SAYISAL ANALİZ

Dr. Öğr. ÜYESİ Abdullah SEVİN

SAYISAL ANALİZ

DENKLEM ÇÖZÜMLERİ

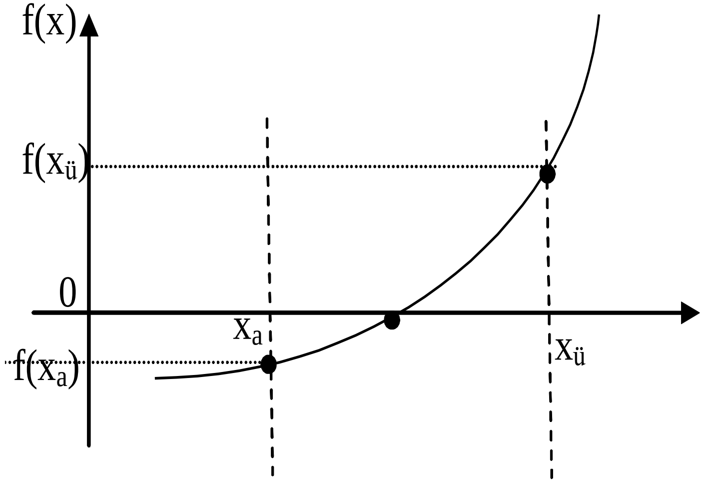
(Açık Yöntemler)

İÇİNDEKİLER

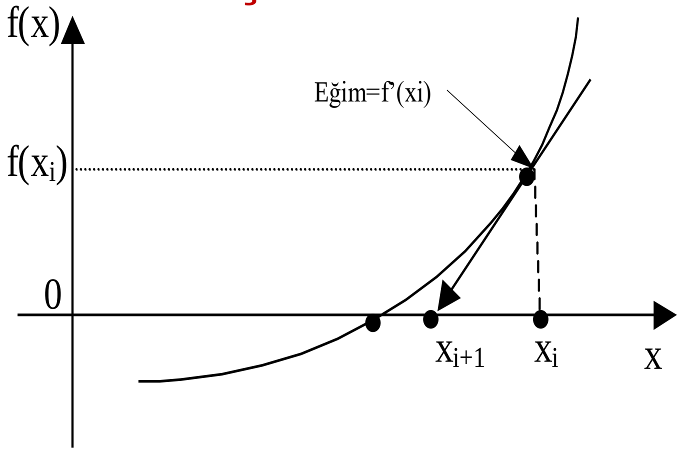
Denklem Çözümünde Açık Yöntemler

- Bu yöntem, x'in yalnızca başlangıç değeri kullanılan ya da kökü kapsayan bir aralık kullanılması gerekmez.
- Açık yöntemler hızlı sonuç vermesine karşın, başlangıç değeri uygun seçilmediğinde ıraksayabilir.
- Kökü iki başlangıç değeri arasında kıskaca alma ($f(x_a).f(x_ü) < 0$) sorgulaması yoktur.
- *Tüm açık yöntemler, kökün bulunması için matematiksel bir formül kullanır.*

- Kapalı Yöntem



- Açık Yöntem



Serhat Yılmaz'ın Sunusundan Alınmıştır.

Basit Sabit Noktalı İterasyon Yöntemi

❶ f(x) fonksiyonu $f(x)=0$ denkliği $x=g(x)$ formuna getirilir.

- **Örnek:** $f(x)= x^2 - x + 3 = 0 \Rightarrow x = x^2 + 3$

  $f(x)= \cos x = 0 \Rightarrow x = \cos x + x$

- Bu eşitliğin anlamı $y=x$ doğrusu ile $y=g(x)$ fonksiyonunun kesişim noktasını bulmaktır.

❷ Bir x_0 başlangıç değeri seçilir,

- x_0 , $| g'(x_0) | < 1$ şartını sağlar ise köke yakınsama olur.

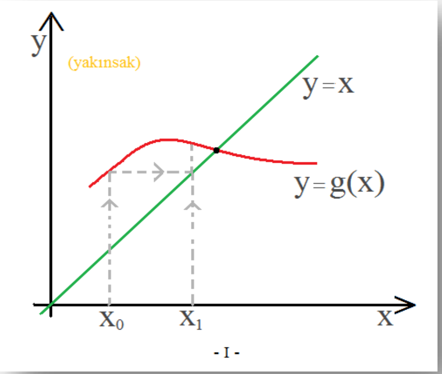
❸ $x_{n+1} = g(x_n)$ formu ile iterasyon gerçekleştirilir.

- $x_1=g(x_0)$
- $x_2=g(x_1)$
- …
- $x_n=g(x_{n-1})$

❹ Durdurma şartı

- $| x_{n+1} - x_n | < \varepsilon_s$ sağlanıncaya kadar
- Ya da belirli iterasyonda durdurulabilir

$$\% \varepsilon = \frac{|x_{i+1} - x_i|}{x_{i+1}} * 100$$



Basit Sabit Noktalı İterasyon Yöntemi

- **Örnek:** $f(x)= 3e^{-0.5x} – x$ fonksiyonunun kökünü mutlak hata $\varepsilon_a = 0.07$ sınırlamasına göre $x_0 = 8$ değerinden başlayarak hesaplayınız.
  - Her adım (iterasyon) için yeni x, g(x) ve hatayı hesaplayınız.

❶ f(x)=0 denkliği x=g(x) formuna getirilir.

- $x = 3e^{-0.5x}$

❷ $g(x) = 3e^{-0.5x}$ fonksiyonu $x_0 = 8$ başlangıç değeri ve $\varepsilon_a = 0.07$ hata sınırlamasına göre iterasyona tabi tutuluyor.

❸ 13. iterasyondan sonra $\varepsilon_a = 0.07$ hata ile kök değeri x=1.4 elde edilir.

(Yakınsak iterasyon)

iterasyon sayısı	x	g(x)	$h=\|x_n-x_{n-1}\|$
1	**8**	0,054946917	7,945053083
2	0,054946917	2,918701514	2,863754597
3	2,918701514	0,697161304	2,221540209
4	0,697161304	2,117066992	1,419905688
5	2,117066992	1,040892786	1,076174206
6	1,040892786	1,782765652	0,741872867
7	1,782765652	1,230264839	0,552500813
8	1,230264839	1,621707926	0,391443087
9	1,621707926	1,333435008	0,288272918
10	1,333435008	1,540173057	0,206738049
11	1,540173057	1,388919019	0,151254038
12	1,388919019	1,498032798	0,109113779
13	1,498032798	1,418494205	**0,079538593**
14	1,418494205	1,476043484	

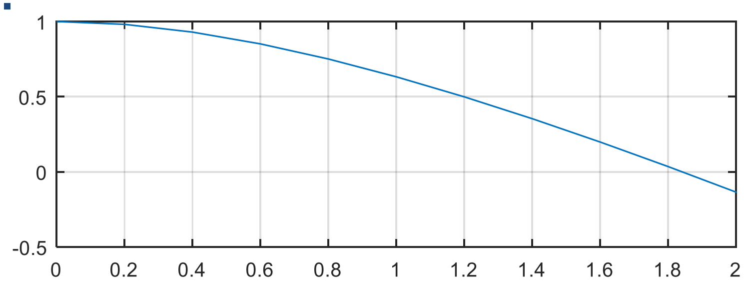
Basit Sabit Noktalı İterasyon Yöntemi

- **Örnek: f(x)= 2 – x – e^{-x} denklemi için 0 < x <2 aralığı için**
  - **a) Denklemin kökünü grafik ve iki eğriliği grafik yöntemi ile kabaca bulunuz.**
  - **b) Basit sabit noktalı köküne $x_0 = 0$ değerinden başlayarak 2 iterasyon yaklaşınız. Her iterasyon da yaklaşık mutlak hatayı hesaplayınız.**

➢ **Çözüm: (a)**

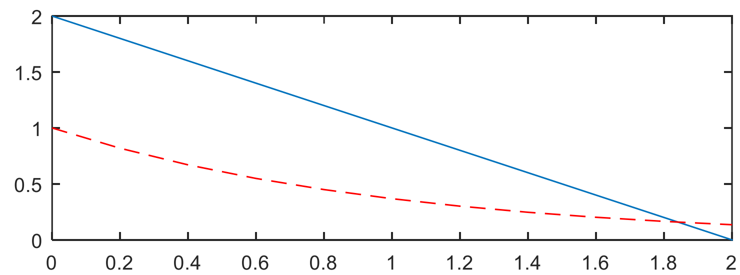
❶ **Denklemin kökü grafik yöntemi ile bulunur.**

```
x=0:0.2:2;
y=2-x-exp(-x);
subplot(2,1,1); plot(x,y);
grid on;
```



❷ **f(x) denklemi iki kısma ayrılır. 2 – x = e^{-x}**

```
f1=2-x;
f2=exp(-x);
subplot(2,1,2); plot(x,f1,x,f2, '-- r');
```



Basit Sabit Noktalı İterasyon Yöntemi

- **Örnek (devam): $f(x) = 2 - x - e^{-x}$ denklemi için**

b) Basit sabit noktalı iterasyon yöntemini kullanarak köke $x_0 = 0$ değerinden başlayarak 2 iterasyon yaklaşınız. Her iterasyon da yaklaşık mutlak hatayı hesaplayınız.

- **$f(x)=0$ denkliği $x=g(x)$ formuna getirilir.**
  - $x = 2 - e^{-x}$
- $x_{n+1} = g(x_n)$

1. iterasyon $x_0 = 0$

$x_1 = g(x_0) = 2 - e^{-x} \Rightarrow x_1 = 2 - e^0 = 2 - 1 = 1$

$\varepsilon = | x_{n+1} - x_n | = | 1 - 0 | = 1$

2. iterasyon $x_1 = 1$

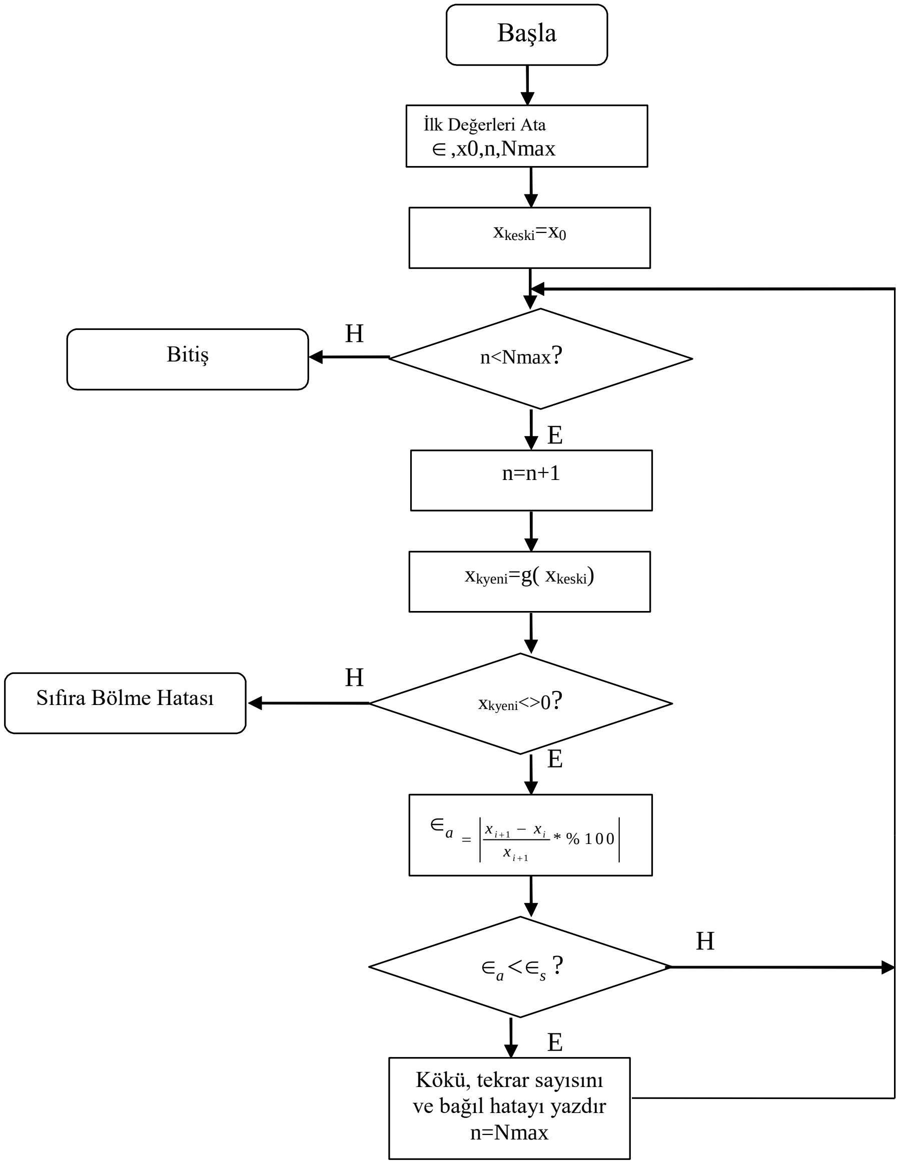
$x_2 = g(x_1) = 2 - e^{-x} \Rightarrow x_2 = 2 - e^{-1} = 2 - 0.3678 = 1.6321$

$\varepsilon = | x_{n+1} - x_n | = | 1.6321 - 1 | = 0.6321$

Basit Sabit Noktalı İterasyon Yöntemi

- **Örnek:** $f(x) = xe^x - 1$ denkleminin kökünü Basit Sabit Noktalı İterasyon yöntemini kullanarak yaklaşık mutlak hata $\varepsilon_s = 0.005$ sınırlamasının altına ininceye kadar $x_0 = 0.7$ değerinden başlayarak hesaplayınız. Başlangıç değeri için yakınsama şartını kontrol ediniz.

Basit Sabit Noktalı İterasyon Yöntemi Algoritması ve Matlab Kodu



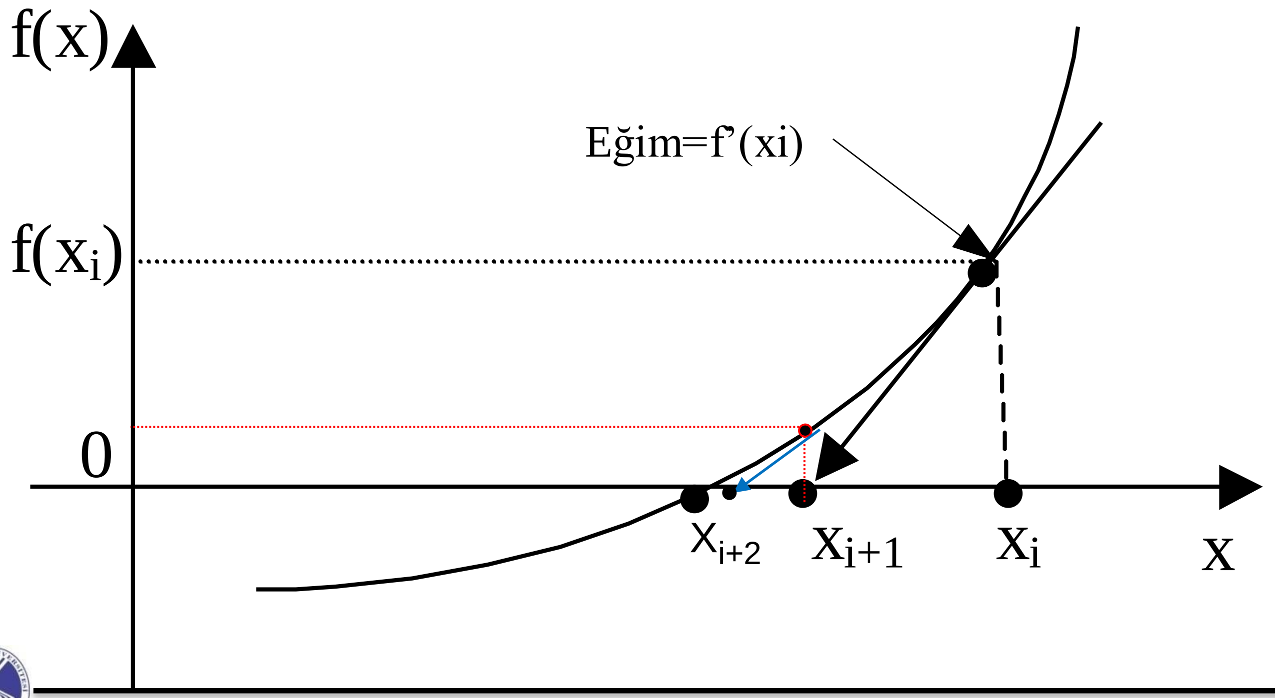

```
% f(x)=(e^-x)-x
% g(x)=(e^-x)
%%%%%%%%%%%%%%%%%%%%%

x0=1; es=0.01; n=0; Nmax=100;
gtx=-1*exp(-x0);

if abs(gtx)<1
xkeski=x0;
while (n<Nmax)
  xkyeni=exp(-xkeski); %g(xkeski)
  if xkyeni~=0
    ea=abs((xkyeni-xkeski))
    if ea<es
      disp('Kök='); disp(xkyeni);
      disp('Tekrar Sayisi='); disp(n);
      disp('Mutlak Hata=');
      disp(ea);
      n=Nmax;
    end
  else
    disp('Sifira bolme hatasi');
  end
  xkeski=xkyeni;
  n=n+1;
end
else
  disp('yakınsama olmaz!!')
end
```

Newton-Raphson Yöntemi

- En çok kullanılan yöntemlerden biridir.
- Köke, teğetler ile yaklaşılır.
  - Başlangıç değerinin fonksiyonu kestiği noktadan, çizilen teğetin yatay ekseni kestiği yeni nokta başlangıç değeri ile değiştirilerek köke yaklaşmaya çalışmaktır.
  - Bir noktadaki türev, o noktadan geçen teğetin eğimine eşittir.



$$f'(x_i) = \frac{f(x_i) - 0}{x_i - x_{i+1}}$$

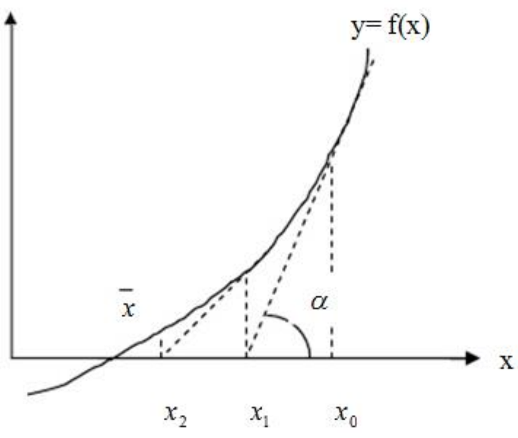
$$\boxed{x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}}$$

Newton-Raphson Yöntemi

❖ Yakınsaklık Koşulu

❶ **Başlangıç noktasındaki türev ile köke yaklaşma**

$$\mathrm{Tan}(\alpha_1) = \frac{f(x_0)}{(x_0 - x_1)} = f'(x_0)$$



❷ **x_i yalnız bırakılırsa, ifade basit iterasyondaki gibi $x_{n+1} = g(x_n)$ formuna dönüştürülür**

$$x_1 = \underbrace{x_0 - \frac{f(x_0)}{f'(x_0)}}_{g(x_0)}$$

❸ **Yakınsaklık koşulu,**

$$|g'(x_0)| < 1$$

$$g'(x_0) = \left(x_0 - \frac{f(x_0)}{f'(x_0)}\right)' = \left|\frac{f''(x_0).f(x_0)}{(f'(x_0))^2}\right| < 1$$

Newton-Raphson Yöntemi

❖ **Örnek : f(x) = x² - 10** denklemini Newton-Raphson yöntemini kullanarak $x_0 = 3$ değerinden başlayarak, iki iterasyon için çözünüz?

❶ **$f(x) = x^2 - 10 \Rightarrow f'(x) = 2x$**

❷ **$x_0 = 3$ 'ten başlayarak köke doğru yaklaşalım**

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

❸ **Durdurma Kriteri (Hata Sınırlaması)**

$| x_{i+1} - x_i | < \varepsilon_s$ yada iterasyon

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 3 - \frac{(-1)}{6} = \frac{19}{6} = 3{,}166$$

$$|x_1 - x_0| = |3{,}166 - 3| = 0.166$$

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = 3{,}166 - \frac{(0{,}023556)}{6{,}332} = 3{,}162$$

$$|x_2 - x_1| = |3{,}162 - 3{,}166| = 0.004$$

Serhat Yılmaz'ın Sunusundan Alınmıştır.

Newton-Raphson Yöntemi

❖ **Örnek :** **f(x) = e^{-2x} - x + 2** denkleminin kökünü Newton-Raphson yöntemini kullanarak x_0= 1 değerinden başlayarak, 3 iterasyon için çözünüz?

Not: Her iterasyonda yaklaşık bağıl hata yüzdesini de hesaplayınız. Tüm değerler virgülden sonra 4 basamak alınacak.

❶ **f(x) = e^{-2x} - x + 2** ⇒ **f'(x) = = -2e^{-2x} -1**

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \qquad \%\varepsilon_r = \frac{|x_{i+1} - x_i|}{x_{i+1}} * 100$$

❷ **x_0 = 1 'den başlayarak köke doğru yaklaşalım**

1. iterasyon

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 1 - \frac{(0.1353 - 1 + 2)}{-0.2706 - 1} = 1 - \frac{1.1353}{-1.2706} = 1 - (-0.8935) = 1.8935 \qquad \%\varepsilon_r = \frac{|1.8935 - 1|}{1.8935} * 100 = \%47.18$$

2. iterasyon

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = 1.8935 - \frac{(0.0226 - 1.8935 + 2)}{-0.0452 - 1} = 2.017 \qquad \%\varepsilon_r = \frac{|2.017 - 1.8935|}{2.017} * 100 = \%6.12$$

3. iterasyon

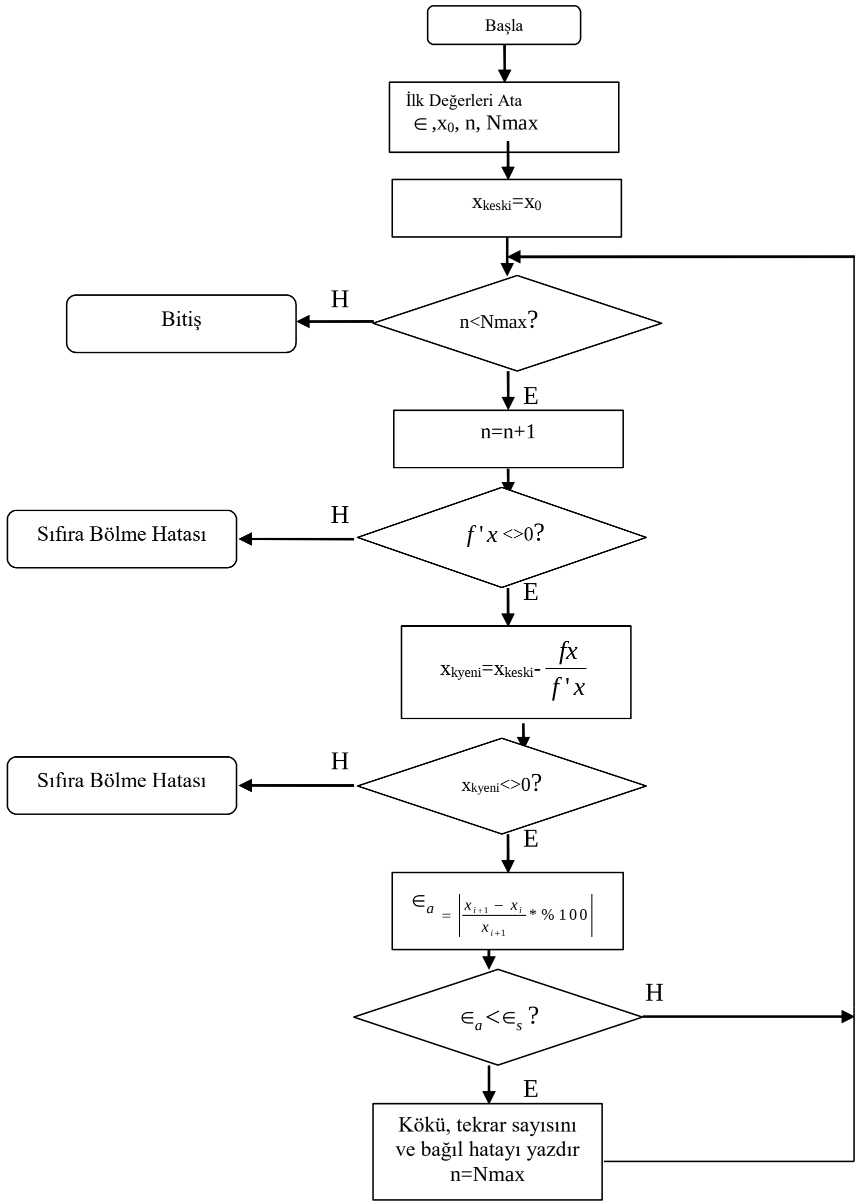
$$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)} = 2.017 - \frac{(0.0177 - 2.017 + 2)}{-0.0354 - 1} = 2.0176 \qquad \%\varepsilon_r = \frac{|2.0176 - 2.017|}{2.0176} * 100 = \%0.06$$

Newton-Raphson Yöntemi

❖ **Örnek :** **$f(x) = x - e^{-x}$** denkleminin kökünü Newton-Raphson yöntemini kullanarak $x_0 = 2$ değerinden başlayarak, **yaklaşık yüzde bağıl hata değeri**, en az 1 anlamlı basamak veren hata sınırlaması altına ininceye kadar hesaplayınız?

Not: Tüm değerler virgülden sonra 4 basamak alınacak.

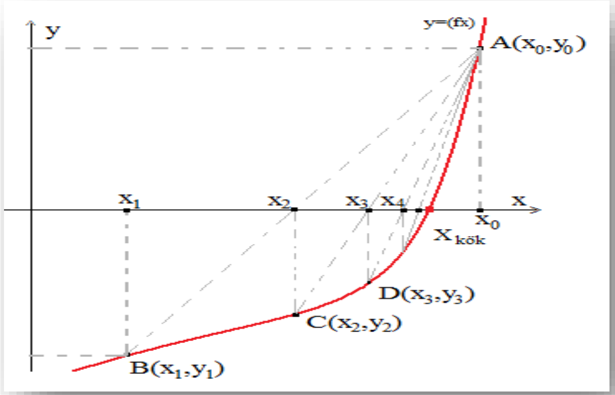
Newton-Raphson Yöntemi Algoritması



Kiriş (Secant) Yöntemi

- Newton-Raphson yönteminin en büyük problemlerinden biri, bazı fonksiyonların/denklemlerin türevini almanın oldukça zor olabileceğidir. Bu işlemler uzun zaman alabilir.
- Türev almadan çözüm için Kiriş (secant) yönteminden yararlanılır.
- Şekildeki A-B noktaları arasında, x_0 ve x_1 başlangıç değerleri kullanılarak türev alınmadan gerçek köke daha yakın bir kök bulunabilir.

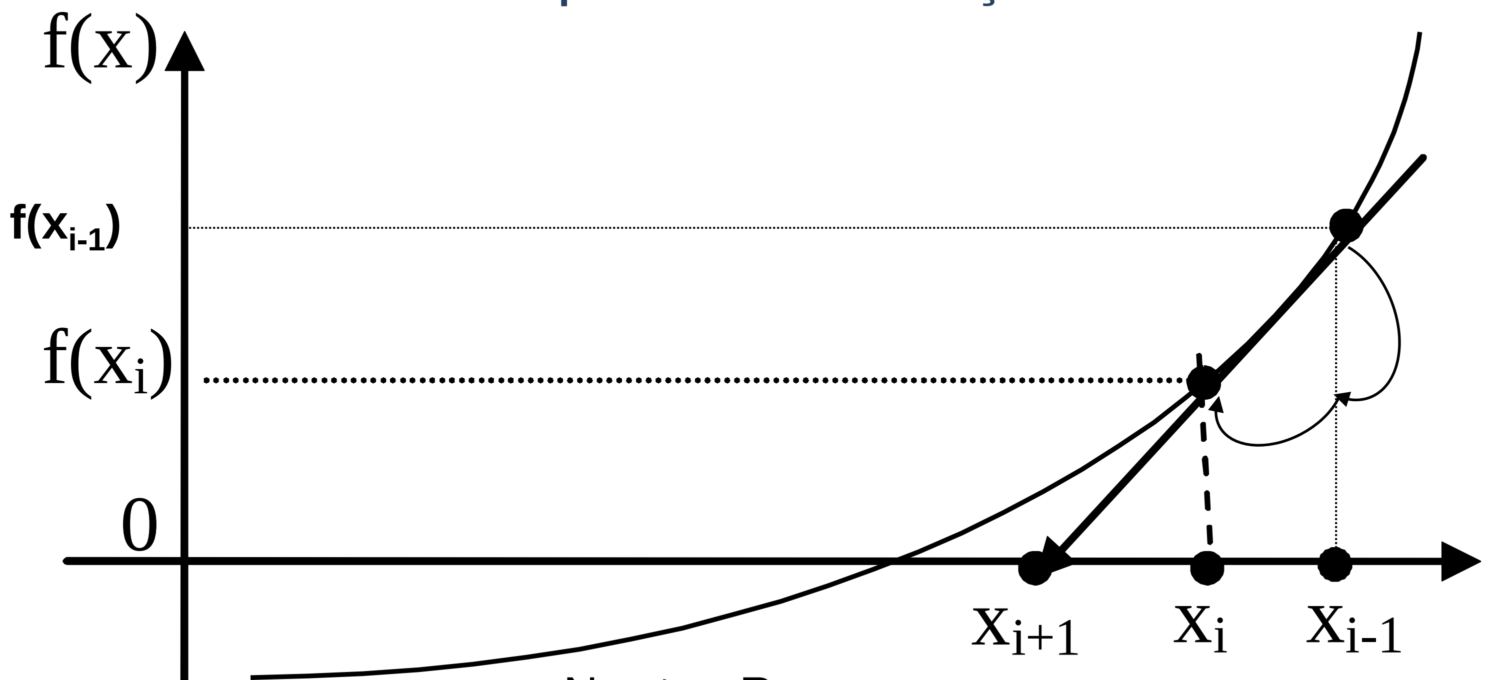
$$x_{i+1} = x_i - \frac{(x_i - x_{i-1})}{(y_i - y_{i-1})} y_i$$



- Kirişin x eksenini kestiği nokta köke yakın noktadır.
- Her yeni iterasyonda yeni bir kiriş noktaları bulunarak kirişlerin x eksenini kestiği yeni noktalar ile köke yaklaşılır.
- Bu işlem diğer yöntemlerdeki gibi belirli bir hata sınırlamasına kadar tekrar edilir.
- y=f(x)

Kiriş (Secant) Yöntemi

- Secant Yöntemi ile Newton-Raphson Yönteminin İlişkisi



Newton R

$$f'(x_i) = \frac{f(x_i) - 0}{x_i - x_{i+1}}$$

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

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

$$\boxed{x_{i+1} = x_i - \frac{f(x_i)(x_{i-1} - x_i)}{f(x_{i-1}) - f(x_i)}}$$

Kiriş (Secant) Yöntemi

Secant Yönteminin Regula-Falsi Yöntemi İle Karşılaştırılması

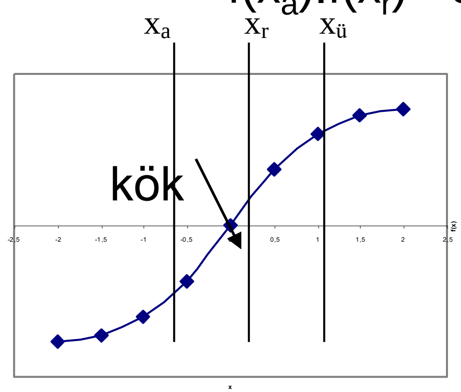
İkisinde de iki ilk tahmin değeri var

Regula Falsi

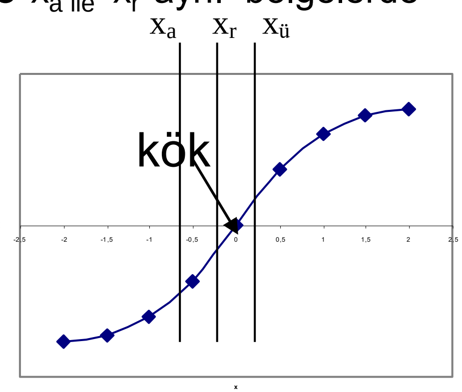
$$x_r = x_ü - \frac{f(x_ü)(x_a - x_ü)}{f(x_a) - f(x_ü)}$$

Güncellenecek sınır

- $f(x_a).f(x_r) < 0$ x_a ile x_r farklı bölgelerde xü(yeni)=xr
- $f(x_a).f(x_r) > 0$ x_a ile x_r aynı bölgelerde xa(yeni)=xr



Kök, x_a, x_r arasında



Kök, x_r, $x_ü$ arasında

Sekant

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

X_{i+1} X_i X_{i-1}

Kiriş (Secant) Yöntemi

❖ **Örnek :** **f(x) = e^-x^ - x** denkleminin köklerini $x_0 = 0$ ve $x_1 = 1$ ilk tahminlerinden başlayarak Secant Yöntemi ile çözünüz?

$f(0) = 1.0 \quad f(1) = -0.632\,120\,559 \Rightarrow f(0)f(1) < 0$ olduğundan aralıkta kök vardır.

$$x_0 = 0, \quad y_0 = f(x_0) = 1 \quad x_1 = 1 \quad y_1 = f(x_1) = -0.632\,120\,559$$

- $$x_2 = x_1 - \frac{x_1 - x_0}{y_1 - y_0} y_1 = 1 - \frac{1-0}{-0.632120559 - 1}(-0.632120559) = 0.612\,699 \,,$$

$$f(0.612\,699) = y_2 = -0.07\,081\,27$$

- $$x_3 = x_2 - \frac{x_2 - x_1}{y_2 - y_1} y_2 = 0.612\,699 - \frac{(0.612\,699 - 1)}{(-0.07\,081\,27 + 0.632\,120\,)}(-0.07\,081\,27) = 0.563\,838$$

$$|x_3 - x_2| = |0.563\,838 - 0.612\,699| = 0.048\,861$$

$$f(0.563\,838\,) = y_3 = 0.00518297,$$

- $x_4 = 0.567170 \qquad |x_4 - x_3| = |0.567170 - 0.563\,838\,| = 0.003\,332$
- $x_5 = 0.567143 \qquad |x_5 - x_4| = |0.567143 - 0.567170| = 2.7 \times 10^{-5}$
- $x_6 = 0.567143 \qquad |x_6 - x_5| = |0.567143 - 0.567143| = 0$

O halde verilen denklemin yaklaşık kökü $x = 0.567143$ dir.

Kiriş (Secant) Yöntemi

❖ **Örnek : f(x) = x³ -3x² –x + 9** denkleminin köklerini x_0 = -2 ve x_1 = -1 ilk tahminlerinden başlayarak Secant Yöntemi ile çözünüz?

$$x_{i+1} = x_i - \frac{f(x_i)\left(x_{i-1} - x_i\right)}{f(x_{i-1}) - f(x_i)}$$

$$x_0 = -2 \Rightarrow f(x_0) = (-2)^3 - 3*(-2)^2 - (-2) + 9 = -9$$

$$x_1 = -1 \Rightarrow f(x_1) = (-1)^3 - 3*(-1)^2 - (-1) + 9 = 6$$

1. iterasyon

$$x_2 = x_1 - \frac{f(x_1)\left(x_0 - x_1\right)}{f(x_0) - f(x_1)} = -1 - \frac{6*\left(-2-(-1)\right)}{-9-6} = -1 - 0.4 = -1.4$$

2. iterasyon

$$x_2 = -1.4 \Rightarrow f(x_2) = (-1.4)^3 - 3*(-1.4)^2 - (-1.4) + 9 = 1.776$$

$$x_3 = x_2 - \frac{f(x_2)\left(x_1 - x_2\right)}{f(x_1) - f(x_2)} = -1.4 - \frac{1.776*\left(-1-(-1.4)\right)}{6-1.776} = -1.4 - 0.1681 = -1.5681$$

ÖDEV

Ödevler dersin Araştırma Görevlisine, takiben eden hafta teslim edilecektir.

Not: Vaktinde teslim edilmeyen ödevler alınmayacaktır.

- $f(x) = \sin(x) - x.e^x + 3$ denkleminin köküne Newton Raphson yöntemini kullanarak $x_0 = 1$ değerinden başlayarak yaklaşık yüzde bağıl hata $\varepsilon_s = \%3$ 'ün altına ininceye kadar yaklaşınız.

Ödev Hakkında Bilgilendirme:

- Ödev çıktı olarak teslim edilecektir.
- Ödev hem program dosyasını (MATLAB) hem de el ile çözümünü içerecektir.

KAYNAKLAR

- Serhat YILMAZ, "*Bilgisayar İle Sayısal Çözümleme*", Kocaeli Üniv. Yayınları, No:168, Kocaeli, 2005.
- Steven C. Chapra, Raymond P. Canale (Çev. H. Heperkan ve U. Kesgin), "*Yazılım ve Programlama Uygulamalarıyla Mühendisler İçin Sayısal Yöntemler*", Literatür Yayıncılık.
- İlyas ÇANKAYA, Devrim AKGÜN, Sezgin KAÇAR "*Mühendislik Uygulamaları İçin MATLAB*", Seçkin Yayıncılık
- Prof.Dr. Asaf Varol, Sayısal Analiz Ders Notları, Fırat Üniversitesi
- Fahri VATANSEVER, "*İleri Programlama Uygulamaları*",Seçkin Yayıncılık
- Yüksel YURTAY, Sayısal Analiz Ders Notları, Sakarya Üniversitesi