Optimizasyon Teknikleri

Ders Notu – 3

TEK DEGİŞKENLİ OPTİMİZASYON

Prof. Dr. Bilal ALATAŞ

İÇERİK

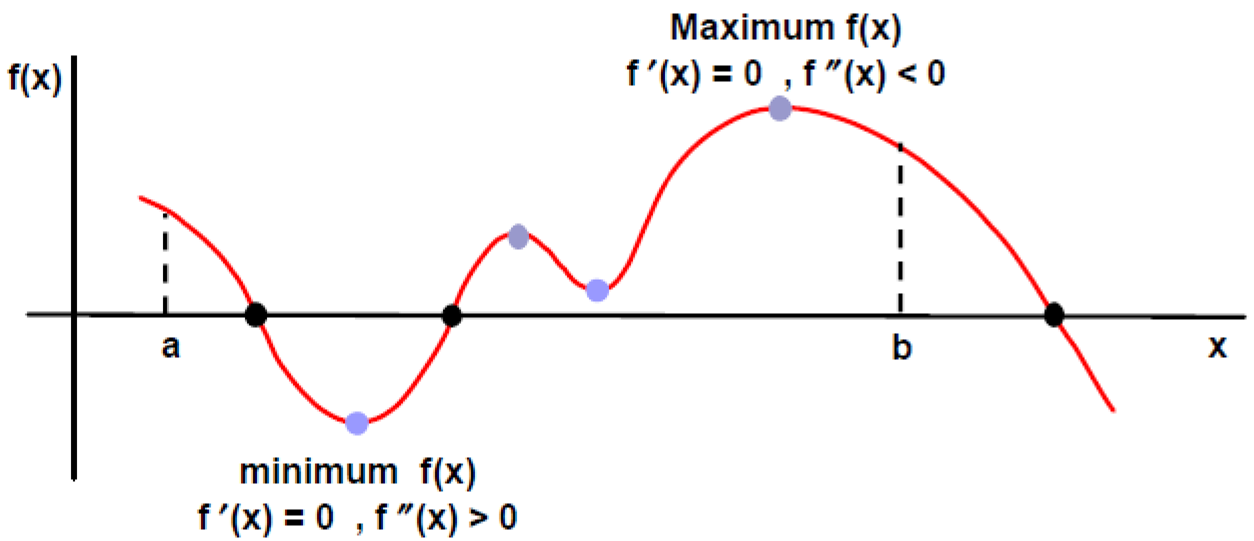
- **TEK DEĞİŞKENLİ OPTİMİZASYON METOTLARI**
  - **GOLDEN SECTION METODU**
  - **BISECTION METODU**
  - **POLINOM METODU**
  - **NEWTON-RAPHSON METODU**
  - **SECANT METODU**
- **METOTLAR HAKKINDA GENEL DEĞERLENDİRME**

TEK DEGISKENLI OPTIMIZASYON METOTLARI

<u>Tek degiskenli optimizasyon ifadesi</u>:

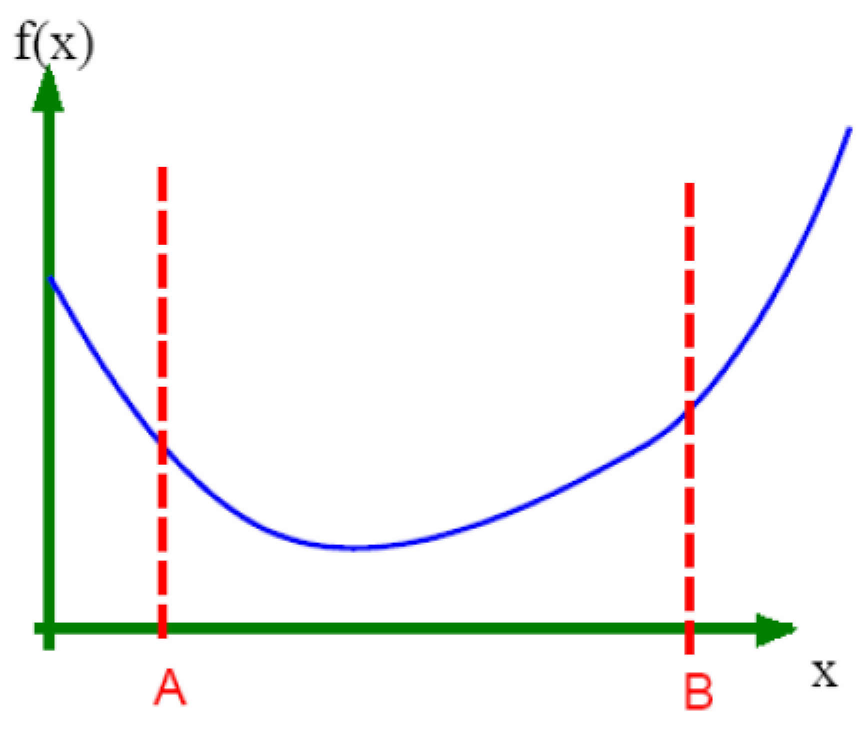
Minimize: f(x)

x: optimizasyon degiskeni



TEK DEGISKENLI OPTIMIZASYON METOTLARI

<u>Optimizasyon probleminin çözümü</u>:



Verilen f (x) fonksiyonunun [A,B] araliginda minimumunu bulmak istiyoruz.

Minimumu en az hesapla (f(x) degeri kullanarak) bulmak istiyoruz.

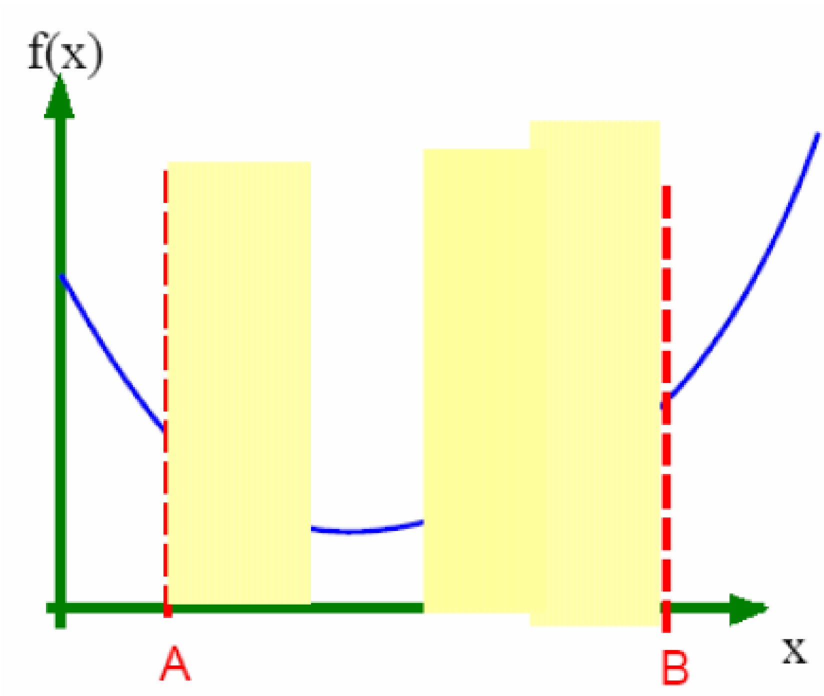
f(x) fonksiyonunun açik halinin (analitik ifadesi) bilinip bilinmedigi durumlarda minimum nasil bulunur?

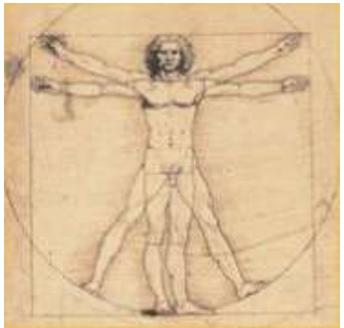
GOLDEN SECTION METODU

- Golden section (Altin Bölme) metodun temeli Italyan matematikçi Fibonacchi tarafindan atilmistir (1202).

- Metot, optimumun arastirilacagi alani sürekli daraltmaya dayanir.

- Optimizasyon sonunda çok küçülen alanin optimum çözümü barindirdigi düsünülür.





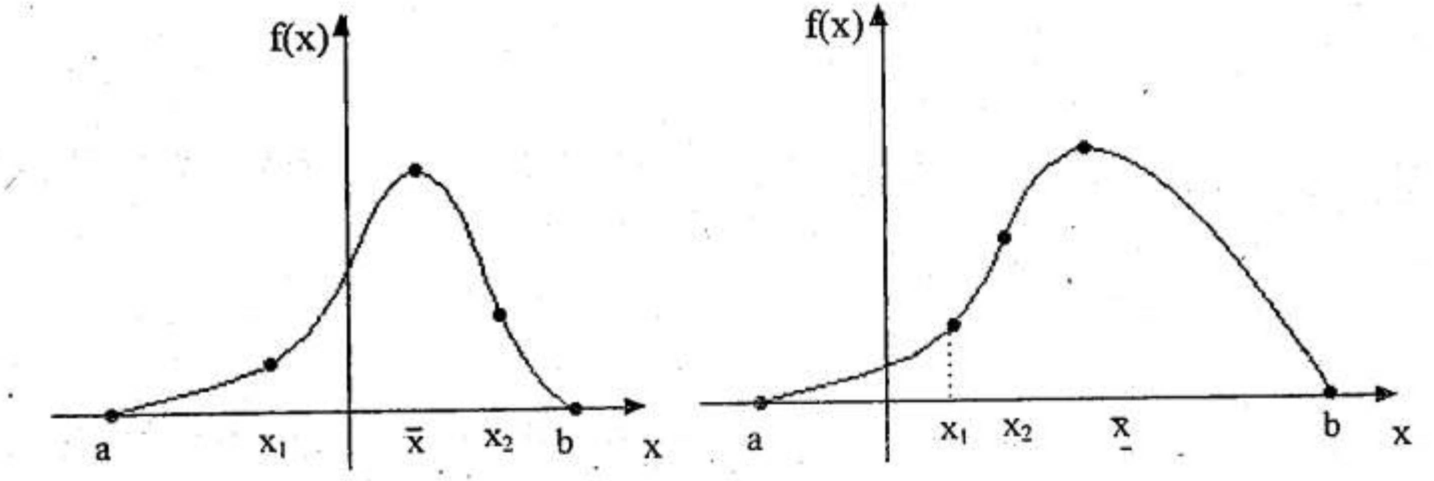
- **f’(x) bulunamadığında veya f’(x) =0 eşitliğinin çözülemediği durumlarda tek değişkenli bir fonksiyonun daha önce anlattığımız yollarla çözülmesi zor olabilir.**
- **Bu durumda tek değişkenli f(x) fonksiyonunun en iyi değerinin araştırılmasında tek değişkenli araştırma teknikleri kullanılabilir. Bu tekniklerin kullanılabilmesi için fonksiyonun tek modlu olması şarttır.**
- **f(x) fonksiyonu $a \leq x \leq b$ koşulu altında maksimize edilmek istensin. Bu problem genel gösterimle aşağıdaki gibi ifade edilir:**
  - **enb f(x), $a \leq x \leq b$**

<u>TANIM:</u> **f(x), $[a,b]$kapsamındaki bir nokta ($\bar{x}$) için $[a,\bar{x}]$ aralığında hızla artarken, $[\bar{x},b]$ aralığında hızla azalıyorsa tek modlu bir fonksiyondur.**

Eğer f(x) tek modlu bir fonksiyon ise bahsedilen problemin en iyi çözümü $[a,b]$ aralığındadır. En iyi çözümün bulunduğu kesin olan bu aralığın küçültülmesi uygun olur. Bunun için $x_1 < x_2$ olmak üzere iki nokta seçilir ve fonksiyonun bu noktalardaki değeri hesaplanır. $f(x_1)$ ve $f(x_2)$ değerlerinin karşılaştırılması sonucunda, üç durumdan biriyle karşılaşılır:

Durum1: $f(x_1) < f(x_2)$

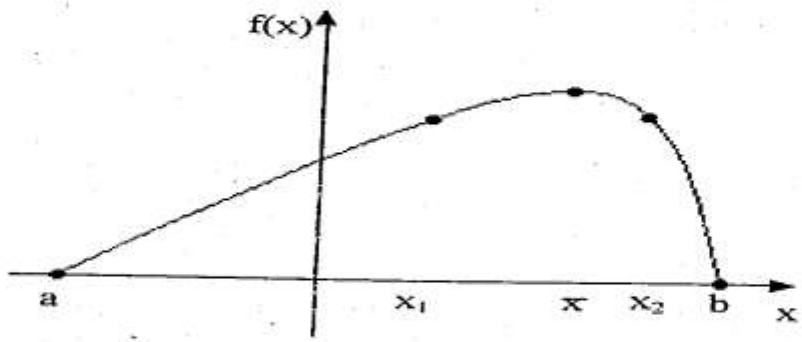
F(x) tek modlu olduğundan $[x_1, x_2]$ alt aralığının en azından bir bölümünde artacaktır. Bu durumda problemin en iyi çözümü $[a, x_1]$ aralığında bulunamaz. En iyi çözüm $\bar{x} \in (x_1, b]$ olur.



Şekil 6.21
$f(x_1) < f(x_2)$ ise, $\bar{x} \in (x_1, b]$

Durum2: $f(x_1) = f(x_2)$

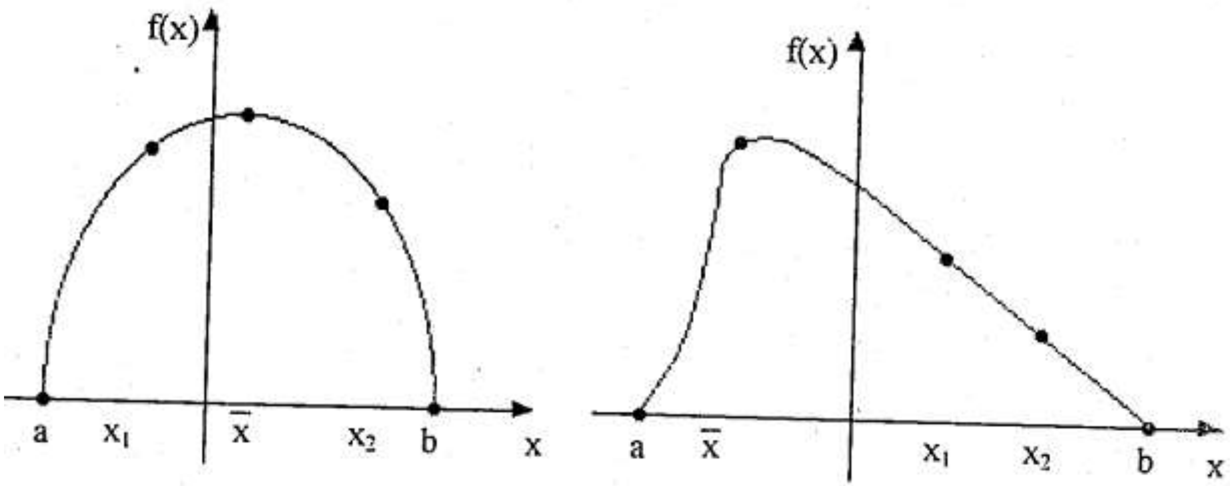
f(x) tek modlu olduğundan, $[x_1, x_2]$ alt aralığının belirli bir bölümünde azalır. U durumda, problemimizin en iyi çözümü $\bar{x} \in$ [a,x_2) aralığında olur.



Şekil 6.22
$f(x_1) = f(x_2)$ ise, $\bar{x} \in [a, x_2)$

Durum3: $f(x_1) > f(x_2)$

Bu durumda fonksiyonun eğrisi x_2 ye ulaşmadan önce azalmaya başlar. Buna göre $\bar{x} \in [a, x_2)$ olur.



Şekil 6.23
$f(x_1) > f(x_2)$ ise, $\bar{x} \in [a, x_2)$

Bu durumda f(x)in en büyük noktası $[a, x_2)$ veya $(x_1, b]$ aralıklarından birinde bulunur. $\bar{x}$ ın bulunduğu aralığa belirsizlik aralığı denir.

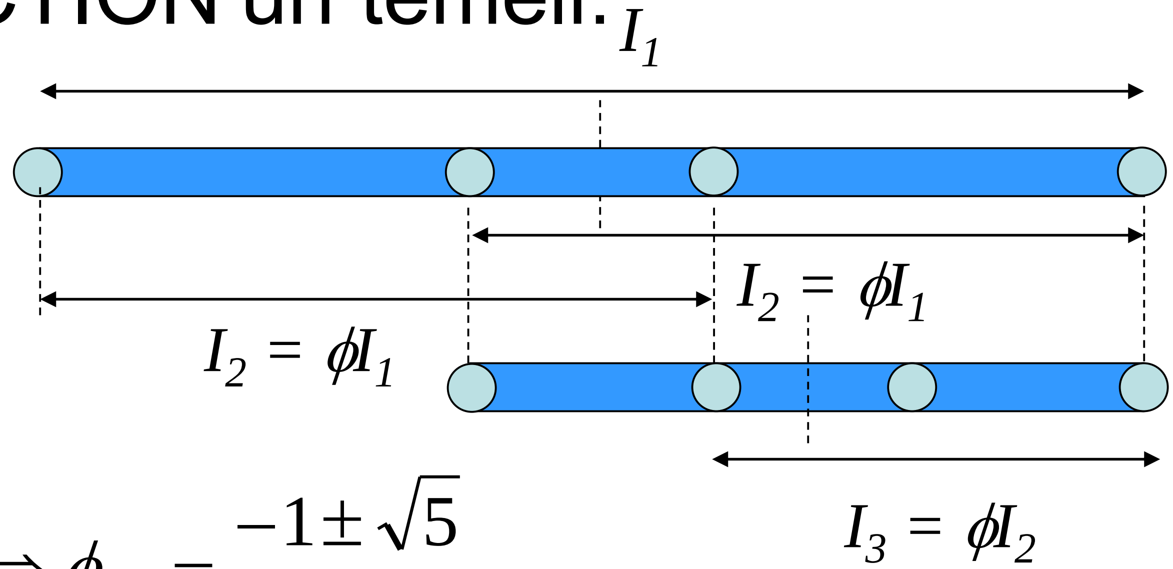
<u>Adım1:</u> **fonksiyonun tanımlandığı $[a,b]$kapalı aralığı x'in belirsizlik aralığı olarak seçilir. $[a,b]$ araştırmanın başlatıldığı ilk aralık olduğundan $[a,b]$ ye ilk veya başlangıç belirsizliği denir. Bu aralıktan uygun olarak x_1 ve x_2 gibi herhangi iki nokta seçilir ve f(x) in bu noktalardaki değerleri hesaplanır.**

<u>Adım2:</u> **$f(x_1)$ ve f (x_2) değerleri karşılaştırılarak yukarıda açıklanan üç durumdan hangisinin ortaya çıktığı belirlenir. Saptanan durum doğrultusunda başlangıç belirsizlik aralığı daraltılabilir. Bu adımda belirlenen daha dar aralık üçüncü adımın belirsizlik aralığı olur.**

<u>Adım3:</u> **ikinci adımda belirlenen belirsizlik aralığından aynı yaklaşımla iki yeni nokta seçilir ve fonksiyonun bu noktalardaki değeri hesaplanır. Hesaplanan değerler karşılaştırılarak 2. Adımda belirlenenden daha kısa olan bir aralık saptanır. Belirlenen aralığın uzunluğu yeterince kısa değilse ikinci adıma dönülür. Bu adımlar yeterince kısa adımlar elde edinceye kadar tekrar edilir.**

Bu adımların kullanıldığı Golden kesit araştırmasında noktaların seçilmesi için golden oran adı verilen ve mimarlıkta sıklıkla kullanılan bir oran kullanılır.

- GOLDEN SECTION'un temeli:



$$I_1 = I_2 + I_3$$

$$\Rightarrow I_1 = \phi I_1 + \phi^2 I_1$$

$$\phi^2 + \phi - 1 = 0 \Rightarrow \phi_{1,2} = \frac{-1 \pm \sqrt{5}}{2}$$

$$\phi = \frac{\sqrt{5} - 1}{2} \approx 0.618034$$

- Son Aralık: $I_N = \phi^N I_1$

$r^2 + r = 1$ kareli eşitliğinin tek pozitif kökü r olsun. Bu durumda golden oran (r);

$$r = \frac{5^{\frac{1}{2}} - 1}{2} = 0.618$$

olarak belirlenir.

Golden kesit araştırmasının ilk adımında f(x) in $x_1 = b - r(b-a)$ ve $x_2 = a + r(b-a)$ olarak belirlenen noktalardaki değerleri hesaplanır.

<u>Yeni sol taraf noktası:</u> yürürlükteki belirsizlik aralığının sağ bitim noktasından r kesri ile belirlenen noktaya hareket edilir.

<u>Yeni sağ taraf noktası:</u> yürürlükteki belirsizlik aralığının sol bitim noktasından r kesri ile belirlenen noktaya hareket edilir.

GOLDEN SECTION METODU

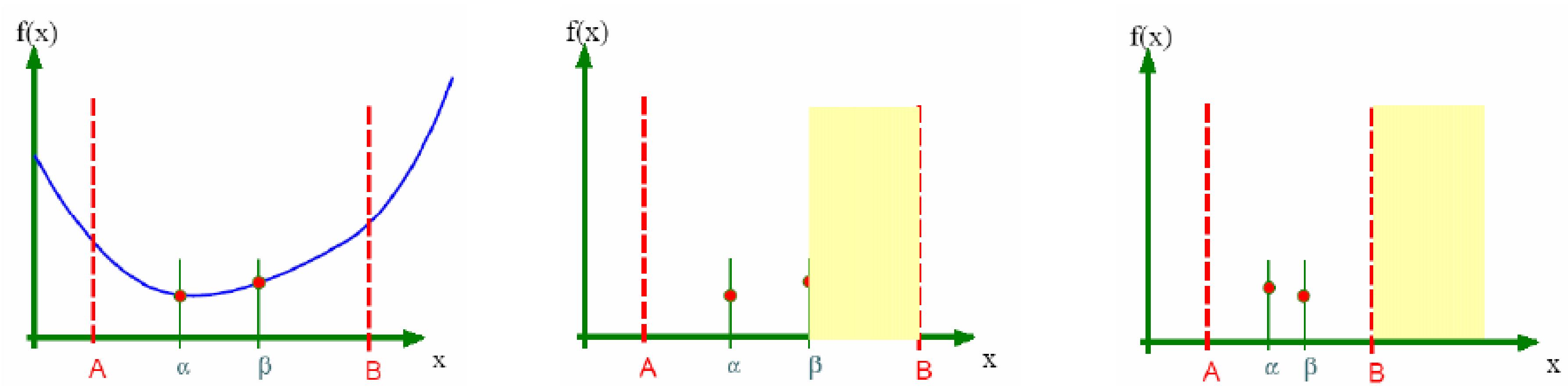
<u>Golden Section metodu çözüm algoritmasi</u>:

- Her iterasyonda, bir önceki alaninin bir kismi atilarak alan daraltilir.

- f(x)'in türevi (gradyenti) kullanilmadigindan, alanin hangi bölgesinin atilacagina f(x) fonksiyonunun 2 noktadaki degeri kullanilarak karar verilir.

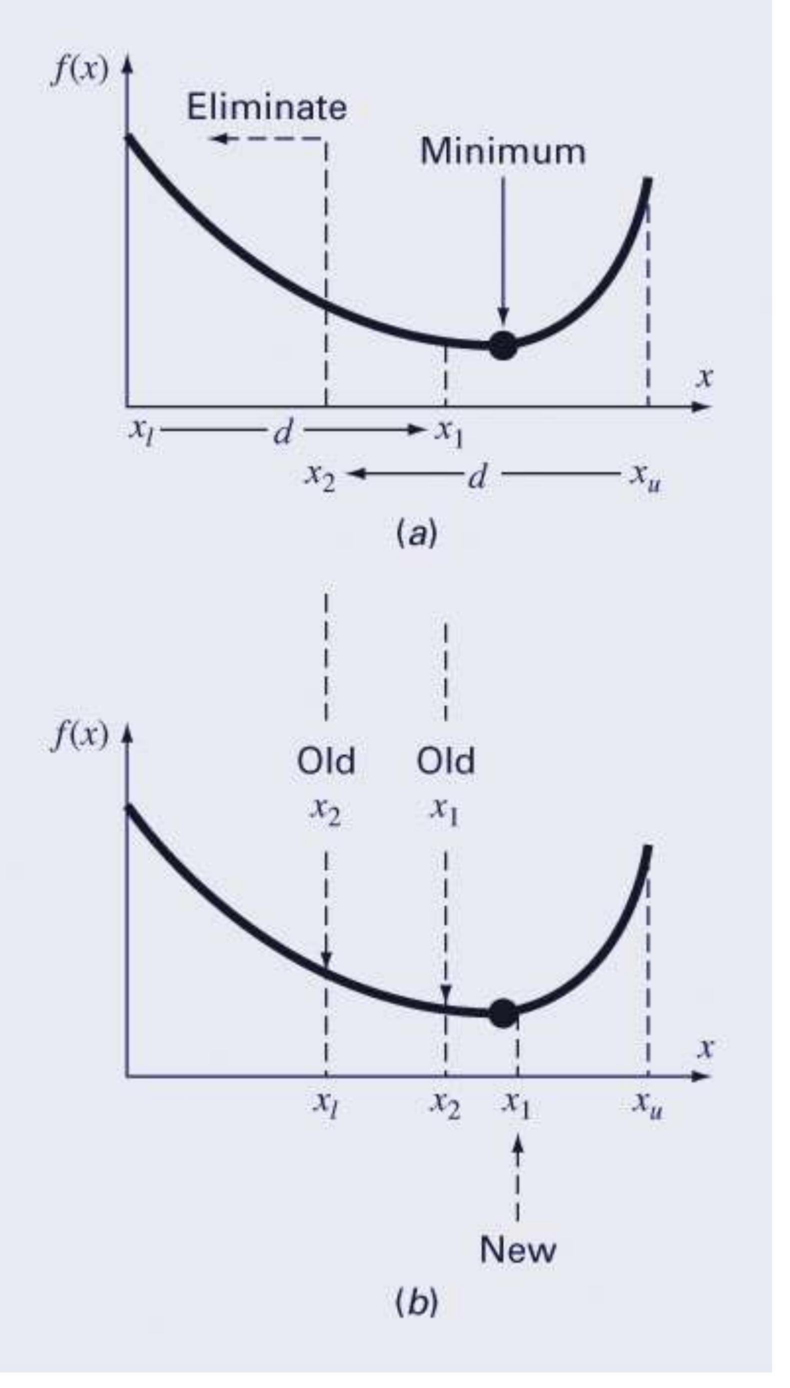
- f(a) < f(ß) ise, [ß, B] araligi atilir ve bir sonraki iterasyonda B= ß olur
- f(a) > f(ß) ise, [A, a] araligi atilir ve bir sonraki iterasyonda A= a olur

- Noktalarin (a, ß) yerlerinin seçiminde altin bölme (Golden Section) kurali uygulanir. Bu islem yenilenmis [A, B] araligi ile çözüm yakinsayana kadar tekrar edilir.

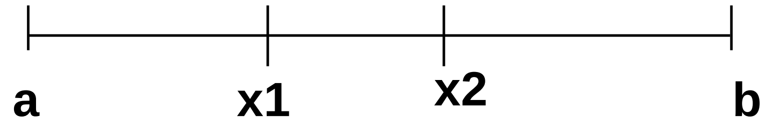


- $f(x_1) < f(x_2) \rightarrow x_2$ yeni alt limittir; x_1 de yeni x_2 dir.

- $f(x_2) < f(x_1) \rightarrow x_1$ yeni üst limittir; x_2 de yeni x_1 dir.



```
Initialize:
x1 = a + (b-a)*0.382
x2 = a + (b-a)*0.618
f1 = f(x1)
f2 = f(x2)
Loop:
if f1 > f2 then
  a = x1; x1 = x2; f1 = f2
  x2 = a + (b-a)*0.618
  f2 = f(x2)
else
  b = x2; x2 = x1; f2 = f1
  x1 = a + (b-a)*0.382
  f1 = f(x1)
endif
```



$\tau = (\sqrt{5} - 1)/2$

$x_1 = a + (1 - \tau)(b - a);\ f_1 = f(x_1)$

$x_2 = a + \tau(b - a);\ f_2 = f(x_2)$

while $((b - a) > tol)$ **do**

 if $(f_1 > f_2)$ **then**

  $a = x_1$

  $x_1 = x_2$

  $f_1 = f_2$

  $x_2 = a + \tau(b - a)$

  $f_2 = f(x_2)$

 else

  $b = x_2$

  $x_2 = x_1$

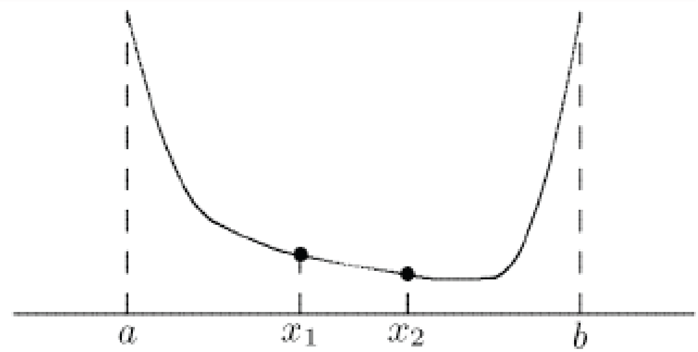
  $f_2 = f_1$

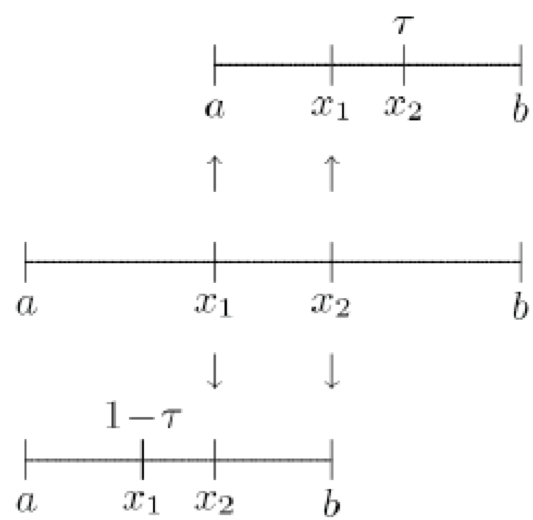
  $x_1 = a + (1 - \tau)(b - a)$

  $f_1 = f(x_1)$

 end

end

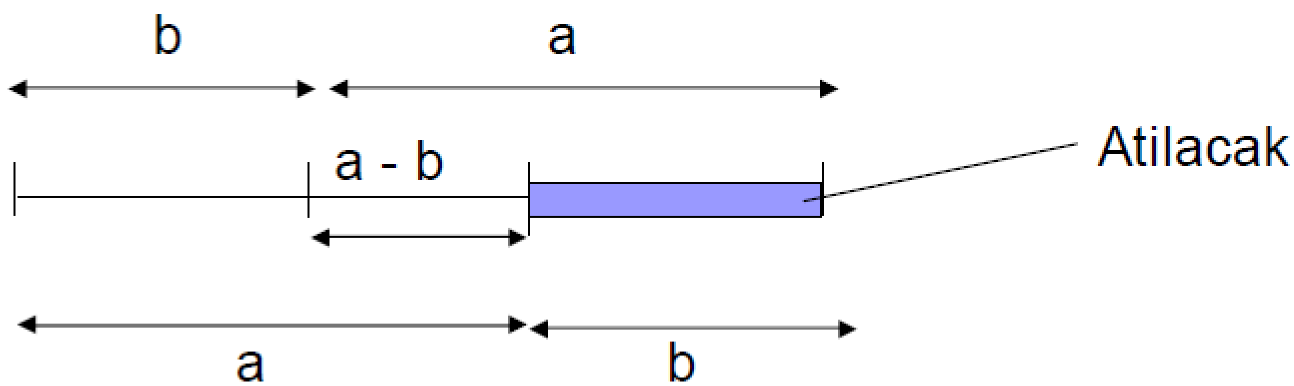




GOLDEN SECTION METODU

Altin Bölme (Golden Section) kurali:

✓ Hesap maliyetini azaltmak için Golden Section metodunda aralik sürekli belirli oranda (% 38 oraninda) daraltilir.



b/(a-b) = a/b >> b*b = a*a – ab

Çözüm:

a = (b ± b* sqrt(5)) / 2 >> a/b = -0.618 or 1.618 (Golden Section orani)

Not: 1/1.618 = 0.618

- Example
  - Find maximum of $f(x) = \sin x - x^2$
  - Start with x_l=0 and x_u=1

$$x_1 = 0 + \frac{\sqrt{5}-1}{2}(1-0) = 0.618 \quad \Rightarrow f(x_1) = 0.197$$

$$x_2 = 1 - \frac{\sqrt{5}-1}{2}(1-0) = 0.382 \quad \Rightarrow f(x_2) = 0.226$$

$$f(x_2) > f(x_1)$$

$$x_{u,new} = x_1 = 0.618$$

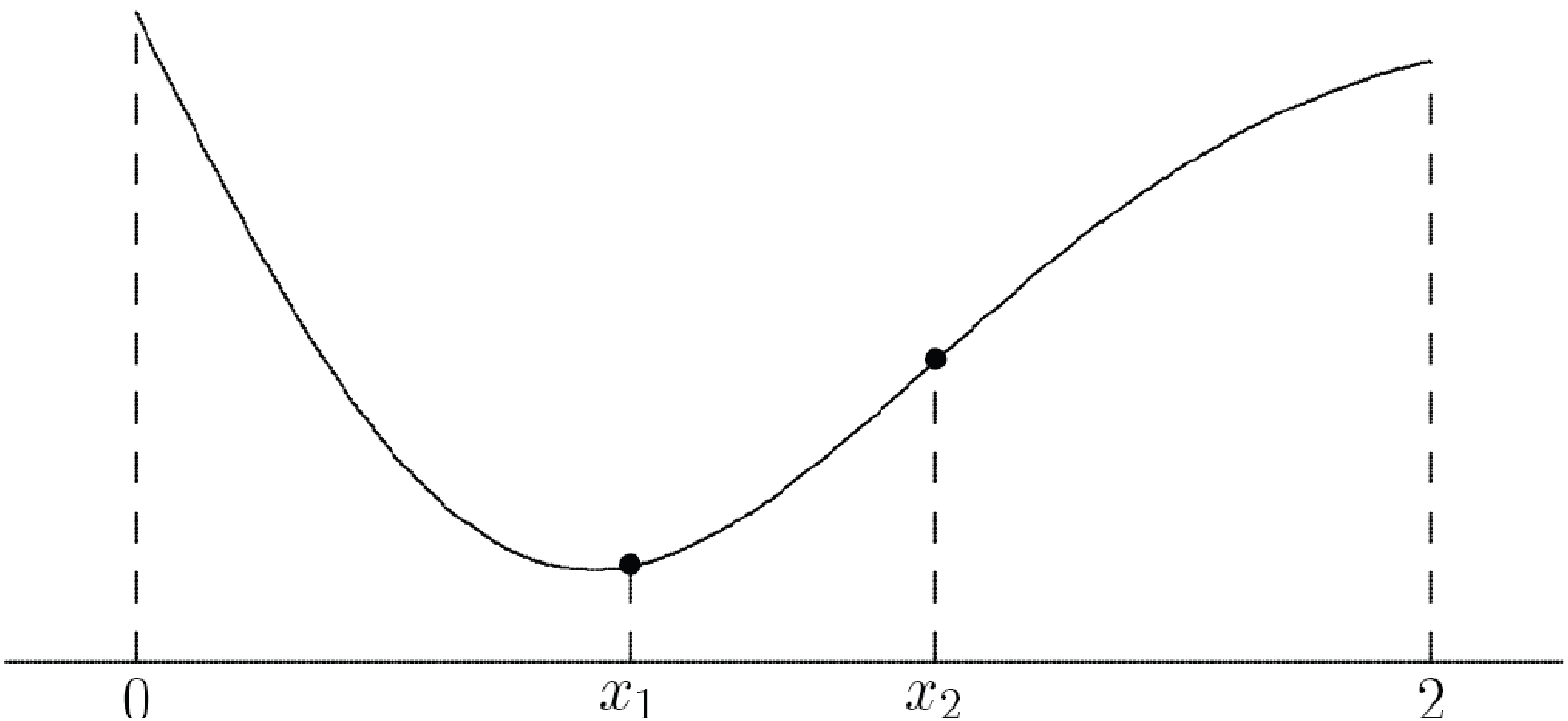
$$x_{1,new} = x_2 = 0.382$$

$$\begin{aligned} x_{2,new} &= x_{u,new} - \frac{\sqrt{5}-1}{2}\left(x_{u,new} - x_l\right) \\ &= 0.618 - 0.618(0.618 - 0) \\ &= 0.236 \end{aligned}$$

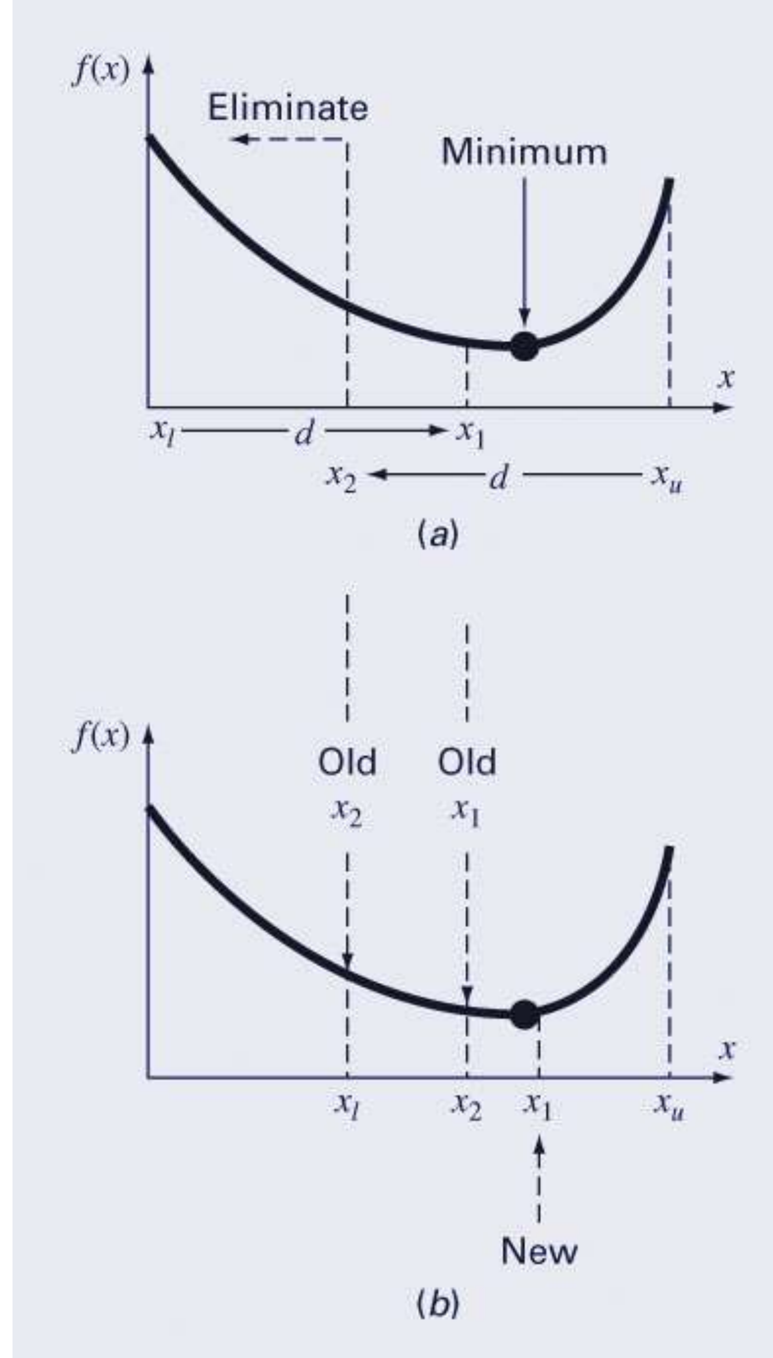
i	x_l	x_2	x_1	x_u	$f(x_2)$	$f(x_1)$
0	0.0000	0.3820	0.6180	1.0000	0.2268	0.1975
1	0.0000	0.2361	0.3820	0.6180	0.1782	0.2268
2	0.2361	0.3820	0.4721	0.6180	0.2268	0.2319
3	0.3820	0.4721	0.5279	0.6180	0.2319	0.2250
4	0.3820	0.4377	0.4721	0.5279	0.2323	0.2319
5	0.3820	0.4164	0.4377	0.4721	0.2311	0.2323

Örnek-2

Minimize $f(x) = 0.5 - x\exp(-x^2)$



x_1	f_1	x_2	f_2
0.764	0.074	1.236	0.232
0.472	0.122	0.764	0.074
0.764	0.074	0.944	0.113
0.652	0.074	0.764	0.074
0.584	0.085	0.652	0.074
0.652	0.074	0.695	0.071
0.695	0.071	0.721	0.071
0.679	0.072	0.695	0.071
0.695	0.071	0.705	0.071
0.705	0.071	0.711	0.071



GOLDEN SECTION METODU

Genel Degerlendirme:

- Her iterasyon için çözüm araliginin (1 - 0.618) = 38% ‘i elimine edilir.

- n iterasyon sonunda çözüm araligi 0.618^n kat küçülmüs olacaktir.

- n = 10 oldugu düsünülürse bulunan çözüm araligi orjinalinin % 1 ’inden küçük olacaktir.

- Verilen çözüm araligi, içinde tek bir minimum noktasi içeren fonksiyonlarda kullanilabilir.

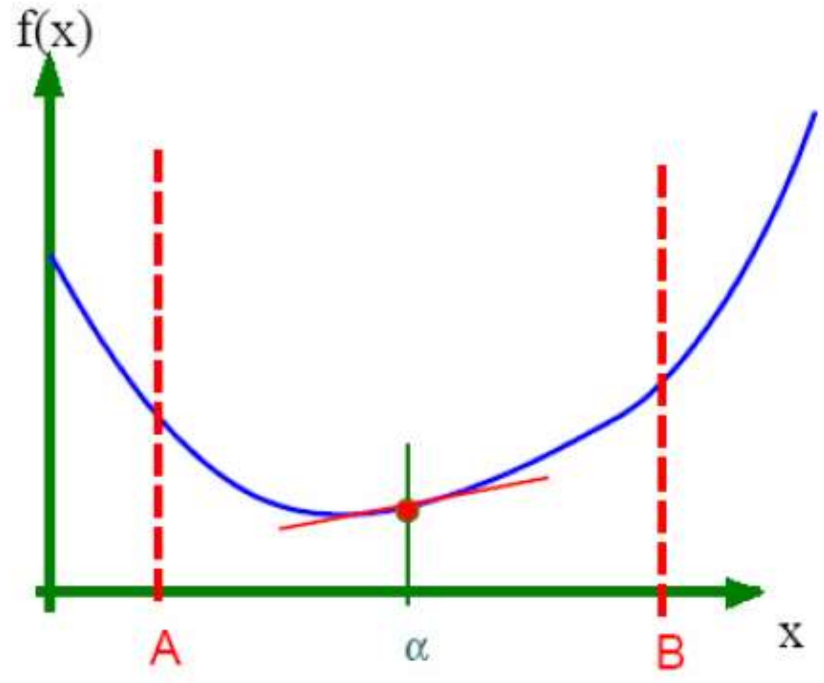
Animasyon

- https://takashiida.floppy.jp/en/education-2/golden-section-algorithm/
- https://github.com/shahrokhx/Golden_Section_Toolbox/blob/main/img/exec_anim.gif

BISECTION METODU

- Aralik daraltma metotlarinda, elde türev bilgisi olursa çözüm hizlandirilabilir.

- Bu gibi durumda araliklari ikiye bölerek (bisection) (daha hizli) daraltmak mümkün olur.



BISECTION (İkiye Bölme) Metodu

Optimizasyonla kök bulma birbirine benzer. Her ikisi de bir fonksiyonun veya türevinin sıfır olduğu yeri bulmaya çalışır.

- Kök bulma: f(x)=0,
- Optimizasyon: f‘(x) = 0

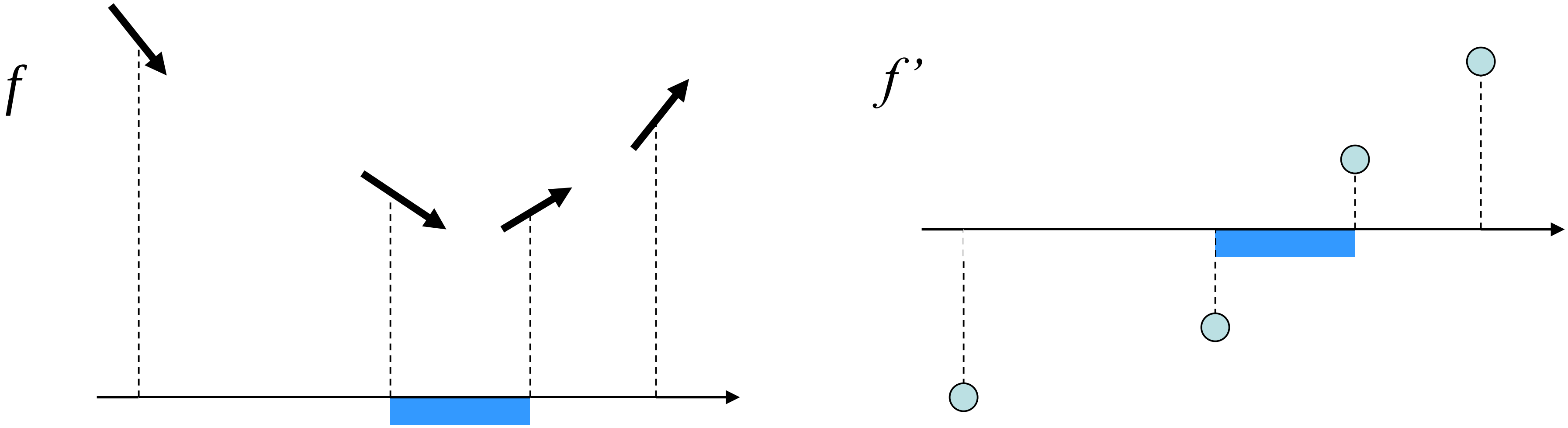
O halde optimum çözüm, f’(x) = 0 probleminin kök bulma yöntemleri ile çözümünden elde edilebilir (eğer fonksiyon türetilebilir ise).

f ’ köklerinin bulunmasına eşdeğerdir…

Minimumu sınırlayan iki nokta (x_a, $x_ü$) bulunur.

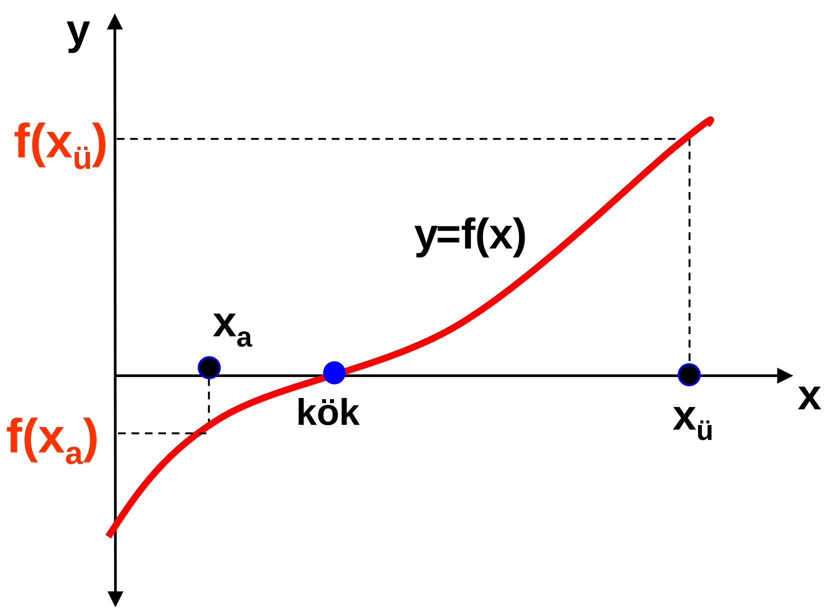
$f'(x_a)<0$ ve $f'(x_ü)>0$

- Golden sectiona benzer ancak türev kullanır



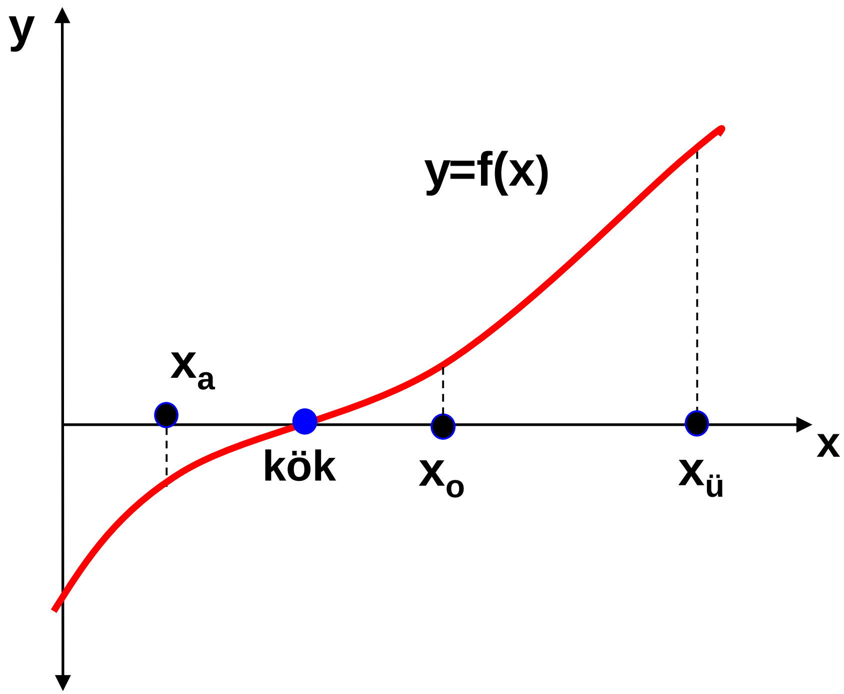
HATIRLAMA : İkiye Bölme Yöntemi

Genel olarak x_a ve $x_ü$ aralığında fks **sürekli** ve $f(x_a)$ ile $f(x_ü)$'nün işaretleri ters ise yani $f(x_a).f(x_ü) < 0$ ise bu aralıkta bir **kök** vardır.



İkiye bölme yönteminde **kökün bulunduğu aralık** adım adım ***daraltılarak*** gerçek köke ulaşılmaya çalışılır.

HATIRLAMA : İkiye Bölme Yöntemi



f(x) = 0 'ı sağlayan kökün içinde bulunduğu aralığın ***alt*** ve ***üst*** ***değeri*** biliniyorsa bu iki değerin **orta noktası** için ***değeri*** bulunabilir.

$$x_o = \frac{x_a + x_ü}{2}$$

HATIRLAMA : İkiye Bölme Yöntemi

İşlem adımları

1) Kökün bulunduğu aralık için x_a ve $x_ü$ değerleri tahmin edilir ve **f(xa).f(xü) < 0** şartı aranır.

2) Üst ve alt değerlerle **orta değer** (x_o) hesaplanır.

$$x_o = \frac{x_a + x_ü}{2}$$

3) $f(x_o)$ değeri hesaplanır

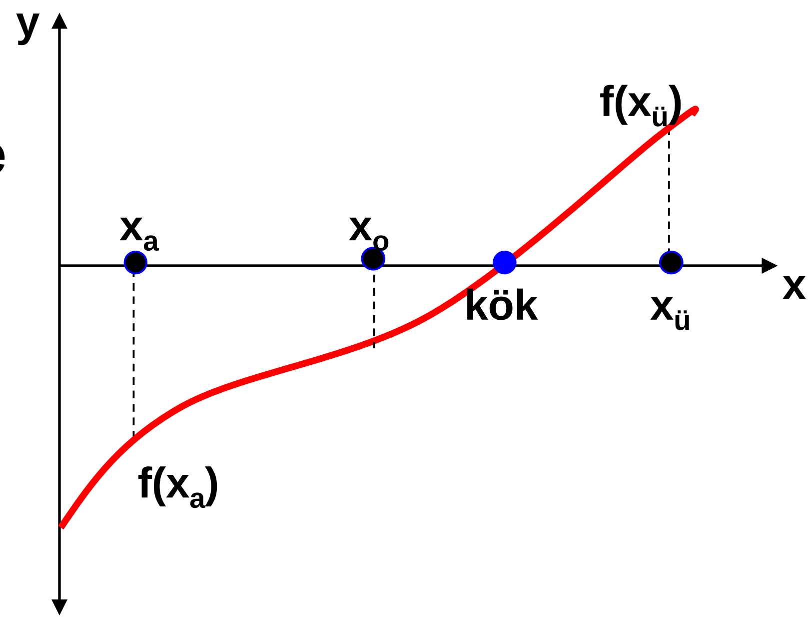
Eğer $f(x_o) = 0$ ise **kök** x_o'dır.

Eğer $f(x_o) \neq 0$ ise **işleme devam** edilir

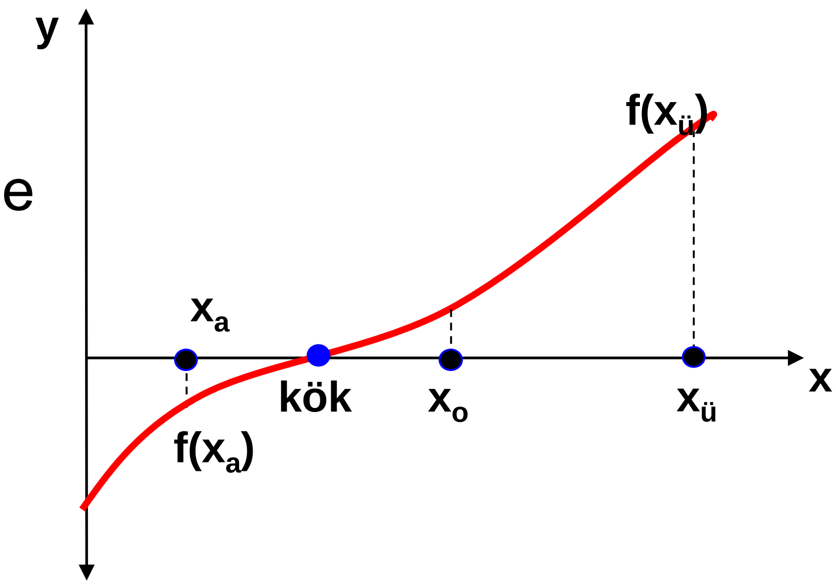
4) $f(x_a)$ hesaplanır

HATIRLAMA : İkiye Bölme Yöntemi

5. a)
   $f(x_a).f(x_o) > 0$ ise
   x_a yerine x_o yazılarak işleme devam edilir



b)
$f(x_a).f(x_o) < 0$ ise
$x_ü$ yerine x_o yazılarak işleme devam edilir.



HATIRLAMA : İkiye Bölme Yöntemi

İşleme son verme

1) $f(x_o)=0$ olunca işleme son verilir

Kök x_o'dır.

2) $|\varepsilon_t| < \varepsilon_k$ ise işleme son verilir.

$$\varepsilon_t = \frac{\text{son deger - bir önceki deger}}{\text{son deger}} = \frac{x_{o,k+1} - x_{o,k}}{x_{o,k+1}}$$

HATIRLAMA : İkiye Bölme Yöntemi

İkiye Bölme Yöntemi Özet Adımlar

Adım 1: $f(x_a)\, f(x_{ü})<0$ şartını sağlayan x_a ve $x_{ü}$ değerlerini tahmin et ve yaklaşım kriteri için uygun bir % sapma değeri (ε_k) kabul et.

Adım 2: aşağıdaki eşitliği kullanarak kök için tahmini bir değer hesapla

$$x_o=(x_a+x_{ü})/2$$

Adım 3: Eğer $f(x_a)\, f(x_{ü})<0 \Rightarrow x_{ü} = x_o$ yazarak Adım 2' ye dön

Eğer $f(x_a)\, f(x_{ü})>0 \Rightarrow x_a = x_o$ yazarak Adım 2' ye dön

Adım 4:

$$\varepsilon_t = \left| \frac{x_o^{yeni} - x_o^{eski}}{x_o^{yeni}} \right| \leq \varepsilon_k$$

oluncaya kadar bu işlemlere devam et.

Buradaki mantıkla f ' köklerinin bulunması işi yapılır …

Minimumu sınırlayan iki nokta (x_a, $x_ü$) bulunur.

$f'(x_a)<0$ ve $f'(x_ü)>0$

ve işlemler f' için açıklanan şekilde uygulanır…

Türevsiz Algoritma

algorithm *MinBisect* $(f, a, b, tol, maxits)$

---

for $k := 1$ **to** $maxits$ **do**

$m := a + (b - a)/2$

if $f(m - \epsilon) < f(m + \epsilon)$ **then**

$b := m + \epsilon$

else

$a := m - \epsilon$

endif

if $|a - b| \leq tol$ **then return** m

endfor

endalg {MinBisect}

Örnek

Max $f(x) = 12x - 3x^4 - 2x^6$

	$df(x)/dx$	$\underline{x}$	$\overline{x}$	New x'	$f(x')$
0					
1					
2					
3	4.09	0.75	1	0.875	7.8439
4	-2.19	0.75	0.875	0.8125	7.8672
5	1.31	0.8125	0.875	0.84375	7.8829
6	-0.34	0.8125	0.84375	0.828125	7.8815
7	0.51	0.828125	0.84375	0.8359375	7.8839

BISECTION METODU

Avantaj:

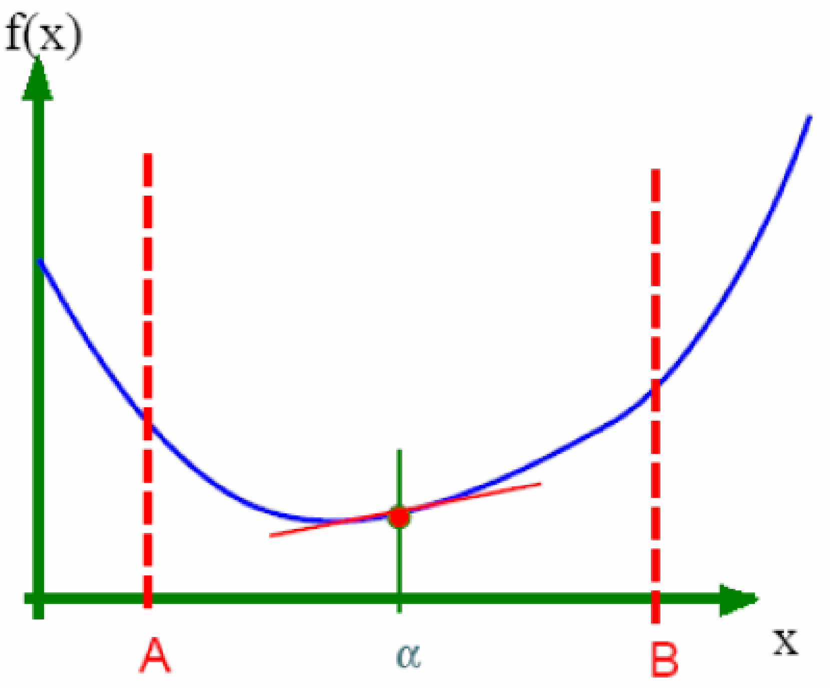
- Daha hizli yakinsama. Örnegin, 10 iterasyondan sonra alan %99 oraninda daraltilmis olur.

Dezavantaj:

- Türev bilgisi gerekli. Türev hesabi genellikle biraz fazla hesap gerektirir.

- Türev bilgisi kullanildigi zaman fonkiyonun sürekli (türetilir) oldugu kabul edilmis olur. Golden section metodunda bu sinirlama yoktur.

- Bisection metodu, golden section metodundan daha az garantilidir.



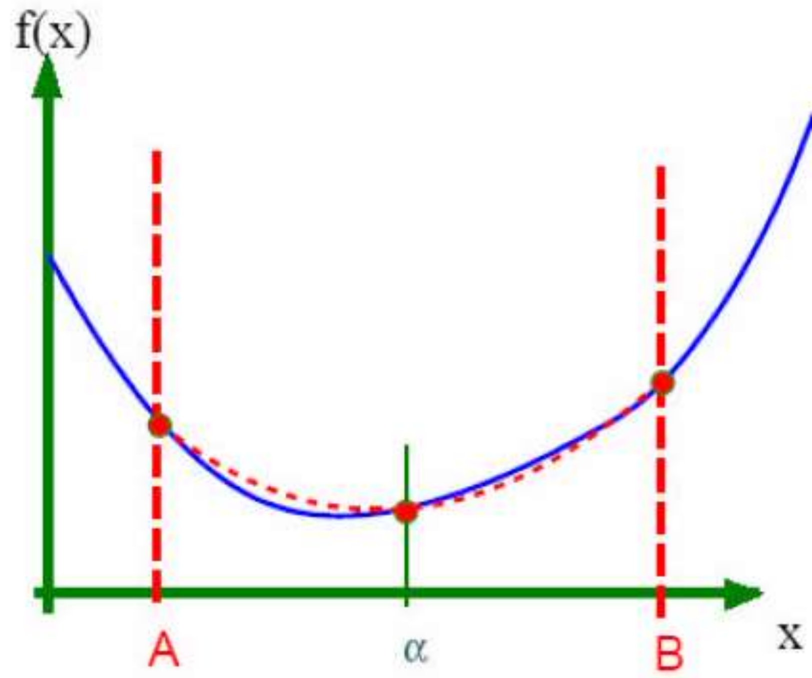
POLINOM METODU

- Araligin uç ve orta noktalarinda fonksiyon degerleri hesaplanir.

- Hesaplanan 3 degerden bir parabol geçirilir.

- Parabolün minimum noktasi (yeni nokta) ve karsi gelen fonksiyon degeri analitik olarak hesaplanir.

- Yeni nokta, eski noktalardan en kötü olanin yerini alarak iterasyon yakinsayana kadar devam edilir.



POLINOM METODU

Avantaj:

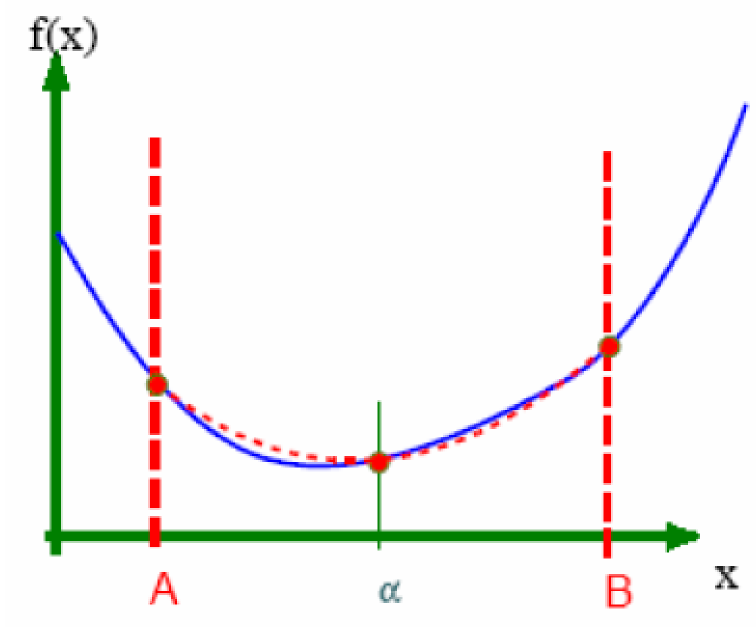
- Fonksiyon düzgünse yakinsama çok hizlidir.

- Türev bilgisi gerekmez. Her iterasyonda fazladan bir nokta için fonksiyon degeri hesaplanir.

Dezavantaj:

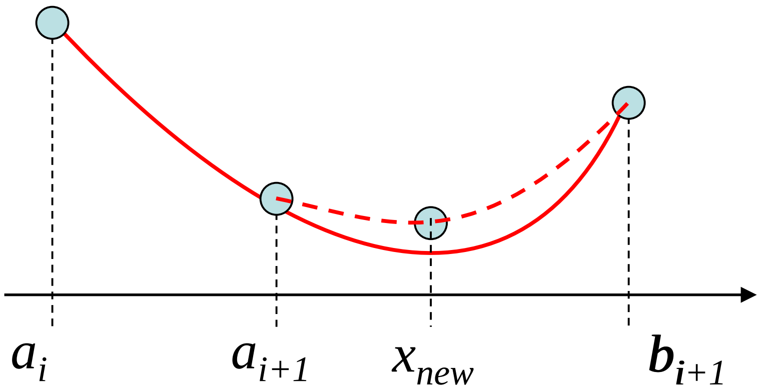
- Algoritma non-konveks fonksiyonlara karsi çok hassastir.

- Optimizasyon için fonksiyonun 2. mertebeden türetilir olmasi lazimdir.



Not: Genel uygulamalar için tavsiye edilmemesine ragmen, özel uygulamalarda faydali olabilir.

$$\tilde{f}(x) = ax^2 + bx + c$$



- Yeni nokta parabolun minimumunda değerlendirilir:

$$\tilde{f}' = 2ax + b = 0 \Rightarrow x_{new} = \frac{-b}{2a}$$

- Minimum için: $a > 0$!
- Mevcut noktaya çok yakınsa x_{new} kaydır

Üç nokta quadratik yaklaşım

- Üç noktadaki fonksiyonun değerine ihtiyaç gösterir (x_1,f_1), (x_2,f_2), (x_3,f_3)
- $q(x)=a_0+a_1(x-x_1)+a_2(x-x_1)(x-x_2)$
- $x=x_1 \rightarrow q=f_1 \quad a_0=f_1$
- $x=x_2 \rightarrow q=f_2 \quad f_2=f_1+a_1(x_2-x_1) \rightarrow a_1=(f_2-f_1)/(x_2-x_1)$
- $x=x_3 \rightarrow q=f_3 \quad f_3=f_1+[(f_2-f_1)/(x_2-x_1)](x_3-x_1)(x_3-x_2) \rightarrow a_2=[(f_3-f_1)/(x_3-x_1)-(f_2-f_1)/(x_2-x_1)]/(x_3-x_2)$
- $dq/dx=0 \rightarrow x^*$ (opt. noktayı buluruz)

Örnek

- $f(x)=(2x^3+16)/x \quad 1 \le x \le 5$
- $x_1=1 \rightarrow f_1=18$
- $x_2=3 \rightarrow f_2=23.33$
- $x_3=5 \rightarrow f_3=53.2$

Formüllerle

$a_0=18,\ a_1=2.67,\ a_2=3.07$

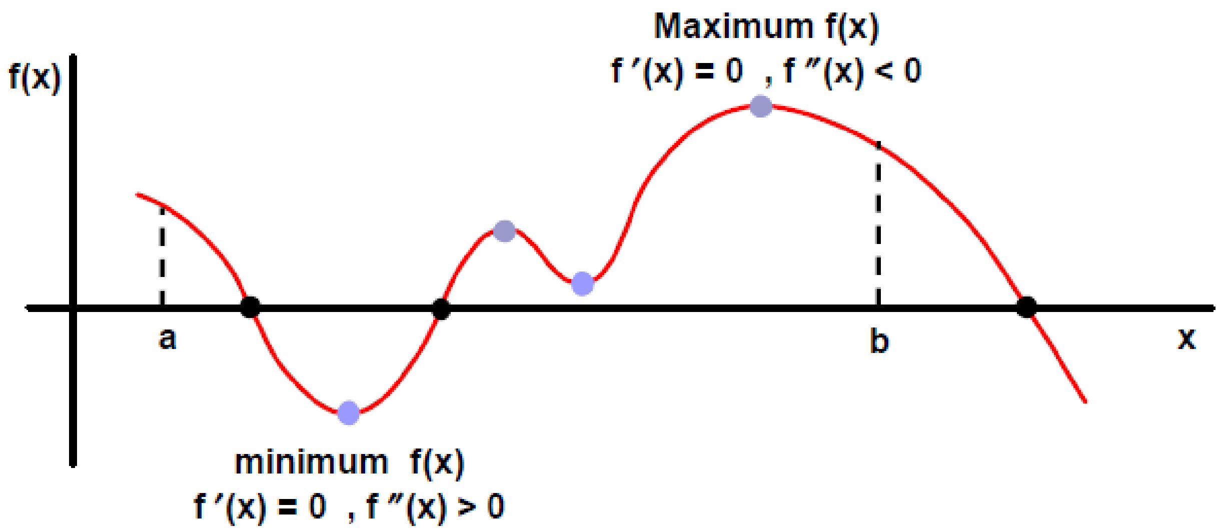
- $q(x)=18+2.67(x-1)+3.07(x-1)(x-3)$
- $dq/dx=0 \rightarrow x^*=[(x_2+x_1)/2](-a_1/2a_2)=1.565$
- Gerçek $x^*=1.5874$

NEWTON-RAPHSON METODU

- Optimizasyonla kök bulma birbirine benzer. Her ikisi de bir fonksiyonun veya türevinin sifir oldugu yeri bulmaya çalisir.

✓ Kök bulma: $f(x)=0$,

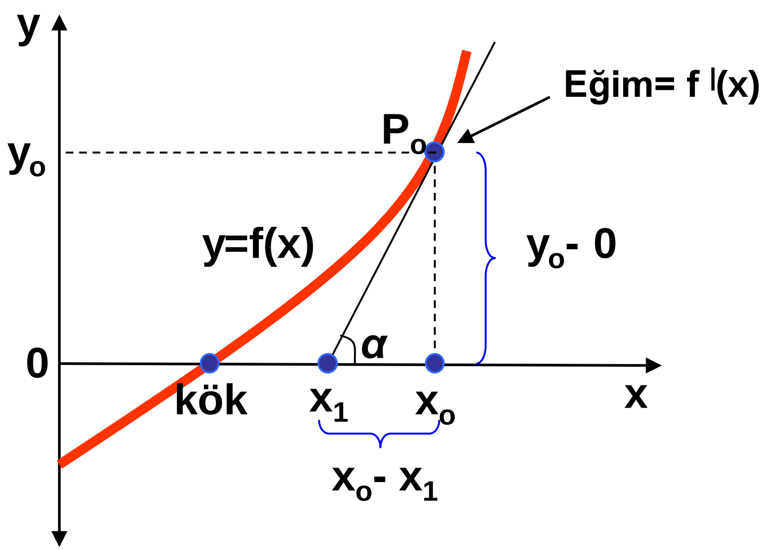
✓ Optimizasyon: $f'(x) = 0$



✓ O halde optimum çözüm, $f'(x) = 0$ probleminin kök bulma yöntemleri ile çözümünden elde edilebilir (eger fonksiyon türetilebilir ise).

Hatırlama (Kök Bulma)

Eğer kökün ilk tahmini x_o ise, $[x_o, f(x_o)]$ noktasındaki teğet uzatılabilir. Uzatılan teğetin x eksenini kestiği nokta genellikle kökün daha iyi bir tahminini verir.

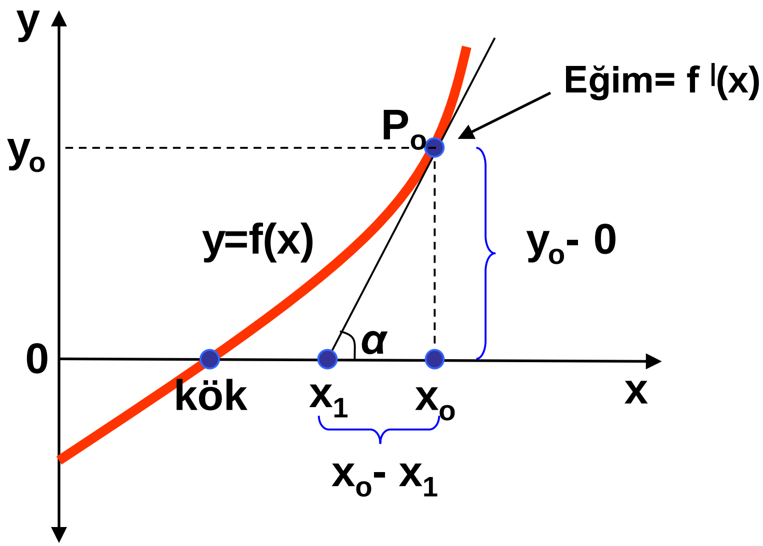


y= f(x) fks.nun x_o değeri y_o= f(x)'dır. P_o noktasından çizilen teğetin eğimi tg(α) 'dır.

$$tg\alpha = \frac{y_o - 0}{x_o - x_1} = \frac{f(x_o) - 0}{x_o - x_1}$$

Newton-Raphson yönteminin grafik gösterimi. x_{i+1} kökünü tahmin edebilmek için x_i 'de fonksiyonun teğeti [yani *f'(x)]* x eksenini kesecek şekilde uzatılmaktadır. Teğetin x eksenini kestiği nokta kök için yeni değerdir

Hatırlama (Kök Bulma)



Aynı zamanda x_o noktasındaki eğim bu noktadaki fonksiyonun türevine eşit olacağından :

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

Buradan x_1 değeri ; $x_1 = x_o - \dfrac{f(x_o)}{f'(x_o)}$

Bir sonraki adımdaki değer: $x_2 = x_1 - \dfrac{f(x_1)}{f'(x_1)}$

En genel şekilde: $\boxed{x_{k+1} = x_k - \dfrac{f(x_k)}{f'(x_k)}}$

Hatırlama (Kök Bulma)

- Newton-Raphson yönteminde iterasyona iki şekilde son verilir.

1. Bulunan **x** değeri için **f(x)** fks.nun değerinin **0**’a yaklaşımına bakarak,
2. **x** değerinin bir önceki hesaplanan değerine **εk** kadar yaklaşmasına bakarak;

- İterasyona son verilir.

NEWTON-RAPHSON METODU

Taylor seri açilimi:

$$f(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2} f''(x_k)(x - x_k)^2 + \ldots$$

f(x)'in x'e göre türevi alinirsa:

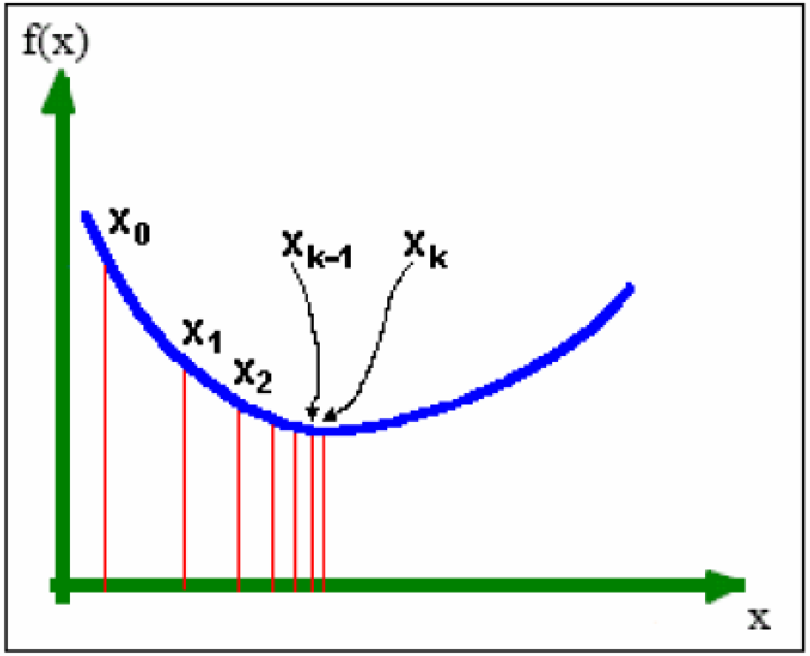
$$f'(x) = 0 + f'(x_k) + f''(x_k)(x - x_k)$$

f '(x)=0 noktasi, Newton-Raphson metodunda optimum noktayi verir:

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$

NEWTON-RAPHSON METODU

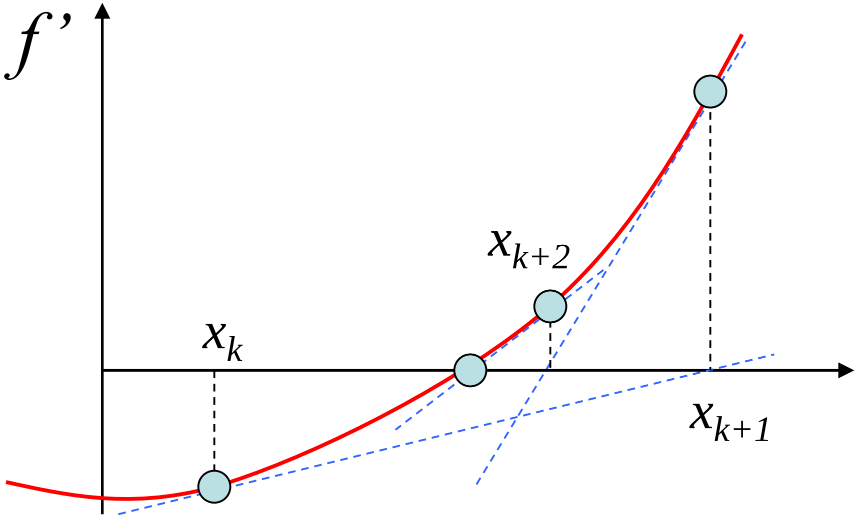
- Optimizasyon problemi iteratif olarak çözer.
- Çözüm için f ”(x)=0 olmamali.

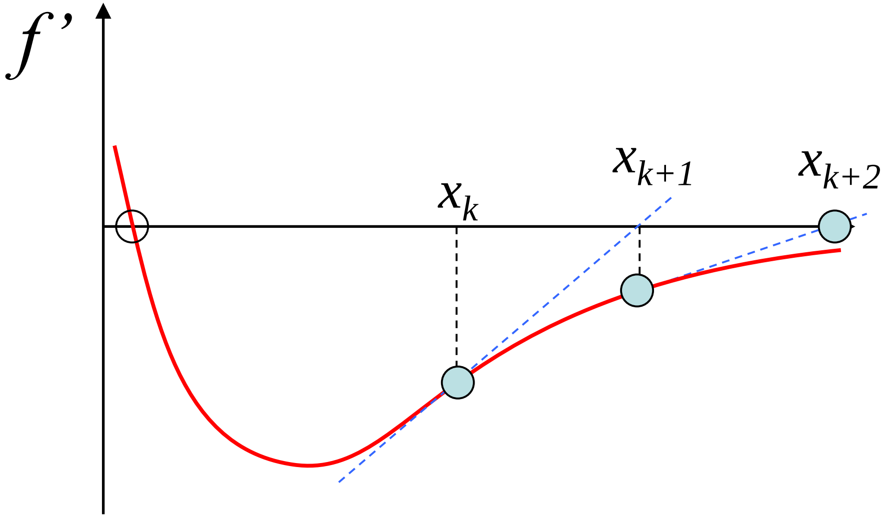


$$x_1 = x_0 - \frac{f'(x_0)}{f''(x_0)} \rightarrow x_2 = x_1 - \frac{f'(x_1)}{f''(x_1)} \rightarrow \ldots \rightarrow x_k = x_{k-1} - \frac{f'(x_{k-1})}{f''(x_{k-1})}$$

NEWTON-RAPHSON METODU

- Tüm metotların en iyi yakınsayanıdır:





Örnek

Max $f(x) = 12x - 3x^4 - 2x^6$

Select $\varepsilon = 0.00001$, and choose $x_1 = 1$.

Iteration i	x_i	$f(x_i)$	$f'(x_i)$	$f''(x_i)$	x_{i+1}
1					
2					
3	0.84003	7.8838	-0.1325	-55.279	0.83763
4	0.83763	7.8839	-0.0006	-54.790	0.83762

Örnek-2

- Use Newton's method to minimize $f(x) = 0.5 - x\exp(-x^2)$
- First and second derivatives of f are given by

$$f'(x) = (2x^2 - 1)\exp(-x^2)$$

and

$$f''(x) = 2x(3 - 2x^2)\exp(-x^2)$$

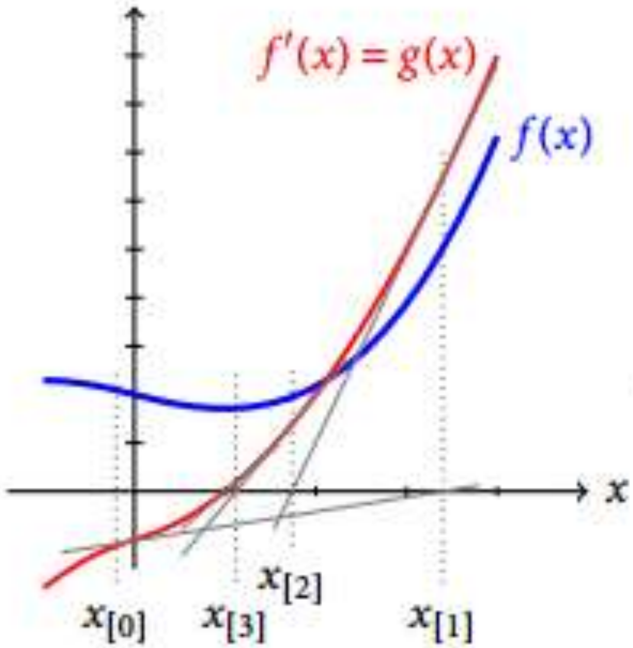
- Newton iteration for zero of f' is given by

$$x_{k+1} = x_k - (2x_k^2 - 1)/(2x_k(3 - 2x_k^2))$$

- Using starting guess $x_0 = 1$, we obtain

x_k	$f(x_k)$
1.000	0.132
0.500	0.111
0.700	0.071
0.707	0.071

Örnek-3



$$f(x) = \frac{1}{3}x^3 + \frac{1}{4}x^2 - 0.5x + 1$$

$$f'(x) = x^2 + \frac{1}{2}x - 0.5$$

$$f''(x) = 2x + \frac{1}{2}$$

x	$f(x)$	$f'(x)$ $= g(x)$	$f''(x)$ $= g'(x)$
$x_{[0]} = -0.1$	1.0522	−0.54	0.3
$x_{[1]} = 1.7$	2.5102	3.24	3.9
$x_{[2]} = 0.8692$	0.9732	0.6902	2.2385
$x_{[3]} = 0.5609$	0.8570	0.0951	1.6218
$x_{[4]} = 0.5023$	0.8542	0.0034	1.5046
$x_{[5]} = 0.5000$	0.8542	0.0000	1.5000

SECANT METODU

Newton metodunda:

- 2. türev bilgisinin elde edilmesi çok hesap gerektirebilir. (özellikle çok degiskenli problemlerde).
- 2. türevin yaklasikliginin kullanilmasi hesaplari çok kolaylastirir.

2. türevin, 1. türev bilgisi kullanarak sonlu farklarla yaklasikligi:

$$f''(x) = \frac{f'(x_k) - f'(x_{k-1})}{x_k - x_{k-1}}$$

2. türev ifadesi Newton-Rapson formülünde yerine konursa Secant metodu elde edilir

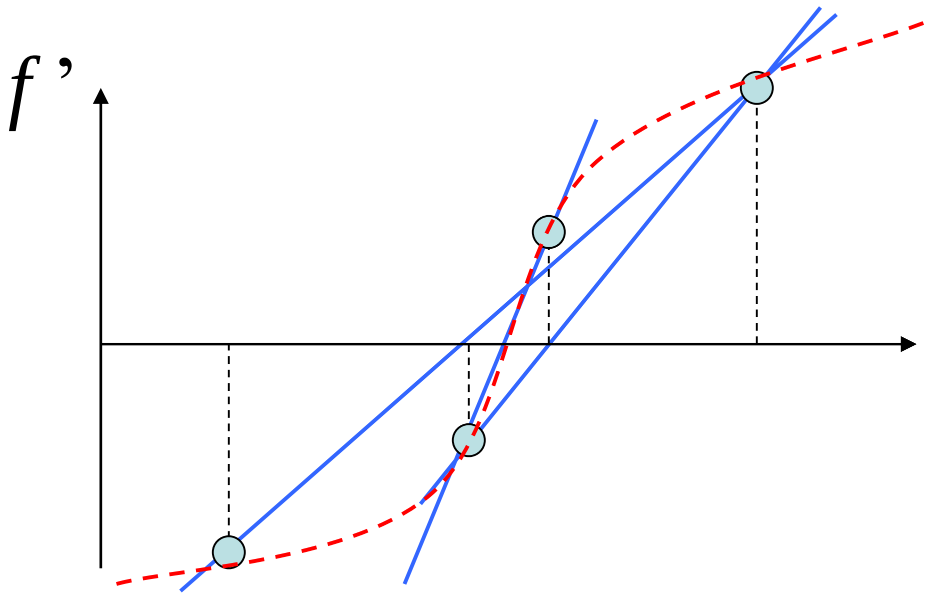
$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f'(x_k) - f'(x_{k-1})} f'(x_k)$$

Not: Secant metodu, yüksek dereceli problemleri Newton-Raphson metodu gibi iteratif çözer.

SECANT METODU

f’ kökünü bulma temellidir.

Lineer interpolasyon kullanır



Örnek

Minimize

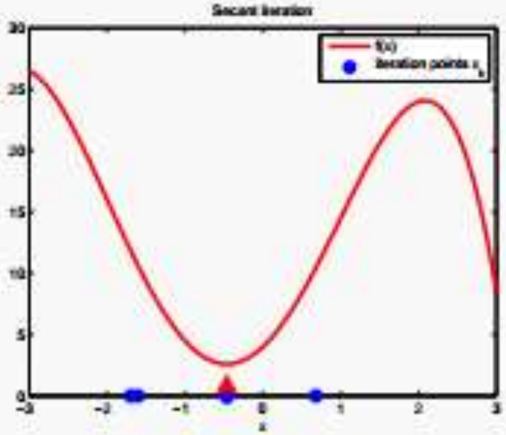
$$f(x) = -\frac{1}{2}x^4 - x^3 + 6x^2 + 6x + 4.$$

Then

- gradient
  $f'(x) = -2x^3 - 3x^2 + 12x + 6$

gives secant iterations

$x_0 = -1.6000, \quad x_1 = 0.6820,$
$x_2 = -0.4345, \quad x_3 = -0.4709,$
$x_4 = -0.4630, \quad x_5 = -0.4630$



initial condition $x_0 = -1.6$
initial condition $x_{-1} = -1.7$
convergence to $x^* = -0.4630$
gradient $f'(x^*) = 5.96 \times 10^{-5}$

- Tek-değişken yöntemler
  - 0. derece (sadece f)
  - 1. derece (f ve f')
  - 2. derece (f, f' ve f'')

• Dichotomous sectioning • Fibonacci sectioning • Golden ratio sectioning • Quadratic interpolation	0^{th} order
• Cubic interpolation • Bisection method • Secant method	1^{st} order
• Newton method	2^{nd} order

METOTLAR HAKKINDA GENEL DEGERLENDIRME

Golden Section, Bisection ve Polinom metotlari çözüm bölgesini sürekli belirli oranda daraltarak optimumu arastirirlar.

Golden Section:

- Fonksiyon degerlerini kullanir.
- yakinsama hizlari çözülen problemin tipinden bagimsizdir.
- Çözüm garantidir (fonksiyon sürekli olsun veya olmasin).

Bisection

- Türev bilgisini kullanir (fonksiyon 1. dereceden sürekli olmak zorunda).
- yakinsama hizi çözülen problemin tipine baglidir.

Polinom metodu

- Fonksiyon degerlerini kullanir.
- yakinsama hizi çözülen problemin tipine baglidir.
- Keskin dönüslü problemlerde zorlanirlar (fonksiyon 1. ve 2. dereceden sürekli olmak zorunda).

METOTLAR HAKKINDA GENEL DEGERLENDIRME

Newton Raphson ve Secant metotlari bir noktadan baslayarak adim adim (iteratif olarak) optimumu arastirirlar.

Newton Raphson:

- Fonksiyonun 1. ve 2. türevlerini kullanir.
- yakinsama hizi çözülen problemin tipine baglidir.
- Keskin dönüslü problemlerde zorlanirlar (fonksiyon 1. ve 2. dereceden sürekli olmak zorunda).
- Çözüm garanti degildir

Secant metodu:

- Fonksiyonun 1. türevini kullanir.
- yakinsama hizi çözülen problemin tipine baglidir.
- Keskin dönüslü problemlerde zorlanirlar (fonksiyon 1. dereceden sürekli olmak zorunda).
- Çözüm garanti degildir.

Not: Optimizasyon problemlerinin çözümünde genellikle Polinom veya Golden Section kullanilir.

Kaynaklar

- Hasan KURTARAN, “Mühendislik tasarımların optimizasyonu”, 2005
- Edwin K.P. Chong Stanislaw H.Zak,**“**An Introduction to Optimization”, 2001, John Wiley&Sons
- Jorge Nocedal, Stephan J. Wright, “Numerical Optimization”, 1999, Springer-Verlag
- R. Fletcher, “Practical Methods of Optimization”, 1987, John Wiley&Sons
- David G.Luenberger, “Introduction to Linear and Nonlinear Programming”, 1987, Addison-Wesley
- S.G. Nash, A. Sofer, “Linear and Nonlinear programming”, 1996, McGraw Hill
- Dimitri P. Bertsekas, “Constrained Optimization and Lagrange Multiplier Methods”, 1982, Academic Press
- Abbas Azimli, “Matematiksel Optimizasyon”, 2011, Papatya Yayınevi