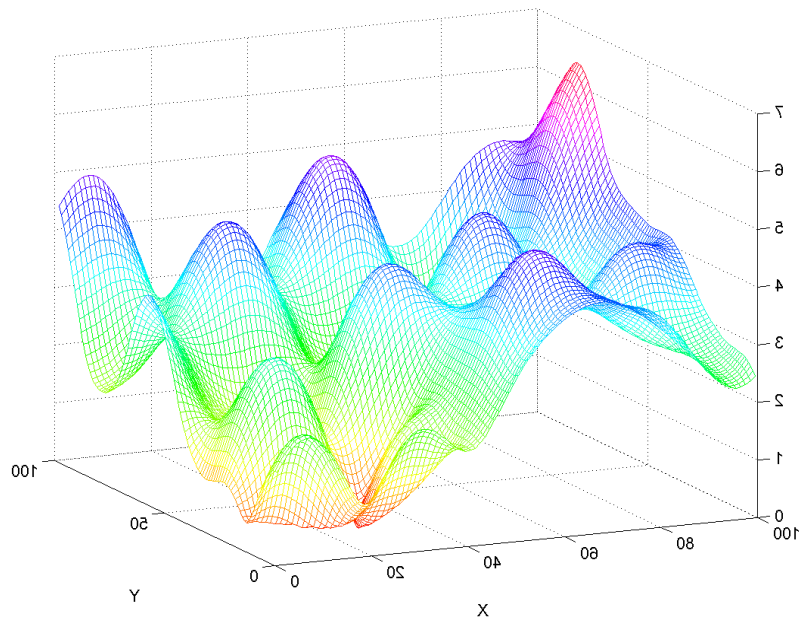
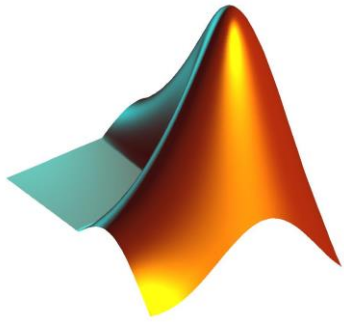


Sayısal Analiz

Matlab Ders Notu

01
000
11101
111
0010101



1.1 MATLAB



MATLAB; (**MA**TriX **LAB**oratory); 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 hazırlanmış olmakla beraber son sürümleri (1999 yılı itibariyle 5.3 dür) C dilinde hazırlanmıştır.

MATLAB mühendislik alanında (hesaplamalarında); 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 CACSD (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.

1.2 KULLANIM AMACI VE YERİ

MATLAB tüm mühendislik alanında, sayısal hesaplamalar, veri çözümlenmesi ve grafik işlemlerinde kolaylıkla kullanılabilen bir programlama dilidir. FORTRAN ve C dili gibi yüksek seviyeden programlama dilleri ile yapılabilen hesaplamaların pekçoğunu MATLAB ile yapmak mümkündür. Buna karşılık MATLAB’ın fonksiyon kütüphanesi sayesinde FORTRAN ve diğer programlama dillerine göre MATLAB’ta daha az sayıda komutla çözüm üretmek mümkündür

Gerçekte MATLAB, M-dosyaları (M-files) olarak bilinen pek çok sayıda fonksiyon dosyalarından (altprogramlardan) ibarettir. M-dosyaları ASCII formatında olup okunabilirliği olan, MATLAB programlama dili kodlarından oluşmuştur. Bu yüzden M-dosyaları kütüphanesi kullanıcı tarafından müdahale edilebilirdir. Ancak bu kodlara zorunlu olunmadıkça müdahale edilmemesi, değiştirilmesi gerekir.

MATLAB’ın kullanım yerleri;

- Denklem sistemlerinin çözümü, doğrusal ve doğrusal olmayan diferansiyel denklemlerinin çözümü, integral hesabı gibi sayısal hesaplamalar,
- Veri çözümleme işlemleri,
- İstatistiksel hesaplamalar ve çözümlemeler,
- Grafik çizimi ve çözümlemeler,
- Bilgisayar destekli denetim sistemi tasarımı.

1.3 KURULMASI ve ÇALIŞMA ORTAMI

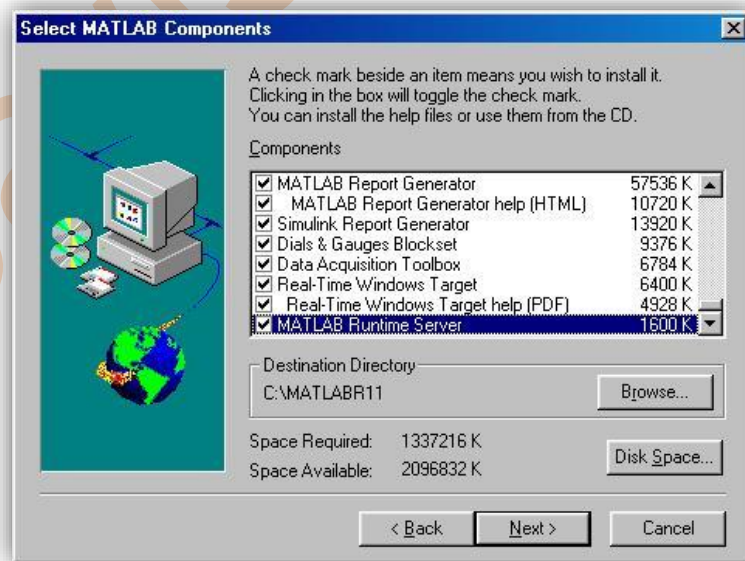
MATLAB’ın değişik bilgisayar sistemlerinde çalışan çok çeşitli sürümleri vardır. Burada anlatılacak olan MATLAB’ın 5.3.10183 (R11) sürümüdür. MATLAB’ın bu sürümünü bilgisayara kurmak önceki sürümlerine göre daha kolaydır. Kurma işlemini başlatmak için

MATLAB dizini altındaki “setup.exe” dosyasına çift tıklanır. Setup programı yüklendikten sonra gelecek olan ekranlardaki direktifler yerine getirilip ileri butonuna basılır. Kurulumun ilk aşaması lisans sözleşmesidir. Aşağıdaki görüntü sözleşme kabul edildikten sonra alınmıştır.



Resim 1 : MATLAB kurulum penceresi

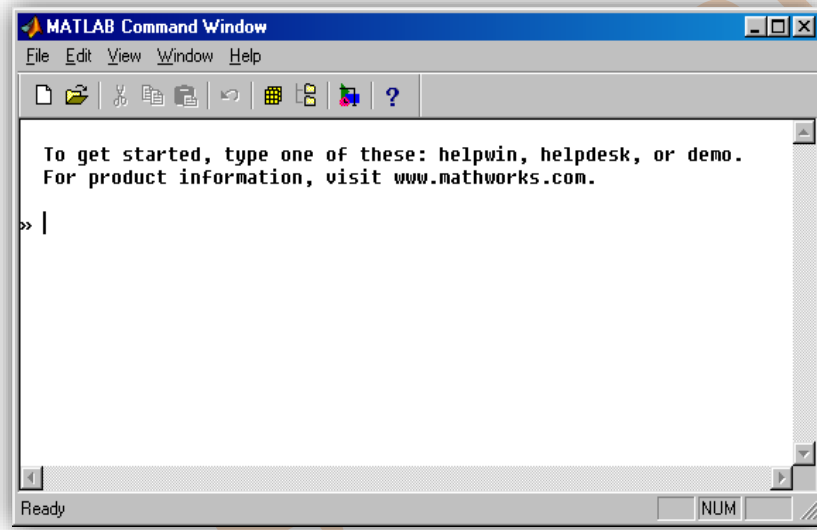
Kurulum için gerekli lisans numarası da girildikten sonra şimdi sıra, seçilmiş MATLAB bileşenlerini yüklemeye geldi. Bu ekranda sizin için gerekli MATLAB araçlarını yükleyebilirsiniz.



Resim 2 : Yüklenecek bileşenlerin seçildiği ekran

MATLAB'ı tüm komponentleriyle birlikte bir PC' ye kurabilmek için yaklaşık 1.5GB disk boşluğu ve en az 16 MB bellek gereklidir. Buradaki 1.5 GB boşluk dosyaların extract işlemi için gereklidir. Kurulum sonrasında MATLAB sabit diskte 300 MB kadar yer kaplayacaktır. Ayrıca kurulum aşamasında Türkçe dil özelliği yüklü olan sistemler için Japanese Help'leri ve MATLAB Tour'u seçmek sorun çıkartmaktadır. Bunların seçilmemesi başarılı bir kurulum için gereklidir.

Yürütme: MATLAB programı bilgisayara yüklendikten sonra, aktif masa üstüne ve programlar mönüsüne kendi kısayollarını yerleştirir. MATLAB bu seçenekler vasıtasıyla belleğe yüklenebilir. MATLAB ortamına girildiğinde, (>>) biçiminde MATLAB iletişi ile karşılaşılır. Bu ileti kullanıcıya, bir komut girmek üzere MATLAB'ın hazır bulunduğunu söyler.



Resim 3 : MATLAB komut penceresi ekranı

Demo : MATLAB'ı ve MATLAB'ın versiyon yeniliklerini öğrenmek için *demo* komutunu kullanabilirsiniz. Listedeki yapacağınız uygun bir seçimle MATLAB'ın çeşitli konularda göstereceği icraatları adım adım ekrandan izleyebilirsiniz.

Çıkış (quit ve exit) : MATLAB ortamından çıkış *quit* veya *exit* komutları ile gerçekleştirilir.

Saklama ve Yükleme (save & load) : Çıkıştan önce MATLAB ortamında elde edilen değişkenlerin içeriklerinin saklanması gerekiyorsa *save* komutu kullanılır. Bu durumda tüm değişkenler *matlab.mat* adlı bir dosya içinde, ikili sayılar kodunda otomatik olarak saklanır. Tekrar MATLAB ortamına girildiğinde, gerekli görüldüğünde saklanan değişkenler *load* komutu ile yüklenebilir.

Çalışma Ortamı : MATLAB programı Windows kaynaklarını çok iyi kullanır. Önceki (4.2 ve öncesi) MATLAB versiyonları DOS tabanlı olarak çalışırdı. Bu yüzden MATLAB'ın konsolunu ve grafik görüntüleme arayüzü aynı anda kullanılamazdı.

MATLAB'ın son sürümlerinde Windows işletim sistemi ile birlikte bu iki pencereyi de aynı anda kullanmak mümkündür. Bu da kullanıcıya problem çözümlerinde büyük hız ve esneklik getirir.

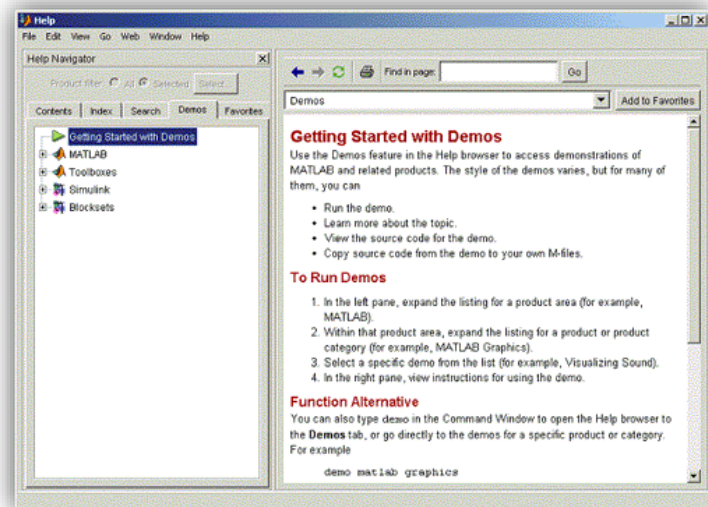
Bir çalışma sırasında komut penceresi silinmek istenirse `clc` komutu kullanılır. Bazı durumlarda çalışma ortamında oluşturulan değişkenler çok fazla yer tutmaya başlar ve bu yüzden küçük bilgisayar sistemlerinde bellek sorunu (OUT OF MEMORY) ortaya çıkabilir. Bu durumdan kurtulmak için `clear` ve `pack` komutları kullanılabilir. `Clear` komutu bellekte yer alan, kullanıcı tarafından tanımlanan tüm değişkenleri ve fonksiyonları siler. Eğer değişkenlerin program içinde tekrar kullanılması gerekiyorsa `pack` komutu daha kullanışlıdır. `pack` komutu bilgileri gerekli en küçük bellek alanına sıkıştırarak bellekte belli bir serbest alan oluşturur. Böylece hem tanımlanan değişkenler silinmemiş ve hem de belli bir yer açılmış olur.

İşleyişin sona erdirilmesi: Özellikle bilgisayarın sonsuz bir döngüye girdiği veya sonu gelmez gibi görünen veri görüntüleme ve yazdırma işlemleri gibi durumlarda komut icrasının kesilmesi gerekebilir. MATLAB ortamında bir komutun icrasını sona erdirmek için `ctrl+c` tuşlarına aynı anda basmak gerekir. Bu durumda `^C` iletisi karşımıza çıkar ki bu da komut icrasının kullanıcı tarafından sona erdirildiğini gösterir.

Büyük ve küçük harf duyarlılığı (`casesen off`, `casesen`): MATLAB büyük ve küçük harf karakterlerini ayırt eden bir dildir. Buna göre; bir değişken `TIME` adı ile atanabilirken, diğer bir değişken `time` adı ile ve diğer başka değer de `Time` adı ile atanabilir. Bu durumda bir karaktere birden fazla değer atamak mümkündür. Ayrıca `casesen off` komutu ile bu durum sona da erdirilebilir. Böylece büyük ve küçük harf karakterlere duyarlılık ortadan kalkmış olur.

Değişkenlerin ve matrislerin durumu (`who`, `whos`, `what`, `size`): 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 olmayan sanal kısımları olup olmadığını da görüntüler. `what` komutu kullanıcı diskinin hali hazır dizininde mevcut M-dosyaları listesini verir.

Yardım (`help`): Yardım imkanı MATLAB'ta en önemli kaynaktır. Çalışma ortamında sadece `help` komutu ile, önce yardım alınabilecek dosyaların bulunduğu dizinler ve sonrada yardım alınabilecek konuların listesi görüntülenir. Kullanıcı daha sonra `help` komutu ile birlikte yardım istediği konuyu yan yana yazarak istediği konuda yardım alabilir.



Örnek :

help step

komutu ile step konusunda gerekli yardımı almak mümkündür.

Böylece arzu edilen konu veya komutun ne şekilde ve nerelerde kullanılabileceği hakkında gerekli bilgi sağlanmış olur. Yardım kaynağını iyi bir biçimde kullanmak suretiyle, kullanıcı kullanım klavuzuna gerek kalmaksızın MATLAB işlemlerini kolaylıkla yapma olanağına sahip olur. Ayrıca örneğin helpwin step komutu ile ayrı bir yardım penceresinden bilgi almak mümkündür.

Komut çizgisinde Yardım Alma: Belirli bir fonksiyon hakkında yardım almak için üç temel fonksiyon bulunmaktadır: help,helpwin(yardım penceresi için kısayol) and doc dökümantasyon için kısayol). help ve helpwin fonksiyonları aynı bilgiyi farklı pencerede vermektedir, doc komutu ise HTML sayfası üzerinde daha geniş bilgi sunar.

Yazım (type): type komutu belli bir dosyanın içeriğini görüntüler. M-dosyalarının görüntülenmesinde dosya adından sonra M uzantısının yazılmasına gerek yoktur. Örneğin type polyfit dosyasında yer alan % işareti ile gizlenen açıklama satırlarını görüntüler.

MATLAB'ta DOS: MATLAB eski sürümlerinden kalma bir özelliği bünyesinde bulundurulur. MATLAB komut satırında (>>) “!” (ünlem) karakteri kullanılarak dos komutları işletilebilir. Bu durum da MATLAB dos komut satırını bir nevi simüle eder. Komutların çıktısını MATLAB konsolu üzerinde görmek mümkündür.

Örneğin program çalıştırmak için !nc , !pc gibi.

MATLAB Komut Penceresi Menüleri:

Bir çok programda olduğu gibi MATLAB'da da komut penceresi menüleri büyük kolaylıklar sağlar.

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 seçebileceğiniz 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 içn yüklü bulunan bir yazıcıya döker. Eğer seçilen metin yoksa, tüm MATLAB oturumu boyunca girilen metni yazdırır.

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

Exit MATLAB: MATLAB oturumunu kapatır.

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.

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.

1.4 KULLANILAN BELLİ BAŞLI KOMUT İŞARETLERİ

[]

Köşeli parantezler, vektörleri ve matrisleri biçimlendirmek için kullanılır.

Örneğin [6.9,9.64, sqrt(-1)] elemanları virgüllerle (veya boşluklarla) ayrılmış üç elemanlı bir vektördür. Buna karşılık [11 12 13;21 22 23], iki satır ve üç sütundan ibaret 2x3 lük bir matrisi gösterir.

Burada noktalı virgül ";" matris oluşturulmasında bir satırı sona erdirip bunu izleyen satırın oluşmasını sağlar. Böylece mxn boyutundaki bir matrisi kolayca oluşturmak mümkündür. Vektör ve matrisler [] köşeli parantez içinde gösterilir.

Örneğin bir A matrisinin satır sayısı B matrisinin satır sayısına eşit ve B nin satır sayısı ve A nın sütun sayısı artı B nin sütun sayısı C nin sütun sayısına eşit iys [A B; C] biçiminde yeni bir matris oluşturmakta mümkündür.

A=[]

Biçiminde bir bildirim boş bir matrisi A değişkenine atar.

()

MATLAB içinde normal parantez çeşitli amaçlarla çeşitli yerlerde kullanılır. Olağan yoldan; aritmetik deyimlerin üstünlüğünü belirtmek ve fonksiyon argümanlarını (arguments) kapatmak için kullanılır.

Daha genel anlamda da vektör ve matrislerin indislerini kapatmak için kullanılır.

Eğer X ve V bir vektör ise X(V) de bir vektördür ve

$$[X(V(1)), X(V(2)) , \dots , X(V(N))]$$

şeklinde gösterilir. V'nin elemanları en yakın tamsayıya yuvarlatılarak indisler olarak kullanılır.

Eğer bu tür bir indisin değeri 1 den küçük veya X'in boyutundan küçük olacak olursa hata ortaya çıkar. Bazı örnekler verecek olursak; X(3) X vektörünün 3'ncü elamanını gösterir.

X([1 2 3])

ve benzer şekilde

$$X([sqrt(2), sqrt(3), 4*atan(1)])$$

X vektörünün ilk üç elemanını göstermek için kullanılır.

Eğer X'in N adet elemanı varsa

$$X(N:-1:1)$$

komutu yukarıdaki işlemi ters çevirir.

Benzer şekilde dolaylı indisleme matrislerde de kullanılır.

Örneğin V'nin M elemanı ve W'nun N elemanı mevcut olsun. Bu durumda A(V,M) A'nın elemanlarından oluşmuş MxN boyutlu bir matrisdir. Burada !'nın indisleri V ve W'nun elemanlarıdır.

Örnek olarak;

$$A([1,5],:) = A([5,1]),:)$$

İfadesi A matrisinin 1 ve 5'nci satırları arasında yer değiştirme sağlar.

•

Kesir ayırma işareti. 314/100, 3.14 ve .314e1 aynı anlama gelir.

Nokta işareti aynı zamanda eleman elemana çarpma ve bölme işleminde de kullanılır. Kullanış biçimleri; .*, .^, ./, .\ veya .' şeklindedir.

Örneğin, C=A./B elemanları c(i,j)=a(i,j)/b(i,j) olan bir matrisdir.

...

Bir tek satıra sığmayan ifadelerin bir alttaki satırda devam ettiğini gösterir.

,

Matris indislerini ve fonksiyon argümanlarını ayırmak için kullanılır. Virgül işareti aynı zamanda bir satıra birden fazla bildirim yazılması halinde bildirim komutlarını ayırmak için de kullanılır.

;

Noktalı virgül bir bildirimde elde edilen sonuçların program icrası sonunda ekranda görüntülenmesini önlemek için kullanılır. Ayrıca köşeli parantezler içinde matrislere ait satırları sona erdirmek için de kullanılır.

%

Yüzde işareti açıklama satırları için kullanılır. Başında % işareti bulunan satırlar icra edilmez. Bir çizgi üzerinde bulunan % işareti, çizginin mantıksal bir sonu olduğunu gösterir.

!

'!' işaretini izleyen herhangi bir yazı DOS komutu işlemi görür.

Böylece MATLAB içinde DOS komutlarını çalıştırma imkanı sağlanır.

:

Sütun işareti; indislerde, FOR iterasyonlarında ve muhtemelen her yerde kullanılabilir. Belli başlı kullanım biçimleri aşağıda olduğu gibidir.

J:K, J den başlayarak 1 er 1 er K ya kadar artan bir dizi oluşturur. Ve [J, J+1 ... , K] aynı anlama gelir.

J:K da J>K ise boş bir vektörü sonuçlandırır.

J:I:K J ile başlayan I aralığında artarak K' ya kadar giden bir dizi oluşturur ve

$$[J, J+1, J+2I, \dots, K]$$

ifadesi ile aynı anlama gelir.

J:I:K ifadesinde $I > 0$ ve $J > K$ veya $I < 0$ ve $J < K$ olacak olursa boş bir matris sonuçlanır.

Sütun işareti aynı zamanda vektörlerin ve matrislerin seçilen satırlarını, sütunlarını ve elemanlarını ayırmak için de kullanılır. Bu durum aşağıdaki örneklerle de gösterilmiştir.

A(:) Tek bir sütun olarak A'nın tüm elemanlarıdır. Atanan bir bildirimin sol tarafında A(:) önceki biçimini koruyarak A'yı doldurur.

A(: , J) A'nın J'inci sütunu ile aynıdır.

A(: , J+1) A'nın J'den itibaren birer birer artarak K ya kadar giden elemanlarını gösterir ve A(J), A(J+1), ..., A(K) ile aynıdır.

A(: , J : K) Benzer şekilde A(: , J), A(: , J+1), ..., A(: , K) ile aynıdır.

,

Tırnak işareti matrislerin transpozisini almayı sağlar. X' , X matrisinin karmaşık eşlenik transpozisini ve X.' İse eşlenik olmayan transpozisini sonuçlandırır.

Ayrıca 'ANY TEXT' şeklindeki aktarım işlemlerinde karakterler için elemanları ASCII kodlarında bir vektörü göstermek için kullanılır.

+

Toplama, X+Y iki matrisin toplamını alır. Bu toplamın gerçekleştirilmesi için matrislerin aynı boyutlarda olması gerekir. Yalnızca matrislerden birisi skaler olduğunda toplam gerçekleştirilebilir. Bir skaler herhangi bir şeye eklenir.

-

Çıkarma, X-Y X matrisinden Y matrisini çıkarır. Burada da toplama işlemindeki benzer koşulların gerçekleştirilmesi gerekir.

*

Matrisyel çarpım işlemcisi. X*Y X ve Y matrislerinin matris çarpımını gösterir. Herhangi bir skaler (1x1 elemanlı matris) herşey ile çarpılabilir. Aksi takdirde X matrisinin sütun sayısı Y matrisinin satır sayısına eşit olmalıdır.

*

Eleman elemana çarpım işlemcisi. X.*Y eleman elemana çarpma işlemini gösterir. Birisi skaler olmadıkça, X ve Y matrislerinin eleman sayıları eşit olmalıdır. Yalnız bir skaler her şey ile çarpılabilir.

\

Matrisyel soldan bölme işlemcisi. $A \setminus B$ B matrisinin A matrisine bölümünü gösterir ki bu da hemen hemen $\text{inv}(A) * B$ işlemi ile aynıdır.

Hesaplama biçimleri farklıdır. Eğer A NxN elemanlı bir matris ve B de N elemanlı Sütun vektörü ise $X = A \setminus B$ ifadesi Gaussian eliminasyonu yoluyla hesaplanan $A * X = B$ denklemi için bir çözümdür.

Eğer A matrisi kötü bir şekilde ölçeklendirilmiş veya hemen hemen tekil ise bir uyarı mesajı alınır. $A \setminus \text{eye}(A)$ ifadesi A'nın tersini sonuçlandırır.

Eğer A mxn olacak şekilde m x n elemanlı bir matris ve B de m elemanlı sütun vektörü ise $X = A \setminus B$ en küçük kareler cinsinden altında ve üstünde saptanan sistem denklemlerinin, $A * X = B$ bir çözümdür.

./

Skaler veya eleman elemana soldan bölme işlemcisi. $A ./ B$ deyimi; elemanları $B(i, J) \setminus A(i, J)$ olan bir matrisdir. Yalnız burada A ve B'nin aynı boyutta olması gerekir.

/

Matrisyel sağ bölme işlemcisi. B / A B matrisinin A matrisine bölümünü gösterir ki bu da hemen hemen $B * \text{inv}(A)$ ile aynıdır. Yalnız hesaplama yöntemleri bakımından farklılık gösterir. Diğer bir biçimi ile $B / A = (A' \setminus' B)$ dır.

./

Eleman elemana bölme işlemcisi. $B ./ A$ eleman elemana bölme işlemini gösterir. Bu durumda A ve B matrisinin aynı boyutta olması gerekir. Yalnız birini skaler olması halinde buna gerek yoktur. Çünkü bir skaler herşey ile bölünebilir.

$3 ./ A$ ile $3 ./ A$ aynı işlem olmadığına dikkat etmek gerekir. Çünkü birinci ifadede nokta 3 rakamının kesir işaretini gösterdiğinden matris bölme işlemini yaptırır ve ikinci ifade de ise nokta eleman elemana bölme işlemini işaret eder.

^

Matrisyel kuvvet alma işlemcisi. $Z = X^y$ ifadesi y'nin skaler olması halinde X'in Y'inci kuvvetini aldırır. Eğer y birden büyük bir tam sayı ise kuvvet alma işlemi tekrarlı çarpma yolu ile hesaplanır. y'nin diğer değerleri için hesaplama özdeğerler ve özvektörler yolu ile gerçekleştirir.

$Z = x^Y$ ifadesinde x'in Y'inci kuvvetinin alınmasında eğer Y bir matris ve x bir skaler ise hesaplama işlemi özdeğerler ve özvektörler kullanılarak yapılır. $Z = X^Y$ de X ve Y'nin her ikisinin de matris olması halinde hata oluşur.

^

Eleman elemana kuvvet alma işlemcisi. $Z=X.^Y$ ifadesi eleman elemana kuvvet alma işlemini gösterir. Ancak birinin skaler olması halinde bu koşul aranmaz. Çünkü bir skaler ile işlem görebilir.

<

...den küçük bağıntı işlemcisi. “<=” ...den küçük veya eşittir işlemcisi. $C=A<B$ bildirimi A ve B matrisleri arasında eleman elemana karşılaştırma yapar ve aynı boyutta bir matrisi sonuçlandırır. C matrisinin elemanları; bağıntının gerçek olması halinde bir ve gerçek olmaması halinde de sıfır olarak atanır.

A ve B matrisleri aynı boyutta olmalıdır. Yalnız birinin skaler olması halinde bu şarta gerek yoktur. Çünkü bir skaler ile karşılaştırılabilir. Diğer bağıntı işlemcileri; >, >=, ==, ~= şeklindedir.

>

...den büyük bağıntı işlemcisi. “>=” ...den veya eşittir işlemcisi.

=

Bildirimleri atamak için kullanılır.

==

Mantıksal eşittir bağıntı (relational) işlemcisi.

&

Mantıksal AND (ve): $C=A \& B$ bildirimi A ve B matrislerinin her ikisi de sıfırdan farklı elemanlara sahip olduğunda 1 ve sadece bir tanesi sıfır elemana sahip olduğunda da sıfır elemanlı bir matris sonuçlandırır. A ve B aynı olduğunda da sıfır elemanlı bir matris sonuçlandırır. A ve B nin aynı boyutta matrisler olması gerekir. Yalnız birinin sıfır olması halinde bu koşula gerek yoktur.

|

Mantıksal OR (veya): $C=A | B$ bildirimi A ve B matrisleri sıfırdan farklı elemana sahip olduğunda, 1 ve her ikisinden birisi sıfır olduğunda sıfır elemanlı bir matris sonuçlandırır.

~

Mantıksal tamamlayıcı (complemet) NOT (değil) işlemcisi \sim : Eşit değildir işlemcisi. $B\sim$ bildirimi; A matrisi bir sıfır elemanına sahip olduğunda 1 ve sıfırdan farklı elemanlara sahip olduğunda 1 ve sıfırdan farklı elemanlara sahip olduğunda da sıfır elemanlı bir matrisi sonuçlandırır.

1.5 VERİLERİN GİRİLMESİ VE PROGRAMLARIN ÇALIŞTIRILMASI

MATLAB genelde bir sayısal hesaplama dilidir. MATEMATICA’da olduğu gibi sembolik ifadelerin çözümlenmesinde genellikle kullanılmaz. Yalnız 4.0 ve daha sonraki sürümlerinde The Symbolic Math Toolbox ilavesi ile sembolik işlemlerde de kullanılabilmesi sağlanmıştır.

MATLAB, matris laboratuvarı kelimelerinin kısaltmasından türetilmiş olarak bütün işlemlerini matris esasına dayalı olarak yapar. Burada kastedilen matris en genel anlamda $m \times n$ elemandan ibaret dikdörtgen bir matris olabileceği gibi tek bir elemandan ibaret bir skaler da olabilir. Tek bir sütun veya bir satırdan ibaret özel matrisler ise vektörler olarak bilinir. Bu açıdan MATLAB ortamındaki işlemler ve komutlar matris cinsinden icra edilir. Dolayısıyla da MATLAB ile çalışırken matrislerle ilgili işlemlerin çok iyi bilinmesi gerekir.

Matrislerin girilmesinde ise genellikle aşağıdaki yollardan birisi kullanılır.

- Kesin elemanlar listesi olarak girme
- MATLAB’ın kendi içindeki bildirimler ve fonksiyonlar yolu ile matris oluşturulması,
- M-dosyaları yolu ile matris oluşturma,
- Harici veri dosyalarından yükleme,

MATLAB’daki işlemlerde ve hesaplamalarda kullanılacak verilerin ve parametrelerin girilmesinde köşeli parantezler kullanılır. Buna karşılık MATLAB’da bulunan bir program (M-dosyası) yoluyla çözümü istenen parametre ve veriler normal parantez içinde yazılır.

Tek bir sayıdan ibaret veri ise köşeli parantez kullanmadan da girilebilir.

Örnek 1.1 :

```
dat = ...  
[ 0      0  
  1      10  
  2      20  
  3      25  
  4      30  
  5      35  
  6      40  
  7      45  
  8      50  
  9      55  
 10      60  
 11      65 ];
```

bildirimi ile 2 sütun 12 satırdan ibaret verilerin dat değişkenine saklanması sağlanmış olur. Burada (=) işlemcisinden sonra gelen (...) işlemcisi veri girişinin (yani satırın) devam edeceğini gösterir.

MATLAB ifadeleri ve fonksiyonları yolu ile matris oluşturulmasında MATLAB dilinde boyut ifadelerine ve tip bildirimlerine gerek yoktur. Oluşturulan sayılar dizisi istenen sırada otomatik olarak saklanır. Örneğin,

X=0:0.1:10;

komutu ile 0'dan başlayıp her adımda 0.1 artan ve 10'a kadar olan sayılar "x" vektörü olarak daha sonra kullanılmak üzere saklanır. Diğer taraftan MATLAB'ın özel fonksiyonları yolu ile de matrisleri oluşturmak mümkündür. Örneğin;

```
a = ones( 5 );
```

komutu ile tüm elemanları 1 olan 5*5 lik bir "a" matrisi oluşturulabilir. Bunun gibi zeros fonksiyonu ile tüm elemanları 0 ve rand komutu ile de rastgele (random) sayılar matrisi oluşturmak mümkündür. Burada ones, zeros, vb.

MATLAB içinde bulunan hazır ifadeler veya fonksiyonlardır. Benzer şekilde a=ones(1,5); komutu ile de 1*5 lik 5 elemandan ibaret bir satır matrisi veya vektör oluşturmak mümkündür.

1.6 VERİLERİN M-DOSYALARI BİÇİMİNDE OLUŞTURULMASI

MATLAB ortamına dışarıdan çağırılacak tüm programlar ".m" uzantılı dosyalar (M-file) olarak bilinir. MATLAB'ın kendi hazır program algoritmaları M-dosyaları şeklinde olabileceği gibi MATLAB için yazılan programlar da M-dosyaları şeklinde saklanıp MATLAB içinde çağrılabilir. Benzer şekilde veriler de M-dosyaları şeklinde oluşturup saklanarak daha sonra MATLAB içinde kullanılabilir. Örnek 1'de gösterildiği gibi verilerinizi (MATLAB da kullanılacak veriler) iki sütun ve on satırdan ibaret bir matris biçiminde uygun bir editörde TEXT kodunda girilerek saklanabilir.

Dosya ismini veri.m ile belirlersek daha sonra MATLAB içinde veri komutu ile verileri istediğimiz kadar çağırabiliriz. Bu veriler MATLAB'ta dat değişkeni altında 2 sütun x 10 satırlık bir matris olarak tanımlanmaktadır. Bu matristeki veriler farklı yerlerde kullanmak istersek yeni değişkenler atamamız gerekir. Örneğin 1. sütundaki verileri;

```
X=dat(:,1);
```

Komutu ile "x" ve 2. sütundaki verileri de

```
Y=dat(:,2);
```

Komutu ile "y" ye atayabiliriz. Benzer şekilde,

```
A=dat(2,3);
```

Komutu ile dat matrisinin 2. satır ve 3. sütunundaki bir veriyi "a" ya atamak mümkündür.

1.7 MATLAB DIŞINDA OLUŞTURULAN VERİ DOSYALARININ YÜKLENMESİ

MATLAB ortamından veriler; MAT-dosyaları ve TEXT dosyaları biçiminde iki şekilde çağrılabilir. TEXT-dosyası , yukarıda da bahsedildiği gibi TEXT karakterleri cinsinden saklanan bilgileri içerir ve bellekte fazla zaman kaybı yaratmayan dosyalardır. MAT-dosyaları

MATLAB içinde “save” komutu ile saklanan ve MATLAB’ta hızlı işlem yapmak için tercih edilen dosyalardır. Örneğin

Save dat1 x y;

Komutu ile x ve y matrisleri “dat.mat” isimli dosyada saklanır. Burada “.mat” uzantısı MATLAB tarafından otomatik bir şekilde oluşturulur. Bu matrisler MATLAB içinde

Load dat1;

Komutu ile çağrılabilir. Benzer şekilde MATLAB içinde oluşturulan veriler ASCII karakterleri cinsinden de saklanabilir. Örneğin,

Save dat1.m z /ascii

Komutu ile z matrisi dat1.m dosyası içinde ASCII karakterleri (herhangi bir TEXT editörü altında okunabilir ve yazılabilir karakterler) cinsinde saklanabilir. TEXT dosyaları MATLAB programları dışındaki programlar (C, FORTRAN, BASIC vb. gibi) tarafından kullanılması gereken veriler için gereklidir.

MATLAB içinde tekrar tekrar kullanılan verilerin ise “.MAT” (binary-ikili sayılar) uzantılı saklanması tercih edilmemelidir. (Burada binary saklanması anlatılmak istenen şey; MATLAB’ın bu dosyaları yazarken kendi yazım algoritmasını kullanmasıdır, bu yüzden de çıktı anlamsız, okunamaz olmaktadır)

1.8 BİLDİRİMLER VE DEĞİŞKENLER

MATLAB bir deyim dilidir. Kullanıcı tarafından yazılan deyimler MATLAB sistemi tarafından yorumlanır ve değerlendirilir. MATLAB bildirimleri çoğunlukla aşağıdaki şekildedir.

değişken=deyim

ve basitçe

deyim(expression)

olabilir. Deyimler işlemci işaretlerinden, diğer özel karakterlerden fonksiyonlardan ve değişken isimlerinden ibaret olabilir. Deyimin değerlendirilmesi ve çözümü sonucunda bir matris oluşturulur.

Bu sonuç ekranda görüntülenebilir ve gerekirse daha sonra kullanılmak üzere bir değişkene atanabilir. Eğer değişken adı ve “=” işareti belirtilmezse, değişken, answer (cevap) kelimesinin kısaltması olan “ans” ile birlikte otomatik olarak oluşturulur.

Örneğin MATLAB ortamında

1900/81

deyimi yazılıp “enter” tuşuna basılınca MATLAB

ans=23.4568

sonucunu oluşturur ve ekranda görüntüler. Örneğin,

$$a = \expm(x);$$

bildirimi ile x matrisinin üstel değeri hesaplanarak sonuç “a” değişkenine atanır ve daha sonra kullanılmak üzere saklanmış olur. Burada bildirimin sonuna konan “;” işareti konulmayan bildirimlerin sonuçları hem saklanır hem de o anda ekranda görüntülenir. Özellikle elde edilen sonuçların çok büyük yer tuttuğu durumlarda bunların o anda ekranda görüntülenmemesi daha uygun olur.

Ayrıca program icrası bittikten sonra istenen veriye ait değişken yazdırılarak ekranda görüntülemek de mümkündür. Örneğin yukarıda “x” matrisinin hesaplanan değeri “a” değişkenine atandıktan sonra, MATLAB ortamından ayrılmadıkça ve “a” değişkenine başka bir atama yapılmadığı müddetçe ekranda “a” yazarak sonucu görüntülemek veya bu değişkeni herhangi bir hesaplamada kullanmak mümkündür.

Ayrıca bu değişkenleri “save “ komutu ile saklayarak daha sonraki MATLAB çalışmalarında da kullanmak mümkündür.

MATLAB ortamından ayrılmadıkça kendine bir değer atanan değişken tekrar tekrar kullanılabilir. Yalnız bu durumda aynı değişkene yeni bir değer atanmaması gerekir. Yeni bir değer atanması durumunda değişken en son atanan değeri saklar.

Bildirim, normal olarak satır sonu dönüşü veya “enter” tuşu ile sona erdirilir. Bildirim sonlarına konan “;” işareti ekranda görüntülenmeyi durdurmakla beraber aynı satırda daha fazla bildirim yazılmasına engel değildir. Diğer bir deyişle, aynı satırda birden fazla bildirim yazmak ve çalıştırmak mümkündür. Bir satırda yazılabilecek karakter sayısı genellikle 80 karakterle sınırlıdır. Deyimin çok karmaşık olduğu ve tek satıra sığdırılmadığı durumlarda, satır sonuna doğru ard arda konan üç noktaya (...) bildirimde süreklilik sağlar. Böylece bir tek bildirimin müteakip satırlarda da devam ettiği bildirilmiş olur.

Örneğin,

$$S = 1 - 1/2 + 1/3 - 1/4 + 1/5 - 1/6 + 1/7 \dots \\ - 1/8 + 1/9 - 1/10 + 1/11 - 1/12;$$

bildirimi iki satırdan ibaret dizinin toplamını hesaplar ve sonucu “s” değişkenine atar. Birinci satırın sonunda yer alan “...” işareti işlemin ikinci satırda devam edeceğini söyler. Bu durumda ikinci satır elemanları yazılıp “enter” tuşuna basılmadıkça hesaplama yapılmaz.

Değişken ve fonksiyon isimleri bir harf ve bunu izleyen rakam ve harflerden meydana gelebilir. Bir ismin yalnızca ilk 19 karakteri hatırlanır. Dolayısıyla 19 harften uzun isimlerde son harflerin bir önemi yoktur.

2.1 PROGRAM KODLAMA

MATLAB’ta algoritmaları bilinen programlar hazırlamak ve çalıştırmak çok kolaydır. Ayrıca FORTRAN, BASIC, C ve PASCAL gibi programlama dillerinde hazırlanmış programları MATLAB için uyarlamak da mümkündür. Bu durumda çoğunlukla aynı program için daha az sayıda satır kullanmak yeterlidir. MATLAB’ın hazır M-dosya paketlerini kullanmak suretiyle programlamayı çok kısa tutmak mümkündür.

2.2 M-DOSYALARI

MATLAB'ın kendisi büyük oranda herhangi bir program içinde çağrılıp kullanılabilen altprogramlardan ibaret hazır M-dosyalarından oluşmuştur. MATLAB'ta M-dosyaları okunabilir ve yazılabilir TEXT dosyalarıdır. Bu nedenle MATLAB'a ait M-dosyalarının yanlışlıkla değiştirilmemesi gerekir. Aksi takdirde orijinal görevini yerine getiremez. Bu nedenle de bir programcı kendi hazırladığı M-dosyalarını kendine ait bir alt dizinde saklanması tavsiye olunur.

MATLAB'ta çalıştırılmak üzere programların yazılıp saklandığı M-dosyaları iki biçimde hazırlanabilir. Birincisi, uzun komutlar dizisi biçiminde yazılan düz yazı (script) M-dosyalarıdır. İkincisi ise MATLAB'ın kendi M-dosyalarına benzer biçimde hazırlanan fonksiyon dosyaları herhangi bir ana program içinde bir altprogram gibi kullanılabilirler.

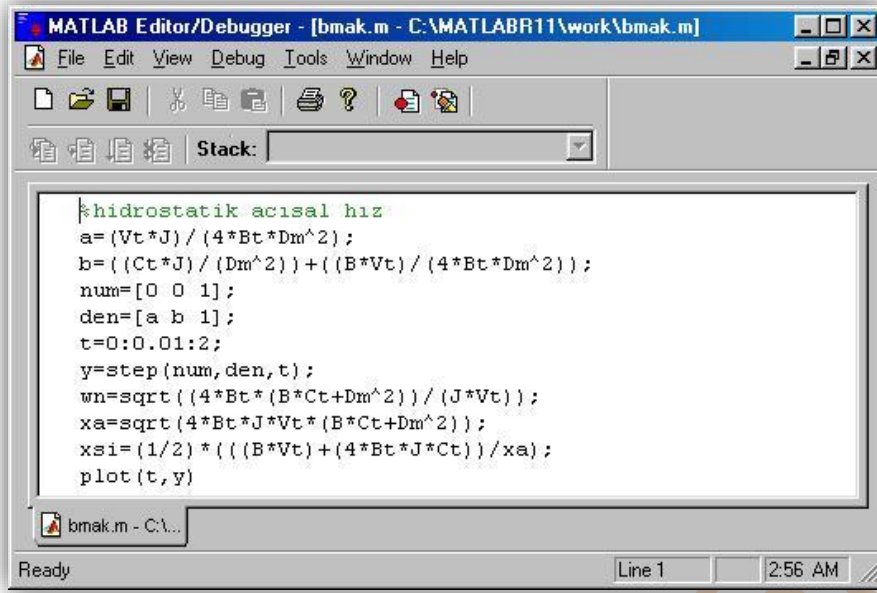
2.3 DÜZ YAZI DOSYALARI

Bir dizi komutlardan ibaret düz yazı dosyaları, MATLAB içinde çağrıldığında dosya içinde bulunan komutlar otomatik olarak çalıştırılır. Böylece her seferinde klavyeden komutları tekrar girmeye gerek kalmaz. Bir düz yazı dosyası içinde yer alan bildirimler MATLAB çalışma ortamında yer alan verileri işletir ve sonuçlandırır.

Düz yazı dosyaları, MATLAB ortamında etkileşimli biçimde çalışılmayacak uzun komutlar dizisi gerektiren analizlerin icrası, problemlerin çözümü veya tasarım yapılmasında kullanışlı olmaktadır. Düz yazı dosyalarının yazımı için MATLAB'ın dahili editörü kullanılabilir. Bu editörü çalıştırmak için MATLAB komut satırında edit komutunu yazmak gerekir

Örneğin aşağıda verilen MATLAB komutları dizisi “bmak.m” adı verilen bir dosyada oluşturulup saklanabilir.

MATLAB ortamında “bmak” yazmak (“.m” uzantısı yazılmaz) sureti ile program çalıştırılabilir. Yalnız bunu için program içinde yer alan; B, Ct, Dm, J, Vt parametre değerlerinin çalıştırılmadan önce atanmış olması gerekir. Bu, ya “bmak” komutu yazılmadan önce klavyeden yukarıda belirtilen parametre değerlerini girmek sureti ile ya da,



Resim 4 : MATLAB editörü ve hata ayıklayıcısı

daha önce sözü edildiği gibi saklanmış bir dosyadan verileri çağırmak sureti ile yapılabilir. Bu şekilde “bmak” çalıştırıldığı zaman MATLAB dosya içinde yazılı komutları icra eder ve sonuçları saklar ve en sonunda plot komutu ile sonuçlar “a” ve “b” değişkenlerine atanır. Daha sonraki 3. ve 4. satırlarda “num” ve “den” değişkenleri yukarıda hesaplanan değerler atanır. 5. satırda ise 0 dan t_f değerine kadar dt artımında bir dizi oluşturur ve “t” değişkenine atanır. 6. satırda MATLAB’a ait bir altprogram (M-file) olan step komutu ile payı “num”, paydası “den”, değişkeni ile verilen bir sistem “t” zaman dizisinde basamak cevabı sonuçları hesaplanır ve bir sayılar dizisi şeklinde “y” değişkenine atanır. Son satırda plot komutu ile sonuçların grafiği elde edilmiş olur. Ayrıca, program içersinde son üç satırda wn, xa ve xsi gibi değerler de hesaplanabilir.

MATLAB içinde desteklenen “demos” daha karmaşık işlemlerin yerine getirilmesinde kullanılacak pek çok örnek içermektedir.

2.4 FONKSİYON DOSYALARI

Fonksiyon dosyaları ilk satırda “function” kelimesi bulunan “.m” uzantılı dosyalardır. MATLAB içinde bulunan tüm M-dosyaları fonksiyon biçiminde olup bunlar hazırlanan herhangi bir program için altprogram görevi görürler. Diğer bir deyişle PASCAL ve benzeri programlama dillerinde kullanılan altprogramlar (subroutines) gibi işlem görürler.

Fonksiyon dosyalarında yer alan tüm değişkenler, düz yazı dosyalarında olduğu gibi MATLAB çalışma ortamında görüntülenemez ve gerektiğinde başka bir bildirimde kullanılamazlar. Fonksiyon dosyası; “function” satırında yer alan giriş verisine göre çıkış argümanı veya argümanlarını sonuçlandırır. Bu argümanlar (function tarafından bulunan sonuçlar) istenirse diğer hesaplamalarda kullanılabilir.

Fonksiyon dosyaları MATLAB’ın kendi dilini kullanmak sureti ile yeni MATLAB fonksiyonlarını oluşturarak MATLAB’ta mevcut dosyaları genişletmesi açısından çok kullanışlıdır.

Fonksiyon dosyasının oluşturulmasını ve çalıştırılmasını aşağıda verilerin basit bir örnekle açıklayabiliriz. Burada MATLAB'ta mevcut bir vektörün ortalamasını hesaplayan “ortalama.m” dosyasına ait bildirimler listesi verilmiştir.

```
function y=ortalama(x)
% ortalama (x) her bir
% sütunda ortalama
% değeri olan bir satır vektörüdür.
[m,n]=size(x)
if m == 1
m=n;
end
y=sum(x)/m;
```

Bu şekilde yazılıp yine “ortalama.m” olarak saklanan (dosya adı ile fonksiyon adı aynı olmalı) yeni fonksiyon dosyasının kullanımı herhangi MATLAB dosyasının kullanımından farklı değildir. Örneğin z, 1’den 99’a kadar tamsayıları gösteren bir vektör olsun,

z=1:99;

Komutu ile z vektörü veya satır matrisi oluşturulmuş olur. Bu vektör içinde yer alan sayıların ortalama değeri ise aşağıda gösterildiği gibi “ortalama” ifadesini kullanmak sureti ile yapılır.

ortalama(z)

veya

a=ortalama(z)

birimci durumda “z” matrisinde yer alan sayıların ortalama değerleri “ortalama.m” fonksiyon dosyası yolu ile hesaplanıp ekranda

ans=50

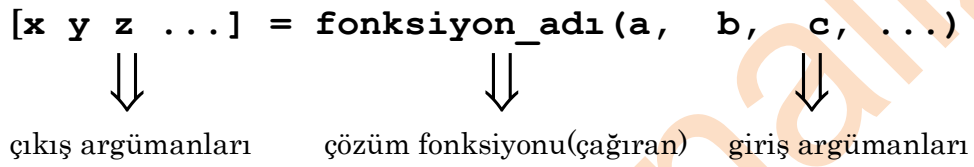
şeklinde görüntülenecektir. İkinci durumda ise “z” ortalama değeri hesaplanıp görüntüledikten sonra aynı zamanda “a” değişkenine atanarak saklanacaktır.

“ortalama.m” fonksiyon dosyasının bazı temel özelliklerini aşağıdaki şekilde sıralayabiliriz.

1. Birinci satır fonksiyon ismini, giriş argümanlarını ve çıkış argümanlarını beyan eder. Bu satır bulunmadığı zaman, dosya düz yazı dosyası gibi işlem görür.
2. % işareti ise ilgili satırların açıklama satırı olduğunu görür. Bu satırlarda yazılanlar MATLAB tarafından icra edilmez.
3. MATLAB ortamında “help ortalama” yazılınca % işaretlerinin bulunduğu satırlardaki açıklamalar ekranda görüntülenir.
4. Fonksiyon dosyası içinde yer alan m, n ve y gibi değişkenler “ortalama” dosyasının çalışması sırasında kendi içinde geçerli olup daha sonra çalışma ortamında

görüntülenmez ve kullanılamazlar. Diğer taraftan bu değişkenler “ortalama” dosyası çalıştırılmadan önce çalışma ortamında mevcut iseler değerleri değişmeden kalırlar.

5. Hesaplatılması gereken değişkenlerin fonksiyon dosyası içinde yer alan değişkenler cinsinden atanması şart değildir. Örneğimizde 1 den 99 a kadar olan tam sayılar, fonksiyon dosyası içinde yer alan x cinsinden değil de z cinsinden atanmıştır. Veriler x cinsinden de veya z dışındaki bir karakter olarak da atanabilirdi. 1 den 99 a kadar sayıları içeren z vektörü, “ortalama” içine geçirilerek veya kopya edilerek onun x adı ile bilinen yerel değişken haline gelir.
6. Benzer şekilde çıkış argümanında fonksiyon dosyasında tanımlanan karakterle aynı olması gerekmez Örneğimizde hesaplanan sonuç “a” şeklinde bir değişkene atanmıştır ki bu “ortalama” fonksiyon” dosyasında “y” ile belirtilen değişkene atanmaktadır.



Şekil 1: Bir fonksiyon dosyasının genel kullanım biçimi

2.5 FONKSİYON FONKSİYONU DOSYALARI

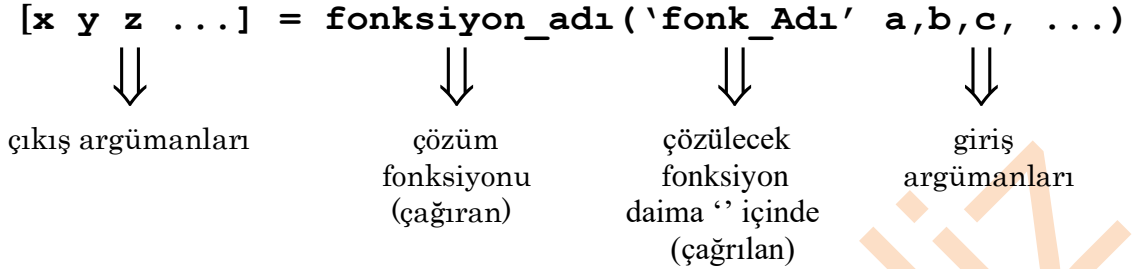
MATLAB’ta normal fonksiyon dosyaları yanında kullanıcı tarafından hazırlanan fonksiyonları çözmek amacı ile kullanılan fonksiyonlar da mevcuttur. Bunlar fonksiyonları çağırın ve çözen fonksiyonlar olduğundan fonksiyon fonksiyonu adını alır. Diğer taraftan doğrusal olmayan sayısal hesaplama yöntemleri olarak da bilinirler.

MATLAB içinde yer alan fonksiyon fonksiyonları ve kullanım amaçları aşağıda verilmiştir.

Ayrıca OPTIM toolbox içinde özel fonksiyon fonksiyonları da mevcuttur.

fmin	Tek değişkenli bir fonksiyonu sadeleştirir.
Fmins	Birkaç değişkenli bir fonksiyonu sadeleştirir.
fplot	Fonksiyonun grafiğini çizer.
fzero	Tek değişkenli bir fonksiyonu sıfır yapan değeri bulur.
ode23	Diferansiyel denklemleri çözer (düşük derece yöntemi)
ode45	Diferansiyel denklemleri çözer (yüksek derece yöntemi)
quad	Sayısal integral çözüm yöntemi (düşük derece yöntemi)
quad8	Sayısal integral çözüm yöntemi (yüksek derece yöntemi)

Diferansiyel denklem çözüm fonksiyonlarının (ode23, ode 45) kullanım şekli ileriki bölümlerde ayrıntılı bir şekilde ele alınacaktır.



Şekil 2 : *Fonksiyon fonksiyonunun kullanım biçimi*

2.6 ÇIKIŞ VERİLERİNİN SAKLANMASI

MATLAB grafik esaslı bir programdır. MATLAB ortamında oluşturulan tüm verileri ve grafikleri saklamak mümkündür. Saklanan veriler tekrar tekrar MATLAB içinde, hatta farklı programlarda da kullanılabilir. Saklanan grafikleri ise tekrar program için geri çağırılmaz, ancak bir ekranda görüntülenir veya yazıcıdan çıkış olarak alınabilir.

Hesaplan verilerin saklanması, daha önce de sözü edildiği gibi save komutu ile yerine getirilmektedir. Çeşitli bilgisayar sistemlerinde farklı biçimlerde kullanılan “save” komutunun kişisel bilgisayarlarda (PC) kullanımı aşağıdaki şekildedir.

Save

MATLAB çalışma ortamında yer alan tüm verileri otomatik olarak oluşturduğu “matlab.mat” isimli dosyada ikili sayılar formatı ile saklar. Bu veriler daha sonra “load” komutu ile çağrılır.

save dosya_adı

Veriler “matlab.mat” yerine “dosya_adı.mat” adında bir dosyada saklanır.

save dosya adı X Y Z

Hali hazırda çalışma ortamında oluşturulmuş bulunan X Y Z değişkenleri “dosya_adı.mat” dosyasında saklanır.

```
save dosya_adı X Y Z /ascii
```

X Y Z değişkenleri “dosya_adı” dosyasında TEXT biçiminde saklanır. Çeşitli yerlerde kullanılabilen bu verilerin tekrar MATLAB ortamında kullanılabilmesi için, veriler MATLAB’ın anlayabileceği matris biçiminde sokulmasına ve dosya adına “.m” uzantısı eklenmesi gerekir. Bu veriler MATLAB’ta “load” komutu ile yüklenemez. M-dosyası şeklinde çağırılması gerekir. Veriler sadece MATLAB içinde çağrılacaksa “.mat” uzantılı saklanması tavsiye edilir.

```
save dosya_adı X Y Z /ascii / double
```

X Y Z değişkenleri “dosya_adı” dosyasında 16-digit ASCII biçiminde saklar.

```
save dosya_adı X Y Z /ascii / tabs
```

X Y Z değişkenleri “dosya_adı” dosyasında TEXT kodunda, tablo biçiminde saklanır.

Yukarıdaki komutların yazılmasında komut ve değişkenler arasında “/” işaretinden önce boşluk bırakılması gerekir. MATLAB’da değişkenler birbirlerinden aralarına virgül konularak ayırt edilebildikleri gibi aralarına boşluk koymak suretiyle de ayırt edilebilirler.

Yalnız giriş argümanlarında boşluktan ziyade virgül konulması gerekir.

3.1 TEMEL MATLAB İŞLEMLERİ

Her ne kadar MATLAB 4.0 dan itibaren MATLAB’ta The Symbolic Math Toolbox ilave edilerek sembolik matematiksel işlem yapma olanakları sağlanmışsa da, MATLAB temelde matrisler ile sayısal işlem yapmak üzere hazırlanmış bir paket programlama dilidir.

MATLAB’ın matrislerde işlem yapması demek skaler işlemleri yapamaması anlamına gelmemelidir. Diğer taraftan bir skaler tek bir satır ve tek bir sütundan ibaret bir matris olarak ele alınabilir. Benzer şekilde bir tek satır veya bir tek sütundan ibaret bir matris satır vektörü veya sütun vektörü veya kısaca vektör adını alır.

MATLAB’ın matrisler ile işlem yapması demek; verileri matrisler biçiminde değerlendirmesi ve çözümleri matris esasına göre yapması ve sonuçları da matris biçiminde düzenlenmesi demektir.

Verilerin tanımlanması ve değişkenlerin atanmasında kullanılan matrisler MATLAB’ta köşeli parantez “[]” içinde yazılır. Örneğin,

```
A=[3.5] veya a=3.5  
B=[1 2 3 4 5]  
C=[1 2 3; 4 5 6; 7 8 9];
```

şeklinde veriler matris biçiminde tanımlanabilir.

Birinci satırda A değişkenine atanan 3.5 değerindeki skaler gereğinde köşeli parantez kullanılmaksızın da atanabilir. Buna karşılık ikinci satırda bir dizin şeklindeki veriler bir satır vektörü biçiminde köşeli parantez içinde atanmak zorundadır.

Üçüncü satırdaki C matrisi örneğinde ise köşeli parantez içinde 3x3 lük matris oluşturulmaktadır. Burada kullanılan noktalı virgül işareti, “;” matris satır elemanlarının sonunu belirler. Yine satır sonunda köşeli parantezin dışında yer alan “;” işareti ise; MATLAB ortamında verilerin görüntülenmesini önler. Bu işaret bulunmadığında üçüncü satırdaki ifade yazılıp enter tuşuna basıldıktan sonra ekranda

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

şeklinde matris görüntülenir.

MATLAB ile ilk defa çalışmaya başlayan ve matrisleri tanımaya ç.alışan kullanıcıların matrisleri görüntülemesi önerilir. Buna karşılık m-dosyaları biçiminde hazırlanan uzun programlarda satır sonlarına “;” işareti konarak ekranda görüntüleme işlemini önlenebilir ve böylece program icrası sırasında ortaya çıkacak zaman kayıpları önlenmiş olur.

3.2 MATRİSLER

Matris elemanları alelade rakamlardan ibaret olabileceği gibi MATLAB deyimlerinden ibaret de olabilir. Örneğin

```
x=[-1.3  sqrt(3)  (1+2+3)*4/5]
bildirimi
x= -1.3000  1.7321  4.8000
sonucunu doğurur.
```

Bireysel matris elemanları parantez “(.)” içindeki indislerle ilişkilendirirler.

Yukarıdaki örneğe devam edecek olursak

```
x(5)=abs(x(1))
bildirimi
x= -1.3000  1.7321  4.800  0.0000  1.3000
sonuçlandırır.
```

Burada x, 5 elemanlı olarak tanımlandığından ilk örnekten atanan 3 elemanı, ilk üç eleman olarak yerine yerleştirmiş 4. elemanı otomatik olarak sıfır aldıktan sonra 5. elemanı x2in 1. elemanının mutlak değerini alarak yerleştirmiştir.

Küçük matrisleri eleman olarak kullanıp daha büyük matrisler oluşturmak mümkündür. Örneğin yukarıda verilen C matrisine aşağıda gösterildiği bir satır ekleyebiliriz.

```
C=[C;  [10  11  12]]
```


Bu durumda

C=

1	2	3
4	5	6
7	8	9
10	11	12

olarak sonuçlanır. “:” işareti kullanmak sureti ile büyük bir matristen daha küçük bir matris oluşturulabilir. Örneğin

```
C=C(1:3, :)
```

Bildirimi ile yukarıda en son oluşturulan C matrisinin ilk 3 satırı ve tüm sütunları alınarak sol taraftaki C ye atanır ve böylece ilk orijinal C matrisi elde edilmiş olunur. Verilerin değerlendirilmesinde sütun işlemcisi “:” çok kullanışlıdır. Örneğin

```
x=data(:,1)  
y=data(:,2)
```

bildirimleri ile data adı altında matris biçiminde tanımlanan verilerin 1. sütun x değişkenine ve 2. sütunu da y değişkenine atanmış olur. Benzer şekilde;

```
x=data(1,:)
```

data matrisindeki verilerin 1. satırını x değişkenine atar.

```
X=data(:,2:3)
```

Data matrisindeki verilerin 2. ve 3. sütunlarını x değişkenine atar.

```
X=data(1:3,:)
```

Data matrisindeki verilerin 1,2 ve 3 üncü satırlarını x değişkenine atar.

Bu şekilde mevcut bir veri tablosundan çok değişik matrisler oluşturmak mümkündür.

MATLAB’ta kullanıcı, verileri kendi matrisleri ile tanımlayabileceği gibi MATLAB’ın kendi özel matrislerinden de faydalanabilir. Bunların belli başlıları; ones, eye, zeros, magic, rand, randn şeklindedir.

Ones(n) :Tüm elemanları 1 olan nxn elemanlı kare matris oluşturur.

Ones(1,n):Tüm elemanları 1 olan n elemanlı bir satır matrisi oluşturur.

Ones(n,1):Tüm elemanları 1 olan n elemanlı bir sütun matrisi oluşturur.

Benzer şekilde eye ile köşegen elemanları 1 olan birim matris, zeros ile tüm elemanları sıfır olan matris, rand, randn ile elemanları rasgele sayılardan oluşan bir matris oluşturulabilir. help olanakları ile daha ayrıntılı bilgiler elde edilebilir.

3.3 SABİTLER

MATLAB programlarında kullanılabilen skaler değerler aşağıda tanımlanmıştır. Bu değişkenlerin içerikleri MATLAB komut satırında yazılıp, enter'a basılarak görüntülenebilir.

`pi` (π)

π değeri `pi` adı altında otomatik olarak saklanmıştır. Programlar içinde kullanılan `pi` kelimesi doğrudan π değerine karşılık gelir.

`i`, `j` ($\sqrt{-1}$)

`i`, `j` harfleri doğrudan $\sqrt{-1}$ değerine ayarlanmıştır.

`Inf` (∞)

Bu kelime MATLAB'ta sonsuz değeri için atanmış bir değişkendir ve sıfıra bölme işlemlerinde ortaya çıkar. Eğer sıfıra bölme işlemi görüntülenmek istenirse bir uyarı mesajı alınır ve sonuç ∞ işareti şeklinde görüntülenir veya basılır.

`NaN`

Bu değer Not-a-Number (rakam değil) anlamına gelir ve sıfır bölü sıfır bölümünde olduğu gibi tanımlanmamış deyimlerde ortaya çıkar.

`eps`

Bu değer fonksiyon, kullanılmakta olan bilgisayar için floatingpoint (virgüllü sayılar) tamlığını içerir. Bu epsilon tamlığı 1.0 ve bunu izleyen enbüyük decimal (onlu sayılar) arasındaki farktır.

`ans`

Bu değişken bir deyim tarafından hesaplanan fakat bir değişken ismi altında saklanmayan değerleri saklamak için kullanılır.

3.4 ELEMAN ELEMANA HESAPLAMA İŞLEMLERİ

Eleman elemana hesaplama işlemi eleman eleman icra edilir.

Örneğin A ve B'nin 5'er elemanlı birer satır vektörü olduğunu varsayalım. Bu değerler ile yeni bir C satır vektörü oluşturmanın bir yolu, aşağıda gösterildiği gibi A ve B deki karşılık gelen değerlerin çarpımlarını almaktır.

$$\begin{aligned} C(1) &= A(1) * B(1); \\ C(2) &= A(2) * B(2); \\ C(3) &= A(3) * B(3); \\ C(4) &= A(4) * B(4); \\ C(5) &= A(5) * B(5); \end{aligned}$$

Bu komutlar esasında skaler komutlardır. Çünkü herbir komut tek bir skaler değeri diğer bir tek skaler değer ile çarparak çarpımı üçüncü bir değer olarak saklamaktadır.

MATLAB’da aynı boyutlu iki matris arasında eleman eleman çarpma işleminin icrası aşağıdaki bildirimde gösterildiği gibi,”*” işlemcisi ile çok daha kısa yoldan yerine getirilir.

$$C = A.*B$$

Toplama ve çıkarma işleminde eleman elemana hesaplama ve matris işlemleri aynıdır. Dolayısıyla bunlarda farklı işlemci kullanmaya gerek yoktur. Buna karşılık eleman eleman işlemlerinde çarpma, bölme ve üst alma, matris işlemlerindeki çarpma, bölme ve üst alma işlemlerinden farklılık gösterir. Burada en önemli fark çarpma, bölme ve üst alma işlemcileri önünde nokta, “.” İşareti gelmesidir.

Eleman elemana hesaplama işlemleri yalnızca aynı boyutlu iki matris arasında uygulanmayıp aynı zamanda skaler ve skaler olmayan değerler arasında da uygulanır. Bununla beraber bir matrisin bir skaler ile çarpımı ve bölümünde işlemciler noktalı ve noktasız olarak kullanılabilir. Buna göre skaler olmayan bir A matrisi için aşağıda verilen bildirim takımları birbirine denktir.

$$B=3*A; \text{ veya } B=3.*A;$$
$$C=A/5; \text{ veya } C=A./5;$$

Sonuç matrisler B ve C her iki durumda da A ile aynı boyutta bir matris olur.

Eleman elemana çarpım: Vektörlerde kullanılan eleman elemana çarpma işlemini göstermek üzere aşağıda verilen iki satır vektörünü ele alalım.

$$A=[2 \ 5 \ 6]; \ B=[2 \ 3 \ 5];$$
$$C=A.*B$$

şeklinde verilen bir komut

$$C=[4 \ 15 \ 30]$$

eleman elemana çarpma sonucunu verecektir.

Eleman elemana bölme: MATLAB ileri veya sağdan bölme, ./ ve geri veya soldan bölme, ./ olmak üzere iki bölme işlemcisi kullanır. Sağdan eleman elemana bölme komutu

$$C=A./B$$

biçiminde olup bu da

$$C=[1 \ 1.667 \ 1.2]$$

A’nın her bir elemanın B tarafından bölümünü sonuçlandırır. Buna karşılık geri ve soldan bölme (ters bölme) işlemi

$$C=A.\B$$

komutu ile gerçekleşir ki bu B’nin A tarafından bölümünü sonuçlandırır.

$$C=[1 \ 0.6 \ 0.833]$$

Eleman elemana üst alma: Bu işlemde eleman elemana üst alınır. Yukarıda tanımlanan A ve B vektörlerini ele alacak olursak;

$$C=A.^2;$$
$$D=A.^B;$$

komutları yolu ile C ve D vektörleri elde edilir.

$$C=[4 \ 25 \ 36];$$
$$D=[4 \ 125 \ 7776];$$

Aynı zamanda bir skaler tabanın vektör üssünü almak mümkündür.

Örneğin

$$C=3.^0.^A;$$

bildirimi

$$C=[9 \ 243 \ 729];$$

Sonucunu doğuracaktır. Bu vektör aynı zamanda aşağıdaki bildirimle de hesaplanabilir.

$$C=3.^A;$$

Burada MATLAB nokta işaretini “.” 3 sabitinin bir parçası sayacak ve ona göre işlem yapacaktır. Bu da matris işlemine karşılık aşağıda geleceğinden kastedilene göre yanlış olacaktır. Buna karşılık aşağıda gösterildiği gibi nokta ile sabit arasında bir boşluk konacak olursa yine doğru sonuç alınır.

$$C=3 \ .^A;$$

Bu örnekler eleman elemana hesaplama işlemi yapılırken çok dikkatli olunması gerektiğini göstermektedir. Yazılan ifadenin doğruluğundan tam olarak emin olunmadığı durumlarda basit örneklerle bazı testler yapılması yerinde olacaktır. Bunu aşağıdaki örneklerle gösterebiliriz.

Bundan önceki örneklerde eleman elemana hesaplama işlemlerinde, vektörler kullanılmıştır. Aşağıdaki örneklerde görüldüğü gibi eleman elemana hesaplama işlemlerinde matrisleri de kullanmak mümkündür.

$$d=[1:5; -1:-1;-5]$$
$$z=\text{ones}(d);$$
$$s=d-z;$$
$$P=d.*s;$$
$$Sq=d.^3;$$

Hesaplama sonuçları aşağıda gösterildiği gibidir.

D=

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ -1 & -2 & -3 & -4 & -5 \end{bmatrix}$$

Z=

1	1	1	1	1
1	1	1	1	1

S=

0	1	2	3	4
-2	-3	-4	-5	-6

Sq=

1	8	27	64	125
-1	-8	-27	-64	-125

4.1 DENKLEM SİSTEMLERİNİN ÇÖZÜMLERİ

Mühendislik problemlerinde doğrusal ve doğrusal olmayan çeşitli denklem ve denklem takımları ile karşılaşılır. Herhangi bir denklem en basitinden bir doğru veya bir eğriyi tanımlar. n. Dereceden tek değişkenli bir polinom biçimindeki bir denklemde, denklemin köklerini bulmak esastır. Birden fazla değişkeni bulunan denklem takımlarında ise denklemleri teşkil eden bilinmeyenlerin çözümü sağlanır.

Doğrusal denklemlerin çözümü doğrusal olmayan denklemlere göre daha kolaydır.

MATLAB'ta doğrusal denklemlerin çözümü matris işlemleri yolu ile çok basit şekilde çözülebilir. Polinom biçiminde doğrusal bir denklemin kökleri roots fonksiyonu yolu ile çözülebilir. Doğrusal denklem takımları ise sağ veya sol matris bölme işlemleri ve inv (matrisin tersini alma) fonksiyonu yardımı ile çözülür.

Doğrusal olmayan denklemlerin çözümünde; MATLAB içinde yer alan fmin, fmins, fplot, fzero fonksiyon fonksiyonları kullanılabilir.

Doğrusal olmayan denklem sistemlerinin çeşitli sınır şartları altında çözümünde kullanılan çok değişik fonksiyonlar mevcuttur. Bunlar içinde en kullanışlısı fsolve fonksiyonudur.

4.2 DOĞRUSAL(Linear) DENKLEM ÇÖZÜMLERİ

Tek değişkenli bir doğrusal sistemin denklemi, n. dereceden bir plinom biçiminde;

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_2 x^2 + a_1 x + a_0$$

tanımlanır. Burada $f(x)=0$ biçiminde denklemin köklerini bulmak için MATLAB içinde yer alan roots fonksiyonu veya fzero fonksiyon fonksiyonu kullanılabilir. Fzero fonksiyon fonksiyonu daha çok doğrusal olmayan denklemlerin çözümünde kullanılır.

Roots fonksiyonunun kullanılışı, aşağıda verildiği gibi çok basittir.

```
roots( [an, an-1, ..., a2, a1, a0])
```


Burada `roots` fonksiyonu için giriş argümanlarını, $a_n, a_{n-1}, \dots, a_2, a_1, a_0$ biçiminde polinomun en yüksek dereceden başlayarak azalan kuvvetlerinin katsayıları olarak tanımlamak yeterlidir. Sonuçta çıkış olarak denklemin kökleri bulunur.

Örnek :Aşağıda verilen 8. dereceden polinomun köklerini bulunuz.

$$f(x) = 3x^8 + 4x^7 - 9x^6 + 13x^5 - x^4 + 1.5x^3 - 10.5x^2 + 15x - 5$$

Çözüm:

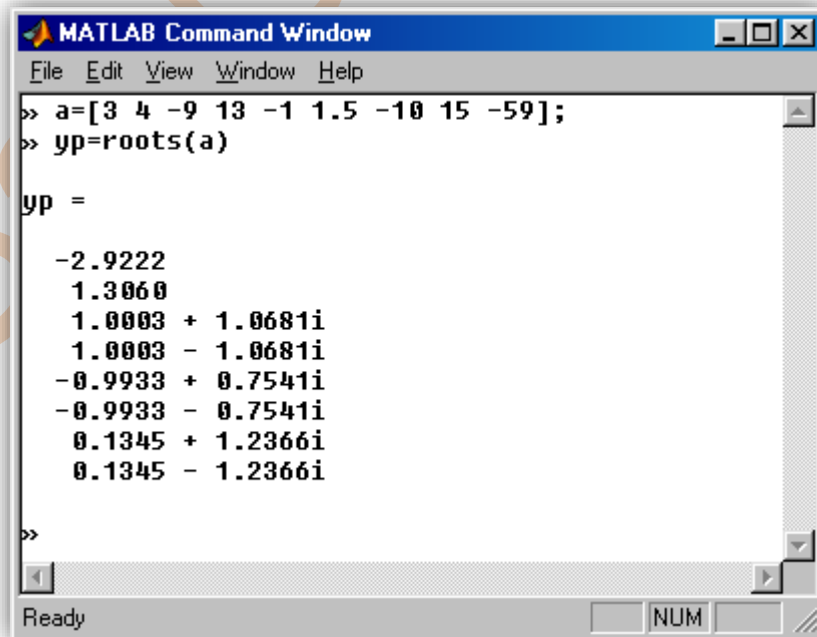
```
>> a=[3 4 -9 13 -1 1.5 -10.5 15 -5];  
>> yp=roots(a)
```

veya

```
yp=roots( [3 4 -9 13 -1 1.5 -10.5 15 -5] )
```

bildirimi ile sonuç şu şekilde elde edilir.

```
X1= -2.9170  
X2= -0.7475 + 0.7096i  
X3= -0.7475 - 0.7096i  
X4= 0.6024 + 1.0396i  
X5= 0.6024 - 1.0396i  
X6= 0.7060 + 0.5552i  
X7= 0.7060 - 0.5552i  
X8= 0.4618
```



Resim 5: Örneğin MATLAB ortamındaki çözümü ve çıktısı

4.3 DOĞRUSAL DENKLEM SİSTEMLERİNİN ÇÖZÜMLERİ

n. dereceden doğrusal denklem takımı

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

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

.....

$$a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n$$

biçiminde verilir. Bu denklem takımının matris biçiminde gösterimi

$$[A][x] = [B]$$

şeklinde tanımlanabilir. Burada;

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

katsayılar matrislerini ve

$$X = [x_1 \ x_2 \ \dots \ x_n]$$

çözümü istenen değişkenler matrisini (vektörünü) gösterir.

Bu tür istenen denklem sistemlerinin çözümü için, içinde yer alan özel fonksiyonlar yoktur. Bu denklemlerin çözümünde matris işlemlerinden yararlanılabilir. $AX=B$ biçiminde verilen denklem takımının çözümünde $A \setminus B$ biçiminde soldan (bölen bölme işaretinin solunda yer almakta) matris bölme işlemi ile yerine getirilir. $XA=B$ biçiminde tanımlanan matris denklemin çözümünde B/A şeklinde sağdan matris bölme işlemi kullanılır. Sağdan ve soldan matris bölme işleminde sayısal Gauss eliminasyon tekniği kullanılır.

Denklem Sistemlerinin Ters Matris İşlemi Yolu ile Çözümü

$AX=B$ biçiminde verilen ve B'nin satır matrisi olarak tanımlandığı matris denkleminin her iki tarafını A-1 ile çarparsak

$$A^{-1} \quad AX = A^{-1}B$$

elde edilir. Burada A-1 A, I olarak tanımlanan birim matrise denktir.

Buna göre

$$IX = A^{-1}B$$

veya

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

elde edilir.

MATLAB ortamında bu çözüm;

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

komutu ile elde edilebilir. Diğer taraftan B'nin sütun matrisi olarak tanımlandığı, $XA=B$ biçiminde ifade edilen denklem takımının çözümü için, her iki taraf A-1 ile çarpılır ve gerekli düzenlemeler yapılırsa

$$X=BA^{-1}$$

elde edilir. MATLAB ortamında

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

bildirimi ile gerekli çözüm elde edilmiş olur.

Örnek : Aşağıda verilen denklem takımının çözümünü elde ediniz.

$$\begin{cases} x_1 + 4x_2 - x_3 + x_4 = 2 \\ 2x_1 + 7x_2 + x_3 - 2x_4 = 16 \\ x_1 + 4x_2 - x_3 + 2x_4 = 1 \\ 3x_1 - 10x_2 - 2x_3 + 5x_4 = -15 \end{cases}$$

Çözüm : Çözüm ilk önce soldan ve sağdan matris bölme işlemlerine göre ele alınacak ve daha sonra da ters matris işlemine göre çözülecektir. $AX=B$ biçiminde matris denklemi verildiğinde çözüm soldan bölme işlemine göre aşağıdaki bildirimlerle yerine getirilebilir.

```
a=[ 1 4 -1 1; 2 7 1 -2; 1 4 -1 2; 3 -10 -2 5 ]
b=[ 2 16 1 -15 ]
>> x=a\b;
```

Denklem takımları $XA=B$ biçiminde matris denklemi ile tanımlandığında çözüm sağdan bölme işlemi ile aşağıdaki şekilde sağlanır. Burada A ve B matrisleri bir önceki orijinal halinin traspozesi olmaktadır.

```
a=[ 1 2 1 3; 4 7 4 -10; -1 1 -1 -2; 3 -10 -2 5]
b=[2 16 1 -15]
x=b/a
```

Ters matris işlemi ile çözümde; MATRİS denklemi $AX=B$ biçiminde verildiğinde, A ve B matrisi

```
a=[ 1 4 -1 1; 2 7 1 -2; 1 4 -1 2; 3 -10 -2 5 ];
b=[ 2 16 1 -15 ]
```

biçiminde tanımlandıktan sonra

```
x=inv(a) *b;
```

bildirimi ile çözüm elde edilir: Benzer şekilde Matris denklemi $XA=B$ biçiminde verildiğinde A ve B matrisleri,

```
a=[ 1  2  1  3; 4  7  4 -10; -1  1 -1 -2; 3 -10 -2  5]
b=[2  16  1 -15]
x=b*inv(a)
```

bildirimi ile çözüm elde edilir. Yukarıda verilen bildirimler yolu ile x çözümü için

x= 2.0000 1.0000 3.0000 -1.0000

şeklinde elde edilmiş olur. Burada $x_1=2$, $x_2=1$, $x_3=3$ $x_4=-1$ 'dir.

4.4 DOĞRUSAL(lineer) OLMAYAN DENKLEMLERİN ÇÖZÜMÜ

Doğrusal olmayan denklemlerin doğrusal denklemlerde olduğu gibi tek bir standart biçimi yoktur.

Gerek MATLAB içinde gerekse Optimization Toolbox içinde, gerek tek değişkenli ve gerekse çok değişkenli denklemlerin çözümünde kullanılan çeşitli çözüm fonksiyonları vardır. Doğrusal olmayan denklemlerin çözümü, doğrusal denklem çözümü kadar basit olmayıp bunların çözümü için ayrıca bir fonksiyon dosyası hazırlanması gerekir.

Burada, MATLAB içinde yer alan fzero fonksiyon fonksiyonu ile Optimization Toolbox içinde yer alan fsolve fonksiyon fonksiyonu ayrıntılı bir biçimde ele alınacaktır. Ayrıca diğer doğrusal olmayan fonksiyon fonksiyonlarının kısaca tanımları gözden geçirilecektir.

fzero: Fonksiyon fonksiyonu; tek değişkenli bir fonksiyonun sıfırını hesaplar. Genel kullanım biçimleri aşağıda olduğu gibidir.

```
z=fzero('function',X0);
z=fzero('function',X0,tol);
z=fzero('function',X0,tol, trace);
```

fun(x) biçimindeki bir fonksiyonun, X0 ile tanımlanan değere yakın olan tek bir sıfırını hesaplar. Burada, fonksiyonu sıfır yapan, yani x eksenini kesen bir sıfır değeri hesaplanır.

İkinci bildirimde yer alan tol isimli, seçimli argüman bağıl hata toleransını belirler. Üçüncü bildirimde yer alan seçimli trace argümanı her bir hesap yineleme işlemindeki bilgileri görüntüler.

Fonksiyon fonksiyonu olan fzero fonksiyonunu kullanmak için ayrıca function ile başlayan bir fonksiyon dosyası hazırlanması gerekir.

Örnek: $f(x)=x^3-2x-5$ fonksiyonunun bir sıfırını bulunuz.

Çözüm: Önce bir `fx.m` adı ile fonksiyon dosyası hazırlanır.

```
Function y=fx(x)
y=x^3-2*x-5
```

Burada dosya adı 'fx' ile fonksiyon adı 'fx' aynı olması gerektiğine dikkat edilmelidir. Daha sonra MATLAB ortamında

```
z=fzero('fx',2)
```

bildirimi ile

```
z=2.0946
```

sonucu elde edilir. Burada $X_0=2$ olarak tahmini bir başlangıç değeri verilmiştir.

$f(x)$ fonksiyonu gerçekten bir polinom olduğuna göre aşağıda verilen `roots` komutu ile

```
p=roots([1 0 -2 -5])
```

Aynı fonksiyonu sıfır yapan gerçekteki değer ile birlikte karmaşık eşlenik kökleri de;

```
p=
2.0946
-1.0473 + 1.1359i
-1.0473 - 1.1359i
```

olarak elde edilmiş olur

Örnek : $e^{2x}-x-2$ biçiminde verilen doğrusal olmayan fonksiyonun bir adet sıfırını bulunuz.

Çözüm:Burada $f(x)$ fonksiyonu;

```
f(x)=e^{2x}-x-2 biçimine sokulabildiğine göre fonksiyon dosyası;
function y=fex(x)
y=exp(2*x)-x-2;
```

biçiminde hazırlanır. Daha sonra MATLAB ortamında;

```
z=fzero('fex',1)
z0.4475
```

elde edilir.

Yukarıdaki örneklerden de görüldüğü gibi `fzero` fonksiyonu herhangi bir fonksiyonun tahmini bir sıfırının hesaplanmasında ve/veya doğrusal olmayan denklemlerin çözümünde daha kullanışlıdır.

Doğrusal denklemlerin aynı anda tüm köklerini çözmek gerektiğinde `roots` fonksiyonunu kullanmak daha pratik olacaktır.

4.5 DOĞRUSAL OLMAYAN DENKLEM SİSTEMLERİNİN ÇÖZÜMÜ

Doğrusal olmayan denklem sistemlerinin çözümünde, Optimization Toolbox içinde yer alan fsolve fonksiyon fonksiyonu kullanılır. fsolve fonksiyonu doğrusal olmayan denklem takımının çözümünü sağlar.

fsolve fonksiyonunun belli başlı kullanım biçimleri aşağıda olduğu gibidir.

```
x=fsolve('fun',x0)
x=fsolve('fun',x0,options)
x=fsolve('fun',x0,options,'grad')
x=fsolve('fun',x0,options,'grad',p1,p2,...)
[x,options]=fsolve('fun',x0,...)
```

fsolve doğrusal olmayan denklemlerin köklerini hesaplar. Çıkış argümanı olan X değerleri; $F(x)=0$ şeklinde hesaplanır. Burada $F(x)$ ve X skaler, vektör veya matrislerden ibaret olabilir.

$x=fsolve('fun',x0)$ bildirimi, fun.m isimli M-dosyasında tanımlanan denklemleri, X0 tahmini başlangıç değerlerinden başlayarak çözer ve sonucu X değişkenine atar. Burada X0 boyutu x değişken sayısı kadar olmalıdır.

İkinci bildirimde yer alan seçimli argüman options seçimli parametreler vektörünü tanımlar. options için pek çok seçenek mevcuttur. Bunlar ile ilgili bilgiler help fsolve yolu ile sağlanabilir.

Üçüncü bildirimde yer alan grad, X noktasında fonksiyonların kısmi türevlerini (Jacobianlarını) df/dx , $df=grad(x)$ elde etmek için kullanılır. df 'in i'inci sütunu f'deki fonksiyonun i'inci kısmi türevine karşılık gelir.

Örnek: Bir metal kesme işlemine ait denklem takımı

$$C=1.2+11.62323v-1f-1+5.7449x10-8v3f0.16d1.14$$
$$0.0499v0.95 f0.78 d0.75=20$$

biçiminde tanımlanmaktadır.

Burada c 1.27 ile 1.28 arasında bir değerdir. Tamamen nonlinear olan bu denklemlerin çözümü için gerekli fonksiyon dosyası

```
function f=nlmr(x)
&x(1)=v, x(2)=f, x(3)=d, değişkenlerine .karşılık gelmektedir.
F(1)=-0.08+11.6323/(x(1)*x(2))+5.744e-8 . . .
(x(1).^3)* (x(2).^0.16) * (x(3).^1.14);
f(2)=12.7 -0.015*(x(1).^(-1.52))* (x(2).^1.004) . . .
* ( x(3) . ^0.25);
f(3)=20-0.0449*(x(1).^0.95)*(x(2).^0.78)
* (x(3) . ^0.75);
```

hazırlanabilir.

Daha sonra MATLAB ortamında fsolve ile aşağıdaki bildirimler yolu ile çözülür. Burada en önemli husus başlangıç değerinin seçimidir. Uygun bir çözüm elde edilene kadar başlangıç değerlerinin seçimi değiştirilebilir.

```
x=fsolve('nlmr',[0.5 112.5 20]);
```

Bildirimi ile

```
X= 0.6686 217.4627 18.2436
```

Sonucu elde edilir. Bu sonuç f fonksiyonları ile test edildiğinde

```
f= 1.0e-008 *(-0.0573 -0.5991 0.5776)
```

sıfıra çok yakın değerler elde edildiği görülür. Buna karşılık X0 başlangıç değerleri aşağıda olduğu gibi seçilecek olursa

```
>>x=fsolve('nlmr',[1 2 100]);
```

Warning: Matrix is close to singular or badly scaled.
Results may be inaccurate. RCOND= 6.79344e-017
Maximum number of iterations has been exceeded

Biçiminde bir uyarı alınır. Buda çözümün yeterli tamlıkta gerçekleşmediğini gösterir. Gerçekten de f fonksiyon değerleri aşağıdaki biçimde yazılarak test edildiğinde

```
>> F(1)=-0.08+11.6323/(x(1)*x(2))+5.744e-8 ...  
(x(1).^3)*(x(2).^0.16)*(x(3).^1.14);
```

```
>> f(2)=12.7-0.015*(x(1).^(-1.52))*(x(2).^1.004) ...  
*(x(3).^0.25);
```

```
>> f(3)=20-0.0449*(x(1).^0.95)*(x(2).^0.78)  
*(x(3).^0.75);
```

```
f= 0.1715 0.2316 0.5576
```

sıfırdan oldukça farklı değerler elde edildiği görülür. Bu durumda başlangıç değerlerini değiştirmek sureti ile uygun çözümler araştırılmalıdır.

Optimization Toolbox içinde, doğrusal olmayan denklemlerin, değişik şekilde çözümünde kullanılan diğer pek çok fonksiyon vardır. Bunların tanımları aşağıda kullanılan diğer pek çok fonksiyon vardır. Bunların tanımları aşağıda olduğu gibidir. Genel kullanım biçimleri MATLAB ortamında help komutu ile elde edilebilir.

attgoal	Çoklu-amaçlı hedefe ulaşma problemi çözümü
constr	Kısıtlı minimizasyon çözümü
fmin	Skaler kısıtsız minimizasyon çözümü.
fminu, fmins	Kısıtsız minimizasyon çözümü.
Fsolve	Doğrusal olmayan denklem çözümü.
leastssq	Doğrusal olmayan en küçük kareler çözümü.
minimax	Minimum-maksimum çözümü.

seminf	Yarı mutlak minimizasyon
lp	Doğrusal programlama
nnls	Negatif olmayan en küçük kareler çözümü.
qp	Eğrisel programlama.

5.1 DİFERANSİYEL DENKLEM ÇÖZÜMLERİ

Birinci dereceden bir diferansiyel denklemin genelleştirilmiş biçimi

$$y' = \frac{dy}{dx} = g(x, y)$$

şeklinde ifade edilebilir. Burada; x bağımsız değişken ve y de bağımlı değişkendir. Birinci dereceden bir diferansiyel denklemin çözümü $f'(x) = g(x, y)$ fonksiyonundan elde edilen $y=f(x)$ şeklinde bir fonksiyondur. Bir diferansiyel denklem çözümünün sayısal hesaplaması y'

türevinden y nin elde edilmesi için gerekli integrasyon işlemini ($y = \int_{x=a}^{x=b} g(x, y) dx$) kapsar.

Bir diferansiyel denklemin çözümü genellikle bir fonksiyonlar ailesi biçimindedir. Tek bir özel çözümün hesaplanabilmesi için sınır koşullarının ($x=a$, $x=b$) tanımlanması gerekir.

5.2 DİFERANSİYEL DENKLEMLERİN SAYISAL ÇÖZÜMÜ

MATLAB'ta diferansiyel denklemlerin sayısal çözümünde kullanılan; ode23 ve ode45 adında iki adet Ruge-Kutta fonksiyonu mevcuttur. Bunlardan; ode23 fonksiyonu ikinci ve üçüncü dereceden Runge-Kutta integrasyon denklemlerini, ode45 fonksiyonunda dördüncü ve beşinci dereceden Runge-Kutta integrasyon denklemlerini kullanır. ode23 ve ode45 fonksiyonlarında aynı türden giriş ve çıkış argümanları kullanılır.

Ode fonksiyonlarının genel kullanım şekli;

```
[y t] = ode23('yprime', t0, tf, yo, tol, trace);
```

biçimindedir. Giriş argümanlarının açıklaması ise aşağıda olduğu gibidir.

Yprime:

İntegre edilmesi gereken diferansiyel denklem takımını tanımlayan fonksiyon dosyası, m-uzantılı olarak saklanan fonksiyon dosyası ve fonksiyon adı aynı (örneğin yprime) olmalı ve ode fonksiyonları ile çağrıldığında “ (tırnak) içinde gösterilmelidir.

Kullanıcı tarafından hazırlanan fonksiyon dosyası bağımsız değişken, x ve bağımlı değişken, y olmak üzere iki giriş argümanına ve aşağıda gösterilen, durum türevlerinin bir sütun vektörü olan ydot(y') çıkışına sahiptir.

$$y_i' = \frac{dy}{dx} \text{ veya } y_i' = \frac{dy}{dt}$$

Fonksiyon dosyası function ydot=dosya_adı(t,y) bildirimi ile başlar ve çözülecek DD takımı denklemlerinin MATLAB formatında yazılımı ile devam eder.

t0:

İntegrasyon için başlama zamanı veya $y=f(x)$ şeklinde herhangi bir fonksiyonun integre edilmesi halinde integrasyon aralığının başlangıç değeri, a.

tfinal:

İntegrasyon zamanının nihai değeri veya $y=f(x)$ şeklinde herhangi bir fonksiyonun integre edilmesi halinde integrasyon aralığının sona erme değeri, b.

y0:

Bağımsız değişkenin veya sınır değerleri. Bir diferansiyel denklem takımının çözümü halinde başlangıç koşulları bir satır vektörü biçiminde olur.

tol:

Seçime bağlı bir argüman olup çözümde integrasyon işleminin hata sınırlarını belirler. Eğer bu argüman çağırılan fonksiyonda yer almazsa ode23 fonksiyonunda hata sınır değeri $1.0e-3$ ve ode45 ile hata sınır değeri $1.0e-6$ olarak otomatik belirlenir.

trace:

İsteğe bağlı bir argüman olup hesaplama sırasında elde edilen ara değerlerin saklanıp saklanmayacağını belirleyen bir bayrak (flag) tır. 0 değeri saklanmayacağını ve 1 değeri saklanacağını gösterir. Kullanılmaması halinde otomatik olarak 0 kabul edilir.

ode fonksiyonları ile elde edilen çözümün sonuçları bir t ve y matrisleri biçiminde daha sonra kullanılmak üzere saklanır. Burada birinci argüman, t bir sütun vektörü şeklinde zaman değerlerini veya $y=f(x)$ halinde bağımsız fonksiyon değerlerini temsil eder. İkinci argüman, y; diferansiyel denklemin derecesine bağlı olarak durum değişkenleri matrisini temsil eder. Bu matris durum değişkenlerinin sayısı kadar sütuna sahiptir. Satır sayısı ise çözümün durumunu belirler.

ode fonksiyonlarının kullanımı ve bunlar ile birlikte kullanılacak diferansiyel denklem fonksiyon dosyalarının yazılmasını açıklamak üzere basit bir birinci dereceden diferansiyel denklem çözüm örneği verelim.

Örnek: Aşağıda verilen birinci dereceden diferansiyel denklemin $y(0)=1$ başlangıç koşulu ve $[1,10]$ sınır değerleri arasındaki çözümü bulalım.

$$y' = \frac{dy}{dx} = g_1(x,y) = 4x^3 \cos x^2 - x^2$$

Çözüm: Problemin çözümü için, ilkönce aşağıdaki fonksiyon dosyası yazılır. Fonksiyonun yazılması için MATLAB'ın dahili editörü kullanılabilir. Bunun için MATLAB komut satırında `edit` yazılıp `enter`'a basılır.

MATLAB'ta sayısal diferansiyel çözümleri için `ode` fonksiyonları kullanılır. `ode` fonksiyon dosyaları daha önceden de anlatıldığı gibi fonksiyon dosyalarıdır. Yani çözümü üretilecek probleme ait fonksiyonu girdi olarak kabul ederler.

Sonucu ise problemin bağımlı ve bağımsız değişkenlerine karşılık gelecek şekilde matrisel olarak üretir. Bundan dolayıdır ki, eleman sayısı birden fazla olan matrisel sonuçları gözlemlemenin en iyi yolu grafiğini çizdirmektir.

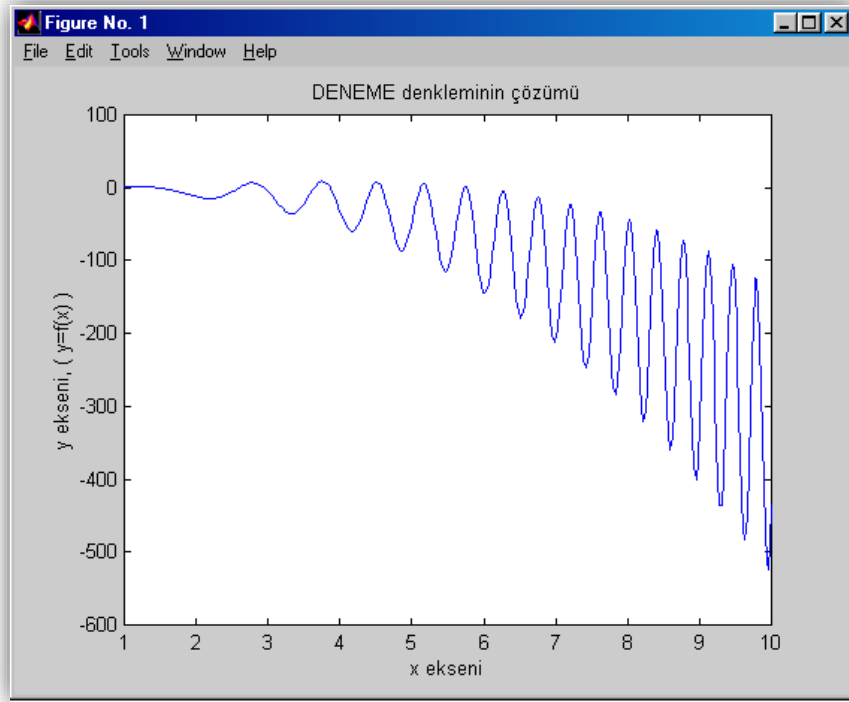
MATLAB programında bu işi yapan fonksiyon ya da komut `plot` (çiz) komutudur. `Plot` komutunun kullanım şekli problemin çözümü sırasında anlatılacaktır

```
function dy =deneme (x,y)
dy=4*x^3*cos (x^2) - x^2;
```

`function` kelimesiyle başlayan dosyanın fonksiyon adı `deneme` olarak belirlendiğinden dosya adı da `deneme.m` olarak saklanmalıdır.

MATLAB ortamında çözüm sağlayan bildirimler ve grafik çıktısı aşağıdaki gibidir.

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



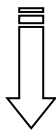
Resim 6: Çözümün grafik çıktısı

MATLAB ortamında çalıştırılan yukarıdaki bildirimler ile aşağıdaki adımlar gerçekleştirilmiştir;

- a) birinci satırda ode fonksiyonu ile çağrılan diferansiyel denklem fonksiyonunun sayısal çözümü gerçekleştirilmektedir; bundan sonra ode işleminin sonucunda a ve b 'nin içerikleri aşağıdaki gibi olmuştur.

a =

1.0000
1.0689
1.1474
1.2191
1.2850
1.3472
1.4055
1.4593
1.5059
1.5507
1.6170
1.7305



b =

1.0000
1.0720
1.1178
1.0967
0.9936
0.7938
0.4889
0.0855
-0.3743
-0.9221
-1.9382
-4.2596



b) ikinci satırda çözüm sonuçlarının `plot` komutu ile çizdirilmesi sağlanmaktadır. Burada `Plot` komutunun kullanım şekli aşağıdaki gibidir.

```
plot(x,y,seçenek)
```

x: Kordinat düzleminde x eksenine karşılık gelecek olan değerler topluluğu.

y: Kordinat düzleminde y eksenine karşılık gelecek olan değerler topluluğu.

seçenek: Seçenek parametresi zorunlu değildir. Grafik çıktısının görsel açıdan özelliklerini belirleyen bir değerdir. Örnekte kullanılan “b-“ ifadesi grafiğin mavi (b:blue) renkte çizgisel (solid) olarak oluşturulacağını belirtir. Kullanılabilecek parametreler aşağıdaki gibidir

renkler		nokta belirteçleri		çizim stilleri	
y	yellow	.	point	-	solid
m	magenta	o	circle	:	dotted
c	cyan	x	x-mark	-.	dashdot
r	red	+	plus	--	dashed
g	green	*	star		
b	blue	s	square		
w	white	d	diamond		
k	black	v	triangle (down)		
		^	triangle (up)		
		<	triangle (left)		
		>	triangle (right)		
		p	pentagram		
		h	hexagram		

c) üçüncü satırda ,açıklayıcı olması açısından grafiğe başlık verilmiştir.

d) dördüncü satırda, grafiğin x eksenine ait bir açıklama verilmiştir.

e) beşinci satırda, grafiğin y eksenine ait bir açıklama verilmiştir.

5.3 DİFERANSİYEL DENKLEMLERİN SEMBOLİK ÇÖZÜMÜ

MATLAB programı her ne kadar sembolik çözümler gerçekleştirmek üzere geliştirilmiş bir paket program olmasa da, MATLAB’a 4.0 sürümünden sonra The Symbolic Math Toolbox adı altında bir paketin eklenmesiyle bu iş içinde kullanılmaya başlanmıştır.

Sembolik çözümlemenin anlamı; bir sayısal çözüm elde etmek yerine, sayısal çözümün bir basamak gerisinde kalan, sayısal çözümü verebilecek bir fonksiyon elde etmektir. MATLAB’ta diferansiyel denklem çözümü için kullanılan fonksiyon fonksiyonu `dsolve` (differential solve: diferansiyel çözümle)’dur.

`dsolve` **fonksiyon fonksiyonu :**

`Dsolve` fonksiyonu sıradan diferansiyel eşitliklerin sembolik çözümünü bulur. `dsolve('eşitlik1','eşitlik2', ... , 'bsart1','bsart2',...)` şeklinde başlangıç şartlarını ve sıradan diferansiyel eşitlikleri parametre olarak alır. Bu şekilde birçok başlangıç şartı ve eşitlik beraberce kullanılabilir.

Dsolve fonksiyon fonksiyonu sunucu default olarak, bağımsız değişken “t” ye göre hesaplar. Bu “t” varsayılını, fonksiyonun sonuna eklenecek bir başka bağımsız değişken ile değiştirilebilir.örneğin;

```
dsolve('eşitlik1','eşitlik2', ... , 'bsart1','bsart2',...,'x')
```

“D” harfi bağımsız değişken sırası ile diferansiyel bulmayı belirtir, genellikle “D”, d/dt dir. Ayrıca $D2=d^2/dt^2$ olarak ifade edilir. Hemen “D” diferansiyel bulma operatörünü izleyen herhangi bir karakter, bağımlı değişken olarak alınır. Örnek olarak, $D3y$ $y(t)$ ’nin 3. türevini belirtir. Dikkat edilmesi gereken bir nokta vardır, buda “D” harfinin eşitliğin herhangi bir yerinde sembolik değişken olarak kullanılmamasıdır.

dsolve fonksiyonunda, y bağımlı değişkenlerden biri ve a ve b sabitler olduğunda $y(a)=b$ veya $Dy(a)=b$ şeklinde başlangıç şartları belirtilebilir. Eğer başlangıç şartlarının sayısı bağımlı değişkenlerin sayısından az verilirse, meydana gelen çözümler rasgele sabitler ($C1, C2, C3, \dots$ gibi) içerir.

dsolve fonksiyonu eğer, belirgin bir çözüm bulduysa bunu ekrana basar, eğer belirgin bir çözüm bulunamazsa bir uyarı verilir ve boş bir sym(symbolic) değeri geri döndürülür. Ayrıca dsolve fonksiyonu bazı durumlarda diferansiyel eşitlik veya integral içeren ifadeler içeren çözümler bulabilir.

5.3.1 Birinci Mertebeden Adi Diferansiyel Denklemler

Örnek 1: $(3x^5y^5-2y)dx + (5x^6y^4+x)dy=0$ denklemini çözelim.

Çözüm:

Denklemi dsolve fonksiyon fonksiyonunu kullanarak çözebilmek için aşağıda olduğu gibi ifade etmek gerekir. Burada matematiksel operatörlere ve bunları çevreleyen parantezlerin konumlarına dikkat etmek gerekir. Burada fazla parantez kullanmak lehimize olacaktır, böylece yazabileceğimiz olası hatalı ifadelerden de kurtulmuş oluruz.

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

Denklemin çözümü için MATLAB komut satırına aşağıdaki komutu girmek yeterlidir.

```
>> dsolve('(3*x^5*y^5-2*y)*Dx+(5*x^6*y^4+x)*Dy=0')
```

bu komuttan sonra MATLAB komut penceresinde görüntülenen çözüm fonksiyonu aşağıdaki gibidir.

ans=

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

Bu sonucu daha anlaşılır ifade etmek gerekirse;

$$\implies x^3y^5 + x^{-2}y = C \text{ 'dir.}$$

Örnek 2: $(x^2+3y^2) dx + 2xy dy=0$ denkleminin çözümünü bulalım.

Çözüm:

Bu denklemin çözümü için gerekli MATLAB komutu aşağıdaki gibidir.

```
>> dsolve('(x^2+3*y^2)*Dx+(2*x*y)*Dy=0')
```

Bu komutun çıktısı şöyle olur.

ans =

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

Bu sonucu daha anlaşılır ifade etmek gerekirse;

$$\Rightarrow \frac{x^5}{5} + x^3 y^2 = c \text{ 'dir.}$$

5.3.2 Homogen diferansiyel denklemler

Birinci mertebeden bir lineer adi diferansiyel denklemin

$$\frac{dy}{dx} = f(x, y)$$

şeklinde verildiğini biliyoruz. Eğer, y/x veya x/y nin bir g -fonksiyonu bulunabilirse, öyle ki,

$$\frac{dy}{dx} = f(x, y) = g(y/x)$$

olsun o zaman $f(x,y)$ fonksiyonuna homogen fonksiyon ve yukarıdaki denkleme homogen diferansiyel denklem denir.

Örnek 1: $(y + \sqrt{x^2 + y^2})dx - xdy = 0$ denklemini çözelim.

Çözüm:

Çözüm için gerekli MATLAB komutu aşağıdaki gibidir.

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

Burada `sqrt` fonksiyonu karekök alma fonksiyonudur. Problemin çözümü olarak;

cozum =

```
[ (1-2*C1*y(t))^(1/2)/C1  
 [ -(1-2*C1*y(t))^(1/2)/C1]
```

şeklinde bir sütun matris oluşmuştur. Burada matrisin 1. elemanı ile 2. elemanı arasındaki tek fark işaretlerinin farklı olmasıdır. Bu da problemin iki farklı çözümü olduğunu göstermektedir.

Burada sonuç ifadesinin okunabilirliği azdır. MATLAB'ta bunu normal, matematiksel tarzda ifade etmeye yarayan bir fonksiyon vardır. Bu fonksiyon aşağıdaki gibidir.

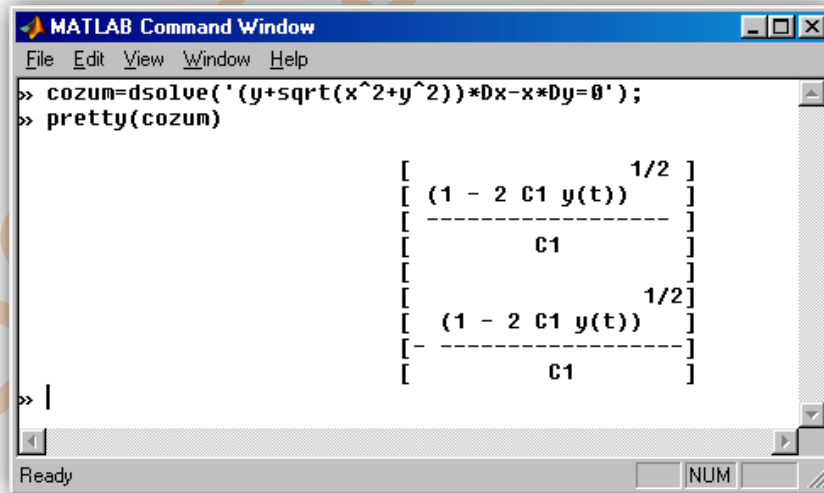
```
pretty(symbolic)
```

symbolic ifadesi normal matematiksel tarzda yazılması istenen fonksiyona karşılık gelmektedir. pretty fonksiyonun ile kullanılacak parametre mutlaka sembolik olmalıdır. Örnek olarak bulduğumuz cozum adındaki sütun matrisi sembolik biçimdedir. Bunu örneğimize uygularsak, yazacağımız komut satırı ve örnek ekran çıktısı aşağıdaki gibidir.

```
>> pretty(cozum)
```

veya

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



Resim 7: pretty komutunun kullanımı

5.3.3 Birinci mertebeden lineer diferansiyel denklemler

$\frac{dy}{dx} + P(x)y = Q(x)$ şeklindeki denklemlere lineer diferansiyel denklem denir.

Örnek 1: $y' + (1/x)y = \sin x$ denklemini çözelim

Çözüm:

Burada $y' = dy / dx$ olduğundan bağımsız değişken x 'tir. `dsolve` fonksiyonun sonunda belirtilir. Çözüm için gerekli MATLAB komutu aşağıdaki gibidir.

```
>> pretty(dsolve('Dy+(1/x)*y=sin(x)','x'))
```

elde edilen sonuç;

$$\frac{\sin(x) - x \cos(x) + C1}{x} \text{ 'gibidir.}$$

Örnek 2: $\frac{dy}{dx} = e^{2x} + 3y$ denklemini çözelim.

Çözüm:

MATLAB'ta kullanılabilecek bir "e" sabiti yoktur. "e" üzerili ifadeleri belirtmek için `exp(p)` fonksiyonu kullanılır. Buradaki p "e" nin üssü olan değerdir. Örneğin `exp(1)= 2.7183` 'dür. Çözüm için gerekli MATLAB komutu aşağıdaki gibidir.

```
>> pretty(dsolve('Dy=exp(2*x)+3*y','x'))
```

elde edilen sonuç;

$-\exp(2x) + \exp(3x) C1$ olur.

5.3.4 BERNOULLİ DENKLEMİ

Birinci mertebeden bir adi diferansiyel denklem,

$$\frac{dy}{dx} + P(x)y = y^n Q(x)$$

şeklinde ise bu diferansiyel denkleme Bernoulli denklemi denir.

Örnek: $\frac{dy}{dx} - \frac{y}{x} = -\frac{y^2}{x}$ denklemini çözelim

Çözüm:

Çözüm için gerekli MATLAB komutu aşağıdaki gibidir.

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

elde edilen sonuç;

$$\frac{x}{x + C1} \text{ olur.}$$

KAYNAKLAR

- [HTTP://WWW.MATHWORKS.COM](http://www.mathworks.com)
- [HTTP://EDUCATION.MATHWORKS.COM](http://education.mathworks.com)
- **DİFERANSİYEL DENKLEMLER VE UYGULAMALARI**
M.AYDIN
G. GÜNDÜZ
B. KURYEL
G.OTURANÇ
- **MATLAB İLE MÜHENDİSLİK SİSTEMLERİNİN ANALİZİ VE ÇÖZÜMÜ**
Prof.Dr. İbrahim YÜKSEL U.Ü.Makine Mühendisliği Bölümü 1996