**M Dosyasında işlemler**

Komut dosyalarının nasıl oluşturulacağını ve MATLAB'da nasıl çalıştırılacağını öğrenmek. Komut dosyası, içinde MATLAB komutları dizisi bulunan, kullanıcı tarafından oluşturulan bir dosyadır. Dosya bir. m uzantısını ismine ekler, böylece onu bir M-dosyası yapar. Komut istemine adı (.m uzantısı olmadan) yazılarak bir komut dosyası yürütülür.

**Komut Dosyaları Hakkında Notlar**

• Komut dosyası, program olarak da adlandırılan MATLAB komutları dizisidir.

• Bir komut dosyası çalıştığında (yürütüldüğünde), MATLAB komutları, Komut Penceresine yazılmış gibi yazıldıkları sırada yürütür.

• Bir komut dosyası, dosyası bir çıktı oluşturan bir komuta sahipse (örn., Sonunda noktalı virgül olmadan bir değişkene bir değer atanması), çıktı Komut Penceresinde görüntülenir.

• Bir komut dosyası kullanmak uygundur çünkü düzenlenebilir (düzeltilebilir veya başka şekilde değiştirilebilir) ve birçok kez çalıştırılabilir.

• Komut dosyası dosyaları herhangi bir metin düzenleyicide yazılabilir ve düzenlenebilir ve ardından MATLAB düzenleyicisine yapıştırılabilir.

• Komut dosyası dosyaları, kaydedildiklerinde. m uzantısı kullanıldığından M dosyaları olarak da adlandırılır.

Pencere açıldıktan sonra, komut dosyasının komutları satır satır yazılır. MATLAB, Enter tuşuna her basıldığında otomatik olarak yeni bir satır numaralandırır. Komutlar ayrıca herhangi bir metin düzenleyicisine veya kelime işlemci programına yazılabilir ve ardından Düzenleyici / Hata Ayıklayıcı Penceresine kopyalanıp yapıştırılabilir. Editör / Hata Ayıklayıcı Penceresine yazılan bir kısa program örneği aşağıdaki Şekilde gösterilmektedir. Bir komut dosyasındaki ilk birkaç satır, komut dosyasında yazılan programı tanımlayan tipik olarak yorumlardır (satırdaki ilk karakter% olduğu için yürütülmez).

****

Bir komut dosyasının çalıştırılabilmesi için kaydedilmesi gerekir. Bu, Dosya menüsünden Farklı Kaydet ... seçilerek, bir konum seçilerek (birçok öğrenci, dizinde Sürücü (F :) veya (G :)) olarak görünen bir flash sürücüye kaydeder ve dosya kaydedildiğinde, MATLAB isme .m uzantısını ekler. Bir m.file dosyasını isimlendirme kuralları, bir değişkeni isimlendirme kurallarına uyar (bir harfle başlamalıdır, rakam ve alt çizgi içerebilir, boşluk içermez ve en fazla 63 karakter uzunluğunda olabilir). Kullanıcı tanımlı değişkenlerin, önceden tanımlanmış değişkenlerin ve MATLAB komutlarının veya işlevlerinin adları, komut dosyalarının adları olarak kullanılmamalıdır.

**Dosyanın Çalıştırılması:**

Bir komut dosyası, doğrudan Düzenleyici Penceresinden Çalıştır simgesine tıklanarak (bkz. Şekil) veya Komut Penceresine dosya adı yazıp ardından Enter tuşuna basılarak yürütülebilir. Bir dosyanın yürütülebilmesi için, MATLAB'ın dosyanın nereye kaydedildiğini bilmesi gerekir. Dosya, dosyanın kaydedildiği klasör MATLAB'ın mevcut klasörü ise veya klasör, arama yolunda listelenmişse yürütülecektir.

**Değişken Olarak Dizgi ve Dizgiler**

Bir dizge bir karakter dizisidir. Karakterler tek tırnak içine alınarak oluşturulur.

* Dizeler harfleri, rakamları, diğer sembolleri ve boşlukları içerebilir.
* Dizge örnekleri: 'ad ef', '3% fr2', '{edcba: 21!', 'MATLAB'.
* Tek tırnak içeren bir dize, dize içinde iki tek tırnak yazarak oluşturulur.
* Bir dizge yazılırken, ilk tek alıntı yazıldığında ekrandaki metnin rengi kestane rengine dönüşür. Dizenin sonundaki tek tırnak yazıldığında, dizenin rengi mora döner.
* MATLAB'da dizelerin birkaç farklı kullanımı vardır. Metin mesajlarını görüntülemek için çıktı komutlarında, çizimlerin biçimlendirme komutlarında ve bazı işlevlerin girdi argümanları olarak kullanılırlar. Bu amaçlar için dizeler kullanıldığında ilerleyen derlerimizde deyineceğiz.
* Dizeleri biçimlendirmede (eksenlere etiketler, başlık ve metin notlarında) dizeler kullanıldığında, dizedeki karakterler belirli bir yazı tipi, boyut, konum (büyük harf, küçük harf), renk vb. Olacak şekilde biçimlendirilebilir.
* Dizeler, aşağıdaki örneklerde gösterildiği gibi, değişkenlere atama operatörünün sağ tarafına yazarak da atanabilir:

a='FRty 8'

a =FRty 8

>> B='Merhaba ben Mehmet'

B =Merhaba ben Mehmet

Bir değişken bir dizge olarak tanımlandığında, dizenin karakterleri tıpkı sayılar gibi bir dizide saklanır. Boşluk dahil her karakter dizideki bir öğedir. Bu, tek satırlık bir dizenin, eleman sayısının karakter sayısına eşit olduğu bir satır vektörü olduğu anlamına gelir. Vektörlerin elemanları pozisyona göre adreslenir. Örneğin, yukarıda tanımlanan B vektöründe 4. eleman h harfi, 13. eleman M'dir ve bu böyle devam eder.

Dizeler ayrıca bir matrise yerleştirilebilir. Sayılarda olduğu gibi, bu da noktalı virgül yazarak yapılır; (veya Enter tuşuna basarak) her satırın sonunda. Her satır bir dize olarak yazılmalıdır, bu da tek tırnak içine alınması gerektiği anlamına gelir. Ek olarak, sayısal bir matriste olduğu gibi, tüm satırlar aynı sayıda öğeye sahip olmalıdır. Bu gereksinim, amaç belirli ifadelere sahip satırlar oluşturmak olduğunda sorunlara neden olabilir. Boşluklar eklenerek satırların aynı sayıda öğeye sahip olması sağlanabilir.

MATLAB, char adında bir dizi oluşturan yerleşik bir işleve sahiptir. hepsi aynı uzunlukta olmayan bir satır girişinden aynı sayıda karaktere sahip satırlar. MATLAB, kısa satırların sonuna boşluklar ekleyerek tüm satırların uzunluğunu en uzun satırın uzunluğuna eşit yapar. Char işlevinde, satırlar aşağıdaki biçime göre virgülle ayrılmış dizeler olarak girilir:

**değişken\_adı = char ('dize 1', 'dize 2', 'dize 3')**

>>Info=char('Student Name:','John Smith','Grade:','A+')

Info =

Student Name:

John Smith

Grade:

A+

Bir değişken, bir sayı veya aynı rakamlardan oluşan bir dize olarak tanımlanabilir. Örneğin, aşağıda gösterildiği gibi, x, 536 sayısı olarak tanımlanır ve y, 536 rakamlarından oluşan bir dize olarak tanımlanır.

>> x=536

x =

 536

>> y='536'

y =

536

Ekranda aynı görünseler bile iki değişken aynı değildir. X = altındaki satırdaki 536 karakterlerinin girintili olduğuna ve y = altındaki satırdaki 536 karakterlerinin girintili olmadığına dikkat edin. X değişkeni matematiksel ifadelerde kullanılabilirken y değişkeni kullanılamaz.

**ALIŞTIRMALAR**

1. m dosyası açarak içerisine clc; clear all; a=2 b=5 ve c=a\*b yazınız dosyanızı kaydederek çalıştırınız. Komut penceresinde sonuçlara bakınız. Daha sonra tüm işlemlere “;” koyarak çalıştırınız ve komut penceresine bakınız.
2. m dosyası açarak yukarıdaki 1 nolu ödevdeki işlemlerinizi içerisine yazınız. Dosyayı Matris işlemler olarak masaüstüne kaydediniz.
3. Yeni bir m file açarak dosyanıza a=1:100000 yazınız ve dosyayı çalıştırınız. Daha sonra işlemin sonuna; koyunuz.
4. Clear all; x=1+13^2-sgrt(16) ifadesini yazarak çalıştırınız. Dosyanıza y=x+10 yazarak çalıştırınız.
5. Ders1 isimli bir m-file dosyası oluşturunuz içerisine x=10; y=-16; yazınız aşağıdaki işlemleri içerisine sırayla yazarak çalıştırınız.
6. a=abs(y) yazınız. abs işlemi ne işe yarıyor?
7. B=sqrt(a)
8. ceil(4.2) ve floor(4.2) round(4.2)
9. c=x/a c\_ceil=ceil(c); c\_floor=floor(c); c\_round=round(c)
10. exp(1)
11. sign(-10) ve sign(10)
12. disp(x+y-a)
13. girdi=input(‘bir sayi giriniz’)
14. Ders2 isimli bir m dosyası oluşturunuz ve içerisine

clc; clear all;

sayi\_1=input(‘birinci sayiyi giriniz’);

sayi\_2=input(‘ikinci sayiyi giriniz’);

sonuc=sayi\_1^2+sqrt(abs(sayi\_2));

disp(sonuc);

4 ve -9 için el ile ve yazdığınız program ile hesabı yapınız.

1. x=linspace(0,10,5) 0 ile 10 arasında doğrusal aralıklarla 5 elemanlı bir x vektörü oluşturur.

Trigonometrik fonksiyonlar sin, cos vb. ve ayrıca temel matematik fonksiyonlar sqrt, exp, log, vb. vektörler üzerinde terim bazında çalışır.

y=sin(x)

z=sqrt(x).\*y

MATLAB çalışma alanı, bir MATLAB oturumu sırasında tanımlanan ve depolanan değişkenler kümesinden (adlandırılmış diziler) oluşur. Komut Penceresinde tanımlanan değişkenleri ve komut dosyaları yürütüldüğünde tanımlanan değişkenleri içerir. Bu, Komut Penceresinin ve komut dosyalarının bilgisayar içinde aynı bellek bölgesini paylaştığı anlamına gelir. Bu, bir değişken çalışma alanında olduğunda, tanındığı ve kullanılabileceği ve hem Komut Penceresinde hem de komut dosyalarında yeni değerlerin yeniden atanabileceği anlamına gelir. Bir sonraki derste açıklanacağı gibi, MATLAB'da değişkenlerin de tanımlanabildiği fonksiyon dosyası adı verilen başka bir dosya türü vardır. Ancak bu değişkenler, ayrı bir çalışma alanı kullandıkları için normalde programın diğer bölümleriyle paylaşılmaz.



Bir değişkene çift tıklamak, değişkenin içeriğinin bir tabloda görüntülendiği Değişken Düzenleyici Penceresini açar.



Değişken Düzenleyici Penceresindeki öğeler düzenlenebilir. Çalışma Alanı Penceresindeki değişkenler seçilerek ve ardından klavyedeki sil tuşuna basılarak veya düzenleme menüsünden sil seçeneği seçilerek silinebilir. Bu, Komut Penceresine clear variable\_name komutunu girmekle aynı etkiye sahiptir.

**Yazı dosyasına girişler (m. file)**

Bir komut dosyası yürütüldüğünde, dosya içindeki hesaplamalarda kullanılan değişkenlerin atanmış değerleri olmalıdır. Başka bir deyişle, değişkenler çalışma alanında olmalıdır. Bir değişkene bir değerin atanması, değişkenin nerede ve nasıl tanımlandığına bağlı olarak üç şekilde yapılabilir.

Bu durumda, değişkene bir değerin atanması, komut dosyasının bir parçasıdır. Kullanıcı dosyayı farklı bir değişken değeriyle çalıştırmak isterse, dosya düzenlenmeli ve değişkenin ataması değiştirilmelidir. Dosya kaydedildikten sonra tekrar çalıştırılabilir.

* Değişken tanımlanır ve komut dosyasında bir değer atanır

% Komut dosyası, üç oyunda alınan ortalama puanları hesaplar.

% Noktaların değerlerinin atanması, kod dosyasının bir parçasıdır.

game1=75;

game2=93;

game3=68;

ave\_points=(game1+game2+game3)/3

çıktısı:

ave\_points =

78.6667

* Değişken tanımlanır ve Komut Penceresinde bir değer atanır

Bu durumda değişkene bir değerin atanması Komut Penceresinde yapılır. (Değişkenin komut dosyasında tanındığını hatırlayın.) Kullanıcı komut dosyasını değişken için farklı bir değerle çalıştırmak isterse, Komut Penceresinde yeni değer atanır ve dosya yeniden çalıştırılır. Komut dosyasının, üç oyunda alınan puanların ortalamasını hesaplayan bir programa sahip olduğu önceki örnek için, komut dosyası (Oyun\_ort olarak kaydedilir):

% Komut dosyası, üç oyunda alınan ortalama puanları hesaplar.

% Noktaların değerlerinin değişkenlere atanması

% game1, game2 ve game3 Komut Penceresinde yapılır.

ave\_points = (game1 + game2 + game3) / 3

Komut penceresine aşağıdaki gibi tanım yapılıp

>> game1 = 67;

>> game2 = 90;

>> game3 = 81;

Dosya çalıştırılabilir, çıktısı

ave\_points =

79.3333

* Değişken komut dosyasında tanımlanır, ancak komut dosyası yürütüldüğünde Komut Penceresinde belirli bir değer girilir.

Bu durumda değişken komut dosyasında tanımlanır ve dosya yürütüldüğünde, kullanıcıdan Komut Penceresindeki değişkene bir değer ataması istenir. Bu, değişkeni oluşturmak için input komutu kullanılarak yapılır.

Giriş komutunun şekli:

değişken\_adı = input('bir mesaj içeren dizge Komut Penceresinde görüntülenir ')

Komut dosyası çalışırken input komutu yürütüldüğünde, dize Komut Penceresinde görüntülenir. Dize, kullanıcıdan değişkene atanmış bir değer girmesini isteyen bir mesajdır. Kullanıcı değeri yazar ve Enter tuşuna basar. Bu, değeri değişkene atar. Herhangi bir değişkende olduğu gibi, değişken ve atanan değeri, giriş komutunun en sonuna bir noktalı virgül yazılmadığı sürece Komut Penceresinde görüntülenecektir. Her oyunda alınan puanları, puanların ortalamasını hesaplayan programa girmek için input komutunu kullanan bir komut dosyası aşağıda gösterilmiştir.

% Bu komut dosyası, üç oyun içinde alınan puanların ortalamasını hesaplar.

% input komutunu kullanarak Her oyundan alınan puanlar değişkenlere atanır. %.

game1 = input ('İlk oyunda atılan puanları girin');

game2 = input ('İkinci oyunda atılan puanları girin');

game3 = input ('Üçüncü oyunda atılan puanları girin');

ave\_points = (game1 + game2 + game3) / 3

Dosya çalıştırıldığında komut penceresinde gelen komuta karşılık değerler girilir.

İlk oyumda alınan puanları girin 67

İkinci oyumda alınan puanları girin 91

Üçüncü oyumda alınan puanları girin 70

ave\_points =

76

Bu örnekte, değişkenlere skalar atanmıştır. Ancak genel olarak vektörler ve diziler de atanabilir. Bu, diziyi genellikle bir değişkene atandığı şekilde yazarak yapılır (sol köşeli parantez, ardından satır satır yazarak ve sağ köşeli parantez).

Girdi komutu, bir değişkene bir dizge atamak için de kullanılabilir. Bu, iki yoldan biriyle yapılabilir. Bunun bir yolu, komutu yukarıda gösterilenle aynı biçimde kullanmaktır ve komut istemi mesajı göründüğünde, bir dizenin bir değişkene input komutu olmadan atanması gibi, dize iki tek tırnak arasına yazılır. İkinci yol, girdi komutunda, girilen karakterleri dizge olarak tanımlayan bir seçenek kullanmaktır. Komutun şekli:

değişken\_adı = input (‘bilgi istemi’, ‘s’)

komutun içindeki "s", bir dizge olarak girilecek karakterleri tanımlar. Bu durumda, bilgi istemi mesajı göründüğünde, metin tek tırnak işaretleri olmadan yazılır, ancak değişkene bir dizge olarak atanır.

**Çıktı Oluşturan Komutlar**

Daha önce tartışıldığı gibi, MATLAB, bazı komutlar yürütüldüğünde otomatik olarak bir ekran oluşturur. Örneğin, bir değişkene bir değer atandığında veya önceden atanmış bir değişkenin adı yazıldığında ve Enter tuşuna basıldığında, MATLAB değişkeni ve değerini görüntüler. Komutun sonuna bir noktalı virgül yazılırsa, bu tür bir çıktı görüntülenmez. Bu otomatik ekrana ek olarak, MATLAB'ın ekran oluşturmak için kullanılabilecek birkaç komutu vardır. Ekranlar bilgi, sayısal veriler ve grafikler sağlayan mesajlar olabilir. Çıktı oluşturmak için sıklıkla kullanılan iki komut disp ve fprintf'dir. Disp komutu çıktıyı ekranda görüntülerken, fprintf komutu çıktıyı ekranda görüntülemek veya çıktıyı bir dosyaya kaydetmek için kullanılabilir.

Komutlar, Komut Penceresinde, bir komut dosyasında ve daha sonra gösterileceği gibi bir işlev dosyasında kullanılabilir. Bu komutlar bir komut dosyasında kullanıldığında, oluşturdukları görüntü çıktısı Komut Penceresinde görüntülenir.

**Disp Komutu:**

Disp komutu, değişkenin adını görüntülemeden bir değişkenin öğelerini görüntülemek ve metni görüntülemek için kullanılır. Disp komutunun biçimi şu şekildedir:

disp (değişkenin adı) veya disp ("dize olarak metin")

abc = [5 9 1; 7 2 4];

>> disp(abc)

5 9 1

7 2 4

>>disp('The problem has no solution.')

The problem has no solution.

**Örnek**

game1=input('Enter the points scored in the first game ');

game2=input('Enter the points scored in the second game ');

game3=input('Enter the points scored in the third game ');

ave\_points=(game1+game2+game3)/3;

disp(' ')

disp('The average of points scored in a game is:')

disp(' ')

disp(ave\_points)

89 60 ve 82 değerlerini sorduka biz giriyoruz Çıktısı

Enter the points scored in the first game 89

Enter the points scored in the second game 60

Enter the points scored in the third game 82

The average of points scored in a game is:

77

**fprintf Komutu:**

fprintf komutu, çıktıyı (metin ve veri) ekranda görüntülemek veya bir dosyaya kaydetmek için kullanılabilir. Bu komutla (disp komutunun aksine) çıktı biçimlendirilebilir. Örneğin, değişkenlerin metin ve sayısal değerleri karıştırılarak aynı satırda görüntülenebilir. Ayrıca sayıların formatı kontrol edilebilir.

Mevcut birçok seçenekle, fprintf komutu uzun ve karmaşık olabilir. Karışıklığı önlemek için komut kademeli olarak sunulur. İlk olarak, bu bölüm metin mesajlarını görüntülemek için komutun nasıl kullanılacağını, ardından sayısal verilerin ve metnin nasıl karıştırılacağını, ardından sayıların görüntüsünün nasıl biçimlendirileceğini ve son olarak çıktının bir dosyaya nasıl kaydedileceğini gösterir.

* Metni görüntülemek için fprintf komutunu kullanma:

fprintf (‘dize olarak yazılan metin’)

**Örnek:**

fprintf('Lutfen problemi kontrol ediniz')

Bu satır bir komut dosyasının parçasıysa, satır yürütüldüğünde Komut Penceresinde aşağıdakiler görüntülenir.

Lutfen problemi kontrol ediniz

printf komutu ile yeni satıra başlamak mümkündür. Örneğin, önceki örnekte ilk cümleden sonra \ n eklemek şunu verir:

fprintf('Lutfen problemi kontrol ediniz \nverileri kontrol ediniz.')

Lutfen problemi kontrol ediniz

verileri kontrol ediniz.

\ n'ye bir çıkış karakteri denir. Ekranı kontrol etmek için kullanılır. Dizeye eklenebilecek diğer çıkış karakterleri şunlardır:

\ b Geri al.

\ t Yatay sekme.

* Metin ve sayısal verilerin bir karışımını görüntülemek için fprintf komutunu kullanma:

fprintf ("% -5.2f ek metin dizesi olarak metin",değişken ismi)

Biçimlendirme öğeleri şunlardır:

-Alandaki sayıyı sola yaslar.

Alan genişliği ve hassasiyeti (önceki örnekte 5.2) isteğe bağlıdır. İlk sayı (örnekte 5), ekrandaki minimum basamak sayısını belirten alan genişliğidir. Gösterilecek sayı alan genişliğinden daha kısa ise, sayının önüne boşluklar veya sıfırlar eklenir. Hassasiyet ikinci sayıdır (örnekte 2). Ondalık ayırıcının sağında görüntülenecek basamak sayısını belirtir.

Biçimlendirme öğelerindeki gerekli son öğe, sayının görüntülendiği gösterimi belirten dönüştürme karakteridir.

Bazı yaygın gösterimler şunlardır:

e Küçük e kullanarak üstel gösterim (ör. 1.709098e + 001).

E Büyük harf E kullanan üstel gösterim (ör. 1.709098E + 001).

f Sabit noktalı gösterim (ör. 17.090980).

g e veya f gösterimlerinin daha kısa olanı.

G E veya f gösterimlerinin kısası.

i Tamsayı.

**Örnek:**

game(1) = input('Enter the points scored in the first game ');

game(2) = input('Enter the points scored in the second game ');

game(3) = input('Enter the points scored in the third game ');

ave\_points = mean(game);

fprintf('An average of %f points was scored in the three games.',ave\_points)

çıktısı

Enter the points scored in the first game 75

Enter the points scored in the second game 60

Enter the points scored in the third game 81

An average of 72.000000 points was scored in the three games.

fprintf komutuyla, metne birden fazla sayı (bir değişkenin değeri) eklemek mümkündür. Bu, metinde sayıların ekleneceği yerlere% g (veya% ve ardından herhangi bir biçimlendirme öğesi) yazarak yapılır. Ardından, komutun dize bağımsız değişkeninden sonra (virgülden sonra), değişkenlerin adları metne eklendikleri sırayla yazılır. Genel olarak komut şöyle görünür:

fprintf (‘.. text ...% g ...% g ...% f ...’, değişken1, değişken2, değişken3)

**Örnek:**

v=1584; % Initial velocity (km/h)

theta=30; % Angle (degrees)

vms=v\*1000/3600;

t=vms\*sind(30)/9.81;

d=vms\*cosd(30)\*2\*t/1000;

fprintf('A projectile shot at %3.2f degrees with a velocity

of %4.2f km/h will travel a distance of %g km.\n',theta,v,d)

Çıktısı

A projectile shot at 30.00 degrees with a velocity of 1584.00 km/h will travel a distance of 17.091 km.

Görüntülenen metne tek bir tırnak işareti koymak için, komutun içindeki dizeye iki tek tırnak işareti yazın.

• Metinde% karakterini görüntülemek için %% yazın.

• fprintf komutu vektörleştirilmiştir. Bu, komuta bir vektör veya matris olan bir değişken dahil edildiğinde, komutun tüm öğeler görüntülenene kadar kendini tekrarladığı anlamına gelir. Değişken bir matris ise, veriler sütun sütun kullanılır. Örneğin, aşağıdaki komut dosyası, ilk satırın 1'den 5'e kadar sayıları içerdiği ve ikinci satırın karşılık gelen karekökleri gösterdiği bir 2x5 T matrisi oluşturur.

x=1:5;

y=sqrt(x);

T=[x; y]

fprintf('If the number is: %i, its square root is: %f\n',T)

çıktısı

T =

1.0000 2.0000 3.0000 4.0000 5.0000

1.0000 1.4142 1.7321 2.0000 2.2361

If the number is: 1, its square root is: 1.000000

If the number is: 2, its square root is: 1.414214

If the number is: 3, its square root is: 1.732051

If the number is: 4, its square root is: 2.000000

If the number is: 5, its square root is: 2.236068

**Çıktıyı bir dosyaya kaydetmek için fprintf komutunu kullanma:**

Çıktıyı Komut Penceresinde görüntülemenin yanı sıra, fprintf komutu, çıktının kaydedilmesi gerektiğinde çıktıyı bir dosyaya yazmak için kullanılabilir. Kaydedilen veriler daha sonra MATLAB'da ve diğer uygulamalarda görüntülenebilir veya kullanılabilir.

Bir dosyaya çıktı yazmak için üç adım gerekir:

a) fopen komutunu kullanarak bir dosya açmak.

b) Çıktıyı fprintf komutunu kullanarak açık dosyaya yazmak.

c) fclose komutunu kullanarak dosyayı kapatmak.

Adım a:

Veriler bir dosyaya yazılmadan önce dosyanın açılması gerekir. Bu, yeni bir dosya oluşturan veya mevcut bir dosyayı açan fopen komutuyla yapılır. fopen komutu şu biçime sahiptir:

fid = fopen ("dosya\_adı", "izin")

fid, dosya tanımlayıcısı olarak adlandırılan bir değişkendir. fopen çalıştırıldığında fid'e skaler bir değer atanır. Dosya adı (uzantısı dahil) tek tırnak içinde bir dizge olarak yazılır. İzin, dosyanın nasıl açıldığını söyleyen bir koddur (ayrıca bir dize olarak yazılır). Daha yaygın izin kodlarından bazıları şunlardır:

‘r’ Dosyayı okumak için açın (varsayılan).

"w" Dosyayı yazmak için açın. Dosya zaten varsa, içeriği silinir. Dosya yoksa, yeni bir dosya oluşturulur.

"a" Dosya varsa, yazılı verilerin dosyanın sonuna eklenmesi dışında "w" ile aynıdır.

'r +' Dosyayı okumak ve yazmak için açın (oluşturmayın).

"w +" Dosyayı okumak ve yazmak için açın. Dosya zaten varsa, içeriği silinir. Dosya yoksa, yeni bir dosya oluşturulur.

"a +" Dosya varsa, yazılı verilerin dosyanın sonuna eklenmesi dışında "w +" ile aynıdır.

Komutta bir izin kodu yoksa, dosya varsayılan 'r' koduyla açılır. Ek izin kodları yardım menüsünde açıklanmıştır.

Adım b:

Dosya açıldığında, çıktıyı dosyaya yazmak için fprintf komutu kullanılabilir. fprintf komutu, fid değişkeninin komutun içine yerleştirilmesi dışında, Komut Penceresinde çıktıyı görüntülemek için kullanılanla tamamen aynı şekilde kullanılır. fprintf komutu daha sonra şu forma sahiptir:

fprintf (fid, "metin% -5.2f ek metin", değişken\_adı)

Adım c:

Verilerin dosyaya yazılması tamamlandığında, dosya fclose komutu kullanılarak kapatılır. fclose komutu şu biçime sahiptir:

fclose(fid)

Çıktıyı bir dosyaya kaydetmek için fprintf komutunun kullanımına ilişkin ek notlar:

• Oluşturulan dosya mevcut dizine kaydedilir.

• Birkaç farklı dosyaya yazmak için fprintf komutunu kullanmak mümkündür.

Bu, önce dosyaları açarak, her birine farklı bir etiket atayarak (ör. fid1, fid2, fid3, vb.) Ve ardından o dosyaya yazmak için fprintf komutundaki belirli bir dosyanın etiketi (fid)'i kullanılarak yapılır.

%fprintf'in çıktıyı dosyalara yazmak için kullanıldığı% Script dosyası.

% İki dönüştürme tablosu oluşturuldu ve iki farklı dosyaya kaydedildi.

% Biri mi/sa'yı km/saate dönüştürür, diğeri lb'yi N'ye dönüştürür.

Vmph = 10: 10: 100;

Vkmh = Vmph. \* 1.609;

TBL1 = [Vmph; Vkmh];

Flb = 200: 200: 2000;

FN = Flb. \* 4.448;

TBL2 = [Flb; FN];

fid1 = fopen ('VmphtoVkm.txt', 'w');

fid2 = fopen ('FlbtoFN.txt', 'w');

fprintf (fid1, 'Hız Dönüşüm Tablosu \ n \ n');

fprintf (fid1, 'mil / s km / s \ n');

fprintf (fid1, '% 8.2f% 8.2f \ n', TBL1);

fprintf (fid2, 'Kuvvet Dönüşüm Tablosu \ n \ n');

fprintf (fid2, 'Pounds Newton \ n');

fprintf (fid2, '% 8.2f% 8.2f \ n', TBL2);

fclose (fid1);

fclose (fid2);

Yukarıdaki komut dosyası çalıştırıldığında, VmphtoVkm ve FlbtoFN adlı iki yeni .txt dosyası oluşturulur ve geçerli dizine kaydedilir. Bu dosyalar, .txt dosyalarını okuyabilen herhangi bir uygulama ile açılabilir. Aşağıdaki şekiller, Microsoft Word ile açıldığında iki dosyanın nasıl göründüğünü gösterir.





**SAVE ve LOAD komutları**

Save (kaydet) ve load (yükle) komutları, MATLAB'da kullanılmak üzere verileri kaydetmek ve geri almak için en yararlıdır. Save komutu, o anda çalışma alanında bulunan değişkenleri kaydetmek için kullanılabilir ve load komutu, daha önce kaydedilmiş değişkenleri çalışma alanına almak için kullanılır. Çalışma alanı, MATLAB bir platform türünde (örneğin, PC) kullanıldığında ve başka bir platformda (örneğin, Mac) MATLAB'da kullanılmak üzere alındığında kaydedilebilir. Save ve load komutları, MATLAB dışındaki uygulamalarla veri alışverişi yapmak için de kullanılabilir. Bu amaçla kullanılabilecek ek komutlar gelecek bölümde verilmiştir.

**Save komutu**

Save komutu, çalışma alanında depolanan değişkenleri (tümü veya bir kısmı) kaydetmek için kullanılır. Kaydet komutunun en basit iki biçimi şunlardır:

save dosya\_adı ve save("dosya\_adı")

Bu komutlardan biri yürütüldüğünde, şu anda çalışma alanında bulunan tüm değişkenler, geçerli dizinde oluşturulan dosya\_adı.mat adlı bir dosyaya kaydedilir. İkili formatta yazılan mat dosyalarında her değişken adını, türünü, boyutunu ve değerini korur. Bu dosyalar diğer uygulamalar tarafından okunamaz. Save komutu, çalışma alanında bulunan değişkenlerin yalnızca bazılarını kaydetmek için de kullanılabilir. Örneğin, var1 ve var2 adlı iki değişkeni kaydetmek için komut şu şekildedir:

save dosya\_adı var1 var2 veya save("dosya\_adı", "var1", "var2")

Save komutu, MATLAB dışındaki uygulamalar tarafından okunabilen ASCII formatında kaydetme için de kullanılabilir. ASCII biçiminde kaydetme, komuta -ascii bağımsız değişkeni eklenerek yapılır (örneğin, save dosya\_adı -ascii). ASCII biçiminde değişkenin adı, türü ve boyutu korunmaz. Veriler, boşluklarla ayrılmış, ancak değişken adları olmadan karakterler olarak kaydedilir. Örneğin, aşağıda iki değişkenin (bir vektör ve bir matris) Komut Penceresinde nasıl tanımlandığı ve ardından ASCII formatında DatSavAsci adlı bir dosyaya kaydedildiği gösterilmektedir:

**>> V=[3 16 -4 7.3];**

**>> A=[6 -2.1 15.5; -6.1 8 11];**

**>> save -ascii DatSavAsci**

Dosya kaydedildikten sonra ASCII dosyalarını okuyabilen herhangi bir uygulama tarafından açılabilir. Örneğin, Şekil, dosya Not Defteri ile açıldığında verileri gösterir.



Dosyanın değişkenlerin adlarını içermediğine dikkat edin, sadece değişkenlerin sayısal değerlerinin (önce A ve sonra V) listelendiğini unutmayın.

**Load Komutu**

Load komutu, kaydetme komutuyla kaydedilen değişkenleri çalışma alanına geri almak ve diğer uygulamalarla oluşturulan ve ASCII formatında veya metin (.txt) dosyalarında kaydedilen verileri içe aktarmak için kullanılabilir. .mat dosyalarında kaydet komutuyla kaydedilen değişkenler şu komutla alınabilir:

load dosya\_adı load(‘dosya\_adı’)

Komut yürütüldüğünde, dosyadaki tüm değişkenler (kaydedilen adı, türü, boyutu ve değerleri ile) çalışma alanına eklenir (geri yüklenir). Çalışma alanında, load komutuyla alınan bir değişkenle aynı ada sahip bir değişken zaten varsa, alınan değişken mevcut değişkenin yerini alır. Load komutu, kaydedilen .mat dosyasında bulunan değişkenlerin yalnızca bazılarını almak için de kullanılabilir. Örneğin, var1 ve var2 adlı iki değişkeni almak için komut şu şekildedir:

load dosya\_adı var1 var2 veya load("dosya\_adı", "var1", "var2")

Load komutu, ASCII veya metin (.txt) olarak kaydedilen verileri çalışma alanına aktarmak için de kullanılabilir. Ancak bu, yalnızca dosyadaki veriler MATLAB'da bir değişken biçiminde olduğunda mümkündür. Böylece, dosya bir sayıya (skaler), bir satıra veya bir sayı sütununa (vektör) veya her birinde aynı sayıda sayıya sahip satırlara (matris) sahip olabilir.

Örneğin, Şekilde gösterilen veriler, yükleme komutuyla yüklenemez (kaydetme komutuyla ASCII formatında kaydedilmiş olsa bile), çünkü öğelerin sayısı tüm satırlarda aynı değildir. (Bu dosyanın iki farklı değişken kaydedilerek oluşturulduğunu hatırlayın.)

Veriler bir ASCII veya metin dosyasından çalışma alanına yüklendiğinde, bir değişken adına atanmalıdır. ASCII formatındaki veriler, yükleme komutunun aşağıdaki iki formundan biriyle yüklenebilir:

load file\_name veya VarName = load ("dosya\_adı")

Veriler bir metin dosyasındaysa, dosya adına .txt uzantısı eklenmelidir. Yükleme komutunun biçimi şu şekildedir:

load file\_name.txt veya VarName = load ("dosya\_adı.txt")

Komutun ilk biçiminde veriler, dosyanın adını taşıyan bir değişkene atanır. İkinci formda, veriler VarName adlı bir değişkene atanır.

Örneğin, Şekilde gösterilen veriler (3x3 bir matris) Not Defteri'nde yazılır ve ardından DataFromText.txt olarak kaydedilir.



Ardından, metin dosyasındaki verileri MATLAB'ın Çalışma Alanına aktarmak için iki yükleme komutu biçimi kullanılır. İlk komutta veriler DfT adlı bir değişkene atanır. İkinci komutta veriler otomatik olarak, verilerin kaydedildiği metin dosyasının adı olan DataFromText adlı bir değişkene atanır.

>> DfT=load('DataFromText.txt')

DfT =

56.0000 -4.2000

3.0000 7.5000

-1.6000 198.0000

>> load DataFromText.txt

>> DataFromText

DataFromText =

56.0000 -4.2000

3.0000 7.5000

-1.6000 198.0000

Bir sonraki bölümde sunulan MATLAB komutları ile diğer uygulamalara veri almak (veya dışarı aktarmak) da yapılabilir.

**VERİLERİ İÇE AKTARMA VE DIŞA AKTARMA**

MATLAB genellikle deneylerde kaydedilen veya diğer bilgisayar programları tarafından oluşturulan verileri analiz etmek için kullanılır. Bu, önce verileri MATLAB'a aktararak yapılabilir. Benzer şekilde, MATLAB tarafından üretilen verilerin bazen diğer bilgisayar uygulamalarına aktarılması gerekir. Çeşitli veri türleri (sayısal, metin, ses, grafikler ve resimler) vardır. Bu bölümde, yalnızca MATLAB'ın yeni kullanıcıları tarafından aktarılması gereken en yaygın veri türü olan sayısal verilerin nasıl içe ve dışa aktarılacağı açıklanmaktadır. Diğer veri aktarımı türleri için, Dosya I / O altındaki Yardım Penceresine bakın.

Verilerin içe aktarılması, komutlar veya İçe Aktarma Sihirbazı kullanılarak yapılabilir. Komutlar, içe aktarılan verilerin formatı bilindiğinde kullanışlıdır. MATLAB, çeşitli veri türlerini içe aktarmak için kullanılabilecek birkaç komuta sahiptir. İçe aktarma komutları, komut dosyası yürütüldüğünde verilerin içe aktarılması için bir komut dosyası dosyasına da dahil edilebilir. İçe Aktarma Sihirbazı, verilerin biçimi (veya verileri içe aktarmak için uygun komut) bilinmediğinde kullanışlıdır. İçe Aktarma Sihirbazı verilerin biçimini belirler ve otomatik olarak içe aktarır.

**Verileri İçe ve Dışa Aktarma Komutları**

Bu bölümde ayrıntılı olarak verilerin Excel elektronik tablolarına nasıl aktarılacağı açıklanmaktadır. Microsoft Excel, verileri depolamak için yaygın olarak kullanılır ve Excel, birçok veri kayıt cihazı ve bilgisayar uygulamasıyla uyumludur. Birçok kişi ayrıca çeşitli veri biçimlerini Excel'e ve Excel'den içe ve dışa aktarabilir. MATLAB ayrıca, verileri doğrudan csv ve ASCII gibi formatlara ve ayrıca Lotus 123 elektronik tablo programına aktarmak için komutlara sahiptir. Bunların ve diğer birçok komutun ayrıntıları, Dosya I / O altındaki Yardım Penceresinde bulunabilir.

**Verileri Excel'e ve Excel'den dışa aktarma:**

Verilerin Excel'den içe aktarılması xlsread komutu ile yapılır. Komut yürütüldüğünde, elektronik tablodaki veriler bir değişkene bir dizi olarak atanır. xlsread komutunun en basit şekli şudur:

değişken\_adı = xlsread ("dosya adı")

* "Dosya adı" (bir dize olarak yazılır) Excel dosyasının adıdır. Excel dosyasının dizini ya geçerli dizin olmalı ya da arama yolunda listelenmelidir.
* Excel dosyasının birden fazla sayfası varsa, veriler ilk sayfadan alınacaktır.

Bir Excel dosyasının birkaç sayfası olduğunda, xlsread komutu, verileri belirli bir sayfadan içe aktarmak için kullanılabilir. Komutun biçimi o zaman:

değişken\_adı = xlsread ("dosya adı", "sayfa\_adı")

• Sayfanın adı bir dizge olarak yazılır.

Diğer bir seçenek de elektronik tablodaki verilerin yalnızca bir bölümünü içe aktarmaktır.

Bu, komuta ek bir argüman yazarak yapılır:

değişken\_adı = xlsread ("dosya adı", "sayfa\_adı", "aralık")

"Aralık" (dize olarak yazılır), bölgenin zıt köşelerindeki hücrelerin adresleriyle (Excel gösteriminde) tanımlanan elektronik tablonun dikdörtgen bir bölgesidir. Örneğin, "C2: E5" 2., 3., 4. ve 5. satırlar ile C, D ve E sütunlarından oluşan bir bölgedir.

MATLAB'dan bir Excel elektronik tablosuna veri aktarımı, xlswrite komutu kullanılarak yapılır. Komutun en basit şekli şudur:

xlswrite ("dosya adı", değişken\_adı)

"Dosya adı" (bir dize olarak yazılır) verilerin dışa aktarıldığı Excel dosyasının adıdır. Dosya geçerli dizinde olmalıdır. Dosya yoksa, belirtilen adda yeni bir Excel dosyası oluşturulur.

• değişken\_adı, dışa aktarılan atanmış verilerle MATLAB'daki değişkenin adıdır.

• "sayfa\_adı" ve "aralık" bağımsız değişkenleri, sırasıyla belirli bir sayfaya ve belirli bir hücre aralığına dışa aktarmak için xlswrite komutuna eklenebilir.

Örnek olarak, Şekilde gösterilen Excel elektronik tablosundaki veriler xlsread komutu kullanılarak MATLAB'a aktarılır.



Elektronik tablo, A sürücüsündeki bir diskte TestData1 adlı bir dosyaya kaydedilir. Geçerli Dizin A sürücüsüne değiştirildikten sonra, veriler MATLAB değişkenine DATA atanarak içe aktarılır:

>> DATA = xlsread('TestData1')

DATA =

11.0000 2.0000 34.0000 14.0000 -6.0000 0 8.0000

15.0000 6.0000 -20.0000 8.0000 0.5600 33.0000 5.0000

0.9000 10.0000 3.0000 12.0000 -25.0000 -0.1000 4.0000

55.0000 9.0000 1.0000 -0.5550 17.0000 6.0000 -30.0000

**İçe Aktarma Sihirbazını Kullanma**

İçe Aktarma Sihirbazını kullanmak, kullanıcının verinin biçimini bilmesi veya belirtmesi gerekmediğinden, verileri MATLAB'a aktarmanın muhtemelen en kolay yoludur. Alma Sihirbazı, Komut Penceresinin Dosya menüsünde Verileri İçe Aktar seçilerek etkinleştirilir. (Ayrıca uiimport komutu yazılarak da başlatılabilir.) Alma Sihirbazı, Sihirbaz tarafından tanınan tüm veri dosyalarını gösteren bir dosya seçim kutusu görüntüleyerek başlar. Kullanıcı daha sonra içe aktarılacak verileri içeren dosyayı seçer ve Aç'ı tıklatır. İçe Aktarma Sihirbazı dosyayı açar ve verilerin bir kısmını önizleme kutusunda görüntüler, böylece kullanıcı verilerin doğru seçim olduğunu doğrulayabilir. İçe Aktarma Sihirbazı verileri işlemeye çalışır ve sihirbaz başarılı olursa, verilerin bir kısmıyla oluşturduğu değişkenleri görüntüler. Kullanıcı ileriye tıklar ve sihirbaz kullanılan Sütun Ayırıcıyı gösterir. Değişken doğru verilere sahipse, kullanıcı sihirbaza devam edebilir (ileriye tıklayın); aksi takdirde kullanıcı farklı bir Sütun Ayırıcı seçebilir. Bir sonraki pencerede sihirbaz, MATLAB'da oluşturulacak değişkenin adını ve boyutunu gösterir. (Veriler sayısal olduğunda, MATLAB'daki değişken, verilerin içe aktarıldığı dosyayla aynı ada sahiptir.) Sihirbaz sona erdiğinde (bitişe tıklayın), veriler MATLAB'a aktarılır.

Örnek olarak, İçe Aktarma Sihirbazı bir .txt dosyasına kaydedilen sayısal ASCII verilerini almak için kullanılır. TestData2 dosya adıyla kaydedilen veriler, Şekil'de gösterilmiştir.

****

TestData2 dosyası için içe aktarma işlemi sırasında İçe Aktarma Sihirbazının görüntüsü Şekil lerde gösterilmektedir. Aşağıdaki şekil, MATLAB'deki değişkenin adının TestData2 ve boyutunun 3x5 olduğunu gösterir.





MATLAB'ın Komut Penceresinde, içe aktarılan veriler değişkenin adı yazılarak görüntülenebilir.

>> TestData2

TestData2 =

5.1200 33.0000 22.0000 13.0000 4.0000

4.0000 92.0000 0 1.0000 7.5000

12.0000 5.0000 6.5300 15.0000 3.0000