T.C.

ONDOKUZ MAYIS ÜNİVERSİTESİ LİSANSÜSTÜ

EĞİTİM ENSTİTÜSÜ BİLGİSAYAR

MÜHENDİSLİĞİ

OPTİMİZASYON-BM615

Optimizasyon Yöntemlerinden Quadratic ve Cubic İnterpolasyon Raporu

FIRAT KAAN BİTMEZ -

23281855

SAMSUN, 2024-2025 Eğitim Öğretim Yılı Güz Yarıyılı

1. GİRİŞ

Bu rapor, optimizasyon yöntemlerinden **quadratic** ve **cubic interpolasyon** tekniklerini detaylı bir şekilde ele alarak hazırlanılmıştır. Optimizasyon problemlerinde, fonksiyonların minimum noktalarını belirlemek için kullanılan interpolasyon yöntemleri, özellikle analitik türev bilgisine ulaşmanın zor olduğu durumlarda kritik bir öneme sahiptir. Özellikle bu iki yöntemin nasıl çalıştığını teorik ve pratik olarak gözlemledik en sonunda yöntemleri kıyaslama yaparak çalışmamızı tamamladık.

İncelenen yöntemler hakkında kısaca bahsetmem gerekirse:

1. **Quadratic interpolasyon**, ikinci dereceden polinomlar kullanarak minimum noktayı tahmin eder ve basit yapısı sayesinde hızlı bir yakınsama sağlar.
2. **Cubic interpolasyon** ise üçüncü dereceden polinomlar ve türev bilgilerinden faydalanarak daha hassas sonuçlar elde eder.

2. TEORİK ARKA PLAN

Bu bölümde, quadratic ve cubic interpolasyon yöntemlerinin matematiksel altyapısı, uygulama prensipleri ve kullanım detayları açıklanmaktadır. Bölümde ayrıca unimodal fonksiyonların tanımı ve bu fonksiyonların optimizasyon yöntemlerindeki önemi de ele alınmıştır. Açıklamalar, **Engineering Optimization** kitabında sunulan bilgiler doğrultusunda geliştirilmiştir.

2.1. Unimodal ve Multimodal Fonksiyonlar

Unimodal Fonksiyon Nedir?

Bir fonksiyon, yalnızca bir global minimum veya maksimum noktası içeriyorsa **unimodal fonksiyon** olarak tanımlanır. Örneğin, $f(x)=(x-2)^2$ fonksiyonu, x=2'de bir minimuma sahiptir ve unimodal bir fonksiyondur.

Multimodal Fonksiyon Nedir?

Bir fonksiyon, birden fazla minimum veya maksimum noktası içeriyorsa **multimodal fonksiyon** olarak tanımlanır. Örneğin, f(x)=sin(x) fonksiyonu, birçok maksimum ve minimum içerdiği için multimodaldir.

Farkları:

- **Unimodal:** Tek bir global minimum veya maksimum bulunur.
- **Multimodal:** Birden fazla yerel ve/veya global minimum veya maksimum bulunabilir.

Quadratic ve cubic interpolasyon gibi optimizasyon yöntemleri, bir fonksiyonun **unimodal** olduğunu varsayar. Bu varsayım:

- Aralığın sistematik olarak daraltılmasına,
- İşlemlerin daha hızlı ve doğru bir şekilde tamamlanmasına olanak tanır.

Eğer fonksiyon multimodal ise, her bir unimodal bölge ayrı ayrı incelenmeli ve optimize edilmelidir. Bu nedenle, optimizasyon algoritmalarında unimodal varsayımı kritik bir temel oluşturur.

2.2. Quadratic İnterpolasyonun Teorik Temelleri

Polinomların Rolü:

Quadratic interpolasyon, bir fonksiyonun ikinci dereceden bir polinom ($h(x)=ax2+bx+c$) ile yaklaşık modellenmesine dayanır. Bu süreçte, polinomun türevini sıfıra eşitleyerek minimum nokta hesaplanır:

$$h'(x)=2ax+b \implies x_{min}=-2ab$$

Matematiksel Dayanak:

- **İkinci Dereceden Polinom Kullanımı:** Bu yaklaşım, bir fonksiyonun üç noktasındaki değerlerini **(x_1, x_2, x_3)** kullanarak katsayıların çözümünü sağlar.
- **Interpolasyonun Etkinliği:** Fonksiyonun unimodal olduğu varsayımı altında, bu yöntem minimum noktayı yüksek bir doğrulukla tahmin edebilir.

Quadratic interpolasyonun temel avantajı, türev bilgisine ihtiyaç duymadan çalışabilmesidir. Bununla birlikte:

- Katsayıların hesaplanması sırasında doğruluk, seçilen noktaların aralığına bağlıdır.
- Çok keskin eğriler veya osilasyonlar içeren fonksiyonlarda yeterince hassas olmayabilir.

2.3. Cubic İnterpolasyonun Teorik Temelleri

Polinomların Rolü:

Cubic interpolasyon, bir fonksiyonun üçüncü dereceden bir polinom ($h(x)=ax3+bx2+cx+d$) ile modellenmesine dayanır. Bu yöntem, türev bilgilerini de kullanarak daha hassas sonuçlar elde eder.

Matematiksel Dayanak:

Bir üçüncü dereceden polinomun türevi sıfıra eşitlenerek minimum nokta hesaplanır:

$h'(x)=3ax2+2bx+c \implies$ Minimum noktalar türev köklerinden bulunur.

Türev Kullanımının Önemi:

Cubic interpolasyon, türev bilgilerinin dahil edilmesiyle quadratic interpolasyondan daha karmaşık ancak daha doğru sonuçlar sağlar. Ancak bu, aşağıdaki matematiksel gereklilikleri doğurur:

- Türevin hesaplanabilir olması.
- Dört farklı noktadaki (x0,x1,x2,x3) türev ve fonksiyon değerlerinden hareketle katsayıların çözümü: a,b,c,d katsayıları bir matris denkleminden çözülür.

2.4. Quadratic ve Cubic İnterpolasyon Arasındaki Teorik Farklılıklar

Polinom Derecesi:

- **Quadratic İnterpolasyon:** İkinci dereceden polinom kullanır ve yalnızca üç nokta gerektirir.
- **Cubic İnterpolasyon:** Üçüncü dereceden polinom kullanır ve dört nokta ile türev bilgisine ihtiyaç duyar.

Yaklaşım Doğruluğu:

- **Quadratic:** Daha az bilgi ile çalışır, ancak karmaşık fonksiyonlarda doğruluk azalır.
- **Cubic:** Daha fazla bilgi kullanarak doğruluğu artırır, ancak hesaplama maliyeti yüksektir.

Teorik Dayanak:

- Quadratic interpolasyon, fonksiyonun yalnızca ikinci dereceden bir modele uygun olduğunu varsayar. Cubic interpolasyon ise, fonksiyonun daha yüksek dereceli bir modelle daha iyi temsil edilebileceği durumlar için uygundur.

Avantaj ve Dezavantajların Matematiksel Temelleri:

Özellik	**Quadratic İnterpolasyon**	**Cubic İnterpolasyon**
Türev Gereksinimi	Gerektirmez	Gerekir
Hesaplama Maliyeti	Düşük	Yüksek
Doğruluk	Orta	Yüksek
Teorik Karmaşıklık	Basit	Karmaşık

2.5. Quadratic ve Cubic İnterpolasyonun Uygulama Alanlarına Katkıları

Optimizasyon Sürecindeki Roller:

- **Quadratic İnterpolasyon:** Daha düşük doğruluk gerektiren, hızlı ve maliyet etkin çözümler sağlar. Bu, özellikle fonksiyonların türevlerinin bilinmediği durumlar için uygundur.
- **Cubic İnterpolasyon:** Daha yüksek doğruluk gerektiren uygulamalarda, özellikle fonksiyonun türevlerinin mevcut olduğu durumlarda tercih edilir.

3. QUADRATIC VE CUBIC INTERPOLASYON YÖNTEMLERİNİN UYGULAMA ADIMLARI

3.1. Quadratic İnterpolasyon Yönteminin Adımları

Quadratic interpolasyon, bir fonksiyonu ikinci dereceden bir polinom $h(x)=ax2+bx+c$ ile modelleyerek minimum noktanın belirlenmesini sağlar. Bu yöntem türev bilgisi gerektirmez ve yalnızca üç farklı noktadaki fonksiyon değerlerini kullanır.

Adım Adım Quadratic İnterpolasyon Süreci:

1. **Başlangıç Noktalarının Seçimi:**
    - Optimizasyon yapılacak fonksiyonun, minimum değerinin bulunabileceği üç farklı nokta seçilir: $x0,x1,x2$.
    - Bu noktaların aralık içinde uygun bir dağılımda olması kritik öneme sahiptir. Noktalar fonksiyonun gerçek minimum değerine yakın seçilirse yöntem daha hızlı yakınsama sağlar.
2. **Fonksiyon Değerlerinin Hesaplanması:**
    - Seçilen her bir başlangıç noktası için fonksiyon değerleri hesaplanır: $f(x0),f(x1),f(x2)$.
    - Bu değerler, ikinci dereceden polinomun katsayılarını hesaplamak için kullanılacaktır.
3. **Polinom Katsayılarının Hesaplanması:**
    - İkinci dereceden polinomun katsayıları (a,b,c) şu formüllerle hesaplanır:

      $a=(x2-x1)(x2-x0)(f(x2)-f(x1))-(x1-x0)(x2-x0)(f(x1)-f(x0))$

      $b=x2-x1f(x2)-f(x1)-a(x1+x2)$

      $c=f(x0)-ax02-bx0$

4. **Minimum Noktanın Tahmini:**
    - İkinci dereceden polinomun minimum noktası, türev sıfıra eşitlenerek hesaplanır: $h'(x)=2ax+b \implies x_{min}=-2ab$.
5. **Yeni Nokta Seçimi:**
    - Hesaplanan xmin, başlangıç noktalarından biri ile değiştirilir. Bu seçim, xmin 'in en düşük fonksiyon değerine sahip olduğu duruma göre yapılır.
    - Örneğin, $f(x_{min})<f(x0),f(x1),f(x2)$'den daha düşükse, xmin eski noktalardan biriyle değiştirilir.
6. **Yakınsama Kontrolü:**
    - Minimum noktanın hesaplanan değerinin, belirlenen bir tolerans (epsilon) içinde olup olmadığı kontrol edilir.
    - Eğer $|x_{min,new}-x_{min,prev}|<Epsilon(\epsilon)$ koşulu sağlanıyorsa işlem sonlandırılır. Aksi takdirde süreç baştan tekrarlanır.

Dikkat Edilmesi Gerekenler:

- Başlangıç noktalarının doğru seçilmesi yöntemin başarısı için önemlidir.
- Aralığın çok geniş ya da çok dar olması yöntemin başarısız olmasına neden olabilir.
- Fonksiyonun sürekli olması yeterlidir; türev bilgisi gerektirmez.

3.2. Cubic İnterpolasyon Yönteminin Adımları

Cubic interpolasyon, bir fonksiyonu üçüncü dereceden bir polinom $h(x)=ax3+bx2+cx+d$ ile modelleyerek minimum noktayı belirler. Bu yöntem, quadratic interpolasyondan farklı olarak türev bilgisini de kullanır ve dört farklı noktadaki değerler üzerinden işlem yapar.

Adım Adım Cubic İnterpolasyon Süreci:

1. **Başlangıç Noktalarının Seçimi:**
    - Dört farklı başlangıç noktası seçilir: $x0,x1,x2,x3$.
    - Noktaların aralık içinde minimum noktaya yakın yerleştirilmesi önemlidir. Bu seçim, interpolasyonun doğruluğunu doğrudan etkiler.
2. **Fonksiyon ve Türev Değerlerinin Hesaplanması:**
    - Seçilen dört noktada fonksiyon değerleri hesaplanır:
    - $f(x0),f'(x0),f(x1),f'(x1),f(x2),f'(x2),f(x3),f'(x3)$
    - Türev değerleri mevcut değilse ya da hesaplanamıyorsa cubic interpolasyon uygulanamaz.

3. **Polinom Katsayılarının Çözülmesi:**
    - Üçüncü dereceden polinomun dört bilinmeyenli (a,b,c, d) katsayıları, seçilen noktalar üzerinden bir matris çözüm yöntemiyle bulunur.
    - Polinom katsayılarının çözümünde şu denklem sistemi kullanılır:

$$h(x) = \begin{bmatrix} [x0^3 & x0^2 & x0 & 1] \\ [x1^3 & x1^2 & x1 & 1] \\ [3x0^2 & 2x0 & 1 & 0] \\ [3x1^2 & 2x1 & 1 & 0] \end{bmatrix} * \begin{bmatrix} [a] \\ [b] \\ [c] \\ [d] \end{bmatrix} = \begin{bmatrix} [f(x0)] \\ [f(x1)] \\ [f'(x0)] \\ [f'(x1)] \end{bmatrix}$$

    - Bu denklem, minimum noktanın bulunduğu türev köklerini elde etmek için çözülür.
4. **Minimum Noktanın Hesaplanması:**
    - Çözülen katsayılarla birlikte türev sıfıra eşitlenerek minimum nokta tahmin edilir: $h'(x)=3ax2+2bx+c \implies xmin=solve(3ax2+2bx+c=0)$.
    - Burada, türev denkleminin birden fazla kökü olabilir. Köklerin sayısı ve değerleri, fonksiyonun yapısına bağlıdır. Minimum noktayı belirlemek için her kökün fonksiyon değerleri kontrol edilir.
5. **Yeni Nokta Seçimi:**
    - Minimum noktanın bulunduğu yeni değerler, başlangıç noktalarından biriyle değiştirilir. Bu seçim, minimum değere en yakın olan ve en düşük fonksiyon değerine sahip olan nokta olarak yapılır.
6. **Yakınsama Kontrolü:**
    - Minimum noktanın hesaplanan değerinin belirlenen tolerans (ϵ\epsilon$\epsilon$) içinde olup olmadığı kontrol edilir.
    - Eğer $|xmin,new-xmin,prev|<\epsilon$ koşulu sağlanıyorsa işlem durdurulur. Aksi takdirde süreç baştan tekrarlanır.

Dikkat Edilmesi Gerekenler:

- Cubic interpolasyon, türev bilgisine bağımlıdır. Bu bilgi eksikse yöntem kullanılamaz.
- Yüksek doğruluk sağlamasına rağmen, hesaplama maliyeti quadratic interpolasyona göre daha yüksektir.

- Uygun başlangıç noktalarının seçimi ve doğru türev değerlerinin hesaplanması, yöntemin başarısını belirler.

4. KOD ANALİZİ

4.1 Quadratic Interpolation

Quadratic Interpolation fonksiyonu, bir unimodal fonksiyonun minimum noktasını bulmak için şu adımları izler:

Adım 1: Parametrelerin Tanımlanması

```
def quadratic_interpolation(f, x0, x1, x2, tolerance=1e-5, max_iter=100):
```

- **f**: Minimumu bulunacak fonksiyon.
- **x0, x1, x2**: Üç başlangıç noktası.
- **tolerance**: Yakınsama için tolerans sınırı.
- **max_iter**: İzin verilen maksimum iterasyon sayısı.

Adım 2: Başlangıç Bilgilerinin Yazdırılması

```
print(f"Başlangıç noktaları: x0={x0}, x1={x1}, x2={x2}\n")
```

Kullanıcıya başlangıç noktaları hakkında bilgi verilir. Bu noktalar, algoritmanın başlangıç aralığını belirler.

Adım 3: Iterasyon Döngüsü

Kod, iteratif bir yapı kullanarak minimum noktayı bulmaya çalışır.

```
for iteration in range(max_iter):
```

Her iterasyon sırasında:

1. **Fonksiyon Değerlerini Hesapla:**

```
f0, f1, f2 = f(x0), f(x1), f(x2)
```

- f(x0), f(x1), f(x2) değerleri hesaplanır.
- Bu değerler, katsayıların hesaplanmasında kullanılır.

2.Parabol Katsayılarının Hesaplanması:

```
numerator = (f0 * (x1**2 - x2**2) + f1 * (x2**2 - x0**2) + f2 * (x0**2 -
x1**2))
denominator = (f0 * (x1 - x2) + f1 * (x2 - x0) + f2 * (x0 - x1))
```

Numerator ve denominator ifadeleri, minimum noktayı tahmin etmek için kullanılır.

3. **Minimum Noktanın Hesaplanması:**

```
x_min = 0.5 * (numerator / denominator)
f_min = f(x_min)
```

- Parabolün minimum noktası formülle hesaplanır.
- f(x_min) ile bu noktadaki fonksiyon değeri elde edilir.

4. Yakınsama Kontrolü:

```
if abs(x_min - x1) < tolerance and abs(f_min - f1) < tolerance:
    return x_min
```

x_min ile önceki nokta x1 arasındaki fark tolerans değerinin altına düşerse algoritma sonuçlanır.

5. Aralığın Güncellenmesi:

- Algoritma, x_min'e göre başlangıç noktalarını günceller

```
if x_min < x1:
    if f_min < f1:
        x2, x1 = x1, x_min
    else:
        x0 = x_min
else:
    if f_min < f1:
        x0, x1 = x1, x_min
    else:
        x2 = x_min
```

6.Bilgi Yazdırma:

```
print(f"Yeni noktalar: x0={x0}, x1={x1}, x2={x2}")
print(f"Aralık boyutları: |x2 - x0| = {abs(x2 - x0):.6f}\n")
```

Her iterasyonda güncellenen noktalar ve aralığın boyutu yazdırılır.

Adım 4: Sonuç veya Uyarı

Eğer maksimum iterasyon sınırı aşılırsa:

```
print("Maksimum iterasyon sayısına ulaşıldı, yakınsama sağlanamadı.\n")
```

Algoritma, minimumu bulamadan durur.

Cubic Interpolation

Cubic Interpolation algoritması, Quadratic Interpolation'a benzer bir yapıya sahiptir, ancak türev bilgilerini ve dört başlangıç noktasını kullanır.

Adım 1: Parametrelerin Tanımlanması

```
def cubic_interpolation(f, x0, x1, x2, x3, tolerance=1e-5, max_iter=100):
```

Bu algoritma, dört başlangıç noktası alır.

Adım 2: Katsayıların Hesaplanması

1. **Fonksiyon Değerlerini ve Katsayıları Hesapla**

```
A = np.array([
    [x0**3, x0**2, x0, 1],
    [x1**3, x1**2, x1, 1],
    [x2**3, x2**2, x2, 1],
    [x3**3, x3**2, x3, 1]
])
b = np.array([f0, f1, f2, f3])
coeffs = np.linalg.solve(A, b)
```

- Katsayılar, matris çözümü ile hesaplanır.
- np.linalg.solve, matris denklemini çözmek için kullanılır.

2. **Türev ve Kritik Noktalar:**

```
derivative = lambda x: 3 * a * x**2 + 2 * b * x + c
critical_points = np.roots([3 * a, 2 * b, c])
```

Polinomun türevi alınır ve kritik noktalar (roots) hesaplanır.

Adım 3: Minimum Noktayı Bulma

Kritik noktalar arasında minimumu bulmak için:

```
if np.isreal(root) and x0 < root < x3:
    real_root = np.real(root)
    f_root = f(real_root)
    if f_root < min_value:
        x_min = real_root
        min_value = f_root
```

- np.isreal: Gerçek kökleri kontrol eder.
- f(real_root): Minimum noktadaki fonksiyon değerini hesaplar.

Adım 4: Yakınsama Kontrolü

Yakınsama kontrolü Quadratic Interpolation ile benzerdir:

```
if abs(x_min - x1) < tolerance and abs(min_value - f1) < tolerance:
    return x_min
```

Adım 5: Güncellemeler

Yeni aralık ve noktalar güncellenir:

```
if x_min < x1:
    if min_value < f1:
        x3, x2, x1 = x2, x1, x_min
    else:
        x0 = x_min
else:
    if min_value < f1:
        x0, x1, x2 = x1, x_min, x2
```

```
    else:
        x3 = x_min
```

4.3. Quadratic ve Cubic Karşılaştırılması

Özellik	Quadratic İnterpolasyon	Cubic İnterpolasyon
Türev Kullanımı	Gerekli değil	Gerekli
Katsayı Hesaplama	3 noktadan polinom katsayıları	4 noktadan ve türev bilgisi ile katsayılar
Hassasiyet	Orta düzey	Yüksek
Hesaplama Maliyeti	Düşük	Yüksek
Kod Karmaşıklığı	Basit	Daha karmaşık

5.KOD ÇIKTILARININ DEĞERLENDİRİLMESİ

Kod Çıktıları

Quadratic Interpolation Çıktısı:

```
Quadratic Test:
==== Quadratic Interpolation Başladı ====
Başlangıç noktaları: x0=0, x1=2, x2=4

Iterasyon 1: f(x0)=4.000000, f(x1)=0.000000, f(x2)=4.000000
Tahmini minimum: x_min=2.000000, f(x_min)=0.000000
1. iterasyonda 2.000000 değerine yakınsandı.
```

Cubic Interpolation Çıktısı:

```
Cubic Test:
==== Cubic Interpolation Başladı ====
Başlangıç noktaları: x0=0, x1=1, x2=2, x3=3

Iterasyon 1: f(x0)=1.000000, f(x1)=5.000000, f(x2)=3.000000, f(x3)=1.000000
Tahmini minimum: x_min=1.000000, f(x_min)=5.000000
1. iterasyonda 1.000000 değerine yakınsandı.
```

Quadratic Interpolation:

- Doğru bir şekilde çalışmış, minimum noktayı hızlı ve doğru bir şekilde bulmuştur.
- Bu yöntem, türev bilgisi gerektirmediği ve yalnızca üç nokta ile çalıştığı için daha basit fonksiyonlarda etkili bir çözüm sunar.

Cubic Interpolation:

- Doğru teorik minimum noktayı (x=1.0x=1.0x=1.0) bulmuş ancak fonksiyon değerini tahmin ederken beklenenden farklı bir sonuç vermiştir.
- Bu durum, algoritmanın türev bilgisine olan hassasiyetine ve başlangıç noktalarına bağlı olarak iyileştirilebilir.
- Daha fazla iterasyon ile daha doğru sonuçlara ulaşılabilir.

6. AVANTAJLAR, DEZAVANTAJLAR VE KULLANIM ALANLARI

Yöntem	Avantajlar	Dezavantajlar
Quadratic İnterpolasyon	Basit, hızlı, türev gerektirmez.	Daha karmaşık fonksiyonlar için yetersiz.
Cubic İnterpolasyon	Daha hassas, türev bilgisi mevcutsa etkili.	Daha karmaşık, türev bilgisi gerektirir.

7. SONUÇ

Bu çalışma kapsamında quadratic ve cubic interpolasyon yöntemleri teorik ve pratik düzeyde analiz edilmiştir. Her iki yöntem de optimizasyon problemleri için önemli araçlardır ve farklı senaryolara uygun özellikler sunar:

1. Quadratic interpolasyon, türev bilgisi gerektirmeyen, hızlı ve basit bir yöntemdir. Özellikle düşük doğruluk gerektiren veya başlangıç aşaması tahminleri için idealdir.
2. Cubic interpolasyon, türev bilgisi gerektirir ve doğruluk açısından üstün bir yöntemdir. Daha karmaşık ve hassas sonuçlar gerektiren mühendislik problemlerinde tercih edilir.

Sonuç olarak, quadratic ve cubic interpolasyon yöntemleri, mühendislik ve matematikteki optimizasyon problemlerine etkili çözümler sunar. Hangi yöntemin kullanılacağına, problemin doğasına, fonksiyonun özelliklerine ve istenen doğruluk seviyesine bağlı olarak karar verilmelidir.

8.KAYNAKLAR

- Rao, S. S. (1996). ***Engineering Optimization: Theory and Practice*** (3rd ed.). Wiley-Interscience. (Page: 310-340)
- Jain, A. (2023). Handling Missing Data in Pandas (Part 2). ***Medium***. https://medium.com/@abhishekjainindore24/handling-missing-data-in-pandas-part-2-8d06c4075bb6
- Bitmez, F. K. Optimization Repository. ***GitHub***. https://github.com/firatkaanbitmez/Optimization
- Rey, P. (2023). Interpolation. ***Medium***. https://medium.com/@polluxrey/interpolation-54cf494e07f3