

### 3.3 Karşılaştırma İşlemleri

MATLAB programlama dilinde karşılaştırma amacı ile kullanılan işlemler aşağıdaki çizelgede verilmiştir.

Çizelge 3.1 MATLAB karşılaştırma işlemleri

Matematik ifade	MATLAB Karşılığı	Örnek	Açıklama
>	>	A>B	Büyüktür
<	<	A<5	Küçüktür
≥	>=	F(5) >= 2.7	Büyük-eşit
≤	<=	G(:,1) <= 0	Küçük-eşit
=	==	K(5,:) == 1	Eşittir
≠	~=	Y~=0	Eşit değil
"and"	&	N<0 & N>-5	ve
"or"		A=='e'   A=='E'	veya

Karar verme süreçleri, veri ayıklama, dizeler içinde arama gibi uygulamalarda da sıkça karşılaşılan işlemler yukarıda verilen işlemler ile kolayca yürütülebilmektedir. Çizelge 3.7'de verilen tüm örnekler birer mantıksal önermedir. Önermelerin sonucu 1 veya 0 olabilir. Önerme doğru ise 1, yanlış ise 0 değerini alan mantıksal (logical) sonuçlar üretilir. Bu sonuçlar karar vermede ve dizelerinde kullanılabilmektedir. Tüm karşılaştırma işlemlerinde karşılaştırılan dizeler ya aynı büyüklükte olmalı ya da karşılaştırılanlardan biri bir skaler olmalıdır. Karşılaştırma her bir dizeler elemanı için ayrı ayrı yapılır ve önermenin doğru ya da yanlış olmasına göre ilgili eleman için bir mantıksal değer atanır. Örneğin 5x5 boyutlarındaki D dizelerinin elemanlarının sıfırdan büyük olup olmadıkları yönünde bir karşılaştırma yapmak için:

```
>> D=10*randn(5,5)
D =
    0.8593   -6.1560  -14.0227   14.1931    6.9662
   -14.9159    7.4808  -14.2238    2.9158    8.3509
    -7.4230   -1.9242    4.8819    1.9781   -2.4372
   -10.6158    8.8861   -1.7738   15.8770    2.1567
    23.5046   -7.6485   -1.9605   -8.0447  -11.6584

>> L=D<0
L =
     0     1     1     0     0
     1     0     1     0     0
     1     1     0     0     1
     1     0     1     0     0
     0     1     1     1     1
```

şeklindeki MATLAB ifadeleri yeterli olur. L mantıksal bir değişken olarak bellekte saklanır ve yalnızca önermeyi doğrulayan elemanlarda 1 değeri bulunmaktadır. Aynı dizey içinde değeri [-5,5] aralığında olanlar için bir mantıksal karşılaştırma yapılmak istenirse

```
>> L1=D>=-5&D<=5
L1 =
     1     0     0     0     0
     0     0     0     1     0
     0     1     1     1     1
     0     0     1     0     1
     0     0     1     0     0
```

sonucu üretilmiş olur. L1 dizeyi içerisinde değeri 1 olan dizey elemanlarının bu önermeyi doğruladığı kolayca görülebilir. Aynı boyutta iki dizeyi karşılaştırmak için ise aşağıdaki örnekteki yazılışlar kullanılabilir:

```
>> s1=[1 2 -1 0;3 1 5 7;0 2 6 8]
s1 =
     1     2    -1     0
     3     1     5     7
     0     2     6     8
>> s2=[1 0 3 0;4 6 5 -1;1 2 0 7]
s2 =
     1     0     3     0
     4     6     5    -1
     1     2     0     7
>> s1==s2
ans =
     1     0     0     1
     0     0     1     0
     0     1     0     0
```

son adımda üretilen mantıksal değışkende değeri 1 olan konumlarda s1 ve s2 dizyelerinin elemanlarının birbirlerine eşit oldukları görülebilir. Örnekler benzer şekilde diğer işleçler için çoğaltılabilir. Bunlar bölüm sonundaki alıştırmalarda verilecektir.

Karşılaştırma sonucunda üretilen mantıksal değerler, dizyelerin indekslenmesinde kullanılabilir. Dizyelerin belirli koşulları sağlayan elemanları bu sayede başkaca bir program parçası yazmaya gerek olmadan ayıklanabilmektedir. Örneğin 1x6 boyutlarındaki p dizyesinin negatif elemanlarını görmek için

```
>> p=[1 3.4 -1 0 -2.7 4]
p =
Columns 1 through 4
     1.00     3.40    -1.00     0.00
     -2.70     4.00
>> p(p<0)
ans =
```

```
-1.00 -2.70
```

ifadeleri yazılabilir. Karşılaştırma işlemi sayısal olmayan değişken içerikleri için de yapılmaktadır.

Yukarıda sayılan karşılaştırma işleçleri dışında dizey içi arama ve ayıklama işlemleri için kullanılan **find** MATLAB fonksiyonuna özel olarak yer ayırmak gerekir. Fonksiyon çeşitli seçenekler girilerek değişik amaçlar için kullanılabilir. Bu seçenekleri görmek için komut satırında **doc find** yazmak yeterlidir. Burada fonksiyonun jeofizik veri işlem, grafikleme ve diğer analiz işlemlerinde sık kullanıldığı şekli ile ilgili örnekler verilecektir. Aşağıdaki örnekte 1x5 boyutlarındaki **f** dizeyinin sıfırdan küçük elemanlarının yerlerinin bulunması gösterilmiştir. **find** fonksiyonunun içerisine önceki bölümde anlatılan mantıksal bir işleç yazılabilir.

```
>>f=[-2 4 0 30 -6];
>>[yer]=find(f<0)
yer =
     1     5
```

Verilen örnekte **f** dizeyinin sıfırdan küçük elemanlarının bulunduğu sütun numaraları **yer** isimli değişkene aktarılmaktadır. 1 ve 5 numaralı elemanların koşulu sağladığı görülmektedir. İki boyutlu ise satır sütun numaraları belirlenebilir. Örneğin

```
[sat,sut]=find(Lat>=100&Lat<500);
```

yazılışı ile **Lat** isimli değişkenin elemanlarından verilen koşulları sağlayanların bulunduğu satır ve sütun numaraları sırası ile **sat** ve **sut** isimli değişkenlerde saklanır. Verilen şartı sağlayan eleman bulunmaması durumunda **sat** ve **sut** değişkenleri boş dizeydir (empty matrix). Birden çok koşul **&** ve **|** işleçleri ile birleştirilerek **find** fonksiyonu ile birlikte kullanılabilir. Aşağıdaki örnekte **A** dizeyinin birinci sütununda değeri 5 olan ve ikinci sütunda değeri 0.5'den daha küçük olan elemanların bulunduğu satır numaraları bulunabilir:

```
[bul]=find(A(:,1)==5&A(:,2)<0.5);
```

**find** fonksiyonu mantıksal işleçlerde olduğu gibi dizey indeklemeye de kullanılabilir.

```
>> h=randn(4,6)
h =
     0.5377     0.3188     3.5784     0.7254    -0.1241     0.6715
     1.8339    -1.3077     2.7694    -0.0631     1.4897    -1.2075
    -2.2588    -0.4336    -1.3499     0.7147     1.4090     0.7172
     0.8622     0.3426     3.0349    -0.2050     1.4172     1.6302
>> h(find(h>-0.25&h<0.25))
```

```
ans =
-0.0631
-0.2050
-0.1241
```

### Alıştırmalar

1.  $n \times l$  boyutlarındaki  $s$  dizeyinin sıfırdan farklı elemanlarının sayısını bulan MATLAB ifadelerini yazınız.
2. T1 ve T2 dizeyleri aynı boyutlardadır. İki dizeyin eşit olan elemanlarına 0 değeri atayan MATLAB ifadelerini yazınız.
3. İçeriği aşağıdaki gibi olan  $m12$  dizeyinde 0.07 değerinin tekrarlandığı yerleri bulan MATLAB ifadelerini yazınız.

M12 =

```
0.25  0.32  -0.42  1.03  -0.21  -0.44  1.14  0.18  2.00  -0.36  1.03
0.07  -1.34  -0.14  -0.35  -0.87  0.07  0.53  0.97  0.95  0.07  1.46
-0.61  -1.03  0.90  1.01  -1.04  0.98  0.97  -0.41  -0.43  1.42  0.05
-1.22  0.07  -0.30  0.53  -0.27  -0.30  -0.52  -0.44  0.65  -1.60  0.07
```

4.  $f(x) = \begin{cases} x^2, & x < 0 \\ 1, & x = 0 \\ \sqrt{x}, & x > 0 \end{cases}$  şeklinde parçalı olarak tanımlanan fonksiyonu aşağıdaki  $x$  değerleri

için hesaplayınız. Hesaplamalarda  $x$  ile aynı boyutta olan  $f$  dizeyinin içeriğini mantıksal indeksleme ile doldurunuz.

```
x=[-6.1687  -0.7989  -11.4068  -12.8326
    2.7484   8.9848  -10.9334  -23.2895
    6.0110   1.8370   -4.3361   9.0193
    0.9231   2.9079   -1.6847  -18.3564
   17.2984   1.1294   -2.1853   0.6676
   -6.0856   4.3995   0.0000   0.3548
   -7.3706   1.0166   3.8927  22.2717
  -17.4988  27.8734   7.5123  -0.6921
    9.1048  -11.6667  17.7826  -5.0732
    8.6708  -18.5430  12.2306   2.3581];
```

### 3.4 IF Yapısı

IF, bilgisayar programlama dillerinin birinci derece karar verme amaçlı kullanılan yapısıdır. Farklı dillerde yazım olarak değişse de yapısal olarak büyük benzerlikler göstermektedir. IF yapılan bir ya da daha çok önermeyi kontrol ederek ve sonucun doğru çıkması durumunda bir grup ifadeyi çalıştırmaktadır. Genel yazım biçimi MATLAB dili için

```
if <koşul 1>
    <işlemler 1>
elseif <koşul 2>
    <işlemler 2>
else
    <işlemler>
end
```

şeklinde dir. Burada koşul yerine önceki bölümlerde anlatıldığı gibi bir karşılaştırma ya da mantık işleci kullanılarak bir önerme yazılır. Yazılan önerme doğru ise hemen altındaki satırdan başlayan işlemler gerçekleştirilir. Doğru değil ise sonraki kontrol deyimine kadar atlanır. Yukarıdaki yazımda `elseif` diğer bir koşulda yapılacaklar `else` ise sayılan koşullardan hiçbiri sağlanmadığında yapılacakları denetlemektedir. Tüm `if` yapıları bir `end` kelimesi ile sona erer. `elseif` ve `else` yapıları kullanılmak zorunda değildir. Karar verme adımının niteliğine bağlı olarak kullanımı düşünülebilir. Bu durumda en basit hali ile `if` yapısı

```
if <koşul>
    <işlemler>
end
```

halini alır. Birden çok koşul `&` ve `|` işleçleri ile birleştirilebilmektedir. Buna ek olarak birden çok `if` yapısı birbirini kesmeyecek şekilde iç içe kullanılabilir. Aşağıdaki `if` yapılarının kullanımı ile ilgili iki MATLAB programı verilmiştir.

```
% Bu program katsayıları verilen ikinci dereceden bir polinomun köklerini hesaplar.
a=2;b=1;c=-11;
delta=b^2-4*a*c;
if delta<0
    disp('Gerçek kök yoktur')
elseif delta==0
    disp('Birbirine eşit iki kök vardır')
    x1_2=-b/(2*a);
else
    disp('İki farklı kök vardır')
    x1=(-b+sqrt(delta))/(2*a);
    x2=(-b-sqrt(delta))/(2*a);
end
```

Yukarıda verilen örnekte ikinci dereceden bir polinomun katsayıları diskriminant (delta) değerine bağlı olarak `if` yapısı ile denetlenerek hesaplanmaktadır. Aşağıdaki örnekte ise komut satırından girilen bir `x` değeri için parçalı olarak tanımlanmış `f` fonksiyonunun değeri hesaplanmaktadır. Buradaki koşullarda

---

x değişkeninin aldığı değer hangi aralığa düştüğü belirlenerek sağlanan şartın devamındaki satır işletilmektedir. Program içerisinde daha önce değinilmeyen **input** deyimi komut satırından bir mesaj görüntülenerek bir giriş yapılması için kullanılmaktadır. Komut satırından girilen değer eşittir işaretinin solundaki değişkene aktarılmaktadır. **disp** deyimi ise sonrasında gelen parantez içinde yazılı karakter dizisinin komut satırında görünmesini sağlar.

```
% Bu program girilen bir x değeri için parçalı olarak tanımlanmış f(x)
fonksiyonunun alacağı değeri hesaplar
x=input('Bir x değeri giriniz : ');
if x<-3
    f=1;
elseif x>=-3&x<0
    f=exp(abs(x));
elseif x>=0&x<3
    f=exp(x);
else
    f=0;
end
f
```