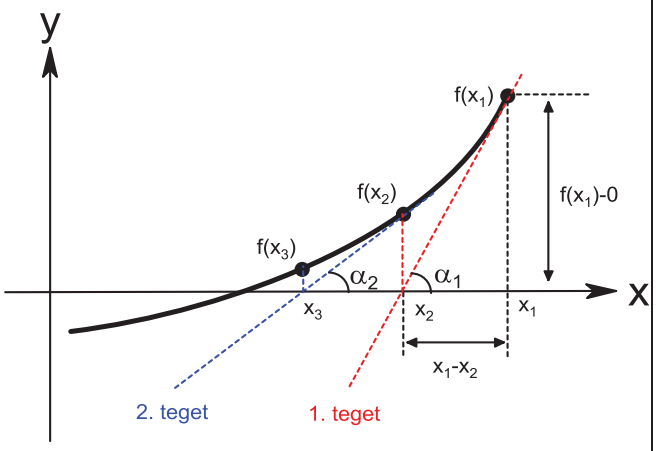
## 4- Newton Raphson Yöntemi




Bu yöntemde köklere teğetler ile yaklaşılır.
Rasgele bir $x_1$ noktası alınır ve bu noktada fonksiyonun teğeti çizilir. Bu teğetin eğimi hesaplanır.

Fonksiyonun o noktadaki teğeti aynı zamanda o noktadaki türevine eşittir. Bu iki eşitlik kullanılarak teğetin x eksenini kestiği $x_2$ noktası bulunur.

Aynı işlemler $x_2$ noktası için tekrar edilir ve $x_3$ noktası bulunur. İşlemlere devam edilirse sonlu adım sonra köke yaklaşılır.

1.teğetin eğimi

$$\tan(\alpha_1) = f'(x_1) = \frac{f(x_1) - 0}{x_1 - x_2}$$

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

2.teğetin eğimi

$$\tan(\alpha_2) = f'(x_2) = \frac{f(x_2) - 0}{x_2 - x_3}$$

$$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)}$$

Genel Kural
(Newton Raphson Fomülü)

$$\Rightarrow \quad x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

$$\Delta x = -\frac{f(x_i)}{f'(x_i)}$$

$$x_{i+1} = x_i + \Delta x$$

Not: Fonksiyonun işaret değiştirip değiştirmediğine bakılmadığı için bu yöntem ile katlı kökler de bulunabilir.

| | | |
|---|---|---|
| ÖRN: | $f(x)=x^3+7x^2-5x-20$ | $X_0$ =8 alarak Newton rapson yöntemi ile kök bulunuz. Kökler: (1.8162, -7.3097 , -1.5065) |
| | | |
| ÖRN: | $f(x)=x^3+7x^2-5x-20$ | $x_1$=-10 ve $x_1$=-4 için |

| | | |
|---|---|---|
| ÖRN: | $f(x)=x^3-5$ | $x_1$=1 ve $x_1$=-5 ($x_0$=1.709976) |

## ALGORİTMA

1- Rastgele bir başlangıç değeri ($x_1$ ) ve hata sınırı belirle ( ε ),

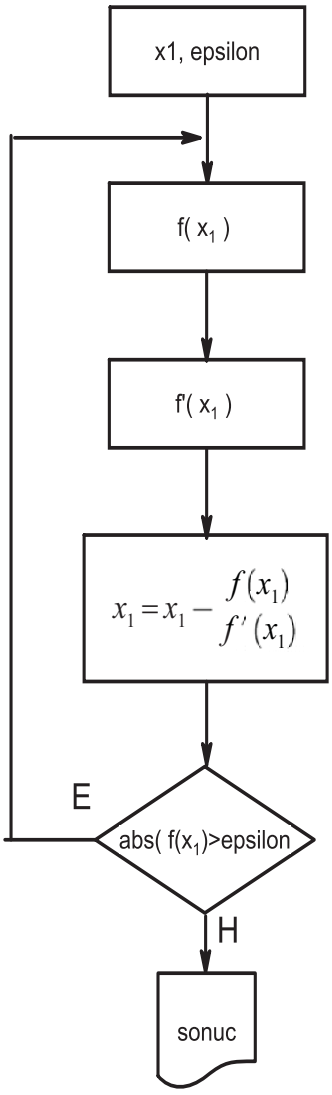
2- $f(x_1)$ ve $f'(x_1)$ 'i hesapla,

3- $x_{i+1} = x_i - \dfrac{f(x_i)}{f'(x_i)}$ değerini hesapla,

4- $abs\left(f(x_{x+1})\right) > \varepsilon \longrightarrow x_i = x_{i+1}$ atamasını yap ve adım 2 'ye atla,

5- Sonuçları yazdır ve çık.

## AKIŞ ŞEMASI




## c# kodları

```
namespace nr
{
   class Program
   {
          public delegate double Function(double x);

      static double F1(double x)
      {return x*x-4*x-10;}

      static double F1_turev(double x)
      {    return 2*x-4;}

      public static double NewtonRaphsonMethod(Function f,
Function fprime,
      double x0, double epsilon)
      {
         double f0 = f(x0);
         double x = x0;
           int i=0;
         while (Math.Abs(f(x)) > epsilon)
         {
            i++;
Console.WriteLine("iterasyon: {0}",i + "   x:" +
x.ToString("0.000000") + "  gercek_turev:" +
F1_turev(x).ToString("0.000000"));
            x -= f0 / fprime(x);
            f0 = f(x);
         }
         return x;
      }

      static void Main(string[] args)
      {
         double epsilon=0.0001, x1=3.0;
Console.WriteLine("\n\nTesting Testing Newton-Raphson
Method\n");
double x = NewtonRaphsonMethod(F1, F1_turev, x1, epsilon);
Console.WriteLine("\n\nNR Sonuc:" + x.ToString());
         Console.WriteLine("NR Test:f(x)=" + F1(x).ToString());
         Console.ReadLine();
      }
   }
}
```

## 4- 1 Sayısal Türevli Newton Raphson Formülü

Newton Raphson formülü içerisinde fonksiyonun 1. Türevine ihtiyaç vardır. Bu türev analitik olarak hesaplanmaktadır. Polinom veya bir çok fonksiyon için 1. türevin bulunması kolay olsada, türevlerinin bulunması zor veya zaman alıcı fonksiyonlar olabilir. Bu durumlarda türevin sayısal hesaplanması gerekir. Sayısal türev hesabı ileriki haftalarda Sayısal türev konusunda ayrıntılı olarak işlenecektir. Burada sadece bir sayısal türev formülüne değinilip geçilecektir.

Genel Kural
(Newton Raphson formülü) $\Rightarrow$ $$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

$f'(x_0) = ?$

Herhangi bir $x_i$ noktasında fonksiyonun türevi; ACD üçgeninden

$$f'(x_i) = \tan\alpha_1 = \frac{|AC|}{|DC|} = \frac{|BC|}{h}$$

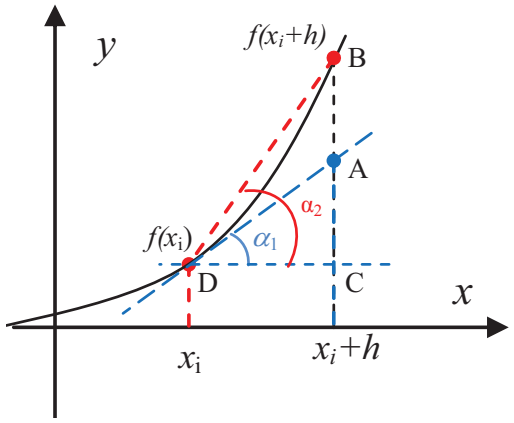
A noktası hesap edilemediğinden yani AC uzunluğu bilinemediğinden bu üçgen yerine BCD üçgeni kullanılarak yaklaşık olarak türev hesaplanır.

BCD üçgeninden

$$\tan\alpha_1 = f'(x_i) \cong \tan\alpha_2 = \frac{|BC|}{|DC|} = \frac{|BC|}{h}$$

$$f'(x_i) = \frac{f(x_i + h) - f(x_i)}{h}$$

Newton formülü içine yazılırsa;




**ÖRNEK**

$f(x) = x^2, f'(2) = ?$

Analitik olarak;

- ✓ $f'(x) = 2x \rightarrow f'(2) = 4$

Sayısal Olarak;h=0.1;

- ✓ $f'(2) = \frac{f(x+h) - f(x)}{h}$
- ✓ $f'(2) = \frac{f(2+0.1) - f(2)}{0.1} = \frac{2.1^2 - 2^2}{0.1} = 4.1$

Sayısal Olarak;h=0.01;

- ✓ $f'(2) = \frac{f(2+0.01) - f(2)}{0.01} = 4.01$

| ÖRN: | $f(x)=x^3-5$ | $x_1$=1 ve h=0.01 ($x_0$=1.709976) |
|---|---|---|

| | |
|---|---|
| SORU | $f(x)=x^2-10Sin(x)-3$ fonksiyonunun bir kökünü ε=0.001 alarak program ile bulunuz. |
| SORU | $f(x)=x^5-2x^4+3x^3-x+5$ fonksiyonunun tüm kökleri ni ε=0.001 alarak program ile bulunuz. |

## ALGORİTMA

1- Rastgele bir başlangıç değeri ($x_1$ ) ve hata sınırı belirle ( ε ),

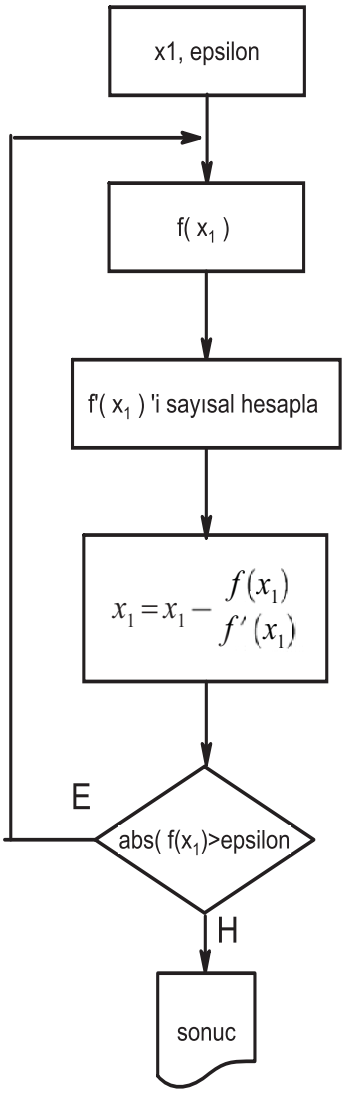
2- f($x_1$) ve $x_1$ noktasındaki sayısal türevi 'i hesapla,

3- $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$ değerini hesapla,

4- $abs\left(f(x_{x+1})\right) > \varepsilon \rightarrow x_i = x_{i+1}$ atamasını yap ve adım 2 'ye atla,

5- Sonuçları yazdır ve çık.

## AKIŞ ŞEMASI



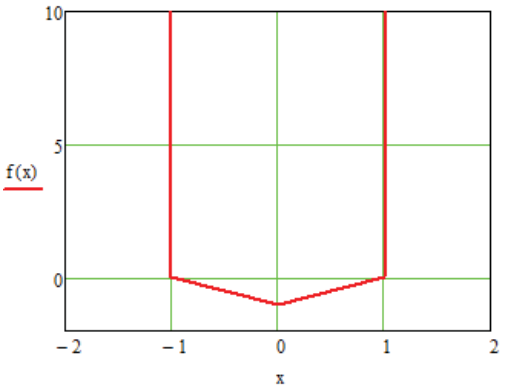

## c# kodları

```
namespace ConsoleApplication1
{
   class Program
   {
      public delegate double Function(double x);

      static double F1(double x)
      {return x * x - 4*x - 10; }

      static double F1_turev(double x)
      {    return 2*x-4;}

      static double F1say_tur(double x) //sayısal turev
      {double h = 0.001;
         return (F1(x + h) - F1(x)) / h; }

      public static double NewtonRaphsonMethod(Function f,
Function fprime, double x0, double epsilon)
      {
         double f0 = f(x0);
         double x = x0;
         while (Math.Abs(f(x)) > epsilon)
         {
            Console.WriteLine("x:" + x.ToString("0.000000") + "
gercek_turev:" + F1say_tur(x).ToString("0.000000") +
"  sayisal_turev:" + F1say_tur(x).ToString("0.000000"));
            x -= f0 / fprime(x);
            f0 = f(x);
         }
         return x;
      }

      static void Main(string[] args)
      {
        double epsilon = 0.0001, x1 = 3.0;
Console.WriteLine("\n\nTesting Testing Newton-Raphson
Method\n");
double x = NewtonRaphsonMethod(F1, F1say_tur, x1, epsilon);
         Console.WriteLine("\n\nNR Sonuc:" + x.ToString());
         Console.WriteLine("NR Test:f(x)=" + F1(x).ToString());
         Console.ReadLine();
      }

   }
}
```

**NR Yönteminin zayıflıkları;**

NR yöntemi çok etkili olmasına rağmen, özellikle katlı kökler ve bazı basit kök aramasında zayıf kalır.

1.
Örnek : $f(x)=x^{10}-1$ fonksiyonun $x_1$=0.5 civarındaki kökünü NR yöntemi ile bulmaya çalışalım;

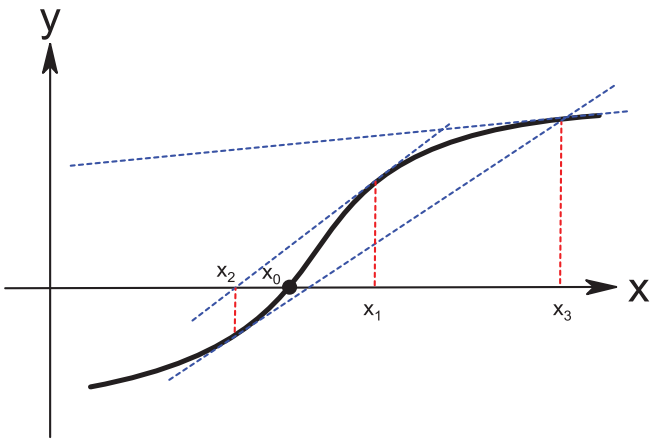



$$f(x)=x^{10}-1 \rightarrow f'(x)=10\cdot x^{9}$$

$$x_{i+1}=x_i-\left(\frac{x_i^{10}-1}{10\cdot x_i^{9}}\right)$$

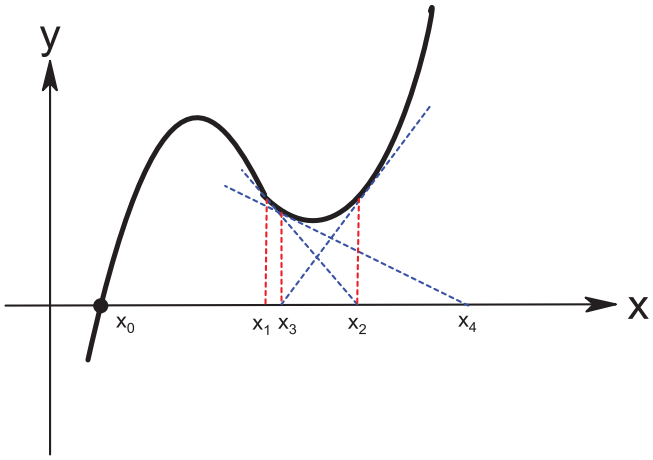
| i | 0 | 1 | 2 | 3 | 4 | 5 | . | . | ∞ |
|---|---|---|---|---|---|---|---|---|---|
| $x_i$ | 0.5 | 51.65 | 46.485 | 41.8365 | 37.6528 | 33.8875 | . | . | 1.00 |

2.
Kötü bir ilk tahminden sonra, köke yakınsama çok yavaştır.




İlk tahmin $x_1$ 'den itibaren itersayon adımları kökten uzaklaşmaktadır.

3.

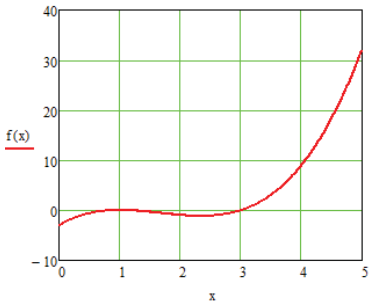



NR yöntemi, yerel max ve min civarında salınma özelliği gösterir.

NR yöntemi için genel bir yakınsama kriteri yoktur. Yakınsama fonksiyonun doğasına ve ilk tahmin değerinin dogruluğuna bağlıdır. Çözüm, köke yeterince yakın ilk başlangıç noktasının seçilmesidir.

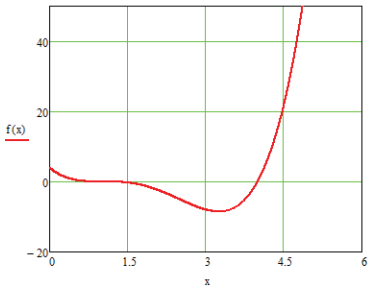
İyi tahminler, fiziksel problemin bilinmesi veya çözümün davranışı hakkında bilgi ve grafikler ile bulunabilir.

4.
Katlı kökler




$$f(x) = (x-3)\cdot(x-1)\cdot(x-1)$$

Çift katlı kökler ekseni kesmez.




$$f(x) = (x-4)\cdot(x-1)\cdot(x-1)\cdot(x-$$

Üç katlı kök eksene teğettir ve ekseni keser.

Çift katlı köklerde; f(x) işaret değiştirmediğinden kapalı yöntemler kullanılamaz ayrıca çift ve üç katlı köklerde, kök civarında f(x) ve f'(x) sıfıra çok yakın olduğundan yuvarlama hataları oluşur.

Bu sorunları çözmek için Geliştirilmiş NR yöntemi kullanılır.

## 4.2 GELİŞTİRİLMİŞ NEWTON RAPHSON YÖNTEMİ

f(x)' in kökünü bulmak için yardımcı bir fonksiyon tanımlanır (G(x)). Bu fonksiyonun kökü f(x) ile aynı olmalıdır.

Yardımcı fonksiyon;

$$G(x)=\frac{f(x)}{f'(x)}$$

f(x) = 0 → G(x) = 0 'dır.

Olarak tanımlanır.

G(x)=0 yapacak x=a değeri f(x) 'in de köküdür.

$$x=a \rightarrow f(a)=0 \Rightarrow G(a)=0$$

Bu yöntemle fonksiyon katlı kökten kurtulmuş olur.

Bu sebeple f(x) 'in yerine G(x)' in kökü Newton Raphson Formülü ile bulunur.

ÖRN $f(x)=x^3-2x^2-5$

$$G(x)=\frac{f(x)}{f'(x)}=\frac{x^3-2x^2-5}{3x^2-4x}$$

ÖRN $f(x)=(x-2)^3$

$$G(x)=\frac{f(x)}{f'(x)}=\frac{(x-2)^3}{2\cdot(x-2)^2}=2\cdot(x-2)$$

### 4.2.1 G(x) Fonksiyonuna Newton Raphson Formülünün Uygulanması

$$G(x)=\frac{f(x)}{f'(x)} \quad ve \quad x_{i+1}=x_i-\frac{G(x_i)}{G'(x_i)}$$

$$G'(x)=\frac{d}{dx}\left[\frac{f(x)}{f'(x)}\right]=\frac{\left[f'(x)\right]^2-f''(x)\cdot f(x)}{\left[f'(x)\right]^2}$$

$$x_{i+1}=x_i-\frac{\frac{f(x_i)}{f'(x_i)}}{\frac{\left[f'(x_i)\right]^2-f''(x_i)\cdot f(x_i)}{\left[f'(x_i)\right]^2}}$$

$$x_{i+1}=x_i-\frac{f(x_i)\cdot f'(x_i)}{\left[f'(x_i)\right]^2-f''(x_i)\cdot f(x_i)}$$

Not: Burada fonksiyonun ikinci türevinin de hesabı gerekmektedir. Şimdilik fonksiyonun 2. Türevi analitik olarak hesaplanarak kullanılacaktır. Sonraki bölümlerde sayısal türev konusunda 2. Türev hesabı için formüller verildiğinde onlar kullanılacaktır.

Algoritma, ve program N-R ile aynıdır.

| ÖRN | $f(x)=(x-3)\cdot(x-1)\cdot(x-1)=x^3-5x^2+7x-3$ fonksiyonun $x_1=0$ , $x_1=4$ civarındaki kökünü NR ve GNR hesaplayınız. |
|---|---|

| | |
|---|---|
| Not: Başlangıç değeri kompleks sayı girilirse kompleks kökler de bulunabilir. | |
| ÖRN | $f(x)=x^2+x+1$<br>x=i noktası civarındaki kökünü bulunuz. $\left(x_{1,2}=-\frac{1}{2}\pm\frac{\sqrt{3}\,i}{2}\right)$ |

## 4- Basit iterasyon Yöntemi

Bu yöntemde f(x)=0 denklemi x=F(x) formuna getirilir. Eğer yakınsama koşulunu sağlıyorsa;

$x_{i+1} = F(x_i)$ formülü ile iterasyona başlanır.
$|x_i - x_{i+1}| < \varepsilon$ oluncaya kadar işleme devam et.

**Yakınsama koşulu;**
[a , b] deki bütün x' ler için $(\forall x \in [a,b])$ için $|F'(x)| < 1$ sağlanmalıdır.

ÖRN: $f(x) = x^2 - 3x + 1$ denkleminin bir kökünü [0, 1 ] aralığında basit iterasyon yöntemi ile 3 adımda bulunuz.

ÖDEV: $f(x) = x^2 + 2x - 2 = 0$ denkleminin bir kökünü [0, 1 ] aralığında basit iterasyon yöntemi ile 3 adımda bulunuz.

## ALGORİTMA

1- Rastgele bir başlangıç değeri ($x_1$ ) ve hata sınırı belirle ( ε ),

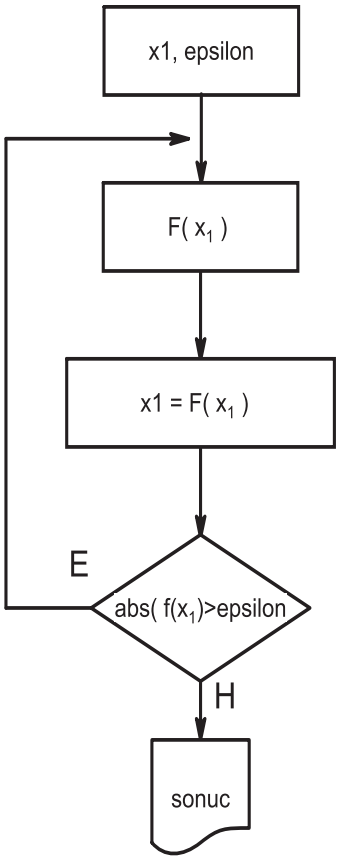
2- F($x_1$) 'i hesapla,

3- $x_1$= F($x_1$) atamasını yap,

4- $abs\left(f(x_1)\right) > \varepsilon \rightarrow$ adım 2 'ye atla,

5- Sonuçları yazdır ve çık.

## AKIŞ ŞEMASI




## c# kodları

```
namespace ConsoleApplication1
{
   class Program
   {
      public delegate double Function(double x);

                        //f(x)=x * x - 3 * x + 1
      static double F1(double x)
      {
         return (x * x + 1) / 3;
      }
      static double F(double x)
      {
         return x * x - 3 * x + 1;
      }


      static void Main(string[] args)
      {
         Console.WriteLine("\n\nBasit iterasyon yontemi\n");
                              double x,x0=0.5, epsilon=0.0001;
                              int i=0;
                              do
         {
            i++;
           x = F1(x0) ;
           x0 = x;
Console.WriteLine("iterasyon:" + i + "   x:" +
x0.ToString("0.00000000"));
         } while (Math.Abs(F (x0)) > epsilon);

         Console.WriteLine("\n\nNR Sonuc:" + x.ToString());
         Console.WriteLine("BIY Test:f(x)=" + F(x).ToString());
         Console.ReadLine();
      }

   }
}
```