

MATLAB' TA DÖNGÜLER VE ŞARTLI DEYİMLER

Bilgisayar programlama dilleri ve programlanabilir hesap makineleri karar yürütme esasına dayanan komut icrası akışının denetimine olanak tanır. Programlama dillerinde çoğunlukla, bu akış denetimini sağlayan fonksiyonlar şartlı deyimler ve döngü komutları olarak bilinir.

MATLAB; FORTRAN, C, BASIC gibi yüksek seviyeden programlama dillerindekine benzer komut icrası akışının denetimine yarayan fonksiyonlara sahiptir. Bu fonksiyon deyimleri MATLAB'ı basit bir hesap işlemcisiinden ziyade yüksek seviyeli bir matris dilli haline sokar. Şartlı deyimler ve döngüler, geçmişte yapılan hesaplamaların etkisini gelecek işlemlere aktarabilen çok güçlü bir programlama özelliğidir. MATLAB karar yürütme veya akış denetimi yapıları altında sunulan bu özellikler; for döngüleri, while döngüleri, if-else-end şartlı deyimleri ve switch-case şartlı deyimleridir.

if Şartlı Deyimi

if mantıksal deyimi değerlendirir ve bu deyimin değerini esas alan bir grup ifadeyi yürütür. En basit biçimdeki sözdizimi;

```
if mantıksal deyim
    komut ifadeleri
end
```

Eğer mantıksal deyim doğru (1) ise, MATLAB if ve end satırları arasında kalan tüm komut ifadelerini yürütür. End komutunu izleyen satırda yürütmeyi sürdürür. Eğer koşul yanlış (0) ise MATLAB if ve end satırları arasında kalan tüm komut ifadelerini atlayıp geçer ve end komut satırından sonraki satırda yürütmeye devam eder.

Basit bir if deyimi örneği;

```
if a < 50
    say=say+1;
    top=top+a;
end
```

Burada a' nın bir skalar olduğunu varsayarsak a < 50 olması halinde say her seferinde 1 sayısı kadar artırılır ve a da top' a ilave edilir. Aksi taktirde ise, if yapısı arasında kalan bu komut atlanır. Eğer a bir skalar değilse say yalnızca a' nın 50 den küçük olması halinde 1 kere artırılır ve a da top' a ilave edilir.

else ve elseif Şartlı Deyimleri

if deyiminin daha fazla koşullara bağlanmasını sağlar. else deyimi hiçbir mantık koşuluna sahip değildir. Eğer kendinden önceki if (veya olası elseif koşulu) koşulu yanlış (0) ise else deyimi ile ilişkili ifadeler yürütülür. Bunun genel çatısı;

```
if deyim
    komut ifadeleri % if doğru ise değerlendirilecek
else
    komut ifadeleri % if yanlış ise değerlendirilecek
end
```

biçiminde kurulabilir. elseif deyimi bir mantıksal koşula sahip olup, eğer kendisinden önceki if (ve

olası elseif koşulu) koşulu yanlış (0) ise değerlendirme yapar. Eğer elseif`ın mantıksal koşulu doğru (1) ise, onunla ilişki ifadeleri yürütülür. Bir if bloğu içinde birden fazla elseif kullanılabilir.

switch-case Yapıları

switch-case yapıları bir değişken veya deyimın değerine dayanan belli ifadeleri yürütür. Temel kullanım yapısı;

switch deyim (skalar veya karakter)

case değer1

komut ifadeleri

% eğer deyim değeri1 ise yürütülür

case deger2

komut ifadeleri

% eğer deyim değeri2 ise yürütülür

.

otherwise

komut ifadeleri

% eğer deyim hiçbir case ile denk düşmezse yürütülür

end

biçiminde verilir. Bu program aşağıda verilen kısımları içerir.

- Switch kelimesini değerlendirecek deyim izler
- Case grupları herhangi bir sayıdadır. Bu gruplar, hepsi tek bir satır üzerinde olmak üzere deyim için olası bir değerın izlediği case kelimesinden ibarettir. Bir case grubu MATLAB bir sonraki case ifadesi veya otherwise ifadesi ile karşılaşınca sona erer.
- Bir seçmeli otherwise vardır. Bu otherwise kelimesinden ibaret olup, bunu önceki herhangi bir case gruplarının ele almadığı komut ifadelerini yürütür. otherwise grubunun yürütülmesi end komut ifadesi ile son bulur.

while Döngüsü

while döngüsü, denetleyen deyim doğru (1) olduğu sürece bir ifadeyi veya ifade grubunu yürütür. Bunun söz dizimi aşağıda olduğu gibidir.

while deyim

Komut ifadeleri

end

Eğer deyim bir matrisi değerlendiriyorsa, yürütmenin devam edebilmesi için matrisin tüm elemanları 1 olmalıdır. Bir matrisi bir skalar değere indirgemek için all ve any fonksiyonları kullanır. Aşağıda while döngüsü, 100 haneli bir n! için ilk tam sayı n`i bulur.

n=1;

while prod(1:n) < 1e100

n=n+1;

end

break fonksiyon ifadesi kullanarak while döngüsünden herhangi bir anda çıkılabilir.

for Döngüsü

for bir komut ifadesi veya komut ifadesi grubu önceden saptanan bir sayı kere yürütülür. Bunun sözdizimi aşağıda verilmiştir.

```
for indeks=başlangıç:artış:son
    Komut ifadeleri
end
```

varsayılan artış 1dir. Negatif sayıda dahil olmak üzere herhangi bir artış belirlenebilir. Aşağıdaki döngü 5 kez yürütülür.

```
for i= 2:6
    x(i)=2*x(i-1 );
end
```

break ifadesi

break ifadesi bir for döngüsü veya while döngülerinin yürütülmesini sona erdirir. break ifadesi ile karşılaşıldığında yürütme döngünün dışındaki bir komut ifadesinde devam eder. İç içe döngülerde, break yalnızca en içteki döngüden çıkar. Aşağıda verilen örnek while döngüsünün bir fft.m dosyası içeriğinin bir MATLAB karakter dizimine okuyuşunu göstermektedir. İlk boş satır ile karşılaştığında while döngüsünden çıkmak için break ifadesi kullanılmıştır. Fft.m programı için sonuç M-dosyası yardımı içeren karakter dizimidir.

```
fid= fopen ('fft.m','r' );
s = ' ^';
while feof (fid )
    line= fgetl (fid );
    if isempty (line ), break, end
    s = strvcats (s,line );
end
disp (s )
```

return Deyimi

return komutların halihazır sıralanmasını sona erdirir ve uyarılan fonksiyonu veya klavyeyi denetime geri döndürür. return ayrıca klavye modunu sona erdirmek içinde kullanılır. Fonksiyonun sonuna ulaştığında, çağırılan fonksiyon normal olarak denetimi uyarılan fonksiyona aktarılır. return erken sona erdirme ve denetimi uyarılan fonksiyona aktarmak için çağırılan fonksiyon içine sokulabilir.

Continue Deyimi

Continue komutu, for veya while döngüsünde kontrolü bir sonraki yinelemeye geçirir. Dolayısıyla continue deyiminden sonraki komutlar işleme tabii tutulmaz. Aşağıda verilen örnekte continue deyimi içeren bir döngü(loop) verilmiştir. Continue komutu dosyadan okunan satırların boş satır olması halinde bir sonraki satırın okumasını sağlamakta ve okunan dosyadaki boş olmayan satırların sayısının elde edilmesini sağlamaktadır.

```
count = 0;
```

```

while ~feof(fid)
    line = fgetl(fid);
    if isempty(line)
        continue
    end
    count = count + 1;
end
disp(sprintf('%d lines', count));

```

Gauss Eleminasyon Yöntemi ile Doğrusal Denklem Sistemi Çözümü:

Kaynak Kod:

```

function [x]=gauselm(a,b)
% program Gauss eliminasyon yöntemi ile Ax=b yapısında doğrusal
% denklem sistemi için x bilinmeyenleri hesaplar.
% not: pivotlama işlemi yapılmaz
% Ü. Dikmen, 1999
%
[row,col]=size(a);
g=a;
g(:,col+1)=b;
for r=1:row
    g(r,:)=g(r,:)/g(r,r);
    for k=1:row
        if ne(k,r)
            g(k,:)=g(k,:)-g(r,:)*g(k,r);
        end;
    end;
end;
x=g(:,end)';

```

Örnek: Verilen örnek kod kullanılarak a katsayı dizeyi ve b yöneyi MATLAB' da çalışma ortamına girilirse,

```

>> a =
    2   -3    2
    1    1   -2
    3   -2   -1

```

```

>> b =
   -11
     8
    -1

```

```

>> x= gauselm(a,b)

```

```

x=

```

```

1.0000  3.0000 -2.0000

```

Örnek:

```
>>a =
-0.0400  0.0400  0.1200
 0.5600 -1.5600  0.3200
-0.2400  1.2400 -0.2800
```

```
>>b =
 3
 1
 0
```

```
>> x= gauselm(a,b)
```

```
>>x
```

```
7.0000  7.0000 25.0000
```

Gauss-Jordan Yöntemi ile Doğrusal Denklem Sistemi Çözümü:

Öğrendiklerimizden hatırlayacağımız gibi, Gauss eliminasyon yönteminde, katsayı dizeyi bir üst üçgen dizey yapısına dönüştürülürken, Gauss-Jordan yönteminde ise Gauss eliminasyon yönteminde izlenen işlem adımları uygulanmasıyla, katsayı dizeyi birim dizeye dönüştürülür. Bunun için MATLAB programlama dilinde bir dizeyi birim dizeye dönüştürebilen

```
[r,jr] = rref(a)
```

fonksiyonundan yararlanılır. rref() fonksiyonu Gauss-Jordan yöntemi kullanarak bir A dizeyini birim dizeye kısmi pivotlama yaparak dönüştürür. $Ax=b$ doğrusal denklem sisteminin çözümünde kullanmak üzere, aşağıdaki gösterilen işlem yapılmalıdır:

```
a(:,end+1)=b ;
```

bu şekilde, katsayı dizeyinin son kolonuna ilave bir kolon açılmakta ve b yöneyi bu kolona aktarılmaktadır.

Örnek:

```
>>a =
-0.0400  0.0400  0.1200
 0.5600 -1.5600  0.3200
-0.2400  1.2400 -0.2800
```

```
>>b =
```

3
1
0

```
>> a(:,end +1)=b ;
```

```
>> [r,jr]=rref(a)
```

```
r =
```

```
1  0  0  7
0  1  0  7
0  0  1 25
```

```
jr =
```

```
1  2  3
```

Dikkat edilirse r dizeyinin 4. Kolonu bizim için bilinmeyenlerin yer aldığı kolonu gösterir.

Ayrıştırma(Cholesky,Lu) Yöntemi ile Doğrusal Denklem Sistemi Çözümü:

$Ax=b$ şeklinde bir doğrusal denklem sistemini, ayrıştırma yöntemi ile çözmek istediğimizde MATLAB'ın

```
[l,u,p] = lu(a);
```

hazır fonksiyonundan yararlanılır. Burada l, alt üçgen dizey, u, üst üçgen dizey ve p, permutasyon dizeyidir. Böylece, $l*u = p*a$ sağlanır. Şimdi bir önceki örnekte vermiş olduğumuz örneği bu kez ayrıştırma yöntemi ile çözelim:

Örnek:

```
>> a
```

```
a =
```

```
-0.0400  0.0400  0.1200
 0.5600 -1.5600  0.3200
-0.2400  1.2400 -0.2800
```

```
>> b
```

```
b =
```

```
3
1
0
```

```
>> [l,u]=lu(a)
```

```
l =
```

```
-0.0714 -0.1250  1.0000
 1.0000      0      0
-0.4286  1.0000      0
```

```
u =  
  0.5600 -1.5600  0.3200  
  0      0.5714 -0.1429  
  0      0      0.1250
```

```
>> z=inv(l)*b
```

```
z =  
  1.0000  
  0.4286  
  3.1250
```

```
>> x=inv(u)*z
```

```
x =  
  7.0000  
  7.0000  
 25.0000
```