

BİLGİSAYAR PROGRAMLAMA
MATLAB
DERS NOTU

Prof. Dr. Mehmet ARDIÇLIOĞLU
2017

PROGRAMLAMA MANTIĞI

Program: Belirli bir amaca yönelik olarak hazırlanmış ve bilgisayardan yerine getirilmesi istenen bir dizi komut ya da işlem adımlarının tümüne program adı verilir.

- Program bir işin nasıl yapılacağını gösteren emirler dizisidir.
- Problemin tanımı ve analizinin tam olarak yapılması gereklidir.
- Bu amaçla aşağıdaki sorulara cevap aramak gerekir

1-Programda yapmak istediğimiz (programın görevi) nedir.

2- Gerekli giriş bilgileri nelerdir.

3- Programın çıktıları ne olacaktır.

4- Sonucu elde etmek için gerekli işlemler.

5- Program kaç kez kullanılacak.

Örnek: $ax^2+bx+c=0$ formundaki ikinci dereceden bir bilinmeyenli denklemin çözümünü inceleyelim

1. a, b,c değerlerinin bilinmesi gereklidir
2. Diskriminantın değeri hesaplanmalıdır $\Delta=b^2-4ac$
3. $\Delta < 0 \rightarrow$ reel kök yoktur.
4. $\Delta=0 \rightarrow$ kökler çakışıktır $x_1 = x_2 = -b/2a$
5. $\Delta > 0 \rightarrow$ çözüm kümesi iki elemanlıdır
6. $x_{1,2} = \frac{-b \pm \sqrt{\Delta}}{2a}$ olarak bulunacaktır.

Algoritma; bir problemin nasıl çözüldüğünü gösteren işlem adımlarıdır.

Algoritma Hatırlama Kuralları:

1-herhangi bir bilgisayar programında, aksi belirtilmedikçe program satırları yukarıdan aşağıya doğru sırayla icra edilecektir. Ancak programın akışı içerisinde belirli bir şarta bakılarak yada herhangi bir şarta bakılmaksızın satırlar arasında yönlendirme ile program akışını değiştirmek mümkündür. Program akışını değiştirmek için kullanılacak algoritma **GİT** kelimesidir. Bu komut kullanılırken hangi satıra gidilmesi gerektiği de belirtilmelidir. Dolayısıyla program akışı içerisinde tekrar dönülmesi gereken satırlar varsa, bunlara birer tam

sayı olarak satır numarası verilmesi zorunludur. Satır numaraları verilirken pozitif tam sayılarda istenen değeri kullanmak mümkündür.

.....
5 A' ya değer al
B=2*A
Yaz A, B' nin değerini
Git 5'e

2-Bir algoritma sonlu sayıda adımdan sonra sona erebilmektedir. Yani algoritmanın işleyişinin bir yerde durması gerekir.

Örneğin;
5 A' ya değer al
B=2*A
Yaz A' nın değerini
Git 5'e

Programı çalıştırmaya başlayınca A değişkeni için kullanıcıdan bir değer isteyecek, sonraki adımda aldığı bu değerini iki katını B değişkenine aktaracak ve devamındaki adımda **GİT 5'e** komutu ile tekrar başa dönecek ve aynı işlemleri tekrarlayacaktır. Yani program akışı sürekli bir döngüye girecek ve program bir yerde durmayacaktır.

Bu durum bir hatadır. Belirli şartlara göre algoritmanın işleyişini durduran kontrollerin program içerisine dahil edilmesi gerekir.

3-Algoritmanın programla ilgili olarak ortaya çıkabilecek her türlü durumu içermesi gerekir. Bilgisayarın program içerisinde istenmeyen bir durumla karşılaşması halinde, yorum yada tahmin yapmak gibi bir kabiliyeti bulunmamaktadır. Dolayısıyla problemin çözümü sırasında karşılaşılabilecek her türlü durumda yapılması gereken işlemler program içerisinde verilmelidir.

4-Algoritma aynı tür problemler içinde geçerli yani genel olmalıdır. Yani ikinci dereceden bir denklemin çözümünde, problemin bazı sabitler için değil de benzer türdeki problemlerin de çözümünü içerecek şekilde bir düzenleme yapılmalıdır.

Örnek: bilgisayarda girdi olarak verilen 10 sayının toplamını hesaplayıp, sonucu kullanıcıya yazılı olarak veren algoritma:

Herhangi bir problemin çözümünde birden fazla yol olabileceği unutulmamalıdır. Bu yollardan en basit ve en sade olanı tercih edilmelidir. Yapılması gereken ilk iş problemi tanımak yani analiz etmektir.

Burada amaç bilgisayarın kullanıcıdan alacağı herhangi 10 adet sayının toplamını bulması ve sonucu yazılı olarak vermesidir. Yani programda istenenler:

1-10 sayı veri olarak alınmalı

2-Sayılar toplatılmalı

3-Sonuç yazılı olarak verilmeli

Bu problemde çözüm yolu tek değildir. En basit iki çözüm:

1-10 adet değişken tanımlanarak değerleri aynı anda bilgisayara verip bunların toplamı bulunabilir.

A, B, C, D, E, F, G, H, I, J, değişkenleri için doğrudan alınabilir ve

$$\mathbf{TOP=A+B+C+D+E+F+G+H+I+J}$$

İşlemi yaptırılarak sonuç bulunabilir.

2-Bir değişken tanımlanarak her defasında bu değişkene bir değer aldırılır ve alınan değerler bir toplama ilave edilebilir.

A değişkenine değer alınabilir.

$\mathbf{TOP=TOP+A}$ işlemi ile \mathbf{TOP} ' un önceki değerlerinin üzerine \mathbf{A} değeri eklenir.10 sayı toplanmışsa sona \mathbf{TOP} yazılır.

Değişik çözüm yolları olması halinde en kolay ve/veya en az bir bilgisayar hafızası gerektiren çözüm seçilmelidir

Uygun olan ikinci çözüm için program algoritması:

Başla

$\mathbf{TOP=0}$

5 A değişken değerini al

$\mathbf{TOP=TOP+A}$

Alınan sayı adedi 10' dan az ise GİT 5' e

\mathbf{TOP} ' un değerini yaz

Dur

Bu problemde ;

Alınan sayı adedi 10' dan az ise GİT 5'e komut satırı açık değildir. Çünkü alınan sayı adedinin 10' dan az olup olmadığı nasıl bilinecektir? Programcının ben olsam nasıl yapardım sorusuna cevap araması gereklidir. Cevap: "her sayı aldığımda sayardım ve sayı adedi 10' dan az ise tekrar sayı alıp aynı işlemi tekrarlardım." olacaktır.

Yani programa alınan sayı adedini sayan bir sayaç koymak ve sayaç ile 10 değerini karşılaştıran mantıksal bir komut kullanmak yeterli olacaktır.

Algoritmayı doğru ve eksiksiz olarak yazarsak:

Başla

N=1 (N değişken sayaç)

TOP=0 (top değişkeni toplam)

5 A değişkenine değer al

TOP=TOP+A

N=N+1

Eğer N<=10 İSE git 5' e

Sonucu (TOP değerini)yaz

Dur

Program Yaparken Karşılaşılan Hata Türleri

1. **Yazılım Hatası (Syntax Error):** Kullanılan programlama diline ait tanımlar dışındaki yazılımlar hata mesajını verir. Örneğin PRINT yerine PRINTT yazılması gibi.
2. **Mantık hatası (Logic Error):** Örneğin $y=ax+b$ şeklindeki bir denklemi çözmek istediğimizde $y=ax-b$ ifadesi girilmiş ise program herhangi bir hata mesajı vermeden çalışacak ve bir sonuç verecektir. Burada oluşan durum sonucun bulunmasına engel olmadığı için bilgisayar açısından bir hata olmayacaktır. Ancak verilen sonuç amaç dışı olacaktır.

Sabit ve Değişken Kavramı:

Programda değeri hiç değişmeyen gösterimlerdir. Örneğin $y=2x+1$ ifadesinde 2 ve 1 her zaman aynı matematiksel anlamda birer sabittir.

Değişik zamanlarda farklı değerler alabilen gösterimlere değişken adı verilir. BASIC alfabetinde değişkenler A-Z arası alfabetik harflerle başlayıp istenirse 0-9 arası rakamlarla devam edebilir. Örneğin TOPLAM, AD5, SN3,SAY birere değişken adıdır. Değişkenler

sadece sayısal deęerler almaz. Deęişkenler sayısal deęişkenler ve alfa sayısal deęişkenler olmak üzere iki guruba ayrılır.

İçerikleri matematiksel büyüklükler olan ve matematiksel işlemlere girebilen deęişkenler sayısal deęişkenler olarak adlandırılır.

İçerikleri matematiksel bir anlam ifade etmeyen, karakter yada karakter guruplarından oluşan deęişkenler ise alfa sayısal deęişken olarak adlandırılır. BASIC programlama dilinde bir deęişkenin alfa sayısal olduğunu belirtmek için deęişken adının yanına \$ işareti eklenir. Örneğin AD\$, SOY\$, C\$ gibi deęişkenler alfa sayısaldır. Alfa sayısal deęişkenler çarpma, bölme gibi matematiksel bir işleme girmez.

Atama Deyimi:

Deęişken=Aritmetik ifade

= işaretine aktarma yada atama deyimi adı verilir. = işaretinin sağında aritmetik ifadenin gösterdiği deęer , sol tarafta ise deęişkenin adı yazılır.

TP=1	TP=1
SN=3	SN=3
KT1= TP + SN	KT1=4

Sayaç Kavramı:

$K=K+1$ komut satırında , K nın deęeri 1 artırılır ve sonucu K deęişkenine yeni deęer olarak atar.

K=1	
K=K+1	K=2

Sayaç deęerinin her defasında ne kadar artacağı veya azalacağı belirlenir.

K=20	
K=K-2	K=18
	K=16

Örnek: Kullanıcının vereceęi iki sayıdan büyük olanı bulan algoritma

A ve B deęişkenine deęer al

EĞER $A>B$ ise Git 10 a

YAZ “ İkinci sayı daha büyüktür”

Git 15 ‘e

10 YAZ “ Birinci sayı daha büyüktür”

15 DUR

Matematiksel İşlemler

İşlem	Sembol	Matematiksel Gösterim	Basic Gösterimi
Toplama	+	$Y=a+b$	$Y=A+B$
Çıkarma	-	$Y=a-b$	$Y=A-B$
Çarpma	*	$Y=ab$	$Y=A*B$
Bölme	/	$Y=a/b$	$Y=A/B$
Üs alma	^	$Y=x^b$	$Y=X^B$

Örnek: $y=2a+3b^4$ $Y=2*A+3*B^4$
 $y=bd+d/k$ $Y=B*D+D/K$

İşlem Öncelikleri:

1. Parantez içleri
 2. Hazır fonksiyonlar
 3. Üs alma
 4. Çarpma bölme
 5. Toplama çıkarma
- Herhangi bir satırda birden fazla aynı önceliğe sahip işlem bulunursa bu durumda işlem önceliği soldan sağa doğru kullanılır.
 - Gerekli yerlerde parantez kullanılır Açılan parantez kadar kapatmak suretiyle istenildiği kadar parantez kullanmak mümkündür.

Örnek: $y=ab+\frac{c+k}{d-a}$ ifadesini Basic formuna uygun olarak yazınız.

$Y=A*B+C+K/D-A$ Yanlış $y=ab+c+\frac{k}{d}-a$

$Y=A*B+(C+K)/(D-A)$ Doğru

A- Hazır Fonksiyonlar:

İşlem	Fonksiyon Adı	Örnek
Mutlak Değer(x)	ABS	ABS(X)
Karekök (\sqrt{x})	SQR	SQR(X)
Exponansiyel (e^x)	EXP	EXP(X)
Logaritma (Ln x)	LOG	LOG(X)
Sinüs (Sin x)	SIN	SIN(X)
Cosinüs (Cos x)	COS	COS(X)
Tanjant (Tan x)	TAN	TAN(X)
Tamdeğer(x)	INT	INT(X)
İşaret fonk. (Sign x)	SGN	SGN(X)

Örnek:

$$y = \sin 2x + \sqrt{b + a^2}$$

$$Y = \text{SIN}(2 * X) + \text{SQR}(B + A^2)$$

$$y = e^{3x^2} + |4x^3 + 5x|$$

$$Y = \text{EXP}(3 * X^2) + \text{ABS}(4 * X^3 + 5 * X)$$

$$y = 2 \cos 4x + \frac{e^{(4x)} + \sqrt{3x^3 - 1}}{2x^2 + 5x}$$

$$Y = 2 * \text{COS}(4 * X) + (\text{EXP}(4 * X) + \text{SQR}(3 * X^3 - 1)) / (2 * X^2 + 5 * X)$$

B- Mantıksal Karşılaştırmalar:

İşlem	Sembol
Eşit	=
Eşit Değil	<>
Küçüktür	<
Büyüktür	>
Küçük eşit	<=
Büyük eşit	>=

Mantıksal karşılaştırmalarda birden fazla şartı bağlamada kullanılan iki bağlaç **VE (AND)** ile **VEYA (OR)** dir.

P1 ve P2 birer mantıksal şart olmak üzere bu bağlaçların doğruluk tablosu aşağıdaki gibidir.

P1	P2	P1 AND P2		P1	P2	P1 OR P2
Yanlış	Yanlış	Yanlış		Yanlış	Yanlış	Yanlış
Yanlış	Doğru	Yanlış		Yanlış	Doğru	Doğru
Doğru	Yanlış	Yanlış		Doğru	Yanlış	Doğru
Doğru	Doğru	Doğru		Doğru	Doğru	Doğru

Örnek: -1-7 arası tamsayıların toplamını bulan algoritma

Burada yapılacak iş $1+2+3+4+5+6+7$, sayılarının toplanmasıdır. 1 den başlayıp 1 er artarak 7 ye kadar devam eden sayılar basit bir sayaçla elde edilir. Böylece değer üretecek ve bu değerler bir yerde toplanacaktır.

X=0

T=0

5 X=X+1

T=T+X

X ≥ 7 ise GİT 8 'e

GİT 5'e

8 YAZ T değerini

DUR

Örnek 20'den 50'ye kadar olan sayıların toplamını bulan algoritma yazınız.

S=20 : T=0 (Sayı 20 T değeri 0 ile başla)

1 T=T+S (T'ye sayıyı ekle T'yi göster.)

S=S+1 (Sayıyı bir artır.)

S<50 ise 1'e git. (Eğer sayı 50'den küçük ise 1. Satıra git)

T'yi göster. (T'nin değerini göster.)

Dur

Örnek Bir tamsayının faktöriyelini hesaplayan algoritmayı yazınız.

Girdi : Bir tamsayı, n Çıktı : sayının faktöriyel fakt

İlgili formül: Faktöriyel(n)=1*2*...*n

N sayısını al.

FAKT = 1

SAYAÇ = 1

4 Eğer N <= SAYAÇ ise 8. e' a git.

SAYAÇ = SAYAÇ + 1

FAKT = FAKT * SAYAÇ

4. satır'a git.

8 N yaz.

FAKT yaz.

DUR.

Örnek Klavyeden girilen 10 tane sayının toplamını ve ortalamasını bulan algoritmayı yazınız.

SAYAÇ = 0

TOP = 0

3 Bir SAYI al

SAYAÇ = SAYAÇ + 1

TOP = TOP + SAYI

Eğer SAYAÇ = 10 ise 8. Adım' a git.

3. Adım' a git.

8 ORT = TOP / SAYAÇ

TOP yaz.

ORT yaz.

DUR

Örnek İkinci dereceden bir Polinomunu köklerini bulan algoritmayı yazınız.

$$y=ax^2+bx+c$$

POLİNOMUN KATSAYILARINI GİRİN A, B, C

DISK = B ^ 2 - 4 * A * C

EĞER DISK > 0 Git 10

EĞER < 0 Git 20

X = -B / IKIA

YAZ " KÖKLER ÇAKIŞIKTIR, X1=X2= "; X

Git 30

10 AD = SQR(DISK)

X1 = (-B + AD) / IKIA

X2 = (-B - AD) / IKIA

YAZ X1, X2

Git 30

20 YAZ " KÖKLER KOMPLEKSTİR:"

30 END

Örnek: X= 2B+3A+1 eşitliğini A'nın 0-3 arası B'nin 0-4 arası bütün tam değerleri için hesaplayan ve her bir A,B,X değerini yazan algoritma

A=0

40 B=0

10 X=2B+3A+1

YAZ A,B,X

EĞER B=4 ise GİT 20 ye

B=B+1

GİT 10 'a

20 EĞER A=3 ise GİT 30 'a

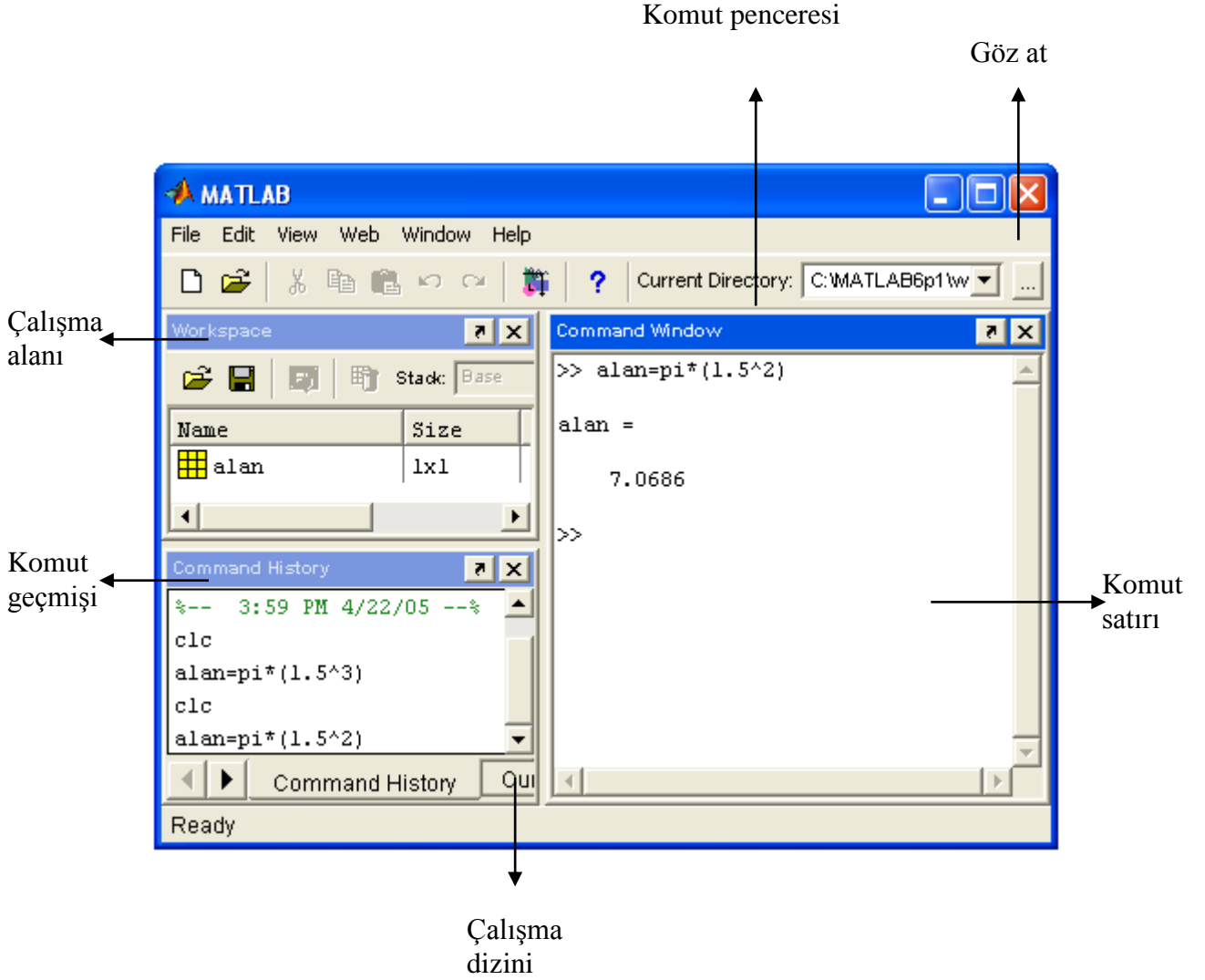
A=A+1

GİT 40'a

30 DUR

MATLAB

(**MAT**rix **LAB**rotary):1985 de C.B Moler tarafından geliştirilmiş ve özelliklele matris esaslı matematik ortamında kullanılabilen etkileşimli bir paket programlama dili olarak tanımlanabilir. C dilinde hazırlanmıştır.



Matlab Arayüzü

Komut satırına yazılan komutlar arasında gezmek için

↑, CTRL+P	: Bir önceki satırı yazar
↓, CTRL+N	: Bir sonraki satırı yazar
←, CTRL+B	: Bir karakter sola kayar
→, CTRL+F	: Bir karakter sağa kayar
CTRL+L	: Bir kelime sola kayar
CTRL+R	: Bir kelime sağa kayar
Home, CTRL+A	: Satır başına gider
End, CTRL+E	: Satırın sonuna gider
Esc, CTRL+U	: Şu anki satırı siler

Matlab Çabuk Erişim Kılavuzu

Çalışma ortamı komutları

help	: fonksiyon yardım komutu
demo, intro	: demo ve tanıtım komutu
who	: kayıtlı değişkenleri görüntüleme
whos	: kayıtlı değişken türlerini görüntüleme
what	: çalışma dizinindeki dosyaları görüntüleme
cd	: klasör gösterme değiştirme komutu
size	: matris boyut komutu
length	: matris uzunluk komutu
tic, toc	: zaman başlatma ve görüntüleme komutu
quit, exit	: Matlab dan çıkış komutu
clc	: çalışma ortamını temizler
clear	: çalışma ortamındaki tüm değişkenleri silme komutu
clear x	: çalışma ortamındaki x değişkenini silme komutu
save	: çalışma ortamını matlab.mat olarak kaydetme komutu
save ad	: çalışma ortamını ad.mat olarak kaydetme komutu
load ad	: çalışma ortamına matlab.mat ı yükleme komutu
print	: çalışma alanını veya şekli yazdırma komutu

Semboller

!	: Windows, Dos komutu başlatma komutu
%	: Açıklama getirme ifadesi
[]	: matris girme ifadesi
()	: indis ve değer girme ifadesi
=	: eşitlik ifadesi
:	: kolon ifadesi
;	: matris satır belirtme ifadesi
...	: bir ekrana sığmayan komutlar için devam ifadesi

(Tek bir satırda izin verilen karakter sayısı 4096 adet)

Matematiksel operatörler

+	: toplama
-	: çıkarma
*	: çarpma
/	: bölme
^	: üst alma
.*	: elemanter çarpma
./	: elemanter bölme
.^	: elemanter üst alma
sqrt	: karekök alma

Mantıksal operatörler

>	: büyüktür ifadesi
>=	: büyük eşittir ifadesi
<	: küçüktür ifadesi
<=	: küçük eşittir ifadesi
~=	: eşit değildir ifadesi
& ~	: ve, veya değil ifadeleri

Sabit ifadeler

pi	: 3,1416
inf	: sonsuz
nan	: sayı değil

i,j : imajiner i ve j

Trigonometri fonksiyonları

cos, sin, tan, cot : trigonometrik ifadeler

acos, asin, atan, acot : ters trigonometrik ifadeler

cosh, sinh, tanh, coth : hiperbolik ifadeler

Logaritma fonksiyonları

exp : eksponansiyel ifadesi

log : ln anlamlı logaritma ifadeleri

log10 : 10 tabanında logaritma

Kompleks fonksiyonları

abs : mutlak değer

angle : radyan olarak faz açısı

real : reel kısım

Matematiksel işlemler

sum, cumsum : toplama ve kümülatif toplama

diff : çıkarma

mean : ortalama ifadesi

median : orta değer gösterme komutu

min, max : minimum ve maksimum ifadeleri

factorial : Faktöryel alma

Lineer cebir

inv : tersini alma

eig : özdeğer bulma

det : determinant alma

' : transpoze alma

roots : kök bulma

Eğri uydurma

polyfit : polinom uydurma komutu
polyval : polinom değerleri bulma komutu

Rakamlar

Matlab rakamlar için önünde artı veya eksi işareti ve ondalık (desimal) işaretler sistemini kullanır. Bilimsel işaretler sistemi 10 tabanına göre kuvvet belirtmek için “e” harfi kullanılır. Sanal rakamlar, son takı olarak i veya j harfi kullanılır.

3 -99 0.0011 9.6397 1.6021e-21
6.0225e23 1i -3.1415j 3e5i

MATLAB TEMELLERİ

Temel yapı birimi dizilerdir. Dizi, satır veya sütunlar kullanılarak yapılandırılmış, belirli sayıda değerler içeren bir yapıdır. Tek boyutlu diziler “vektör” ve iki boyutlu diziler “matris” olarak tanımlanır.

$a = \begin{bmatrix} 2 & 1 \\ 4 & 3 \\ 6 & 5 \end{bmatrix}$ dizisi 3x2 boyutunda 6 elemanlı bir matris.

$b = [1 \ 5 \ 7 \ 10]$, 1x4 boyutunda 4 elemana sahip bir satır vektörüdür.

$a(3,2)=5$ $b(3)=7$

Matlab’da değişkenler, bilgisayar hafızasında belli bir yer kaplayan, kullanıcının belirlediği isme sahip olan dizilerdir.

MATLAB değişkenleri ve kuralları

- Değişkenler; herhangi bir tür bildiri veya boyut ifadeleri gerektirmez. A\$, A(5) gibi.
- Matlab yeni bir değişkenle karşı karşıya geldiğinde otomatik olarak değişken oluşturur ve yeteri kadar bellek ayırır.
- En basit değişken ismi tek bir harften (karakter) ibarettir.
- Değişken isimleri küçük/büyük harf kullanımına duyarlıdır. Baysi, baysi, baySi, BAYSI farklı değişkenlerdir.

- Değişken isimleri en çok 63 karakter içerir.
- Değişken isimleri daima bir harf ile başlamalı ve bunu herhangi bir sayıda harfler, rakamlar veya altçizgi “_” izleyebilir. Noktalama işaretleri değişken ismi olarak kullanılamaz.

Artış miktarı düzenli olan dizinlerin değişkenlere atanması

x=[J:I:K] J ile başlayan I aralığında artarak K' ya kadar giden dizi oluşur.
(J, J+K, J+2I, ... , K)

>>x=1:3:12 x=1 4 7 10
>>y=[1:3] y=1 2 3

x(:,J) x' in J' inci sütunu

>>x(:,3) 7

x(J:K) x' in J' den itibaren birer birer artarak K' ya giden elemanlarını
A(J), A(J+1) , A(J+2), ..., A(K)

>>x(2:3) 4 7

x(2:end) 2. elemandan başlayarak son elemana kadar dizileri adresler.

>>x(2:end 4 7 10)

Matris adresleme: iki boyutlu dizim.

x (i,j) i= eleman satır no, j= sütun no.

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

x= 1 2 3 “ ; “ işlemcisi elemanların satır sonlarını belirler.
4 5 6 Enter komutu “ ; “ yerine kullanılır.
7 8 9

x(2,3) “6” 2. satır 3. sütun

x(:,1) “1 4 7” x matrisinin 1. sütunundaki tüm elemanları

x(1,:) “1 2 3” x matrisinin 1. satırındaki tüm elemanları
x(2:3,:) “4 5 6 x matrisinin 2. ve 3. satırındaki tüm elemanları adresler.
 7 8 9”

Bir değişkene dışarıdan değer alınması:

```
x=input('Bir değer girişi yapınız: ');
```

Üretilen sonuçların sunulması:

<u>Fonksiyon</u>	<u>Açıklama</u>	<u>Örnek</u>
format short	Virgülden sonra 4 basamaklı gösterim	3.1416
format long	Virgülden sonra 14 basamaklı gösterim	3.14159265358979
format rat	iki sayının oranı şeklinde gösterim	355/113
format bank	Virgülden sonra 2 basamaklı gösterim	3.14

disp Fonksiyonu

```
>>b=' MATLAB';  
>>disp(b)  
MATLAB
```

fprintf Fonksiyonu

```
fprintf(bicim, deger)
```

“bicim” sunulacak bilginin şeklini, “deger” ise sunulacak bilgiyi göstermektedir

```
>>fprintf('pi sayisi = %5.2f \n\n ', pi)  
pi sayisi =    3.14
```

```
>>
```

%5; eşitlikten sonraki boşluğu (5 karakter)

.2f; ondalık hane sayısını (2 hane)

\n; yazının altındaki boşluğu (2 satır) belirlemektedir.

Veri (data) Dosyaları

Çalışma alanındaki verilerin kaydedilmesi ve yüklenmesi “save” ve “load” komutları ile yapılır. save komutu çalışma alanı içeriğini bir ikili sayılar (binary) mat dosyası olarak kaydeder. Bu dosya daha sonra “load” komutu ile geri çağırılabilir.

Save dosyası degisken1 degisken2 degisken3

>>“save uygulama

Tüm çalışma alanı içeriğini “uygulama.mat” dosyası içine saklar. Gerekirse dosya adından sonra değişken isimleri belirleyerek yalnızca belli değişkenlerin kaydedilmesi sağlanır.

>>save uygulama x y z

yalnızca x, y, z değişkenleri saklanmış olur.

load komutu daha önceden save komutu ile oluşturulan mat dosyasını çalışma alanına yükler.

>>load uygulama”

komutu “uygulama.mat” çalışma alanına yükler.

who ve whos komutları

“who” ve “whos” komutları kullanılarak o andaki çalışma alanı içindeki değişkenlerle ilgili bilgi verir.”who” komutu sadece değişkenlerin isimlerini ,”whose” ayrıca boyutu ve veri türü bilgileri de verir.

Not:

- Çalışma alanında yer alan tüm değişkenleri silmek için “clear” komutu kullanılır.
- Çalışma ekranını temizlemek “clc” komutu kullanılır.
- Çalışılan klasör yolunu görüntülemek için “cd” komutu kullanılır.

>>cd (enter)

C:\MATLAB1\work

- Mevcut yolu değiştirmek için;
>>cd\MATLAB\work\yeni
- Çalışılan klasördeki dosyaları görmek için
>>dir (enter)

- Sonuçların o anda ekranda görüntülenmesini istenmiyorsa komut satırı sonunda ‘;’ noktalı virgül koymak gerekir.

Skaler ve Matrislerde Dört İşlem

Skaler işlemleri

3^((5+3)/4) işleminin sonucu $=3^{(8/4)}=3^2=9$

```
>>3^((5+3)/4)
ans=          (ans=answer)
          9
```

Matris işlemleri

Çıkarma işlemi

$x = \begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix}$ ve $y = \begin{bmatrix} 2 & 8 \\ 5 & 9 \end{bmatrix}$ için $z=x-y$ işlemi

```
>>x=[2 4;6 8];y=[2 8;5 9]
>>z=x-y
z=
     0     -4
     1     -1
```

Matris işlemleri bir matris ile bir skaler arasında gerçekleştirmek mümkündür.

$a = \begin{bmatrix} 1 & 3 \\ 5 & 7 \end{bmatrix}$ ve $b=a+1$

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

İki matrisin çarpılabilmesi için birinci matrisin sütun sayısının ikinci matrisin satır sayısına eşit olması gerekir

$$z(i, j) = \sum_{k=1}^n x(i, k) * y(k, j)$$

$$x = \begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix} \text{ ve } y = \begin{bmatrix} -1 & 3 \\ 5 & 4 \end{bmatrix} \quad z = x * y$$

```
>>x=[1 2;4 5];y=[-1 3;5 4]
```

```
>>z=x*y
```

```
z=
```

```
     9     11
    21     32
```

Matlab da matrisleri eleman elemana işleme sokmak için kullanılan özel bir operatör vardır

.işlem

```
>>z=x.*y
```

```
z=
```

```
    -1     6
    20    20
```

Matris Transpozesi: Matrisin satırları, eleman sıralamaları korunarak sütuna dönüştürülür.

```
>>a=[1 2 3];
```

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

```
>> c=a'
```

```
A=
```

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

      1     4
      2     5
      3     6
```

Matrisin tersi (inversi) nxn boyutunda bir kare matris ve I birim matris olmak üzere $A*B=B*A=I$ eşitliğini sağlayan B matrisine A metrisinin tersi denir ve A^{-1} ile gösterilir.

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

```
>>inv(A)
```

```
ans =
```

```
     7     -3     -3
    -1     1     0
    -1     0     1
```

```
>>A*A^-1
```

```
>>inv(A)*A
```

ans=

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

ans=

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

MATLAB da polinomlar

Fonksiyon

conv(p1,p2)

deconv(p1,p2)

poly(a)

polyder(p)

polyfit(x,y,z)

polyval(p,x)

polyvalm(v,x)

residue(b,a)

roots(p)

Tanım

Polinom çarpma.

Polinom bölme.

Kökleri belli polinomun katsayılarını bulur.

Polinomun türevini bulur.

Polinom eğri uydurma

Elemanları verilen polinomun değerlerini hesaplar.

Elemanları verilen matris pol.değerlerini hesaplar.

Polinomu kısmi kesirlere ayırır.

Polinomun köklerini bulur.

Polinomların Gösterilmesi:

Matlab'ta polinomlar en yüksek dereceden en düşük dereceye azalan sırada Polinom katsayılarını içeren bir satır vektörü ile gösterilir.

Örneğin 4.derece bir Polinom $p(x)=x^5-2x^4+2x^3+3x^2+x+4$

matlab'ta bu polinomu girmek için

$p=[1 \ -2 \ 2 \ 3 \ 1 \ 4];$

İfadesi kullanılır. Bu matlab ortamında yukarıdaki polinomu tanımlar.

Köklerin bulunması $\gg r = \text{roots}(p)$

r=

1.5336 + 1.4377i

1.5336 - 1.4377i

-1.0638

-0.0017 + 0.9225i

-0.0017 - 0.9225i

solve komutu:

$x^2-4x+4=0$ denkleminin kökü

```
>> a=solve('x^2-4*x+4')
```

a =

[2]

[2]

$\sin(x)=\frac{\sqrt{2}}{2}$ denklemini sağlayan x değeri

```
>>x=solve('sin(x)=sqrt(2)/2')
```

x=

1/4*pi

Kökleri verilen bir polinomun katsayılarını bulmak için poly fonksiyonu kullanılır. yukarıdaki probleme uygularsak

```
pp=poly(r)
```

pp=

1.0000 -2.0000 2.0000 3.0000 1.0000 4.0000

$x_1=2$ ve $x_2=2$ iki kökü bulunan polinomun katsayıları

```
>>poly([2;2])
```

ans= 1 -4 4 x^2-4x+4

polyval(x,y) fonksiyonu

$p=s^3-2s^2+5s$ polinomunun $s=5$ için değeri

```
>>p=[1 -2 5 0];
```

```
>>polyval(p,5)
```

ans=100

Polinomun türevi ve integrali

$p=s^3-2s^2+5s$ polinomunun türevi $\frac{dp}{ds} = 3s^2 - 4s + 5$

```
>>p=[1 -2 5 0];
```

```
>>trev=polyder(p)
```

trev=

3 -4 5

$P(x)=3x^2+5x-4$ polinomunun $Q(x)=\int P(x)dx$ şeklindeki belirsiz integralinin hesabı

```
>>p_x=[3 5 -4];
```

```
>>int_polx=polyint(p_x)
```

```
int_polx =
```

```
1.0000 2.5000 -4.0000 0
```

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

Polinom çarpımı: conv fonksiyonu ile yerine getirilir.

$p1=x^3+2x^2+3x+4$ ve $p2=x^3+4x^2+9x+16$

```
>> p1=[1 2 3 4]; p2=[1 4 9 16]
```

```
>> pc=conv(p1,p2)
```

```
pc=
```

```
1 6 20 50 75 84 64
```

$$pc=x^6+6x^5+20x^4+50x^3+75x^2+84x+64$$

Polinom toplama: Matlab'ta Polinom toplamında doğrudan kullanılacak bir fonksiyon yoktur. Her iki Polinom vektörü aynı boyutta ise standart dizim toplama işlemi çalışır.

Yukarıdaki iki Polinom toplamak için;

```
>>pt=p1+p2
```

```
pt=
```

```
2 6 12 20
```

sonucu elde edilir. Buda $pt=2x^3+6x^2+12x+20$ polinom katsayılarıdır. İki fonksiyonun dereceleri farklı olduğunda, derecesi düşük olan polinoma sıfırlar ilave edilerek yüksek dereceden polinom ile aynı derece etkisine getirilmesi gerekir. pc ve pt polinomlarının toplamı;

```
>>pt1=pc+[0 0 0 pt]
```

```
pt1=
```

```
1 6 20 52 81 96 84
```

$$pt1=x^6+6x^5+20x^4+52x^3+81x^2+96x+84$$

Polinom bölme; Polinomların bölme işlemi *deconv* fonksiyonu ile yerine getirilir. Yukarıdaki pc fonksiyonunu p1 fonksiyonuna bölmek için

```
>>[q r]=deconv(pc,p2)
```

```
q =1   2   3   4           bölüm
```

```
r =0   0   0   0   0   0   0   kalan
```

Polyfit fonksiyonu: Bu fonksiyon, verilen bir veri takımı için derecesi belli polinomun en iyi biçimde uydurulmasının hesaplanmasını sağlar:

```
p = polyfit(x,y,z)
```

Polyval fonksiyonu: katsayıları belli bir polinomun verilen bir bağımsız değişkene karşılık gelen bağımlı değişkenini hesaplar.

```
y = polyval(p,xi)
```

```
>>x=[1 2 3 5 6 7]
```

```
>>y=[1 8 27 125 216 343]
```

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

```
c =
```

```
12.0000    -41.0000    36.0000
```

```
c= 12x2-41x+36
```

```
>>polyval(c,4)
```

```
ans
```

```
64.0000
```

LİNEER DENKLEM SİSTEMLERİ

Mühendislikte bir çok problemin çözümünde çok sayıda bilinmeyen ve genellikle buna eşit sayıdaki denklemden oluşan sistemler ile karşılaşılır. Lineer denklem sistemlerinde tüm bilinmeyenler, birinci derecededir. Böyle bir sistemin genel hali

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n = b_2$$

.....

$$a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \dots + a_{mn}x_n = b_m$$

Bu denklem matris biçiminde gösterimi $[A][X] = [B]$ şeklinde tanımlanabilir.

Burada A: katsayılar matrisi, X: bilinmeyenlere ait sütun vektörü ve B ise eşitliğin diğer tarafındaki sütun vektörüdür ve aşağıdaki gibi gösterilirler:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \quad X = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} \quad B = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix}$$

Lineer denklem sistemlerinin çözümünde çeşitli yöntemler mevcuttur.

Katsayılar matrisinin inversi yardımı ile çözüm

$A.X=B$ biçiminde verilen ve B'nın satır matrisi olarak tanımlandığı matris denkleminin her iki tarafını A^{-1} ile çarparsak

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

Burada $A^{-1}.A = I$

olarak tanımlanan birim matristir. Yani; $I.X = A^{-1}.B$ veya $x=inv(A)*B$ komutu ile elde edilir.

Matlab ortamında bu çözüm

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

Aşağıda verilen denklem takımının çözümü

$$x_1 + 3x_2 + x_3 = 2$$

$$2x_1 - x_2 + 3x_3 = 1$$

$$x_1 + 2x_2 - x_3 = 3$$

$$\gg A = [1 \ 3 \ 1; \ 2 \ -1 \ 3; \ 1 \ 2 \ -1];$$

$$\gg B = [2; 1; 3];$$

$$\gg X = \text{inv}(A) * B$$

X =

$$1.6667$$

$$0.3333$$

$$-0.6667$$

Not: Bu yöntemin kullanılabilmesi için katsayılar matrisinin kare olması gerekir.

A\B komutunun kullanılması aynı sonucu verecektir.

Gauss eliminasyon yöntemi

Bu yöntemin kullanılabilmesi için katsayılar matrisinin kare olması şartı yoktur.

$$x_1 + 4x_2 - x_3 = 7$$

$$2x_1 + 3x_2 - 2x_3 = 9$$

$$x_1 + x_2 + x_3 = 4$$

$$3x_1 + 12x_2 + x_3 = 21$$

Bu denklem sisteminin çözümü

$$\gg A = [1 \ 4 \ -1; \ 2 \ 3 \ -2; \ 1 \ 1 \ 1; \ 3 \ 12 \ 1];$$

$$\gg B = [7; 9; 4; 21];$$

$$\gg C = [A \ B];$$

$$\gg y = \text{rref}(C)$$

y =

$$1 \ 0 \ 0 \ 3$$

$$0 \ 1 \ 0 \ 1$$

$$0 \ 0 \ 1 \ 0$$

$$0 \ 0 \ 0 \ 0$$

$$\gg x = y(:, \text{end})$$

x =

$$3$$

1
0
0

elde edilir. Yani denklem sisteminin çözüm kümesi: $x_1=3$, $x_2=1$ ve $x_3=0$ dır.

Yukarıdaki lineer denklem sisteminin çözümü için *pinv* komutu kullanılabilir

```
>>A=[1 4 -1; 2 3 -2; 1 1 1; 3 12 1];
```

```
>>B=[7; 9; 4; 21];
```

```
>>x=pinv(A)*B
```

x =

3.0000

1.0000

-0.0000

Cramer yöntemi

Katsayılar matrisinin kare matris olması durumunda kullanılır.

$$x + 2y + z = 5$$

$2x + 2y + z = 6$ lineer denklem sisteminin Cramer yöntemi ile çözümü

$$x + 2y + 3z = 9$$

$$\gg C=[1; 2; 1]; \quad D=[2; 2; 2]; \quad E=[1; 1; 3];$$

$$\gg A=[C, D, E];$$

$$\gg B=[5; 6; 9];$$

$$\gg x=\det([B, D, E])/\det(A)$$

x=

1

$$\gg y=\det([C, B, E])/\det(A)$$

y=

1

$$\gg z=\det([C, D, B])/\det(A)$$

z=

2

lu Fonksiyonu kullanarak çözüm

$$x + 2y + z = 5$$

$$2x + 2y + z = 6$$

$$x + 2y + 3z = 9$$

$$\gg A=[1 \ 2 \ 1; 2 \ 2 \ 1; 1 \ 2 \ 3];$$

$$\gg B=[6; 6; 9];$$

$$\gg [L,U]=lu(A);$$

$$\gg Y=inv(L)*B;$$

$$\gg X=inv(U)*Y$$

X =

1

1

2

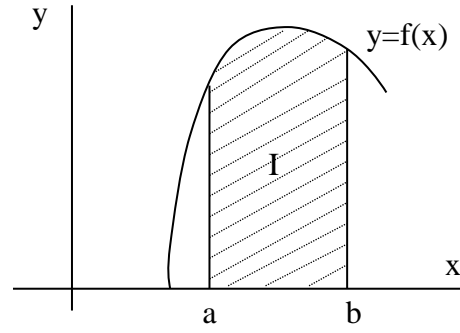
MATLAB DA SAYISAL İNTEGRAL

Herhangi bir $y=f(x)$ fonksiyonunun, $f(x) \geq 0$ olmak koşulu ile $[a-b]$ aralığındaki integrali, o eğrinin $x=a$ ve $x=b$ doğruları ile x ekseninde sınırladığı bölgenin alanına eşittir. Bu integralin

değeri; $I = \int_a^b f(x)dx$ şeklinde ifade edilir. Bir

fonksiyonun $[a-b]$ aralığındaki integralinin değeri

analitik olarak hesaplanamıyor ise sayısal çözüm yöntemleri kullanılır.



Quad) fonksiyonu

$y = x^3 - 3x$ Fonksiyonunun $[0,2]$ aralığındaki $Q = \int_0^2 (x^3 - 3x)dx$ integralini hesaplamak için

```
>>Q=quad('x.^3-3*x',0,2)
```

Q=

-2.0000

Trapz.fonksiyonu

$Q = \int_0^5 (x^2 + 5x)dx$

```
>>x=0:1:5;
```

```
>>y=x.^2+5*x;
```

```
>>Q=trapz(y)
```

Q=

105

$Q = \int_0^{\pi/2} \sin(x)dx$

```
>>x=0:pi/4:pi/2;
```

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

```
>>trapz(y)
```

Q=

1.2071

Daha yaklaşık bir sonuç elde edebilmek için fonksiyonu $trapz(x,y)$ şeklinde kullanmak gerekir

```
>>x=0:pi/4:pi/2;
```

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

```
>>trapz(x,y)
```

Q=

1.8961

Birim aralıklı hesaplatılan sayısal integralde x aralıklarını daraltmak daha kesin sonuç verir.

```
>>x=0:pi/60:pi/2;
```

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

```
>>trapz(x,y)
```

Q=

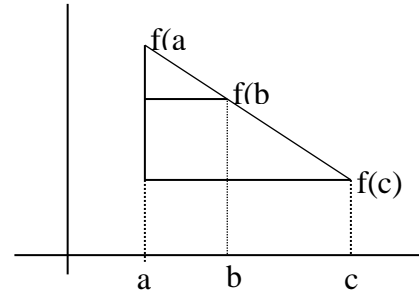
1.8961

MATLAB'DA ENTERPOLASYON, ARA DEĞER HESABI VE EĞRİ UYDURMA

DeneySEL bir gözlem veya yapılan bir dizi ölçüm sonucu elde edilen veriler, uygun bir $f(x)$ fonksiyonuna ait noktalmış gibi düşünülebilir. Enterpolasyonun temeli bir $f(x)$ fonksiyonunun bazı değişkenler için bilinen değerlerinden yararlanarak ve bu değerleri çeşitli yöntemlerde kullanarak, o fonksiyonun herhangi bir değişken için değerini uygun bir yaklaşımla tespit etmeye dayanır.

İki veri arasındaki ilişkiyi fonksiyonel bir şekilde yapılandırmanın en basit yolu lineer enterpolasyon

kullanmaktır. Lineer enterpolasyonda iki farklı değişkene karşılık gelen fonksiyon değerleri bir doğru ile birleştirilir. Bu doğrunun denkleminin elde edilmesi ile b gibi bir ara değişken için fonksiyonun değerini hesaplamak mümkün olur.



Enterpolasyon fonksiyonunu elde edebilmek için aşağıdaki iki noktası bilinen doğru denklemi formülünden yararlanılır.

$$\frac{f(a) - f(b)}{a - b} = \frac{f(a) - f(c)}{a - c}$$

interp1 fonksiyonu

Tabloda 15° ve 35° arasındaki çeşitli hava sıcaklığı değerleri için deniz suyu sıcaklıkları ölçülmüştür. 25° için deniz suyu sıcaklığı tahmin edilmek istenmektedir. Burada deniz suyu sıcaklığı hava sıcaklığının bir fonksiyonu olarak düşünülmüştür.

```
>>sicaklik=[15 18 25 30 35];
>>deniz_suyu=[7.5 10 16 22.5 25];
>>deniz_20=interp1(sicaklik, deniz_suyu, 20)

deniz_20=
    11.7143
```

Sıcaklık °C	Deniz suyu sıcaklığı °C
15	7.5
18	10
25	16
30	22.5
35	25

Farklı enterpolasyon komutları

$Y=x^3-x^2$ fonksiyonuna ait bazı değerler verilmiş ve $x=4$ için değişik enterpolasyon komutu ile fonksiyonun değeri hesaplanmıştır

```
>>x=[1 2 3 4 5];
>>y=[0 4 18 48 100];
>>yy=interp1(x,y,4,'linear')
yy= 59
>>yy=interp1(x,y,4,'nearest')
yy= 100
>>yy=interp1(x,y,4,'spline')
yy= 48.0000
>>yy=interp1(x,y,4,'pchip')
yy= 49.1983
>>yy=interp1(x,y,4,'v5cubic')
yy= 48.0000
>>yy=spline(x,y,4)
yy= 48.0000
>>yy=pchip(x,y,4)
yy=
    49.1983
```

x	y
1	0
2	4
3	18
4	48
5	100

interp2 fonksiyonu ve iki boyutlu enterpolasyon

Bir verinin diğer iki değişkenin fonksiyonu olduğu durumlarda bu iki verinin bazı değerlerine karşılık gelen fonksiyon değerini bulmak için kullanılır. Kullanım şekli

$$zI=interp2(x,y,z,xI,yI)$$

Ortam sıcaklığı	5	10	20	30
Zaman	Motor sıcaklığı			
0	5	10	20	30
1	7	13	25	35
2	10	15	29	37
4	11	18	32	38
5	11.5	20	35	40

Özellikleri tabloda verilen elektrik motorunun 15 °C ortam sıcaklığında çalışmaya başladıktan 3 dakika sonra sahip olacağı sıcaklık değerini hesaplayan komut dizisi;

```
>> x=[5 10 20 30];  
>> y=[0 1 2 4 5];  
>> z=[5 10 20 30  
7 13 25 35  
10 15 29 37  
11 18 32 38  
11.5 20 35 40]
```

z =

```
5.0000 10.0000 20.0000 30.0000  
7.0000 13.0000 25.0000 35.0000  
10.0000 15.0000 29.0000 37.0000  
11.0000 18.0000 32.0000 38.0000  
11.5000 20.0000 35.0000 40.0000
```

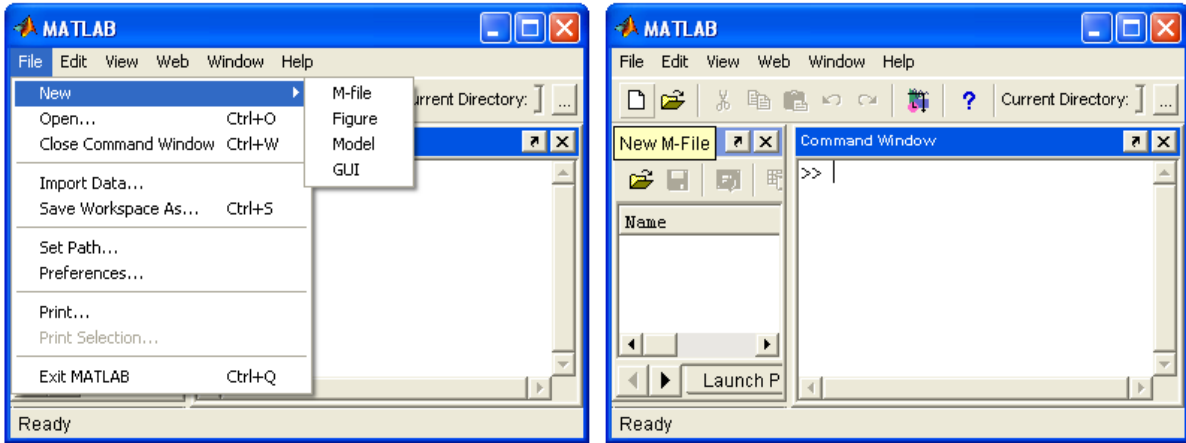
```
>> zz=interp2(x,y,z,15,3)
```

zz =

```
23.5000
```

MATLAB PROGRAMLAMA DOSYALARI

Basit problemleri MATLAB komut penceresinde tasarlamak ve uygulamak kolaydır. Ancak deęişken sayısının artması ve oluşan sonuçlara göre algoritmanın deęişiklik göstermesi durumunda komut penceresinden program hazırlamak kullanışlı olmaz. Bu amaçla MATLAB derleyicisi (M-Dosya, M-File) ile oluşturulan komutlar kullanılabilir.



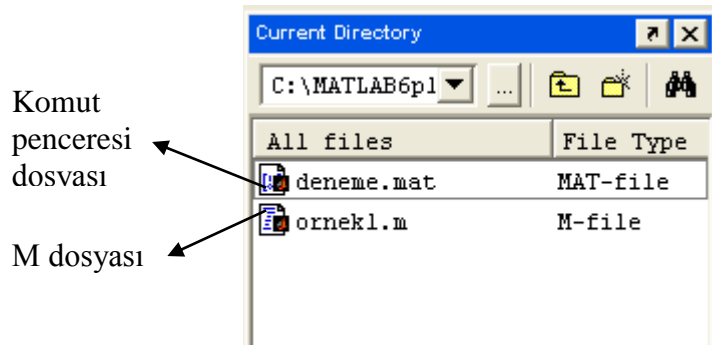
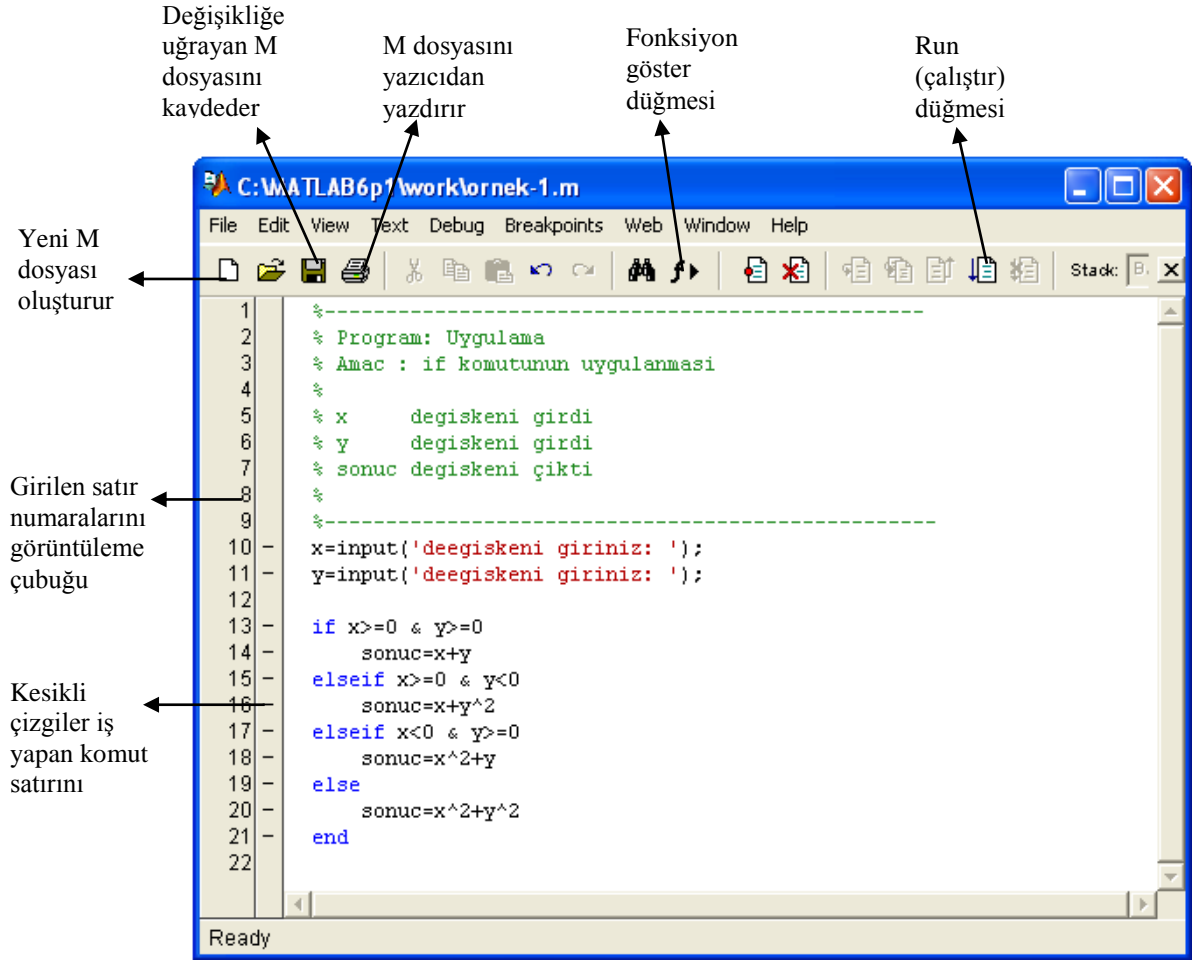
Şekil 1

MATLAB Editör/Debugger i açmak için şekilde görülen File menüsündeki “New” sekmesinden “M-file seçilebileceęi gibi Matlab çubuğundaki boş sayfaya basılarak yeni M dosyası açılabilir veya Matlab komut satırında >> **edit** komutu yazılır.

M-Dosyaları=Matlab'ta algoritmaları bilinen programları hazırlamak ve çalıştırmak için kullanılır. Matlab dil kodu içeren dosyaları M-dosyaları (M-files) adı verilir. Bir M-dosyasının Matlab tarafından bir dosya olarak onaylanması için ‘.m’ uzantısına sahip olması gerekir.

Hazırlanan M-dosyaları editör penceresinde yer alan File menüsünden Save komutu yardımı ile belirlenen bir isim altında uygun bir klasöre kaydedilir. Hazırlanan programın Matlab'da çalıştırılabilmesi için, kayıt edilen klasörün Matlab'ın tanıyabileceęi bir yol üzerinde bulunması veya yolun tanıtılması gerekir. Matlab ana klasörü altında work (çalışma) adı altındaki klasör çalışmalar için kullanılabilir. Matlab derleyicisinde oluşturulan M-Dosya File menüsünden **Save** seçeneęi ile çalışma dizini olan “work” altına saklanır. Bir M-Dosyasının çalıştırılması ile bu m-Dosyanın içerdiği tüm komutların tek tek komut satırında yazılarak çalıştırılmasının oluşturacağı sonuçlar aynıdır. Bir m Dosyasının “input” komutu kullanmadan

girdi ifadesi alma imkânı yoktur. Şekil 2 de MATLAB editör e ait pencere ve özellikleri görülmektedir.



Şekil 2

***if* Şartlı Deyimi**

if bir mantıksal deyimi değerlendirir ve bu deyimin değerini esas alan bir grup ifadeyi yürütür.

if mantıksal deyim

Komut ifadeler

End

Eğer mantıksal ifade doğru (1) ise Matlab *if* ve *end* satırları arasında kalan tüm komut ifadelerini yürütür. *End* komutunu izleyen satırda yürütmeyi sürdürür. Eğer koşul yanlış ise (0) Matlab *if* ve *end* satırları arasında kalan tüm komut ifadelerini atlayıp geçer ve *end* komut satırından sonraki satırda yürütmeyi sürdürür.

***else* ve *elseif* Şartlı Deyimleri**

İf deyiminin daha fazla koşullara bağlanmasını sağlar. *Else* deyimi hiçbir mantık koşuluna sahip değildir. Eğer kendinden önceki *if* (ve olası *elseif* koşusu) yanlış (0) ise *else* deyimi ile ilişkili ifadeler yürütülür.

if deyim

Komut ifadeleri % *if* doğru ise değerlendirilecek

else

Komut ifadeleri % *if* yanlış ise değerlendirilecek

end

elseif deyimi bir mantıksal koşula sahip olup, eğer kendisinden önceki *if* (ve olası *elseif*) koşulu yanlış (0) ise değerlendirme yapar. Eğer *elseif* mantıksal koşulu doğru (1) ise onunla ilişkili ifadeleri yürütür.

if deyim1

Komut ifadeleri % deyim1 doğru ise değerlendirilecek

elseif deyim2

Komut ifadeleri % deyim2 doğru ise değerlendirilecek

elseif deyim3

Komut ifadeleri % deyim3 doğru ise değerlendirilecek

elseif

else

Komut ifadeleri % eğer diğer hiçbir deyim doğru değilse değerlendirilecek

end

Örnek: $f(x) = \begin{cases} x + y & \rightarrow x \geq 0 \text{ ve } y \geq 0 \\ x + y^2 & \rightarrow x \geq 0 \text{ ve } y < 0 \\ x^2 + y & \rightarrow x < 0 \text{ ve } y \geq 0 \\ x^2 + y^2 & \rightarrow x < 0 \text{ ve } y < 0 \end{cases}$

İfadesinin (x,y), (2,3), (1,-5), (-2,1), (-3,-1) değerleri için “input” komutu kullanarak M-dosya formunda yazılması için yukarıdaki şekilde görülen komutlar yazılır.

%-----

% program: uygulama

% amac: if komutunun uygulanması

%

% x degiskeni girdi

% y degiskeni girdi

% sonuc degiskeni çıktı

%

%-----

x=input('x degiskenini giriniz: ');

y=input('y degiskenini giriniz: ');

if x>=0 & y>=0

sonuc=x+y

elseif x>0 & y<0

sonuc=x+y^2

elseif x<0 & y>=0

sonuc=x^2+y

else

sonuc=x^2+y^2

end

```
>> uyg1
```

```
x degiskenini giriniz: 2
```

```
y degiskenini giriniz: 3
```

```
sonuc = 5
```

```
>> uyg1
```

```
x degiskenini giriniz: -3
```

```
y degiskenini giriniz: -1
```

```
sonuc = 10
```

Örnek: İkinci dereceden denklemin kökünü bulan program

```
%-----  
% İkinci dereceden denklemin kökünü bulan program  
%-----  
clc  
clear  
A=input('x^2 li terimin katsayisini giriniz= ');  
B=input('x li terimin katsayisini giriniz= ');  
C=input('sabit terimi giriniz= ');  
delta=B^2-4*A*C;  
if delta<0  
    S='Kokler Kompleks';  
    disp(S);  
else  
    if delta > 0  
        disp('Kokler Reel')  
    else  
        if delta == 0  
            disp ('Cakisik Kokler var')  
        end  
    end  
end  
end
```

```
X1=(-B+sqrt(delta))/(2*A);
X2=(-B-sqrt(delta))/(2*A);
X1=num2str(X1);
X2=num2str(X2);
fprintf('Birinci kok: %s\n',X1);
fprintf('Ikinci kok: %s\n',X2);
```

Örnek $x^2-4x+4=0$ denkleminin kökü

```
>> kok
```

```
x^2 li terimin katsayisini giriniz= 1
```

```
x li terimin katsayisini giriniz= -4
```

```
sabit terimi giriniz= 4
```

```
Cakisik Kokler var
```

```
Birinci kok: 2
```

```
Ikinci kok: 2
```

for Döngüsü

for döngüsü bir komut ifadesi veya komut ifadesi gurubunu önceden saptanan bir sayı kadar yürütür.

```
for indeks=başlangıç:artış:son
```

```
    Komut ifadeleri
```

```
end
```

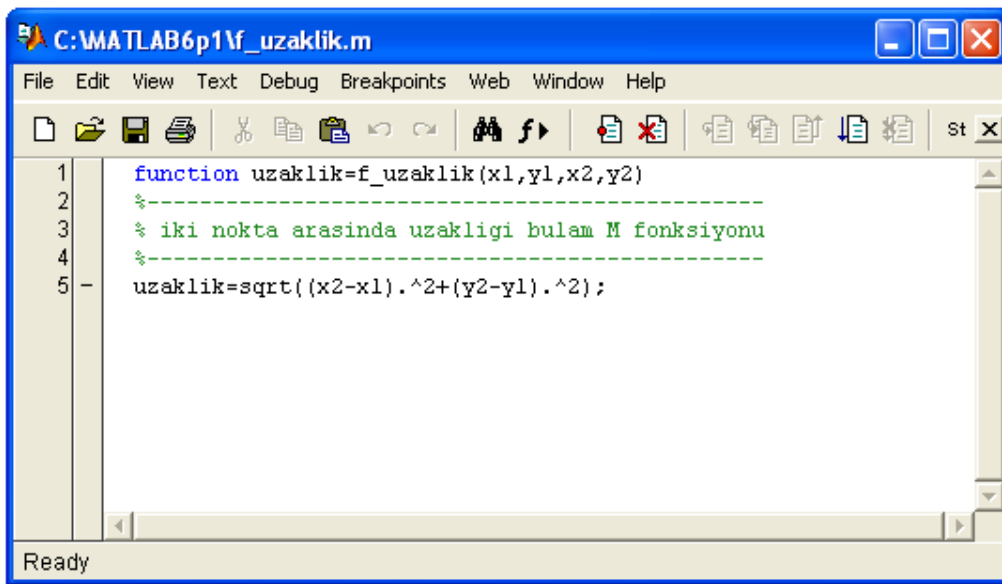
Örnek: Faktöriyel hesaplayan program

```
%-----  
%  
% Faktöriyel hesaplayan program  
%  
%-----  
clc  
clear  
N=input('Faktöriyeli hesaplanacak sayıyı giriniz= ');  
Nfak=1;  
for Isay=1:N  
    Nfak=Nfak*Isay;  
end,  
fprintf('Faktöriyel değeri=%d\n',Nfak);  
  
>> fak  
Faktöriyeli hesaplanacak sayıyı giriniz= 5  
Faktöriyel değeri=120
```

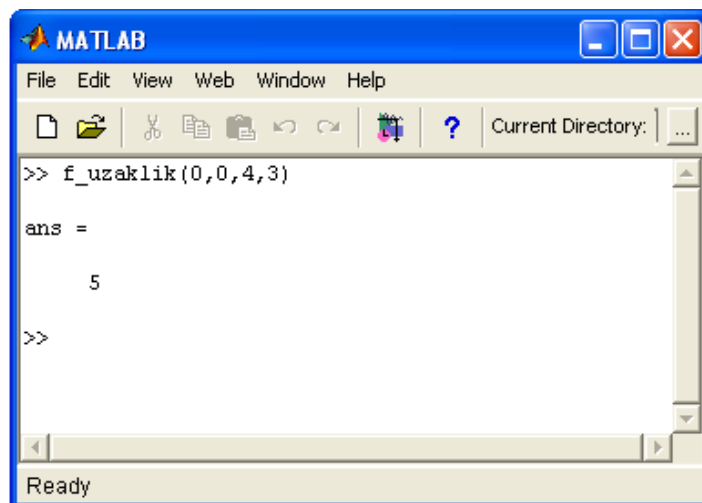

Fonksiyon Dosyaları:

Fonksiyon dosyaları ilk satırında “function” kelimesi bulunan “.m”uzantılı dosyalardır. Hazırlanan herhangi bir program içinde çağrılabilir. M fonksiyonlar kendi çalışma alanına sahip olan özel bir tür M-Dosya lardır. M-Fonksiyonlar input komutu kullanmadan giriş ifadeleri alır ve bu ifadeleri belirtilen kodlamalara göre yapılan işlemler sonrası çıkış ifadelerine atar.

function cikis_ifadesi=fonksiyon_adi (giris_ifadesi1, giris_ifadesi2,)



```
1 function uzaklik=f_uzaklik(x1,y1,x2,y2)
2 %-----
3 % iki nokta arasinda uzakligi bulan M fonksiyonu
4 %-----
5 uzaklik=sqrt((x2-x1).^2+(y2-y1).^2);
```



```
>> f_uzaklik(0,0,4,3)

ans =

     5

>>
```

Şekil 3

```
function uzaklik=f_uzaklik(x1,y1,x2,y2)
% -----
% iki nokta arasındaki uzaklığı bulan program
% -----
uzaklik=sqrt((x2-x1).^2+(y2-y1).^2);
```

```
function shacim=f_hacim(r,h)
% -----
% silindirin hacmini bulan program
% -----
shacim=pi*r^2*h;
```

```
>> f_hacim(1,2)
```

```
ans =
```

```
6.2832
```

MATLAB'DA GRAFİK İŞLEMLERİ

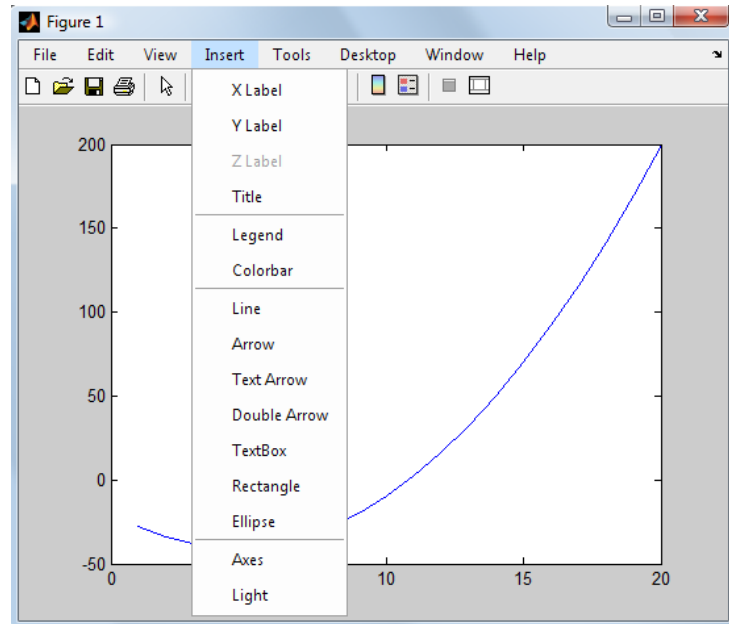
Matlab da grafik çizdirmenin en kolay yolu tanımlanan aynı uzunluktaki iki vektörü birer değişkene atamak ve bu değişkenleri “*plot*” fonksiyonu ile birlikte kullanmaktır.

Örnek

$y = x^2 - 9x - 20$ fonksiyonunun x in 1 ile 20 arasında 1 er artan değerleri için grafiğini aşağıdaki gibi çizdirebiliriz.

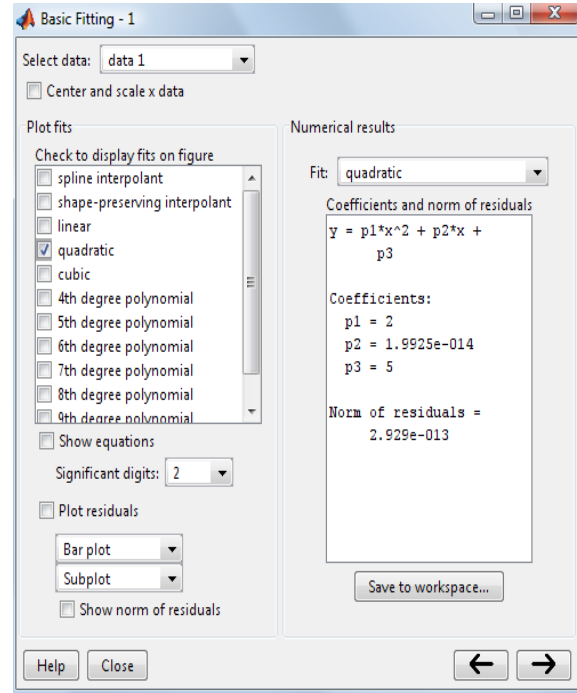
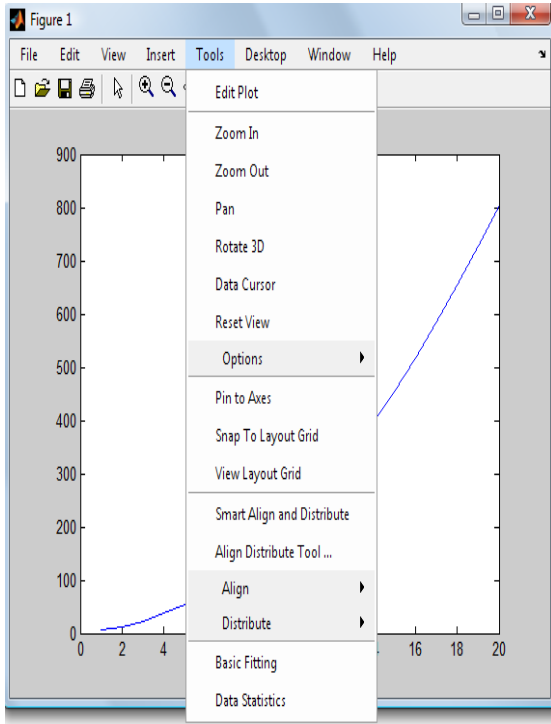
```
>> x=1:20;  
>> y=x.^2-9*x-20;  
>> plot(x,y)
```

Matlab grafik işlemlerini şekil penceresi adı verilen ve komut penceresinden ayrı bir pencereye gönderir. Matlab içinde bu pencere (figure) bir şekil olarak görülür.



Örnek

>> x=1:20;



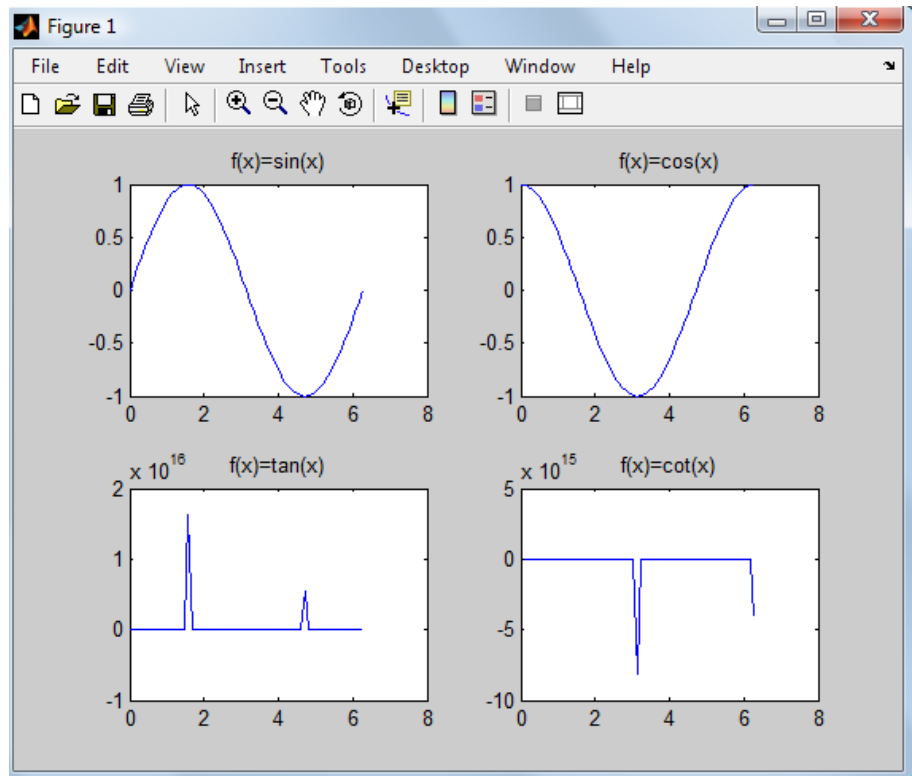
>> y=2*x.^2+5;

subplot fonksiyonu ve alt grafikler

Matlab da aynı anda birden fazla grafik penceresi açmak ve bunlarda farklı fonksiyonların grafiklerinin görüntülenmesini sağlamak mümkün olduğu gibi, aynı grafik penceresinde birden fazla grafiğe yer vermek mümkündür. Bu işlemi gerçekleştirmek için genel kullanımı subplot (a, b, c) şeklinde olan “subplot” fonksiyonu kullanılır. Bu fonksiyon ile grafik penceresi axb (a satır, b sütun sayısı) boyutunda olacak şekilde yapılandırılır. c ise grafik komutlarının işletileceği alt pencere numarasıdır.

Örnek:

```
>> subplot(2,2,1)
>> x=0:pi/30:2*pi;
>> y1=sin(x);
>> plot(x,y1)
>> title('f(x)=sin(x)')
>> subplot(2,2,2)
>> y2=cos(x);
>> plot(x,y2)
>> title('f(x)=cos(x)')
>> subplot(2,2,3)
>> y3=tan(x);
>> plot(x,y3)
>> title('f(x)=tan(x)')
>> subplot(2,2,4)
>> y4=cot(x);
>> plot(x,y4)
>> title('f(x)=cot(x)')
```



hold Komutu

Matlab aynı grafik penceresinde çok sayıda grafiğe yer verebildiği gibi aynı eksen takımında birden fazla fonksiyon grafiği çizdirebilir.

```
>> x=0:pi/30:2*pi;
```

```
>> y1=sin(x);
```

```
>> plot(x,y1,'k*:')
```

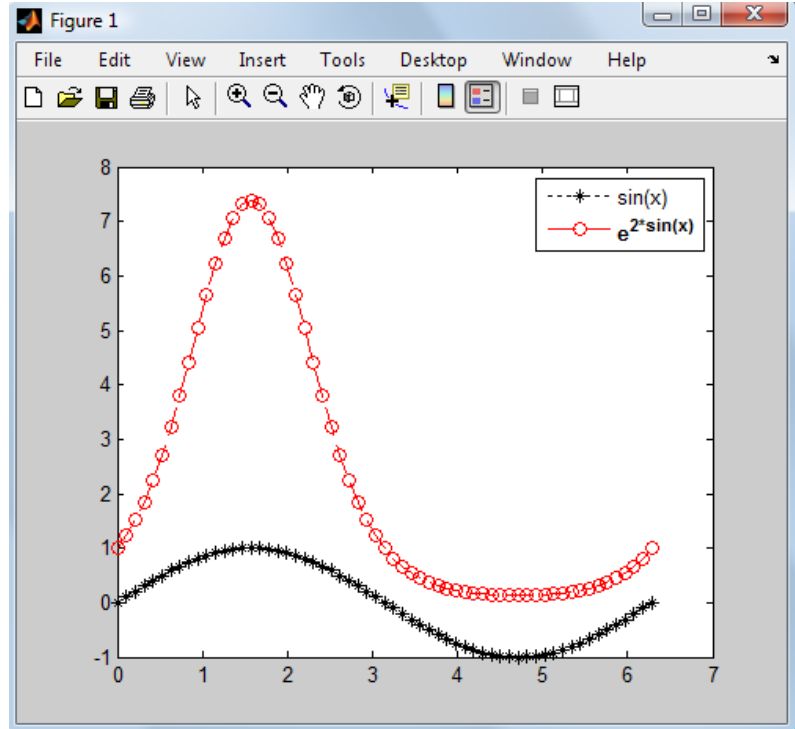
```
>> hold
```

Current plot held

```
>> y2=exp(2*sin(x));
```

```
>> plot(x,y2,'ro--')
```

```
>> legend('sin(x)', '\bf{e}^{2*sin(x)}')
```



3_Boyutlu Çizgi Grafikleri

plot3 fonksiyonu 3 boyutlu çizgi grafik için kullanılır. Genel formatı

`plot3 (x1, y1, z1, S1, x2, y2, z2, S1,)` şeklindedir.

Burada x_n , y_n , z_n vektör veya matris, ve S_n renk, çizgi stili veya işaretleyici simgesini belirleyen seçimli karakter koşuludur.

Aşağıda verilen komut ifadeleri üç boyutlu bir helis oluşturur.

```
>>t=0:pi/50:10*pi;
```

```
>>plot3(sin(t), cos(t),t)
```

