

İÇİNDEKİLER

ÖNSÖZ.....	I
İÇİNDEKİLER.....	II
ÖZET.....	VI
ABSTRACT.....	VI
ŞEKİL LİSTESİ.....	VIII
TABLO LİSTESİ.....	XI

BÖLÜM 1

1. MATLAB'A GİRİŞ.....	1
1.1 MATLAB ürün ailesi.....	1
1.2 MATLAB araç kutuları(Toolboxs).	3
1.3 MATLAB'ın kullanım amacı ve alanı.....	4
1.4 MATLAB'ı kurulumu ve çalıştırılması.....	4
1.4.1 MATLAB'ın Kurulumu.....	4
1.4.2 MATLAB'ın başlatılması.....	5
1.4.3 MATLAB komut penceresi.....	5
1.4.4 Genel Komutlar.....	5
1.4.5 Özel Karakterler.....	8
1.5 MATLAB Komut Penceresi Menüleri.....	10
1.5.1 File (Dosya) Menüsü.....	11
1.5.2 Edit (Düzen) Menüsü.....	11
1.5.3 Options (Seçenekler) Menüsü.....	11
1.5.4 Windows Menüsü.....	12
1.5.5 Help (Yardım) menüsü.....	12
1.5.6 Şekil Penceresi Menüleri.....	12
1.5.6.1 File (Dosya).....	12
1.5.6.2 Edit (Düzen)	12

BÖLÜM 2

2. MATRİSLER.....	14
-------------------	----

2.1 İfadeler.....	14
2.2 Değişkenler.....	14
2.3 Sayılar.....	15
2.4 Operatörler(sayısal işlemciler).....	15
2.5 Kompleks Sayılar.....	16
2.6 Fonksiyonlar.....	17

BÖLÜM 3

3. MATLAB'DA MATRİSLER VE TEMEL MATRİS İŞLEMLERİ.....	19
3.1 Matrisleri Girmek.....	19
3.2 Matrislerde Elemanlar.....	20
3.3 Genel Matrisler.....	21
3.4 Matrislerde İşlemler.....	22
3.4.1 Matrislerin Transpozu.....	22
3.4.2 Matrislerde Toplam ve Fark İşlemleri.....	23
3.4.3 Matrislerin Çarpılması.....	24
3.4.4 Matrislerin Bölünmesi.....	25
3.4.5 Matrislerin kuvvetini alma işlemi.....	26
3.4.6 Matrislerin Üstel Fonksiyonları.....	26
3.4.7 Matrislerin Tersini Alma.....	27

BÖLÜM 4

4. MATLAB'DA GRAFİKLER.....	29
4.1 2 Boyutlu(2D) Grafikler.....	29
4.1.1 2D Grafik Fonksiyonları.....	29
4.1.2 Bir plotun oluşturulması.....	35
4.1.3 Grafik Çiziminde Çizgi Stilleri, İşaretler ve Renkler.....	36
4.1.4 Matrislerin Çizdirilmesi.....	37
4.2 3 Boyutlu(3D) Grafikler.....	37
4.2.1 3D Grafik Fonksiyonları.....	38
4.2.2 3-Boyutlu Bir Grafiğe Bakış Açısı.....	41

BÖLÜM 5

5. SIMULINK VE SIMULASYON.....	44
5.1 Simulink'e giriş.....	44
5.2 Simulink Blokları.....	44
5.3 Simulink Modellerinin Oluşturulması ve Çalıştırılması.....	45
5.4 Adım adım örnek bir modelin oluşturulması ve çalıştırılması.....	46
 KAYNAKLAR.....	 50
EK:MATLAB komutları.....	51
ÖZGEÇMİŞ.....	56

Ö Z E T:

Bu alıřmada, zellikle teknik alanlarda

- matematik ve hesaplama
- algoritma geliřimi
- modelcilik , benzetim ve prototip
- veri analizi , arařtırma ve canlandırma
- bilimsel ve mhendislik grafikler
- uygulama geliřimi , grafiksel kullanıcı arabirimi yapısı

gibi tasarım ve veri özmlmeleri yapanlar iin nemli bir kaynak teřkil edecek řekilde genel ve z bir řekilde MATLAB anlatılmıřtır.

Tezin beř blm iersinde MATLAB'ın kullanımı, temel iřlemleri, matris uygulamaları, řartlı deyimler, grafik ve SIMULINK iřlemleri arpıcı rneklerle birlikte ele alınmıřtır. Buradaki iřlemlerde tezdten istifade etmek isteyenler iin sıkıcı olmayan ve de en ok ihtiya duyulan temel konulara deyinilmiřtir.

ABSTRACT

In this work especially in technical areas

- Math and computation
- Algorithm development
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including Graphical User Interface building

the ones who are making design and data analysis, this MATLAB has been briefly and generally explained so as to constitute a significant source.

In the five section of the thesis, the use of MATLAB, basic operations conditioned expressions, graphic and SIMULINK oprations have been explained with splendid examples. In these operations the ones who want to make use of this thesis ,the most needed basic subjects have been told in a non-boring way.

ŞEKİL LİSTESİ :

Şekil1.1-Matlab Ailesi.....	2
Şekil1.2-Matlab Komut Penceresi.....	5
Şekil4.1-plot fonksiyonuna bir örnek.....	30
Şekil4.2-loglog fonksiyonuna bir örnek.....	31
Şekil4.3-semilog7 fonksiyonuna bir örnek.....	32
Şekil4.4-title fonksiyonuna bir örnek.....	33
Şekil4.5-text fonksiyonuna bir örnek.....	34
Şekil4.6-Bir matrisin çizimi.....	37
Şekil4.7-plot3 fonksiyonuna bir örnek.....	38
Şekil4.8-meshc fonksiyonuna bir örnek.....	39
Şekil4.9-meshz fonksiyonuna bir örnek.....	40
Şekil4.10-surface fonksiyonuna bir örnek.....	41
Şekil4.11-3 boyutlu bir grafiğe bakış açısı.....	42
Şekil4.12-3 boyutlu bir grafiğin normal görünümü.....	42
Şekil4.13-Aynı grafiğin farklı bir yönden görünümü.....	43
Şekil4.14-Aynı grafiğin farklı bir yönden görünümü.....	43
Şekil4.15-Aynı grafiğin farklı bir yönden görünümü.....	43
Şekil5.1-Simulink Blokları.....	44
Şekil5.2-Yeni bir simulink çalışma penceresi.....	45
Şekil5.3 Yeni bir simulink çalışma penceresinin açılması.....	46
Şekil5.4-Bir eleman kümesi.....	46
Şekil5.5-Elemanların taşınması.....	47
Şekil5.6-Elemanların taşınmış hali.....	47
Şekil5.7-Multiplexirin ayak sayısının ayarlanması.....	47
Şekil5.8-Bağlantı yapımı.....	48
Şekil5.9-Bağlantıların yapılması.....	48
Şekil5.10-Bir modelin tamamlanmış hali.....	48
Şekil5.11-Bir modelin scope çıkışı.....	49

TABLO LİSTESİ:

Tablo1.1.....	8
Tablo1.2.....	8
Tablo1.3.....	12
Tablo2.1.....	15
Tablo2.2.....	16
Tablo2.3.....	18
Tablo3.1.....	21
Tablo3.2.....	27
Tablo3.3.....	27
Tablo4.1.....	35
Tablo4.2.....	36

BÖLÜM 1

1. MATLAB'A GİRİŞ:

MATLAB; (MATrix LABoratory); ilk defa 1985'de C.B Moler tarafından matematik ve özellikle de matris esaslı matematik ortamında kullanılmak üzere geliştirilmiş etkileşimli bir paket programlama dilidir. İlk sürümleri FORTRAN diliyle yazılmış olmakla beraber son sürümleri (2000 yılı itibariyle 5.1 dir) C dilinde hazırlanmıştır. MATLAB mühendislik alanında; sayısal hesaplama, veri çözümleri ve grafik işlemlerinde kullanılabilecek genel amaçlı bir program olmakla beraber özel amaçlı modüler paketlere de sahiptir. Control Toolbox, Signal Toolbox gibi paket programlar (bilgisayar destekli denetim sistemi tasarımı) paketler olup bunlar denetim sistemlerinin tasarımında çok etkili araçlardır. Ayrıca WINDOWS ortamında çalışan SIMULINK, etkileşimli benzetim programlarının hazırlanması ve çalıştırılmasında büyük kolaylıklar sağlamaktadır.

Bugün (2000 yılı itibariyle) 5.2 sürümüne ulaşan MATLAB'ın değişik bilgisayar sistemlerinde çalışan çok çeşitli sürümleri vardır. En basiti Öğrenci sürümü olup bu sürüm Profesyonel MATLAB benzeri özelliklere sahiptir. Yalnız bazı sınırlamalar vardır.

1.1 MATLAB ürün ailesi

MATLAB ,

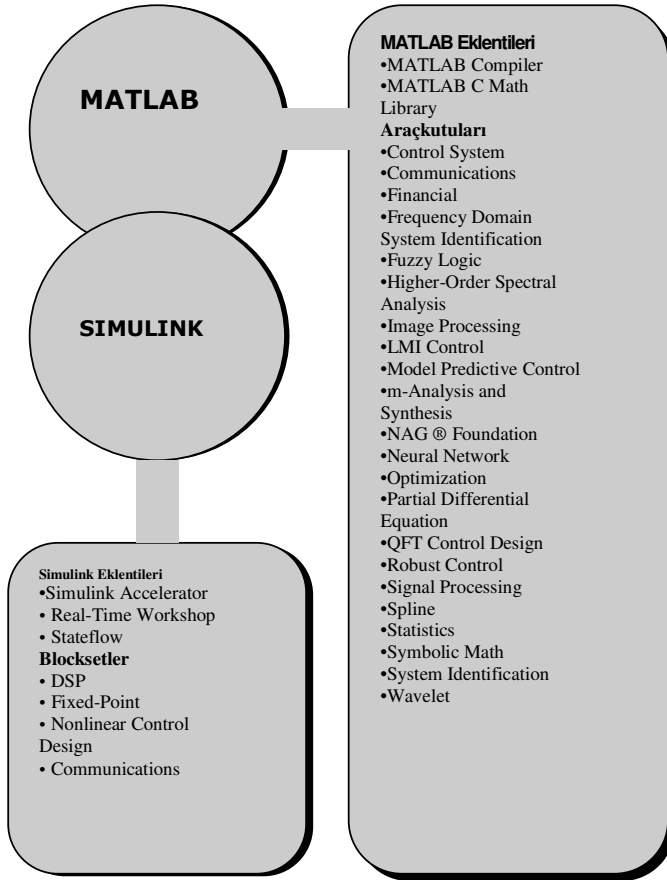
bütün mathworks ürünleri için bir alyapıdır. MATLABta , kullanımı kolay olan tek bir ortamda sayısal hesaplama , 2-d ve 3-d grafikleri ve dil yeteneklerini birleştirilmiştir. MATLAB eklentileri, MATLABta geliştirilen sistemlerin gerçekleştirmesini destekleyen seçimli araçlardırlar .

Araç kutuları,

özel sorunların çözümü için MATLABa uyarlanan MATLAB işlevlerinin bir tür kitaplıklarıdır. Araç kutuları, açık ve eklenebilir olup; kendi algoritmalarını ve ekini görebilirsiniz.

Simulink,

Blok diyagram arayüzü ile doğrusal olmayan simulasyonu ve "canlı" simulasyon yetenekleriyle öz sayıları, MATLAB'ın grafiklerini ve dil fonksiyonlarını birleştiren bir sistemdir.



Şekil 1.1

Simulink eklentileri ,

simulinkte geliştirilen sistemlerin gerçekleştirilmesini destekleyen seçime dayalı araçlardır.

Bloksetler,

özgöl uygulama alanlarında kullanım için tasarlanmış simulink bloğunun toplamlarıdır.

1.2 MATLAB araç kutuları(Toolboxes):

MATLAB'ın kullanılabilirliği ve çok yönlülüğü, uygulamalara özgü çeşitli araç-kutuları eklemekle geliştirilebilir. Araç kutuları, çeşitli bilimsel alan ve konularda yazılan MATLAB fonksiyon dosyalarından oluşurlar. Aşağıda kısaca açıklanan, değişik bilim dalları ile ilgili olarak yazılmış hazır araç-kutuları yer almaktadır.

• **Signal Processing Toolbox (Sinyal İşleme Araç kutusu):** 1-boyutlu ve 2-boyutlu sayısal sinyal işleme (zaman serilerinin analizi) ile ilgili fonksiyonlardan oluşmaktadır. Ayrıca, sayısal filtreler için geliştirilen analiz ve tasarım fonksiyonları ile güç spektrumu analizine ilişkin fonksiyonları da içermektedir.

• **Control Systems Toolbox (Kontrol Sistemleri Araç kutusu):** Durum uzayı tekniklerini kullanarak kontrol mühendisliği ve sistemler teorisi ile ilgili fonksiyonlardan oluşmaktadır.

• **System Identification Toolbox (Sistem Tanımlama Araç kutusu):** Parametrik modelleme ve sistem tanımlama ile ilgili fonksiyonlardan oluşmaktadır.

• **Neural Network Toolbox (Yapay Sinir Ağları Araç kutusu):** Yapay sinir ağları için tasarım ve benzetim (simülasyon) fonksiyonlarından oluşmaktadır. Bu fonksiyonlar birkaç kategoriye ayrılırlar. Bunlar; 1) ağ benzetimi için transfer fonksiyonlarının belirtilmesi, 2) ağ parametrelerini güncelleştirmek için 'kuralların belirtilmesi, 3) veriler üzerinde ağın işlem yapabilmesi için fonksiyonların belirtilmesi. Bazı

fonksiyonlar, lineer ve Hapfield ağlarının doğrudan tasarımı için kullanılabilirler.

- **Spline Toolbox (Oluşum Araç kutusu):** Oluşumlar ile ilgili M-dosyalarından oluşmaktadır. Oluşum araç kutusu fonksiyonel modellemeye kullanılmaktadır. Eğrilerin modellenmesi, verilere göre eğri uydurulması, fonksiyonel denklemlerin çözülmesi vb. için oldukça kullanışlıdır.

- **Robust-Control Toolbox (Robot Kontrol Araç kutusu):** Robot kontrol sistemleri tasarımı ile ilgili fonksiyonlardan oluşmaktadır.

- **μ -Analysis and Synthesis Toolbox (p.Analiz ve Sentez Araç kutusu):** μ -Analiz ve sentez tekniklerinin kullanılarak robot ve lineer kontrol sistemlerinin incelenmesi ve tasarlanması için yazılmış fonksiyonlardan oluşmaktadır.

1.3 MATLAB'ın kullanım amacı ve alanı:

MATLAB tüm mühendislik alanında, sayısal hesaplamalar, veri çözümlenmesi ve grafik işlemlerinde kolaylıkla kullanılabilen bir program dilidir. FORTRAN ve C dili gibi yüksek seviyeli programa dili ile yapılabilen hesaplamaların pek çoğunu MATLAB ile yapmak mümkündür. Ayrıca bunu yanında diğer programlama dillerine göre MATLAB'ta daha az sayıda komutla çözüm üretmek mümkündür. Gerçekte MATLAB, M-dosyaları (M-Files) olarak biline pek çok sayıda fonksiyon dosyalarından, alt programlardan ibarettir. Hazırlanması düşünülen bir program içinde M-dosyalarını kullanmak suretiyle komut sayısını çok kısa tutmak mümkündür.

MATLAB'ın Kullanım Yerleri:

Denklemler takımının çözümü, doğrusal ve doğrusal olmayan diferansiyel denklemlerin çözümü, integral hesabı gibi sayısal hesaplamalar

- Veri çözümlene işlemleri
- İstatiksel hesaplamalar ve çözümlenmeler
- Grafik çizimi ve çözümlenmeler
- Bilgisayar destekli denetim sistemi tasarımı
- Devre analizinde düğüm kol denklemlerinin çözümü

1.4 MATLAB'ı kurulumu ve çalıştırılması:

1.4.1 MATLAB'ın Kurulumu

MATLAB, gerekli kaynak programın bulunduğu disketler ya da CD üzerinden KUR.EXE dosyası çalıştırılarak çok rahat bir şekilde sisteme kurulabilir. Kurulum aşamasında kullanıcıya çeşitli metin ve diğer uyarıcılar ile yardım edilerek kurulum tamamlanır.

1.4.2 MATLAB'ın başlatılması

MATLAB'ı iki şekilde başlatmak mümkündür.

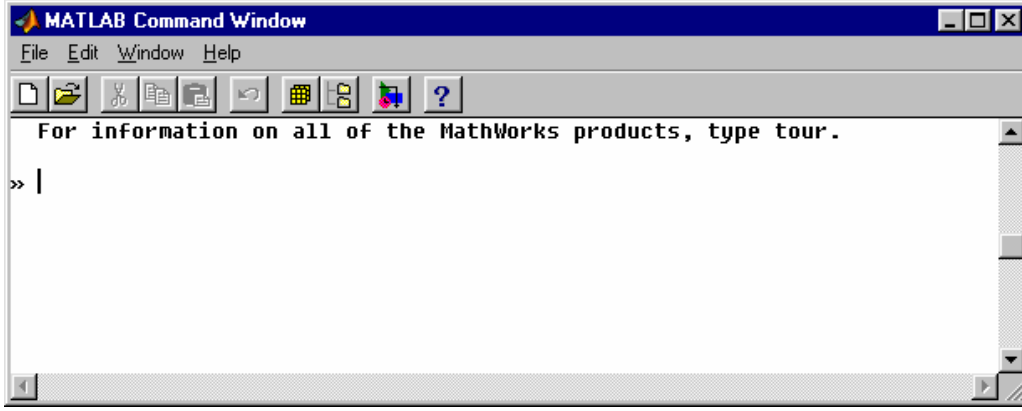
a-Windows program yöneticisinden başlatmak: MATLAB'ı doğrudan program yöneticisi etkinleştirildikten sonra ekranda görünen MATLAB yazılı simgeye çift tıklayarak başlatabiliriz. Bu MATLAB'ı çalıştırarak komut penceresini ekrana getirir.

DOS komut satırından başlatmak: DOS komut satırında MATLAB'ı çalıştırmak için BIN alt dizinine geçmek en uygundur. Bu durumda Sadece MATLAB yazmak yeterli olacaktır. Bu komut önce Windows'u ardından MATLAB'ı başlatır.

1.4.3 MATLAB komut penceresi

MATLAB açıldığında karşımıza gelen pencere (Şekil1) MATLAB'ın komut penceresidir. Komut penceresi kullanıcı ile MATLAB komut yorumlayıcısı arasında iletişimi sağlayan bir ara yüzdür. Yorumlayıcı hazır hale

geldiğinde (>>) iletisi karşımıza gelir. Bu ileti MATLAB'a komut ya da komut dizileri girilebileceğini gösterir.



Şekil 1.2

1.4.4 Genel Komutlar

Açıklama: Osman!
Komut Penceresi Yazmayı
Unutma

Demo komutu(demo): Eğer MATLAB ilk defa kullanılıyor ya da belli komutların çalışmasını merak ediliyorsa `demo` komutu ile demonstrasyon listesini görüntülenebilir. Listedeki yapacağınız bir seçimle seçtiğimiz işlemin icraatlarını adım adım izleme imkanı bulabilirsiniz.

Saklama ve geri çağırma komutları(save-load):

Bilgisayarınızda MATLAB ile çalışırken bilgisayarınızı kapatmayı arzulayabilirsiniz. Daha sonra geri dönerek kaldığınız yerden etmek isteyebilirsiniz. İşte bunu başarmak için kullanıyor olduğunuz bütün değişkenleri yeniden ayarlamadan bilgisayarınızı kapatmadan önce

```
>>save
```

komutunu kullanın.

Bu komut kullanımda olan MAT dosyasını alt dizininde veya MATLAB dosyanızda MATLAB.mat diye yapar veya üstüne yazar. Sonra MATLAB.mat da yer alan çalışma alanını yeniden çağırma ihtiyacı duyduğunuzda

```
>>load
```

komutunu girmelisiniz.

MATLAB.mat haricinde başka bir isim ile de değişkenleri saklayabilirsiniz.

Örneğin

```
>>save dosyam
```

Bunun yanında

```
>>load dosyam
```

komutu ile de saklamış olduğunuz değişkenleri geri çağırabilirsiniz.

Büyük ve Küçük harf duyarlılığı(casesen off, casesen):

MATLAB büyük ve küçük harf karakterlerine duyarlı bir dildir. Yani yani simla verilmiş fakat karakterlerin durumu bakımında farklılık arzeden isimler farklıymış gibi algılanır. Örneğin dscrite, DSCRITE ve Dscrite atamaları farklı şekilde algılanır. Bu durumda bir isme birden fazla değer atamak mümkündür. Eğer istenirse `casesen off` komutu ile bu durum sona erdirilebilir. Böylece büyük-küçük harf duyarlılığı ortadan kalmış olur.

Değişkenlerin ve Matrislerin durumu(who, whos, what, size):

MATLAB'da; kullanıcı tarafından program içinde tanımlanan değişkenlerin ve matrislerin durumunu gözden geçirmek için `who, whos, what, size` gibi kullanışlı komutlar bulunmaktadır.

`who`;komutu kullanıcı tarafından tanımlanan değişkenlerin listesini görüntüler,

`whos`;komutu bu değişkenleri boyutları ile birlikte ve bu değişkenlerin sıfır olamayan sanal kısımları olup olmadığını da görüntüler,

`what`;komutu kullanıcı diskinin halihazır dizininde mevcut M-dosyaları listesini verir.

Yardım Komutu(help): Yardım imkanı MATLAB'ta en önemli bir kaynaktır. Çalışma ortamında `help` komutu ile yardım alınabilecek dosyaların bulunduğu dizinler ve sonrada yardım alınabilecek komutlar

görüntülenir.Ardından `help` komutunun ardına yardım istediği konuyu peşi sıra yazarak istediği konuda yardım alabilir. Örnek olarak

```
>>help sin
```

komutu girilerek

```
SIN          SIN(X) is the sine of the elements of X
```

şeklinde sinüs ile ilgili bir yardım alınabilir.

Yazım(type): `type` komutu ile bir dosyanın içeriğini görüntüler. M-dosyalarının görüntülenmesinde dosya adından sonra M uzantısının yazılmasına gerek yoktur. Örneğin `type device` komutu `device.m` dosyasında yazılı olan tüm satırları görüntüler. Buna karşılık `type device.m` komutu ise `device` dosyasında yer alan % işareti ile belirlenmiş açıklama satırları görüntüler.

Kısaca genel komutların isimleri ve açıklamaları Tablo 1.1 de verilmiştir.

Tablo.1.1

Komut	İşlev
help	MATLAB'ın operatör ve fonksiyonlarını tanımlar
who	Değişkenlerin isimlerini listeler
whos	Değişkenlerin simlerini ve boyutlarını listeler
what	Diskinizdeki M-dosyalarını listeler
size	Argümanların boyutlarını verir
length	Argümanların maksimum boyutlarını verir
clear	Çalışma ortamındaki tüm değişkenleri temizler
quit	MATLAB ortamını sona erdirir
save	MATLAB çalışma ortamında bir MAT-dosyasını saklar

1.4.5 Özel Karakterler

MATLAB'da ifadeler formunda veya MATLAB kontrolünü kullanabilmeniz için özel karakterlere sahiptir. Tablo 1.2'de bu karakterlerin bir özeti verilmiştir.

Açıklama: Osman!
Komut Penceresi Yazmayı
Unutma

[]:Köşeli parantezler, vektörleri ve matrisleri biçimlendirmek için kullanılır. Örneğin [5.45 9.3 sqrt(-6)] elemanları boşluklarla ayrılmış

Tablo 1.2

Karakter	Fonksiyon
[]	Matris ve vektörleri tanımlar
()	Kontrol işlemi üstünlüğü
	Matris indislerini ve fonksiyon argümanlarını ayırmak için kullanılır
;	Matrislere ait sınırları sona erdirir
%	Açıklama eklemek için kullanılır
:	Sütun işareti veya vektör
!	Sistem fonksiyonlarını yerine getirir(MS-DOS)

üç elemanlı bir vektördür. Bunun yanında [5 6 45;12 91 2], üç satır ve üç sütundan ibaret 3x3 lik bir matrisi gösterir. Burada noktalı virgül (;) matris oluşturulmasında bir satırı sona erdirip bunu izleyen satırın oluşmasını sağlar. A=[] biçiminde bir bildirim boş bir matrisi A değişkenine atar.

():MATLAB da normal parantez çeşitli çalışmalarla çeşitli yerlerde kullanılır. En önemlisi; aritmetik deyimlerin üstünlüğünü belirlemek ve fonksiyon argümanlarını kapatmak için kullanılır. Eğer X ve Y bir vektör ise X(Y) de bir vektördür ve [X(Y(1)),X(Y(2)),...,X(Y(N))] şeklinde ifade edilir.

./:Kesir ayırma işaretidir. 1452/10, 1.452 ve .1452e1 aynı anlama gelir. Bu işaret aynı zamanda eleman elemana çarpma ve bölme işleminde de kullanılabilir. Buna göre kullanım biçimleri .*, .^, ./, .\ veya ./ şeklinde.

...:Bir tek satıra sığmayan ifadelerin devam ettiğini belirtmek için kullanılır.

;;:Bu işaret bir bildirimde elde edilen sonuçların programın icrası sonunda akıllarda görüntülenmesini engellemek için kullanılır.

%:Açıklama yapılacağı zaman ilgili satırlara bu işrele başlanarak yapılır. % işareti ile başlayan MATLAB tarafından dikkate alınmaz ve icraya da sokulmaz.

!:Herhangi bir yazı DOS işlemi görür. Böylece MATLAB içerisinde iken DOS komutları ile çalışılabilir.

::İşaret sütun işareti olarak kullanılır ve her yerde kullanılabilir. Örneğin; J:K dan başlayarak 1 er ler K ya kadar artan bir dizi oluşturur ve [J, J+1,...,K] ile aynı anlama gelir.

':Tırnak işareti matrislerin transpozunun alınmasını sağlar. M', M matrisinin karmaşık eşlenik transpozonu M'. ise eşlenik olmayan transpozunu sonuçlandırır. Devam

İlgili Örnekler:

Aşağıdaki deyimlere göre

```
syms a b c d;
```

```
A = [a b; c d];
```

```
A*A/A
```

```
A*A-A^2
```

Entere bastığımızda

```
[ a, b]
```

```
[ c, d] sonucu alınır.
```

Aşağıdaki deyimlere göre

```
syms a11 a12 a21 a22 b1 b2;
```

```
A = [a11 a12; a21 a22];
```

```
B = [b1 b2];
```

```
X = B/A;
```

```
x1 = X(1)
```

```
x2 = X(2)
```

Entere bastığımızda

$x_1 =$

$$(a_{21} \cdot b_2 - b_1 \cdot a_{22}) / (-a_{11} \cdot a_{22} + a_{12} \cdot a_{21})$$

$x_2 =$

$$(-a_{11} \cdot b_2 + a_{12} \cdot b_1) / (-a_{11} \cdot a_{22} + a_{12} \cdot a_{21})$$

sonucu alınır.

1.5 MATLAB Komut Penceresi Menüleri:

Bir çok programda olduğu gibi MATLAB'da da komut penceresi menüleri büyük kolaylıklar sağlar. Buna göre MATLAB'daki menüler ve işlevleri şu şekildedir.

1.5.1 File (Dosya) Menüsü:

File menüsü dosya veya dosyaların oluşturulması ve yazdırma işleminin ayarlamalarının olduğu komutları içerir.

New: Bu komut şu seçenekleri içerir.

M-File, yeni M-dosyası oluşturmak için boş bir pencere açar.

Figure, yeni bir şekil penceresi oluşturur.

Model, ywni bit SIMULINK penceresi oluşturur.

Open M-File: Bir dosya seçebileğiniz pencere ekrana getirerek dosya adı girilmesini veya gereken dosyanın seçilmesini ister ve ardında seçilen yada ismi yazılan dosya metin düzenleyici programı çalıştırılarak açılır.

Open Selected: Komut penceresinde seçilerek belirtilen bir M-dosyasını varsayılan düzenleyiciyi çalıştırarak açar.

Save Workspace As...: Çalışma alanını kaydetmek için bir iletişim kutusu görüntülenir, yeni bir dosya adı girmeniz gerekmektedir.

Run M-File: Dosya Yöneticisi'nin Çalıştır... komutuna benzer. Bir M-dosyası adı girmeniz veya seçmeniz için bir iletişim kutusu görüntülenir ve belirtilen M-dosyasını çalıştırır.

Look for Selected: MATLAB'in `lookfor` komutunu çalıştırır. MATLAB'in arama yolunda bulunan tüm M-dosyalarının içindeki yardım

metinlerinin ilk açıklama satırlarını tarayarak komut penceresi içinde seçilen katarı araştırır ve sonucu ekranda görüntüler.

Print...: Komut Penceresinde seçilen metni o için yüklü bulunan bir yazıcıya döker. Eğer seçilen metin yoksa, tüm MATLAB oturumu boyunca girilen metni yazdırır.

Printer Setup...: O andaki yazıcı ayarlarını ve seçeneklerini(renk tonu gibi) değiştirmek için bir iletişim kutusunu görüntüler.

Exit MATLAB: MATLAB oturumunu kapatır.

1.5.2 Edit (Düzen) Menüsü:

Edit (Düzen) menüsü komutları kullanıcıya düzenleme fonksiyonlarını uygulamak için büyük kolaylıklar sağlarlar.

Cut: Komut Penceresi'nde seçilen metni 'keser' ve ortamda saklar.

Copy: Komut Penceresi'nde seçilen metni 'kopyalar' ve ortamda saklar.

Paste: O andaki pano içeriğini komut satırına yapıştırır.

Clear Session: Komut penceresinin içeriğini siler. Bu komut, `clc` komutu ile aynı görevi icra eder.

1.5.3 Options (Seçenekler) Menüsü:

Bu menü; MATLAB'da pencere seçeneklerini ayarlamak, varsayılan düzenleme programını seçmek ve MATLAB'in `format` ve `echo` komutlarının işlevlerini değiştirmek için kullanılır.

Numeric Format: Ekran çıktı biçimlerini değiştirmek için bu komutu kullanabilirsiniz. Bu komut şu seçeneklerden oluşmaktadır.

Turn Echo On/Off (Yansıma Açık/Kapalı): Yansıma durumu için açık ve kapalı arasında geçiş yapar. `Echo on` ise verilen bir komutun sonucu ekranda görüntülenmesini sağlar.

Enable/Disable Background Process: Artalan işlemlerinin olup olmaması arasında geçişi sağlar.

Font...: Yazı fontları iletişim kutusunu açarak buradan komut penceresinde kullanılan font ve artalan rengini seçebiliriz.

Tablo1.3

Komut	Etki
Short	Sabit-noktalı sayıları 5 haneli olarak gösterir (varsayılan).
Long	Sabit-noktalı sayıları 15 haneli olarak gösterir.
Hex	Onaltılı gösterim biçimini seçer.
Bank	Dolar ve sent (Lira ve Kuruş) gösterim biçimlerini seçer.
Plus	Sayıların önüne pozitif, negatif veya sıfır olma durumlarına göre +, — veya boşluk koyar. Kompleks sayılarda sanal kısım dikkate alınmaz.
Short e	Kayan-noktalı sayıları 5 haneli olarak yazar.
Long e	Kayan-noktalı sayıları 15 haneli olarak yazar.
Rational	Gerçel sayıları, tamsayıların oranı olarak verir.
Loose	Katsayıları, matrisin önünde satır-atla karakteri ile birlikte görüntüler.
Compact	Katsayıları, matrisin önünde satır-atla karakteri olmadan görüntüler.

Editor Preference...: Bir metin düzenleme komutu işletilmek istendiğinde gereken metin düzenleyici programı varsayılan olarak belirtmemizi sağlar.

1.5.4 Windows Menüsü:

Windows (Pencereler) menüsü, MATLAB komut penceresi ve diğer şekil pencerelerini listeler. Bu menüden bir öğenin seçilmesi, o öğenin etkin olmasını sağlar.

1.5.5 Help (Yardım) menüsü:

Help (Yardım) menüsü, MATLAB'e ait etkileşimli yardım programına erişimi sağlar. Bu menü, aşağıdaki komutlardan oluşmaktadır.

Table of Contents (İçindekiler Tablosu):

MATLAB için yardım almak mümkün olan konuların listesini görüntüler.

Index (Dizin):

Yardım almak mümkün olan MATLAB fonksiyonlarını alfabetik bir liste olarak görüntüler.

Help Selected (Seçilen için Yardım):

Komut penceresinde seçilen bir öge için yardım araştırır.

About... (Hakkında...):

MATLAB için tanıtıcı bilgiler içeren Hakkında kutusunu görüntüler.

1.5.6 Şekil Penceresi Menüleri

MATLAB, tüm grafik çıktılarını 'şekil penceresi' olarak adlandırılan ayrı grafik pencerelerinde görüntüler. Bu pencerelerin özelliklerini, MATLAB komut ve fonksiyonları ile kontrol etmek mümkündür. MATLAB, kullanıcıların, istedikleri kadar (sistem kaynakları ile sınırlı olabilir) şekil pencereleri oluşturmalarına ve komut penceresinden işletilen komutlarla grafiklerin bu pencereler içinde çizdirilmelerini sağlar.

1.5.6.1 File (Dosya):

Şekil pencerelerinin açılması, kapatılması, oluşturulması ve yazıcı çıktılarının alınması komutlarını içerir.

Komutların kullanımları yukarıda verilmiştir.

1.5.6.2 Edit (Düzen)

Bu menü aşağıdaki seçeneklerden oluşur.

Copy to Metafile

Şekil penceresinin içeriğini Pano'ya Windows metafile (.wmf) biçiminde kopyalar.

Copy to Bitmap

Şekil penceresinin içeriğini Pano'ya Windows bitmap (.bmp) biçiminde kopyalar.

Clear Figure

Şekil penceresinin içeriğini siler. Bu komut, clf komutu ile eşdeğerdir.

BÖLÜM 2

2.MATRİSLER:

2.1 İfadeler:

MATLAB diğer programlama dillerine göre daha avantajlıdır. Bir çok programlama dilinden farklı matematiksel ifadeler sunar ve bu ifadeler bütün matrisleri içerirler . İfadelerin temel blokları

- Değişkenler
- Sayılar
- Operatörler
- Fonksiyonlar'dır.

2.2 Değişkenler:

MATLAB'da, herhangi bir tip tanımlaması veya boyut ifadesine gerek yoktur. MATLAB, yeni bir değişken ismi ile karşılaştığında, otomatik olarak `ans` isminde bir değişken oluşturur ve uygun bir bellek miktarı ayırır. Eğer değişken zaten varsa, MATLAB gerekli bir bellek ayırdığında içeriği değişir. Örneğin,

```
ogrenci_sayı=51
```

`ogrenci_sayı` diye isimlendirilen 1x1 matrisi oluştur ve 21'i yükle.

Değişken isimleri; bir harfdan, sayıdan veya alt çizgiden oluşur. sayısının tarafından takip edilen harftan oluşurlar. MATLAB, sadece değişken isminin ilk 31 karakterini kullanır. MATLAB, büyük ve küçük harfe duyarlıdır, büyük harf ile küçük harfi ayırdeder. A ve a değişkenleri aynı değildir.

2.3 Sayılar:

MATLAB'da sayılar yaygın olarak kullanılan onluk tabanda ifade edilirler. Bunun yanısıra onluk tabanda üstel olarak veya `i` veya `J` olarak kompleks sayı biçimlerinde de ifade edilebilirler. Örnek olarak,

3	-99	0.0001
9.6397238	1.60210e-20	6.02252e23
1i	-3.14159j	3e5i

sayıları gösterilebilir.

Sayıların duyarlılığını belirtmek için kullanılan eps sayısı onluk tabanda 16 basamaklı olarak gösterilmekte ve yaklaşık olarak 10^{-308} den 10^{-308} e kadar olan tüm sayıları kapsayabilmektedir.

2.4 Operatörler(sayısal işlemciler):

Matematiksel ifadeleri oluşturmak için operatörler ve önceden tanımlanmış sembolleri kullanabilirsiniz. Operatörler özetle şunlardır:

Tablo 2.1

İşlem	Cebirsel Biçimi	MATLAB Karşılığı
Toplama	$a + b$	a+b
Çıkarma	$a - b$	a-b
Çarpma	$a \times b$	a*b
Bölme	a / b	a/b
Sola Bölme	b / a	a\b
Üs Alma	a^b	a^b

İki düz (skaler) sayı atasındaki aritmetiksel işlemler Tablo2.1'de gösterilmiştir. Bir deyim aşağıda olduğu gibi değer atanarak belli bir değer içinde saklanabilir.

$$x=a + b$$

Bu ifadede a ve b nin toplandığı ve x değişkeni içinde saklandığı belirtilmektedir. Bu atama işlemini; a içindeki bir değer b içindeki bir değerle toplanarak bu toplamın x değişkenine atanacağı şeklinde yorumlamak mümkündür. Eğer b,r atama işlemi bu şekilde yorumlanacak olursa, aşağıda verilen bir MATLAB bildirimi de geçerli olur.


```
say = say + 1;
```

Açık bir şekilde bu bildirim geçerli bir cebirsel bildirim olamaz, fakat MATLAB içindeki 1'in `say` içindeki bir değere ilave edileceğini ve sonucun tekrar `say` içinde saklanacağını belirtir. Sonuçta `say` içindeki değer her seferinde 1 artacağını belirtmesine denktir.

Belli bir değişken tanımlamadan girilen deyimlerin icrasında `ans` isimli bir değişken içinde otomatik olarak saklanır. Her defasında `ans` içindeki değer bir öncekinin yerini alır. Burada `ans` İngilizce cevap anlamına gelen `answer` kelimesinin kısaltılmış şeklidir.

Matrislerle yapılan işlemlerde bölme işlemi için iki farklı sembol kullanılmaktadır. Bunun yanında eğer sayılar skaler ise iki bölme işleminin sonucu da aynı değeri gösterecektir. Örneğin $3/2$ ile $2\setminus 3$ ifadelerinin sonuçları aynı olup 1.5'dir.

Aritmetiksel İşlemlerde Öncelik Durumu: Tek bir aritmetiksel durum içinde birden fazla durum bir arada bulunabildiğine göre hangi işlemin öncelik hakkına sahip olunduğunun bilinmesi yerinde olacaktır. Aşağıda MATLAB'da kullanılan işlemlerde işlemlerin öncelik listesi verilmiştir.

Tablo 2.2

Öncelik	İşlem
1	Parantez
2	Üs alma, soldan sağa doğru
3	Çarpma ve bölme, soldan sağa doğru
4	Toplama ve Çıkarma, soldan sağa doğru

2.5 Kompleks Sayılar:

MATLAB'da kompleks sayılar, tüm işlemlerde ve fonksiyonlarda i ve j özel fonksiyonları ile ifade edilir. Örneğin;

$$z=2+5i, z=1+1.83j, z=1+3j$$

Kompleks matrisleri girerken dikkat edilmesi gereken nokta deęerler arasında bırakılacak boşluęun yeridir. Örneęin $3+4i$ ifadesi, iki farklı deęer üretir.

2.6 Fonksiyonlar:

MATLAB'ın çok güçlü ve çok kapsamlı bir fonksiyon yapısı vardır. MATLAB'da bazı fonksiyonlar daha önceden yerleřtirilmiř bazıları ise sonradan dışarıdan yerleřtirilmiř M-dosyalarından oluřan MATLAB kütüphanelerinde tanımlanmıřlardır. Bunun yanında özel uygulamalar için kullanıcıların kendi fonksiyonlarını yazarak kendi kütüphanelerini oluřturması mümkündür. Kullanıcı tanımlı fonksiyonlar da dięer fonksiyonlar gibi kullanılabilirler. MATLAB'da mevcut analitik fonksiyonları genel olarak ařaęıdaki kategorilere ayırabiliriz.

1. Temel matematiksel fonksiyonlar
2. Özel fonksiyonlar
3. Temel matrisler ve matris işlemleri
4. Matrisleri çarpanlarına ayırma
5. Veri analizleri
6. Polinomlar
7. Diferansiyel denklem çözümleri
8. Lineer olmayan denklemler ve optimizasyon
9. Sayısal integral hesaplamaları
10. Sinyal işleme

Bütün fonksiyonların görevini yazmak bu tezi sınırlarını ařacaęından burada sadece bazı temel matematiksel fonksiyonların açıklamaları verilecektir. Ayrıca dięer önemli fonksiyonlar da tezin sonundaki eklerde verilmiřtir.

Tablo 2.3

Fonksiyon	Açıklama
abs	Mutlak değer hesaplar ve karakterleri sayıya dönüştür.
angle	faz açısını radyan cinsinden hesaplar
sqrt	sayıların karekökünü hesaplar
real	bir karmaşık sayının gerçek kısmını hesaplar
imag	bir karmaşık sayının sanal kısmını hesaplar
conj	karmaşık eşlenik sayıyı hesaplar
round	en yakın tamsayıya tamamlar
fix	en yakın tamsayıyı sıfıra doğru yuvarlar
floor	en yakın tamsayıyı eksi sonsuza doğru yuvarlar
ceil	en yakın tamsayıyı artı sonsuza kadar yuvarlar
sign	bir sayı veya matris elemanlarının sıfırdan büyük veya küçük olup olmadığını tespit eder
rem	bir bölme işleminde kalanı hesaplar

MATLAB, standart basit matematiksel fonksiyonlardan abs, sqrt, expin ve sin gibi sayılar içermektedir. Karekök veya negatif sayının logaritmasını alarak hatasız, uygun karmaşık sonucu otomatik olarak üretir. MATLAB, çok gelişmiş matematiksel fonksiyonları kapsayan bessel aynı zamanda sağlar ve gama fonksiyonunu yerine getirir. bu fonksiyonların çoğu karmaşık tartışmaları kabul ederler. Temel fonksiyonlar örnek verilecek olursa:

```
>>rem(12,7)
```

```
ans=5
```

```
>>exp(1)
```

```
ans=2.7183
```

```
>>fix(3.5)
```

```
ans=3
```

```
>>ceil(3.5)
```

```
ans=4
```

```
>>round(3.5)
```

```
ans=4
```

BÖLÜM 3

3. MATLAB'DA MATRİSLER VE TEMEL MATRİS İŞLEMLERİ:

Her ne kadar 4. versiyondan itibaren MATLAB'a The Symbolic Toolbox ilave edilerek sembolik matematiksel işlem yapma olanakları sağlanmış ise de MATLAB elemanları kompleks olabilen dörtgensel boyutlu sayısal matrislerle çalışan bir programdır. Bazı durumlarda özel olarak bu matris 1x1 boyutunda olabilir. Benzer şekilde tek satır veya tek sütundan oluşan vektörler olabilirler. MATLAB'ın matrisler ile çalışması demek; verileri matrisler biçiminden değerlendirmesi ve çözümleri matris esasına göre yapması ve sonuçları da matris biçiminde düzenlenmesi demektir.

3.1 Matrisleri Girmek:

Matrisler girilirken köşeli parantez kullanılarak ([]) ve aşağıdaki yollardan biri kullanılarak girilir.

- Elemanları bir tam liste olarak girmek
- Yerleşik deyim ve fonksiyonları kullanarak matrisler üretmek
- M-dosyalarını kullanarak matrisler oluşturmak
- Dış veri dosyalarından matrisleri yüklemek

Bunların arasından en kolay olanı tam liste olarak yazmaktır. Buna göre bir matrisi girerken ,

1. Elemanlar, baştan ve sondan köşeli parantezlerle kapatılır.
2. Elemanların arasına boşluk veya virgöl konulur.
3. Matris satırlarını ayırmak için de noktalı virgöl kullanılır.

kurallarına dikkate alınmalıdır. Örneğin,

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

```
A =
```

```
1 2 3
```

```
4 5 6
```

şeklinde veriler matrisler biçiminde tanımlanır. MATLAB daha sonra kullanılmak üzere verileri A matrisinde saklar.

Büyük matrisleri girerken noktalı virgül yerine satır atlama karakteri kullanılabilir. Yukarıda A matrisi farklı bir şekilde aşağıda gösterildiği gibi de girilebilir.

```
A = [1 2 3
     4 5 6]
```

Sonuç olarak aynı değerler elde edilir.

3.2 Matrislerde Elemanlar:

Matris elemanları rasgele rakamlardan oluşabileceği gibi MATLAB'ın deyimlerinden de oluşabilir. Örneğin,

```
a = [-1.3 sqrt(3) (1+2+3)*475]
a =
-1.3000  1.7321  4.8000
```

Küçük matrisleri eleman olarak kullanıp daha büyük matrisler oluşturmak mümkündür. Örneğin,

```
M = [1 3 5
     7 8 9]
```

matrisine aşağıda gösterildiği şekilde bir satır eklemesi yapabilir.

```
M = [M; [10 13 15]]
```

Bu durumda M matrisi

```
M = [1 3 5
     7 8 9
     10 13 15]
```

durumunu alır.

Bunun yanında iki nokta üst üste (:) kullanarak büyük matrislerin belli bir kısmını kullanabiliriz ya da diğer ifade ile bu matrisi küçültebiliriz. Örneğin,

```
M=M(1:2, :)
```

bildirimi M matrisinin ilk iki satırını alarak M matrisi olarak saklar. Böylece baştaki orijinal M matrisi elde edilmiş olur.

3.3 Genel Matrisler:

MATLAB'da kullanıcı verileri kendi oluşturacağı matrisler ile tanımlayabileceği gibi MATLAB'ın kendi özel matrislerinden de istifade edilebilir. Bunların önemlileri ones, eye, zeros, magic, rand, randn olup işlevleri şu şekildedirler:

Tablo 3.1

zeros	Tüm elemanları sıfır olan bir matris oluşturur
Ones	Tüm elemanları 1 olan bir matris oluşturur
Rand	Düzgün dağılımlı rasgele sayılar veya matrisler oluşturur
randn	Normal dağılımlı rasgele sayılar veya matrisler oluşturur
Eye	Bir birim matris oluşturur
Magic	Sihirli kare matris oluşturur

MATLAB'ın dört temel matrisi ile ilgili birkaç örnek:

Örnek:1

```
Z = zeros(2,4); % 2 satırdan oluşan ve bütün elemanları 0 olan bir matris üretir
```

```
Z =
    0    0    0    0
    0    0    0    0
```

Örnek:2

```
F = 5*ones(3,3); %3x3 lük oluşacak tüm elemanları 1 olan matris 5 ile çarpılmıştır
```

F =

```

5     5     5
5     5     5
5     5     5

```

Örnek:3

```
N = fix(10*rand(1,10));
```

N =

```

4     9     4     4     8     5     2     6     8     0

```

Örnek:4

```
R = randn(4,4);
```

R =

```

1.0668    0.2944   -0.6918   -1.4410
0.0593   -1.3362    0.8580    0.5711
-0.0956    0.7143    1.2540   -0.3999
-0.8323    1.6236   -1.5937    0.6900

```

3.4 Matrislerde İşlemler:

MATLAB'ın temelini matris işlemler oluşturur. Matris işlemlerinin tümünü MATLAB'ta uygulamak mümkündür. Yalnızca bazı karakterlerde kısıtlama söz konusu olabilir.

3.4.1 Matrislerin Transpozu:

Bir matrisin transpozu satırlarla sütunların yer değiştirdiği başka bir matristir.. Genelde tranpoz üst indis olarak T olarak ifade edilir.

Örneğin bir A matrisi,

$A=[1 \ 2 \ 3; 4 \ 5 \ 6; 6 \ 7 \ 8]$ şeklinde tanımlarsak bu A matrisinin transpozu olan A^T ,

$A^T=[1 \ 4 \ 7; 2 \ 5 \ 8; 3 \ 6 \ 9]$ şeklindedir.

Buradaki yapılan işlem (i,j) konumunda bir değerin (j,i) konuma aktarılmasıdır.

MATLAB'da bir matrisin transpozu (`'`) üstten virgül karakteri ile ifade edilir. Yani A matrisinin transpozu gösterilirken A' şeklinde yazılır. Bilindiği üzere kare matrisin dışındaki matrislerin transpozesi farklı boyuta sahip yeni bir matristir.

Eğer A matrisi karmaşık sayılardan oluşuyorsa A' da karmaşık eşlenik transpozedir. Eşlenik olmayan bir transpoze elde etmek için ya A' ya da `conj(A')` komutları kullanılmalıdır.

Örnekler:

Örnek:1

$$X = \begin{bmatrix} 6 & 0 & -3 \\ 2 & 0 & -1 \\ 8 & 0 & -4 \end{bmatrix}$$

$$Y = X'$$

$$Y = \begin{bmatrix} 8 & 3 & 4 \\ 1 & 5 & 9 \\ 6 & 7 & 2 \end{bmatrix}$$

Örnek:2

$$z = [1+2i \quad 3+4i]$$

$$z' =$$

$$1-2i$$

$$3-4i$$

$$z.' =$$

$$1+2i$$

$$3+4i$$

3.4.2 Matrislerde Toplam ve Fark İşlemleri:

MATLAB'da toplama ve çıkarma işlemleri için + ve - sembolleri kullanılır. Bu işlemler ancak boyutları aynı olan işlemler için tanımlıdır. Örneğin K ve L gibi farklı boyutta iki matris gözönüne alırsak,

$$K+L$$

Toplamı hata sonucunu verir. Çünkü matrislerin boyutları farklıdır. Fakt A ve B gibi eşit boyutlu matrisleri

$$C=A+B$$

Şeklinde toplarsak sonuçta hata olmaz, A ve B matrisleri toplanır ve bu değer C ye atanır. Örneğin,

$$A=[1 \ 2 \ 3; 4 \ 5 \ 6; 7 \ 8 \ 9] \text{ ve}$$

$$B=[1 \ 4 \ 7; 2 \ 5 \ 8; 3 \ 6 \ 0] \text{ matrisleri için}$$

$$C=A+B \text{ toplamının sonucu}$$

$$C=[2 \ 6 \ 10; 6 \ 10 \ 14; 10 \ 14 \ 0] \text{ olur.}$$

Toplama ve çıkarma işlemlerini işlenenlerden biri skaler olduğunda da geçerlidir. Skalari 1x1 boyutunda bir matrisi olarak düşünebiliriz. Bu durumda skaler değer, matrislerdeki tüm değerler ile ayrı ayrı ve tek tek işleme tâbi tutulur.

Örnekler:

$$A =$$

$$1 \ 1 \ 1$$

$$1 \ 2 \ 3$$

$$1 \ 3 \ 6$$

$$B =$$

$$8 \ 1 \ 6$$

$$3 \ 5 \ 7$$

$$4 \ 9 \ 2 \quad \text{matrisleri için}$$

$$X = A + B$$

$$X =$$

$$9 \ 2 \ 7$$

$$4 \ 7 \ 10$$

$$5 \ 12 \ 8$$

$$Y = X - A$$

$$Y =$$

$$8 \ 1 \ 6$$

$$3 \ 5 \ 7 \quad \text{sonuçları alınır.}$$

3.4.3 Matrislerin Çarpılması:

Matrislerde çarpma işlemi için * sembolü kullanılır. Bu işlem işlenen matrislerden her ikisinin iç boyularının aynı olduğu durumlarda geçerlidir. Yani, A*B çarpımında a'nın sütun sayısı ile B'nin satır sayısı eşit ise hesaplama yapılır aksi takdirde hata sonucu çıkar.

Ayrıca matris çarpımı matrislerin karşılık gelen elemanlarının çarpım yolu ile hesaplanamaz. Böyle bir durumda eğer MATLAB'da C=B*A komutu verilecek olursa ,

`C does not exist (C mevcut değildir)`

Şeklindeki bir uyarı mesajı alınır.

A ve B şeklinde iki matrisin matris çarpımının sonucu olan C matrisinin C(i,j) konumundaki değer aşağıda gösterilen toplam denkleminde ifade edildiği gibi birinci matrisin i'nci satırı ile ikinci matrisin J'inci sütununun nokta çarpımıdır.

$$C_{i,j} = \sum_{k=1}^N a_{ik} b_{jk}$$

Örnekler:

Örnek:1

$$A = [1 \ 2; \ 3 \ 4]; \quad B = [5 \ 6; \ 7 \ 8];$$

$$C = A * B$$

$$C =$$

$$19 \ 22$$

$$43 \ 50$$

3.4.4 Matrislerin Bölünmesi:

MATLAB'ta daha önce açıklandığı gibi bölme işlemi için iki farklı (/ ve \) bölme işlemcisi vardır. Eğer A değeri, tekil olmayan bir kare matris ise $A \setminus B$ ve B/A ifadeleri B matrisi ile A matrisinin soldan ve sağdan çarpımları yani, $\text{inv}(A)*B$ ve $B*\text{inv}(A)$

anlamına gelir. Bunun yerine MATLAB'da sonucu doğrudan elde etmek mümkündür. Buna göre;

$X=A \setminus B$ matris denklemleri $AX=B$ çözümünü gösterir.

$X=B/A$ matris denklemleri $XA=B$ çözümünü gösterir.

Soldan bölme, A/B ; B matrisi ile A matrisi satır sayıları aynı ise tanımlıdır.

- Eğer A bir kare matris ise, A/B ifadesi Gauss elimasyonu yardımıyla faktörlere ayrılır. Bu faktörler,

$$AX(:,j)=B(:,j)$$

denklemlerinin çözümünde kullanılır. Buradaki $B(:,j)$ ifadesi B matrisinin j sütununu göstermektedir. Bulunan X matrisi B ile aynı boyutta olan bir matris olur.

- Eğer A bir kare matris değilse $A \setminus B$ ifadesi Householder ortogonelleştirilmesi kullanılarak faktörlere ayrılır. Bu faktörler en küçük kareler yönteminde hesaplanacak denklemler altında kullanılır.

Sonuç $m \times n$ boyutlu bir matristir. Burada m; A matrisinin sütun sayısının ve n; B matrisinin sütun sayısının ifadesidir.

Sağdan bölme, B/A ; $B/A=(A' \setminus B')$ şeklinde soldan bölmede kullanılan terimlerle tanımlanabilir.

3.4.5 Matrislerin kuvvetini alma işlemi:

Bir A matrisinin kuvvetini almak için $A.^p$ deyimini kullanmak yeterlidir. Bu işlem A matrisi içindeki her bir elemanın teker teker p . kuvvetini alır. Burada A bir kare matris, p ise bir skaler sayıdır.

Bir kare matrisin karesi (A^2) $A*A$ işlemine denktir. Benzer şekilde A^4 deyimini de $A*A*A*A$ işlemine denktir. Buna karşılık $A^{2.5}$ gibi kesirli üst lama işlemi öz değerleri ve öz vektörleri de içine alan daha karmaşık bir işlem gerektirir.

İki matris çarpımının yapılabilmesi için birinci matriste bulunan satır sayısının ikinci matrisin sütun sayısına eşit olması gerektiğine göre, bir matrisinin kuvvetinin alınabilmesi için de satır sayının sütun sayısına eşit olması dolayısıyla kare matris olması gerekir.

3.4.6 Marislerin Üstel Fonksiyonları:

Bir A matrisinin üstel fonksiyonu e^A şeklinde ifade edilir ve çeşitli hesaplamalarda önemli yer tutar. MATLAB'ta bu fonksiyonlar $\exp(A)$ şeklinde bir komutla hesaplanır. Bu komut A 'nın herbir elemanı üzerinde eleman elemana işlemine icra eder. Buna karşılık A 'nın matrisi şeklinde üstel ifadesi ise $\expm(A)$ komutu ile hesaplatılır. MATLAB'taki temel matris fonksiyonları şunlardır:

Tablo 3.2

Poly	Kökleri belirten karakterisitik bir polinom oluşturur
Det	Bir matrisin determinantını hesaplar
Trace	Bir matrisin izini hesaplatır
Kron	Bir Kronecter çarpımını hesaplar

MATLAB aynı zamanda ileri düzey matris fonksiyonlarını da içermektedir. Öreğin matrissel üs ve matrissel logaritma hesaplamaları gibi. $\expm(A)$ ve $\sqrt{m}(A)$ fonksiyonlarında olduğu gibi fonksiyon adlarının

sonunda 'm' harfi bulunan fonksiyonlar ileri matris fonksiyonları olarak hesaplanır.

MATLAB'ta tanımlı bazı yerleşik ileri matris fonksiyonları:

Tablo 3.3

expm	e tabanına göre bir matrisin üssünü hesaplar
logm	Bir matrisin doğal logaritmasını hesaplar
sqrtn	Bir matrisin karekökünü hesaplar

3.4.7 Matrislerin Tersini Alma:

Tanıma göre bir kare matris olan A'nın A^{-1} olup AA^{-1} ve $A^{-1}A$ çarpımların her ikisinin de birim matrise eşit olması gerekir. Yani,

$$A A^{-1} = A^{-1} A = I \text{ dir.}$$

Normalde bir matrisin tersini alma işlemleri oldukça sıkıcı olmakla beraber MATLAB'da bu işlem tek bir komutla (inv) çözümlenir. Aşağıda verilen örnek ile bir A matrisinin tersinin alınması ve bu matrisin tersi ile çarpımın nasıl birim matrise eşit olduğu gösterilmiştir.

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

```
A=
```

```
2 3
```

```
3 1
```

```
B=inv(A)
```

```
B=
```

```
-0.1000 0.3000
```

```
0.4000 -0.2000
```

```
A*B
```

```
ans=
```

```
1.0000 -0.0000
```

```
0.0000 1.0000
```

```
B*A
```

```
ans=
```

```
1.0000 -0.0000
```

0.0 1.0000

BÖLÜM 4

4. MATLAB'DA GRAFİKLER:

MATLAB grafik sistemi, verilerin hazırlanmasında ve görselleştirilmesinde çok değişik ve kendine has özellikleriyle bir kullanıcılara büyük kolaylık sağlamaktadır.

MATLAB 3.5 ve daha altındaki sürümlerinde; x-y grafikleri, kutupsal grafikler, çubuk grafikler, yüzey grafikleri 3-D yüzey network mesh'i oluşturmak mümkündür. Bunun yanında MATLAB 4.0 ve daha yukarı sürümlerinde grafiksel özellikler çok daha fazla artırılmış ve yukarıdaki bahsedilen özelliklere ilaveten x-y-z grafikleri de dahil olmak üzere her türlü 3 boyutlu çizgi ve yüzey grafikleri oluşturmak mümkün hale gelmiştir.

4.1 2 Boyutlu(2D) Grafikler:

MATLAB'ta verilerin 2 boyutlu olarak çizdirilmesi ve bu grafiklerin düzenlenmesi ile ilgili bir takım hazır fonksiyonlar mevcuttur. Aşağıda bu fonksiyonlar açıklanarak örnekler verilmiştir.

4.1.1 2D Grafik Fonksiyonları:

x-y düzleminde oluşturulan pek çok grafikte x ve y eksenlerinin eşit aralıklarla bölüdüğü varsayılır. Ve bu tür grafiklerde lineer grafik adını alır. Bazı istisnai durumlarda ya tek eksen üzerinde ya da her iki eksen (x,y) üzerinde de logaritmik ölçeklendirme kullanılabilir.

x ve y vektörlerinin lineer ve logaritmik fonksiyonları aşağıda verilmiştir (eksenler veri sayısına göre ölçeklendirilirler).

Plot

amaç: 2-D lineer çizim

kullanım:

`plot(Y)`


```

plot(X1,Y1,...)
plot(X1,Y1,ÇizgiTürü,...)
plot(...,'Özellikismi',ÖzellikDeğeri,...)

```

```
h = plot(...)
```

açıklama:

x ve y eksenlerini lineer olarak ölçeklendirir ve verilerin grafiğini çizer.

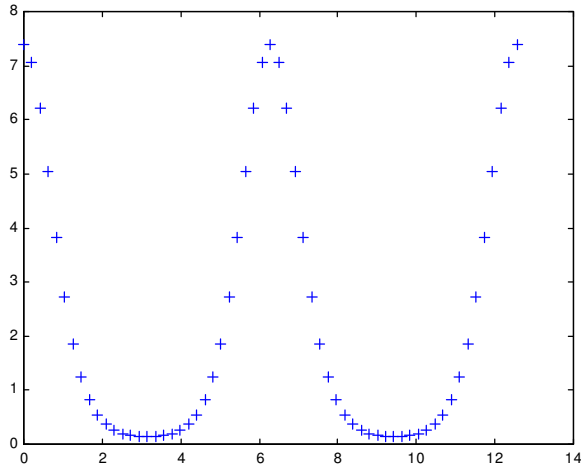
Burada x'ler bağımsız değişken y'ler ise bağımlı değişkeni göstermektedir.

Örnek:

```

X = 0:pi/15:4*pi;
Y = exp(2*cos(X));
plot(X,Y,'b+') herbir veri noktasına mavi artı işareti koyarak çizer

```



Şekil 4.1

Loglog

amaç: Log-log ölçeği çizmek

kullanım:

```

loglog(Y)
loglog(X1,Y1,...)
loglog(X1,Y1,ÇizgiTürü,...)
loglog(...,'Özellikismi', ÖzellikDeğeri,...)
h = loglog(...).plot(Y)

```

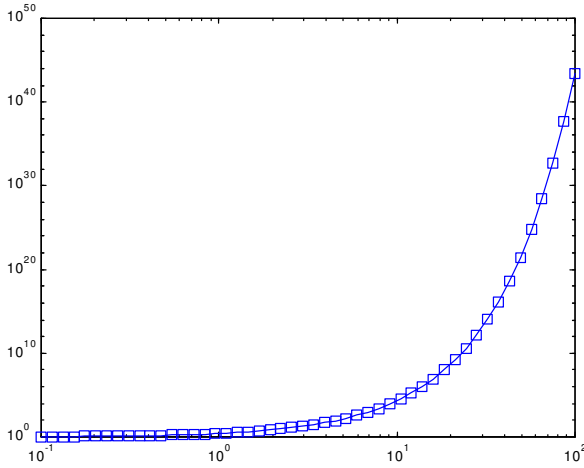
açıklama:

Hem x eksenini hem de y eksenini için logaritmik ölçeklendirme kullanarak x ve y değerlerinin grafiğini oluşturur.

Örnek:

Kare işaretlerle basit bir loglog çizimi oluşturalım.

```
x = logspace(-1,2);
loglog(x,exp(x),'-s')
grid on
```



Şekil 4.2

Semilogx ve semilogy

amaç: Semi-logaritmik grafik çizmek

kullanım:

```
semilogx(Y)
semilogx(X1,Y1,...)
semilogx(X1,Y1, ÇizgiTürü,...)
semilogx(...,'Özellikİsmi',ÖzellikDeğeri,...)
h = semilogx(...)
semilogy(...)
h = semilogy(...)
```

açıklama:

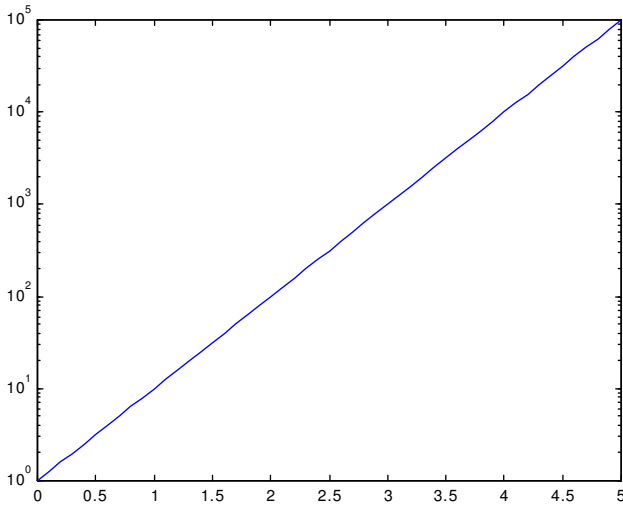
semilogx, x eksenini logaritmik y eksenini lineer olarak ölçeklendirir ve verilerin grafiğini çizer.

semilogy, x eksenini lineer y eksenini doğrusal olarak ölçeklendirir ve verilerin grafiğini çizer.

Örnek:

Örnek bir semilogy plotu:

```
x = 0:.1:10;
semilogy(x,10.^x)
```



Şekil 4.3

MATLAB'da ana grafik fonksiyonlarının yanında çizilen grafikleri üzerine yazı yazdırabileceğimiz etiket komutları da vardır. Belli başlı etiket komutları ve açıklamaları aşağıda verilmiştir.

Title

amaç: Hali hazırdaki grafiğe başlık eklemek

kullanım:

```
title('başlıkyazısı')
title(fismi)
title(...,'Özellikİsmi',ÖzellikDeğeri,...)
```

```
h = title(...)
```

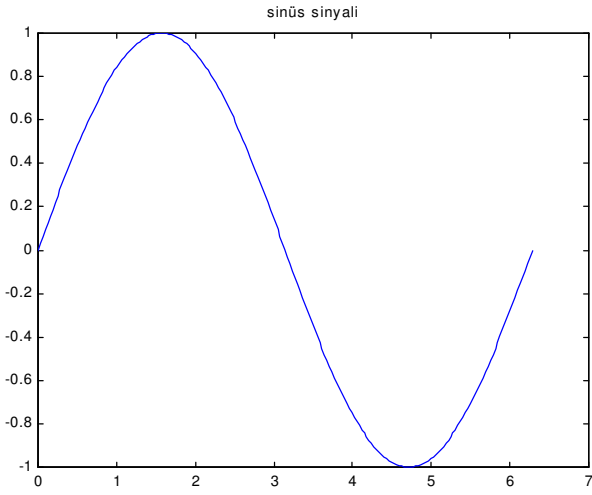
açıklama:

Halihazırdaki grafiğin en üst kısmına merkezleyerek tırnak içindeki kelime ya da kelime grubunu yazdırır.

Örnek:

2π periyotlu bir sinüs sinyali çizdirerek üzerine 'sinüs sinyali' diye yazılması:

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



Şekil 4.4

xlabel ve ylabel

amaç: x ve y eksenleri için etiket tanımlamak

kullanım:

```
xlabel('yazı')
xlabel(fismi)
xlabel(...,'Özellikİsmi',ÖzellikDeğeri,...)
h = xlabel(...)
```



```
ylabel(...)
```

```
h = ylabel(...)
```

açıklama:

xlabel, Halihazırdaki grafiğin x eksenini için etiket yazdırır.

ylabel, Halihazırdaki grafiğin y eksenini için etiket yazdırır.

Text

amaç: x ve y eksenleri için etiket tanımlamak

kullanım:

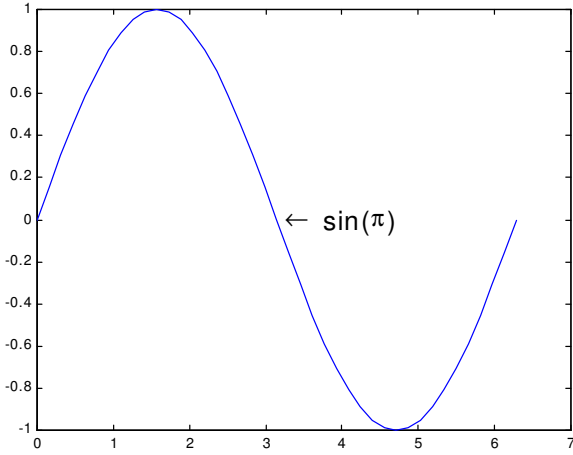
```
text(x,y,'yazı')
text(...'Özellikİsmi',ÖzellikDeğeri...)
h = text(...)
```

açıklama:

Halihazır grafikten alınan eksenleri kullanarak (x,y) koordinatları yolu ile belirlenen grafik eksenine üzerine bir etiket yazdırır.

Örnek:

```
plot(0:pi/20:2*pi,sin(0:pi/20:2*pi))
text(pi,0,' \leftarrow sin(\pi)', 'FontSize',18)
```



Şekil 4.5

Gtext

amaç: Fare yardımıyla grafik üzerine metin yerleştirme

kullanım:

```
gtext('metin')
h = gtext('metin')
```

açıklama:

Bu komut grafik ekran üzerinde fare veya ok tuşları ile gösterilen noktada bir metin veya metinler yazar.

Örnek:

```
gtext('Bu nota dikkat et!')
```

Grid

amaç: Izgara çizgilerini gösterme

kullanım:

```
grid on
grid off
grid
```

açıklama:

Bu komut üzerinde çalışılan grafik için yatay-dikey ölçeklendirme çizgilerini(grid=ızgara) çizdirir.

grid on, üzerinde çalışılan alanda ızgaraları aktif hale getirir.

grid off, aktif olan ızgaraları geri kaldırır.

4.1.2 Bir plotun oluşturulması:

Eğer y bir vektör ise plot(y) fonksiyonu y'nin elemanlarını indislerine göre lineer olarak çizdirir. Argüman olarak iki vektör belirtilirse plot(x,y) fonksiyonu x'e göre y'nin lineer fonksiyonunu çizer.

Bu fonksiyon ile çoklu veri kümelerini çizimin özelliklerini belirterek çizdirmek mümkündür.

Bir plot oluşturulurken izlenecek adımlar ve ilgili örnekler:

Tablo 4.1

Adımlar	İlgili Örnek
1.Verinin hazırlanması	<pre>x = 0:.2:12; y1 = bessell(1,x); y2 = bessell(2,x); y3 = bessell(3,x);</pre>
2.Pencereyi ve çizdirilecek pozisyonunu seçilmesi	<pre>figure(1) subplot(2,2,1)</pre>
3.Çizim fonksiyonunu girilmesi	<pre>h = plot(x,y1,x,y2,x,y3);</pre>
4.Çizgi ve işaret özelliklerinin seçilmesi	<pre>set(h,'Çizgigenişliği',2,'Çizstili',{'-',':','-.'}) set(h,'Renk',{'r','g','b'})</pre>
5.Eksen sınırları, tick izleri ve ızgara çizgilerinin belirlenmesi	<pre>axis([0 12 -0.5 1]) grid on</pre>
6.Eksen etiketi, açıklayıcı bilgi ve text ile grafiğe notların düşülmesi	<pre>xlabel('Zaman') ylabel('Yükselme') legend(h,'Birinci','İkinci','Üçüncü') title('Bessel Fonksiyonlar') [y,ix] = min(y1); text(x(ix),y,'First Min \rightarrow',... 'HorizontalAlignment','right')</pre>
7.Grafik çıktısı al	<pre>print -dps2</pre>

4.1.3 Grafik Çiziminde Çizgi Stilleri, İşaretler ve Renkler:

plot(x,y) komutu x ve y vektörleri ile temsil edilen noktaları çizgi dilimleri ile birleştirerek bir çizgi grafiği oluşturur. Kesikli, noktalı, noktalı - kesikli gibi diğer çizgi türlerini de seçmek mümkündür. Diğer taraftan bir çizgi grafiği yerine bir nokta grafiği de çizdirmek mümkündür.

Aşağıdaki tabloda çizgi ve işaret tiplerinin bir listesi verilmiştir.

Tablo 4.2

Sembol	Renk(RGB)	Çizgi stili	Sembol	Nokta stili	
Y	sarı(110)	.	nokta	-	Çizgi
M	magenta(101)	O	yuvarlak	:	Noktalı
C	ciyan(011)	X	çarpı işareti	-.	çizgili ve noktalı
R	kırmızı(100)	+	artı işareti	--	kesik çizgili
G	yeşil(010)	*	yıldız		

B	mavi(001)	S	karekök		
W	beyaz(111)	D	baklava		
K	siyah(000)	V	üçgen(aşağı)		
		^	üçgen(yukarı)		
		<	üçgen(sola)		
		>	üçgen(sağa)		
		P	pentagram		
		H	hexagram		

Örnek:

plot(x,y,r*) fonksiyonu girildiğinde çizilen grafik kırmızı renkte(r) ve yıldız (*) işaretleriyle çizilmiş olur.

4.1.4 Matrislerin Çizdirilmesi:

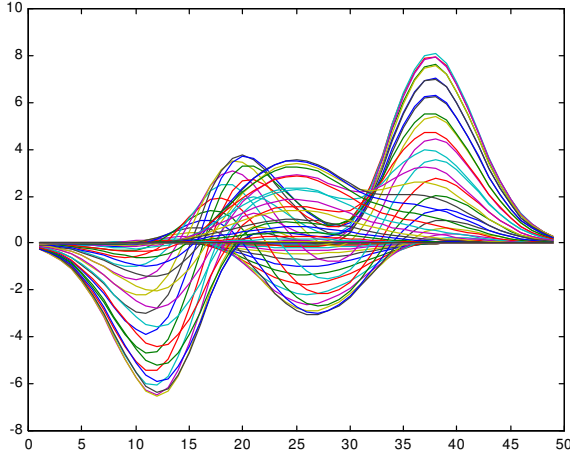
plot fonksiyonu argüman olarak tek bir matris alırsa

`plot(y)`

y matrisinin her birisi ayrı ayrı çizdirilir. X-ekseni ise, 1:m'e kadar olan (m:satır sayısı) indis vektörü ile etiketlenir. Örneğin

`z =peaks`

iki değişkeli fonksiyonu değerlendirerek 49x49 luk matristen oluşan grafiği çizer.



Şekil 4.6

4.2 3 Boyutlu(3D) Grafikler:

MATLAB'ta verilerin 3 boyutlu olarak çizdirilmesi ve bu grafiklerin düzenlenmesi ile ilgili bir takım hazır fonksiyonlar mevcuttur. Bunlar, yüzeylerin ve kafes çerçevesi şeklindeki ağ grafiklerin çizdirilmesinde kullanılmaktadır. Aşağıda bazı temel 3D grafik fonksiyonlar açıklanarak örnekler verilmiştir(diğer grafik fonksiyonları tezin sonundaki ekte verilmiştir).

4.2.1 3D Grafik Fonksiyonları:

plot3

amaç: 3-D lineer çizim

kullanım:

```
plot3(X1,Y1,Z1,...)
plot3(X1,Y1,Z1, Çizgitürü,...)
plot3(...,'Özellikismi',ÖzellikDeğeri,...)
h = plot3(...)
```

açıklama:

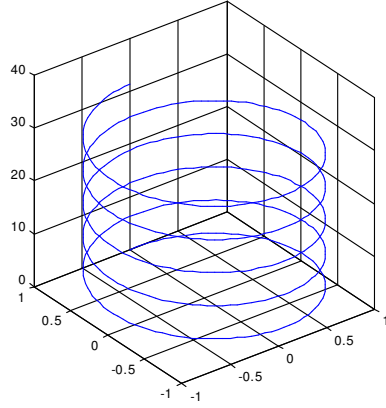
3 boyulu uzayda noktaların ve çizgilerin grafiğini oluşturur. plot(x,y,z) komutu 3 boyutlu uzayda koordinatları x,y ve z'nin elemanları olan

noktalardan geçen tek bir çizgi grafiği oluşturur. Burada x,y ve z'nin aynı boyutta vektörler olması gerekir. çizer.

Örnek:

Üç boyutlu helazonun çizilmesi:

```
t = 0:pi/50:10*pi;
plot3(sin(t),cos(t),t)
grid on; axis square
```



Şekil 4.7

Mesh,meshc,meshz

amaç: Ağ çizimlerini oluşturmak

kullanım:

```
mesh(X,Y,Z)
```

```
mesh(Z)
```

```
mesh(...,C)
```

```
meshc(...)
```

```
meshz(...)
```

```
h = mesh(...)
```

```
h = meshc(...)
```

```
h = meshz(...)
```

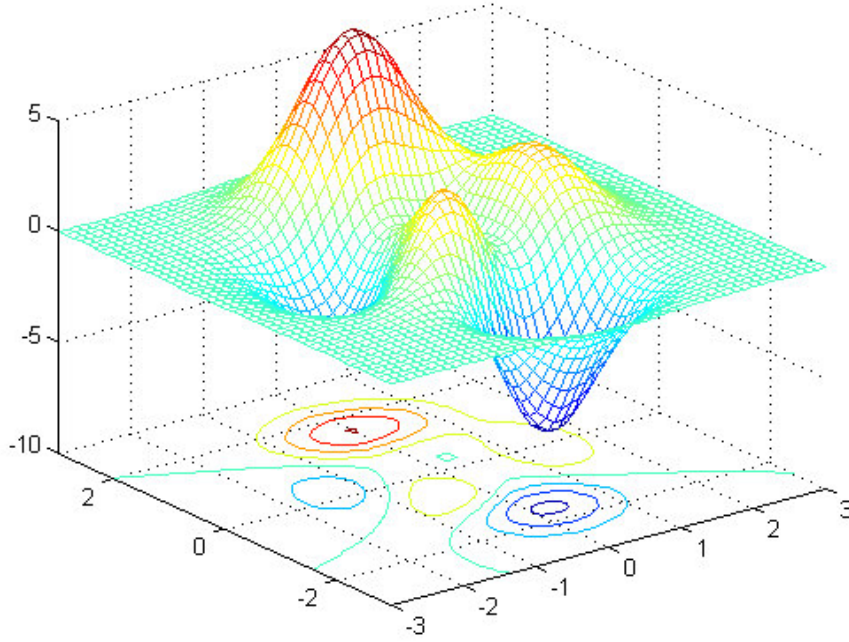
açıklama:

mesh , meshc ve meshz , matris elemanlarının 3 boyulu olarak bşr perspektifinin çizdirirler.

Örnek1:

Birleşme ağını üret ve tepe noktaların kenar çiziminin yapılması.

```
[x,y] = meshgrid(-3:.125:3);  
Z = peaks(X,Y);  
meshc(X,Y,Z);  
axis([-3 3 -3 3 -10 5])
```

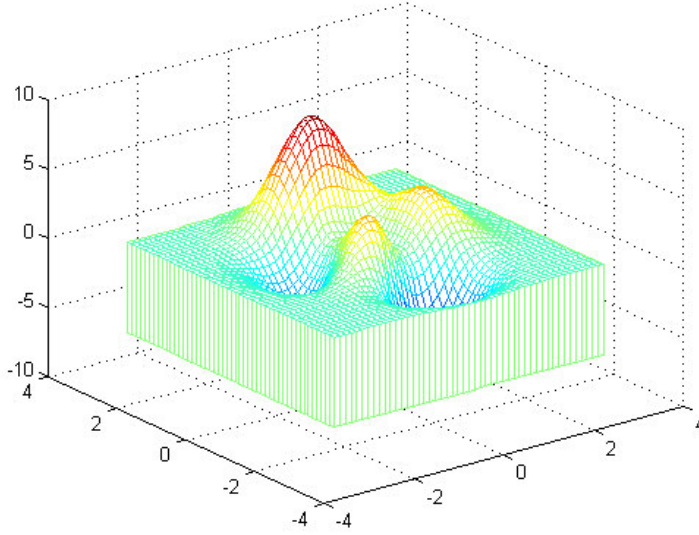


Şekil 4.8

Örnek2:

Tepe noktalar fonksiyonu için perde çiziminin üretilmesi .

```
[X,Y] = meshgrid(-3:.125:3);  
Z = peaks(X,Y);  
meshz(X,Y,Z)
```



Şekil 4.9

Pcolor

amaç: Renkleri matris elemanlarıyla hesaplanan bir dörtgenel hücre dizisinin çizdirilmek.

kullanım:

```
pcolor(C)
pcolor(X,Y,C)
h = pcolor(...)
```

açıklama:

pseudocolor çizimi , 'C' tarafından belirlenmiş renkler ile hücrelerin dikdörtgen biçiminde sıralanmasıdır .

Surface

amaç: Yüzey nesnesi oluşturmak

kullanım:

```
surface(Z)
surface(Z,C)
surface(X,Y,Z)
```

```

surface(X,Y,Z,C)
surface(...'PropertyName',PropertyValue,...)
h = surface(...)

```

açıklama:

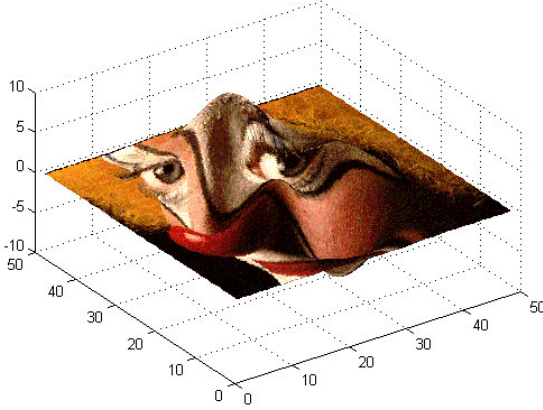
surface, yüzey nesnesi oluşturmak için düşük seviyeli bir grafik fonksiyonudur .

Örnek:

```

load clown
surface(peaks,flipud(X),...
        'FaceColor','texturemap',...
        'EdgeColor','none',...
        'CDataMapping','direct')
colormap(map)
view(-35,45)

```



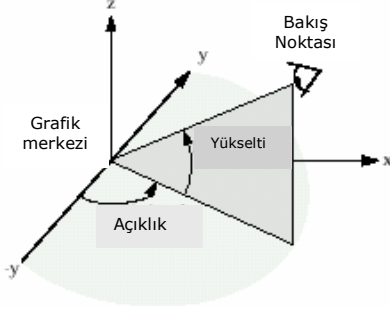
Şekil 4.10

4.2.2 3-Boyutlu Bir Grafiğe Bakış Açısı:

MATLAB, grafiğin yönünü kontrol etmenizde kolaylık sağlar. MATLAB'ın bir grafik için görüş açısını belirtebilir, yönü ve görünüşün izdüşümünü şekil penceresinde görebilirsiniz. Özellikleri görerek bunları bir takım grafik fonksiyonları yoluyla kontrol edebilirsiniz.

view fonksiyonu, küresel koordinatlarda eksenlerin merkezine göre (orjine) açılık ve yükselti açıları belirtilerek bakış noktasını ayarlar.

- Açıklık, x-y düzleminde saat yönünde pozitif değerli kutupsal açıdır.
 - Yükselti ise bakış noktasının x-y düzleminin üzerinde (pozitif) veya altında (negatif) olmasını belirleyen derece cinsinden bir açıdır.
- Aşağıdaki çizim, koordinat sistemini gösterir. Oklar, pozitif yönleri ifade etmektedir.



Şekil 4.11

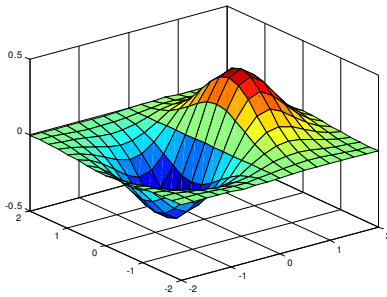
MATLAB , 2-D veya 3-D çizimlerini belirlenmiş görüş açısını otomatik olarak seçer. Buna göre

- 2-D çizimler için, açıklık=0° ve yükselti=90°
- 3-d çizimler için, açıklık=-37.5° ve yükselti=30°

Örnek:Aşağıda peaks matrisinin çizdirilmesinden elde edilen grafiğin dört farklı görünümü yer almaktadır.

1. Açıklık=-37.5° ve yükselti=30° için grafiğin çizimi:

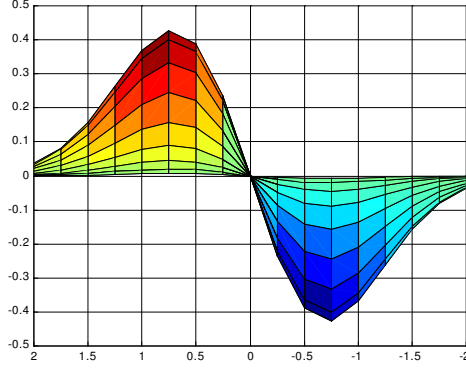
```
[X,Y] = meshgrid([-2:.25:2]);
Z = X.*exp(-X.^2 -Y.^2);
surf(X,Y,Z)
```



Şekil 4.12

2. Açıklık= 180° , yükselti= 0° ve $z=0$ için grafiğin çizimi:

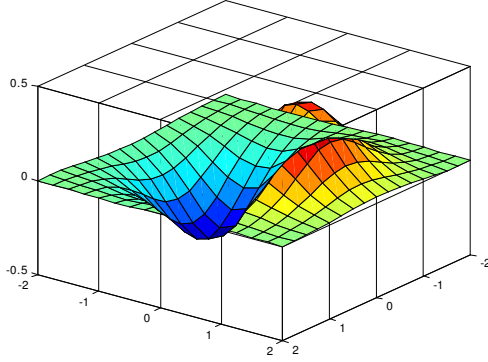
`view[180,0]`



Şekil 4.13

3. Açıklık= -37.5° , yükselti= 30° için grafiğin çizimi:

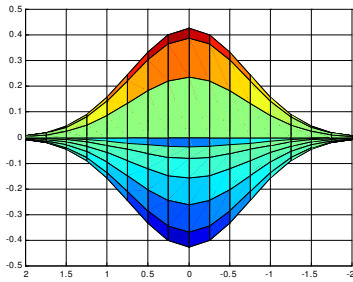
`view([-37.5 -30])`



Şekil 4.14

4. Açıklık= -90° , yükselti= 0° için grafiğin çizimi:

`view([-37.5 -30])`



Şekil 4.15

BÖLÜM 5

5.SIMULINK VE SIMULASYON:

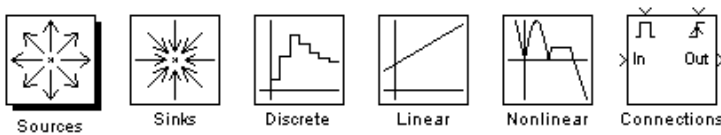
5.1 Simulink'e giriş:

Simulink, modelcilik, simüle etme ve çözümleyiş dinamik sistemler için yazılım paketidir. Sürekli zamanda modellenilen doğrusal ve doğrusal olmayan sistemleri destekler. Sistemler, aynı zamanda çok dereceli olabilirler, farklı dereceli denilen veya güncelleştirilen farklı parçalara sahip olabilirler .

Modelcilik için, simulink, blok çizimleri, kullanma tıkla ve taşı fare operasyonları ile modelleri inşa etmek için grafiksel kullanıcı arabirimi sağlar.

Simulink modellerinin benzetimi diferansiyel denklem takımlarının sayısal integrasyonu ile ilgilidir. Simulink içinde bu tür denklemlerin simulasyonunda kullanılan çeşitli integrasyon yordamları mevcuttur. Aşağıda simulink blokları verilmiştir. (Kullanımları ve özellikleri için Bkz. MATLAB/HELP)

5.2 Simulink Blokları:



Şekil 5.1

- 1.Sources blocks(kaynaklar blokları)
- 2.Sinks blocks
- 3.Discrete blocks
- 4.Linear blocks
- 5.Nonlinear blocks
- 6.Connections blocks

5.3 Simulink Modellerinin Oluşturulması ve Çalıştırılması:

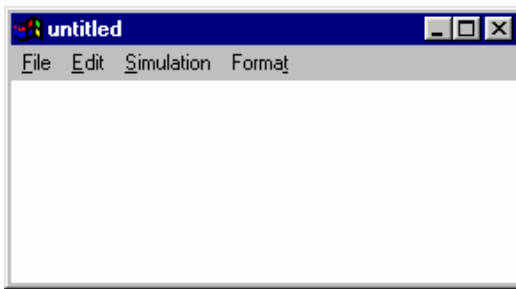
Simulink, bir dinamik sistemin modellenmesi için blok şemalar sunar. Bir sistemin modellenmesinde kullanılan blok elemanları doğrudan simulink blok kütüphanelerinden seçilir. Bunun için MATLAB çalışma ortamında

```
>>simulink
```

komutu girmek yeterlidir. Bu durumda ilk karşımıza çıkan çeşitli blok kütüphanelerini temsil eden kutu şeklindeki şemalardır. (Şekil 5.1) Daha sonra simulink File'den New seçilerek yeni ve boş bir sitem penceresi açılır(Şekil 5.2). Bu pencere simulink çalışma ve ve dinamik model oluşturma ortamıdır ve aynı zamanda isimlendirilmemiş (untitled) bir dosyadır. Modelimiz için gerekli bloklar, blok kütüphanelerinden sürükleyip bırak yada kopyalama ile taşınır.

Açıklama: yakardaki şeklin ismini gireceksin

Açıklama: ismini yaz



Şekil 5.2

Bir blok kütüphanesinde ne türden blok elemanları olduğunu anlamak için fare ile çift tıklamak sureti ile bu kütüphaneyi açmak yeterlidir. Bu

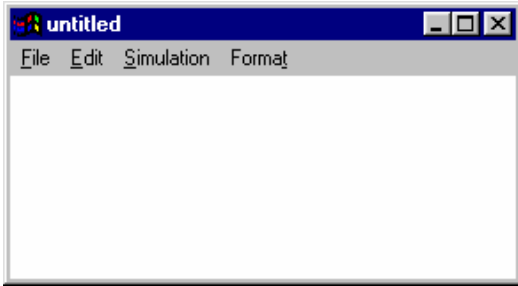
elamanlardan modelimiz için gerekli olanlar fare ile sürüklenerek çalışma ortamına taşınır.

Çalışma ortamına yapılmak istenen model için gerekli elemanlar taşındıktan sonra bu elemanlar uygun biçimde bağlantı yerlerinden birbirlerine bağlanır. İşaret verici kaynak (source) bloklar yalnızca çıkış bağlantı ucuna, işaret gösterici bloklar yalnızca giriş bağlantı ucuna ve diğer bloklar ise hem giriş hem de çıkış bağlantı ucuna sahiptir.

Arzu edilen biçimde model oluşturulduktan sonra .m uzantılı bir isim verilerek (örneğin `modelim.m`) uygun bir dizin altında saklanabilir. Model oluşturma penceresindeki menüden `simulation/start` komutu seçilmek suretiyle modelin simülasyonu çalıştırılır. Scope gibi işaret alıcı blokları kullanmak sureti ile sonuçlar anında alınabilir. Ayrıca çözüm sonuçları bir dosyada saklanarak MATLAB ortamında grafikleri elde edilebilir.

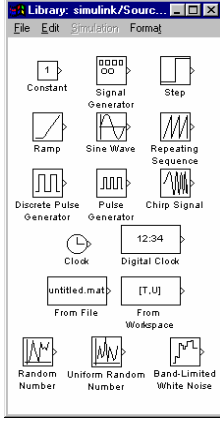
5.4 Adım adım örnek bir modelin oluşturulması ve çalıştırılması:

1.Adım: Yeni bir çalışma ortamı yukarıda anlatıldığı gibi oluşturulur.



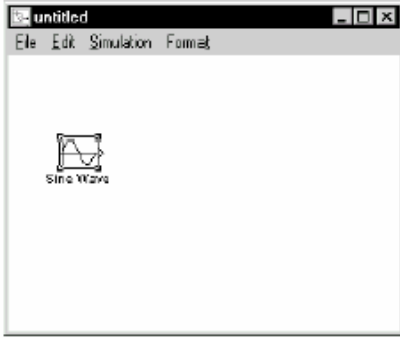
Şekil 5.3

2.Adım: Modelimizde kullanacağımız elemanlar ilgili bloklardan bulunur.



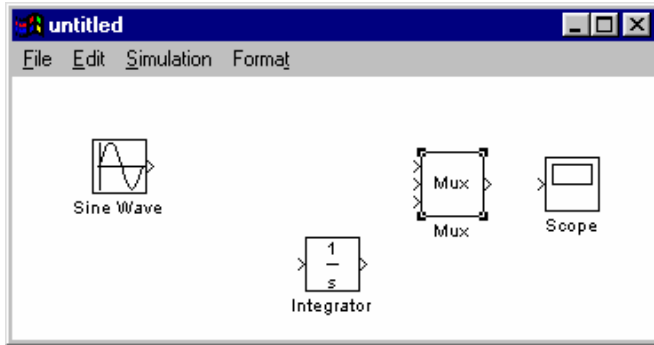
Şekil 5.3

3.Adım: Eleman, bulunduğu pencereden çalışma ortamına fare ile sürüklenerek yerleştirilir.



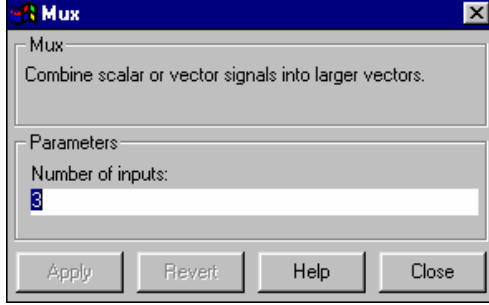
Şekil 5.4

4.Adım: Ve aynı metotla modelin diğer elemanları da çalışma ortamına taşınır.



Şekil 5.5

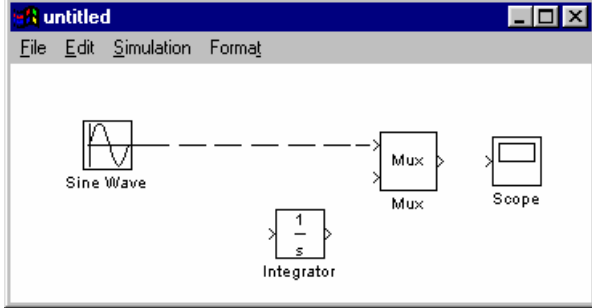
5.Adım:Modelde bulunan multiplexirin giriş uçlarından sadece 2'sini kullanacağımızdan üç girişi ikiye indirebilmek için Mux. üzerine çift



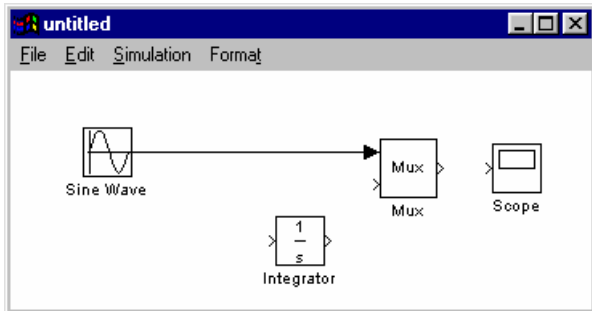
Şekil 5.6

tıklayarak parametere penceresine erişmeliyiz. Ve bu penceredeki 3 değerini 2 yapmalıyız.

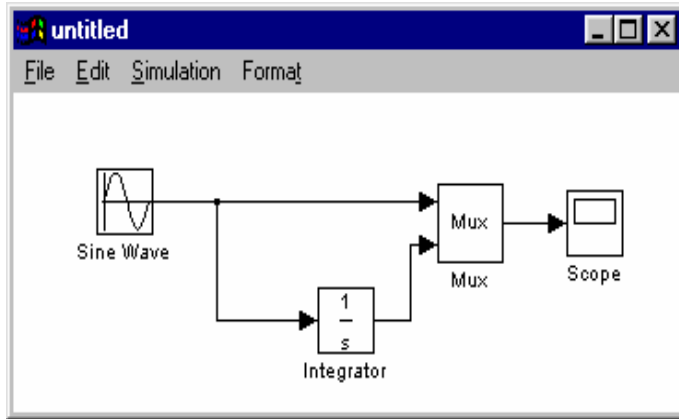
6.Adım:Artık bağlantı işlemine geçerek uygun şekilde aşağıda gösterildiği gibi adım adım bağlantılar yapılır.



Şekil 5.7

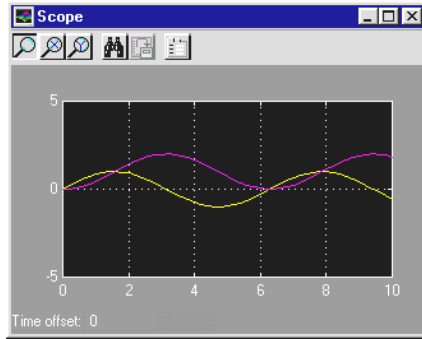


Şekil 5.8



Şekil 5.9

7.Adım:Ve son işlem modeli çalıştırmaktır. Bunun için penceredeki menüde bulunan simülasyon/start vermeliyiz. Eğer scope'nin üzerine çift tıklanırsa modelin sonucunu görebilmek için bir pencere ile karşılaşırız.



Şekil 5.10