17.05.2025

Yıldız Teknik Üniversitesi
Elektrik-Elektronik Fakültesi
Bilgisayar Mühendisliği Bölümü

BLM1022
Sayısal Analiz
Gr: 1

Prof.Dr.Banu DİRİ
Dönem Projesi

İsim:Muhammed Emir Gül

No:23011051

E-posta: emir.gul@std.yildiz.edu.tr

İçindekiler

İçindekiler

Ön Bilgi

Program, 10 tane belirli işlemi yerine getirebilmek için tasarlanmıştır. Bu işlemler sırasıyla şöyledir:

1. Bisection yöntemi
2. Regula-Falsi yöntemi
3. Newton-Rapshon yöntemi
4. NxN'lik bir matrisin tersi
5. Cholesky (ALU) Yöntemi
6. Sayısal Türev
7. Simpson yöntemi
8. Trapez yöntemi
9. Değişken dönüşümsüz Gregory-Newton enterpolasyonu
10. Fonksiyon sonucu bulma

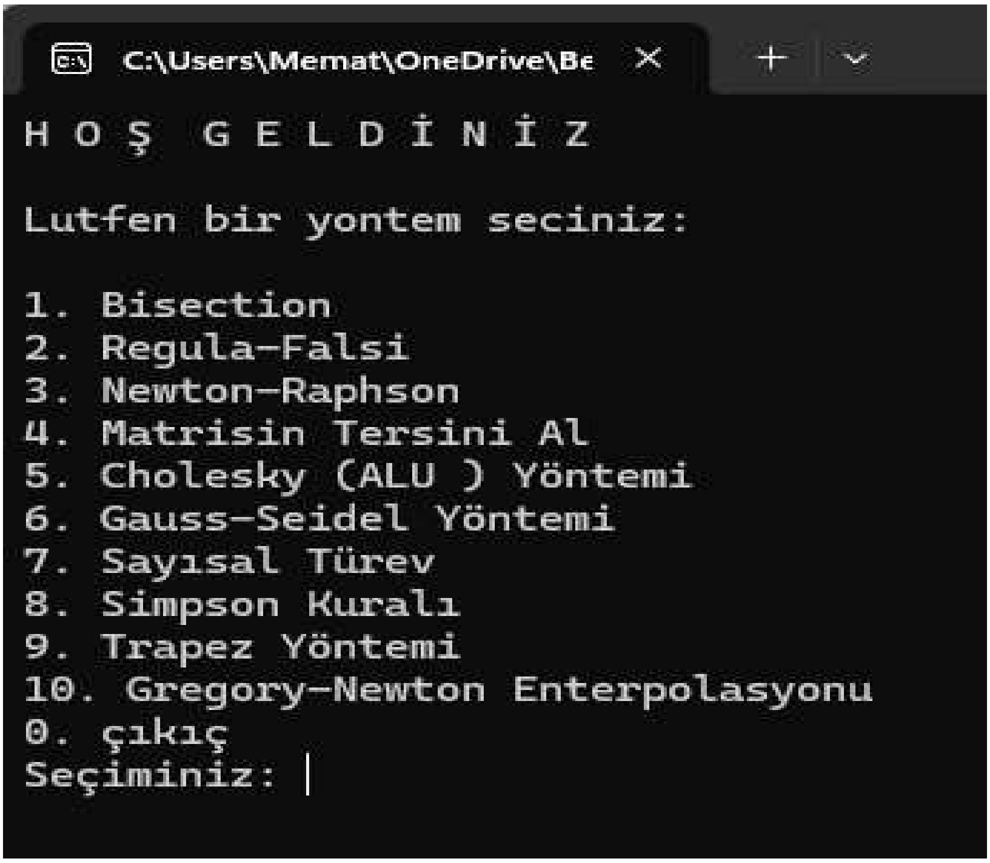
YAPILABİLEN İŞLEMLER

1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	1	1

Ana Menü

Çalıştırılmak istenilen işlem program çalıştırıldıktan sonra numarası girilip gereken parametrelerin verilmesiyle çalışır. Ana menüde '0' girdisi verilene kadar program çalışmaya devam eder.

Ana ekran bu şekildedir .



Desteklenen Fonksiyonlar

Bu çalışmada geliştirilen sayısal analiz programı, kullanıcıdan string olarak alınan matematiksel ifadeleri parçalayıp (tokenize edip), Shunting Yard algoritması ile ters Polonya gösterimine (RPN) dönüştürmekte ve evaluateRPN fonksiyonu yardımıyla verilen nokta üzerindeki sayısal değerini hesaplamaktadır.(Yapılırken İnternetteki kaynaklardan yardım alınmıştır bu parseleme işlemi oldukça zorlamıştır .)

Programda desteklenen temel fonksiyon tipleri ve işlem yapıları aşağıda belirtilmiştir:

1. Polinom Fonksiyonlar

Polinomlar doğal sayılarla tanımlı üs alma (^), çarpma (*), toplama (+) ve çıkarma (-) işlemleri ile ifade edilebilir. Örnek:

- 4*x^3 + 2*x - 7
- x^2 - 5*x + 6

2. Üstel Fonksiyonlar

Program exp(x) ifadesiyle **e tabanlı üstel fonksiyonları** desteklemektedir.

- exp(x) → *e^x*
- 3*exp(x^2) → *3·e^{x²}*

Not: e sabiti doğrudan desteklenmemektedir, ancak exp(1) yazılarak elde edilebilir.

3. Logaritmik Fonksiyonlar

Sistemde yalnızca **10 tabanında logaritma** işlemi desteklenmektedir:

- log(x) → $log_{10}(x)$
- 2*log(x^2+1) gibi bileşik ifadeler mümkündür.

Doğal logaritma ln(x) fonksiyonu ve değişken tabanlı log_base(x) gibi yapılar desteklenmemektedir.

4. Trigonometrik Fonksiyonlar

Aşağıdaki temel trigonometrik fonksiyonlar desteklenmektedir:

- sin(x)
- cos(x)
- tan(x)

Örnek:

- 2*sin(x^2) + 3*cos(x)

Ters trigonometrik fonksiyonlar (arcsin, arccos, arctan, arccot) desteklenmemektedir.

5. Parantezli İfadeler

Öncelik sırasını belirtmek için parantez kullanılabilir:

- sin(x*(x+1)) + log(10/(x+2))

6. Kombinasyonlu Yapılar

Fonksiyonlar, çarpanlarıyla birlikte kullanılabilir:

- 2*x^3 + 4*sin(x^2) + log(x^2+1)

Bu yapı sayesinde hem kök bulma hem integral alma hem de türev hesaplama işlemleri kullanıcıdan alınan ifade doğrultusunda otomatik olarak işlenebilmektedir.

Matris Girişi

Matrisin tersi (4) ve lineer denklem çözümü yöntemleri (5, 6) için ilk istenilen parametre NxN’lik bir kare matris için N değeridir. Bu değer girildikten sonra matrisin elemanları satır satır alınır.

Örnek

$$N = 3, \quad \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

```
MATRİSİN TERSİNİ BULMA
Matrisin boyutunu giriniz: 3
1. satır, 1. sütun elemanını giriniz: 1
1. satır, 2. sütun elemanını giriniz: 2
1. satır, 3. sütun elemanını giriniz: 3
2. satır, 1. sütun elemanını giriniz: 4
2. satır, 2. sütun elemanını giriniz: 5
2. satır, 3. sütun elemanını giriniz: 6
3. satır, 1. sütun elemanını giriniz: 7
3. satır, 2. sütun elemanını giriniz: 8
3. satır, 3. sütun elemanını giriniz: 9
Girdiğiniz Matris:
1,00000 2,00000 3,00000
4,00000 5,00000 6,00000
7,00000 8,00000 9,00000
Determinant: 0,00000000
Girilen matrisin tersi yoktur (determinant sıfır).

Lutfen bir yontem seciniz:

1. Bisection
2. Regula-Falsi
3. Newton-Raphson
4. Matrisin Tersini Al
5. Cholesky (ALU ) Yöntemi
6. Gauss-Seidel Yöntemi
7. Sayısal Türev
8. Simpson Kuralı
9. Trapez Yöntemi
10. Gregory-Newton Enterpolasyonu
```

Bisection Yöntemi

Parametreler

Fonksiyon

bas: Başlangıç değeri

bit: Bitiş değeri

tolerans: Hata miktarı

Durma Koşulu: $\text{fabs(sonucOrta)} >= \text{tolerans} \;\&\&\; \text{iterasyon} < \text{maxIter}$
$\text{Fabs} = \text{mutlağını alma}$

Max iterations: Maksimum iterasyon sayısı

Örnek

Fonksiyon: $x^3 - 7x^2 + 14x - 6$

bas: 0

bit: 1

tolerans: 0.01

Durma Koşulu: : $\text{fabs(sonucOrta)} >= \text{tolerans} \;\&\&\; \text{iterasyon} < \text{maxIter}$ 2^n

```
BISEKSIYON YONTEMI
Bir fonksiyon giriniz (orn: x^6-7x+9, sin(x), log(x) vs.)
x^3 - 7*x^2 + 14*x - 6
Alt siniri girin: 0
Ust siniri girin: 1
Bu aralikta bir kok mevcut.
epsilon giriniz: 0,01

--- 1. Adim ---
Alt Sinir: 0,000000, f(alt): -6,000000
Ust Sinir: 1,000000, f(ust): 2,000000
Orta Nokta: 0,500000, f(orta): -0,625000

--- 2. Adim ---
Alt Sinir: 0,500000, f(alt): -0,625000
Ust Sinir: 1,000000, f(ust): 2,000000
Orta Nokta: 0,750000, f(orta): 0,984375

--- 3. Adim ---
Alt Sinir: 0,500000, f(alt): -0,625000
Ust Sinir: 0,750000, f(ust): 0,984375
Orta Nokta: 0,625000, f(orta): 0,259766

--- 4. Adim ---
Alt Sinir: 0,500000, f(alt): -0,625000
Ust Sinir: 0,625000, f(ust): 0,259766
Orta Nokta: 0,562500, f(orta): -0,161865

--- 5. Adim ---
Alt Sinir: 0,562500, f(alt): -0,161865
Ust Sinir: 0,625000, f(ust): 0,259766
Orta Nokta: 0,593750, f(orta): 0,054047

--- 6. Adim ---
Alt Sinir: 0,562500, f(alt): -0,161865
Ust Sinir: 0,593750, f(ust): 0,054047
Orta Nokta: 0,578125, f(orta): -0,052624

--- 7. Adim ---
Alt Sinir: 0,578125, f(alt): -0,052624
Ust Sinir: 0,593750, f(ust): 0,054047
Orta Nokta: 0,585938, f(orta): 0,001031

 Sonuc: 0,585938 fonksiyonun kokudur.

Cikmak icin Enter'a basin...
```

Max iterations: 100

Regula-Falsi Yöntemi

Parametreler

Fonksiyon

bas: Başlangıç değeri

bit: Bitiş değeri

tolerans: Hata miktarı

Durma Koşulu:
$\mathrm{fabs(fc)} >= \mathrm{tolerans}\ \&\&\ \mathrm{iter} < \mathrm{maxIter}$ ***Max iterations:*** Maksimum iterasyon sayısı

Örnek

Fonksiyon: $x^3 - 2x^2 - 5$

bit: 2

bas: 3

tolerans: 0.01

Durma Koşulu:

$\mathrm{fabs(fc)} >= \mathrm{tolerans}\ \&\&\ \mathrm{iter} < \mathrm{maxIter}$ ***Max iterations:*** Maksimum iterasyon sayısı

Max iterations: 100

```
REGULA-FALSI (Doğrunun Kestiği Yöntem)
Fonksiyonu giriniz (örn: x^3-2x+1, sin(x), log(x) vs.):
x^3-2*x^2-5
Alt sınırı girin: 2
Üst sınırı girin: 3
Aralıkta bir kök var.
Hassasiyet (epsilon) değerini giriniz: 0,01

1. Adım:
Alt Sınır: 2,000000, f(alt): -5,000000
Üst Sınır: 3,000000, f(üst): 4,000000
Yeni Tahmin: 2,555556, f(yeni): -1,371742

2. Adım:
Alt Sınır: 2,555556, f(alt): -1,371742
Üst Sınır: 3,000000, f(üst): 4,000000
Yeni Tahmin: 2,669050, f(yeni): -0,233803

3. Adım:
Alt Sınır: 2,669050, f(alt): -0,233803
Üst Sınır: 3,000000, f(üst): 4,000000
Yeni Tahmin: 2,687326, f(yeni): -0,036321

4. Adım:
Alt Sınır: 2,687326, f(alt): -0,036321
Üst Sınır: 3,000000, f(üst): 4,000000
Yeni Tahmin: 2,690140, f(yeni): -0,005561

Sonuç: 2,690140 fonksiyonun köküdür.
```

Newton-Raphson Yöntemi

Parametreler

Fonksiyon

Başlangıç tahmin değeri***:*** x'in başlangıç değeri

Hata Oranı ***:*** Hata miktarı

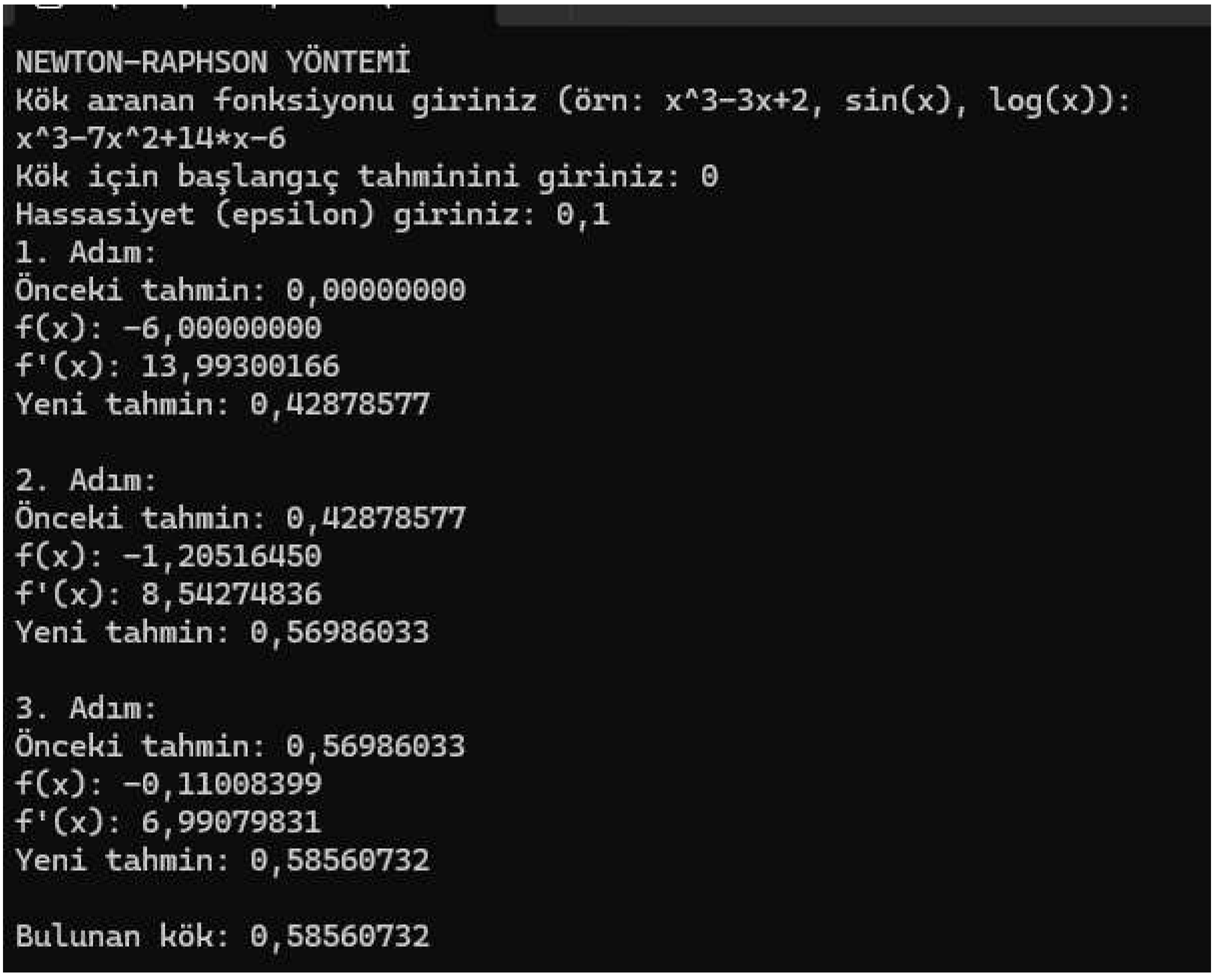
Max iterations: Maksimum iterasyon sayısı

Örnek

Fonksiyon: $x^3 - 7x^2 + 14x - 6$

Başlangıç tahmin değeri***:*** 0

Hata Oranı: 0.1



```
NEWTON-RAPHSON YÖNTEMİ
Kök aranan fonksiyonu giriniz (örn: x^3-3x+2, sin(x), log(x)):
x^3-7x^2+14*x-6
Kök için başlangıç tahminini giriniz: 0
Hassasiyet (epsilon) giriniz: 0,1
1. Adım:
Önceki tahmin: 0,00000000
f(x): -6,00000000
f'(x): 13,99300166
Yeni tahmin: 0,42878577

2. Adım:
Önceki tahmin: 0,42878577
f(x): -1,20516450
f'(x): 8,54274836
Yeni tahmin: 0,56986033

3. Adım:
Önceki tahmin: 0,56986033
f(x): -0,11008399
f'(x): 6,99079831
Yeni tahmin: 0,58560732

Bulunan kök: 0,58560732
```

Matrisin Tersini Alma

Parametreler

Fonksiyon

N : Kare Matrisin Boyutu

Örnek

```
MATRİSİN TERSİNİ BULMA
Matrisin boyutunu giriniz: 3
1. satır, 1. sütun elemanını giriniz: 1
1. satır, 2. sütun elemanını giriniz: 2
1. satır, 3. sütun elemanını giriniz: 5
2. satır, 1. sütun elemanını giriniz: 6
2. satır, 2. sütun elemanını giriniz: 7
2. satır, 3. sütun elemanını giriniz: 3
3. satır, 1. sütun elemanını giriniz: 0
3. satır, 2. sütun elemanını giriniz: 1
3. satır, 3. sütun elemanını giriniz: 2
Girdiğiniz Matris:
1,00000 2,00000 5,00000
6,00000 7,00000 3,00000
0,00000 1,00000 2,00000
Determinant: 17,00000000
Matrisin Tersi:
0,64705882 0,05882353 -1,70588235
-0,70588235 0,11764706 1,58823529
0,35294118 -0,05882353 -0,29411765
```

Cholesky (ALU) Yöntemi

Parametreler

Denklem sisteminin boyutu:
Sağ taraf vektörü:

```
CHOLESKY (ALU) YÖNTEMİ
Denklem sisteminin boyutunu giriniz (n x n): 3
Matris A'nın elemanlarını satır satır giriniz:
A[1][1]: 25
A[1][2]: 15
A[1][3]: -5
A[2][1]: 15
A[2][2]: 18
A[2][3]: 0
A[3][1]: -5
A[3][2]: 0
A[3][3]: 11
Sağ taraf vektörü b'nin elemanlarını giriniz:
b[1]: 350
b[2]: 400
b[3]: 200

Çözüm vektörü x:
x1 = 10,51851852
x2 = 13,45679012
x3 = 22,96296296
```

Gauss-Seidel Yöntemi

Parametreler

N***:*** Çözülecek Matrisin Boyutu

N[][]: Matris değerleri girme

Sonuç vektörü[N]: Sonuç vektörü girme

Tolerans : hata

Örnek

```
C:\Users\Memat\OneDrive\Be
GAUSS-SEIDEL YÖNTEMİ
Kaç bilinmeyenli denklem sistemi çözeceksiniz? 3
1. denklemin katsayılarını ve sonucu giriniz:
x1 katsayısı: 10
x2 katsayısı: 2
x3 katsayısı: 1
Sonuç: 7
2. denklemin katsayılarını ve sonucu giriniz:
x1 katsayısı: 1
x2 katsayısı: 5
x3 katsayısı: 1
Sonuç: -8
3. denklemin katsayılarını ve sonucu giriniz:
x1 katsayısı: 2
x2 katsayısı: 3
x3 katsayısı: 10
Sonuç: 6
Hata (epsilon) değerini giriniz: 0,01
Başlangıç değerlerini elle girmek için 1 yazın, otomatik sıfırdan başlamak için başka bir değer gi
---- 1. iterasyon sonuçları ----
x1 = 0,70000000
x2 = -1,74000000
x3 = 0,98200000
---- 2. iterasyon sonuçları ----
x1 = 0,94980000
x2 = -1,98636000
x3 = 1,00594800
---- 3. iterasyon sonuçları ----
x1 = 0,99667720
x2 = -2,00052504
x3 = 1,00082207
---- 4. iterasyon sonuçları ----
x1 = 1,00002280
x2 = -2,00016897
x3 = 1,00004613

Son çözüm:
x1 = 1,00002280
x2 = -2,00016897
x3 = 1,00004613
```

Sayısal Türev Yöntemi

Parametreler

Fonksiyon

X : Türev Noktası

H : Adım miktarı

Tolerans : hata miktarı

Secim : İleri-geri-merkezi seçme

Örnek

```
SAYISAL TÜREV HESAPLAMA
Fonksiyonu giriniz (örn: 2*x+5, x^3-2*x^2-3*x-15, sin(x), log(x), vs.):
x^2+3*x+2
Türev alınacak x değerini giriniz: 1
Epsilon (adım) değeri giriniz: 0,1

Epsilon (adım) = 0,10000000
Merkezi Fark (Ortalama Türev): 5,0000012
Geri Fark  (Soldan Yaklaşım) : 4,9000011
İleri Fark (Sağdan Yaklaşım) : 5,1000012
```

Simpson 1/3-3/8 Yöntemi

Parametreler

Fonksiyon

bas: integralin başlangıç değeri

bit: integralin bitiş değeri

n = bolum sayisi

h = (bas-bit)/bolum

secim = 1/3 ya da 3/8

Örnek

Trapez Yöntemi

Parametreler

Fonksiyon

bas: integralin başlangıç değeri

bitir : integralin bitiş değeri

bölüm sayısı: aralık sayısı

h : (bitir-basla)/bölüm sayısı

Örnek

```
SIMPSON YONTEMI (Yaklaşık Integral)
Hangi yöntemi kullanmak istiyorsunuz?
1. Simpson 1/3 Kuralı
2. Simpson 3/8 Kuralı
Seçiminiz (1 veya 2): 1
Fonksiyon giriniz (örn: x^2+5, sin(x), log(x)):
x^4-2x+1
Alt sınırı giriniz: 0
Üst sınırı giriniz: 2
n değerini giriniz (çift olmalı!): 6
Simpson 1/3 ile yaklaşık integral sonucu: 4,40329194

Lutfen bir yontem seciniz:
```

Enterpolasyon Yöntemi(İleri Fark)

Parametreler

Kaç tane x,y çifti gireceksiniz
Sırayla x,y değerlerini gir
Bulmak istediğin x değerini gir

Örnek

1. x,y= 1,0
2. X,y=0,1
3. X,y=2,1

4. X,y=3,10

```
GREGORY-NEWTON ENTERPOLASYON
Kaç tane (x, y) çifti gireceksiniz?
4
Sırayla x ve y değerlerini girin:
1. (x, y): 1
0
2. (x, y): 0
1
3. (x, y): 2
1
4. (x, y): 3
10
f(x) değerini bulmak istediğiniz x değerini girin: 2,5

Fark Tablosu:
0,000000        1,000000        -1,000000       10,000000
1,000000        0,000000        9,000000
1,000000        9,000000
10,000000

h = -1,000000
Sonuç: f(2,500) = -6,50000000
```

Proje Yaparken Karşılaşılan Sorunlar

Proje yapılırken karşılaşılan en büyük sorun string olarak alınan veriyi integer haline çevirme ve işlem önceliklerini ona göre yapma . Sorun çözümü için internetten araştırma yapıldı ve Tokenize ve Stack işlemleri ile sorun çözüldü . Ondan sonra tek tek adımlar kodlandı . İçerisine setlocale(LC_ALL, "Turkish") kodu eklendiği için epsilon yazarken 0.1 yazınca kabul olmuyor 0,1 yazmamız gerekiyor.

TEŞEKKÜRLER