.İşlemlerde kullanılan semboller, bazı temel matematiksel fonksiyonlar ve anlamları yandaki tabloda görülmektedir.

- 2) Komut satırına $a=5$ (enter) $b=-3$ (enter) $c=a+3*b$ (enter) yazdığımızda ekranda sırasıyla a, b ve c değişkenlerinin değerleri nelerdir?
C: 5 -3 ve -4

- 3) Hafızadaki değişkenlerin a) sadece adlarını b) her bir değişkenin tipini ve kapladığı alanı görüntülemek için hangi komutlar kullanılır?
C: a) who b) whos

- 4) hafızadaki a) a değişkeninin b) a, b, z değişkenlerinin c) tüm değişkenlerin değerlerini silmek için hangi komutlar kullanılır?
C: a) clear a b) clear a b z c) clear

- 5) Yarıçapı 5 birim olan dairenin alanını buldurmak için hangi girişleri yapmalıyız?
C: $\pi*5^2$ veya $\pi*25$

Matlab'da Temel Kavramlar:

Anahtar Kelimeler: Tüm programlama dillerinde olduğu gibi (Fortran, C, Pascal, Basic vs..) Matlab'ın da özel anlam taşıyan bazı kelimeleri vardır ki, bu kelimeler değişken olarak kullanılamazlar.Bu tür kelimelere **anahtar kelime (keywords)** denir.Bu kelimeler; '**break**' '**case**' '**catch**' '**continue**' '**else**' '**elseif**' '**end**' '**for**' '**function**' '**global**' '**if**' '**otherwise**' '**persistent**' '**return**' '**switch**' '**try**' '**while**' dir. Bu kelimelerin bir listesini almak için komut satırına; **iskeyword** komutunu yazarak elde edebiliriz.
Sabitler, Değişkenler ve Metin Katarları:

Tüm programlama dillerinde olduğu gibi, program içinde değeri değişmeyen değerlere **sabit**, değeri değişebilen bellek alanına işaret eden değerlere **değişken**, değeri karakterlerden oluşan değerlere de **metin katarı (string)** denir.Matlab'da değişkenler büyük küçük harfe duyarlıdır.Örneğin a değişkeni ile A değişkeni farklıdır.

Değişkenlere Değer Atama:

Her hangi bir programlama dilinde olduğu gibi, Matlab'da da bir değişkene değer verme işlemine "değer atamak" denir.Bir değişkene atanan değer, değiştirilmediği sürece aynı kalır.

Değer atamanın genel kullanımı aşağıdaki biçiminde olur:

<Değişken adı>=<Atanacak değer>;

Örnekler:

- 1) **a=1;** işlemi ile a adlı sayısal değişkene 1 sayısını atamış oluruz.
2) **a='İzmir'** işlemi ile a adlı string değişkenine İzmir stringini atamış oluruz.
3) **a=5;b=7;c=a+b;** işlemleri sonucunda a değişkenine 5, b değişkenine 7 ve c değişkenine a ve b değişkenlerinin değerleri toplamı olan 12 sayısını atamış oluruz.
4) Aşağıdaki atamalar sonucunda değişkenlerin son durumlarının ne olacağını bulalım.
 $a=5;b=-3;c=a+2*b;a:=a+b;$

Çözüm:

a	b	c	Açıklama
5	-3	-1	$5+2*(-3)=5-6=-1$
2	-3	-1	$5+(-3)=2$

Özel Sabitler: Matlab'da önceden tanımlanmış bazı sabitlerdir.Bunlar aağıdaki tabloda gösterilmiştir.

Özel Sabit	Anlamı	Değeri
eps	Sıfıra çok yakın bir sayı (epsilon)	2.2204e-016
realmin	Tanımlanabilen en küçük reel sayı	2.2251e-308
realmax	Tanımlanabilen en büyük reel sayı	1.7977e+308
pi	pi sayısı	3.1416
i, j	Karmaşık sayıların sanal birimi	$0 + 1.0000i$
inf	Sonsuz	Inf
computer	Bilgisayarın tipi	PCWIN
version	Matlab'ın versiyonu	6.5.0.180913a (R13)

Matlab'da Dizi (Matris) İşlemleri:

Hasan KORKMAZ- İzmir Fen Lisesi Matematik Öğretmeni

Matlab Ders Notları

Sayılardan oluşan satır ve sütun yapısına matris (dizi) denir.

Örneğin;

d1=[5] 1x1 lik, d2=[2 -7] 1x2 lik,

d3= [1 0 -3]

[5 3 1] 2x3 lük bir dizidir. Matlab da bu dizileri;

Komut satırında; **d1=[5]** veya **d1=5 ile;**

d2=[2 -7] veya **d2=[2,-7]** ile;

d3=[1 0 -3;5 3 1] veya **d3=[1,0,-3;5,3,1]** veya

d3=[1 0 -3

5 3 1] ataması ile oluşturabiliriz

Dizilerin Değerlerinin Değiştirilmesi ve Düzenlenmesi:

Bir dizinin herhangi bir elemanını belirlemek için dizi adından hemen sonra parantez içinde elemanın bulunduğu satır ve sütun sayısı yazılmalıdır.

Örnek:

a) Yukarıda tanımlanan d2 dizisinin -7 elemanını görüntülemek için ne yapılmalıdır?

b) Yukarıda tanımlanan d3 dizisinin 2. satır, 1. sütununda bulunan 5 in değerinin, -7.5 olması için ne yapılmalıdır?

Çözüm: a) d2(1,2) b) d3(2,1)=-7.5;

Not:1) Bir dizinin bir çok elemanını yeniden değer atamak gerekirse, komut satırından atama yapmak uzun zaman alabilir. Bu durumda dizi değişkeninin üzerine çift tıklayarak açılan dizi editörü (array edit) yardımıyla değişiklikleri daha kolay yapabiliriz.

2) Bir diziye düzenli artış (veya azalış) kuralıyla değerler atanmak isteniyorsa bunu;

ilk_değer:artış:son_değer veya **ilk_değer:artış:son_değer** biçiminde yapabiliriz. Ancak artış 1 ise belirtilmeyebilir.

Örnek:

a) **puan** adlı bir diziye 1 den 100 e kadar sayıları atayan;

b) **ortalama** adlı bir diziye 0 dan 5 e kadar 0.5 er artışla elde edilen sayı dizisini atayan;

c) 1. satır 7 den 17 ye kadar olan tam sayılar, 2. satırı 99 dan 89 a kadar azalan tam sayılardan oluşan 2 boyutlu m dizisine atayan işlemleri yazınız.

Çözüm:

a) **puan=[1:1:100];** veya **puan=1:1:100;** veya **puan=1:100;**

b) **ortalama=[0:0.5:5];**

c) **m=[7:17;99:-1:89];**

Özel Dizi (Matris) Oluşturan Bazı Fonksiyonlar:

a) Sıfır Matrisi Oluşturan Fonksiyon:

Her elemanı sıfır olan mxn boyutunda bir matrise **sıfır matrisi** denir. Böyle bir dizi oluşturmak için **zeros** fonksiyonu kullanılır. Kullanımı; **matris_adı=zeros(m,n);** biçimindedir.

Örnek:

3x5 boyutunda s adlı sıfır matrisi oluşturalım.

Çözüm: **s=zeros(3,5);**

b) 1 lerden Oluşan Matris:

Her elemanı 1 olan mxn boyutunda bir matrisi oluşturmak için **ones** fonksiyonu kullanılır.

Kullanımı; **matri_aadı=ones(m,n);** biçimindedir.

Örnek:

2x3 boyutunda b adlı tüm elemanları 1 olan matrisi oluşturalım.

Çözüm: **b=ones(2,3);**

c) Birim Matrisi Oluşturan Fonksiyon:

Esas köşegeni 1 lerden diğer elemanları 0 lardan oluşan matrisie kare matrise (satır ve sütun sayısı eşit olan) **birim matrisi**, kare olmayan matrise de **diyagonal matris** denir. Böyle matrisleri oluşturmak için **eye** fonksiyonu kullanılır. Kullanımı; **matris_adı=eye(m,n);** biçimindedir.

Örnek:

a) 3x3 lük birim matris;

b) 4x3 llük diyagonal matris oluşturalım.

Çözüm: a) **i=eye(3,);** b) **d=eye(4,3);**

d) Rastgele Sayılardan Oluşan Matris ve Fonksiyonu:

Elemanları 0 ile 1 arasındaki rastgele sayılardan oluşan bir matris için **rand** fonksiyonu kullanılır.

Kullanımı; **matris_adı=rand(m,n);** biçimindedir.

Not 1) Üretilen matrisin tüm elemanlarını k gibi bir sayı ile çarparak, sayıları 0 ile k arasına çekebiliriz.

Not 2) Ondalık sayılardan oluşmuş bir matrisin elemanlarını yuvarlayıp tam sayı yapmak için **round** fonksiyonunu kullanırız.

Örnek:

a) 0 ile 1 arasında rastgele sayılardan oluşan 10 elemanlı a adında bir satır matrisi (dizisi, vektörü) oluşturalım.

b) Elemanları 10 ile 50 arasında sayılardan oluşan 5x3 tipinde b matrisini oluşturalım.

c) Elemanları 50 ile 300 arasındaki tamsayılardan oluşan 3x4 tipinde c matrisini oluşturalım.

Çözüm:

a) **a=rand(1,10);** b) **b=10+rand(5,3)*40;**

c) **c=round(50+rand(3,4)*250);**

e) Rastgele Sayılardan Oluşan Normal Dağılımlı Matris ve Fonksiyonu:

Elemanları rastgele sayılardan oluşan bir normal dağılımlı bir matris için **randn** fonksiyonu kullanılır.

Kullanımı; **matris_adı=randn(m,n);** biçimindedir.

Örnek: Rastgele sayılardan oluşan normal dağılımlı 2x3 lük bir n matrisini oluşturalım.

Çözüm: **n=randn(2,3);**

f) Lineer Aralıklı (Aritmetik) Dizi ve Fonksiyonu:

Başlangıç ve bitiş değerleri ve kaç elemandan oluşacağı belirlenen diziye oluşturmak için **linspace** fonksiyonu kullanılır.

Kullanımı;

dizi_adı=linspace(ilk_değer,son_değer,eleman_sayısı); biçimindedir.

Örnek: 10 ile 30 arasına 9 tane daha sayı koyarak a adında bir aritmetik dizi oluşturalım.

Çözüm: 10 ve 30 (ilk ve son terimler) diziye dahil olacağından terim sayısı 11 dir. O halde komut; **a=linspace(10,30,11);**

Matris İşlemleri:

Matlab Ders Notları

Matlab'da sayılardan oluşan matrislerle ilgili bazı işlemler yaptırmak mümkündür.Örneğin 1 den 100 e kadar olan sayıları 1x100 lük bir a matrisine, kareleri dizisini de 1x100 lük bir b matrisine atamak daha sonra da karılıklı elemanları toplamını da bir c matrisine atamak isteyebiliriz.Veya 2x3 lük iki matrisi toplaya bilir, çıkarabilir ya da birincinin 3 katına ikincinin -3 katını ilave edebilir ve sonuç matrisinin tüm elemanlarının 7 fazlasını buldurmak isteyebiliriz.Veya 2x3 lük bir a matrisi ile 3x4 lük bir b matrisinin çarpımını c matrisine atamak isteyebiliriz.İşte bu ve bunun gibi işlemlere **matris işlemleri** denir.Şimdi bu işlemlerin bazılarını görelim.

a) Toplama-Çıkarma Bir Sayı ile Çarpma İşlemi:

İki matrisi toplamak (veya çıkarmak) demek, matrislerin aynı mertebedeki elemanları teker teker toplayıp (veya çıkarıp) aynı mertebeye yazmak demektir.Bu durumda iki matrisin de aynı mertebeden olması gereği açıktır.Bir matrisi sabit bir sayıyla ile toplamak (veya çıkarmak) demek, matrisin elemanlarınının tümünü teker teker o sayıyla toplamak (veya çıkarmak)demektir.Bir matrisi sabit bir sayıyla ile çarpmak demek ise, matrisin elemanlarınının tümünü teker teker o sayıyla çarpmak demektir.

Örnek: $a = [-1 \ 3 \ 5; 2 \ 1 \ 7]$ ve $b = [3 \ -3 \ -4; 1 \ 1 \ 5]$ matrisleri veriliyor.

a) $c = a + b$ toplam matrisini b) $d = a - b$ matrisini c) a matrisinin her elemanınının 5 eksiğine karşılık gelen e matrisini d) $f = 2a - 3b$ matrisini bulduran işlemleri yazalım.

Çözüm:

a) $c = a + b$ b) $d = a - b$ c) $e = a - 5$ d) $f = a + a - b - b$ veya $f = 2a - 3b$

b) İki Matrisin Çarpımı, Bir Matrisin Kuvvetleri ve Çarpma İşlemi:

İki matrisin çarpım işlemi iki biçimde anlaşılır.

1) Aynı mertebeden iki matrisin elemanlarını teker teker , çarpıp, aynı mertebeye yazmak demektir.Bunu \cdot işlemi ile gerçekleştiririz.

2) Matematiksel anlamda iki matrisi çarpmak istediğimizde; birinci matris $m \times n$ türünde ve ikinci matris mutlaka $n \times p$ türünde olmalıdır; yani birinci matrisin sütun sayısı ikinci matrisin satır sayısına eit olmalıdır.Bu durumda birinci matrisin i. sütun elemanları ile, ikinci matrisin j. satırındaki elemanlar karılıklı olarak çarpılır ve sonuçlar toplanır ve bu toplam çarpım matrisinin (i,j) inci mertebeye yazılır.Matrisler arası çarpma işleminin sembolü de \cdot dir.

3) Bir a matrisinin her bir elemanının n. kuvvetlerinden oluşan matrisi bulmak için $a \cdot^n$ işlemi kullanılır.

4) Satır ve sütun sayıları eşit bir kare matrisi ardışık olarak n defa kendisiyle çarparak, a matrisinin n. kuvvetini bulabiliriz.Örneğin a matrisinin karesi için $a \cdot a$ veya a^2 , kübünü buldurmak için $a \cdot a \cdot a$ veya a^3 , dördüncü kuvvetini buldurmak için $a \cdot a \cdot a \cdot a$ veya a^4 işlemiyle buldurabiliriz.Ancak 2005 nci kuvvetini buldurmak için a^{2005} yazmak yeterlidir.

Not) Bir a matrisinin eleman -elemana çarpma işlemine benzer mantıkla, bir matrisin tüm elemanlarının kareleri, küpleri, sinüsleri, kosinüsleri, logaritmalarından ... oluşan matris bulunmak istenirse; bunu sırayla $a \cdot a$

(veya a^2), $a \cdot a \cdot a$, (veya a^3), $\sin(a)$, $\cos(a)$, e tabanında logaritması için $\log(a)$, 10 tabanında logaritmaları için $\log_{10}(a)$... biçiminde gerçekleştirebiliriz.

Örnek: $a = [-1 \ 3 \ 5; 2 \ 1 \ 7]$, $b = [3 \ -3 \ -4; 1 \ 1 \ 5]$ ve $c = [1 \ 0; -1 \ 2; 3 \ 3]$ matrisleri veriliyor.

- a matrisinin elemanları ile b matrisinin elemanlarını karşılıklı çarpımlarından oluşan c1 matrisi varsa bulalım.
- a matrisi ile b matrisinin çarpım matrisi olan c2 varsa bulalım.
- a matrisinin elemanları ile c matrisinin elemanlarını karşılıklı çarpımlarından oluşan c3 matrisi varsa bulalım.
- a matrisi ile c matrisinin çarpım matrisi olan c4 varsa bulalım.
- a matrisinin elemanlarının karelerinden oluşan matris ile b matrisinin kosinüslerinden oluşan matrisler toplamını bulalım.
- $x = [1 \ 0; 0 \ 3]$ matrisinin i) Karesini ii) Kübünü iii) 10. kuvvetini bulalım.

Çözüm:

- İki matrisin karşılıklı elemanlarının çarpımından oluşan matrisin tanımlı olabilmesi için aynı mertebeli olması gerekir.Bu durumda c1 matrisi tanımlıdır ve bunu $c1 = a \cdot b$ işlemi ile gerçekleştirebiliriz.
- İki matrisin çarpılabilmesi için birinci matrisin sütun sayısı ikinci matrisin satır sayısına eşit olmalıdır.Halbuki a matrisi 2×3 b matrisi de 2×3 olduğundan bu iki matris çarpılamaz.
- İki matrisin karşılıklı elemanlarının çarpımından oluşan matrisin tanımlı olabilmesi için aynı mertebeli olması gerekir.Halbuki bu matrisler aynı mertebeden olmadığından bu iki matris eleman-elemana çarpma işlemi gerçeklemez.
- İki matrisin çarpılabilmesi için birinci matrisin sütun sayısı ikinci matrisin satır sayısına eşit olmalıdır.a matrisi 2×3 lük, c matrisi de 3×2 lik olduğundan bu iki matris çarpılabilir ve c4 çarpım matrisi 2×2 lik bir matris olur.c4 çarpım matrisini $c4 = a \cdot c$ işlemi ile buluruz.
- $a^2 + \cos(b)$
- i) x^2 ii) x^3 iii) x^{10}

c) Bir Matrisin Devriğini (Transpozesi) Bulma İşlemi:

Bir matrisin satırlarını sütun, sütunlarını satır olarak yazılmasıyla bulunan matrise, bu matrisin **devriği (transpozesi)** denir.Bir matrisin devriğini \cdot' işlemi ile bulabiliriz.

Örnek: Bir önceki örnekteki a matrisinin devriğini buldurup d matrisine atayalım.

Çözüm: $d = a \cdot'$;

d) İki Matrisin Bölümü, Birim Matris ve Bir Matrisin Ters :

Aynı mertebeden iki matrisin elemanlarını teker teker , bölerek, aynı mertebeye yazılmasına iki matrisin sol bölmesi denir ve bu $/$ işlemi ile yapılır.

a, b ve c aynı mertebeden kare matrisler olmak üzere; $c = a \cdot b$ ise a matrisine c nin b matrisine **bölümü** denir. c bölüm matrisi $/$ işlemi ile yapılır.

Matlab Ders Notları

Esas köşegeni 1 sayılarından diğer elemanları 0 lardan oluşan **kare** matrise **birim matris** denir.

Örneğin 1x1 lik birim matris [1],
2x2 lik birim matris [1 0;0 1],
3x3 lük birim matris [1 0 0;0 1 0;0 0 1],
4x4 lük birim matris [1 0 0 0;0 1 0 0; 0 0 1 0;0 0 0 1] dir.
Birim matris oluşturmak için; **eye** fonksiyonunu kullanırız.

Örneğin;
2x2 lik i2 adlı birim matrisi **i2=eye(2,2);**
3x3 lük i3 adlı birim matrisi **i2=eye(3,3);** işlemi ile oluşturabiliriz.
Aynı mertebeden a ve b kare matrisleri için a ile b nin çarpımı **birim matris** ise b matrisi a matrisinin (aynı biçimde a matrisi de b matrisinin) **ters matrisidir**.

Örneğin 3x3 lük bir a kare matrisinin tersini bulmak için **eye(3,3)/a** veya **inv(a)** işlemini kullanırız.

Örnek:
a=[2 -10 0;1 2 4;3 0 1] matrisi ile b=[1 5 4;1 -1 2;0 1 -1] matrisleri veriliyor.
a) a matrisinin elemanlarını sırasıyla b matrisinin elemanlarına bölerek elde edilen matrisi b1 matrisine atayalım.
b) a matrisinin ta ters matrisini bulalım.
c) a ile ta matrisinin çarpımının 3x3 lük birim matris olduğunu gösterelim.
d) a matrisinin b matrisine bölümünü b2 matrisine atayalım.

Çözüm:
a) b1=a./b b) ta=eye(3,3)/a veya ta=inv(a)
c) a*ta ==eye(3,3) d) b2=a/b

Matrisler İle İlgili Bir Uygulama: Lineer Denklem Sistemlerinin Çözümü:

Matris ile ilgili işlemlerin bir çok uygulama sahası vardır.Bunlardan biri de lineer denklem sistemlerinin çözümüdür.Bunun için önce katsayılar matrisi elde edilir, bu matris a olsun.Denklem sistemindeki eitiklerin sağ tarafındaki sabit sayılardan oluşan matris b olsun. Bilinmeyenlerden oluşan matris x olmak üzere denklem sistemi **ax=b** matris eşitliği biçimine getirilmiş olur. Buradaki x bilinmeyenler matrisini bulmak için, a nın tersi ile b matrisini çarparız yani **inv(a)*b** işlemini yaparız.

Örnek:
2x-3y+z= 15
x-z = -3
x+y+z = 2 denklem sistemini çözelim.

Çözüm:
a=[2 -3 1;1 0 -1;1 1 1]; b=[15;-3;2]; x=inv(a)*b

Matlab'da Programlama

Her hangi bir bilgisayar dilinde program yaparak, istediğimiz bazı işlemleri yaptırabiliriz.Matlab'da da bir program yazarak benzer işlemlerimizi

yaptırabiliriz.Bunun için Matlab'da kullanılan komut ve deyimleri örneklerle inceleyelim.

1) x değişkenine 5 atayarak x in 2 katının 3 eksiğini bulduralım.
Ç: x=5 (enter) 2*x-3 (enter)

2) Girilen bir x değerini için, karesinin 3 katından 5 eksiğini hesaplatan bir program yazalım.
Ç: Bunun için klavyeden girilen değeri x gibi bir değikene atamalıyız.Bunun için **input** komutundan yararlanılır. Kullanımı **değişken=input('açıklayıcı ifade');** biçimindedir.
x=input('sayıyı gir!'); (enter)
3*x^2-5 (enter)

Not:Her ne kadar Matlab'da bu şekilde işlem yaptırabilirsek de; daha uzun işlemler yaptırmak istediğimizde, komutları tekrar tekrar yazmak hem uzun zaman alır, hem de hata durumunda düzeltmesi zor olur.Onun için program için gerekli komutları yazdıktan sonra bunları bir dosyaya kaydedip sonra gerektiğinde bu dosyayı çalıştırabiliriz.Bu amaçla yazılan Matlab dosyalarına **m** dosyaları denir ve bunların uzantısı m dir.Böyle bir dosya yazmak için; **File - New - M-File** sekmesi tıklanırsa; yeni bir m dosyası ekranı gelir.Bu dosyaya Matlab komutları yazılır ve **File-Save** sekmesinden, dosyaya bir ad verilerek kayıt ortamına kaydedilir.Diske kaydedilen bir m dosyasını çalıştırmak için, komut satırından ismi girilerek çalıştırılır.

3) Bu açıklamalar sonucunda yukarıdaki probleme uyan, yani girilen bir sayının karesinin 3 katının 5 eksiğini bulup ekrana yazdıran bir m dosyası yazalım.
Ç:File-New-M-File sekmesi tıklanır, gelen yeni m dosyası sayfasına sırasıyla aşağıdaki komutlar yazılır.

x=input('sayıyı gir');
3*x^2-5

Matlab'da Kullanılan Bazı Komut ve Deyimler: Değişkenlere Değer Atama: input

Amaç: Matlab'da bir değışkene bir değeri atamak.

Kullanımı:

değişken=input('Açıklama') veya
değişken=input('Açıklama','s')

Değişken sayısal değışkense ilk yazılan ifade; karakter dizisi değışkeni (string) ise ikinci ifade kullanılır.

Değişken Değerlerini Ekrana Yazdırma: fprintf ve disp

Amaç: Değişkenlerin değeri ekrana yazdırmak.

Kullanımı:

fprintf('Açıklama <biçim ifadesi>',değişken)
disp(değişken)

Not: "Biçim ifadesi" yerine, değışken string (karakter zinciri) ise **%s** değışken reel sayı (kayan noktalı) ise **%f** , üstel biçimde gösterilecekse **%e** sembolü kullanılır. Ayrıca değışkenin değeri yazdırıldıktan sonra kaç satır atlatılacaksa okadar **\n** ifadesi yazılır.

Örnek:

Matlab Ders Notları

Klavyeden bir kişinin adı, soyadı ve yaşı girilerek; temiz ekrana ilgili kişinin kaç yaşında olduğunu yazdıran bir program yazınız.

Çözüm:

```
ad=input('Adınız :','s');
soyad=input('Soyadınız :','s');
yas=input('Yaşınız :');
clc;
fprintf('Siz %s %s %d yaşındasınız.',ad,soyad,yas);
```

Karar Verme ve Dallanma:

Bazen bir programda, belirli bir şartın gerçekleşmesi durumunda olması gereken işlemleri yaptırabilmek için karar verme deyimleri kullanılır. Matlab'da kullanılan karar verme deyimlerinden biri **if** deyimi, diğeri de **case** deyimidir..

1) İf şartlı deyimi:

Genel Kullanımı aşağıdaki gibidir:

```
if durum_1
(ifadeler_1)
elseif durum_2
(ifadeler_2)
elseif durum_3
(ifadeler_3)
...
else
(ifadeler_n)
end
```

Örnek:

0-100 aralığında girilen puanı 5 üzerinden nota çeviren bir programı if deyimi kullanarak yazalım.

Çözüm:

```
puan=input('Puanı girin :');
if puan<45 fprintf('Değeri : %d',1 );
elseif puan<55 fprintf('Değeri : %d',2 );
elseif puan<70 fprintf('Değeri : %d',3 );
elseif puan<85 fprintf('Değeri : %d',4 );
else fprintf('Değeri : %d',5 );end;
```

Örnek:

Klavyeden girilen sayının negatif, pozitif ya da sıfır olduğunu ekrana yazan bir program yazınız.

Çözüm:

```
sayi=input('Sayıyı giriniz :')
if sayi<0
fprintf('sayınız negatif. ');
elseif sayi>0
fprintf('sayınız pozitif. ');
else
fprintf('sayınız sıfır. ');end;
```

Örnek:

a, b c katsayıları girilen ikinci derece $ax^2+bx+c=0$ denkleminin reel köklerini bulup ekrana yazdıran bir program yazınız.

Çözüm:

```
clc;
a=input('a = ');b=input('b = ');c=input('c = ');
delta=b*b-4*a*c;
```

if delta>0

```
x1=(-b-delta^0.5)/(2*a);x2=(-b+delta^0.5)/(2*a);
fprintf('İki reel kök; x1 = %f x2 = %f ',x1,x2);
elseif delta==0
fprintf('Tek kök var; x1 = x2= %f ',-b/(2*a));
else
fprintf('Kökler sanal ');
end;
```

2) switch end Deyimi:

Genel Kullanımı aşağıdaki gibidir:

switch anahtar-ifade

```
case durum-1
(işlemler-1)
case durum-2
(işlemler-2)
...
case durum-n
(işlemler-n)
otherwise
(diğer işlemler)
end
```

Örnek: Klavyeden girilen 1 ile 5 arasında girilen bir tam sayının yazı ile kaç girildiğini ekrana yazdıran, istenen aralığın dışında bir sayı girilmesi durumunda **'Lütfen 1 ile 5 arasında bir tam sayı girin'** uyarısını yapan bir program yazalım.

Çözüm:

```
s=input('Sayınızı girin :');
switch s
case 1;fprintf('Bir... ');
case 2 ;fprintf('İki... ');
case 3 ;fprintf('Üç... ');
case 4 ;fprintf('Dört... ');
case 5 ;fprintf('Beş... ');
otherwise fprintf('Lütfen 1 ile 5 arasında bir tam sayı girin'); end;
```

Matlab'da Tekrarlı İşlemler ve Döngüler

Belirli bir işlem birden çok tekrar ediyorsa bunu döngü deyimleri ile gerçekleştirebiliriz. Bunlar **for** ve **while** döngüleridir.

for Döngüsü:

Amaç: Bir başlangıç değerinden, son değere kadar artış miktarı kadar aralıklarda işlemleri tekrarlamaya yarar.

Kullanımı:

```
for değişken=başlangıç_değeri:artış:son_değer
(işlemler)
end
```

Not: Artış değeri 1 ise yazılmasa da olur.

Örnek: Temiz ekrana 20 defa alt alta **İzmir Fen Lisesi** yazdıran bir program yazınız.

Matlab Ders Notları

Çözüm:

```
clc;  
for i=1:20  
fprintf('İzmir Fen Lisesi \n');end;
```

Örnek: 1 den 1000 e kadar olan sayıların toplamını buldurup sonucu ekrana yazdıran bir program yazınız.

Çözüm:

```
toplam=0;  
for i=1:1000  
toplam=toplam+i;end;  
fprintf('Toplam = %d ',toplam);
```

Örnek: Girilen bir sayıdan, istenen bir sayıya kadar olan sayıların toplamını bulduran bir program yazınız.

Çözüm:

```
toplam=0;  
ilk=input('Kaçtan itibaren :');  
son=input('Kaça kadar :');  
for i=ilk:son  
toplam=toplam+i;end;  
fprintf('Toplam = %d ',toplam);
```

Örnek: $9^2+13^2+17^2+...+2005^2$ toplamını bulduran bir program yazınız.

Çözüm:

```
toplam=0;  
for i=9:4:2005  
toplam=toplam+i*i;end;  
fprintf('Toplam = %d ',toplam);
```

while Döngüsü:

Amaç: Belirli bir durum gerçekleştikçe istenen işlemleri tekrarlamaya yarar.

Kullanımı:

```
while durum  
(işlemler)  
end;
```

Örnek: $t=1+1/2+1/3+...+1/n$ toplamı gözönüne alınıyor.

a) Baştan ilk 2005 terim toplamını bulduran bir programı while döngüsü kullanarak bulunuz.

b) t toplamının 5 i geçtiği ilk n terim sayısını ve toplamı bulduran bir program yazınız.

c) t toplamının, girilen bir x sayısını geçtiği ilk n terim sayısını ve toplamı bulduran bir program yazınız.

Çözüm:

```
a)  
clc;t=0;n=1;  
while n<=2005  
t=t+1/n;n=n+1;end;  
fprintf('İlk %d terimin toplamı %f dir.',n-1,t);
```

b)

```
clc;t=0;n=1;  
while t<=5  
t=t+1/n;n=n+1;end;  
fprintf('İlk %d terimin toplamı %f dir.',n-1,t);
```

c)

```
clc;t=0;n=1;  
x=input('x değerini giriniz : ');  
while t<=x  
t=t+1/n;n=n+1;end;  
fprintf('İlk %d terimin toplamı %f dir.',n-1,t);
```

Örnek:

Klavyeden girilen negatif sayıların toplamını ve kaç tane olduğunu, pozitif sayıların toplamını ve kaç tane olduğunu bulan sıfır girildiğinde programı sona erdirerek sonuçları temiz ekranda yazdıran bir program yazınız.

Çözüm:

```
clc;x=1;nt=0;pt=0;nsay=0;psay=0;  
while x~=0  
x=input('Sayıyı gir (bitirmek için 0) :');  
if x<0 nt=nt+x;nsay=nsay+1;  
elseif x>0 pt=pt+x;psay=psay+1;end;end;  
fprintf('%d tane negatif sayının toplamı %f ',nsay,nt);  
fprintf('%d tane pozitif sayının toplamı %f dir.',psay,pt);
```

MATEMATİKSEL İŞLEMLER

Matlab'daki matematiksel işlemler ve anlamları aşağıda verilmiştir.

İşlem veya fonksiyon	Anlamı
+	Toplama sembolü
-	Çıkarma sembolü
*	Çarpma sembolü
/	Bölme sembolü
^	Üs alma sembolü

Örnek:

Komut satırına **12+32/(4-2)^3*5** yazıp enter tuşuna bastığımızda; Matlab önce parantez içini yapar (yani $4-2=2$) sonra 2 nin 3. kuvveti alınır 8 bulunur, 32 8 e bölünür (4), bu da 5 ile çarpılır (20), 20 ile 12 toplanarak **32** elde edilir.

TEMEL MATEMATİK FONKSİYONLAR

Matlab'da bazı matematiksel işlemler yaptırmak istediğimizde, **matematiksel fonksiyonları** kullanırız.

Matlab'da matematiksel fonksiyonlar;

a) Temel (**elementary**) fonksiyonlar; **elfun**

b) Özel (**special**) fonksiyonlar; **specfun**

c) Veri (**data**) fonksiyonları; **datafun**

d) Metin (karakter dizisi) (**string**) fonksiyonlar; **strfun**

e) Dosya giriş-çıkış (**input-output**) fonksiyonları **iofun**

f) Tarih-zaman (**time**) fonksiyonları; **timefun** araç kutusundadır. Bir araç kutsundaki fonksiyonları ve anlamlarını görmek için komut satırına;

help araç_kutusu biçiminde yazarız.

Örneğin; temel fonksiyonlar ve anlamlarını görüntülemek için; **help elfun**, metin fonksiyonlarını ve anlamlarını görüntülemek için **help strfun** yazılır.

Bunların bazılarını görelim:

a) Temel (**elementary**) fonksiyonlar; **elfun**

Matlab Ders Notları

Varolan tanımlı fonksiyonları görmek için komut satırına **help elfun** yazarız. Bunlardan bazıları ve anlamları aşağıda verilmiştir.

İşlem veya fonksiyon	Anlamı
+	Toplama sembolü
-	Çıkarma sembolü
*	Çarpma sembolü
/	Bölme sembolü
^	Üs alma sembolü
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 tanjantı
cot(x)	Radyan cinsinden x in cotanjantı
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)
sqrt(x)	\sqrt{x}
fix(x)	x in yukarıya yuvarlanmış
ceil(x)	x in aşağıya yuvarlanmış
floor(x)	x in tamdeğeri, $\lfloor x \rfloor$
sign(x)	x in işareti, sgn(x)
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

b) Özel (**special**) fonksiyonlar; **specfun**
Bunlardan bazıları ve anlamları aşağıda verilmiştir.

İşlem veya fonksiyon	Anlamı
cross(a,b)	a ile b vektörünün vektörel çarpımı
dot(a,b)	a ile b vektörünün skaler çarpımı
factor(n)	n sayısının 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 faktöryel (n!)
nchoosek(n,r)	n nin r li kombinasyonlarının sayısı

c) Veri (**data**) fonksiyonları; **datafun**

İşlem veya fonksiyon	Anlamı
max(a)	a dizisinin en büyük elemanını bulur.

min(a)	a dizisinin en küçük elemanını bulur.
mean(a)	a dizisinin ortalamasını bulur.
median(a)	a dizisinin orta terimini bulur.
std(a)	a dizisinin standart sapmasını bulur.
var(a)	a dizisinin varyansını bulur.
sort(a)	a dizisini artan olarak sıralar
sortrows(a)	a matrisinin satırlarını artan olarak sıralar
sum(a)	a dizisinin elemanlarını toplar
prod(a)	a dizisinin elemanlarını çarpar

d) Metin (karakter dizisi) (**string**) fonksiyonlardan bazıları:

İşlem veya fonksiyon	Anlamı
char(a)	sayısal a dizisini, karakter dizisine dönüştür
double(a)	karakterlerden oluşan a dizisini, sayı dizisine dönüştür
eval(a)	a metnini Matlab ifadesi olarak tanımlar
findstr(a,b)	A ve b metinlerinden kısa olanını uzun olanı içinde arayarak metnin başlangıç değerini bulur
strfind(a,b)	a metni içinde b metnin başlangıç değerini bulur
upper(a)	a stringinin (metnin) harflerinin tümünü büyük harf yapar.
lower(a)	a stringinin (metnin) harflerinin tümünü küçük harf yapar.
num2str	Sayıardan oluşan değeri stringe (metne) dönüştür.
st2num	Rakamlardan oluşan stringi sayıya dönüştür.

Örnek:

1. **a='İzmir';findstr(a,'mi')** komutu sonucu ekranda 3 sayısı görülür.

2. Benzer biçimde **a='İzmir';strfind(a,'mi')** komutu sonucu ekranda 3 sayısı görülür.

3. **findstr('izmir fen lisesi','fen')** komutu sonucu ekranda 7 sayısı görülür.

Not: findstr('izmir fen lisesi','fen') ile findstr('fen','izmir fen lisesi') komutu arasında hiçbir fark yoktur.

4. **upper('izmir')** komutu sonucu ekranda İZMİR kelimesi; **lower('NABER?')** komutu sonucu da ekranda; naber? kelimesi görülür.

Matlab Ders Notları

Matlab'da Sembolik Matematik ve Uygulamaları:

Matlab'da bir denklemin çözümünü bulmak için örneğin $x^2-2x-15=0$ denkleminin çözümünü bir m dosyasına gerekli kodları yazarak yapabiliriz. Bu programın m dosyası aşağıdaki gibi olabilir.

```
clc;
a=input('a = ');b=input('b = ');c=input('c = ');
delta=b*b-4*a*c;
if delta>0
x1=(-b-delta^0.5)/(2*a);x2=(-b+delta^0.5)/(2*a);
fprintf('İki reel kök; x1 = %f x2 = %f ',x1,x2);
elseif delta==0
fprintf('Tek kök var; x1 = x2= %f ',-b/(2*a));
else
fprintf('Kökler sanal ');
end;
```

Program çalıştırıldığında a b c katsayılarına sırasıyla, 1, -2 ve -15 değerlerini girerek denklemin köklerini -3 ve 5 olarak buluruz.

İkinci derece bir denklemin çözümünü veren formüllerini bildiğimizden bunun programını (çok kolay olmasa da) yazabildik. Ya denklem üçüncü dereceden olursa, 4 veya 5. dereceden olursa, ya da $x \cdot \sin x = 1/5$

gibi veya $x^x = 64$ gibi olursa ... bu denklemleri nasıl çözümleriz?

Verilen bir cebirsel ifadeyi, mesela

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

gibi bir ifadeyi sadeleştirebilir

miyiz?

Bir fonksiyonun limitini, türevini ve integralini Matlab'da buldurabilir miyiz?

İşte bu gibi işlemleri yaptırabilmek için Matlab'daki Sembolik Mantık (Sembolik Nesne) kavramını kullanmalıyız. Bu konu ile ilgili açıklama ve yardım almak için komut satırına **help symbolic** yazmak yeterlidir.

Sembolik Matematikte Bazı Komutlar ve Anlamları:

sym ve syms komutları:

Bir değişkeni sembolik nesne yapmaya yarar.

Örneğin x değişkenini sembolik nesne yapmak için; **x=sym('x');** komutu kullanılabilir. Aynı işlemi **syms x;** komutu ile de yapabiliriz.

x, y ve z değişkenlerini sembolik nesne yapmak için; **syms x y z** komutu kullanılabilir.

Bir ondalık sayının kesir olarak karşılığını bulabilmek için sym komutundan faydalanabiliriz.

Örneğin 3.98 ondalık sayısının rasyonel sayı karşılığını bulmak için sym(3.98,'r') veya sym(3.98) komutu kullanılabilir.

Benzer şekilde 22/7 kesrini ondalık sayıya çevirmek için de sym(22/7,'d') komutu kullanılır.

Bir sayısal kesri sadeleştirmek için de sym komutundan faydalanabiliriz.

Örneğin, $\frac{120}{20}$, $\frac{120}{25}$ ve $\frac{2005^3 \cdot 2006}{4032085075}$ kesirlerinin en sade biçimini bulalım.

sym(120/20) yazıp enter tuşuna bastığımızda ekranda kesrin en sade sonucu olan 5 sayısını,

sym(120/25) yazıp enter tuşuna bastığımızda ekranda kesrin en sade sonucu olan 24/5 sayısını,

sym((2005^3*2006)/4032085075) yazıp enter tuşuna bastığımızda ekranda kesrin en sade sonucu olan 4010 sayısını görürüz.

Değişkenlerin sembolik nesne olarak tanımlanabildiği gibi, fonksiyonlar da tanımlanabilir.

Örneğin;

$y = x^3 - 3x^2 + \sin x$ fonksiyonunu sembolik nesne olarak

tanımlamak istersek;

bunu iki yoldan da yapabiliriz:

a) syms x;

$y = x^3 - 3x^2 + \sin(x)$ komutlarıyla veya;

b) $y = \text{sym}('x^3 - 3x^2 + \sin(x)')$ komutuyla yapabiliriz.

pretty komutu:

Sembolik nesnenin görüntüsünü ekranda net olarak anlaşılır biçimde görünmesini sağlayan komuttur.

Örneğin yukarıdaki y fonksiyonunu ekranda anlaşılır biçimde görüntülemek için **pretty(y)** yazmak yeterlidir. Bu durumda ekranda; $y = x^3 - 3x^2 + \sin(x)$ ifadesi görülür.

Harfli İfadeler ve Sadeleştirilmesi:

simplify Komutu

Sembolik nesneleri sadeleştirmeye yarar.

Örneğin;

a) $8 - 5\sin^2x - 5\cos^2x$ trigonometrik ifadesini ve

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

c) $\frac{2x-1}{x+3} + \frac{x+2}{x-1}$ ifadesinin sonucunu en sade biçimde yazmak isteyelim.

Çözüm:

a) Bunun için önce ifadeyi sembolik nesneye dönüştürüp sonra basitleştirme komutunu kullanmalıyız. Bunu iki şekilde yapabiliriz:

I) $y = \text{sym}('8 - 5*\sin(x)^2 - 5*\cos(x)^2');$

simplify(y)

II) simplify(sym('8-5*sin(x)^2-5*cos(x)^2'))

b) Benzer şekilde

$y = \text{sym}('(x^3-8)*(x^2+7*x)/((2*x^2+4*x+8)*(x^2-2*x))');$

simplify(y)

komutları uygulanırsa ekranda ifadenin en sade hali olan;

$1/2*x+7/2$ ifadesi bulunur. Şayet simplify(y) yerine;

Matlab Ders Notları

pretty(simplify(y)) komutu uygulanırsa ekranda daha anlaşılır olan; $1/2 x + 7/2$ ifadesi görülür.

c) $y = \text{sym}('(2*x-1)/(x+3)+(x+2)/(x-1)')$;

pretty(simplify(y)) komutları uygulanırsa ekranda işlemin sonucu olan;

$$\frac{3x^2 + 2x + 7}{(x+3)(x-1)}$$

ifadesi görülür.

expand ve factor komutları:

Sembolik nesnelerden oluşan polinomların kuvvetini açmak için expand komutunu, bir polinomun kuvveti olarak açılmış ifadeyi eski haline getirmek için factor komutunu kullanırız.

Örnek:

a) $(x^2-x+1)^3 - (x^2+x-1)^3$ açılımının sonucunu;

b)

$$-1 + 3x - 3x^2 + 6x^3 + x^4 + 10x^5 - 6x^6 - 10x^7 - 12x^8 + 12x^9$$

ifadesi bir polinomun kuvveti olduğuna göre hangi polinomun kaçınıcı kuvveti olduğunu bulduralım.

Çözüm:

a) $y = \text{sym}('(x^2-x+1)^3-(x^2+x-1)^3')$; pretty(expand(y)) işlemin sonucu ekranda;

$$-6x^5 + 6x^4 - 2x^3 + 6x^2 - 6x + 2 \text{ görülür.}$$

b) $z = \text{sym}('1+3*x-3*x^2+6*x^3+x^4+10*x^5-6*x^6-10*x^7-12*x^8+12*x^9')$; pretty(factor(z)) işlemin sonucu ekranda;

$$(x-1)(x+1)^3 \text{ görülür.}$$

subs komutu:

Bir sembolik ifadenin değişkenine verilen bir değer için sonucunu bulmaya yarar.

Örnek:

$$y = f(x) = \frac{x^2 - 2x + 7}{3x+5} \text{ fonksiyonu veriliyor.}$$

a) f(2005) değerini,

$$\frac{x^3 - 2}{x+3}$$

b) $z = g(x) = \frac{x^3 - 2}{x+3}$ olmak üzere f(g(x)) bileşke

fonksiyonunu bulduralım.

Çözüm:

a) $y = \text{sym}('(x^2-2*x+7)/(3*x+5)')$; subs(y,x,2005) işlemin sonucu ekranda 667.1133 değeri görülür.

b) $y = \text{sym}('(x^2-2*x+7)/(3*x+5)')$; subs(y,x,2005); $z = \text{sym}('(x^3-2)/(x+3)')$; pretty(subs(y,x,z)) işlemlerinin sonucu ekranda aşağıdaki ifade görülür.

$$\frac{(x^3 - 2)(x^2 - 2x + 7)}{(x+3)^3} = \frac{x^5 - 2x^4 - 2x^3 + 14x^2 - 14x + 14}{(x+3)^3}$$

Ancak bu ifadeyi gerekli işlemlerin yapılarak daha da basit biçime getirmek için en sondaki ifade olan pretty(subs(y,x,z)) yerine pretty(simplify(subs(y,x,z))) yazmalıyız. Bu durumda ekranda;

$$\frac{6x^6 - 10x^5 + 79x^4 - 2x^3 + 46x^2 + 7x}{(3x^3 + 9x^2 + 5x)(x+3)}$$

ifadesi görülür.

DENKLEMLER ve Çözümleri:

solve komut:

Matlab'ın en güçlü ve kullanışlı komutlarından biridir. Kısaca verilen her türden denklem (sayısal veya matematik nesnesi olan) veya denklem sistemlerini çözümlenmeye yarar.

Bir Bilinmeyenli Denklem Çözümleri:

Örnek;

a) $2x-6=0$

b) $2x^2+5x=3$

c) $ax^2+bx+c=0$

d) $x^3 + 3x^2 - x - 3 = 0$

e) $x^5 = 16x$

f) $x^2 - 6x - 3 = \sqrt{3x - 5}$

g) $x^x = 64$ denklemlerinin çözüm kümelerini bulalım.

Çözüm:

a) solve(2*x-6) veya solve('2*x-6') veya solve(sym('2*x-6')) işleminin sonucu ekranda; 3 görülür.

b) solve(2*x^2+5*x-3) veya solve('2*x^2+5*x-3') işleminin sonucu ekranda; -3 ve 1/2 görülür.

c) Burada harfli ifadeyi matematiksel nesne olarak yazmak zorundayız. Bunun için komutu solve('a*x^2+b*x+c') veya solve(sym('a*x^2+b*x+c')) biçiminde kullanmalıyız. Bu durumda ekranda; çözüm kümesi;

$$\left[\frac{1}{2}a \cdot (-b + (b^2 - 4ac)^{1/2}) \right]$$

$$\left[\frac{1}{2}a \cdot (-b - (b^2 - 4ac)^{1/2}) \right] \text{ biçiminde görülür.}$$

Matlab Ders Notları

Bunu daha düzenli görüntülemek için pretty komutundan faydalanalım.Yani komutu pretty(solve('a*x^2+b*x+c')) olarak uygularsak ekranda;

$$\begin{bmatrix} 2 & 1/2 \\ -b + (b^2 - 4ac) & \\ 1/2 & \text{-----} \\ a & \end{bmatrix}$$

ifadesi görülür ki bu da, $ax^2+bx+c=0$ ikinci derece denklemin kökler formülünden başka bir şey değildir.

d) solve(x^3+3*x^2-x-3) komutu uygulanırsa ekranda; 1, -3, -1 değerleri görülür.

e) solve(x^5-16*x) komutu uygulanırsa ekranda;

[0]
[2]
[-2]
[2*i]
[-2*i] değerleri görülür.Görülüyor ki denklemin 3 tane reel iki tane de sanal kökleri var.

f) solve(x^2-6*x-3-(3*x-5)^(1/2)) komutu uygulanırsa ekranda; 7 sayısı görülür.

g) solve(x^x-64) komutu uygulanırsa ekranda;

ans =

log(64)/lambertw(log(64))
sonucu görülür ki, bu sonuç bilmediğimiz bir fonksiyonun ürettiği bir değerdir.Bu değeri sayısal değere çevirmek için **double** komutundan yararlanırız.Yani komutu;
double(solve(x^x-64)) biçiminde kullanırsak, ekranda 3.3991 değerini görürüz.

Çok Bilinmeyenli Denklem Çözümleri:

Örnek: Aşağıdaki denklem sistemlerinin çözüm kümelerini bulalım:

- a) $2x-3y=27$
 $5x+2y=1$
b) $17x-3y+4z=7$
 $15x-7y=1$
 $x+y-9z=13$
c) $x^2-2xy+3y^2=17$
 $xy-3x+5=0$

Çözüm:

a) Önce x ve y değişkenlerini sembolik değişken olarak tanımlamalıyız.Yani **syms x y** komutu uygulanmalıdır.Sonra da çözüm sonucu bir değikene örneğin sonuc değişkenine
sonuc=solve(2*x-3*y-27,5*x+2*y-1) komutuyla atanmalıdır.Bu durumda x değeri sonuc.x değişkeninde

y değeri de sonuc.y değişkeninde bulunacaktır.Bunlar ekrana yazılarak sonuçlar görülebilir.O halde özetle, bu denklem sisteminin çözümü için ekrana şunlar sırasıyla yazılmalıdır:

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

sonuc.x
sonuc.y

b) **syms x y z;**
sonuc=solve(17*x-3*y+4*z-7,15*x-7*y-1,x+y-9*z-13);
sonuc.x
sonuc.y
sonuc.z

c) **syms x y;**
sonuc=solve(x^2-2*x*y+3*y^2-17,x*y-3*x+5);
sonuc.x
sonuc.y

LİMİT ve UYGULAMALARI:

limit Komutu:

Sembolik nesnelerden oluşan ifadenin limitini bulmaya yarar.

$\lim_{x \rightarrow a} f(x)$ matematiksel ifadenin Matlab karşılığı;

limit(f,x,a) biçimindedir.

Şayet limit soldan veya sağdan olursa, Matematiksel ve Matlab karşılıkları aşağıdaki gibi bulunur.

lim

$x \rightarrow a^- f(x)$ için **limit(f,x,a,'left')**

lim

$x \rightarrow a^+ f(x)$ için **limit(f,x,a,'right')**

Not:

- 1) Şayet a değeri belirtilmezse 0 için limit bulunur.
- 2) ∞ için limit bulunacaksa a yerine **inf** ifadesi yazılır.
- 3) $-\infty$ için limit bulunacaksa a yerine **inf** ifadesi ve **'left'** ifadesi yazılır.

Örnek:

a) $\lim_{x \rightarrow 1} \frac{3x^2 - 3}{x - 1}$

b) $\lim_{x \rightarrow 0} \frac{2-2\cos x}{x.\sin x}$

c) $\lim_{n \rightarrow \infty} \frac{7n^2-13n+777}{19-3n-n^2}$

d) $\lim_{n \rightarrow \infty} \sqrt{n^2+10n-2005} - \sqrt{n^2-8n+2006}$

e) $\lim_{n \rightarrow \infty} \frac{7^{2n+1} - 49.7^{n+3}}{49^{n-1} + 77}$

f) $\lim_{n \rightarrow \infty} \left(\frac{5n+8}{5n+7} \right)^{10n+9}$

Matlab Ders Notları

g) $\lim_{x \rightarrow -\infty} \frac{x - \sqrt{x^2 + x + 1}}{2x - \sqrt{4x^2 + x}}$

h) $\lim_{x \rightarrow 0^-} \frac{9x}{|x|}$

Çözüm:

a) syms x;
limit((3*x^2-3)/(x-1),x,1) işleminin sonucunda limit 6 olarak bulunur.

b) syms x;
limit((2-2*cos(x))/(x*sin(x)),x,0) veya
limit((2-2*cos(x))/(x*sin(x)))

c) syms n; limit((7*n^2-13*n+77)/(19-3*n-n^2),n,inf)
d) syms n;
limit((n^2+10*n-2005)^(1/2)-(n^2-8*n+2006)^(1/2),n,inf)

e) syms n;
limit((7^(2*n+1)-49*7^(n+3))/(49^(n-1)+77),n,inf)

f) syms n;
limit(((5*n+8)/(5*n+7))^(10*n+9),n,inf)

g) syms x;
limit((x-(x^2+x+1)^(1/2))/(2*x-(4*x^2+x)^(1/2)),x,inf,'left')

h) syms x; limit((9*x)/(abs(x)),x,0,'left')

DİZİLER ve SERİLER ile İlgili Uygulamalar:

symsum komutu:

Toplam sembolü veya seri toplamını bulmaya yarayan komuttur.

b
 $\sum_{k=a}^b f(k)$ toplamını bulmaya yarayan Matlab komutu;
k=a

symsum(f(k),a,b) biçiminde kullanılır.

Örnek:

a) 1+2+3+...+n toplamının formülünü bulduran komutu yazalım.

b) 1²+2²+3²+...+n² toplamının formülünü bulduran komutu yazalım.

c) 4.5.6+5.6.7+6.7.8+...+22.23.24 toplamının sonucunu bulduran komutu yazalım.

d) (2/3)³+(2/3)⁴+(2/3)⁵+...+(2/3)⁹⁹ toplamını bulalım.

e) (2/3)³+(2/3)⁴+(2/3)⁵+... serisinin toplamını bulalım.

f) $\frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \dots$ serisinin sonucu $\frac{\pi^2}{6}$ olduğu bilindiğine göre π sayısını bulunuz.

Çözüm:

a) syms k n; symsum(k,1,n)
komutu sonucunda; 1/2*(n+1)^2-1/2*n-1/2 ifadesi bulunur.

Şayet sonucu daha basit bulmak istersek;
syms k n; simplify(symsum(k,1,n)) bunun sonucunda 1/2*n^2+1/2*n ifadesi bulunur.

syms k n; pretty(simple(symsum(k,1,n))) komutunu uygularsak ekranda 1/2 n (n + 1) sonucu görülür.

b) syms k n; pretty(simple(symsum(k^2,1,n)))

c) syms k n; symsum(k*(k+1)*(k+2),4,22)

d) syms k; symsum((2/3)^k,3,99) komutu sonucu ekranda

152704450587262615335745290072695420044661986
328/17179250691067044367882037658854042423403
5840667

sembolik ifadesi görülür. Bu değer sayısal değerini bulmak için **double(ans)** kullanılırsa 0.8889 gerçek değeri bulunur. Aynı şeyi;

syms k; double(symsum((2/3)^k,3,99))

biçiminde de bulabiliriz.

e)

syms k; symsum((2/3)^k,3,inf) komutu sonucu 8/9 sonucu bulunur.

f) syms n; (6*double(symsum(1/n^2,1,inf)))^(1/2)

TÜREV ve İlgili Uygulamalar:

diff komutu:

Tanımlı sembolik ifadenin türevini bulmaya yarar.

Örnek:

a) $y = x^3 + 6x^2 - 13x + 19$ fonksiyonunun türevini bulalım.

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

c) $y = \sin^2 x \cdot \cos x$ fonksiyonunun türevini bulalım.

d) $z = x^2 y + 3xy - y^2$ fonksiyonunun

i) x e göre türevini

ii) y ye göre türevini

iii) $y' = dy/dx$ türevini bulalım.

Çözüm:

a) syms x; y=sym('x^3+6*x^2-13*x+19'); diff(y)
veya kısaca; diff('x^3+6*x^2-13*x+19')

b) diff('(x^2-3*x+7)/(x^2+5*x-1)') komutu uygulandığında ekranda;

$(2*x-3)/(x^2+5*x-1) - (x^2-3*x+7)/(x^2+5*x-1)^2*(2*x+5)$ ifadesi görülür. Bunu daha anlaşılır biçimde görüntülemek için;

pretty(diff('(x^2-3*x+7)/(x^2+5*x-1)'))

komutunu uygulamalıyız. Bu durumda ekranda;

$$\frac{2x-3}{x^2+5x-1} - \frac{(x^2-3x+7)(2x+5)}{(x^2+5x-1)^2}$$

Bu sonucu daha sade halde görüntülemek için ise;
pretty(simplify(diff('(x^2-3*x+7)/(x^2+5*x-1)')))

komutunu uygulamalıyız. Bu durumda ekranda;

$$\frac{x-2x-4}{(x^2+5x-1)^2}$$

ifadesi görülür.

Matlab Ders Notları

c) pretty(diff('sin(x)^2*x*cos(x)')) komutu sonucu ekranda;

$$2 \sin(x) x \cos(x) + \sin(x)^2 \cos(x) - \sin(x)^3 x$$
 görülür.

d) syms x y; z='x^2*y-3*x*y-y^2';

i) tx=diff(z,x)

ii) ty=diff(z,y)

iii) pretty(-tx/ty)

İNTEGRAL ve İlgili Uygulamaları:

int Komutu:

Tanımlı sembolik ifadenin belirsiz integralini bulmaya yarar.

Örnek:

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

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

c) $\int x^2 \sin x dx$ belirsiz integralini bulalım.

Çözüm:

a) int('3*x^2-2*x+5')

b) pretty(int('(2*x+5)/(x^2+1)'))

c) pretty(int('x^2*sin(x)'))

Matlab'da Karmaşık Sayılarla İşlemler:

Matlab'da bir değişkeni karmaşık sayı olarak atamak için; z=a+bi gibi bir eşitlik gerekir. Örneğin, **z=3-4i** eşitliği ile z değişkenine 3-4i karmaşık sayısı atanmış olur.

Bir karmaşık sayı ile ilgili aşağıdaki işlemler yaptırılabilir.

Fonksiyon	Açıklama
conj(z)	z nin eşleniğini verir
real(z)	z nin reel kısmını verir
imag(z)	z nin sanal kısmını verir
abs(z)	z nin mutlak değerini verir
angle(z)	z nin x eksenine yaptığı açıyı radyan olarak verir
isreal(z)	z nin reel sayı olup olmadığını sorgular

Örnek: z1=3+4i; z2=12-5i karmaşık sayıları için;

a) Ekrana sayıları yazdıran,

b) Toplamlarını

c) 3z1-5z2 sayısını

d) Mutlak değerlerini,

e) Eşleniklerini,

f) Reel ve sanal kısımlarını buldurup ekrana yazdıran bir program yazınız.

Çözüm:

clc; z1=3+4i; z2=12-5i;

disp('z1='); disp(z1);

disp('z2='); disp(z2);

disp('z1+z2 = '); disp(z1+z2);

```
disp('3z1-5z2 = '); disp(3*z1-5*z2);
disp('|z1|='); disp(abs(z1));
disp('|z2|='); disp(abs(z2));
disp('z1 in eşleniği'); disp(conj(z1));
disp('z1 in reel kısmı'); disp(real(z1));
disp('z1 in sanal kısmı'); disp(imag(z1));
disp('z2 nin eşleniği'); disp(conj(z2));
disp('z2 nin reel kısmı'); disp(real(z2));
disp('z2 in sanal kısmı'); disp(imag(z2));
```

Matlab'da Grafik İşlemleri:

1) İki Boyutlu Grafikler (Düzlemde Grafik):

Bu konuyla ilgili komut ve açıklamaları görmek için komut satırına **help graph2d** yazdığımızda aşağıdaki bilgiler gelir.

Two dimensional graphs.

Elementary X-Y graphs.

plot - Linear plot.

loglog - Log-log scale plot.

semilogx - Semi-log scale plot.

semilogy - Semi-log scale plot.

polar - Polar coordinate plot.

plotyy - Graphs with y tick labels on the left and

.....

.....

Biz bu komutlardan bazılarını göreceğiz.

a) plot komutu:

Matlab'da plot komutuyla grafik çizdirmek için, tanımlanan fonksiyonun x değişkeninin başlangıç ve bitiş değerleri arasındaki her değer için ayrı ayrı hesaplatılan grafiğe ait (x,y) noktalarının koordinat düzleminde nokta ile işaretletmeliyiz.

Kullanımı:

değişken_adı=ilk_değeri:artış_miktari:son_değeri;
fonksiyon_değişkeni=fonksiyon_tanımı;
plot(x,y);

Not 1) **plot(x,y)** komutu yerine iki noktayı doğru ile birleştiren komut olan **line(x,y)** komutunu da kullanabiliriz.

Not 2) plot komutuyla çizdirilen grafiğe ait **çizgi özelliklerini** de belirtebiliriz. **Bunu plot(x,y,'çizgi_özellikleri',...); veya plot(x,y,'özellik1',değer1,'özellik2',değer2,...); biçiminde** belirtiriz.

Buradaki çizgi özellikleri ve değerleri şunlardır:

Color: line nesnesinin rengini düzenlemeye yarar.

LineStyle: Çizgi stilini belirleyen özelliktir. Alabileceği değerler; -, --, -., :, ve **none** dir. Değeri none olursa çizgi görünmez.

LineWidth: Çizginin kalınlığını düzenleyen özelliktir.

Marker: Çizginin işaretini belirleyen özelliktir. Marker ile ilgili değerler ve anlamı aşağıdaki tabloda gösterilmiştir.

Matlab Ders Notları

Değer	Anlamı
+	+ işareti
o	daire işareti
*	yıldız işareti
.	nokta işareti
x	çarpı işareti
s	kare (square) işareti
d	elmas (diamond) işareti
^	yukarı gösteren üçgen işareti
v	aşağı gösteren üçgen işareti
>	sağa doğru gösteren üçgen işareti
<	sola doğru gösteren üçgen işareti
p	5 noktalı (pentagon) yıldız işareti
h	6 noktalı (hexagram) yıldız işareti
none	işaretsiz

Not 3) Aynı x değerlerine karşılık birden fazla fonksiyon tanımlanarak (y1,y2,y3, ... gibi) aynı grafik ekseninde çizdirebiliriz.Bunu da;
plot(x,y1,x,y2,x,y3,...); biçiminde belirtiriz.

Örnek:

x=-10, x=10 aralığında, 0.1 artışla ,y=2x-6 doğrusunun grafiğini çizdiren bir program yazınız.

Çözüm:

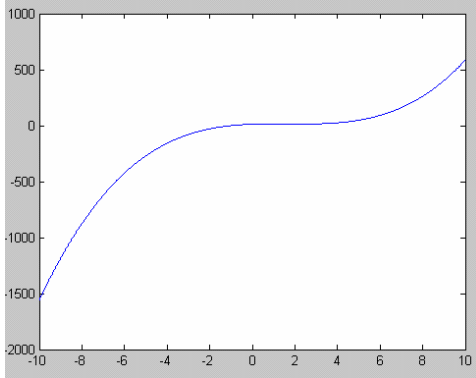
x=-10:0.1:10;
y=2*x-6;plot(x,y);

Örnek:

x=-10, x=10 aralığında, 0.01 artışla,
 $y=x^3 - 5x^2 + 7x + 13$ fonksiyonunun grafiğini çizdiren bir program yazınız.

Çözüm:

x=-10:0.01:10;y=x.^3-5*x.^2+7*x+13;plot(x,y);
Grafik aşağıdaki gibidir:



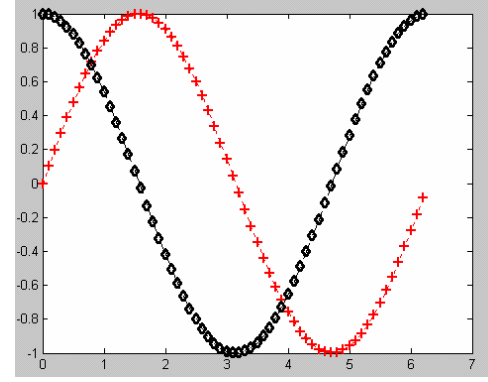
Örnek:

x değerleri (tanım kümesi) $[0, 2\pi]$ aralığı olan $f(x)=\sin x$ ile $g(x)=\cos x$ fonksiyonlarının grafiklerini aynı koordinat düzleminde, tek komutla çizdirelim.Öyle ki;
a) f(x) in rengi **kırmızı**, g(x)in rengi **mavi** olsun
b) f(x) in rengi **kırmızı**, çizgi stili :, noktaların biçimi +, g(x) in rengi **siyah**, çizgi stili --, noktaların biçimi **elmas** ve çizgi kalınlıkları **2** şer birim olsun.

Çözüm:

x=0:0.1:2*pi;f=sin(x);g=cos(x);
a) plot(x,f,'r+',x,g,'b--',2);

b) plot(x,f,'r+',x,g,'black--d','linewidth',2);
Grafiği aşağıda verilmiştir:

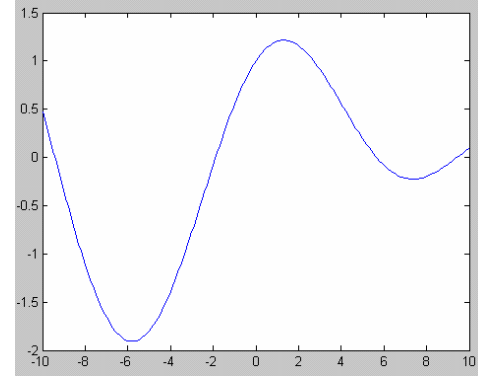


Örnek:

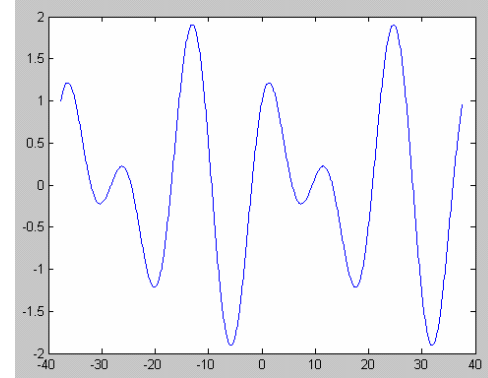
$y=\sin(x/3)+\cos(x/2)$ fonksiyonunun grafiğini;x değerleri 0.1 artışla;
a) $[-10, 10]$ aralığında;
b) Fonksiyonun periyodu T ise, $[-T, T]$ aralığında grafiğini çizdirelim;

Çözüm:

a) x=-10:0.1:10;y=sin(x./3)+cos(x./2); plot(x,y);
Grafiği aşağıdaki gibidir.



b) Fonksiyonun periyodu; $T=\text{OKEK}(6\pi;4\pi)=12\pi$ dir.Buna göre komut satırına aşağıdaki ifadeleri yazmalıyız:
x=-12*pi:0.1:12*pi;y=sin(x./3)+cos(x./2);plot(x,y);
Grafiği aşağıdaki gibidir.



b) loglog, semilogx, semilogy komutları:

1) Bir fonksiyonun grafiğini çizdirdiğimizde x ve y nin aralığı çok geniş olduğunda hem x değerlerini, hem de y değerlerini logaritmik artışla tanımlayabiliriz.Bu durumda grafiği **loglog(x,y)** komutunu kullanırız.

2) x değerleri, y değerlerine göre çok geniş bir aralıkta ise sadece x değeri aralığını logaritmik artışla

Matlab Ders Notları

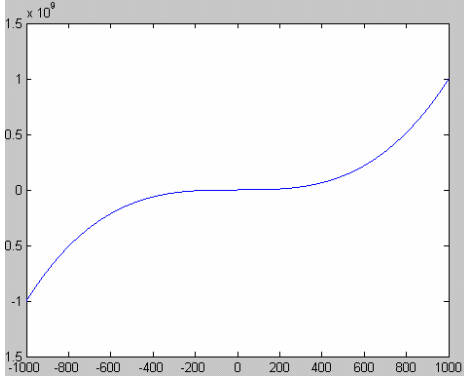
tanımlayarak grafiği çizdirebiliriz. Bu durumda **semilogx(x,y)** komutunu kullanırız.

3) y değerleri, x değerlerine göre çok geniş bir aralıkta ise sadece y değer aralığını logaritmik artışla tanımlayarak grafiği çizdirebiliriz. Bu durumda **semilogy(x,y)** komutunu kullanırız.

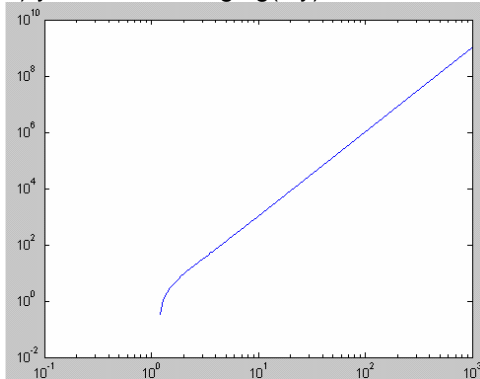
Örnek: x değerleri -1000 ile 1000 arasında olmak üzere $y = x^3 + 3x - 5$ fonksiyonun grafiğini a) Normal b) x ve y değerleri logaritmik artışla c) Sadece x değerleri logaritmik artışla d) Sadece y değerleri logaritmik artışla çizdirelim.

Çözüm: $x = -1000:0.1:1000$;

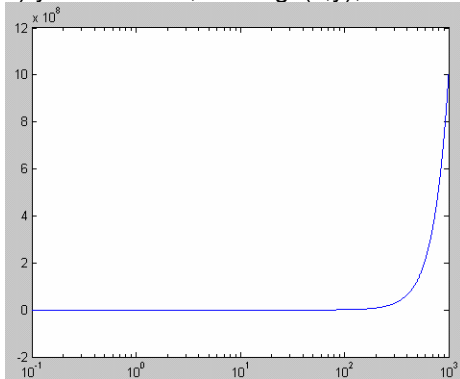
a) $y = x.^3 + 3*x - 5$; `plot(x,y)`;



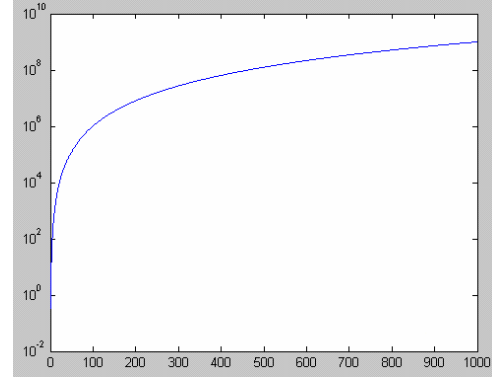
b) $y = x.^3 + 3*x - 5$; `loglog(x,y)`;



c) $y = x.^3 + 3*x - 5$; `semilogx(x,y)`;



c) $y = x.^3 + 3*x - 5$; `semilogy(x,y)`;



hold Fonksiyonu ve Kullanımı:

Bazen aynı ekseninde iki grafik üstüste çizdirerek iki grafiğin birbirine göre durumlarını incelemek isteyebiliriz. İşte bu durumda **hold** fonksiyonu kullanılabilir.

Örnek:

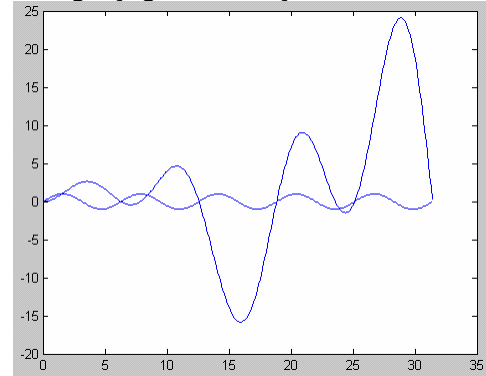
$[0, 10\pi]$ aralığında; $f(x) = \sin(x)$ fonksiyonu ile $g(x) = x \sin(x/2) \cos(x/5)$ fonksiyonunun grafiklerini aynı koordinat düzleminde çizdirelim.

Çözüm:

$x = 0:0.1:10*\pi$; $y1 = \sin(x)$; $y2 = x.*\sin(x./2).*\cos(x./5)$;

`plot(x,y1); hold; plot(x,y2)`;

Grafiği aşağıda verilmiştir.



c) plotyy Fonksiyonu:

Bazen sayısal aralıkları farklı iki fonksiyonu aynı eksen üzerinde görüntülediğimizde, birinin aldığı değerler, diğerine nazaran çok küçük olduğundan tam olarak ayırtedilemez. Bu durumda iki grafiği **plotyy** komutuyla çizdirdiğimizde grafiği daha ayrıntılı ve net görebiliriz.

Kullanımı:

`plotyy(ortak_aralık,fonk1,ortak_aralık,fonk2)`;

Örnek:

$[0, 6\pi]$ aralığında; $f(x) = 2x^2 - 10x + 5$ ile $g(x) = \cos(x/3)$ fonksiyonlarını aynı grafik ekseninde;

a) Normal olarak b) $g(x)$ fonksiyonunu daha belirgin olarak çizdirelim.

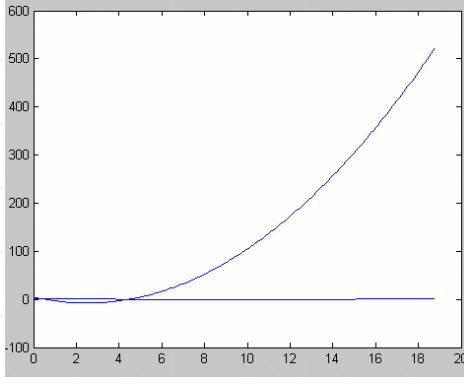
Çözüm:

$x = 0:0.1:6*\pi$; $f = 2.*x.^2 - 10*x + 5$; $g = \cos(x./3)$;

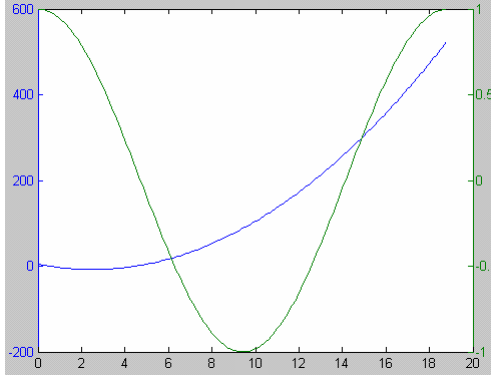
a) `plot(x,f); hold; plot(x,g)`;

Grafik aşağıda verilmiştir.

Matlab Ders Notları



b) `plotyy(x,f,x,g);`
Grafik aşağıda verilmiştir.



d) polar Fonksiyonu:

Kutupsal koordinatlarla verilen bir fonksiyonun grafiğini çizmeye yarar.

Kullanımı:

polar(t,r);

Not: Komuttaki t açısı grafiğe ait noktaya karşılık gelen vektörün Ox eksenine yaptığı açısı, r de bu vektörün uzunluğunu belirtir.

Örnek: t açısının değer aralığı $[0, 10\pi]$ olmak üzere;

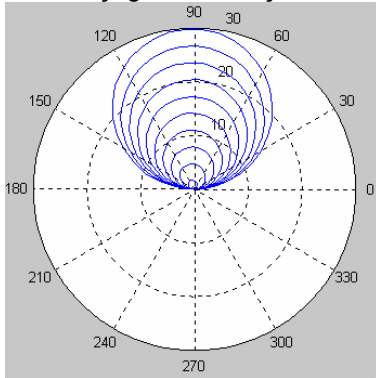
a) $r_1 = \sin(t)$ b) $r_2 = t \sin(t) \cos(t)$ fonksiyonlarının grafiklerini çizdirelim.

Çözüm:

$t = 0:0.1:10*\pi$;

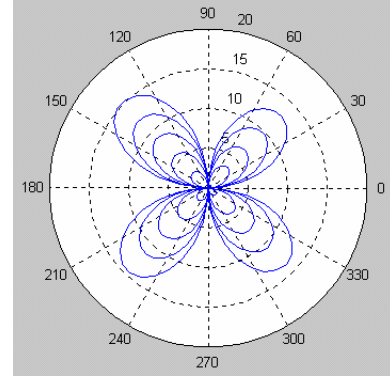
a) $r_1 = \sin(t)$; `polar(t,r1);`

Grafik aşağıda verilmiştir.



b) $r_2 = t \cdot \sin(t) \cdot \cos(t)$; `polar(t,r2);`

Grafik aşağıda verilmiştir.



2) Üç Boyutlu Grafikler (Uzayda Grafik):

Bu konuyla ilgili komut ve açıklamaları görmek için komut satırına **help graph3d** yazdığımızda aşağıdaki bilgiler gelir.

Three dimensional graphs.

Elementary 3-D plots.

`plot3` - Plot lines and points in 3-D space.

`mesh` - 3-D mesh surface.

`surf` - 3-D colored surface.

`fill3` - Filled 3-D polygons.

....

....

Biz bu komutlardan bazılarını göreceğiz.

a) plot3 Fonksiyonu:

Uzayda (x,y,z) koordinatları ile belirli vektörün (vektöre karşılık gelen uç noktasının) grafiğini çizer. `plot` fonksiyonuna benzer biçimde kullanılır.

Kullanımı:

plot3(x,y,z);

plot3(x,y,z,'Çizgi özellikleri',...);

plot3(x,y,z,'özellik1',değer1,'özellik2',değer2,...); biçimindedir.

Not 1) `plot3` komutunu uyguladığımızda, ilgili grafik

Figure (şekil) penceresinde oluşur. Grafiği daha iyi

inceleyebilmek için örneğin grafik derinliğini

algılayabilmek için grafiği bir **kutu** (prizma) içine

alabiliriz. Bunun için komut satırına **box on;** komutunu

girmeliyiz, kutuyu kaldırmak istediğimizde de **box off** komutunu kullanırız.

Not 2) Ayrıca grafik ile ilgili ayarlamalar için; **şekil** penceresinin üstündeki araçlardan faydalanabiliriz. Bu araçlar aşağıda gösterilmiştir:



Grafiği Büyütme Aracı: Bu aracı tıkladıktan sonra, grafik penceresine her tıklanışta grafik bize doğru yaklaşarak büyür.



Grafiği Küçültme Aracı: Bu aracı tıkladıktan sonra, grafik penceresine her tıklanışta grafik bizden uzaklaşarak küçülür.



Grafiği Döndürme Aracı: Bu aracı tıkladıktan sonra, grafik penceresinin köşelerine yakın bir yerden tutularak

Matlab Ders Notları

(farenin sol tuşu basılarak) istediğimiz kadar döndürebiliriz.

Not 3) Ayrıca üç boyutlu koordinat sistemine, incelemeyi daha rahat yapabilmek için ızgara çizgileri de koyabiliriz. Bunun için komut satırına **grid on**; komutunu yazarız. Izzarayı kaldırmak için de **grid off**; komutu uygulanır.

Örnek:

- a) (-3,5,8) noktasının grafiğini çizdiren (noktayı işaretleyen),
b) x değerleri 1 den 10 kadar 1 er artan bir dizide, y değerleri 5 den 50 ye kadar 5 er artan birer dizi olmak üzere; z değerleri de x dizisinin elemanlarının 2 katından y dizisinin değerlerinin 3 katının çıkarılmasıyla oluşturalım. Bu durumda belirlenen (x,y,z) noktalarından oluşan grafiği çizdirelim.
c) b) şıkında tanımlanan grafiğin çizgi rengini **sarı**, çizgi stilini **-**, çizgi noktalarının (marker) işaretini *****, marker kalınlığını **2** birim yapalım.

Çözüm:

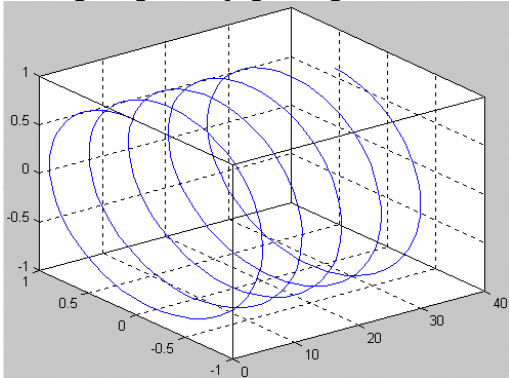
- a) `plot3(-3,5,8);`
b) `x=1:10;y=5:5:50;z=2*x-3*y;plot3(x,y,z);`
c) `x=1:10;y=5:5:50;z=2*x-3*y;plot3(x,y,z,'y-.*','linewidth',2);`

Örnek:

Açı ölçüleri $[0, 10\pi]$ aralığında 0.1 er artışla elde edilen değerleri x dizisine, bu değerlerin sinüslerini y dizisine, kosinüslerini de z dizisine atayalım. Bu durumda elde edilen (x,y,z) üçlülerinin grafiğini çizdirelim. Grafik penceresini hem kutu içine alalım hem de grafik ızgara çizgilerini koyalım.

Çözüm:

- `x=[0:0.1:10*pi];y=sin(x);z=cos(x);`
`plot3(x,y,z);box on;grid on;`
Buna göre grafik aşağıdaki gibi olacaktır.



b) Yüzey Grafikleri:

İki Boyutlu Düzlemin Koordinatlarını Tanımlayan Matris ve meshgrid Fonksiyonu ve Yüzey Grafiği:

Üç boyutlu uzayda, örneğin xOy düzleminde belirli bir dikdörtgen biçimindeki alanın, yatay ve düşey çizgilerle (grid line) daha küçük dikdörtgensel bölgelere ayrıldığını varsayalım. İşte bu çizgilerin kesim noktalarının koordinatları, grafik çiziminde gerekli olacaktır. Bu koordinatları tutan matris **meshgrid** fonksiyonu yardımıyla elde edilir. Bunun için; önce tanımlanmak

istenilen bölgenin x koordinatları bir diziye (örneğin x dizisine), y koordinatları bir diziye (örneğin y dizisine) atanır. Ardından **[X,Y]=meshgrid(x,y)**; komutunu uygulayarak (x ile X in ve y ile Y nin farklı olduğuna dikkat edin!) ilgili bölgenin koordinatları X ve Y dizilerine atanmış olur. Daha sonra X ve Y matrislerine bağlı **Z=f(X,Y)**; gibi bir matris elde edebiliriz. İşte bu Z fonksiyonu, üç boyutlu uzayda bir **yüzey** belirler, örneğin Z fonksiyonu X ve Y ye bağlı birinci dereceden bir fonksiyonsa bir **düzlem**, daha yüksek dereceden veya trigonometrik, üstel ... gibi fonksiyonlardan oluşan bir fonksiyon ise bir **yüzey** belirler. Bu fonksiyonun grafiğini çizdirebiliriz.: Bunun için;
surf(Z) ile ilgili bölgenin **üç boyutlu** yüzey grafiğini, **surface(Z)** ile bölgenin **iki boyutlu** grafiğini, **mesh(Z)** ile fonksiyonun tanımladığı yüzeyin **ağ** grafiğini **contour(Z)** ile de fonksiyonun tanımladığı yüzeyin **seviye** grafiğini çizdirebiliriz.

Örnek: xOy düzleminde;

yatay olarak, $[0 .. 10]$ bölgesini 0.1 er artımlı x dizisine; düşey olarak, $[0 .. 8]$ bölgesini 0.2 er artımlı y dizisine atayalım. Daha sonra, bu dizileri [X,Y] koordinat matrisine atayalım.

a) $Z=X+Y$ matrisine karşılık gelen fonksiyonun

- iki boyutlu düzlem grafiğini,
- üç boyutlu düzlem grafiğini,
- yüzey ağ grafiğini,
- yüzey seviye grafiğini çizdirelim.

b) $P=X.^2+Y.^2$ matrisine karşılık gelen fonksiyonun

- iki boyutlu düzlem grafiğini,
- üç boyutlu düzlem grafiğini,
- yüzey ağ grafiğini,
- yüzey seviye grafiğini çizdirelim.

c) $Q=\sin(X./2)+\cos(Y./3)$ matrisine karşılık gelen fonksiyonun

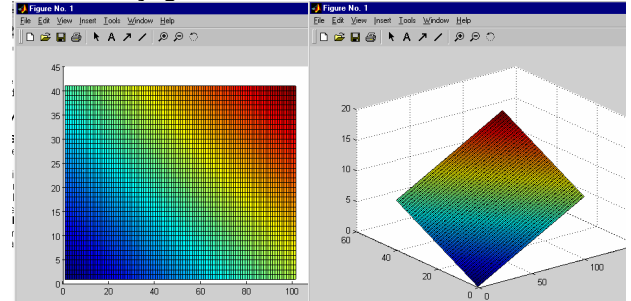
- iki boyutlu düzlem grafiğini,
- üç boyutlu düzlem grafiğini,
- yüzey ağ grafiğini,
- yüzey seviye grafiğini çizdirelim.

Çözüm:

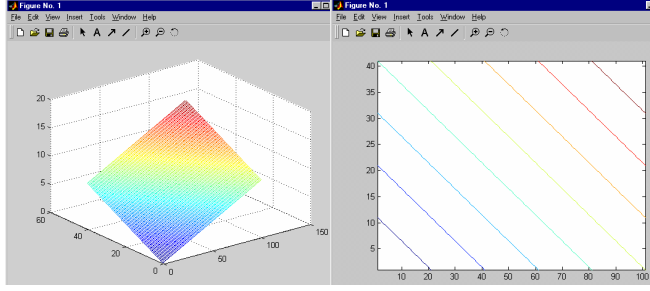
`x=[0:0.1:10];y=[0:0.2:8];[X,Y]=meshgrid(x,y);`
`Z=X+Y;P=X.^2+Y.^2;Q=sin(X./2)+cos(Y./3);`

a) i) `surface(Z)`; ii) `surf(Z)`; iii) `mesh(Z)`; iv) `contour(Z)`;

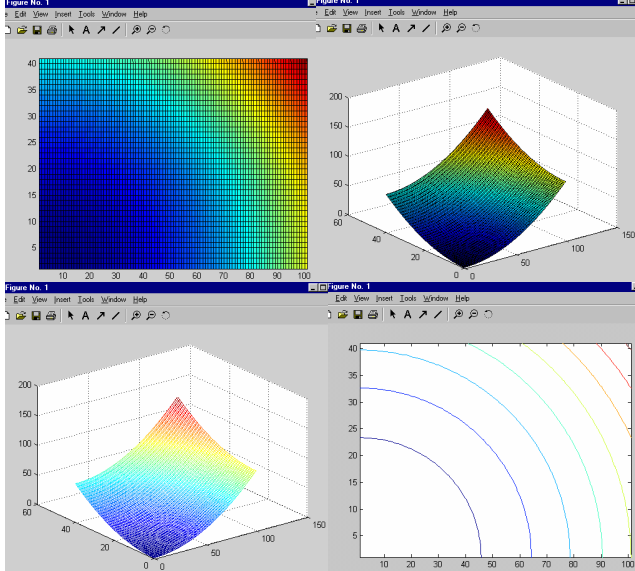
Grafikler aşağıda verilmiştir.



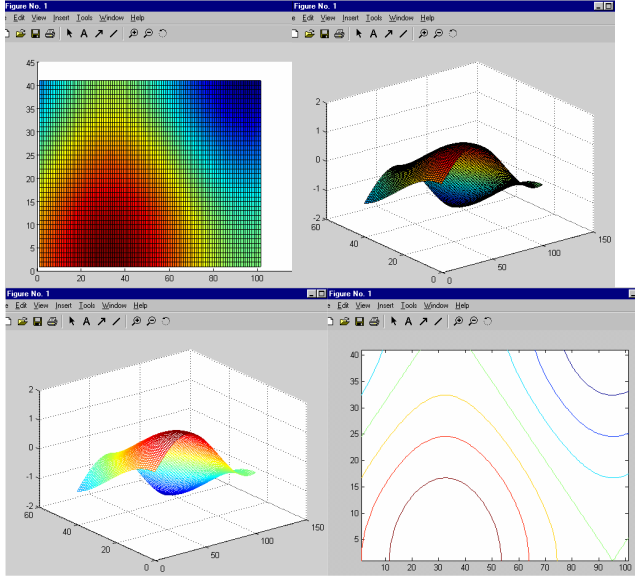
Matlab Ders Notları



b) i) surface(P); ii) surf(P); iii) mesh(P); iv) contour(P); Grafikler aşağıda verilmiştir.



c) i) surface(Q); ii) surf(Q); iii) mesh(Q); iv) contour(Q); Grafikler aşağıda verilmiştir.



peaks Fonksiyonu ile Kare Matris Oluşturma:

$n \times n$ lik bir kare matris oluşturan fonksiyon **peaks** dir. Kullanımı **peaks(n)** biçimindedir. Bu matrisi oluştururken x ve y koordinatlarıyla ilgili $3*(1-x).^2.*\exp(-(x.^2) - (y+1).^2) - 10*(x/5 - x.^3 - y.^5).*\exp(-x.^2-y.^2) - 1/3*\exp(-(x+1).^2 - y.^2)$ fonksiyon kullanılır.

Bu matrisi z gibi bir matris değişkenine **z=peaks(n)**; komutuyla atayıp z matrisi ile ilgili grafikler çizdirilebilir. Örneğin; **surface(z)** ile z nin iki boyutlu alan

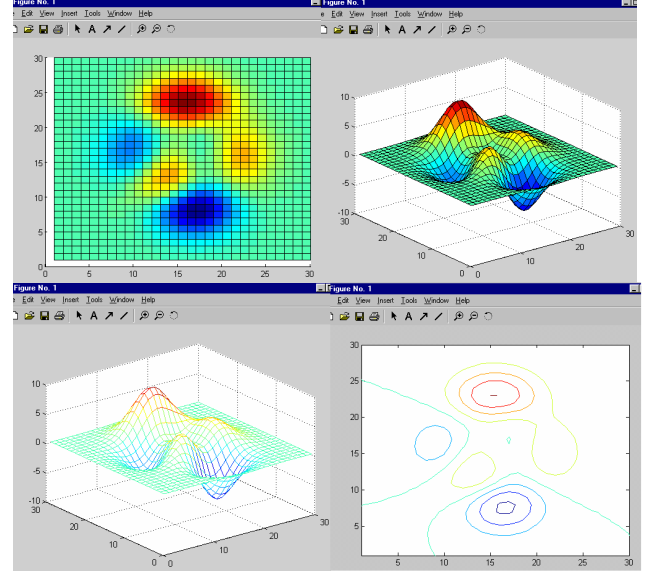
grafığı; **surf(z)** ile z nin yüzey grafığı, **mesh(z)** ile ağ grafığı, **contour(z)** ile yüzey seviye grafığı çizdirilebilir.

Örnek:

- 30x30 boyutlarındaki bir kare matrisi z dizisine atayalım,
- z matrisinin iki boyutlu alan grafığını,
- z matrisinin üç boyutlu yüzey grafığını,
- z matrisinin üç boyutlu yüzeyinin ağ grafığını,
- z matrisinin üç boyutlu yüzey seviye grafığını çizdirelim.

Çözüm:

- z=peaks(30)** b) **surface(z)** c) **surf(z)** d) **mesh(z)** e) **contour(z)** Grafikler aşağıda verilmiştir:



Not 1) Belirlenen yüzeyin rengi Matlab tarafından otomatik olarak belirlenir. Ancak bu renkten başka Matlabın hazır yüzey renk haritalarından birisi ile de boyanabilir. Bu renk haritaları şunlardır:

Color maps.

- hsv** - Hue-saturation-value color map.
- hot** - Black-red-yellow-white color map.
- gray** - Linear gray-scale color map.
- bone** - Gray-scale with tinge of blue color map.
- copper** - Linear copper-tone color map.
- pink** - Pastel shades of pink color map.
- white** - All white color map.
- flag** - Alternating red, white, blue, and black color map.
- lines** - Color map with the line colors.
- colorcube** - Enhanced color-cube color map.
- vga** - Windows colormap for 16 colors.
- jet** - Variant of HSV.
- prism** - Prism color map.
- cool** - Shades of cyan and magenta color map.
- autumn** - Shades of red and yellow color map.
- spring** - Shades of magenta and yellow color map.
- winter** - Shades of blue and green color map.
- summer** - Shades of green and yellow color map.

Aktif penceredeki yüzeyin rengini **bone** (kemik rengi) ile boyamak için, komut satırına; **colormap(bone)**; komutu yazılmalıdır.

Matlab Ders Notları

Not 2) İstersek belirlenen yüzeyi, istediğimiz bir resmin renkleri ile boyayabiliriz (başka bir deyişle yüzeye resmi giydirebiliriz). Bunun için aşağıdaki adımları takip edebiliriz:

- Yüzeyin tutamacını bir değişkene atarız.
Örneğin **yuzey=surf(peaks(40));** gibi.
- İstediğimiz bir resim dosyasının renk bilgisini bir diziye atarız.
Örneğin, **renk=imread('bayrak.jpg');** gib.
- Yüzey tutamacının **Cdata** özelliğine istediğimiz renk bilgisini koyarız.
Örneğin; **set(yuzey,'cdata',renk);** gibi. Ancak bu komuttan sonra, dizil boyutlarının uyuşmadığı gibilerden bazı hata mesajları gelir, bunlara kulak asmayın :)
- Son olarak ta yüzeyin **FaceColor** özelliğine **texturemap** yaparız
Örneğin **set(yuzey,'facecolor','texturemap');** gibi.

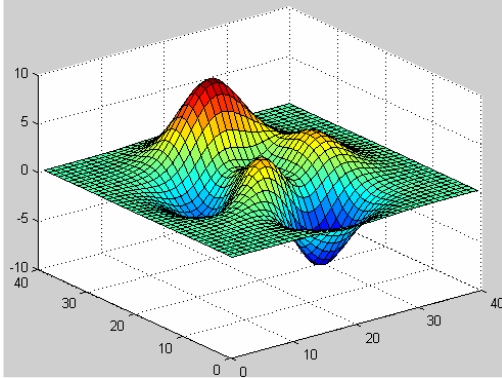
Örnek:

- 40x40 boyutlarında peaks fonksiyonu ile tutamac adı **yuzey** olan luşturulan yüzey nesnesini oluşturalım.
- Yüzeyi **prism** adlı renk haritası ile boyayalım.
- Yüzeye **start.jpg** adlı resmi giydirelim.

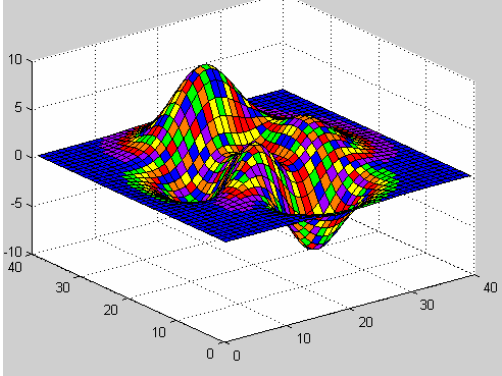
Çözüm:

- yuzey=surf(peaks(40));

Grafik aşağıdaki gibidir:



- yuzey=surf(peaks(40));colormap(prism);

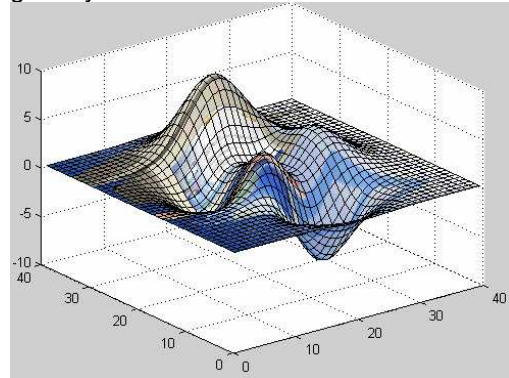


- yuzey=surf(peaks(40));renk=imread('start.jpg');
set(yuzey,'cdata',renk);
set(yuzey,'facecolor','texturemap');

Aşağıdaki resim, **start.jpg** adlı dosyaya aittir.



Yukarıdaki resmin yüzeye giydirilmiş durumu aşağıda görülüyor.



Bazı Özel Grafik Fonksiyonları:

Matlab'da daha bir çok grafik fonksiyonları vardır. Bunları görmek ve kullanımları hakkında bilgi almak için komut satırında **help specgraph** yazabiliriz.

Specialized graphs.

Specialized 2-D graphs.

- | | |
|----------|---|
| area | - Filled area plot. |
| bar | - Bar graph. |
| barh | - Horizontal bar graph. |
| comet | - Comet-like trajectory. |
| compass | - Compass plot. |
| errorbar | - Error bar plot. |
| ezplot | - Easy to use function plotter. |
| ezpolar | - Easy to use polar coordinate plotter. |
| feather | - Feather plot. |
| fill | - Filled 2-D polygons. |
| fplot | - Plot function. |

Matlab Ders Notları

- hist - Histogram.
- pareto - Pareto chart.
- pie - Pie chart.
- plotmatrix - Scatter plot matrix.
- rose - Angle histogram plot.
- scatter - Scatter plot.
- stem - Discrete sequence or "stem" plot.
- stairs - Stairstep plot.

Contour and 2-1/2 D graphs.

- contour - Contour plot.
- contourf - Filled contour plot.
- contour3 - 3-D Contour plot.
- clabel - Contour plot elevation labels.
- ezcontour - Easy to use contour plotter.
- ezcontourf - Easy to use filled contour plotter.
- pcolor - Pseudocolor (checkerboard) plot.
- voronoi - Voronoi diagram.

Specialized 3-D graphs.

- bar3 - 3-D bar graph.
- bar3h - Horizontal 3-D bar graph.
- comet3 - 3-D comet-like trajectories.
- ezgraph3 - General purpose surface plotter.
- ezmesh - Easy to use 3-D mesh plotter.
- ezmeshc - Easy to use combination mesh/contour plotter.
- ezplot3 - Easy to use 3-D parametric curve plotter.
- ezsurf - Easy to use 3-D colored surface plotter.
- ezsurf - Easy to use combination surf/contour plotter.
- meshc - Combination mesh/contour plot.
- meshz - 3-D mesh with curtain.
- pie3 - 3-D pie chart.
- ribbon - Draw 2-D lines as ribbons in 3-D.
- scatter3 - 3-D scatter plot.
- stem3 - 3-D stem plot.
- surf - Combination surf/contour plot.
- trisurf - Triangular surface plot.
- trimesh - Triangular mesh plot.
- waterfall - Waterfall plot.
- Solid modeling.
- cylinder - Generate cylinder.
- sphere - Generate sphere.
- ellipsoid - Generate ellipsoid.
- patch - Create patch.
- surf2patch - Convert surface data to patch data.

.....
Bunların hepsini açıklamaya gerek yok, ancak bazılarını görelim:

Ezplot Fonksiyonu:

Matlab'ın en güçlü fonksiyonlarından biridir. Bu grafik fonksiyon; metin olarak girilen **f(x,y)=0** biçimindeki **kapalı fonksiyon** grafiklerini çizdirmeye yarar.

Kullanımı:

ezplot('kapalı fonksiyon ifadesi'); veya
ezplot('kapalı fonksiyon ifadesi',[xmin xmax]); veya
ezplot('kapalı fonksiyon ifadesi',[xmin xmax ymin ymax]);
biçiminde kullanılabilir. Burada
xmin x değerlerinin başlangıç değeri,
xmax x değerlerinin bitiş değeri,

ymin y değerlerinin başlangıç değeri,
ymax y değerlerinin bitiş değeri.

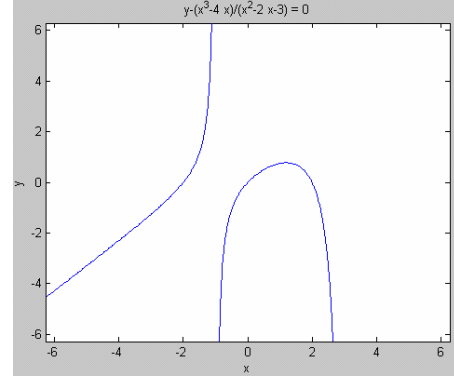
Örnek:

$$y = \frac{x^3 - 4x}{x^2 - 2x - 3}$$
 fonksiyonunun grafiğini ezplot fonksiyonu ile grafiğini;

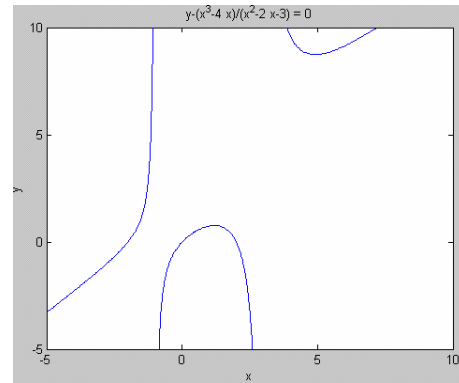
a) Normal olarak b) x değerleri [-5, 10] aralığında
c) x değerlerini [-5, 5] aralığında, y değerlerini [-10, 30] aralığında çizdirelim.

Çözüm:

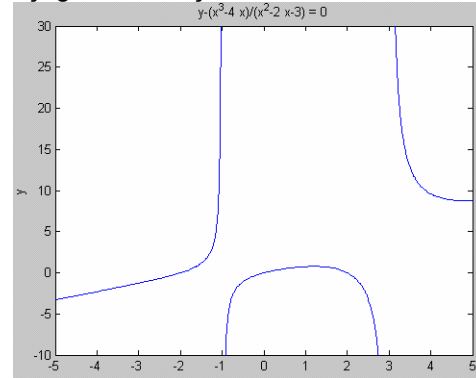
a) ezplot('y-(x^3-4*x)/(x^2-2*x-3)'); grafiği aşağıda verilmiştir.



b) ezplot('y-(x^3-4*x)/(x^2-2*x-3)',[-5 10]); grafiği aşağıda verilmiştir.



c) ezplot('y-(x^3-4*x)/(x^2-2*x-3)',[-5 5 -10 30]); grafiği aşağıda verilmiştir.

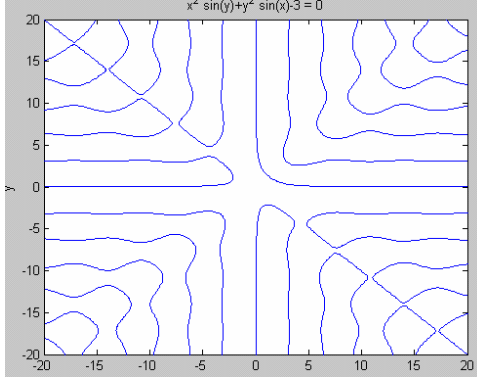


Örnek:

$x^2 \sin y + y^2 \sin x = 3$ bağıntısının grafiğini x ve y değerlerinin her ikisi de [-20, 20] aralığında çizdirelim.

Matlab Ders Notları

Çözüm: `ezplot('x^2*sin(y)+y^2*sin(x)-3',[-20 20 -20 20]);`
grafığı aşağıda verilmiştir.



Ezplot3 Fonksiyonu:

Bu da metin olarak girilen **f(x,y,z)=0** biçimindeki **kapalı fonksiyon** grafiklerini üç boyutlu uzayda çizdirmeye yarar. Burada x, y ve z vektörleri t gibi bir parametreye bağlı birer fonksiyonlardır. Yani, $x=f(t)$, $y=g(t)$ ve $z=h(t)$ gibi birer fonksiyondur.

Kullanımı:

`ezplot3('f(t)','g(t)','h(t)');` veya
`ezplot3('f(t)','g(t)','h(t)',[tmin tmax]);`

Not 1) Burada tmin t parametresinin başlangıç değeri, tmax t parametresinin bitiş değeridir.

Not 2) tmax ve tmin değerleri belirtilmezse Matlab bunu $[0 \ 2\pi]$ olarak alır.

Örnek: $x=\sin(t)$, $y=e^x$ ve $z=t \cdot \cos(t)$ parametrik

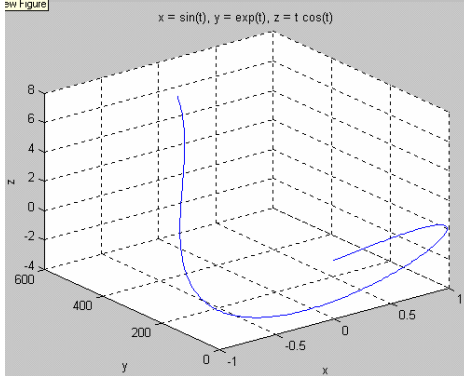
denklemleriyle verilen üç boyutlu eğrinin grafiğini ;

a) Normal varsayılan aralıkta

b) t değerleri $[-10\pi, 10\pi]$ aralığında çizdirelim.

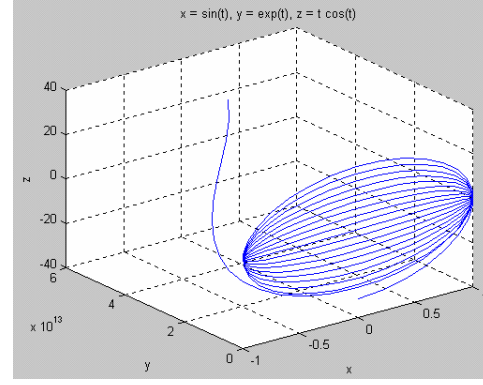
a)

`ezplot3('sin(t)','exp(t)','t*cos(t)');`



b)

`ezplot3('sin(t)','exp(t)','t*cos(t)',[-10*pi 10*pi]);`



cylinder Fonksiyonu:

Silindir çizdirmeye yarayan fonksiyondur.

Kullanımı: **cylinder**; veya **cylinder(r,n)**; biçimindedir.

Not 1) Buradaki r silindirin çapı, n de silindiri oluşturan yüzey sayısıdır. varsayılan n değeri 20 dir.

Not 2) n değerini küçük alırsak, örneğin 5, 10 gibi silindir prizmaya dönüşür.

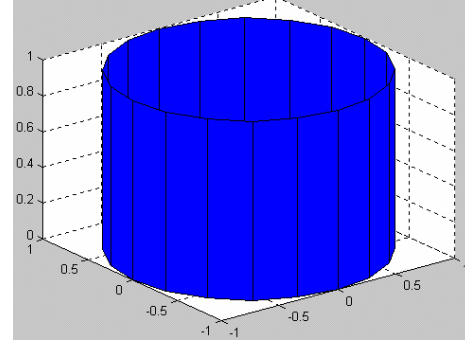
Örnek:

a) Normal varsayılan değerlerle bir silindir

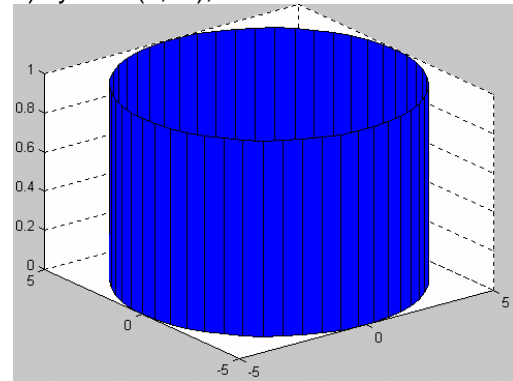
b) Taban çapı 3 birim , yüzey sayısı 50 olan bir silindir;

c) Taban çapı 7 birim olan bir üçgen prizmayı, hsv renk haritasıyla boyayarak çizdirelim.

Çözüm: a) cylinder;

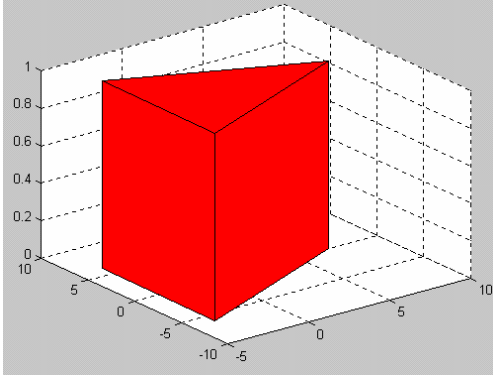


b) `cylinder(3,50);`



c) `cylinder(7,3);colormap(hsv);`

Matlab Ders Notları



sphere Fonksiyonu:

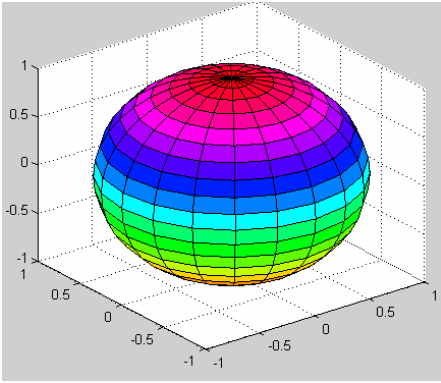
1 birim yarıçapında bir küre çizdirmeye yarayan fonksiyondur.

Kullanımı: **sphere**; veya **sphere(n)**;

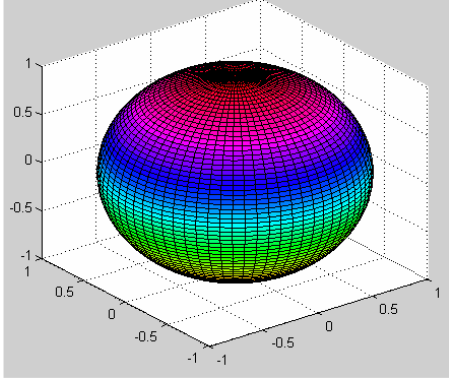
Not: Buradaki n sayısı küre yüzeyini oluşturan yüzeylerin sayısıdır varsayılan değeri 20 dir.

Örnek: a) varsayılan değerlerde bir küre; b) Yüzeyini oluşturan parçaların sayısı 70 olan bir küre çizdirelim.

Çözüm: a)sphere;



b) sphere(70);



eval Fonksiyonu ve Kullanımı:

Matlab'da kullanıcının metin olarak girdiği ifadeyi, yorumlayarak hesaplanacak fonksiyon biçimine getiren fonksiyon **eval** fonksiyonudur.

Kullanımı:

değişken=eval("hesaplanacak-fonksiyon-metni")

Örnek: Kullanıcının girdiği bir x sayısının, istediği fonksiyon altındaki görüntüsünü (değerini) hesaplatan bir program yazalım.

Çözüm:

```
clc;x=input('Sayıyı gir :');
y=input('Fonksiyonu gir f(x)=','s');y=eval(y);
fprintf('f(%d) = %d',x,y);
```

Örnek: x değerleri -5 ile 15 aralığında, 0.1 er artışla tanımlanan sayı aralığında olan, kullanıcının istediği bir fonksiyonun grafiğini çizdiren bir program yazalım.

Çözüm:

```
x=[-5:0.1:10];
ifade=input('Fonksiyonu gir f(x)=','s');
y=eval(ifade);
plot(x,y);
```

Örnek: x değerleri kullanıcının belirlediği bir aralıkta olan ve kullanıcının istediği bir fonksiyonun grafiğini çizdiren bir program yazalım.

Çözüm:

```
clc;x=input('x aralığını gir :','s');x=eval(x);
y=input('Fonksiyonu gir f(x)=','s');y=eval(y);
plot(x,y);
"....
```

Grafik Nesneleri ve Grafiksel Arabirim İle Programlama:

Matlab'da grafiksel işlemler, grafik nesneleri ile yapılır.Grafik nesneleri ve arasındaki hiyerarşi aşağıdaki gibidir:

I. Root(Kök)

II. Figure (Şekil)

a) Axes(Eksen)

1. Image (Resim)
2. Light (Işık)
3. Line (Çizgi)
4. Patch (Yama)
5. Rectangle (Dikdörtgen)
6. Surface (Yüzey)
7. Text (Metin)

b) Uicontrol (Kontrol)

c) Uimenu (Menü)

d) Uicontextmenu (Kısayol menüsü)

Bu nesneler bir hiyerarşi içerisinde birbirlerine bağımlı olarak oluşturulabilirler.Örneğin, grafik kullanıcı arabirimi oluşturmak istediğimizde, düğme, liste kutusu, onay kutusu ... gibi (**uicontrol**) kontrol nesnelerini bir (**figure**) şekil penceresi üzerine yerleştirebilirsiniz.Bu durumda şekil penceresi, kontrol nesnelerinin (**parent**) **ebeveyni** iken, kontrol nesneleri de şekil penceresinin (**children**) **çocuklarıdır**.

Grafik nesnelerini görmek, düzenlemek için ; komut satırında **guide** yazarak veya **File>New>GUI** sekmesi tıklanırsa; karşımıza aşağıdaki GUI (Grafiksel Arabirim) penceresi gelir.

Bu arabirim penceresinde aşağıdaki, **uicontrol** nesnelerini oluşturabiliriz.

- a) pushbutton (komut düğmesi)
- b) togglebutton (açma-kapama düğmesi)
- c) radiobutton (radyo düğmesi)

Matlab Ders Notları

- d) checkbox (onay kutusu)
 - e) edit (metin düzenleme kutusu)
 - f) text (sabit metin kutusu)
 - g) listbox (liste kutusu)
 - h) popupmenu (açılır liste kutusu)
 - i) slider (kaydırma çubuğu)
 - j) frame (çerçeve)
- cc

Bazı Grafik İşleme Fonksiyonları:

a) Bir Nesnenin Özelliklerini Elde Etme: get Kullanımı: get(nesne_tanımlayıcı,'özellik_adı');

Örnek: resim adlı bir figure nesnesi oluşturup bu nesnenin units (birim) özelliğini elde edelim.

Çözüm:
resim=figure; (enter)
get(resim,'units') (enter) tuşuna basıldığında ekranda;
ans =
pixels
görülür. Bunun da resim adlı figure nesnesinin units özelliğinin, pixel olduğu anlaşılır.

b) Bir Nesnenin Özelliklerini Değiştirme: set Kullanımı: set(nesne_tanımlayıcı,'özellik_adı','yeni_değer'); veya set(nesne_tanımlayıcı,'özellik_adı1','yeni_değer1', 'özellik_adı2','yeni_değer2',..., 'özellik_adıN','yeni_değerN');

Örnek: Bir önceki örnekteki resim adlı figure nesnesinin units özelliğini centimeters yapan komutu yazalım.

Çözüm:
set(resim,'units','centimeters')

Örnek:
Bir önceki resim adlı figure nesnesinin pozisyonunu, sol alt köşesinin, ekranın sol alt köşesinden yatay olarak 1 cm, dikey olarak 2 cm uzaklıkta, yatay uzunluğunun 10 cm dikey uzunluğu 7 cm olarak belirleyelim. Resim adlı pencerenin zemin renginin RGB renklerini 0.6 0.7 0.8 oranında olmasını (açık mavi) ve pencerenin ismini 'Şekil Penceresi' olmasını sağlayalım.

Çözüm:
set(resim,'position',[1 2 10 7],
'color',[0.6 0.7 0.8],'name','Şekil Penceresi')

c) Bir Nesneyi Silmek: delete Kullanımı: delete(nesne_tanımlayıcı)

Örnek: Bir figure penceresinde adı eksen olan bir eksen nesnesi ve yazı adında üzerinde Deneme yazan bir text nesnesi açıp sonra bunları teker teker silen komutları yazalım.

Çözüm:
eksen=axes
yazi=uicontrol('style','text','string','Deneme')
Şimdi de teker teker silelim:

delete(yazi)
delete(eksen)

Grafik Nesneleri ve Özellikleri:

1. root Nesnesi

Matlab'da hiyerarşinin en tepesinde olan nesnesidir. Bu nesne Matlab'ı çalıştırdığımızda otomatik olarak oluşur. Bu nedenle root nesnesinin tutamacı (handle) yani tanımlayıcı sayısı daima 0'dır. Bu nesne ekranımızın çözünürlüğü, boyutu, o anda açık bulunan pencerelerinin (children) tanımlayıcı sayıları, farenin ekrandaki konumunu ve Matlab komut penceresi gibi bir çok ayarı tutan bir nesnedir.

Örnek:

- a) root nesnesinin tüm özelliklerini ve o andaki ayarlarını görüntülemek için;
- b) root nesnesinin tüm özelliklerini ve olası ayarlarını görüntülemek için komut satırına hangi komut yazılmalıdır?

Çözüm: a) get(0) b) set(0)

root Nesnesinin Bazı Özellikleri:

CurrentFigure Aktif olan pencere tutamacını tutan özelliktir.

Format: Komut satırında sayıların nasıl görünmesi gerektiğini belirleyen özelliktir.

FormatSpacing: Komut satırında satırlar arası boşlukları tutan özelliktir. Değeri loose veya compact olabilir, loose değeri verilirse satır aralı boşluklar bulunur, compact değeri verilirse boşluk olmaz.

Language: İşletim sisteminin bölgesel ayarlardaki tanımlanan dili tutan özelliktir.

PointerLocation: Farenin ekrandaki o andaki konumunu tutan özelliktir.

Units: Uzunluk ve konum bilgilerinin birimini tutan özelliktir, centimeters, pixels, inches, characters ... gibi değerler alabilir.

Tag: İstenen herhangi bir metni tutmaya yarayan kullanıcıya ayrılmış bir özelliktir.

UserData: İstenen bir veriyi (dizi, hücre, yapı, sınıf vs.) tutmaya yarayan kullanıcıya ayrılmış bir özelliktir.

Örnek:

Komut satırında sayıların a) Noktadan sonra 4 basamaklı (tek hassasiyetli) b) Noktadan sonra 14 basamaklı (çift hassasiyetli) c) Rasyonel (kesir) olarak görünmelerini sağlayan komuları yazalım.

Çözüm: a) set(0,'format','short') b) set(0,'format','long')
c) set(0,'format','rational')

Örnek: Ekran üzerindeki uzunluk birimini a) bildiren b) cm c) pixel olarak düzenleyen komutu yazalım.

Çözüm: a) get(0,'units') b) set(0,'units','centimeters')
c) set(0,'units','pixels')

Örnek: Farenin ekrandaki konumunu a) bildiren b) Ekranın sol alt köşesinden yatay olarak 3 cm sağ, dikey olarak 5 cm yukarıda olmasını sağlayan komutu yazalım.

Çözüm: a) get(0,'pointerlocation')

Matlab Ders Notları

b) set(0,'units','centimeters');set(0,'pointerlocation',[3 5]);

2. figure Nesnesi ve Özellikleri:

Programların grafik arabirimlerini, verilerin görselleştirilme işlemlerinin üzerinde yapmayı sağlayan şekil penceresi nesnesidir.

figure Nesnesinin Bazı Özellikleri:

Children: Şekil penceresinin çocuk (axes, uicontrol, uimenu, uicontextmenu gibi) nesnelerinin tutamaçlarını tutan dizidir.

Color: Şekil penceresinin zemin rengini düzenlemeye yarayan 1x3 boyutlu bir dizidir. Dizinin elemanları yerine 0 ile 1 arasında değişen reel sayılar yazılabilir ki bu değerler tanımlanan rengin RGB renk oranlarını gösterir. [0 0 0] değeri siyah rengi, [1 1 1] değeri beyaz rengi, [1 0 0] değeri kırmızıyı, [0 1 0] değeri yeşili, [0 0 1] değeri maviyi, [0.6 0.7 0.8] değeri açık maviyi belirtir.

MenuBar: Şekil penceresinin üst tarafında bulunan standart menüleri kapatıp-açmaya yarar.

Name: ekil penceresi üstündeki başlığı düzenlemeye yarayan özelliktir.

NextPlot: Grafik işlemleri aktif pencere üzerinde nasıl gerçekleşeceğini belirleyen özelliktir. Değeri **add**, **replace** veya **replacechildren** olabilir

NumberTitle: Şekil penceresinin üst bölümündeki şekil penceresi numarasının görünüp görünmemesini sağlayan özelliktir. Değeri **off** yaparak numaranın görünmemesi, **on** yaparak görünmesi sağlanır.

Pointer: Pencere üzerine geldiğinde, farenin biçimini belirleyen özelliktir. Bu değerler **crosshair**, **fullcrosshair**, **arrow**, **ibeam**, **watch**, **right**, **bottom**, **circle**, **cross** gibi olabilir.

Position: Şekil penceresinin ekran üzerindeki konumunu belirleyen 1x4 boyutlu bir diziyi düzenleyen özelliktir.

Resize: Şekil penceresinin boyutlarını değiştirmeye yarayan özelliktir. Değeri **on** olursa şekil penceresi boyutları değiştirilebilir, **off** olursa değiştirilemez.

Units: Şekil penceresinin uzunluk ve konum bilgilerinin birimini tutan özelliktir, **centimeters**, **pixels**, **inches**, **characters** ... gibi değerler olabilir.

3. uicontrol Nesnesi

Bir şekil penceresi üzerinde kullanıcı ile iletişim sağlayacak bazı nesneler oluşturmaya ve düzenlemeye yarar. uicontrol nesneleri aşağıda verilmiştir:

- pushbutton (komut düğmesi)
- togglebutton (açma-kapama düğmesi)
- radiobutton (radyo düğmesi)
- checkbox (onay kutusu)
- edit (metin düzenleme kutusu)
- text (sabit metin kutusu)
- listbox (liste kutusu)
- popupmenu (açılır liste kutusu)
- slider (kaydırma çubuğu)
- frame (çerçeve)

uicontrol Nesnesinin Bazı Özellikleri:

BackgroundColor: Nesnenin arka plan rengini belirlemeye yarayan özelliktir.

Callback: Kontrol düğmesinin fare ile tıklandığında tetiklenecek geri çağırılabilir fonksiyonu belirleyen özelliktir.

CData: Komut düğmesi ya da açma-kapama düğmesi üzerinde görüntülenebilecek gerçek renklerden oluşan resmi tanımlayan bir matristir.

Enable: Kontrolün kullanılıp-kullanılmayacağını belirleyen özelliktir. Değeri **on** ise düğme kullanılabilir ve normal renktedir, **off** ise düğme kullanılamaz ve rengi siliktir, **inactive** ise düğme yine kullanılamaz fakat rengi normaldir.

FontAngle: Nesnenin string özelliğinin tuttuğu yazı biçimini düzenlemeye yarayan özelliktir, **normal**, **italic** veya **oblique** değerlerini alabilir.

FontName: Nesnenin string özelliğinin tuttuğu yazı fontunu düzenlemeye yarayan özelliktir..

FontSize: Nesnenin string özelliğinin tuttuğu yazı boyutunu düzenlemeye yarayan özelliktir.

FontUnits: Fontsize özelliği ile belirlenen yazıtipi büyüklüğünü birimini belirlemeye yarayan özelliktir. Değeri **centimeters**, **inch**, **points** veya **normalized** olabilir.

FontWeight: Nesnenin string özelliğinin tuttuğu yazı tipinin ağırlığını düzenlemeye yarayan özelliktir. **normal**, **bold** veya **light** olabilir

ForegroundColor: Nesnenin string özelliğinin tuttuğu yazının rengini (RGB) tutan 1x3 boyutunda bir diziyi belirlemeye yarar.

HorizontalAlignment: Metin kutusu ve statik metin nesnelerinin etiket metnini hizalamaya yarayan özelliktir. **left**, **right** veya **center** değerlerini alabilir.

Position: Nesnenin şekil penceresi üzerindeki pozisyonunu belirleyen 1x4 boyutlu bir diziyi düzenlemeye yarayan özelliktir.

String: Nesnenin üzerinde bulunması gereken yazıyı düzenlemeye yarayan özelliktir.

Style: Kontrol nesnesinin tipini belirleyen özelliktir.

Değeri **pushbutton** olduğunda nesne komut düğmesi, **togglebutton** olduğunda nesne açma-kapama düğmesi, **radiobutton** olduğunda nesne radyo düğmesi, **checkbox** olduğunda nesne onay kutusu, **edit** olduğunda nesne metin düzenleme kutusu, **text** olduğunda nesne sabit metin kutusu, **listbox** olduğunda nesne liste kutusu, **popupmenu** olduğunda nesne açılır liste kutusu, **slider** olduğunda nesne kaydırma çubuğu, **frame** olduğunda nesne çerçeve nesnesine dönüşür.

Tag: Nesneye bir etiket ismi vermeye yarayan özelliktir. Bu özellik sayesinde nesne **findobj** komutuyla bulunabilir.

TooltipString: Fare nesnenin üzerine geldiğinde bir ipucu ya da açıklama metni görüntülemeye yarayan özelliktir.

Units: Nesnenin uzunluk ile ilgili birimini düzenlemeye yarayan özelliktir. Değeri **centimeters**, **pixels**, **inches**, **characters** ... olabilir.

4. uimenu Nesnesi

Şekil penceresi üzerinde oluşturulabilecek menü nesneleridir. Üst seviye ve alt seviye menüler olabilir. Üst seviye menülerinin ebeveyni (parent) şekil penceresidir, alt seviye bir menünün ebeveyni ise başka bir menüdür.

Matlab Ders Notları

uimenu Nesnesinin Bazı Özellikleri:

Callback: Menü nesnesinin fare ile tıklanarak seçildiğinde tetiklenecek geri çağırılmalı fonksiyonu belirleyen özelliktir.

Checked: Menü seçeneği tıklandığında, sol tarafında seçildiğine dair onay işareti belirlenmesini sağlayan bir özelliktir. Değeri **on** olduğunda onay işareti konur, **off** olduğunda konmaz.

Enable: Menü seçeneğinin kullanılıp kullanılamayacağını belirleyen özelliktir. Değeri **on** ise seçenek kullanılabilir ve normal renktedir, **off** ise düğme kullanılamaz ve rengi siliktir.

ForegroundColor: Menü seçeneğinin string (etiket yazısının) rengini belirlemeye (RGB kodlarını tutan 1x3 boyutunda bir dizi) yarayan özelliktir.

Label: Menü seçeneğinin üstünde görüntülenecek metni belirlemeye yarayan özelliktir. Menü açıkken, kısa yoldan çalışmasını istersek, ilgili kısayol karakterinin önüne **&** işareti konmalıdır.

Position: Menü seçeneğinin, ebeveyni içerisindeki göreceli pozisyonunu belirlemeye yarayan özelliktir. İlk menü değeri 1 dir.

Separator: Menü nesnesi üzerinde bir ayırıcı işaret konulmasını düzenler. Değeri **on** ise ayırıcı işaret konur, **off** ise konmaz.

Tag: Menüye bir etiket ismi vermeye yarayan özelliktir. Bu özellik sayesinde nesne **findobj** komutuyla bulunabilir.

Visible: Menü'nün görünüp-görünmemesini düzenleyen özelliktir. Değeri **on** ise menü görünür, **off** ise görünmez.

Örnek: Tutamac ismi **hesap** , etiket ismi **Hesapla** , şekil penceresinde ana menünün 3. sırasında bulunan ve kısayol tuşu **H** harfi olan bir menü oluşturalım.

Çözüm:

```
hesap=uimenu('label','&Hesapla','position',3,  
'parent',figure);
```

5. axes Nesnesi

Üzerinde çizim, animasyon, resim ve video görüntüleme gibi bir çok işlemler yapmaya yarayan nesnedir.

axes Nesnesinin Bazı Özellikleri:

Box: Grafiğin, iki boyutlu çizimlerde bir dikdörtgen içinde, üç boyutlu çizimlerde bir küp içinde görünüp görünmemesini sağlayan özelliktir. Değeri **on** ya da **off** olabilir.

CameraPosition: Eksen koordinatlarında kameranın sahneyi gördüğü pozisyonu tanımlayan 1x3 lük bir diziyi belirleyen özelliktir.

CameraPositionMode: Kamera pozisyonu ayarını düzenleyen özelliktir. Değeri **auto** olduğunda Matlab pozisyonu otomatik belirler, **manual** olduğunda ise CameraPosition daki değer geçerlidir.

CameraTarget: Eksen koordinatlarında kameranın baktığı noktayı tanımlamaya yarayan 1x3 lük diziyi tutan özelliktir.

CameraTargetMode: Kamera hedefini tayin etmeye yarayan özelliktir. Değeri **auto** olduğunda Matlab kamera hedefini otomatik belirler, **manual** olduğunda ise CameraTarget daki değer geçerlidir.

CameraUpVector: Kameranın CameraTarget ve CameraPosition ile tayin edilen bakış doğrultusu etrafındaki dönmeyi belirleyen 1x3 lük vektördür.

CameraUpVectorMode: Kameranın bakış doğrultusu etrafındaki dönmelerini tayin etmeye yarayan özelliktir. Değeri **auto** olduğunda Matlab kameranın bakış doğrultusu etrafındaki dönmeyi otomatik belirler, **manual** olduğunda ise CameraUpVector daki değer geçerlidir.

CameraViewAngle: Görme alanını belirleyen açıdır. 0° ile 180° arasında bir değer olabilir.

CameraViewAngleMode: Görme alanını belirleyen açının modunu belirleyen özelliktir. Değeri **auto** olduğunda Matlab kameranın görme açısını bütün sahneyi görecektir biçimde en küçük açığa ayarlar, **manual** olduğunda ise CameraViewAngle daki değer geçerlidir.

Children: Eksen üzerinde oluşturulmuş çocuk nesnelerinin (line, light, patch, surface, rectangle ...gibi) tutamaçlarını tutan dizidir.

Color: Eksenin arka rengini tanımlayan 1x3 lük dizidir. Ayrıca değeri **none** de olabilir. Bu durumda eksen şeffaflaşarak arkasındaki nesneleri de gösterir.

DataAspectRatio: Eksen çizgilerinin (x, y ve z) birbirlerine göre oranını belirleyen 1x3 lük bir diziyi tutan özelliktir.

DataAspectRatioMode: Eksen çizgilerinin birbiri ile ilgili oranını belirlemeye yarayan özelliktir. Değeri **auto** olduğunda Matlab eksen çizgilerinin birbirine oranını otomatik olarak ayarlar, **manual** olduğunda ise DataAspectRatio daki değer geçerlidir.

FontAngle: Eksen çizgilerinin etiketleri ve çentik değerleri için kullanılan yazı biçiminin eğimini düzenlemeye yarayan özelliktir, **normal,italic** veya **oblique** değerlerini alabilir.

FontName: Eksen çizgilerinin etiketleri ve çentik değerleri için kullanılan yazı fontunu düzenlemeye yarayan özelliktir..

FontSize: Eksen çizgilerinin etiketleri ve çentik değerleri için kullanılan yazı tipi boyutunu düzenlemeye yarayan özelliktir.

FontUnits: Fontsize özelliği ile belirlenen yazıtipi büyüklüğünü birimini belirlemeye yarayan özelliktir. Değeri **centimeters, inch, points** veya **normalized** olabilir.

FontWeight: Eksen çizgilerinin etiketleri ve çentik değerleri için kullanılan yazı tipinin ağırlığını düzenlemeye yarayan özelliktir. **normal, bold** veya **light** olabilir

GridLineStyle: Çizim alanını eksen çizgilerine dik olarak bölen çizgilerin (grid lines) stilini belirleyen özelliktir. Değeri: -, --, :, -. ve **none** olabilir.

Layer: Eksen çizgilerinin eksen çizim alanının üst ya da alt tarafında olmasını sağlayan özelliktir. Değeri **top** veya **bottom** olabilir.

LineStyleOrder: Çoklu bir grafik çizimi yapıldığında, sırasıyla grafik çizgilerini belirlemeye yarayan özelliktir. Örneğin ilk grafik * lardan, ikincisi - lardan üçüncüsü + lardan oluşması isteniyorsa; bu özelliğe atanan değer '*|-|+' olmalıdır.

LineWidth: Eksen çizgilerinin kalınlıklarını düzenlemeye yarayan özelliktir. Varsayılan değer 0.5 noktadır.

Matlab Ders Notları

NextPlot: Grafik işlemlerinin aktif eksen üzerinden nasıl gerçekleşeceğini belirleyen özelliktir. Üç farklı değer alabilir bunlar: **add**, **replace** ve **replacechildren** dir.

Position: Eksen nesnesinin şekil penceresi üzerindeki pozisyonunu belirleyen 1x4 boyutlu bir diziye düzenlemeye yarayan özelliktir.

Projection: Eksen üzerindeki grafiğin üç boyutlu görüntüleme yöntemini belirleyen özelliktir. İki değer alabilir, bunlar: **ortographic** ve **perspective** dir.

Tag: Eksen nesnesine bir etiket ismi vermeye yarayan özelliktir. **get** fonksiyonu ile değerini öğrenebilir **set** fonksiyonu ile istediğimiz bir değeri atayabiliriz.

Title: Eksenin başlık metninin tutamacıdır.

Units: Eksen nesnesinin uzunluk ile ilgili birimini düzenlemeye yarayan özelliktir. Değeri **centimeters**, **pixels**, **inches**, **characters** ... olabilir.

Visible: Eksen nesnesinin görünüp-görünmemesini düzenleyen özelliktir. Değeri **on** ise eksen görünür, **off** ise görünmez.

XColor, YColor, ZColor: x, y, z eksenlerinin renklerini düzenleyen özelliktir.

XDir, YDir, ZDir: X, y, z eksen değerlerinin artış yönünü belirler. Değeri **normal** ise artış normal yöndedir, **reverse** ise artış ters yöndedir.

XGrid, YGrid, ZGrid: Çizim alanı eksen çizgilerine dik yönde bölen çizgilerin (gridlines) görünüp görünmemesini belirleyen özelliktir. Değeri **on** ya da **off** olabilir.

XLabel, YLabel, ZLabel: Eksen çizgi etiketlerini düzenlemeye yarayan özelliktir.

XLim, YLim, ZLim: Eksenin x, y ve z çizgilerinin sayısal aralıklarını belirleyen özelliktir. Örneğin x çizgisinin sayısal değeri 10 ile 120 arasında olmasını istersek; **XLim** özelliğinin değeri [10 120] olmalıdır.

XLimMode, YLimMode, ZLimMode: Eksen çizgi değerlerinin sayı aralıklarını düzenlemeye yarayan özelliktir. Değeri **auto** olduğunda Matlab eksen çizgilerinin sayısal aralıkları otomatik olarak belirlenir, **manual** olduğunda **XLim**, **ZLim** veya **ZLim** özelliğinin değeri geçerlidir.

XScale, YScale, ZScale: Eksen çizgilerinin sayısal ölçeğini belirlemeye yarayan özelliktir. Değeri **linear** olduğunda sayı artışı düzgün artar, **log** olduğunda logaritmik artar.

Örnek: Adı **eksen** olan, şekil penceresinin sol alt köşesinden yatay ve düşey olarak 1 cm, 1 cm uzaklıkta olan ve yatay uzunluğu 8 cm düşey uzunluğu 5 cm olan bir eksen nesnesi oluşturalım.

Çözüm: **eksen=axes('units','centimeters','position',[1 1 8 5]);**

Örnek: Bir önceki örnekte tanımlanan eksen adlı eksen nesnesinin zemin rengini [0.3 0.4 0.5] kodlu (lacivert), yapan komutu yazalım.

Çözüm: **set(eksen,'color',[0.3 0.4 0.5])**

Örnek: Eksen başlığına Grafik başlığını verelim.

Çözüm: **baslik=get(eksen,'title');set(baslik,'string','Grafik');**

Örnek: Grafiğin x ekseninin etiketini **x Eksen**i, y ekseninin etiketini de **y Eksen**i olarak düzenleyen komutları yazalım.

Çözüm: **xetiket=get(eksen,'xlabel');yetiket=get(eksen,'ylabel');set(xetiket,'string','x Eksen');set(yetiket,'string','y Eksen');**

Örnek: x ekseninin sayı değerlerini 5 ile 25 arasında, y ekseninin sayı değerlerini de -10 ile 50 değerleri arasında düzenleyelim.

Çözüm: **set(eksen,'xlim',[5 25],'ylim',[-10 50])**

Örnek: Eksen çizgi kalınlıklarını 2 nokta yapan komutu yazalım.

Çözüm: **set(eksen,'linewidth',2)**

Örnek: x eksen sayı değerlerini 5 ile 25 arasında 2 şer birim aralıklarla, y eksen sayı değerlerini de -10 ile 50 arasında 3 er birim aralıklarla işaretleyen komutu yazalım.

Çözüm: **set(eksen,'xtick',[5:2:25],'ytick',[-10:3:50])**

Örnek: Kamera pozisyonunu x=1, y=0, z=9 olacak biçimde düzenleyelim.

Çözüm: **set(eksen,'box','on');set(eksen,'cameraposition',[1 0 9]);**

Örnek:

a) Ekranda $y=\sqrt{25-x^2}$ fonksiyonun (yarım çember) grafiğini çizdirelim.

b) Grafiğin tam bir çember parçası değil, U harfine benzediğini görürüz. Şekil penceresinin boyutlarını kareye benzettiğimizde ancak yarım çember tam olarak görünür. Sebebi de şekil penceresine göre değişen **dataaspectratio** oranının 1 olmamasından kaynaklanır. Yarım dairenin, şekil penceresinin boyutlarına bağımlı kalmaksızın tam olarak yarım daire görünmesini sağlayan işlemleri yapalım.

Çözüm:

a) x=-5:0.1:5;y=sqrt(25-x.^2);plot(x,y);

b) sekil=get(0,'currentfigure');

eksen=get(sekil,'currentaxes');

get(eksen,'dataaspectratio');

set(eksen,'dataaspectratio',[1 1 1]);

Not: Aynı işlemi **axis equal** komutuyla da yapabiliriz.

6. image Nesnesi

Eksen nesnesi üzerinde resim dosyalarını görüntülemeye yarayan grafik nesnesidir. Matlab'da resimler **indeksli (indexed)**, **gerçek renkli (true color)** veya **gri tonlu (gray scale)** olabilirler. Bir resmin ölçüleri genişlik ve yüksekliğine düşen piksel sayısı ile belirlenir. Örneğin 150x140 ölçülerinde bir resim nesnesi, yatay 150 piksel, düşey 140 piksel büyüklüğündedir. Bu durumda bu resmi temsil edebilecek matris 150 satır, 140 sütunlu yani 140x150 mertebesindedir.

Matlab Ders Notları

a) İndeksi Resimler: İndeksli resimler, resmin ölçülerinde bir matris ile birlikte mx3 boyutlu bir renk haritası matrisi (Colormap) ile belirlenir. Resmi oluşturan renk sayısı, renk haritasındaki satır sayısı kadardır.

b) Gri Tonlu Resimler: Bu resimler de, indeksli resimler gibidir, ancak renk haritası 2x3 boyutunda bir matristir.

c) Gerçek Renkli Resimler: 150x140 ebadında bir gerçek renkli resim için Matlab 140x150x3 ölçülerinde bir matris ile temsil edilir.

image Nesnesinin Bazı Özellikleri:

AlphaData: Resim ölçülerinde ve resmin her bir pixelinin şeffaflığını tutan bir dizidir.

AlphaDataMapping: AlphaData dizisinin nasıl yorumlanarak kullanılacağını belirleyen özelliktir.

CData: İndeksli resimlerde mxn, gerçek renkli resimlerde ise mxnx3 boyutlarında bir matristir

Parent: Resim nesnesinin ebeveynini belirleyen özelliktir. Daima ebeveyni eksen nesnesidir. Bu durumda **parent** özelliği resim nesnesinin üzerinde görüntülenecek eksen nesnesinin tutamacını belirlemeye yarar.

Selected: Resmi seçili duruma getirmeye yarar. Değeri **on** ise resim seçilir, **off** yapılırsa seçili durum iptal edilir.

SelectionHighlight: Seçili olan resmin kenarlarına 8 adet kulp ile görselleştirmesi sağlanır. Değeri **on** ise resim kulp takılır, **off** yapılırsa kulp durumu iptal edilir.

Tag: Image nesnesine bir etiket ismi vermeye yarayan özelliktir. **get** fonksiyonu ile değerini öğrenebilir **set** fonksiyonu ile istediğimiz bir değeri atayabiliriz.

Type: Grafik nesnesinin türünü tutan özelliktir, her zaman değeri **image** olur.

Visible: Resim nesnesinin görünüp-görünmemesini düzenleyen özelliktir. Değeri **on** ise resim görünür, **off** ise görünmez.

XData: Resim nesnesini oluşturan CData dizisinin eksen nesnesinin x eksenini üzerinde hangi aralığa konumlanacağını belirleyen 1x2 lik bir dizidir. İlk değer başlangıç, ikinci değer de bitiş değerini temsil eder. Varsayılan olarak bu değer [1 size(CData,2)] dir.

YData: Resim nesnesini oluşturan CData dizisinin eksen nesnesinin y eksenini üzerinde hangi aralığa konumlanacağını belirleyen 1x2 lik bir dizidir. İlk değer başlangıç, ikinci değer de bitiş değerini temsil eder. Varsayılan olarak bu değer [1 size(CData,1)] dir.

Örnek: a) iflbmp.bmp adlı resim dosyası hakkında bilgi edinelim

b) Bu resim dosyasını okutalım ve **ans** değişkenini inceleyerek dosyanın hangi tür resim dosyası olduğunu söyleyelim.

c) Bu resim dosyasını Matlab şekil penceresinde görüntüleyelim.

d) resume.tif adlı dosyayı okutarak res adlı değişkene atayalım.

e) res değişkenini incelediğimizde size değerinin 3296x2528 olduğunu varsayalım. Buradan resim dosyası özellikleri hakkında ne diyebiliriz?

Çözüm:

a) `imfinfo('iflbmp.bmp')`
b) `imread('iflbmp.bmp')` komutu sonucu workspace (çalışma alanında) oluan **ans** değişkeninin **size** (boyutu) değerini incelersek; **239x324x3** olduğunu görürüz. Buradan resim dosyasının 324 piksel yatay, 239 dikey piksel boyutlarında ve gerçek renkli (true color) olduğunu anlayabiliriz.
c) `imageview('iflbmp.bmp');`
d) `res=imread('resume.tif');`
e) Resim dosyasının 2528 piksel yatay, 3296 dikey piksel boyutlarında olduğunu anlayabiliriz.
res değişkenini görüntülemeye çalıştığımızda, dosyanın içinde 0 ve 1 lerden oluştuğunu görebiliriz. Bu da resmin 0 (siyah), 1 (beyaz) renklerden oluşan bir resim olduğunu söyleyebiliriz.

Örnek: a) bmp resim formatlı **iflbmp.bmp** resim dosyasını ve gif formatlı **iflgif.gif** adlı resim dosyalarını sırasıyla **res1** ve **res2** değişkenlerine, **colormap** (renk haritalarını) da **harita1** ve **harita2** değişkenlerine atayalım.

b) harita1 ve harita2 renk haritalarını incelediğimizde, harita1 değişkeninin boş bir dizi, harita2 değişkeninin ise 256x3 boyutlarında bir dizi olduğunu görebiliriz. **res1** ve **res2** değişkenlerini resim olarak şekil penceresinde inceleyelim. **res1** in normal görüntülenmesine rağmen, **res2** nin düzgün görüntülenmediğini görürüz. **res2** nin düzgün görünmesi için ne yapılması gerektiğini araştıralım.

Çözüm:

a) `[res1,harita1]=imread('iflbmp.bmp');`

`[res2,harita2]=imread('iflgif.gif');`

b) Renk haritası olan resim dosyalarını düzgün görüntülemek için `imageview` fonksiyonunu kullanırken ikinci parametre olarak renk haritası dosyasını belirtmektir. O halde komutu aşağıdaki gibi kullanmalıyız: **imageview(res4,harita4);**

7. line Nesnesi

Eksen (axes nesnesi) üzerinde verilen x, y (ve z) koordinatları noktayı belirleyen grafik nesnesidir. Noktaların birleşiminden de **çizgi** (line) oluşur. line nesnesi yardımıyla çizginin, rengi, biçimi, kalınlığı ... gibi bir çok özelliği düzenlenebilir.

line Nesnesinin Bazı Özellikleri:

Color: line nesnesinin rengini düzenlemeye yarar.

LineStyle: Çizgi stilini belirleyen özelliktir. Alabileceği değerler; '-', '--', ':-.', ':', ve **none** dir. Değeri none olursa çizgi görünmez.

LineWidth: Çizginin kalınlığını düzenleyen özelliktir.

Marker: Çizginin işaretini belirleyen özelliktir. Marker ile ilgili değerler ve anlamı aşağıdaki tabloda gösterilmiştir.

Değer	Anlamı
+	+ işareti
o	daire işareti
*	yıldız işareti
.	nokta işareti
x	çarpı işareti
s	kare (square) işareti
d	elmas (diamond) işareti

Matlab Ders Notları

^	yukarı gösteren üçgen işareti
v	aşağı gösteren üçgen işareti
>	sağa doğru gösteren üçgen işareti
<	sola doğru gösteren üçgen işareti
p	5 noktalı (pentagon) yıldız işareti
h	6 noktalı (hexagram) yıldız işareti
none	işaretsiz

MarkerEdgeColor: Çizgi ile çizilecek kapalı bölge çeperinin (sınırının) rengini belirleyen özelliktir.

MarkerFaceColor: Çizgi ile çizilecek kapalı bölgenin rengini belirleyen özelliktir.

MarkerSize: Çizgi işaretlerinin büyüklüğünü belirleyen özelliktir. Matlab bu büyüklüğü 6 olarak varsayar.

Selected: Çizgiyi seçili konuma getirmeye yarayan özelliktir. Değeri **on** veya **off** olabilir.

SelectionHighlight: Selected ile seçilmiş çizginin etrafına 8 adet kulp koymaya yarayan özelliktir. Değeri **on** veya **off** olabilir.

Visible: Çizgi nesnesinin görünüp-görünmemesini düzenleyen özelliktir. Değeri **on** ise çizgi görünür, **off** ise görünmez.

XData, YData, ZData: Çizginin x, y, z koordinatlarını belirleyen vektörlerdir.

Örnek: (20,30) noktasını grafik olarak gösteren komutu yazalım.

Çözüm: `line(20,30);`

Örnek: $y=x^2-8x+5$ fonksiyonunun grafiğini x değerlerini [-5 15] aralığında, 0.01 artış ile çizdiren komutları yazalım.

Çözüm:

`x=-5:0.01:15;y=x.^2-8*x+5; line(x,y);`

Not: **line(x,y);** yerine **plot(x,y);** fonksiyonunu da kullanabilirdik.

Örnek:

a) $f(x)=x^3-4x+7$ fonksiyonu ile $g(x)=x\sin x-5\cos x$ fonksiyonlarının grafiklerini, x değerleri [-5 5] arasında çizdiren komutları yazalım.

b) $f(x)$ in grafiğinin rengini mavi, $g(x)$ in grafiğini kırmızı renkte olmasını sağlayalım.

c) $f(x)$ in grafiğinin çizgi biçimini `:`, $g(x)$ in çizgi biçimini `-` lmasını sağlayalım.

d) Çizgi kalınlıklarını 2 şer birim ve 1. grafiği **yukarıya bakan üçgenlerle**, 2. grafiği **yıldızlarla** işaretleyelim.

Çözüm:

a) `x=-5:0.1:5;`

`f=x.^3-4*x+7;g=x.*sin(x)-5*cos(x);`

`gra1=line(x,f);`

`gra2=line(x,g);`

Not: `gra1` ve `gra2` tanımlamaları yapılmadan da grafikler sadece **line(x,f);line(x,g);** komutları ile çizdirilebilir. Ancak grafiklerde başka değişiklikler yapacağımızdan dolayı `gra1` ve `gra2` değişkenlerine (tutamaçlarına) ihtiyaç duyarız.

b) `set(gra1,'color','b');`; `set(gra2,'color','r');`

c) `set(gra1,'linestyle',':');`; `set(gra2,'linestyle','-');`

d) `set(gra1,'linewidth',2,'marker','^');`

`set(gra2,'linewidth',2,'marker','p');`

8. rectangle Nesnesi

Eksen (axes nesnesi) üzerinde belirlenen koordinat ve ölçülerde dikdörtgen çizmeye yarayan grafik nesnesidir.

rectangle Nesnesinin Bazı Özellikleri:

Curvature: Dikdörtgenin köşelerini eğriye dönüştürme oranlarını belirlemeye yarayan 1x2 lik bir dizidir. Sayılar 0 ile 1 arasında değişir. [0 0] dizisi dikdörtgeni korur, [1 1] dizisi elipse dönütürür, dikdörtgen kare ise çembere dönüştür.

EdgeColor: Dikdörtgenin kenar çizgisinin (sınırının) rengini belirleyen özelliktir.

FaceColor: Dikdörtgenin iç bölgesinin rengini belirleyen özelliktir.

LineStyle: Dikdörtgenin kenar stilini belirleyen özelliktir. Alabileceği değerler; `-`, `--`, `-.`, `:`, ve **none** dir. Değeri **none** olursa dikdörtgen görünmez.

LineWidth: Dikdörtgenin kenar kalınlığını düzenleyen özelliktir.

Position: Dikdörtgenin şekil penceresi üzerindeki pozisyonu ve ölçülerini belirleyen özelliktir. İlk iki değer dikdörtgenin sol alt köşesinin şekil penceresinin sol alt köşesine göre koordinatlarını, üçüncü sayı dikdörtgenin yatay uzunluğunu dördüncü sayı da dikey uzunluğunu belirtir.

Selected, SelectionHighlight, Visible: Bu özellikler de **line** nesnesinin özellikleri ile aynıdır.

Örnek: Tutamaç değeri `dd` olan, şekil penceresinin sol alt köşesine göre koordinatları 5 5, yatay uzunluğu 6 dikey uzunluğu 4 birim olan kırmızı kenarlı dikdörtgeni çizdiren komutu yazalım..

Çözüm: `dd=rectangle('position',[5 5 6 4],'edgecolor','r');`

Örnek: Eksen nesnesinin boyutları dikdörtgenin boyutlarında olduğu için, dikdörtgen eksenini kapladı. Eksen nesnesinin boyutlarını, x eksenini 15 birim, y eksenini 10 birim olarak düzenleyelim

Çözüm: Bunun için **rectangle** nesnesinin ebeveyni olan eksen nesnesinin tutamacına ihtiyacımız var. Daha sonra **xlim** ve **ylim** değerlerini **set** fonksiyonu ile düzenleyebiliriz. Buna göre komut şöyle olmalıdır: **set(get(dd,'parent'),'xlim',[0 15],'ylim',[0 10]);**

9. patch (yama) Nesnesi

Bir çok kenarlardan oluşan iki veya üç boyutlu karmaşık şekli görselleştirmeye yarayan, Matlab'ın en zor kullanılan grafik nesnesidir. Bu nesne aslında çok kenarlıdır. Bir yama en az üç olmak üzere bir çok **bağlantı noktasından (vertices)** oluşur. Yama olarak adlandırılmasının sebebi, sadece düzlemsel değil üç boyutlu olabilmesidir. İlk bağlantı noktası ile son bağlantı noktası birleştirildiğinde **yama** tamamlanmış olur. Yamanın iç bölgesi ise **yüz (face)** olarak adlandırılır.

patch Nesnesinin Bazı Özellikleri:

CData: Yama rengini tanımlayan özelliktir. Her bir bağlantı noktası (vertex), yüz (face) veya bütün yama (patch) için ayrı ayrı renk tanımlanabilir.

Matlab Ders Notları

DiffuseStrength: Yama üzerine düşen ışığın yayılma yoğunluğunu belirten 0 ile 1 arasında bir sayıdır.

EdgeAlpha: Yamanın kenar çizgilerinin şeffaflığını belirleyen 0 ile 1 arasında değişen sayılara karşılık gelen özelliktir.

EdgeColor: Yamanın kenar çizgilerinin rengini belirleyen özelliktir.

FaceAlpha: Yama yüzlerinin şeffaflığını belirleyen 0 ile 1 arasında değişen sayılara karşılık gelen özelliktir.

FaceColor: Yama yüzlerinin rengini belirleyen özelliktir.

Faces: Bir yüzeyin **vertices** dizisinin hangi satırındaki bağlantı noktalarından oluşturulacağını belirleyen özelliktir.

LineStyle, LineWidth: rectangle nesnesinde açıklanan özelliklerle aynıdır.

Marker: Bağlantı noktalarının işaretini belirleyen özelliktir.

MarkerEdgeColor: Bağlantı noktalarında görünmesi istenen kapalı şekildeki işaretlerin çerper (sınır) rengini belirleyen özelliktir.

MarkerFaceColor: Bağlantı noktalarında görünmesi istenen kapalı şekildeki işaretlerin iç bölgesi rengini belirleyen özelliktir.

MarkerSize: Bağlantı noktalarında görünmesi istenen kapalı şekildeki işaretlerin büyüklüğünü belirleyen özelliktir..Matlab bu büyüklüğü 6 olarak varsayar.

Selected, SelectionHighlight, Visible: Dikdörtgen nesnesindeki özelliklerle aynıdır.

XData, YData, ZData: Bağlantı noktalarının x, y, z koordinatlarını belirleyen vektörlerdir.

Örnek: Düzlemde adı **yama** olan, (1,3), (2,5), (4,7), (7,5), (3,2) koordinatlı noktalardan oluan yama esnesini oluşturalım.

Çözüm: Önce x ve y koordinatlarının dizilerini oluturalım
 $x=[1;2;4;7;3]; y=[3;5;7;5;2];$

Yama düzlemsel olacağından z koordinatları 0 lardan oluşacaktır.

$z=[0;0;0;0;0];$

Sonra yamayı oluşturan komutu yazabiliriz.

$yama=patch('xdata',x,'ydata',y,'zdata',z);$

Not 1) z vektörünü; $z=zeros(size(x,1),1);$ biçiminde de oluşturabiliriz.

Not 2) Aynı yamayı, köşe koordinatlarından oluşan **k** matrisinde tutarak ta gerçekleştirebiliriz.

$k=[1\ 3\ 0;2\ 5\ 0;4\ 7\ 0;7\ 5\ 0;3\ 2\ 0];$

$yama=patch('vertices',k);$

Örnek: Önceki örnekte oluşturulan tutamaç adı **yama** olan yamanın, dış çerper rengini kırmızı, iç bölge rengini mavi renkli yapalım.

Çözüm: $set(yama,'edgecolor','r','facecolor','b');$

10. surface Nesnesi

x ve y değişkenlerine bağlı z fonksiyonunun oluşturduğu ($z=f(x,y)$) yüzeyi görselleştiren nesnedir.

surface Nesnesinin Bazı Özellikleri:

CData, EdgeAlpha, EdgeColor, FaceAlpha, FaceColor, LineStyle, LineWidth, Marker, MarkerSize, Selected, SelectionHighlight, Visible,

XData, YData, ZData: Bu özellikler **patch** nesnesinin özellikleri ile aynıdır.

Vertices: Yamanın bağlantı noktalarını koordinatlarını tutan, (x, y, z) üçlülerinin matrisi.

Örnek:

Örnek: Bu kadar öğrendiklerimizle, basit bir grafik arabirimli program yazalım.Sol alt köşesi, ekranın sol alt köşesinden yatay uzaklığı 5 cm, düşey uzaklığı 8 cm, yatay uzunluğu 15 cm, yüksekliği 10 cm olan **sekil** adlı **figure** penceresinde, pencerenin sol alt köşesinden 1 cm yatay, 5 cm düşey uzaklığında, 10 cm uzunluğunda, 1 cm yüksekliğinde **mesaj** adlı **static text** nesnesine, 20 punto büyüklüğünde **İzmir Fen Lisesi** yazdıran program yazalım.

Çözüm:

```
function mesaj;  
sekil=figure('units','centimeters','position',[5 8 15 10]);  
mesaj=uicontrol('style','text','units','centimeters','position',  
[1 5 10 1],'string','İzmir Fen Lisesi','fontsize',20);
```

Klavyeden Giriş Alma:

Bunun için **inputdlg** fonksiyonu kullanılır.

Genel kullanımı aşağıdaki gibidir:

girdiler={'1.girdi başlığı','2. girdi başlığı','3. girdi başlığı...'};
değişken=inputdlg(girdiler);

Örnek: Klavyeden girilen bir mesajı, bir önceki örnekteki figure penceresindeki pozisyonda, beyaz zeminde , mavi renkli olarak yazdıran bir program yazalım.

Çözüm:

```
function mesaj2;  
sekil=figure('units','centimeters','position',[5 8 15 10]);  
mesaj={'Mesaj: '};  
cevap=inputdlg(mesaj);  
yazi=uicontrol('style','text','units','centimeters',  
'position',[1 5 10 1], 'fontsize',20,'backgroundcolor','w',  
'foregroundcolor','b'); set(yazi,'string',cevap);
```

Not: Matlab'da renk tanımlaması bir kaç yoldan yapılabilir.Birincisi, RGB (Red Green Blue) renk oranlarını 0 ile 1 arasında olacak şekilde sayılar vererek oluşturmaktır.İkinci olarak bazı özel renkleri, ister tamamını istenirse baş harfini yazarak tanımlamaktır.Örneğin; kırmızı 'red' veya 'r' ile, yeşil 'green' veya 'g' ile, mavi 'blue' veya 'b' ile,... belirtilebilir. Aşağıdaki tablo renk tanımlamaları için bir fikir verebilir:

RGB değeri	Kısa Kul.	Uzun Kul.
[1 0 0]	'r'	'red'
[0 1 0]	'g'	'green'
[0 0 1]	'b'	'blue'
[1 1 0]	'y'	'yellow'
[1 0 1]	'm'	'magenta'
[0 1 1]	'c'	'cyan'
[1 1 1]	'w'	'white'
[0 0 0]		'black'

Örnek: Klavyeden girilen iki sayının çarpımını, beyaz zeminde , mavi renkli olarak; örneğin sayılar, -13 ve 7 girilmişse; figure penceresinde **-13 x 7 = 91** biçiminde yazdıran bir program yazalım.

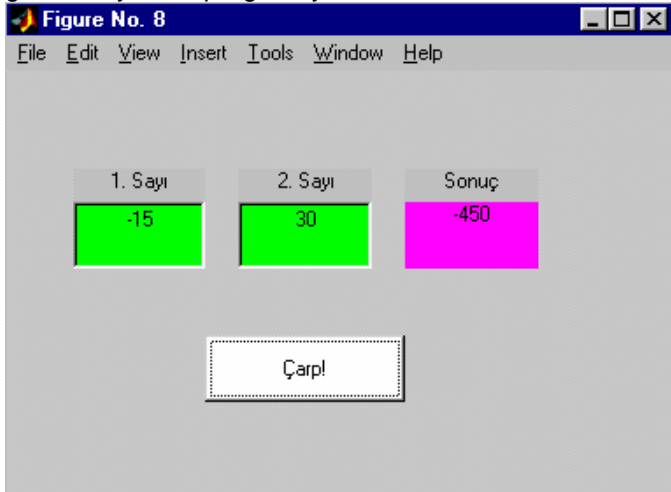
Matlab Ders Notları

Çözüm:

```
function ikisayicarpimi;
sekil=figure('units','centimeters','position',[5 8 15 10]);
sorular={'1. Sayı: ','2. sayı: '};
cevaplar=inputdlg(sorular);
sayi1=cevaplar{1};sayi2=cevaplar{2};
sayi1=str2num(sayi1);sayi2=str2num(sayi2);
carpim=sayi1*sayi2;carpim=num2str(carpim);
```

```
yazi1=uicontrol('style','text','units','centimeters',
'position',[1 5 2 1], 'fontsize',20,'backgroundcolor','w',
'foregroundcolor','b'); set(yazi1,'string','1. sayı');
yazi2=uicontrol('style','text','units','centimeters',
'position',[3 5 1 1], 'fontsize',20,'backgroundcolor','w',
'foregroundcolor','b'); set(yazi2,'string','x');
yazi3=uicontrol('style','text','units','centimeters',
'position',[4 5 2 1], 'fontsize',20,'backgroundcolor','w',
'foregroundcolor','b'); set(yazi3,'string','2. sayı');
yazi4=uicontrol('style','text','units','centimeters',
'position',[6 5 1 1], 'fontsize',20,'backgroundcolor','w',
'foregroundcolor','b'); set(yazi4,'string','=');
yazi5=uicontrol('style','text','units','centimeters',
'position',[7 5 5 1], 'fontsize',20,'backgroundcolor','w',
'foregroundcolor','b'); set(yazi5,'string','carpim);
```

Örnek: Aşağıdaki şekil penceresinde görüldüğü gibi, 1. sayı ve 2. sayıyı girdikten sonra, **Çarp** butonuna basıldığında çarpım sonucunu, **Sonuç** bölümünde görüntüleyen bir program yazalım.

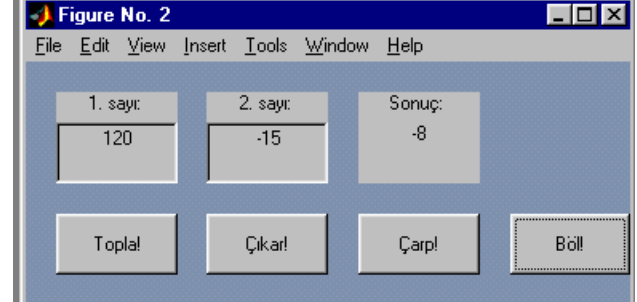


Çözüm:

```
function carpma;
sekil=figure('units','centimeters','position',[1 1 12 8]);
uicontrol('units','centimeters','style','text',
'position',[1 5.5 2 1], 'string','1. Sayı');
uicontrol('units','centimeters','style','text',
'position',[3.5 5.5 2 1], 'string','2. Sayı');
uicontrol('units','centimeters','style','text',
'position',[6 5.5 2 1], 'string','Sonuç');
sayi1=uicontrol('units','centimeters','style','edit','position',[
1 5 2 1], 'backgroundcolor',[0 1 0], 'string','', 'tag','sayi1');
sayi2=uicontrol('units','centimeters','style','edit','position',[
3.5 5 2 1], 'backgroundcolor',[0 1 0], 'string','', 'tag','sayi2');
sonuc=uicontrol('units','centimeters','style','text','position',[
6 5 2 1], 'backgroundcolor',[1 0 1], 'tag','sonuc');
carpimdugmesi=uicontrol('units','centimeters',
'style','pushbutton','position',[3 3 3 1], 'backgroundcolor',
[1 1 1], 'string','Çarp!', 'callback', @carpim);
function carpim(tutamac, bos);
sayi1=findobj(gcf, 'Tag','sayi1'); x=get(sayi1, 'string');
x=str2num(x);
sayi2=findobj(gcf, 'Tag','sayi2'); y=get(sayi2, 'string');
```

```
y=str2num(y);
sonuc=findobj(gcf, 'tag','sonuc'); s=x*y; s=num2str(s);
set(sonuc, 'string', s);
```

Örnek: Aşağıdaki şekil penceresinde görüldüğü gibi, 1. sayı ve 2. sayıyı girdikten sonra, **Topla!**, **Çıkar!**, **Çarp!** ve **Böl!** butonlarından hangisine basılırsa, ilgili işlemin sonucunu **Sonuç** nesnesinde görüntüleyen bir program yazalım. Örneğin 1. sayı ve 2. sayı bölümlerine sırasıyla 120 ve -15 girilip Böl! butonuna basıldığında, Sonuç bölümünde -8 görüntülensin.



Çözüm:

```
function dortislem;
sekil=figure('units','centimeters',
'position',[1 1 10 7], 'color',[0.5 0.6 0.7]);
uicontrol('units','centimeters','style','text',
'string','1. sayı', 'position',[0.5 5.5 2 1]);
uicontrol('units','centimeters','style','text',
'string','2. sayı', 'position',[3 5.5 2 1]);
uicontrol('units','centimeters','style','text',
'string','Sonuç', 'position',[5.5 5.5 2 1]);
sayi1=uicontrol('units','centimeters','style','edit',
'position',[0.5 5 2 1], 'tag','sayi1');
sayi2=uicontrol('units','centimeters','style','edit',
'position',[3 5 2 1], 'tag','sayi2');
sonuc=uicontrol('units','centimeters','style','text',
'position',[5.5 5 2 1], 'tag','sonuc');

topla=uicontrol('units','centimeters','style','pushbutton',
'position',[0.5 3.5 2 1], 'tag','topla',
'string','Topla!', 'callback', @topla);
cikar=uicontrol('units','centimeters','style','pushbutton',
'position',[3 3.5 2 1], 'tag','cikar',
'string','Çıkar!', 'callback', @cikar);
carp=uicontrol('units','centimeters','style','pushbutton',
'position',[5.5 3.5 2 1], 'tag','carp',
'string','Çarp!', 'callback', @carp);
bol=uicontrol('units','centimeters','style','pushbutton', 'posi
tion',[8 3.5 2 1], 'tag','bol', 'string','Böl!', 'callback', @bol);
```

```
function toplu(tut,bosver);
sayi1=findobj(gcf, 'tag','sayi1'); x=get(sayi1, 'string');
x=str2num(x);
sayi2=findobj(gcf, 'tag','sayi2'); y=get(sayi2, 'string');
y=str2num(y);
s=x+y; s=num2str(s); sonuc=findobj(gcf, 'tag','sonuc');
set(sonuc, 'string', s);
```

```
function cikar(tut,bosver);
sayi1=findobj(gcf, 'tag','sayi1'); x=get(sayi1, 'string');
x=str2num(x);
sayi2=findobj(gcf, 'tag','sayi2'); y=get(sayi2, 'string');
y=str2num(y);
s=x-y; s=num2str(s); sonuc=findobj(gcf, 'tag','sonuc');
set(sonuc, 'string', s);
```

```
function carp(tut,bosver);
```

Matlab Ders Notları

```
sayi1=findobj(gcf,'tag','sayi1');x=get(sayi1,'string');  
x=str2num(x);  
sayi2=findobj(gcf,'tag','sayi2');y=get(sayi2,'string');  
y=str2num(y);  
s=x*y;s=num2str(s);sonuc=findobj(gcf,'tag','sonuc');  
set(sonuc,'string',s);
```

```
function bol(tut,bosver);  
sayi1=findobj(gcf,'tag','sayi1');x=get(sayi1,'string');  
x=str2num(x);  
sayi2=findobj(gcf,'tag','sayi2');y=get(sayi2,'string');  
y=str2num(y);  
s=x/y;s=num2str(s);sonuc=findobj(gcf,'tag','sonuc');  
set(sonuc,'string',s);
```

Derleme Nedir ve Matlab Dosyaları Nasıl Derlenir?

Bir programlama dili ile yazılan bir programın, hazırlanan ortamdan bağımsız olarak çalıştırılması için, yapılan işleme **derleme (compile)** denir. Böylece ilgili dilde yazılan komut ve deyimler, işletim sisteminin yorumlayıp çalışmasını sağlayan komutlar topluluğu haline gelir ki, bu dosyaların genel olarak uzantısı **exe** (executable) olur.

Aynı mantıkla, Matlab dosyalarını da derleyerek, matlab ortamı dışında çalıştırılabilir dosyalar biçimine getirmek için derlemek gerekir. Ancak matlab **m fonksiyon** dosyaları derlenebilir, **m dosyaları** derlenemezler!

Derleme komutu ve kullanımı:

Derleme komutu **mcc** (Matlab C/C++) dir.

Komut satırında kullanımı aşağıdaki gibidir:

mcc -seçenek dosyaadı

Burada dosyaadı yerine work klasörüne kaydedilmiş m fonksiyon dosyasının adı yazılır. Seçenek yerine ise aşağıdaki ifadeler yazılabilir.

Seçenek	Açıklama
-m	C tabanlı exe uzantılı uygulama oluşturmaya yarar
-p	C++ tabanlı exe uzantılı uygulama oluşturmaya yarar
-B sgl	C tabanlı grafik kütüphanesi ve grafik arabirimli matlab uygulama dosyaları oluşturmaya yarar.
-B sgldcpp	C++ tabanlı grafik kütüphanesi ve grafik arabirimli matlab uygulama dosyaları oluşturmaya yarar.

Örnek: Temiz ekrana **İyi Günler!** Mesajı yazdıran bir programı DOS ortamında çalıştırabilir biçime getiren işlemleri yapalım.

Çözüm: Önce aşağıdaki m dosyasını yazalım.

```
clc; disp('İyi Günler!');
```

Bu dosyayı **iyigun.m** adıyla work dizinine kaydedelim. Komut satırında **iyigun** yazıp (enter) tuşuna basarak çalışma sonucunu görebiliriz.

Bu dosyayı **mcc -m iyigun** komutuyla derlemeye çalıştığımızda; aşağıdaki hata mesajlarını görürüz:

mcc -m iyigun

??? Error: File "iyigun" is a script M-file and cannot be compiled with the current Compiler.

Error in ==> C:\MATLAB6P5\toolbox\compiler\mcc.dll

Çünkü derlemeye çalıştığımız dosya bir m dosyasıdır, m fonksiyon dosyası değil! Bunun için bu dosyayı m fonksiyon dosyası biçimine getirmemiz gerekir. Bunun için aşağıdaki basit eklentileri yapıp dosyayı yeniden kaydedelim:

```
function laf=iyigun  
clc; disp('İyi Günler!');  
laf=1;
```

Sonra **mcc -m iyigun** komutuyla derleyelim. Bu durumda hiçbir hata mesajı ile karşılaşmadan derlendiğini ve work klasöründe **iyigun.exe** adlı dosyanın oluştuğunu görebiliriz. Dos ortamında **iyigun** yazıp enter'e basarak dosyanın çalışmasını görebiliriz