

# Convergence

- We have tried to find some  $x^k$  that converge to the desired point  $x^*$  (most often a local minimizer). The fundamental question is how fast the convergence is.
- Let's define errors at each iteration step as  $|e_k| = |x^k - x^*|$  and  $|e_{k+1}| = |x^{k+1} - x^*|$



## ChatGPT

Bu slayt "Yakınsama" (Convergence) konusunu ele almaktadır. Yakınsama, bir dizi iterasyon sonucunda elde edilen çözümün, aranan optimal noktaya (genellikle yerel bir minimum) doğru ilerlemesi sürecini tanımlar. Slaytta bahsedilen " $x^*$ " terimi, bulmak istediğimiz hedef noktayı ifade eder. Buradaki temel soru, yakınsamanın ne kadar hızlı gerçekleştiğidir. Her iterasyon adımındaki hataları tanımlamak için  $|e_k| = |x_k - x^*|$  ve  $|e_{k+1}| = |x_{k+1} - x^*|$  ifadeleri kullanılır. Burada  $|e_k|$ , k'inci iterasyondaki tahmini çözüm ile gerçek optimal çözüm arasındaki mutlak hata farkını,  $|e_{k+1}|$  ise (k+1)'inci iterasyondaki tahmini çözüm ile gerçek optimal çözüm arasındaki mutlak hata farkını gösterir.



## Convergence

- Linear Convergence :

There exist a constant  $0 < c_1 < 1$

where  $|e_{k+1}| \leq c_1 * |e_k|$

- If  $c_1$  is very close to 0 then superlinear convergence.

- Quadratic Convergence :

There exist a constant  $0 < c_2 < 1$

where  $|e_{k+1}| \leq c_2 * |e_k|^2$

## Convergence Example

- Two methods are given; one of them showing linear convergence and the other one showing quadratic convergence.
- After certain number of steps errors reach 3 digit precision for both of the methods ( $|e_k| < 0,001$  ).
- How many more steps (iterations) will be necessary if we require 12 digits of precision ?
- $c_1 = c_2 = 1/2$



## ChatGPT

Bu slayt, iki farklı türdeki yakınsama türünü açıklamaktadır: Doğrusal ve Karesel Yakınsama.

### Doğrusal Yakınsama:

- Bir sabit  $C_1$  için,  $0 < C_1 < 1$  şartı sağlanırsa ve  $|e_{k+1}| \leq C_1 * |e_k|$  eşitsizliği her  $k$  iterasyonu için geçerliyse, doğrusal yakınsamadan bahsedilir. Eğer  $C_1$  değeri 0'a çok yakınsa, bu süper doğrusal yakınsama olarak adlandırılır.

### Karesel Yakınsama:

- Bir sabit  $C_2$  için,  $0 < C_2 < 1$  şartı sağlanırsa ve  $|e_{k+1}| \leq C_2 * |e_k|^2$  eşitsizliği her  $k$  iterasyonu için geçerliyse, karesel yakınsamadan bahsedilir.

### Yakınsama Örneği:

- İki yöntem verilmiş: biri doğrusal yakınsama gösteriyor, diğeri ise karesel yakınsama gösteriyor.
- Belirli sayıda adımdan sonra hatalar her iki yöntem için de 3 basamaklı bir doğrulukla 0.001'den küçük oluyor.
- Eğer 12 basamaklı bir doğruluk istenirse, kaç adım (iterasyon) daha gerekli olacak?
- Verilen  $C_1$  ve  $C_2$  değerleri için  $C_1 = C_2 = 1/2$  olarak belirtilmiş.

Bu slaytta, doğrusal ve karesel yakınsamanın tanımı ve bu iki yöntemin karşılaştırmalı analizi üzerine odaklanılmıştır. Ayrıca, daha fazla doğruluk gerektiğinde gerekli iterasyon sayısının nasıl hesaplanacağına dair bir soru örneği verilmiştir.

## Convergence Example

- In case of Linear Convergence :  $c_1=0.5$
- $|e_{k+1}| \leq 0.5 * |e_k|$
- $|e_{k+2}| \leq 0.5 * |e_{k+1}|$
- $|e_{k+2}| \leq 0.5 * 0.5 * |e_k|$
- ...
- $|e_{k+n}| \leq 0.5^n * |e_k|$
- $|e_k| \leq 0.001$  (It is given.)
- $|e_{k+n}| \leq 10^{-12}$  (It is required.)
- $|e_{k+n}| \leq 0.5^n * 10^{-3}$
- $0.5^n \approx 10^{-9}$
- $\log_{10} 2^{-n} = \log_{10} 10^{-9}$
- $-n * \log_{10} 2 = -9$  ( $\log_{10} 2 \approx 0.301$ )

## Convergence Example

- In case of quadratic Convergence :  $c_2=0.5$
- $|e_{k+1}| \leq 0.5 * |e_k|^2$
- $|e_{k+2}| \leq 0.5 * |e_{k+1}|^2$
- $|e_{k+2}| \leq 0.5 * (0.5 * |e_k|^2)^2$
- $|e_{k+3}| \leq 0.5 * (0.5 * (0.5 * |e_k|^2)^2)^2$
- $|e_k| \leq 10^{-3}$  (It is given.)
- $|e_{k+n}| \leq 10^{-12}$  (It is required.)
- $|e_k|^2 = (10^{-3})^2 = 10^{-6}$
- $(|e_k|^2)^2 = 10^{-12}$
- $n = 2$

Bu slaytlar, doğrusal ve karesel yakınsamanın matematiksel örneklerini gösteriyor.

#### **Doğrusal Yakınsama Örneği:**

- Sabit  $c_1=0.5$  için,  $|e_{k+1}| \leq 0.5 * |e_k|$  olacak şekilde hataların nasıl azaldığını gösterir.
- Bu ilişki,  $n$  iterasyonundan sonra da devam eder:  $|e_{k+n}| \leq 0.5^n * |e_k|$
- Verilen bir hata eşiği  $|e_k| \leq 0.001$  ve istenen doğruluk düzeyi  $|e_{k+n}| \leq 10^{-12}$  olduğunda,
- Hata eşiğini başlangıç hatası olarak kabul edersek,  $0.5^n \leq 10^{-9}$  olur.
- Bu durumda, logaritma kullanılarak  $n$ 'nin değeri hesaplanır:  $-n * \log_2 = -9 (\log_2 \approx 0.301)$
- Sonuç olarak,  $n$  sayısını bulmak için bu eşitliği çözebiliriz.

#### **Karesel Yakınsama Örneği:**

- Sabit  $c_2=0.5$  için,  $|e_{k+1}| \leq 0.5 * |e_k|^2$  olacak şekilde hataların nasıl azaldığını gösterir.
- Bu durumda, bir sonraki iterasyon için hata şöyle hesaplanabilir:  $|e_{k+2}| \leq 0.5 * |e_{k+1}|^2$
- Bu ilişki devam eder ve  $|e_{k+3}|$ ,  $|e_{k+2}|$ 'nin karesinin yarısı kadar olur.
- Verilen hata eşiği  $|e_k| \leq 10^{-3}$  ve istenen doğruluk düzeyi  $|e_{k+n}| \leq 10^{-12}$  olduğunda,
- İlk iterasyondaki hatayı kullanarak  $|e_{k+2}| \leq (10^{-3})^2 = 10^{-6}$  elde edilir.
- İkinci iterasyonda ise  $|e_{k+2}|^2 \leq 10^{-12}$  olur ve  $n=2$  iterasyonunun yeterli olduğu sonucuna ulaşılır.

Bu slaytlar, yakınsama teorisinin pratikte nasıl uygulanacağını gösteren adım adım bir rehber olarak düşünülebilir. Doğrusal ve karesel yakınsama için gerekli iterasyon sayısını bulmak üzere yapılan hesaplamalar matematiksel olarak gösterilmiştir.





## with N dimensions

- $f(x) = (1/2) * x^T * q * x + b^T * x + c$
- $f: \mathbb{R}^n \rightarrow \mathbb{R}^1$
- $q \rightarrow n * n$
- $b \rightarrow n * 1$
- $c \rightarrow 1 * 1$
- $x \rightarrow n * 1$
- 
- $df = q * x + b$
- $ddf = q$
- `opt_Ndim_v2.m`

## To avoid local minimums

- Random restart
- Non Derivative Techniques



## ChatGPT

Bu slaytta, N boyutlu bir optimizasyon problemi ve yerel minimumlardan kaçınma teknikleri anlatılmaktadır.

### N Boyutlu Optimizasyon Problemi:

- Verilen fonksiyon  $f(x) = \frac{1}{2}x^T \cdot q \cdot x + b^T \cdot x + c$ , burada  $x$  bir N boyutlu vektör,  $q$  bir N x N matrisi,  $b$  bir N boyutlu vektör ve  $c$  bir sabittir.
- $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , yani  $f$  fonksiyonu N boyutlu gerçel sayı uzayından gerçel sayılara bir fonksiyondur.
- Türev  $df = q \cdot x + b$  ve ikinci türev  $ddf = q$  ile temsil edilir.
- Ayrıca bir MATLAB dosyası olan "opt\_Ndim\_v2.m" dosyası da bahsedilmiş.

### Yerel Minimumlardan Kaçınma:

- Rastgele yeniden başlatma (Random restart), yerel minimumlarda takılıp kalmamak için kullanılan bir yöntemdir.
- Türevsiz Teknikler (Non Derivative Techniques), fonksiyonun türevini hesaplamadan optimum noktaya ulaşmayı hedefleyen tekniklerdir.

Bu slayt, karmaşık optimizasyon problemlerinde optimizasyon tekniklerinin nasıl uygulanabileceğine dair genel bir bakış sağlar ve özellikle yerel minimum noktalarında optimizasyonun durmaması için kullanılabilecek stratejilere değinir.