

21.05.2023



Yıldız Teknik Üniversitesi
Elektrik-Elektronik Fakültesi
Bilgisayar Mühendisliği Bölümü

BLM1022

Sayısal Analiz

Gr: 2

Prof. Dr. Banu Diri

Dönem Projesi

İsim: Burak Atalay

No: 22011641

E-posta: burak.atalay2@std.yildiz.edu.tr

21.05.2023

İçindekiler

Ön Bilgi	3
Ana Menü	4
Polinom	5
Bisection Yöntemi	8
Regula-Falsi Yöntemi	9
Newton-Raphson Yöntemi	10
$N \times N$ 'lik Bir Matrisin Tersi	12
Gauss Eliminasyon Yöntemi	14
Gauss-Seidel Yöntemi	16
Sayısal Türev	19
Simpson Yöntemi	20
Trapez Yöntemi	21
Değişken Dönüşümsüz Gregory-Newton Enterpolasyonu	22

21.05.2023

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. $N \times N$ 'lik bir matrisin tersi
5. Gauss eliminasyon yöntemi
6. Gauss-Seidel yöntemi
7. Sayısal Türev
8. Simpson yöntemi
9. Trapez yöntemi
10. Değişken dönüşümsüz Gregory-Newton enterpolasyon

YÖNTEMLERİN YAPILIP YAPILMADIĞINI AŞAĞIDAKİ TABLODA GÖSTERİLDİĞİ GİBİ 1/0 OLARAK GÖSTERİNİZ									
1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	1	1

21.05.2023

Ana Menü

Çalıştırılmak istenilen işlem program çalıştırıldıktan sonra numarası girilip gereken parametrelerin verilmesiyle çalışır. İstenilen program çalıştırıldıktan ve bittikten sonra ikinci fotoğraftaki soru sorulur. 1'e basılırsa ana menüye dönülür, 0'a basılırsa program sonlanır. Ayrıca 1, 2, 3, 7, 8, 9, 10. yöntemler yalnızca Polinom fonksiyonlar için desteklenir.

```
Bisection Method: 1
Regula-Falsi Method: 2
Newton-Raphson Method: 3
Inverse of a matrix: 4
Gauss elimination: 5
Gauss seidel Method: 6
Numerical Differentiation: 7
Simpson Method: 8
Trapezoidal Method: 9
Gregory Newton Interpolation: 10
Method of choice:
```

```
enter 1 if you want to return to the menu otherwise enter 0:
```

21.05.2023

Polinom

Polinom fonksiyonlarla işlem yapmak için 4 adet fonksiyon oluşturdum.

Polinomu kullanıcıdan almak için: `getPolynomial`

Parametreler:

`coefficient`: katsayıların tutulduğu dizi

`degree`: derecelerin tutulduğu dizi

`n`: polinomun terim sayısı

Kullanıcıya fonksiyonda kaç terim olduğunu sorup terimli karışık olarak girmesi beklenir. Önce katsayısı, sonra derecesi. Bunlar parametre olarak alınan dizilere kaydedilir.

Polinomu yazdırmak için: `sortAndPrintPolynomial`

Parametreler:

`coefficient`: katsayıların tutulduğu dizi

`degree`: derecelerin tutulduğu dizi

`n`: polinomun terim sayısı

Kullanıcıdan alınan `n` terimli fonksiyonu insertion sort kullanarak derecelerine göre sıralar ve ekrana yazdırır.

21.05.2023

Örnek

Fonksiyon: $x^3 + 2x - 3$, Terim Sayısı: 3

```
how many elements are there in the polynomial: 3
the coefficient of 1.th value: -3
the degree of 1.th value: 0
the coefficient of 2.th value: 5
the degree of 2.th value: 3
the coefficient of 3.th value: 2
the degree of 3.th value: 1
Function...
(5.000)x^3 + (2.000)x^1 + (-3.000)x^0
Beginning point of the interval:
```

Polinomda değer hesaplamak için: `calculatePolynomial`

Parametreler:

x: Polinomdaki değeri hesaplanacak sayı

coefficient: katsayıların tutulduğu dizi

degree: derecelerin tutulduğu dizi

n: polinomun terim sayısı

Bu fonksiyon polinomda istenen bir x değerini hesaplamak için hiçbir hazır fonksiyon kullanmaz.

21.05.2023

Kod:

```
double calculatePolynomial(double x, double coefficient[max], int degree[max], int n){
    int i, j;
    double sum = 0.0, temp = 1.0;
    for(i=0; i<n; i++){
        temp = 1.0;
        for(j=0; j<degree[i]; j++){
            temp = temp * x;
        }
        sum += coefficient[i] * temp;
    }
    return sum;
}
```

Polinomun birinci türevinde değer hesaplamak için: calculateDerivative

Parametreler:

x: Polinomdaki değeri hesaplanacak sayı

coefficient: katsayıların tutulduğu dizi

degree: derecelerin tutulduğu dizi

n: polinomun terim sayısı

Bu fonksiyon polinomun birinci türevinde istenen bir x değerini hesaplamak için hiçbir hazır fonksiyon kullanmaz.

Kod:

```
double calculateDerivative(double x, double coefficient[max], int degree[max], int n){
    int i, j;
    double sum = 0.0, temp = 1.0;
    for(i=0; i<n; i++){
        temp = 1.0;
        for(j=0; j<degree[i]-1; j++){
            temp = temp * x;
        }
        sum += coefficient[i] * degree[i] * temp;
    }
    return sum;
}
```

21.05.2023

Bisection yöntemi

Parametreler:

Fonksiyon

Beginning point of the interval: Başlangıç değeri

Ending point of the interval: Bitiş Değeri

expected error: Beklenen Hata

Örnek

Fonksiyon: $x^3 - 7x^2 + 14x^1 - 6x^0$

Beginning point of the interval: 0

Ending point of the interval: 1

expected error: 0.01

```
Function...
(1.000)x^3 + (-7.000)x^2 + (14.000)x^1 + (-6.000)x^0
Beginning point of the interval: 0
Ending point of the interval: 1
expected error: 0.01
root: 0.625000
```


21.05.2023

Regula-Falsi yöntemi

Parametreler:

Fonksiyon

Beginning point of the interval: Başlangıç Değeri

Ending point of the interval: Bitiş Değeri

expected error: Beklenen Hata

Örnek

Fonksiyon: $x^2 - 3x^1 + 2x^0$

Beginning point of the interval: 1

Ending point of the interval: 3

expected error: 0.01

```
Function...
```

```
(1.000)x^2 + (-3.000)x^1 + (2.000)x^0
```

```
Beginning point of the interval: 1
```

```
Ending point of the interval: 3
```

```
expected error: 0.001
```

```
root: 1.000000
```

```
enter 1 if you want to return to the menu otherwise enter 0: ☐
```

21.05.2023

Newton-Raphson Yöntemi

Parametreler:

Fonksiyon

left: Kök Aranacak Aralığın Başlangıç Değeri

right: Kök Aranacak Aralığın Bitiş Değeri

x0: başlangıç değeri

expected error: Beklenen Hata

Not: left ve right değerlerini alma sebebim aralıkta kök var mı tespit edip ıraksamayı engellemek.

Örnek:

Fonksiyon: $x^3 - 7x^2 + 14x^1 - 6x^0$

left: 0

right: 1

x0: 0

expected error: 0.000001

21.05.2023

```
Function...
(1.000)x^3 + (-7.000)x^2 + (14.000)x^1 + (-6.000)x^0
Beginning point of the interval: 0
Ending point of the interval: 1
if you want to give initial value press 1 otherwise press 0 (Then initial value will be the beginning point of the interval...): 0
expected error: 0.000001
root: 0.585786
```

21.05.2023

NxN'lik Bir Matrisin Tersi

Parametreler:

İnt n: matrisin boyutu

Double matrix: Tersi Alınacak Matris

Double inverse_matrix: Birim matris

Not: Birim matris yardımıyla ters alma işlemi yapılıyor. matrix birim matris haline getirilirken, ilk başta birim matris halinde bulunan inverse_matrix girilen matrisin tersi haline geliyor.

Örnek

n: 4

matrix: 1.2000 -3.0000 2.1000 5.0000

9.0000 2.5000 -4.0000 1.0000

11.0000 6.0000 -3.1000 7.0000

9.1000 -2.0000 0.0000 6.5000

İnverse_matrix: -0.5555 -0.2925 0.0012 0.4711

-0.4358 -0.3592 0.1682 0.2094

-1.3614 -1.0579 0.1203 1.0804

0.6436 0.2990 0.0501 -0.4412

```
n: 4
1. row and 1. column: 1.2
1. row and 2. column: -3
1. row and 3. column: 2.1
1. row and 4. column: 5

2. row and 1. column: 9
2. row and 2. column: 2.5
2. row and 3. column: -4
2. row and 4. column: 1

3. row and 1. column: 11
3. row and 2. column: 6
3. row and 3. column: -3.1
3. row and 4. column: 7

4. row and 1. column: 9.1
4. row and 2. column: -2
4. row and 3. column: 0
4. row and 4. column: 6.5

Matrix...

1.2000 -3.0000 2.1000 5.0000
9.0000 2.5000 -4.0000 1.0000
11.0000 6.0000 -3.1000 7.0000
9.1000 -2.0000 0.0000 6.5000

Inverse Matrix...

-0.5555 -0.2925 0.0012 0.4711
-0.4358 -0.3592 0.1682 0.2094
-1.3614 -1.0579 0.1203 1.0804
0.6436 0.2990 0.0501 -0.4412
```

21.05.2023

Gauss Eliminasyon Yöntemi

Parametreler:

İnt n: matrisin boyutu

Double matrix: Genişletilmiş Matris (Augmented Matrix)

Örnek

n: 4

matrix: 5.1000 -3.0000 1.3000 8.0000 1.0000
6.0000 -1.0000 0.0000 2.6000 5.3000
4.0000 -2.0000 5.0000 6.4000 -3.5000
7.8000 -7.0000 1.5000 -2.0000 7.0000

Not: Genişletilmiş matrise önce katsayıları sonra sonuçları alıp kullanıcıya gösteriyoruz.

21.05.2023

```
-----  
4.th root: -0.3533  
3.th root: -1.0681  
2.th root:  0.0330  
1.th root:  1.0419
```

```
n: 4  
coefficients of 1.th equation...  
coefficient of 1.th unknown: 5.1  
coefficient of 2.th unknown: -3  
coefficient of 3.th unknown: 1.3  
coefficient of 4.th unknown: 8  
  
coefficients of 2.th equation...  
coefficient of 1.th unknown: 6  
coefficient of 2.th unknown: -1  
coefficient of 3.th unknown: 0  
coefficient of 4.th unknown: 2.6  
  
coefficients of 3.th equation...  
coefficient of 1.th unknown: 4  
coefficient of 2.th unknown: -2  
coefficient of 3.th unknown: 5  
coefficient of 4.th unknown: 6.4  
  
coefficients of 4.th equation...  
coefficient of 1.th unknown: 7.8  
coefficient of 2.th unknown: -7  
coefficient of 3.th unknown: 1.5  
coefficient of 4.th unknown: -2  
  
Constant matrix...  
results of 1.th equation: 1  
results of 2.th equation: 5.3  
results of 3.th equation: -3.5  
results of 4.th equation: 7  
  
The augmented matrix...  
5.1000 -3.0000  1.3000  8.0000  1.0000  
6.0000 -1.0000  0.0000  2.6000  5.3000  
4.0000 -2.0000  5.0000  6.4000 -3.5000  
7.8000 -7.0000  1.5000 -2.0000  7.0000  
  
The augmented gauss-elimination matrix...  
1.0000 -0.5882  0.2549  1.5686  0.1961  
0.0000  1.0000 -0.6047 -2.6930  1.6302  
0.0000  0.0000  1.0000  0.2566 -1.1588  
0.0000  0.0000  0.0000  1.0000 -0.3533  
-----  
4.th root: -0.3533  
3.th root: -1.0681
```

21.05.2023

Gauss-Seidal Yöntemi

Parametreler:

İnt n: matrisin boyutu

Double matrix: Genişletilmiş Matris (Augmented Matrix)

expected_error: beklenen hata

max_iteration: maksimum iterasyon sayısı

initial values: her bir bilinmeyen için kullanıcıdan alınan başlangıç değerleri

Not: Gauss-Seidal yöntemi için verilen matrisin diagonally dominant halde olması gerekiyor. Bu her sütündaki mutlak değerce en büyük sayı köşegene gelecek şekilde ayarlanınca sağlanıyor. Ancak eğer iki sütundaki en büyük değerler aynı satırda ise bahsettiğimiz ayarlamayı yapsak bile sonuç büyük ihtimalle ıraksıyor. Bu sebeple ben girilen matrisi diagonally dominant mı diye kontrol ediyorum. Bunu da araştırarak öğrendiğim kadarıyla bir satırdaki köşegen dışındaki elemanların mutlak değerce toplamı köşegenden mutlak değerce küçük ve eşit ise kuralıyla yapıyorum.

Örnek

n: 3

matrix: 3.1000 1.5000 -0.8000 1.5000

5.0000 8.0000 -2.0000 -3.7000

4.5000 -3.0000 9.1000 -4.0000

expected_error: 0.001

max_iteration: 45

initial values: x1=1, x2=2, x3=3

n: 3

please provide a diagonally dominant matrix...

coefficients of 1.th equation...

coefficient of 1.th unknown: 3.1

coefficient of 2.th unknown: 1.5

coefficient of 3.th unknown: -0.8

coefficients of 2.th equation...

coefficient of 1.th unknown: 5

coefficient of 2.th unknown: 8

coefficient of 3.th unknown: -2

coefficients of 3.th equation...

coefficient of 1.th unknown: 4.5

coefficient of 2.th unknown: -3

coefficient of 3.th unknown: 9.1

constant matrix...

results of 1.th equation: 1.5

results of 2.th equation: -3.7

results of 3.th equation: -4

expected error: 0.001

maximum iteration: 45

Initial values for unknowns...

Initial value for 1. unknown: 1

Initial value for 2. unknown: 2

Initial value for 3. unknown: 3

1. iteration...

1. unknown: 0.290323

2. unknown: 0.106048

3. unknown: -0.548166

2. iteration...

1. unknown: 0.291095

2. unknown: -0.781476

3. unknown: -0.841138

```
3. iteration...
1. unknown: 0.644937
2. unknown: -1.075870
3. unknown: -1.113167

4. iteration...
1. unknown: 0.717184
2. unknown: -1.189032
3. unknown: -1.186200

5. iteration...
1. unknown: 0.753093
2. unknown: -1.229733
3. unknown: -1.217375

6. iteration...
1. unknown: 0.764742
2. unknown: -1.244807
3. unknown: -1.228105

7. iteration...
1. unknown: 0.769267
2. unknown: -1.250318
3. unknown: -1.232160

8. iteration...
1. unknown: 0.770887
2. unknown: -1.252344
3. unknown: -1.233629

9. iteration...
1. unknown: 0.771488
2. unknown: -1.253087
3. unknown: -1.234171

The augmented matrix..

3.1000  1.5000 -0.8000  1.5000
5.0000  8.0000 -2.0000 -3.7000
4.5000 -3.0000  9.1000 -4.0000

Results...

1th unknown:0.771488
2th unknown:-1.253087
3th unknown:-1.234171
```

21.05.2023

Sayısal Türev

Parametreler:

Fonksiyon:

x: fonksiyonda türevi alınacak değer

h: formüldeki h

choice: 3 türev şeklinden hangisinin yapılmak istendiği

Örnek

Fonksiyon: $x^2 - 2x^1 + x^0$

x: 3

h: 0.00001

choice: 1 (geri fark ile türev)

```
Function...
(1.000)x^2 + (-2.000)x^1 + (1.000)x^0
enter the value of which you want to calculate the deriative: 3
h: 0.00001
Backward differentiation: 1
Central differentiation: 2
Forward Differentiation: 3
1
result: 3.999990
```

21.05.2023

Simpson Yöntemi

Parametreler:

Fonksiyon

left: integralin başlangıç değeri

right: integralin bitiş değeri

m: bölünecek aralık sayısı

Örnek

Fonksiyon: $(1.000)x^3 + (-3.000)x^2 + (6.000)x^0$

left: 0

right: 2

m: 4

```
Function...
(1.000)x^3 + (-3.000)x^2 + (6.000)x^0
Beginning point of the interval: 0
Ending point of the interval: 2
Enter the amount of subdivisions(m):4
Result for Simpson 1/3 rule: 8.000000
Result for Simpson 3/8 rule: 7.757813
```

21.05.2023

Trapez Yöntemi

Parametreler:

Fonksiyon

left: integralin başlangıç değeri

right: integralin bitiş değeri

m: bölünecek aralık sayısı

Örnek

Fonksiyon: $x^3 - 6x^1 + 9x^0$

left: 0

right: 2

m: 5

```
Function...  
(1.000)x^3 + (-6.000)x^1 + (9.000)x^0  
Beginning point of the interval: 0  
  
Ending point of the interval: 2  
  
Enter the amount of subdivisions(m):5  
  
result: 10.160000
```

21.05.2023

Değişken Dönüşümsüz Gregory-Newton Enterpolasyonu

Parametreler

n: tablodaki nokta sayısı

x_0 : ilk nokta

h: her nokta arasındaki uzaklık

Her bir x 'in fonksiyon değerleri:

x: fonksiyondaki değeri hesaplanacak x değeri

Örnek

n: 6

x_0 : 1

h: 2

Her bir x 'in fonksiyondaki değerleri: $f(1)=4$, $f(3)=-6$, $f(5)=8$, $f(7)=7$, $f(9)=2$, $f(11)=3$

x: 4

21.05.2023

```
How many points will you give: 6
First point (x0): 1
The difference between each point (h): 2
Value of the function for 1.00: 4
Value of the function for 3.00: -6
Value of the function for 5.00: 8
Value of the function for 7.00: 7
Value of the function for 9.00: 2
Value of the function for 11.00: 3
-----
The Difference Table...

1.00  4.00  0.00  0.00  0.00  0.00  0.00
3.00 -6.00 -10.00  0.00  0.00  0.00  0.00
5.00  8.00 14.00 24.00  0.00  0.00  0.00
7.00  7.00 -1.00 -15.00 -39.00  0.00  0.00
9.00  2.00 -5.00 -4.00 11.00 50.00  0.00
11.00  3.00  1.00  6.00 10.00 -1.00  0.00
```

```
Which point you want to calculate: 4
-----
```

```
RESULT: 1.609375
```