23.05.2023

Yıldız Teknik Üniversitesi

Elektrik-Elektronik Fakültesi

Bilgisayar Mühendisliği Bölümü

BLM1022

Sayısal Analiz

Gr: 2

Öğr. Gör. Dr. Ahmet ELBİR

Dönem Projesi

İsim: MEHMET ALİ DURAN

No: 21011090

E-posta: ali.duran@std.yildiz.edu.tr

İçindekiler

ÖN BİLGİ

Bu program sayısal analiz dersi kapsamında proje ödevi olarak 10 adet sayısal yöntemin c dilinde kodlanmasıyla oluşturulmuştur. Program çalıştığında ilk olarak ana menü ile karşılaşıyoruz. Bu menüde yapmak istediğimiz işlemi seçip yapabiliriz, programdan çıkılmadığı sürece her yöntem bitiminde yeni bir yöntemin yapılması istenilip istenilmediği sorulur. '0' tuşu ile de programdan çıkılır. Bütün yöntemler modüler olarak kodlanmıştır. Ana menü de ayrı bir fonksiyon olarak kodlanmış olup main fonksiyonunun içinden çağrılır. Programda ayrıca diğer yöntemlerin çalışması için gerekli olan yardımcı işler ile ilgili de yardımcı programlar fonksiyon olarak kodlanmıştır. Stdio.h kütüphanesi haricinde başka kütüphane kullanılmamıştır. Func tipi isminde polinom fonksiyonlar ile işlem yapmayı kolaylaştırmak için bir struct yapısı tutulmuştur. Makrolar ile MAX ve mMAX sabit değerleri sırasıyla dizilerin ve matrislerin boyutlandırılması için tespit edilmiştir. LAPTÜ kısaltmalı fonksiyon türlerinden sadece polinom fonksiyonlar için çalışmaktadır.

10 ADET YÖNTEMDEN YAPILIP YAPILMAYANLARIN 1/0 OLARAK İŞARETLENMESİ									
1	2	3	4	5	6	7	8	9	10
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

ANA MENÜ

Menüde yapılabilecek yöntemler 1-10 arası numaralandırılmıştır. Çıkış ise ‘0’ tuşu ile yapılmaktadır. İstenilen yöntem numarası girilerek hesaplamalar yapılabilir.

```
************Menu*************

0-Quit
1-Bisection Method
2-Regula-Falsi Method
3-Newton-Rapshon Method
4-Inverse Matrix
5-Gauss Elimination
6-Gauss-Seidal
7-Numerical Differentiation
8-Simpson Method
9-Trapez Method
10-Gregory Newton Interpolation

press 0 to exit or operation number to select operation:
```

Bisection Yöntemi

Bu yöntem seçildiğinde ilk olarak kullanıcıdan polinom fonksiyonun en büyük derecesini daha sonra katsayıları sıra ile alarak fonksiyonun tamamını alıyor. Sonrasında kökün arandığı aralığın alt ve üst sınırları olan a ve b değerlerini sırası ile alıyor ve en son olarak durma koşulu olan hata hassasiyetini alıyor. Bu girdiler doğrultusunda girilen a ve b değerleri için sonuçların çarpımı sıfır ise köklerden biri ya da her ikisi sıfır demektir. Buna göre bir çıktı veriyor. Eğer sonuç pozitif ise arada kök yoktur ya da vardır bunu bilemeyiz. Son olarak negatif ise en az bir kök olduğu kesindir. Bu doğrultuda her iterasyonda a ve b değerleri uygun şekilde yarılama yapılarak köke yaklaşılmaya çalışılıyor.

Örnek girdi ve çıktıları:

X^3 – 7X^2 +14X -6 a:0 ,b:1 Hata toleransı: 0.01

```
0-Quit
1-Bisection Method
2-Regula-Falsi Method
3-Newton-Rapshon Method
4-Inverse Matrix
5-Gauss Elimination
6-Gauss-Seidal
7-Numerical Differentiation
8-Simpson Method
9-Trapez Method
10-Gregory Newton Interpolation

press 0 to exit or operation number to select operation: 1
what is the exponent of the function: 3
enter the coefficients of function from lowest exponent to highest exponent
aX^0, a= :-6
bX^1, b= :14
cX^2, c= :-7
dX^3, d= :1
please enter the root interval numbers a and b
a: 0
b: 1
please enter the error sensitivity which stop condition: 0.01
b: Your function is :

-6.000000X^0 + 14.000000X^1 + -7.000000X^2 + 1.000000X^3

there is at least one root between a and b
about root is 0.562500
```

Burada da aralık değerlerinden birini kök olduğu örnek verilmiştir:

X^2 -3X +2 a:1 ,b:1.5 Hata toleransı:0.01

```
0-Quit
1-Bisection Method
2-Regula-Falsi Method
3-Newton-Rapshon Method
4-Inverse Matrix
5-Gauss Elimination
6-Gauss-Seidal
7-Numerical Differentiation
8-Simpson Method
9-Trapez Method
10-Gregory Newton Interpolation

press 0 to exit or operation number to select operation: 1
what is the exponent of the function: 2
enter the coefficients of function from lowest exponent to highest exponent
aX^0, a= :2
bX^1, b= :-3
cX^2, c= :1
please enter the root interval numbers a and b
a: 1
b: 1.5
please enter the error sensitivity which stop condition: 0.01
b: Your function is :

2.000000X^0 + -3.000000X^1 + 1.000000X^2

1.000000 is one of roots of function
```

Son olarak da pozitif değer üreten bir aralık verelim.

X^3 +1 a:1 ,b:3 Hata Toleransı: 0.01

```
************Menu************

0-Quit
1-Bisection Method
2-Regula-Falsi Method
3-Newton-Rapshon Method
4-Inverse Matrix
5-Gauss Elimination
6-Gauss-Seidal
7-Numerical Differentiation
8-Simpson Method
9-Trapez Method
10-Gregory Newton Interpolation

press 0 to exit or operation number to select operation: 1
what is the exponent of the function: 3
enter the coefficients of function from lowest exponent to highest exponent
aX^0, a= :1
bX^1, b= :0
cX^2, c= :0
dX^3, d= :1
please enter the root interval numbers a and b
a: 1
b: 3
please enter the error sensitivity which stop condition: 0.01
b: Your function is :

1.000000X^0 + 0.000000X^1 + 0.000000X^2 + 1.000000X^3

no root between these two values or too many roots
```

Regula-Falsi Yöntemi

Bu yöntem bisection metodu ile aynı mantıkta çalışır. Fakat bu yöntemde yakınsama işlemi yarıya bölme ile değil a ve b aralığında oluşan üçgenlerin birbirine olan benzerliğinden kaynaklanır. Girdi parametreleri aynı olup önce fonksiyon alınır daha sonra aralık ve hata hassasiyeti alınır.

Örnek girdi ve çıktılar:

X^3 -2X^2 -5 a:2 ,b:3 Hata hassasiyeti: 0.01

```
0-Quit
1-Bisection Method
2-Regula-Falsi Method
3-Newton-Rapshon Method
4-Inverse Matrix
5-Gauss Elimination
6-Gauss-Seidal
7-Numerical Differentiation
8-Simpson Method
9-Trapez Method
10-Gregory Newton Interpolation

press 0 to exit or operation number to select operation: 2
what is the exponent of the function: 3
enter the coefficients of function from lowest exponent to highest exponent
aX^0, a= :-5
bX^1, b= :0
cX^2, c= :-2
dX^3, d= :1
please enter the root interval numbers a and b
a: 2
b: 3
please enter the error sensitivity which stop condition: 0.01
b: Your function is :

-5.000000X^0 + 0.000000X^1 + -2.000000X^2 + 1.000000X^3

there is at least one root between a and b
about root is 2.692730
```

Newton-Rapshon Yöntemi

Bu yöntemde de diğerlerinde olduğu gibi öncelikle fonksiyonun derecesi ve katsayıları alınır. Bu yöntemde bir ilk başlangıç değeri verilebileceği gibi aralık da verilebilir. Ben aralık olarak alıp küçük değeri başlangıç değeri olarak ayarladım. Bu yöntemde her iterasyonda fonksiyon ve türevi işleme sokularak bir sonraki değer bulunur ve böylece köke yaklaşılır. Iraksama ihtimali de vardır.

Örnek girdi ve çıktılar:

X^3 -7X^2 +14X -6 a:2 ,b:3 Hata hassasiyeti: 0.0001

```
************Menu*************

0-Quit
1-Bisection Method
2-Regula-Falsi Method
3-Newton-Rapshon Method
4-Inverse Matrix
5-Gauss Elimination
6-Gauss-Seidal
7-Numerical Differentiation
8-Simpson Method
9-Trapez Method
10-Gregory Newton Interpolation

press 0 to exit or operation number to select operation: 3
what is the exponent of the function: 3
enter the coefficients of function from lowest exponent to highest exponent
aX^0, a= :-6
bX^1, b= :14
cX^2, c= :-7
dX^3, d= :1
enter the a and b range, a: 2
b: 3
enter the error sensitivity: 0.0001
3.000000 is root
```

NxN Matrisin Tersini Alma

Bu yöntemde girilen N değerine göre NxN kare matrisin tersi alınır. N değeri 1 ise tersi 1/SAYI değeri olacaktır. N 1 için sayı sıfır ise matrisin tersi olamaz. N 2 ise 2x2 lik matrisin tersini kısa yoldan alma formülüne göre tersi alınır. N 3 veya daha büyük bir değer ise bu sefer gauss eliminasyon yöntemi ile matrisin tersi alınır. Her turda birbirinin aynı satırlar olması hasebiyle determinantın sıfır olup olmadığı kontrol edilir. Örnek girdiler ve sonuçları şöyledir:

N = 1 için:

```
************Menu*************

0-Quit
1-Bisection Method
2-Regula-Falsi Method
3-Newton-Rapshon Method
4-Inverse Matrix
5-Gauss Elimination
6-Gauss-Seidal
7-Numerical Differentiation
8-Simpson Method
9-Trapez Method
10-Gregory Newton Interpolation

press 0 to exit or operation number to select operation: 4
Please enter the N that dimension of matrix: 1
Please enter the elements of NxN matrix
matrix[0][0]: 5
```

N = 1 ve sayı sıfır için:

```
************Menu*************

0-Quit
1-Bisection Method
2-Regula-Falsi Method
3-Newton-Rapshon Method
4-Inverse Matrix
5-Gauss Elimination
6-Gauss-Seidal
7-Numerical Differentiation
8-Simpson Method
9-Trapez Method
10-Gregory Newton Interpolation

press 0 to exit or operation number to select operation: 4
Please enter the N that dimension of matrix: 1
Please enter the elements of NxN matrix
matrix[0][0]: 0
There isn't inverse of matrix because determinant is equal to zero
```

N = 2 için:

```
************Menu************

0-Quit
1-Bisection Method
2-Regula-Falsi Method
3-Newton-Rapshon Method
4-Inverse Matrix
5-Gauss Elimination
6-Gauss-Seidal
7-Numerical Differentiation
8-Simpson Method
9-Trapez Method
10-Gregory Newton Interpolation

press 0 to exit or operation number to select operation: 4
Please enter the N that dimension of matrix: 2
Please enter the elements of NxN matrix
matrix[0][0]: 2
matrix[0][1]: 4
matrix[1][0]: 1
matrix[1][1]: 3
1.500000        -2.000000
-0.500000       1.000000
```

Son olarak N = 3 için bir örnek:

```
1-Bisection Method
2-Regula-Falsi Method
3-Newton-Rapshon Method
4-Inverse Matrix
5-Gauss Elimination
6-Gauss-Seidal
7-Numerical Differentiation
8-Simpson Method
9-Trapez Method
10-Gregory Newton Interpolation

press 0 to exit or operation number to select operation: 4
Please enter the N that dimension of matrix: 3
Please enter the elements of NxN matrix
matrix[0][0]: 1
matrix[0][1]: 2
matrix[0][2]: 3
matrix[1][0]: 0
matrix[1][1]: 1
matrix[1][2]: 4
matrix[2][0]: 5
matrix[2][1]: 6
matrix[2][2]: 0

Inverse of matrix:

-24.000000      18.000000       5.000000
20.000000       -15.000000      -4.000000
-5.000000       4.000000        1.000000
```

Gauus Eliminasyon Yöntemi

Bu yöntemde lineer denklem sistemleri gauss eliminasyon yöntemi ile çözülmektedir. Yöntemin işe yarayabilmesi için denklem ve bilinmeyen sayıları birbirine eşit olmalıdır. Bunun içi yöntem başlatıldığında program bir hatırlatma yapmaktadır. Daha sonra kaç adet denklem olduğu bilgisi kullanıcıdan alınır ve sırasıyla denklemler ve sonuçları alınır. Yapılan işlemler sonucu hesaplanan matris ekrana yazdırılır.

3 bilinmeyenli 3 eşitlik için girdi ve çıktılar:

```
press 0 to exit or operation number to select operation: 5
---Reminding----
This method only solves equations with the same number of equations and unknowns
Enter the equation quantity: 3

enter your equations
coefficient of the x for 1st equation: 1
coefficient of the y for 1st equation: 2
coefficient of the z for 1st equation: -1
equal to ?: -2
coefficient of the x for 2st equation: 1
coefficient of the y for 2st equation: 0
coefficient of the z for 2st equation: 1
equal to ?: 0
coefficient of the x for 3st equation: 2
coefficient of the y for 3st equation: -1
coefficient of the z for 3st equation: -1
equal to ?: -3

Result:
x = -1.000000
y = 0.000000
z = 1.000000
```

Sayısal Türev

Bu yöntemde verilen bir denklemin türevi denklemin türevi alınmadan kendi üzerinden hesaplanır. İlk olarak fonksiyon alınır ve gerekli işlemler yapılarak ileri ve geri türev hesaplanıp ekrana yazdırılır.

Örnek girdi ve çıktı: 2X^2+1 h: 0.1 x:2 f’(2) = 8

```
what is the exponent of the function: 2
enter the coefficients of function from lowest exponent to highest exponent
aX^0, a= :1
bX^1, b= :0
cX^2, c= :2
enter the x value you want to calculate: 2
enter the h that difference value: 0.1
Your function is :

1.000000X^0 + 0.000000X^1 + 2.000000X^2

forward_derivative: 7.999973
backward_derivative: 7.999992
```

3X^3-2X+5 h:0.01 x:1 f2(1) = 7

```
what is the exponent of the function: 3
enter the coefficients of function from lowest exponent to highest exponent
aX^0, a= :5
bX^1, b= :-2
cX^2, c= :0
dX^3, d= :3
enter the x value you want to calculate: 1
enter the h that difference value: 0.01
Your function is :

5.000000X^0 + -2.000000X^1 + 0.000000X^2 + 3.000000X^3

forward_derivative: 6.999373
backward_derivative: 6.999350
```

Simpson Yöntemi

Bu yöntemde integrali hesaplanmak istenilen bir alan analitik olarak değil de sayısal yaklaşımlar ile yaklaşık olarak hesaplanır. Çeşitli zamanlarda buna ihtiyaç duyarız. Bu yöntem ile trapez yönteminden farklı olarak yamuklar yerine paraboller oluşturularak fonksiyon belirli alanlara bölünür ve bu alanların toplamıyla yaklaşık sonuç bulunur. Bunun için ilk önce fonksiyon alınır daha sonra alt ve üst limitler alınır ve en son kaça bölünerek hesaplanması istendiği sorularak gerekli işlemler yapılır ve sonuç ekrana basılır. Örnek girdi ve çıktı:

0.5X^2 -2X -8   [7, 10] aralığı   n=10   gerçek sonuç = 34,5…:

```
************Menu************

0-Quit
1-Bisection Method
2-Regula-Falsi Method
3-Newton-Rapshon Method
4-Inverse Matrix
5-Gauss Elimination
6-Gauss-Seidal
7-Numerical Differentiation
8-Simpson Method
9-Trapez Method
10-Gregory Newton Interpolation

press 0 to exit or operation number to select operation: 8
enter the function whose area you want to calculate
what is the exponent of the function: 2
enter the coefficients of function from lowest exponent to highest exponent
aX^0, a= :-8
bX^1, b= :-2
cX^2, c= :0.5
enter lower and upper limits, lower: 7
upper: 10
enter how many parts you want to divide(it should be even): 10
size of the field is 34.500000 br^2
```

Trapez Yöntemi

Bu yöntemde integrali hesaplanmak istenilen bir alan analitik olarak değil de sayısal yaklaşımlar ile yaklaşık olarak hesaplanır. Bu yöntemde fonksiyon belirlenen aralıklarda eşit dikdörtgenlere ayrılır ve yamuklar oluşur. Bu yamukların alanı toplanarak yaklaşık sonuç bulunur. Bunun için ilk önce fonksiyon alınır daha sonra alt ve üst limitler alınır ve en son kaça bölünerek hesaplanması istendiği sorularak gerekli işlemler yapılır ve sonuç ekrana basılır. Örnek girdi ve çıktı:

2X^2 +3X -1   [1,3] aralığı   n=10   gerçek sonuç = 27,33..:

```
enter the function whose area you want to calculate
what is the exponent of the function: 2
enter the coefficients of function from lowest exponent to highest exponent
aX^0, a= :-1
bX^1, b= :3
cX^2, c= :2
enter lower and upper limits, lower: 1
upper: 3
enter how many parts you want to divide: 10
size of the field is 27.360001 br^2
```

Gregory Newton Enterpolasyonu

Bu yöntemde elimizdeki noktalardan yola çıkarak bilmediğimiz ara değerleri tahmin etmeye çalışırız. Parametre olarak ilk önce kaç noktamız olduğunu giriyoruz. Daha sonra noktalar x,y ikilileri olarak giriliyor ve son olarak rahmin etmek istediğimiz değeri giriyoruz.

Örnek girdi ve çıktıları:

Nokta sayısı:5   noktalar: [(2,10),(4,50),(6,122),(8,226),(10,362)]
x=9

```
how many dots will you enter: 5
x of 1. point: 2
y of 1. point: 10
x of 2. point: 4
y of 2. point: 50
x of 3. point: 6
y of 3. point: 122
x of 4. point: 8
y of 4. point: 226
x of 5. point: 10
y of 5. point: 362
which point do you want to find the estimated value of: 9

Dot pairs:
(2,10)
(4,50)
(6,122)
(8,226)
(10,362)

 The forward difference table is:

2       10      40      32      0
4       50      72      32
6       122     104
8       226
10      value of 9.000000 is 290.000000
--------------------------------
```

Ayrıca x=8 için tahmin yapıldığında gerçek nokta ile uyuştuğu görülüyor:

Nokta sayısı:5   noktalar: [(2,10),(4,50),(6,122),(8,226),(10,362)]
x=8

```
how many dots will you enter: 5
x of 1. point: 2
y of 1. point: 10
x of 2. point: 4
y of 2. point: 50
x of 3. point: 6
y of 3. point: 122
x of 4. point: 8
y of 4. point: 226
x of 5. point: 10
y of 5. point: 362
which point do you want to find the estimated value of: 8

Dot pairs:
(2,10)
(4,50)
(6,122)
(8,226)
(10,362)

 The forward difference table is:

2       10      40      32      0
4       50      72      32
6       122     104
8       226
10      value of 8.000000 is 226.000000
```

Gauus – Seidel Yöntemi

Bu yöntem de eşit denklem ve bilinmeyene sahip lineer denklem sistemlerinin çözümünde kullanılan bir yöntemdir. Parametre olarak denklem sayısını ve akabinde denklemleri alır. Daha sonra bilinmeyenlerin başlangıç değeri alınır ve en son hata hassasiyeti alınır. Sonuç olarak bilinmeyenlerin yaklaşık değeri yazdırılır. Örnek girdi ve çıktılar şöyledir:

Denklem sayısı:3

Denklemler:

-x+4y-3z = -8 x, y, z başlangıç değeri = 1;

3x +y-2z = 9 hata hassasiyeti : 0.000001;

x -y +4z = 1 gerçek değerler -> x=3, y=-2, z=-1;

```
---Reminding----
This method only solves equations with the same number of equations and unknowns
Enter the equation quantity: 3

enter your equations
coefficient of the x for 1st equation: -1
coefficient of the y for 1st equation: 4
coefficient of the z for 1st equation: -3
equal to ?: -8
coefficient of the x for 2st equation: 3
coefficient of the y for 2st equation: 1
coefficient of the z for 2st equation: -2
equal to ?: 9
coefficient of the x for 3st equation: 1
coefficient of the y for 3st equation: -1
coefficient of the z for 3st equation: 4
equal to ?: 1
enter the initial value of x: 1
enter the initial value of y: 1
enter the initial value of z: 1
enter the error sensitivity: 0.000001
result value of x is 2.999994: result value of y is -2.000000: result value of z is -0.999998:
```

Bir başka örnek:

Denklem sayısı:3 başlangıç değerleri -> x=1, y=0, z=1 ;

Denklemler: hata hassasiyeti : 0.00001

12x+3y-5z = 1 Gerçek değerler -> x=1, y=3, z=4;

x+5y+3z = 28

3x+7y+13z = 76

```
---Reminding----
This method only solves equations with the same number of equations and unknowns
Enter the equation quantity: 3

enter your equations
coefficient of the x for 1st equation: 12
coefficient of the y for 1st equation: 3
coefficient of the z for 1st equation: -5
equal to ?: 1
coefficient of the x for 2st equation: 1
coefficient of the y for 2st equation: 5
coefficient of the z for 2st equation: 3
equal to ?: 28
coefficient of the x for 3st equation: 3
coefficient of the y for 3st equation: 7
coefficient of the z for 3st equation: 13
equal to ?: 76
enter the initial value of x: 1
enter the initial value of y: 0
enter the initial value of z: 1
enter the error sensitivity: 0.00001
result value of x is 1.000011: result value of y is 2.999993: result value of z is 4.000001:
```