3.7 Dosya Okuma / Yazma İşlemleri

MATLAB dosya okuma yazma işlemleri için değişik düzeylerde çok sayıda kütüphane fonksiyonu sunmaktadır. Ancak dersin ve bu notların amacı MATLAB programlama dilinin tüm ayrıntılarını anlatmak değildir. MATLAB bir araç olarak kullanılmak üzere çeşitli hesaplama ve görselleştirme işlemlerinin yürütülebilmesine yönelik yeteneklerin geliştirilmesi amaçlanmaktadır. Bu nedenle yalnızca sık kullanılan dosya okuma yazma yöntem ve araçları anlatılacaktır.

MATLAB programları veya komut satırı işlemlerinden üretilen verilerin saklanması ve belleğe tekrar aktarılmasında genel tüm gereksinimler **save** ve **load** fonksiyonları ile karşılanabilmektedir. Ancak özel biçimde (formatlı) veri yazma ve okuma işlemleri bu fonksiyonlar ile yürütülememektedir. Biçimli yazma okuma işlemleri ayrı bir başlık olarak anlatılacaktır. Veri saklama düzenleme ve paylaşımında sık kullanılan Microsoft Excel dosyaları da MATLAB tarafından yazılıp okunabilmektedir. İzleyen bölümde sırasıyla belirtilen dosya yazma ve okuma işlemleri anlatılacaktır.

save

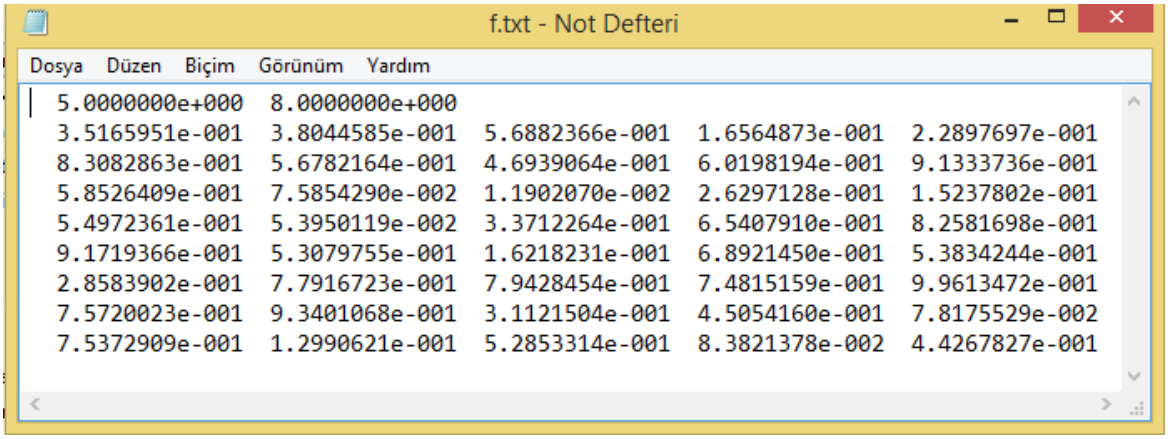
Çalışma belleğindeki değişkenlerin bir dosyaya kaydedilmesinin en kolay yolu **save** fonksiyonunu kullanmaktır. Bu fonksiyon çeşitli kullanım seçenekleri sunarak farklı biçimde dosyaların oluşturulmasına olanak vermektedir. Bilgisayar disk sürücülerinde dosyalar "binary" (bilgisayar dili) ya da ASCII kodlama türleri ile kaydedilebilmektedir. ASCII kodu ile kaydedilen belgeler kullanıcı tarafından platform bağımsız olarak açılıp okunabilen hatta içeriği düzenlenebilen metin dosyalarıdır. *.txt dosyaları buna örnek verilebilir. Aşağıdaki **save** fonksiyonunun değişik kullanım şekilleri ile MATLAB çalışma belleğinde olan değişkenlerin ASCII koduyla dosyaya kaydedilmesine örnekler verilmiştir:

```
f=rand(8,5);
save f.txt f -ASCII
```

Verilen örnekte 8x5 boyutlarında rastgele sayılardan oluşan bir f dizeyi oluşturulup f.txt isimli bir dosyada saklanmaktadır. `f.txt` yazılacak dosya adını, `f` dosyaya kaydedilecek değişken adını, `-ASCII` işaretçisi de kayıt biçimini belirtmektedir. Bellekteki birden çok değişken aralarına boşluk işareti konularak aynı dosyaya yazılabilir:

```
j=[5 8];
save f.txt j f -ASCII
```

Yukarıdaki ifadelerle oluşturulan f.txt dosyası geçerli çalışma dizinine kaydedilir. Dosya içeriği bir metin düzenleyicide açılarak görülebilir ve düzenlenebilir. Kayıt işleminde sayı biçimi olarak üstel gösterim kullanılmaktadır. İstenilen özel biçimlerde yazdırılması konusu daha sonra anlatılacaktır. Dosya içeriği aşağıdaki gibi görünecektir:



Şekil 3.1 **save** fonksiyonu ile kaydedilen dosyanın içeriği

Yukarıdaki yazılıştan farklı olarak dosya ve değişken isimlerinin daha esnek kullanımını sağlamak amacı ile özellikle program içerisinden çağrılmak üzere aşağıdaki yazım biçimi seçilebilir:

```
save('f.txt','f','-ASCII')
```

Bu yazılışın sağladığı üstünlük dosya ve değişken isimlerinin dinamik olarak değiştirilebilmesidir. Aşağıdaki örnekte bellekte sistematik olarak a1, a2, a3, a4 ve a5 şeklinde isimlendirilmiş değişkenlerde bulunan bilgilerin her birinin değişken adı ile uyumlu dosyalara yazılması gerçekleştirilmektedir:

```
a1=1; a2=2;a3=3;a4=4;a5=5;
for k=1:5
    d_adi=['a',num2str(k)];
    save([d_adi,'.txt'],d_adi,'-ASCII')
end
```

Bir başka örnek ise belirli bir aralıktaki açıların sinüsünü içeren bir tablonun bir metin dosyasına kaydedilmesi için verilebilir:

```
x=-2*pi:0.01:2*pi;
y=sin(x);
t=[x y];
save sintablo.txt t -ASCII
```

İkinci dosya kaydetme biçimi ***.mat** dosyalarının kullanımıdır. Bu dosyalar yalnızca MATLAB içerisinden kaydedilebilmekte ve okunabilmektedir. mat-dosyaları bellekte bulunan değişkenlerin türüne ve yapısına bakılmaksızın kaydedilebildiği MATLAB dosyalarıdır. Değişken isimleri ve yapısı aynen korunabilmektedir. Bu dosyalar MATLAB dışında metin düzenleyici ya da başka programlarca

açılamaz. Bunun dışında kayıt önceki bölümde anlatıldığı gibi **save** fonksiyonu ile yapılmaktadır. Bazı yazılış örnekleri aşağıda verilmiştir.

```
save tumu.mat %Bellekteki tüm değişkenleri tumu.mat dosyasına kaydeder.
```

```
save veriler.mat a b c %a,b,c değişkenlerini kaydeder
```

```
dosya='dinamik.mat';
save(dosya, 'X','C'); %X ve C değişkenlerini dinamik.mat dosyasına kaydeder
```

load

save fonksiyonu ile yukarıda anlatılan yollardan biri ile bir dosyaya kaydedilen veriler **load** fonksiyonu ile tekrar çalışma belleğine yüklenilebilir. ASCII kodlaması ile kaydedilmiş dosyaların içeriğinin bu şekilde okunabilmesi için dosya içeriğinin aynı tür veriden (sayısal) oluşması ve bir çizelge şeklinde olması gereklidir. Başka bir anlatımla tüm satırlarda aynı sayıda sütun olmalıdır. Fonksiyonun kullanımı metin dosyaları için aşağıdaki gibidir. Aynı dosyanın üç farklı yazım ile okunması gösterilmiştir. Yazımların birbirinden farkı açıklama olarak verilmiştir.

```
load veri.txt %veri.txt içeriğindeki bilgiler veri değişkenine aktarılır

a=load('veri.txt');%veri.txt içeriğindeki bilgiler a değişkenine aktarılır

file='veri.txt'%dosya adı file isimli değişkene atanmıştır
load(file)%değişkende saklanan dosya adı okunur
```

Metin dosyalarının bu şekilde yazılıp okunmasında dikkat edilecek iki önemli konu vardır. İlk olarak dosya isminin duruma göre aynı zamanda değişken adı olarak kullanılabileceğini göz önüne alarak sayılarla başlatmamak ve dosya isminde Türkçe karakter kullanmamaktır. Diğeri ise dosya içeriğinin bir tablo olarak düzenlenmesi gerekliliğidir.

mat-dosyalarının içeriğinin belleğe aktarılmasında da **load** fonksiyonu kullanılmaktadır. Ancak mat dosyalarında bilgiler değişken isimleri ile birlikte kaydedildiğinden değişkenler kaydedildiği isimleri ile belleğe aktarılacaktır. Bellekte aynı isimde değişken varsa silinerek dosya içeriğindeki ile değiştirilecektir. Aşağıda çeşitli durumlar için örnekler verilmiştir:

```
x=-3:.2:3;
y=0:5;
alfa=[0.25]';
f=exp(-alfa*x.^2);
save bilgi.mat x alfa f y
clear all

load bilgi.mat %dosya içeriğindeki tüm değişkenler belleğe okunur
load('bilgi.mat','x','alfa','f')%dosya içeriğinden yalnızca adları yazılı
%değişkenler belleğe okunur
```

Dosya ve değişken isimleri daha önce anlatıldığı gibi dinamik olarak değiştirilebilmektedir. Bu sistematik isimlendirilmiş dosya okumalarında büyük kolaylık sağlamaktadır. Bunun yanında daha önce tanımlanmış bir listede yer alan dosya isimleri sırasıyla işleme konulabilmektedir.

Formatlı yazma ve okuma

save ve **load** fonksiyonları dosya yazma ve okumada büyük kolaylıklar sunmaktadır. Ancak bazı durumlarda yetersiz kalmaktadır. mat-dosyaları yalnızca MATLAB kullanıcıları tarafından açılabileceğinden paylaşım sorunları ile karşılaşılabilmektedir. Diğer bir sorun ise çeşitli program veya yöntemlerde standartlaşmış yazım biçimleri kullanılıyor olması ve anılan fonksiyonların yalnızca tablo halinde veri yazma okumaya izin veriyor olmasıdır. MATLAB program veya işlemlerinin bu biçime uygun çıktı üretebilmesi ya da belirli biçimde yazılmış dosyaları okuyabilmesi gereklidir. Aşağıda verilen fonksiyonlar ile çok farklı biçim ve seçenekler ile metin dosyalarına yazma ve okuma işlemlerini gerçekleştirilebilmektedir. Ancak burada yine bu fonksiyon tüm yönleri ile değil en temel düzeyde kullanımları verilmiştir. Daha karmaşık durum ve seçenekler için ilgili fonksiyonlar için MATLAB yardım dosyalarına başvurulabilir.

MATLAB programlarından veya komut satırından bir dosyaya özel bir biçimde bilgi yazmak ya da okumak için dosyanın erişime açılması gereklidir. Bu işlem **fopen** fonksiyonu ile gerçekleştirilmektedir. Fonksiyonun genel kullanımı

```
fileID=fopen(<dosyaad>,<izin>)
```

şeklindedir. Burada dosyaad yerine geçerli dizinde bulunan ve bir dosyanın tam adı yazılmalıdır. Dosyanın hangi izinleri içerecek şekilde açılacağı ise izin alanına yazılır. Aşağıdaki çizelgede çeşitli dosya açma izinleri için bu alana neler yazılabileceği ve ne anlama geldikleri verilmiştir.

Çizelge 3.1 Dosya erişim izinleri

Erişim İzni	Anlamı
'r'	Dosyayı okumak için aç
'w'	Yeni ya da var olan dosyayı yazmak için aç. Dosya zaten varsa içeriğini sil.
'a'	Yeni ya da var olan dosyayı yazmak için aç. Dosya zaten varsa içeriğini koru.
'r+'	Dosyayı okumak ve yazmak için aç.
'w+'	Dosyayı okumak ve yazmak için aç. Dosya zaten varsa içeriğini sil.

Dosya açma işlemi başarılı olduğunda dosyaya bir erişim numarası atanır (fileID). Erişim numarası kullanıcı tarafından tanımlanan bir değişkende saklanılabilir. Bu dosya ile ilgili okuma ya da yazma işlemleri bu erişim numarası kaynak gösterilerek yapılabilir. Dosya açma işlemi başarısız olduğunda erişim numarası olarak -1 atanır. Dosya açma işleminin başarısız olma nedenleri, dosya adı ya da yolunun doğru ve tam yazılmamış olması, dosyanın başka bir programca kullanılıyor olması ya da böyle bir dosyanın olmaması sayılabilir. Aşağıda **fopen** ile dosyaları erişime açma için çeşitli örnekler verilmiştir:

```
%Ornek 1: veri.txt dosyası okunmak üzere açılıyor
fid=fopen('veri.txt','r');

%Ornek 2: Geçerli dizinden farklı bir konumda A1.dat isimli dosya
okumak/yazmak için açılıyor
dosya_no=fopen('C:\Belgeler\MATLAB\A1.dat','w+');

%Ornek 3: Sistematik adlandırılmış bir dizi dosya açılıyor (Olcum_1.dat,
Olcum_2.dat... )

for k=1:10
dosyaadi=['Olcum_',num2str(k),'.dat'];
Y1=fopen(dosyaadi,'w+');
...
...
end

%Ornek 3: Açılacak dosyanın Windows dosya açma diyaloğu ile belirlenmesi
[file,path]=uigetfile('*.txt');
fno=fopen([path file], 'r');
```

Dosya yukarıda anlatıldığı gibi açıldıktan sonra artık okunup / yazılabilir durumdadır. Uygun izin ile açılmış bir dosyaya bilgi yazmak için **fprintf** fonksiyonu kullanılır. Fonksiyonun genel kullanımı aşağıdaki gibidir:

```
fprintf(<fileID>,<biçim tanımlayıcı(lar)>,<değişkenler>)
```

Bu yazılışta fileID alanına dosya erişim numarası, biçin tanımlayıcılar alanına yazdırma biçimi (formatı) işaretçileri, değişkenler alanına da verilen biçimle dosyaya yazılacak değişken isimleri gelecektir. Biçim tanımlayıcılar için geçerli seçeneklerden bazıları aşağıdaki çizelgede verilmiştir.

Çizelge 3. 2 Dosya yazma ve okumada kullanılan bazı biçim tanımlayıcılar

Tanımlayıcı	Açıklama
`'%d'`	Tam sayı, 567 gibi
`'%f'`	Ondalıklı sayı, 0.652 gibi
`'%e'`	Üstel gösterim, 1.037e-05 gibi
`'%g'`	`%e` ve `%f` gibi ancak sonuna sıfırlar eklenmiyor
`'%c'`	Tek bir karakter, b gibi
`'%s'`	Bir dizi karakter, Ocak gibi
`'\n'`	Yeni satıra geç
`'\t'`	Sekme (tab) ekle

Dosyaya yazılacak bilgilerin türü ya da biçimi kadar hangi uzunlukta/kaç haneli bir alana yazılacağı da önemlidir. Örneğin deneme.txt adındaki dosyaya 5x1 boyutlarındaki N dizeyinin içeriği bir başlıkla birlikte yazdırılmak istendiğinde aşağıdaki MATLAB kodu yazılabilir.

```
dosya1=fopen('deneme.txt','w+');
N=[1.3 0.6 2 111.7]';
fprintf(dosya1,' Sayılar\n---------\n');
fprintf(dosya1,'%8.3f\n',N);
type deneme.txt
```

Verilen program parçası çalıştırıldığında oluşan dosyanın içeriği

```
 Sayılar
---------
   1.300
   0.600
   2.000
 111.700
```

şeklinde olacaktır. Burada biçim tanımlayıcısı `%8.3f` sayıların ondalıklı yazılacağını, bu sayı için toplam 8 karakter alanı kullanılabileceğini belirtmektedir. Bunlardan biri nokta işaretinin kendisi için ayrılmaktadır. Kalan 7 haneden üçü ondalıklı kısmı için dördü ise tam sayı kısmı için kullanılmaktadır. Yazdırılmak istenilen duyarlılığa göre bu hanelerin sayıları artırılıp azaltılabilir. Örnek program parçasındaki \n belirteci kendisinden sonra gelen ifade ve sayıların bir alt satırdan başlanarak yazılacağını göstermektedir. İki boyutlu bir dizi yukarıdakine benzer şekilde biçimli yazdırılmak istendiğinde her bir sütunu için ayrı bir format belirteci tanımlanmalıdır.

Biçimlendirilmiş bir metin dosyasından veri okunmasında ise **fscanf** fonksiyonu kullanılmaktadır. Fonksiyonun genel yazımı

```
A=fscanf(<fileID>,<biçim tanımlayıcı(lar)>,<boyut>)
```

şeklindedir. Burada fileID alanına dosya erişim numarası, biçim tanımlayıcı alanına Çizelge 3.8'de verilen seçeneklerden biri veya birkaçı, boyut alanına ise belirtilen biçime uyan kaç adet ya da hangi kaç satır kaç sütun veri okunacağı yazılır. Okunan değerler A değişkenine aktarılır. **fscanf** ile başlayan bir okuma işleminin sonucu tek bir dizeye aktarılabilir. Diğer bir deyişle bir **fscanf** komutu ile birden çok dizeyin içeriği okunamaz. Aşağıda verilen içerikteki dosyanın **fscanf** komutları ile okunmasına ait program konunun anlaşılması için faydalı olacaktır.

```
A10 Numaralı Istasyon
0 0 51
24
10       2        46
13       2        45
16       2        42.5
20       2        42
25       2        41.5
35       2        41
...     ...       ...
```

```
clear all
close all
fclose all
xyz_ist=[];
data=[];
fid=fopen(['A10_araziVeri.txt'],'r');
aciklama=fgetl(fid); %dosyadan bir satırın içeriğini okur
xk=fscanf(fid,'%g',[1 3]);%x,y,z koordinatları 1x3 boyutlarında
nd=fscanf(fid,'%d',1);%veri sayısı okunuyor
istv=fscanf(fid,'%g',[nd,3]);%ndx3 boyutlarında bir dizi okunuyor. Tüm sütunlar %g
biçiminde
```

Yukarıda temel bilgileri verilen okuma ve yazma işlemleri için daha ayrıntılı bilgilere MATLAB komut satırında `doc fprintf` veya `doc fscanf` yazarak erişilebilir. Bundan başka Microsoft Excel dosyalarından veri okumak üzere kullanılan `xlsread` fonksiyonu ile ilgili kullanım bilgileri dördüncü bölümdeki uygulamalar içerisinde verilecektir.

Alıştırmalar

1. $g(t) = \cos(\pi t) + \cos(10\pi t) + 0.5\cos(40\pi t)$ fonksiyonunu 0-5 saniye zaman aralığında 0.01s adımlarla örnekleyiniz. Fonksiyonun sayısal ve analitik türevlerini hesaplatınız. Sayısal ve analitik türevler arasındaki yüzde görecel hatayı da hesaplatarak aşağıdaki biçimde bir metin dosyasına yazdırınız.

```
   Zaman     Bekl.    Hesap.     % Hata
_____________________________________
  0.0050 -41.8955 -39.4928   0.0573
  0.0150 -74.1672 -70.2536   0.0528
  0.0250 -22.4609 -22.3697   0.0041
  0.0350  31.4201  27.6801   0.1190
  0.0450   5.4598   3.2048   0.4130
  0.0550 -68.5009 -65.9910   0.0366
  0.0650 -88.3855 -84.4156   0.0449
  0.0750 -22.9478 -22.8565   0.0040
  0.0850  44.6651  40.8688   0.0850
  0.0950  31.0933  28.7311   0.0760
     ...      ...      ...      ...
```

2. Aşağıdaki dosyayı okuyan bir MATLAB programı yazınız. Dosyanın ilk altı satırındaki bilgilerin her biri ayrı bir değişkene aktarılacaktır. Dördüncü satırda bu dosyada sütunlar halinde verilen bilgilerin kaç satır olduğu yazılıdır. Sütunlarda verilen bilgilerin tamamı bir değişkene aktarılacaktır.

```
a1-1a - pole-dipole array
1.50000
6
15
1
0
1.5  1.5     1        374.856
2.25 1.5     2        178.683
3    1.5     3        142.021
3.75 1.5     4        130.066
4.5  1.5     5        96.399
3    1.5     1        190.576
3.75 1.5     2        153.139
4.5  1.5     3        142.036
5.25 1.5     4        102.263
6    1.5     5        79.263
4.5  1.5     1        210.163
5.25 1.5     2        186.989
6    1.5     3        128.172
6.75 1.5     4        93.307
7.5  1.5      5       88.687
```