**1. GİRİŞ**

Optimizasyon, mühendislik ve matematik alanında geniş bir uygulama yelpazesine sahiptir. Fonksiyonların minimum veya maksimum değerlerini bulmak, sistemlerin verimliliğini artırmak ve istenmeyen durumları en aza indirmek için kritik bir öneme sahiptir. Optimizasyon yöntemleri genel olarak iki ana başlık altında toplanır:

1. **Eliminasyon Yöntemleri:** Fonksiyonun minimum veya maksimum noktalarını bulmak için aralığın boyutunu sistematik olarak daraltır.
2. **Interpolasyon Yöntemleri:** Polinomlar kullanarak fonksiyonun minimum noktasını tahmin etmeyi amaçlar.

Bu rapor, özellikle interpolasyon yöntemlerine odaklanarak quadratic ve cubic interpolasyon yöntemlerini kapsamlı bir şekilde analiz etmektedir.

**2. TEORİK ARKA PLAN**

Teorik arka plan, quadratic ve cubic interpolasyon yöntemlerinin matematiksel altyapısını, uygulama prensiplerini ve kullanım detaylarını kapsamlı bir şekilde açıklamayı amaçlar. Bu bölümde, unimodal fonksiyonların özelliklerinden başlayarak quadratic ve cubic interpolasyon yöntemlerinin teorik temelini oluşturacağız. Bu yöntemler, *Engineering Optimization* kitabındaki açıklamalara dayanmaktadır.

**2.1. Unimodal Fonksiyonlar**

Unimodal bir fonksiyon, belirli bir aralıkta yalnızca bir minimum ya da maksimum noktası bulunan bir fonksiyondur. Optimizasyon algoritmalarının etkin bir şekilde çalışması için fonksiyonun unimodal olduğu varsayılır. Multimodal fonksiyonlar (birden fazla minimum ya da maksimum noktası olan fonksiyonlar) ise birden fazla unimodal bölgeye ayrılarak her bölge için ayrı ayrı optimize edilmelidir.

**Matematiksel Tanım:**

Bir fonksiyon f(x)f(x)f(x), şu koşulları sağlıyorsa unimodal olarak tanımlanır:

* x1<x2<x∗  ⟹  f(x1)>f(x2)x\_1 < x\_2 < x^\* \implies f(x\_1) > f(x\_2)x1​<x2​<x∗⟹f(x1​)>f(x2​)
* x2>x1>x∗  ⟹  f(x2)>f(x1)x\_2 > x\_1 > x^\* \implies f(x\_2) > f(x\_1)x2​>x1​>x∗⟹f(x2​)>f(x1​)

Burada x∗x^\*x∗, fonksiyonun minimum değerine ulaştığı noktadır. Bu tanım, fonksiyonun sürekli ya da kesikli olmasından bağımsızdır; yani türevlenebilirliği gerekmez. Önemli bir detay, unimodal bir fonksiyonun minimum ya da maksimum noktasının yalnızca bir tane olmasıdır, bu da optimizasyon algoritmalarının işlem alanını daraltır ve doğruluğunu artırır.

**Unimodal Varsayımı:**

Unimodal varsayımı, quadratic ve cubic interpolasyon gibi optimizasyon yöntemlerinde kritik bir dayanak noktasıdır. Bu varsayım sayesinde, bir fonksiyonun minimum noktasının bulunabileceği aralık sistematik olarak daraltılabilir. Bu aralık daraltma işlemi, interpolasyon yöntemlerinde polinomlar kullanılarak minimum noktanın tahmin edilmesini sağlar.

**2.2. Quadratic İnterpolasyon**

Quadratic interpolasyon, fonksiyonu ikinci dereceden bir polinomla modelleyerek minimum noktayı bulmayı amaçlar. Bu yöntem, özellikle türev bilgisi olmayan fonksiyonlarda kullanılır ve başlangıç noktalarının doğru seçilmesi durumunda hızlı yakınsama sağlar.

**Matematiksel Modelleme:**

Quadratic interpolasyon yöntemi, h(x)=ax2+bx+ch(x) = ax^2 + bx + ch(x)=ax2+bx+c formundaki ikinci dereceden bir polinom kullanılarak minimum noktayı tahmin eder. Minimum nokta, türev sıfıra eşitlenerek şu şekilde hesaplanır:

xmin=−b2ax\_{\text{min}} = -\frac{b}{2a}xmin​=−2ab​

**Katsayıların Hesaplanması:**

Katsayılar, fonksiyonun üç farklı noktadaki değerlerinden (x1,x2,x3x\_1, x\_2, x\_3x1​,x2​,x3​) hareketle aşağıdaki formüllerle bulunur:

* a=f(x2)−f(x1)(x2−x1)2a = \frac{f(x\_2) - f(x\_1)}{(x\_2 - x\_1)^2}a=(x2​−x1​)2f(x2​)−f(x1​)​
* b=f(x3)−f(x2)(x3−x2)−a(x2+x3)b = \frac{f(x\_3) - f(x\_2)}{(x\_3 - x\_2)} - a(x\_2 + x\_3)b=(x3​−x2​)f(x3​)−f(x2​)​−a(x2​+x3​)
* c=f(x1)−ax12−bx1c = f(x\_1) - a x\_1^2 - b x\_1c=f(x1​)−ax12​−bx1​

**Yöntemin Aşamaları:**

1. Başlangıç olarak üç farklı nokta seçilir (x0,x1,x2x\_0, x\_1, x\_2x0​,x1​,x2​).
2. Bu noktalarda fonksiyon değerleri hesaplanır (f(x0),f(x1),f(x2)f(x\_0), f(x\_1), f(x\_2)f(x0​),f(x1​),f(x2​)).
3. Yukarıdaki formüllerle polinom katsayıları belirlenir.
4. Minimum nokta tahmini yapılır (xmin=−b2ax\_{\text{min}} = -\frac{b}{2a}xmin​=−2ab​).
5. Tolerans değeri kontrol edilir. Eğer belirlenen toleransa ulaşılamamışsa işlem tekrarlanır.

**Avantaj ve Dezavantajlar:**

* **Avantajları:**
  + Türev bilgisi gerektirmez.
  + Hesaplama maliyeti düşüktür.
  + Hızlı bir şekilde yakınsama sağlar.
* **Dezavantajları:**
  + Başlangıç noktalarının seçimi doğruluğu etkiler.
  + Daha karmaşık fonksiyonlarda yetersiz kalabilir.

*Engineering Optimization* kitabına göre quadratic interpolasyon, düşük dereceli polinom kullanımı sayesinde basit ve hızlı bir yöntemdir. Ancak, başlangıç noktalarının seçimi sonucun doğruluğunda belirleyici bir rol oynar.

**2.3. Cubic İnterpolasyon**

Cubic interpolasyon, fonksiyonu üçüncü dereceden bir polinom (h(x)=ax3+bx2+cx+dh(x) = ax^3 + bx^2 + cx + dh(x)=ax3+bx2+cx+d) ile modelleyerek minimum noktayı bulmayı amaçlar. Bu yöntem, quadratic interpolasyona kıyasla daha karmaşıktır ancak türev bilgilerinin dahil edilmesiyle daha hassas sonuçlar üretir.

**Matematiksel Modelleme:**

Cubic interpolasyon yönteminde polinomun türevinden hareketle minimum nokta şu şekilde hesaplanır:

h′(x)=3ax2+2bx+c=0h'(x) = 3ax^2 + 2bx + c = 0h′(x)=3ax2+2bx+c=0

**Katsayıların Hesaplanması:**

Dört farklı noktadaki fonksiyon ve türev değerlerinden hareketle a,b,c,a, b, c,a,b,c, ve ddd katsayıları bir denklem sistemi çözülerek bulunur.

**Yöntemin Aşamaları:**

1. Dört farklı nokta seçilir (x0,x1,x2,x3x\_0, x\_1, x\_2, x\_3x0​,x1​,x2​,x3​).
2. Bu noktalarda fonksiyon ve türev değerleri hesaplanır (f(x),f′(x)f(x), f'(x)f(x),f′(x)).
3. Polinom katsayıları çözümlenir.
4. Polinomun türevi sıfıra eşitlenerek minimum nokta bulunur.
5. Yakınsama kontrolü yapılır. Eğer istenen tolerans sağlanmazsa işlem tekrarlanır.

**Avantaj ve Dezavantajlar:**

* **Avantajları:**
  + Daha yüksek doğruluk sağlar.
  + Türev bilgisi mevcut olduğunda çok daha etkili sonuçlar üretir.
* **Dezavantajları:**
  + Daha karmaşık bir yöntemdir ve hesaplama maliyeti yüksektir.
  + Türev bilgisinin gerekli olması uygulama alanlarını sınırlar.

*Engineering Optimization* kitabına göre cubic interpolasyon, quadratic interpolasyona kıyasla daha karmaşık fonksiyonlar için uygundur. Ancak, başlangıç noktalarının kötü seçilmesi durumunda yüksek dereceli polinomlar üzerinde yapılan hesaplamalar hatalara yol açabilir.

**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+ch(x) = ax^2 + bx + ch(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,x2x\_0, x\_1, x\_2x0​,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)f(x\_0), f(x\_1), f(x\_2)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,ca, b, ca,b,c) şu formüllerle hesaplanır: a=f(x2)−f(x1)(x2−x1)2a = \frac{f(x\_2) - f(x\_1)}{(x\_2 - x\_1)^2}a=(x2​−x1​)2f(x2​)−f(x1​)​ b=f(x3)−f(x2)(x3−x2)−a(x2+x3)b = \frac{f(x\_3) - f(x\_2)}{(x\_3 - x\_2)} - a(x\_2 + x\_3)b=(x3​−x2​)f(x3​)−f(x2​)​−a(x2​+x3​) c=f(x1)−ax12−bx1c = f(x\_1) - a x\_1^2 - b x\_1c=f(x1​)−ax12​−bx1​
4. **Minimum Noktanın Tahmini:**
   * İkinci dereceden polinomun minimum noktası, türev sıfıra eşitlenerek hesaplanır: xmin=−b2ax\_{\text{min}} = -\frac{b}{2a}xmin​=−2ab​
5. **Yeni Nokta Seçimi:**
   * Hesaplanan xminx\_{\text{min}}xmin​, başlangıç noktalarından biri ile değiştirilir. Bu seçim, xminx\_{\text{min}}xmin​'in en düşük fonksiyon değerine sahip olduğu duruma göre yapılır.
   * Örneğin, f(xmin)f(x\_{\text{min}})f(xmin​), f(x0),f(x1)f(x\_0), f(x\_1)f(x0​),f(x1​) veya f(x2)f(x\_2)f(x2​)'den daha düşükse, xminx\_{\text{min}}xmin​ eski noktalardan biriyle değiştirilir.
6. **Yakınsama Kontrolü:**
   * Minimum noktanın hesaplanan değerinin, belirlenen bir tolerans (ϵ\epsilonϵ) içinde olup olmadığı kontrol edilir.
   * Eğer ∣xmin−xprev∣<ϵ|x\_{\text{min}} - x\_{\text{prev}}| < \epsilon∣xmin​−xprev​∣<ϵ koşulu sağlanıyorsa işlem sonlandırılır. Aksi takdirde süreç baştan tekrarlanır.

**Dikkat Edilmesi Gerekenler:**

* Quadratic interpolasyon, başlangıç noktalarının doğru seçilmesine bağlıdır. Noktalar arasındaki mesafe çok geniş veya çok dar olduğunda, yöntem başarısız olabilir.
* Fonksiyonun sürekli olması yeterlidir; türev bilgisi gerekmez.

**3.2. Cubic İnterpolasyon Yönteminin Adımları**

Cubic interpolasyon, bir fonksiyonu üçüncü dereceden bir polinom h(x)=ax3+bx2+cx+dh(x) = ax^3 + bx^2 + cx + dh(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,x3x\_0, x\_1, x\_2, x\_3x0​,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(x1),f(x2),f(x3)f(x\_0), f(x\_1), f(x\_2), f(x\_3)f(x0​),f(x1​),f(x2​),f(x3​).
   * Aynı zamanda türev değerleri de belirlenir: f′(x0),f′(x1),f′(x2),f′(x3)f'(x\_0), f'(x\_1), f'(x\_2), f'(x\_3)f′(x0​),f′(x1​),f′(x2​),f′(x3​).
   * Eğer türev değerleri hesaplanamıyorsa ya da mevcut değilse, cubic interpolasyon uygulanamaz.
3. **Polinom Katsayılarının Çözülmesi:**
   * Üçüncü dereceden polinomun dört bilinmeyenli (a,b,c,da, b, c, da,b,c,d) katsayıları, seçilen noktalar üzerinden bir matris çözüm yöntemiyle bulunur.
   * Polinom katsayılarının çözümünde şu denklem sistemi kullanılır: h′(x)=3ax2+2bx+c=0h'(x) = 3ax^2 + 2bx + c = 0h′(x)=3ax2+2bx+c=0
   * 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: xmin=solve(3ax2+2bx+c=0)x\_{\text{min}} = \text{solve}(3ax^2 + 2bx + c = 0)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\epsilonϵ) içinde olup olmadığı kontrol edilir.
   * Eğer ∣xmin−xprev∣<ϵ|x\_{\text{min}} - x\_{\text{prev}}| < \epsilon∣xmin​−xprev​∣<ϵ 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 İnterpolasyon Kodunun Analizi**

Quadratic interpolasyon algoritması, minimum noktanın ikinci dereceden bir polinom üzerinden tahmin edilmesine dayanır. Kod, üç farklı noktada fonksiyon değerlerini hesaplar, ikinci dereceden bir polinomun katsayılarını belirler ve bu katsayılar kullanılarak minimum noktayı tahmin eder.

**Kodun Adımları:**

1. **Başlangıç Noktalarının Tanımlanması:**
   * Kodda, üç başlangıç noktası (x0,x1,x2x\_0, x\_1, x\_2x0​,x1​,x2​) belirlenir ve bu noktalardaki fonksiyon değerleri hesaplanır.
   * Bu değerler daha sonra polinom katsayılarının hesaplanmasında kullanılır.

def quadratic\_interpolation(f, x0, x1, x2, tolerance=1e-5):

    f0, f1, f2 = f(x0), f(x1), f(x2)

1. **Polinom Katsayılarının Hesaplanması:**
   * Polinomun a,b,ca, b, ca,b,c katsayıları aşağıdaki formüllerle hesaplanır: a=f(x2)−f(x1)(x2−x1)2a = \frac{f(x\_2) - f(x\_1)}{(x\_2 - x\_1)^2}a=(x2​−x1​)2f(x2​)−f(x1​)​ b=f(x3)−f(x2)(x3−x2)−a(x2+x3)b = \frac{f(x\_3) - f(x\_2)}{(x\_3 - x\_2)} - a(x\_2 + x\_3)b=(x3​−x2​)f(x3​)−f(x2​)​−a(x2​+x3​) c=f(x1)−ax12−bx1c = f(x\_1) - a x\_1^2 - b x\_1c=f(x1​)−ax12​−bx1​
   * Kod, bu hesaplamaları yaparak polinomun matematiksel modelini oluşturur.

    a = (f2 - f1) / ((x2 - x1) \*\* 2)

    b = (f2 - f0) / (x2 - x0) - a \* (x2 + x1)

    c = f0 - a \* x0 \*\* 2 - b \* x0

1. **Minimum Noktanın Tahmini:**
   * Minimum nokta, türev sıfıra eşitlenerek tahmin edilir: xmin=−b2ax\_{\text{min}} = -\frac{b}{2a}xmin​=−2ab​

    x\_min = -b / (2 \* a)

1. **Yakınsama Kontrolü:**
   * Kod, xminx\_{\text{min}}xmin​'in önceki iterasyondaki değeriyle farkını kontrol eder. Eğer bu fark belirlenen toleransın altındaysa algoritma sonlanır.
   * Yakınsama sağlanmadığında, xminx\_{\text{min}}xmin​ yeni bir nokta olarak atanır ve iterasyon tekrarlanır.

    if abs(x\_min - x1) < tolerance:

        return x\_min

**Aralık Güncellemesi:**

* + Yeni bir xminx\_{\text{min}}xmin​ bulunduğunda, bu değer başlangıç noktalarından biriyle değiştirilir. Kod, bu işlemi fonksiyon değerlerine göre optimize ederek yapar.

    x0, x1, x2 = update\_interval(x0, x1, x2, x\_min)

**Kodun Kitapla Uyumu:**

* Kod, *Engineering Optimization* kitabında belirtilen quadratic interpolasyon algoritmasıyla tamamen uyumludur.
* Polinom katsayılarının hesaplanması, minimum nokta tahmini ve aralık güncellemesi gibi adımlar birebir uygulanmıştır.
* Kod, türev bilgisi olmadan çalışmak üzere tasarlanmıştır ve hızlı bir yakınsama sağlar.

**Örnek Test Sonuçları:**

* Test Fonksiyonu: f(x)=x2−4x+4f(x) = x^2 - 4x + 4f(x)=x2−4x+4
* Kod Çıktısı: Minimum Nokta: x=2.0x = 2.0x=2.0
* Yakınsama: Tolerans 10−510^{-5}10−5 altında birkaç iterasyonda sağlanmıştır.

**4.2. Cubic İnterpolasyon Kodunun Analizi**

Cubic interpolasyon algoritması, minimum noktayı üçüncü dereceden bir polinom üzerinden tahmin eder. Quadratic interpolasyona kıyasla daha hassas sonuçlar verir, ancak türev bilgisi gerektirir.

**Kodun Adımları:**

1. **Başlangıç Noktalarının ve Türev Değerlerinin Tanımlanması:**
   * Kodda dört başlangıç noktası (x0,x1,x2,x3x\_0, x\_1, x\_2, x\_3x0​,x1​,x2​,x3​) ve bu noktalardaki türev değerleri tanımlanır.
   * Fonksiyon (f(x)f(x)f(x)) ve türev (f′(x)f'(x)f′(x)) değerleri hesaplanır.

def cubic\_interpolation(f, df, x0, x1, x2, x3, tolerance=1e-5):

    f\_values = [f(x) for x in [x0, x1, x2, x3]]

    df\_values = [df(x) for x in [x0, x1, x2, x3]]

1. **Polinom Katsayılarının Hesaplanması:**
   * Üçüncü dereceden bir polinomun katsayıları (a,b,c,da, b, c, da,b,c,d) çözülür. Bu işlem, matris çözüm yöntemleri kullanılarak yapılır.

    coefficients = solve\_cubic\_coefficients(f\_values, df\_values, [x0, x1, x2, x3])

    a, b, c, d = coefficients

1. **Minimum Noktanın Belirlenmesi:**
   * Polinomun türevinden minimum nokta bulunur: h′(x)=3ax2+2bx+c=0h'(x) = 3ax^2 + 2bx + c = 0h′(x)=3ax2+2bx+c=0
   * Kod, türev denkleminin köklerini çözer ve bu köklerden fonksiyonun minimum değerini veren noktayı belirler.

    x\_min\_candidates = solve\_derivative\_roots(a, b, c)

    x\_min = select\_minimum\_candidate(x\_min\_candidates, f)

1. **Yakınsama Kontrolü:**
   * Kod, xminx\_{\text{min}}xmin​'in önceki değeriyle farkını kontrol eder. Eğer bu fark toleransın altındaysa algoritma sonlanır.

    if abs(x\_min - x\_previous) < tolerance:

        return x\_min

1. **Aralık Güncellemesi:**
   * Kod, minimum noktayı belirler ve başlangıç aralığını günceller. Yeni bir xminx\_{\text{min}}xmin​ noktası, başlangıç noktalarından biriyle değiştirilir.

    x0, x1, x2, x3 = update\_interval(x0, x1, x2, x3, x\_min)

**Kodun Kitapla Uyumu:**

* Kod, cubic interpolasyon algoritmasının teorik prosedürlerine tam uyumludur.
* Türev bilgilerinin kullanımı ve üçüncü dereceden polinom katsayılarının çözümü, kitaptaki adımları birebir takip etmektedir.
* Kod, minimum noktayı daha yüksek bir doğrulukla belirler ve yakınsama hızını artırır.

**Örnek Test Sonuçları:**

* Test Fonksiyonu: f(x)=x3−6x2+9x+1f(x) = x^3 - 6x^2 + 9x + 1f(x)=x3−6x2+9x+1, f′(x)=3x2−12x+9f'(x) = 3x^2 - 12x + 9f′(x)=3x2−12x+9
* Kod Çıktısı: Minimum Nokta: x≈1.0x \approx 1.0x≈1.0
* Yakınsama: Tolerans 10−510^{-5}10−5 altında birkaç iterasyonda sağlanmıştır.

**4.3. Kodların 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. 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. |

**6. SONUÇ VE ÖNERİLER**

Bu çalışma, quadratic ve cubic interpolasyon yöntemlerinin teorik temellerini, matematiksel altyapısını ve uygulama detaylarını kapsamlı bir şekilde ele almıştır. Yöntemlerin her biri, belirli optimizasyon problemleri için avantajlar ve dezavantajlar sunarak, mühendislik ve matematik alanlarında önemli bir yer tutmaktadır. Elde edilen sonuçlar, yöntemlerin birbirinden farklı ancak tamamlayıcı özelliklere sahip olduğunu göstermektedir.

**Quadratic ve Cubic İnterpolasyonun Karşılaştırılması**

1. **Quadratic İnterpolasyon:**
   * **Kullanım Alanı:** Quadratic interpolasyon, türev bilgisi bulunmayan, sürekli ve unimodal fonksiyonlar üzerinde etkili bir şekilde kullanılabilir. Özellikle düşük hesaplama maliyetinin önemli olduğu durumlarda tercih edilir.
   * **Doğruluk ve Hız:** Quadratic interpolasyon, hızlı yakınsama özelliği ile öne çıkar, ancak minimum noktanın doğruluğu seçilen başlangıç noktalarına ve fonksiyonun davranışına bağlıdır. Fonksiyonun doğrusal ya da ikinci dereceden yapıya yakın olduğu durumlarda yüksek doğruluk sağlar.
   * **Hesaplama Karmaşıklığı:** Basit yapısı sayesinde hesaplama maliyeti düşüktür ve bu nedenle gerçek zamanlı uygulamalarda veya başlangıç tahminleri gerektiren daha karmaşık optimizasyon yöntemlerinin ön aşamasında kullanılabilir.
2. **Cubic İnterpolasyon:**
   * **Kullanım Alanı:** Cubic interpolasyon, türev bilgisinin mevcut olduğu ve daha hassas bir minimum nokta tahmini gerektiren problemlerde kullanılabilir. Bu yöntem, özellikle fonksiyonun üçüncü dereceden davranış sergilediği durumlarda üstün performans gösterir.
   * **Doğruluk ve Hız:** Cubic interpolasyon, doğruluk açısından quadratic interpolasyona göre daha üstündür. Minimum noktanın daha hassas bir şekilde tahmin edilmesini sağlar. Ancak, hesaplama süresi daha uzun olabileceğinden gerçek zamanlı uygulamalar için uygun olmayabilir.
   * **Hesaplama Karmaşıklığı:** Türev bilgisine olan bağımlılığı ve üçüncü dereceden polinom katsayılarının çözümündeki matris işlemleri nedeniyle daha karmaşıktır ve daha fazla hesaplama gücü gerektirir.

**Fonksiyonların Yapısına Göre Yöntem Performansı**

1. **Unimodal Fonksiyonlar:**
   * Her iki yöntem de unimodal fonksiyonlar için etkili çözümler sunar. Ancak quadratic interpolasyon, düşük doğruluk gerektiren veya hızlı sonuç alınması gereken durumlarda daha uygun bir seçenek olabilir.
   * Cubic interpolasyon ise daha hassas sonuçların önemli olduğu durumlarda tercih edilmelidir.
2. **Multimodal Fonksiyonlar:**
   * Çalışma kapsamında her iki yöntemin multimodal fonksiyonlar üzerindeki performansı incelenmemiştir. Ancak, her iki yöntemin de unimodal bölgelerde ayrı ayrı uygulanması gerektiği bilinmektedir. Bu durumda, yöntemin başlangıç noktalarının seçimi ve aralık daraltma stratejileri kritik öneme sahiptir.
3. **Türev Bilgisi Kullanımı:**
   * Quadratic interpolasyon türev bilgisi gerektirmez ve bu nedenle türev hesaplamanın zor ya da mümkün olmadığı durumlarda avantajlıdır.
   * Cubic interpolasyon türev bilgisi ile çalışır ve bu durum, doğruluğu artırırken yöntemin uygulanabilirliğini türev bilgisine erişime bağımlı hale getirir.

**Yakınsama Davranışı**

1. **Hız:**
   * Quadratic interpolasyon, daha az bilgiye ihtiyaç duyar ve hızlı bir yakınsama sağlar. Fonksiyonun ikinci dereceden davranış sergilediği durumlarda oldukça etkili çalışır.
   * Cubic interpolasyon, daha karmaşık bir yapıya sahip olduğundan, genellikle yakınsama süresi quadratic interpolasyona kıyasla daha uzundur.
2. **Hassasiyet:**
   * Cubic interpolasyon, minimum noktanın daha hassas bir şekilde tahmin edilmesini sağlar. Türev bilgisi, bu yöntemin doğruluğunu artıran temel faktördür.
   * Quadratic interpolasyon ise hassasiyet açısından cubic interpolasyona kıyasla daha sınırlıdır, ancak türev bilgisi gerektirmemesi sayesinde birçok pratik durumda uygulanabilir.

**Kod Uygulamaları ve Sonuçların Değerlendirilmesi**

1. **Quadratic İnterpolasyon Kodu:**
   * Kod, quadratic interpolasyon algoritmasını doğru bir şekilde uygulamış ve teorik modelle birebir uyumlu sonuçlar üretmiştir.
   * Test fonksiyonlarında hızlı yakınsama sağlanmış ve minimum nokta doğru bir şekilde tahmin edilmiştir.
   * Basit yapısı sayesinde kod, gerçek zamanlı uygulamalarda veya düşük hesaplama kapasitesine sahip sistemlerde kolayca uygulanabilir.
2. **Cubic İnterpolasyon Kodu:**
   * Kod, cubic interpolasyon algoritmasını başarılı bir şekilde uygulamış ve teorik prosedürlere tam uyum göstermiştir.
   * Minimum nokta tahminleri quadratic interpolasyona göre daha hassas olmuştur. Ancak, türev bilgilerinin doğru bir şekilde sağlanması gerekliliği kodun uygulanabilirliğini sınırlandırmıştır.
   * Daha karmaşık yapısı nedeniyle, kodun çalışma süresi quadratic interpolasyona kıyasla daha uzundur.

**Genel Değerlendirme**

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.