

### Problem-1.7. Verilen Bir Sayının Karekökünü Hesaplama

Acaba hesap makinesi ya da bilgisayar bir sayının karekökünü nasıl hesaplıyor?  $x_0$  başlangıç değeri olmak üzere

$$x_{n+1} = \frac{1}{2} \left( x_n + \frac{b}{x_n} \right), n = 0, 1, 2, \dots \text{ dizisi } \sqrt{b} \text{ değerine yakınsar}$$

Bu yineleme bağıntısının nereden geldiği daha sonra açıklığa kavuşacaktır.

$$x_1 = 0.5 \left( x_0 + \frac{b}{x_0} \right)$$

$$x_2 = 0.5 \left( x_1 + \frac{b}{x_1} \right)$$

.

.

.

$$x_n = 0.5 \left( x_{n-1} + \frac{b}{x_{n-1}} \right)$$

Yani,  $x_0, x_1, x_2, \dots, x_n \rightarrow \sqrt{b}$  dır. Burada sorun  $x_0$  'ın nasıl belirleneceği ve algoritmanın durdurma kuralının ne olacağıdır. Aslında bir sayının karekökünün bulunması problemi,  $f(x) = x^2 - b$  fonksiyonunun kökünü bulmak demektir. Bu şekilde düşününce, hem fonksiyonun belli bir aralıkta kökünün olup olmadığı belirlenebilir, hem de başlangıç değeri seçilebilir.

#### Teorem (Bolzano Teoremi):

$f(x)$ ,  $[a, b]$  aralığında sürekli bir fonksiyon olmak üzere, eğer  $f(a).f(b) < 0$  ise  $\exists x^*$  vardır öyle ki  $f(x^*) = 0$  olur (Balcı 2008).

$f(x) = x^2 - 24$  fonksiyonu için, Bolzano teoremine göre  $f(4).f(5) < 0$  olduğundan fonksiyonun bu aralıkta bir kökü mevcuttur ve  $x_0$ ,  $[4, 5]$  aralığından herhangi bir değer olarak seçilebilir.  $x_0 = 4$  alınırsa,

$$x_1 = 0.5 \left( 4 + \frac{24}{4} \right) = 5$$

$$x_2 = 0.5 \left( 5 + \frac{24}{5} \right) = 4.9$$

$$x_3 = 0.5 \left( 4.9 + \frac{24}{4.9} \right) = 4.89897959183673$$

$$x_4 = 0.5 \left( 4.89897959183673 + \frac{24}{4.89897959183673} \right) = 4.89897948556636$$

olur. Aslında kökü içeren aralıktan başlangıç değeri seçilmesi algoritmanın hesaplama süresini değiştirir. Genel olarak aşağıdaki şekilde durdurma kuralları belirlenebilir.

- 1) Belirli adım sayısında işlem yapılır.
- 2)  $|f(x_n)| < \varepsilon$  incelenir.
- 3)  $|x_n - x_{n-1}| < \delta$  incelenir.

Belirli bir adım sayısında işlem yapıldığında yakınsamanın durumu hakkında her zaman tam olarak bir şey söylenemez. Farklı başlangıç değerlerine göre yakınsama değişir. 2. durdurma kuralı kullanılırsa fonksiyonun sıfıra ne kadar yakın olup olmadığı kontrol edilmiş olunur. 3. durdurma kuralında  $n$ . adımda bulunan yaklaşık değerle  $n+1$ . adımda bulunan değer birbirine ne kadar yakın olup olmadığına bakılmış olunur. Çok büyük adım sayısı verilerek ve diğer iki durdurma kuralı da kullanılarak algoritmanın her zaman durması ve yakınsamanın gerçekleşmesi sağlanabilir.

a) Durdurma kuralı belirli bir adım sayısı olsun. Buna göre algoritma,

**A1.**  $n$ ,  $x_0$  ve  $b$  sayılarını OKU/GİR

**A2.**  $say=0$  al

**A3.**  $x_1=0.5*(x_0+b/x_0)$  al

**A4.**  $say=say+1$  al

**A5.** YAZ  $say$ ,  $x_1$

**A5.** Eğer  $say \geq n$  ise Dur

**A6.**  $x_0=x_1$  al ve A3' e GİR

'Bir sayının karekökünü bulmak için program

INPUT "karekökü alınacak sayıyı giriniz=",b

INPUT "adım sayısı=",n

INPUT "başlangıç değeri=",x0

say=0

A3:

$x_1=0.5*(x_0+b/x_0)$

say=say+1

PRINT "Adım sayısı=";say;"Yaklaşık karekök=",x1

IF say>n THEN END

$x_0=x_1$

GO TO A3

b) Durdurma kuralı verilen bir  $\varepsilon$  değerine göre  $|f(x_n)| < \varepsilon$  kontrolü olsun. Daha önceden ifade edildiği gibi, Basic programlama dilinde verilen bir sayının mutlak değerini almak için ABS deyimi kullanılır. Kullanımı ABS(Sayı) biçimindedir. Buna göre algoritma ve program aşağıdaki gibidir.

**A1.**  $\varepsilon$ ,  $x_0$  ve  $b$  sayılarını OKU/GİR

**A2.**  $x_1=0.5*(x_0+b/x_0)$  al

**A3.**  $say=say+1$  al

**A4.** YAZ  $say$ ,  $x_1$

**A5.** Eğer,  $|x^2 - b| < \epsilon$  ise DUR

**A6.**  $x_0 = x_1$  al ve A2' e GİT

```
INPUT "karekökü alınacak sayıyı giriniz=", b
INPUT "epsilon değeri=", epsilon
INPUT "başlangıç değeri=", x0
A2:
x1 = .5 * (x0 + b / x0)
say=say+1
PRINT "Adım sayısı=";say;"Yaklaşık karekök=",x1
IF ABS(x1 ^ 2 - b) < epsilon THEN END
x0 = x1
GOTO A2
```

c) Durdurma kuralı, verilen bir delta değerine göre  $|x_n - x_{n-1}| < \delta$  kontrolü olsun. Buna göre algoritma ve program aşağıdaki gibidir.

**A1.**  $\delta, x_0$  ve  $b$  sayılarını OKU/GİR

**A2.**  $x_1 = 0.5(x_0 + b/x_0)$  al

**A3.**  $say = say + 1$  al

**A4.** YAZ  $say, x_1$

**A5.** Eğer,  $|x_1 - x_0| < \delta$  ise DUR

**A6.**  $x_0 = x_1$  al ve A2' e GİT

```
INPUT "karekökü alınacak sayıyı giriniz=", b
INPUT "delta değeri=", delta
INPUT "başlangıç değeri=", x0
A2:
x1 = .5 * (x0 + b / x0)
say=say+1
PRINT "Adım sayısı=";say;"Yaklaşık karekök=",x1
IF ABS(x1 - x0) < delta THEN END
x0 = x1
GOTO A2
```

Bolzano teoremini kullanılarak bir fonksiyonun kökünü bulacak başka bir algoritma geliştirilebilir. Ardışık olarak kökün bulunduğu aralığın uzunluğunu yarıya indirilerek kökü içerecek aralık uzunluğunu istenildiği kadar küçük yapacak bir yöntem geliştirebilir. Bu yöntemin işleyişi adım adım aşağıdaki gibidir,

**Adım-1.** Bolzano teoremine göre  $f(a_0).f(b_0) < 0$  olacak şekilde kökü içeren bir  $I_0 = [a_0, b_0], a_0 < b_0$  aralığı bulunur.

**Adım-2.**  $[a_0, b_0]$  aralığı  $\left[ a_0, \frac{a_0 + b_0}{2} \right]$  ve  $\left[ \frac{a_0 + b_0}{2}, b_0 \right]$  aralıklarına parçalanır. Eğer;

$f\left(\frac{a_0 + b_0}{2}\right) = 0 \Rightarrow \frac{a_0 + b_0}{2}$  köktür değilse;

$$f\left(\frac{a_0 + b_0}{2}\right) \neq 0 \Rightarrow \begin{cases} f(a_0).f\left(\frac{a_0 + b_0}{2}\right) < 0 \Rightarrow I_1 = [a_1, b_1] = \left[ a_0, \frac{a_0 + b_0}{2} \right] \\ f(a_0).f\left(\frac{a_0 + b_0}{2}\right) > 0 \Rightarrow I_1 = [a_1, b_1] = \left[ \frac{a_0 + b_0}{2}, b_0 \right] \end{cases}$$

**Adım-3.**  $I_1$  aralığı için Adım-2'deki işlemler yapılır ve  $I_2$  aralığı elde edilir. Buna göre,  $I_0 > I_1 > I_2 > \dots$  olur.

**Adım-4.** İşlemler  $n$ . adımda durmuş ise kökü içeren aralık  $I_n = [a_n, b_n]$  olmak üzere yaklaşık kök  $\alpha = \frac{a_n + b_n}{2}$  alınır. Bu yöntem **Yarılama yöntemi** denir. Program aşağıdaki gibidir.

'Kökü alınacak sayı d

INPUT "a=", a

INPUT "b=", b

INPUT "d=", d

INPUT "n=", n

say = 0

hesap:

fa = a ^ 2 - d

fb = b ^ 2 - d

c = (a + b) / 2

fc = c ^ 2 - d

say = say + 1

IF say > n THEN

    PRINT "c=", c

    END

END IF

IF fa \* fc < 0 THEN

    b = c

ELSE

```
a = c  
END IF  
GOTO hesap
```