

BANDIRMA ONYEDİ EYLÜL ÜNİVERSİTESİ



SAYISAL ANALİZ

Doç. Dr. Öğretim Üyesi Serhat KILIÇARSLAN

Müller Yöntemi ile Kök Bulma

2211505010 – Uğur ELMA

Mühendislik ve Doğa Bilimleri Fakültesi

Yazılım Mühendisliği

2024-2025, Mayıs

Müller Yöntemi ile Kök Bulma

a. Kod

```
import cmath

# Fonksiyon sınıfını tanımladık
class Fonksiyon:

    # Sınıf oluşturulduğunda boş bir fonksiyon olan kurucu metottur
    def __init__(self):
        self.katsayilar = []
        self.kokler = []

    # Fonksiyonun sonucunu bulan metot oluşturduk
    def F(self, x):
        toplam = 0
        for i in range(len(self.katsayilar)):
            toplam += self.katsayilar[i] * x ** i
        return toplam

    # Fonksiyonun katsayılarını belirledik
    def KatsayilariEkle(self, derece):
        for i in range(derece + 1):
            while True:
                try:
                    katsayi = float(input("x^{i} in katsayisi: ".format(i)))
                    self.katsayilar.append(katsayi)
                    break
                except:
                    print("Sadece sayi giriniz!")

    # Fonksiyonun koklerini belirledik
    def KokleriEkle(self):
        while True:
            try:
                kokSayisi = int(input("Kok sayisini gir: "))
                if kokSayisi > 0:
                    for i in range(kokSayisi):
                        kok = float(input("x^{i}: ".format(i)))
                        self.kokler.append(kok)
                    break
            except:
                print("Gecerli kok sayisi giriniz!")
                print("Sadece sayi giriniz!")

print("Muller Yontemi ile Kok Bulma\nFonksiyon Bilgilerini Gir")

# Fonksiyon sınıfından bir nesne türettik
fonksiyon = Fonksiyon()

# Fonksiyonun katsayılarını kullanıcıdan aldık
derece = int(input("Fonksiyonun derecesini gir: "))
fonksiyon.KatsayilariEkle(derece)

# Kökleri alıyoruz
fonksiyon.KokleriEkle()

def ItereGore(sinir):
    for i in range(sinir):
        h0 = fonksiyon.kokler[1] - fonksiyon.kokler[0]
        h1 = fonksiyon.kokler[2] - fonksiyon.kokler[1]
```

```

        S0 = (fonksiyon.F(fonksiyon.kokler[1]) -
fonksiyon.F(fonksiyon.kokler[0])) / h0
        S1 = (fonksiyon.F(fonksiyon.kokler[2]) -
fonksiyon.F(fonksiyon.kokler[1])) / h1
        a = (S1 - S0) / h0 + h1
        b = h1 * a + S1
        c = fonksiyon.F(fonksiyon.kokler[2])
        x3 = fonksiyon.kokler[2] - (2 * c / (b + (cmath.sqrt(b ** 2 - 4 * a *
c))))
        fonksiyon.kokler.append(x3)
        fonksiyon.kokler.pop(0)

    print("{} iterasyon sonunda kok: {}".format(sinir, x3))

def HatayaGore(sinir):
    adim = 0
    bagilHata = -1
    while bagilHata <= sinir:
        h0 = fonksiyon.kokler[1] - fonksiyon.kokler[0]
        h1 = fonksiyon.kokler[2] - fonksiyon.kokler[1]
        S0 = (fonksiyon.F(fonksiyon.kokler[1]) -
fonksiyon.F(fonksiyon.kokler[0])) / h0
        S1 = (fonksiyon.F(fonksiyon.kokler[2]) -
fonksiyon.F(fonksiyon.kokler[1])) / h1
        a = (S1 - S0) / h0 + h1
        b = h1 * a + S1
        c = fonksiyon.F(fonksiyon.kokler[2])
        x3 = fonksiyon.kokler[2] - (2 * c / (b + (cmath.sqrt(b ** 2 - 4 * a *
c))))
        bagilHata = cmath.fabs((x3 - fonksiyon.kokler[2]) / x3)
        fonksiyon.kokler.append(x3)
        fonksiyon.kokler.pop(0)
        adim += 1

    print("{} adim sonunda\n kok: {}\n bagil hata: {}".format(adim, x3,
bagilHata))

# iterasyona göre mi yoksa bağıl hataya göre mi çözülecek
while True:
    try:
        neyeGore = int(input("1 -> Bagil Hata\n2 -> Iterasyon\nNeye gore
cozulecek: "))
        if neyeGore == 2:
            sinir = int(input("Kac iterasyon: "))
            ItereGore(sinir)
            break
        elif neyeGore == 1:
            sinir = float(input("Bagil hata: "))
            HatayaGore(sinir)
            break
        else:
            print("1 ya da 2 giriniz!")
    except:
        print("Sadece sayi giriniz!")

```

b. Örnek

$$f(x) = x^3 - 13x - 11$$

$$x_0 = 4,5 \quad - \quad x_1 = 5,5 \quad - \quad x_2 = 5$$

c. Çıktı

```
Muller Yontemi ile Kok Bulma
Fonksiyon Bilgilerini Gir
Fonksiyonun derecesini gir: 3
x^0 in katsayisi: -11
x^1 in katsayisi: -13
x^2 in katsayisi: 0
x^3 in katsayisi: 1
Kok sayisini gir: 3
x0: 4.5
x1: 5.5
x2: 5
1 -> Bagil Hata
2 -> Iterasyon
Neye gore cozulecek: 2
Kac iterasyon: 4
4. iterasyon sonunda kok: (3.972155447875374-0.0020276552134006493j)
Press any key to continue . . . |
```