

Sayısal Analiz Dersi Dönem Projesi

Öğrenci Adı-Soyadı: İsmail Cem Şahin

Öğrenci No: 19011901

İçerik: Yöntemlerin çalıştığını gösteren rapor (ekran görüntüleri) ve yöntemlerin c dilinde yazılmış kodları.

(hepsi yapıldı)

1-Bisection

Kullanılan denklem: $x^3-7x^2+14x-6=0$

Aralık: [0,1]

Maksimum iterasyon: 20

Hata: 0.0001

-----BISECTION METODU-----

Denklem kacinci dereceden: 3

1 X kat sayi degeri: 1

2 X kat sayi degeri: -7

3 X kat sayi degeri: 14

C sabit degeri: -6

$f(x) = 1.00x^3 + -7.00x^2 + 14.00x^1 + -6$

Fonskiyonun Surekli Oldugu 2 Araligi Giriniz

Aralik 1: 0

Aralik 2: 1

Max Iterasyon Sayisi: 20

Hata Degeri (Lutfen virgul yerine nokta kullaniniz): 0.0001

	a:	b:	f(a):	f(b):	hata:
1. Itersayon -	[0.000000,1.000000]		-6.000000	2.000000	
2. Itersayon -	[0.500000,1.000000]		-0.625000	2.000000	0.125000
3. Itersayon -	[0.500000,0.750000]		-0.625000	0.984375	0.031250
4. Itersayon -	[0.500000,0.625000]		-0.625000	0.259766	0.007813
5. Itersayon -	[0.562500,0.625000]		-0.161865	0.259766	0.001953
6. Itersayon -	[0.562500,0.593750]		-0.161865	0.054047	0.000488
7. Itersayon -	[0.578125,0.593750]		-0.052624	0.054047	0.000122
8. Itersayon -	[0.578125,0.585938]		-0.052624	0.001031	0.000031

Process exited after 67.21 seconds with return value 0

Press any key to continue . . .

2-Regula-Falsi

Kullanılan denklem: $x^3-7x^2+14x-6=0$

Aralık: [0,1]

Hata: 0.0001

```
-----REGULA FALSI METODU-----

Denklem kacinci dereceden: 3
1. X kat sayi degeri: 1
2. X kat sayi degeri: -7
3. X kat sayi degeri: 14
C sabit degeri: -6

f(x)= 1.00x^3 + -7.00x^2 + 14.00x^1 + -6

Fonskiyonun Surekli Oldugu 2 Araligi Giriniz

Aralik 1: 0
Aralik 2: 1
Hata Degeri (Lutfen virgul yerine nokta kullaniniz): 0.0001

      a:      b:      f(a):      f(b):      hata:
1. Itersayon - [0.000000,1.000000] : -6.000000 2.000000
2. Itersayon - [0.000000,0.750000] : -6.000000 0.984375 0.187500
3. Itersayon - [0.000000,0.644295] : -6.000000 0.381777 0.080537
4. Itersayon - [0.000000,0.605752] : -6.000000 0.134249 0.037859
5. Itersayon - [0.000000,0.592495] : -6.000000 0.045571 0.018515
6. Itersayon - [0.000000,0.588028] : -6.000000 0.015283 0.009188
7. Itersayon - [0.000000,0.586534] : -6.000000 0.005105 0.004582
8. Itersayon - [0.000000,0.586036] : -6.000000 0.001703 0.002289
9. Itersayon - [0.000000,0.585870] : -6.000000 0.000567 0.001144
10. Itersayon - [0.000000,0.585814] : -6.000000 0.000190 0.000572
11. Itersayon - [0.000000,0.585796] : -6.000000 0.000063 0.000286
12. Itersayon - [0.000000,0.585790] : -6.000000 0.000021 0.000143
13. Itersayon - [0.000000,0.585787] : -6.000000 0.000008 0.000072

-----
Process exited after 41.75 seconds with return value 0
Press any key to continue . . .
```

3-Newton-Raphson

Kullanılan denklem: $x^3-7x^2+14x-6=0$

Aralık: [0,1]

$X_0 = 0$

Hata: 0.0001

-----NEWTON RAPHSON-----

Denklem kacinci dereceden: 3

1 X kat sayi degeri: 1

2 X kat sayi degeri: -7

3 X kat sayi degeri: 14

C sabit degeri: -6

$f(x) = 1.00x^3 + -7.00x^2 + 14.00x^1 + -6$

Fonskiyonun Surekli Oldugu 2 Araligi Giriniz

Aralik 1: 0

Aralik 2: 1

X_0 : 0

Hata Degeri (Lutfen virgul yerine nokta kullaniniz): 0.0001

	a:	f(a):	f'(a):	b(son kok degeri):	hata:
1. Iterasyon:	0.000000	-6.000000	14.000000	0.428571	0.428571
2. Iterasyon:	0.428571	-1.206997	8.551021	0.569724	0.141152
3. Iterasyon:	0.569724	-0.111039	6.997622	0.585592	0.015868
4. Iterasyon:	0.585592	-0.001328	6.830466	0.585786	0.000194
5. Iterasyon:	0.585786	-0.000001	6.828428	0.585786	0.000000

Process exited after 37.41 seconds with return value 0

Press any key to continue . . .

4-NxN'lik Matrisin Tersi

Kullanılan Matris:

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 3 & 1 & -2 \\ 2 & -1 & 4 \end{bmatrix}$$

-----NxN MATRISININ TERSI-----

Kare matrisin uzunlugu (N*N) N degeri: 3
3X3 Matrisinin degerlerini sirasiyla giriniz

[1][1]: 1
[1][2]: 2
[1][3]: 1
[2][1]: 3
[2][2]: 1
[2][3]: -2
[3][1]: 2
[3][2]: -1
[3][3]: 4

Girilen Matris:

1.000000	2.000000	1.000000
3.000000	1.000000	-2.000000
2.000000	-1.000000	4.000000

Matrisinizin tersi:

-0.057143	0.257143	0.142857
0.457143	-0.057143	-0.142857
0.142857	-0.142857	0.142857

Process exited after 12.87 seconds with return value 10
Press any key to continue . . .

5- Gauss Eliminasyon

Kullanılan lineer denklem sistemleri:

$$2x_1+8x_2+2x_3=14$$

$$x_1+6x_2-x_3=13$$

$$2x_1-x_2+2x_3=5$$

```
-----GAUSS ELIMINASYON-----
```

```
Bilinmeyen Sayisi = Denklem Sayisi: 3
```

```
Satir bazinda matrix degerlerini giriniz:
```

```
[1][1]: 2  
[1][2]: 8  
[1][3]: 2  
[1][4]: 14  
[2][1]: 1  
[2][2]: 6  
[2][3]: -1  
[2][4]: 13  
[3][1]: 2  
[3][2]: -1  
[3][3]: 2  
[3][4]: 5
```

```
Sonuc:
```

```
x1=5.000000  
x2=1.000000  
x3=-2.000000
```

```
-----  
Process exited after 70.77 seconds with return value 0  
Press any key to continue . . .
```

6-Gauss Seidel

Kullanılan:

$$A \cdot x = b$$

$$\begin{bmatrix} 5 & 2 & 1 \\ 2 & 3 & 4 \\ -4 & 5 & 3 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 12 \\ 20 \\ 15 \end{bmatrix}$$

-----GAUSS SEIDEL-----

Toplam Denklem Sayisini Giriniz: 3
Hata degeri: 0.0001

Katsayilarini satir diziliminde Giriniz (Son deger esitli-in sagindaki degerdir):

```
Matrix[1][1] = 5  
Matrix[1][2] = 2  
Matrix[1][3] = 1  
Matrix[1][4] = 12  
Matrix[2][1] = -4  
Matrix[2][2] = 5  
Matrix[2][3] = 3  
Matrix[2][4] = 15  
Matrix[3][1] = 2  
Matrix[3][2] = 3  
Matrix[3][3] = 4  
Matrix[3][4] = 20
```

1. Iterasyon-

```
Y[1]= 2.400000  
Y[2]= 4.920000  
Y[3]= 0.110000
```

2. Iterasyon-

```
Y[1]= 0.410000  
Y[2]= 3.262000  
Y[3]= 2.348500
```

3. Iterasyon-

```
Y[1]= 0.625500  
Y[2]= 2.091300  
Y[3]= 3.118775
```

4. Iterasyon-

```
Y[1]= 0.939725  
Y[2]= 1.880515  
Y[3]= 3.119751
```

5. Iterasyon-

```
Y[1]= 1.023844  
Y[2]= 1.947224  
Y[3]= 3.027660
```

Ekran görüntüsü 1

Ekran görüntüsü 2

```
6. Iterasyon-
Y[1]= 1.015578
Y[2]= 1.995867
Y[3]= 2.995311

7. Iterasyon-
Y[1]= 1.002591
Y[2]= 2.004886
Y[3]= 2.995040

8. Iterasyon-
Y[1]= 0.999038
Y[2]= 2.002206
Y[3]= 2.998827

9. Iterasyon-
Y[1]= 0.999352
Y[2]= 2.000186
Y[3]= 3.000185

10. Iterasyon-
Y[1]= 0.999889
Y[2]= 1.999800
Y[3]= 3.000206

11. Iterasyon-
Y[1]= 1.000039
Y[2]= 1.999908
Y[3]= 3.000050

12. Iterasyon-
Y[1]= 1.000027
Y[2]= 1.999992
Y[3]= 2.999993

SONUC

Y[1]: 1.000027
Y[2]: 1.999992
Y[3]: 2.999993
-----
Process exited after 25.68 seconds with return value 0
Press any key to continue . . .
```


7- Sayısal Türev (merkezi,ileri ve geri)

Kullanılan denklem: $x^3-7x^2+14x-6=0$

$x=1$

$h=0.1$

```
-----SAYISAL TUREV-----
```

```
Denklem kacinci dereceden: 3
```

```
1. X kat sayi degeri: 1
```

```
2. X kat sayi degeri: -7
```

```
3. X kat sayi degeri: 14
```

```
C sabit degeri: -6
```

```
X degeri: 1
```

```
H degeri: 0.1
```

```
f(x)= 1.00x^3 + -7.00x^2 + 14.00x^1 + -6
```

```
x degeri: 1.00
```

```
h degeri: 0.10
```

```
MENU:
```

```
1) Geri Farklar Ile Sayisal Turev
```

```
2) Ileri Farklar Ile Sayisal Turev
```

```
3) Merkezi Farklar Ile Sayisal Turev
```

```
4) Hepsi
```

```
Lutfen kullanmak istediginiz islemin numarasini giriniz: 4
```

```
Geri farklar sonuc: 3.410001
```

```
Ileri farklar sonuc: 2.610006
```

```
Merkezi farklar sonuc: 3.010004
```

```
-----
```

```
Process exited after 25.74 seconds with return value 0
```

```
Press any key to continue . . . █
```


8- a) Simpson Yöntemi (1/3)

Kullanılan:

n=2

$$\int_{-1}^3 (x^3 + 1) dx$$

```
-----SIMPSON 1/3-----
```

```
Denklem kacinci dereceden: 3
```

```
1. X kat sayi degeri: 1
```

```
2. X kat sayi degeri: 0
```

```
3. X kat sayi degeri: 0
```

```
C sabit degeri: 1
```

```
f(x)= 1.00x^3 + 0.00x^2 + 0.00x^1 + 1
```

```
N degerini giriniz (sadece cift sayilar): 2
```

```
Baslangic limit degerini giriniz (a): -1
```

```
Son limit degerini giriniz (b): 3
```

```
Sonuc: 24.000000
```

```
-----
```

```
Process exited after 21.42 seconds with return value 0
```

```
Press any key to continue . . . █
```

8- b) Simpson Yöntemi (3/8)

Kullanılan:

Başlangıç limit değeri a= 0

Son limit değeri b=2

n=6

$$S_2 = \int_0^2 x^3 dx$$

```
-----SIMPSON 3/8-----
```

```
Denklem kacinci dereceden: 3
```

```
1. X kat sayi degeri: 1
```

```
2. X kat sayi degeri: 0
```

```
3. X kat sayi degeri: 0
```

```
C sabit degeri: 0
```

```
f(x)= 1.00x^3 + 0.00x^2 + 0.00x^1 + 0
```

```
N degerini giriniz (sadece 3un katlari): 6
```

```
Baslangic limit degerini giriniz (a): 0
```

```
Son limit degerini giriniz (b): 2
```

```
Sonuc: 4.000000
```

```
-----
```

```
Process exited after 21.05 seconds with return value 0
```

```
Press any key to continue . . . █
```

9- Trapez yöntemi

Kullanılan:

Başlangıç limit değeri a= 0

Son limit değeri b=2

n=4

$$S_2 = \int_0^2 x^3 dx$$

```
-----TRAPEZ YONTEMI-----
```

```
Denklem kacinci dereceden: 3
```

```
1. X kat sayi degeri: 1
```

```
2. X kat sayi degeri: 0
```

```
3. X kat sayi degeri: 0
```

```
C sabit degeri: 0
```

```
f(x)= 1.00x^3 + 0.00x^2 + 0.00x^1 + 0
```

```
N degerini giriniz: 4
```

```
Baslangic limit degerini giriniz (a): 0
```

```
Son limit degerini giriniz (b): 2
```

```
Sonuc: 4.250000
```

```
-----
```

```
Process exited after 23.86 seconds with return value 0
```

```
Press any key to continue . . . █
```

10- Newton Interpolasyon

Kullanılan:

x	0	5	10	15	20
f(x)	1	1.6	3.8	8.2	15.4

```
-----NEWTON INTERPOLASYON-----  
  
Incenecek olan data miktarini giriniz(max 10): 5  
  
---X DEGERLERINI GIRINIZ---  
  
1. x degeri: 0  
2. x degeri: 5  
3. x degeri: 10  
4. x degeri: 15  
5. x degeri: 20  
  
---Y DEGERLERINI GIRINIZ---  
  
1. Y degeri: 1  
2. Y degeri: 1.6  
3. Y degeri: 3.8  
4. Y degeri: 8.2  
5. Y degeri: 15.4  
  
Bulunmasi istenen f(x) degeri x: 3  
  
f(3.000000) = 1.201600  
-----  
Process exited after 22.47 seconds with return value 0  
Press any key to continue . . .
```

C Kodları

Bisection:

```
#include <stdio.h>
int main()
{
    float arraykat[500];
    int arrayderece[500];
    float sabit, aralik1, aralik2, hata, yeniaralik;
    int i, xsayi, derece;
    int temp, sayi;
    int d, g, h;
    float frx;
    float toplam;
    float cevap1, cevap2, cevap3, sonuc;
    int bekle;
    int ais1, ais2, ais3, bilgi;
    int flag, itersayon=1;
    int iterasyonsayi=1;
    int itsinir;
    float kok, koka;
    float hatasinir;
    float kokhata(int iter)
    {
        kok=2;
        for(i=1; i<iter; i++) {
            kok*=2;
        }
        return kok;
    }
    int kontrol(cevap1, cevap2, cevap3)
    {
        if(cevap1>0) {
            ais1=1;
        }
        else{
            ais1=-1;
        }

        if(cevap2>0) {
            ais2=1;
        }
        else{
            ais2=-1;
        }
        if(cevap3>0) {
            ais3=1;
        }
        else{
            ais3=-1;
        }
    }
}
```

```

        if(ais3<ais2 || ais3>ais1){
            bilgi=1;
            return bilgi; // Yeni Aralik Ile 1. Aralik
Arasinda Kok Var
        }
        else{
            bilgi=0;
            return bilgi; // Yeni Aralik Ile 2. Aralik
Arasinda Kok Var
        }
    }
    float aralik (float aralik1, float aralik2){
        yeniaralik = (aralik1+aralik2)/2;
        return yeniaralik;
    }
    float fonkhesap(float xdeger)
    {
        toplam=0;
        for(i=0; i<derece; i++){
            frx=xdeger;
            for(h=1; h<derece-i; h++){
                frx*=xdeger;
            }
            frx*=arraykat[i];
            toplam +=frx;
        }
        toplam+=sabit;
        return toplam;
    }
    printf("-----BISECTION METODU-----\n");
    printf("\n");
    printf("Denklem kacinci dereceden: ");
    scanf("%d",&derece);
    xsayi=derece;
    for(i=0; i<derece; i++) {
        arrayderece[i]=xsayi;
        xsayi--;
    }
    for(i=0; i<derece; i++){
        printf("%d. X kat sayi degeri: ",i+1);
        scanf("%f",&arraykat[i]);
    }
    printf("C sabit degeri: ");
    scanf("%f",&sabit);
    printf("\n");
    printf("f(x)= ");
    for(i=0; i<derece; i++){
        printf("%.2fx^%d + ",arraykat[i],arrayderece[i]);
    }
    printf("%.2f",sabit);
    printf("\n");
    printf("\n");
    printf("Fonskiyonun Surekli Oldugu 2 Araligi Giriniz\n");
    printf("\n");
    printf("Aralik 1: ");
    scanf("%f",&aralik1);
    printf("Aralik 2: ");

```



```

scanf("%f",&aralik2);
printf("Max Iterasyon Sayisi: ");
scanf("%d",&itsinir);
printf("Hata Degeri (Lutfen virgul yerine nokta kullaniniz):
");

scanf("%f",&hatasinir);
printf("\n");
cevap1=fonkhesap(aralik1);
cevap2=fonkhesap(aralik2);
sonuc=cevap1*cevap2;
hata= cevap2-cevap1;
printf("          a:          b:          f(a):          f(b):
hata:\n");
printf("%d. Itersayon - [%f,%f] : %f
%f\n",iterasyonsayi,aralik1,aralik2,cevap1,cevap2);
if(sonuc>0){
    printf("Bu Aralikta Kok Yoktur\n");
    exit(0);
}
while((iterasyonsayi<itsinir) && (hata>hatasinir))
{
    yeniaralik = aralik(aralik1,aralik2);
    cevap3=fonkhesap(yeniaralik);
    flag = kontrol(cevap1,cevap2,cevap3);
    iterasyonsayi++;
    if(flag==1) // 1. Aralik ile yeni aralik
    {
        aralik2=yeniaralik;
        cevap2=cevap3;
        printf("%d. Itersayon - [%f,%f] : %f
%f",iterasyonsayi,aralik1,aralik2,cevap1,cevap2);
        hata = aralik2-aralik1;
        koka= kokhata(iterasyonsayi);
        hata = hata/koka;
    }
    else{ // Yeni Aralik Ile 2. Aralik Arasinda Kok Var
        aralik1=yeniaralik;
        cevap1=cevap3;
        printf("%d. Itersayon - [%f,%f] : %f
%f",iterasyonsayi,aralik1,aralik2,cevap1,cevap2);
        hata = aralik2-aralik1;
        koka= kokhata(iterasyonsayi);
        hata = hata/koka;
    }

    printf(" %f\n",hata);
}
return 0;
}

```

Regula-Falsi:

```
#include <stdio.h>
int main()
{
    float arraykat[500];
    int arrayderece[500];
    float sabit, aralik1, aralik2, hata, yeniaralik;
    int i, xsayi, derece;
    int temp, sayi;
    int d, g, h;
    float frx;
    float toplam;
    float cevap1, cevap2, cevap3=1, sonuc;
    int bekle;
    int ais1, ais2, ais3, bilgi;
    int flag, itersayon=1;
    int iterasyonsayi=1;
    float hatasinir;
    float kok, koka;
    float tt1, tt2, tt3;
    float kokhata(int iter){
        kok=2;
        for(i=1; i<iter; i++){
            kok*=2;
        }
        return kok;
    }
    int kontrol(cevap1, cevap2, cevap3)
    {
        if(cevap1>0){
            ais1=1;
        }
        else{
            ais1=-1;
        }
        if(cevap2>0){
            ais2=1;
        }
        else{
            ais2=-1;
        }
        if(cevap3>0){
            ais3=1;
        }
        else{
            ais3=-1;
        }
        if(ais3<ais2 || ais3>ais1)
        {
            bilgi=1;
            return bilgi; // Yeni Aralik Ile 1. Aralik
```

Arasinda Kok Var

```
    }
    else
    {
        bilgi=0;
        return bilgi;    // Yeni Aralik Ile 2. Aralik
Arasinda Kok Var
    }
}
float aralik(float aralik1,float aralik2)
{
    tt1=(aralik1*cevap2);
    tt2=(aralik2*cevap1);
    tt3=(cevap2-cevap1);
    yeniaralik=(tt1-tt2)/tt3;
    return yeniaralik;
}
float fonkhesap(float xdeger)
{
    toplam=0;
    for(i=0;i<derece;i++){
        frx=xdeger;
        for(h=1;h<derece-i;h++){
            frx*=xdeger;
        }
        frx*=arraykat[i];
        toplam +=frx;
    }

    toplam+=sabit;
    return toplam;
}
printf("-----REGULA FALSI METODU-----\n");
printf("\n");
printf("Denklem kacinci dereceden: ");
scanf("%d",&derece);
xsayi=derece;
for(i=0; i<derece; i++){
    arrayderece[i]=xsayi;
    xsayi--;
}
for(i=0; i<derece; i++){
    printf("%d. X kat sayi degeri: ",i+1);
    scanf("%f",&arraykat[i]);
}
printf("C sabit degeri: ");
scanf("%f",&sabit);
printf("\n");
printf("f(x)= ");
for(i=0;i<derece;i++){
    printf("%.2fx^%d + ",arraykat[i],arrayderece[i]);
}
printf("%.2f",sabit);
printf("\n");
printf("\n");
printf("Fonksiyonun Surekli Oldugu 2 Araligi Giriniz\n");
printf("\n");
```

```

printf("Aralik 1: ");
scanf("%f",&aralik1);
printf("Aralik 2: ");
scanf("%f",&aralik2);
printf("Hata Degeri (Lutfen virgul yerine nokta kullaniniz): ");
scanf("%f",&hata);
printf("\n");
cevap1=fonkhesap(aralik1);
cevap2=fonkhesap(aralik2);
hata= cevap2-cevap1;
sonuc=cevap1*cevap2;
printf("          a:          b:          f(a):          f(b):
hata:\n");
printf("%d. Itersayon - [%f,%f] : %f
%f\n", iterasyonsayi, aralik1, aralik2, cevap1, cevap2);
if(sonuc>0){
    printf("Bu Aralikta Kok Yoktur\n");
    exit(0);
}
while(hata>hata)
{
    yeniaralik = aralik(aralik1, aralik2);
    cevap3=fonkhesap(yeniaralik);
    flag = kontrol(cevap1, cevap2, cevap3);
    iterasyonsayi++;
    if(flag==1) // 1. Aralik ile yeni aralik
    {
        aralik2=yeniaralik;
        cevap2=cevap3;
        printf("%d. Itersayon - [%f,%f] : %f
%f", iterasyonsayi, aralik1, aralik2, cevap1, cevap2);
        hata = aralik2 - aralik1;
        koka= kokhata(iterasyonsayi);
        hata = hata/koka;
    }
    else{ // Yeni Aralik Ile 2. Aralik Arasinda Kok Var
        aralik1=yeniaralik;
        cevap1=cevap3;
        printf("%d. Itersayon - [%f,%f] : %f
%f", iterasyonsayi, aralik1, aralik2, cevap1, cevap2);
        hata = aralik2-aralik1;
        koka= kokhata(iterasyonsayi);
        hata = hata/koka;
    }
    printf(" %f\n", hata);
}
return 0;
}

```

Newton Raphson

```
#include <stdio.h>

int main()
{
    float arraykat[500];
    int arrayderece[500];
    float
sabit, aralik1, aralik2, hata, yeniaralik, ustsinir, altsinir;
    int i, xsayi, derece;
    int temp, sayi;
    int d, g, h;
    float frx, x0;
    float toplam, a, sonuca, sonucb;
    float cevap1, cevap2, cevap3, sonuc;
    int bekle;
    int ais1, ais2, ais3, bilgi;
    int flag, iterasyon=1;
    int iterasyonsayi=1;
    int itsinir;
    float kok, koka;
    float hatasinir;
    int c;
    float aralik(float a, float sonuca, float sonucb) {
        yeniaralik = a - (sonuca/sonucb);
        return yeniaralik;
    }
    float turev (float xdeger) {
        toplam =0;
        for(i=0; i<derece-1; i++) {
            {
                frx=xdeger;
                for (h=2; h<derece-i; h++) {
                    frx*=xdeger;
                }
                toplam+=frx*arraykat[i]*arrayderece[i];
            }
        }
        toplam+=arraykat[derece-1];
        sonucb=toplam;
        return sonucb;
    }
    float fonkhesap(float xdeger) {
        toplam=0;
        for(i=0; i<derece; i++) {
            frx=xdeger;
            for (h=1; h<derece-i; h++) {
                frx*=xdeger;
            }
            frx*=arraykat[i];
            toplam +=frx;
        }
    }
}
```

```

        toplam+=sabit;
        sonuca=toplam;
        return sonuca;
    }
    printf("-----NEWTON RAPHSON-----\n");
    printf("\n");
    printf("Denklem kacinci dereceden: ");
    scanf("%d",&derece);
    xsayi=derece;
    for(i=0; i<derece; i++) {
        arrayderece[i]=xsayi;
        xsayi--;
    }
    for(i=0; i<derece; i++) {
        printf("%d. X kat sayi degeri: ",i+1);
        scanf("%f",&arraykat[i]);
    }

    printf("C sabit degeri: ");
    scanf("%f",&sabit);
    printf("\n");
    printf("f(x)= ");
    for(i=0;i<derece;i++){
        printf("%.2fx^%d + ",arraykat[i],arrayderece[i]);
    }
    printf("%.2f",sabit);
    printf("\n");
    printf("\n");
    printf("Fonskiyonun Surekli Oldugu 2 Araligi Giriniz\n");
    printf("\n");
    printf("Aralik 1: ");
    scanf("%f",&ustsinir);
    printf("Aralik 2: ");
    scanf("%f",&altsinir);
    printf("X0: ");
    scanf("%f",&x0);
    printf("Hata Degeri (Lutfen virgul yerine nokta kullaniniz):
");

    scanf("%f",&hataasinir);
    printf("\n");
    aralik1=x0;
    hata = 999;
    printf("
a: f(a):
f'(a): b(son kok degeri): hata:\n");
    while(hata>hataasinir){
        cevap1=fonkhesap(aralik1);
        cevap2=turev(aralik1);
        aralik2 = aralik(aralik1,cevap1,cevap2);
        printf("%d. Iterasyon: %f %f %f
%f",iterasyon,aralik1,cevap1,cevap2,aralik2);
        hata = aralik2-aralik1;
        if (hata<0){
            hata*=-1;
        }

        aralik1=aralik2;
        iterasyon++;
        printf("%f\n",hata);
    }

```

```
        return 0;  
    }
```

NxN'lik bir matrisin tersi

```
#include<stdio.h>
#include<math.h>
void transpoze(float [][][36], float [][][36], float);
float determinant(float [][][36], float);
void kofaktor(float [][][36], float);
int main()
{
    printf("-----NxN MATRISININ TERSI-----\n\n");
    int i, l;
    float a[36][36];
    float k,d;
    printf("Kare matrisin uzunlugu (N*N)  N degeri: ");
    scanf("%f", &k);
    printf("%.0fX%.0f  Matrisinin  degerlerini  sirasiyla
    giriniz\n", k, k);
    for (i = 0;i < k; i++){
        for (l = 0;l < k; l++){
            printf("[%d][%d]: ",i+1,l+1);
            scanf("%f", &a[i][l]);
        }
    }
    printf("\n");
    printf("Girilen Matris:");
    for (i = 0;i < k; i++){
        printf("\n");
        for (l = 0;l < k; l++){
            printf("%f \t",a[i][l]);
        }
    }
    d = determinant(a, k);
    if (d == 0)
        printf("\nMatrisiniz tersi alinamiyor\n");
    else
        kofaktor(a, k);
}

float determinant(float a[36][36], float k)
{
    float s = 1, determ2 = 0, g[36][36];
    int i, l, v, n, c;
    if (k == 1){
        return (a[0][0]);
    }
    else{
        determ2 = 0;
        for (c = 0; c < k; c++){
            {
                v = 0;
                n = 0;
                for (i = 0;i < k; i++){
                    for (l = 0 ;l < k; l++){
                        g[i][l] = 0;
                        if (i != 0 && l != c)
                        {
```



```

        g[v][n] = a[i][1];
        if (n < (k - 2))
            n++;
        else{
            n = 0;
            v++;
        }
    }
}
determ2 = determ2 + s * (a[0][c] * determinant(g, k
- 1));
s = -1 * s;
}

return (determ2);
}

void kofaktor(float say[36][36], float f)
{
    int p, y, v, n, i, l;
    float g[36][36], fakt[36][36];

    for (y = 0; y < f; y++)
    {
        for (p = 0; p < f; p++)
        {
            n = 0;
            v = 0;
            for (i = 0; i < f; i++){
                for (l = 0; l < f; l++){
                    if (i != y && l != p){
                        g[v][n] = say[i][l];
                        if (n < (f - 2))
                            n++;
                        else{
                            n = 0;
                            v++;
                        }
                    }
                }
            }
            fakt[y][p] = pow(-1, y + p) * determinant(g, f - 1);
        }
    }
    transpose(say, fakt, f);
}

void transpose(float say[36][36], float fakt[36][36], float
e){
    int i, l;
    float g[36][36], ters[36][36], d;

    for (i = 0; i < e; i++)
    {
        for (l = 0; l < e; l++)

```

```
        {
            g[i][l] = fakt[l][i];
        }
    }
    d = determinant(say, e);
    for (i = 0; i < e; i++) {
        for (l = 0; l < e; l++) {
            ters[i][l] = g[i][l] / d;
        }
    }
    printf("\n\nMatrisinizin tersi: \n");
    for (i = 0; i < e; i++)
    {
        for (l = 0; l < e; l++)
        {
            printf("\t%f", ters[i][l]);
        }
        printf("\n");
    }
}
```

Gauss Eliminasyon

```
#include<stdio.h>
int main()
{
    printf("-----GAUSS ELIMINASYON-----\n\n");
    int y,i,n,l;
    float r[30][30],c,x[15],toplam=0.0;
    printf("Bilinmeyen Sayisi = Denklem Sayisi: ");
    scanf("%d",&n);
    printf("\nSatir bazinda matrix degerlerini giriniz:\n\n");
    for(i=1; i<=n; i++)
    {
        for(y=1; y<=(n+1); y++){
            printf("[%d] [%d]: ", i,y);
            scanf("%f",&r[i][y]);
        }
    }
    for(y=1; y<=n; y++){
        for(i=1; i<=n; i++){
            if(i>y){
                c=r[i][y]/r[y][y];
                for(l=1; l<=n+1; l++){
                    r[i][l]=r[i][l]-c*r[y][l];
                }
            }
        }
    }
    x[n]=r[n][n+1]/r[n][n];
    for(i=n-1; i>=1; i--){
        toplam=0;
        for(y=i+1; y<=n; y++){
            toplam=toplam+r[i][y]*x[y];
        }
        x[i]=(r[i][n+1]-toplam)/r[i][i];
    }
    printf("\nSonuc: \n");
    for(i=1; i<=n; i++){
        printf("\nx%d=%f\t",i,x[i]);
    }
    return(0);
}
```

Gauss Seidel

```
#include<stdio.h>
#include<math.h>

int main()
{
    int i, t, limit, iterasyon=1;
    float temp, hata, a, toplam = 0;
    float matrix[10][10], y[10], hatasinir;
    printf("-----GAUSS SEIDEL-----\n");
    printf("\nToplam Denklem Sayisini Giriniz: ");
    scanf("%d", &limit);
    printf("Hata degeri: ");
    scanf("%f", &hatasinir);
    printf("\nKatsayialarini satir diziliminde Giriniz (Son deger
    esitligin sagindaki degerdir):\n");
    for(i = 1; i <= limit; i++)
    {
        for(t = 1; t <= limit + 1; t++)
        {
            printf("Matrix[%d][%d] = ", i, t);
            scanf("%f", &matrix[i][t]);
        }
    }
    for(i = 1; i <= limit; i++)
    {
        y[i] = 0;
    }
    do
    {
        printf("\n");
        printf("%d. Iterasyon- ", iterasyon);
        iterasyon++;
        a = 0;
        for(i = 1; i <= limit; i++)
        {
            toplam = 0;

            for(t = 1; t <= limit; t++)
            {
                if(t != i)
                {
                    toplam = toplam + matrix[i][t] * y[t];
                }
            }
            temp = (matrix[i][limit + 1] - toplam) / matrix[i][i];
            hata = fabs(y[i] - temp);
            if(hata > a)
            {
                a = hata;
            }
            y[i] = temp;
            printf("\nY[%d]=\t%f", i, y[i]);
        }
        printf("\n");
    }
```

```
    }  
    while(a >= hatasinir);  
    printf("\n\nSONUC\n");  
    for(i = 1; i <= limit; i++)  
    {  
        printf("\nY[%d]:\t%f", i, y[i]);  
    }  
    return 0;  
}
```

Sayısal Türev (merkezi, ileri ve geri)

```
#include<stdio.h>

int main (){
    float arraykat[500],sabit,x,h2,toplam,frx,sonuc;
    int arrayderece[500];
    int i,xsayi,derece,islem,h;

    float fonkhesap(float xdeger)
    {
        toplam=0;

        for(i=0;i<derece;i++){
            frx=xdeger;
            for(h=1;h<derece-i;h++)
            {
                frx*=xdeger;

            }

            frx*=arraykat[i];
            toplam +=frx;
        }

        toplam+=sabit;

        return toplam;
    }

    printf("-----SAYISAL TUREV-----\n");
    printf("\n");

    printf("Denklem kacinci dereceden: ");
    scanf("%d",&derece);
    printf("\n");
    xsayi=derece;
    for(i=0; i<derece; i++)    {
        arrayderece[i]=xsayi;
        xsayi--;
    }

    for(i=0; i<derece; i++)    {

        printf("%d. X kat sayi degeri: ",i+1);
        scanf("%f",&arraykat[i]);
    }

    printf("C sabit degeri: ");
    scanf("%f",&sabit);
    printf("X degeri: ");
    scanf("%f",&x);
    printf("H degeri: ");
```

```

scanf("%f",&h2);
printf("\n");

printf("f(x)= ");
for(i=0;i<derece;i++){
    printf("%.2fx^%d + ",arrykat[i],arryderece[i]);

}
printf("%.2f\n",sabit);
printf("x degeri: %.2f\n",x);
printf("h degeri: %.2f",h2);
printf("\n\n\n");

printf("    MENU:\n\n");
printf("1) Geri Farklar Ile Sayisal Turev\n");
printf("2) Ileri Farklar Ile Sayisal Turev\n");
printf("3) Merkezi Farklar Ile Sayisal Turev\n");
printf("4) Hepsi\n\n");
printf("Lutfen kullanmak istediginiz islemin numarasini
giriniz: ");
scanf("%d",&islem);
printf("\n\n");

if(islem==1){
    sonuc = (fonkhesap(x) - fonkhesap(x-h2))/h2;
    printf("Geri farklar sonuc: %f",sonuc);
    return 0;
}
if(islem==2){
    sonuc = (fonkhesap(x+h2) - fonkhesap(x))/h2;
    printf("Ileri farklar sonuc: %f",sonuc);
    return 0;
}
if(islem==3){
    sonuc = (fonkhesap(x+h2) - fonkhesap(x-h2))/(2*h2);
    printf("Merkezi farklar sonuc: %f",sonuc);
    return 0;
}
if(islem==4){
    sonuc = (fonkhesap(x) - fonkhesap(x-h2))/h2;
    printf("Geri farklar sonuc: %f\n",sonuc);
    sonuc = (fonkhesap(x+h2) - fonkhesap(x))/h2;
    printf("Ileri farklar sonuc: %f\n",sonuc);
    sonuc = (fonkhesap(x+h2) - fonkhesap(x-h2))/(2*h2);
    printf("Merkezi farklar sonuc: %f\n\n\n",sonuc);
    return 0;
}
else{
    printf("\n\n\nYanlis deger girdiniz, tekrar kullanmak
icin programi kapatip acin.....\n\n\n");

}

return 0;
}

```


Simpson 1/3

```
#include<stdio.h>
#include<math.h>

int main() {
    double arraykat[500],a,sabit,b,h2,toplam,h,x,frx,sonuc;
    int arrayderece[500],n,i,c,q,derece;
    double tpl=0,xsayi;

    double fonkhesap(double x){
        toplam=0;

        for(i=0;i<derece;i++){
            frx=x;
            for(c=1;c<derece-i;c++){
                frx*=x;
            }
            frx*=arraykat[i];
            toplam +=frx;
        }
        toplam+=sabit;
        return toplam;
    }

    printf("-----SIMPSON 1/3-----\n");
    printf("\n");
    printf("Denklem kacinci dereceden: ");
    scanf("%d",&derece);
    printf("\n");
    xsayi=derece;
    for(i=0; i<derece; i++) {
        arrayderece[i]=xsayi;
        xsayi--;
    }
    for(i=0; i<derece; i++) {
        printf("%d. X kat sayi degeri: ",i+1);
        scanf("%lf",&arraykat[i]);
    }
    printf("C sabit degeri: ");
    scanf("%lf",&sabit);

    printf("\nf(x)= ");
    for(i=0;i<derece;i++){
        printf("%.2lfx^%d + ",arraykat[i],arrayderece[i]);
    }
    printf("%.2lf\n\n",sabit);

    printf("N degerini giriniz (sadece cift sayilar): ");
    scanf("%d",&n);
    if(n%2 != 0){
        printf("Cift sayi girmediniz");
        return 0;
    }
}
```

```
printf("Baslangic limit degerini giriniz (a): ");
scanf("%lf",&a);
printf("Son limit degerini giriniz (b): ");
scanf("%lf",&b);
h=fabs(b-a)/n;
for(q=1;q<n;q++){
    x=a+q*h;
    if(q%2==0){
        tpl=tpl+2*fonkhesap(x);
    }
    else{
        tpl=tpl+4*fonkhesap(x);
    }
}
sonuc=(h/3)*(fonkhesap(a)+fonkhesap(b)+tpl);
printf("\nSonuc: %lf\n",sonuc);
return 0;
}
```

Simpson 3/8

```
#include<stdio.h>
#include<math.h>

int main() {
    double arraykat[500],a,sabit,b,h2,toplam,h,x,frx,sonuc;
    int arrayderece[500],n,i,c,q,derece;
    double tpl=0,xsayi;

    double fonkhesap(double x){
        toplam=0;

        for(i=0;i<derece;i++){
            frx=x;
            for(c=1;c<derece-i;c++){
                frx*=x;
            }
            frx*=arraykat[i];
            toplam +=frx;
        }
        toplam+=sabit;
        return toplam;
    }

    printf("-----SIMPSON 3/8-----\n");
    printf("\n");
    printf("Denklem kacinci dereceden: ");
    scanf("%d",&derece);
    printf("\n");
    xsayi=derece;
    for(i=0; i<derece; i++) {
        arrayderece[i]=xsayi;
        xsayi--;
    }
    for(i=0; i<derece; i++) {
        printf("%d. X kat sayi degeri: ",i+1);
        scanf("%lf",&arraykat[i]);
    }
    printf("C sabit degeri: ");
    scanf("%lf",&sabit);

    printf("\nf(x)= ");
    for(i=0;i<derece;i++){
        printf("%.2lfx^%d + ",arraykat[i],arrayderece[i]);
    }
    printf("%.2lf\n\n",sabit);

    printf("N degerini giriniz (sadece 3un katlari): ");
    scanf("%d",&n);
    if(n%3 != 0){
        printf("3un kati bir sayi girmediniz");
        return 0;
    }
}
```

```
printf("Baslangic limit degerini giriniz (a): ");
scanf("%lf",&a);
printf("Son limit degerini giriniz (b): ");
scanf("%lf",&b);
h=fabs(b-a)/n;
for(q=1;q<n;q++){
    x=a+q*h;
    if(q%3==0){
        tpl=tpl+2*fonkhesap(x);
    }
    else{
        tpl=tpl+3*fonkhesap(x);
    }
}
sonuc=(3*h/8)*(fonkhesap(a)+fonkhesap(b)+tpl);
printf("\nSonuc: %lf\n",sonuc);
return 0;
}
```

Trapez Yöntemi

```
#include<stdio.h>
#include<math.h>

int main() {
    double arraykat[500],a,sabit,b,h2,toplam,h,x,frx,sonuc;
    int arrayderece[500],n,i,c,q,derece;
    double tpl=0,xsayi;

    double fonkhesap(double x){
        toplam=0;

        for(i=0;i<derece;i++){
            frx=x;
            for(c=1;c<derece-i;c++){
                frx*=x;
            }
            frx*=arraykat[i];
            toplam +=frx;
        }
        toplam+=sabit;
        return toplam;
    }

    printf("-----TRAPEZ YONTEMI-----\n");
    printf("\n");
    printf("Denklem kacinci dereceden: ");
    scanf("%d",&derece);
    printf("\n");
    xsayi=derece;
    for(i=0; i<derece; i++) {
        arrayderece[i]=xsayi;
        xsayi--;
    }
    for(i=0; i<derece; i++) {
        printf("%d. X kat sayi degeri: ",i+1);
        scanf("%lf",&arraykat[i]);
    }
    printf("C sabit degeri: ");
    scanf("%lf",&sabit);

    printf("\nf(x)= ");
    for(i=0;i<derece;i++){
        printf("%.2lfx^%d + ",arraykat[i],arrayderece[i]);
    }
    printf("%.2lf\n\n",sabit);

    printf("N degerini giriniz: ");
    scanf("%d",&n);
    printf("Baslangic limit degerini giriniz (a): ");
    scanf("%lf",&a);
    printf("Son limit degerini giriniz (b): ");
    scanf("%lf",&b);
```

```
h=fabs (b-a) /n;
for (q=1;q<n;q++) {
    x=a+q*h;
    tpl=tpl+fonskhesap (x) ;

}
sonuc= (h/2) * (fonskhesap (a)+fonskhesap (b) +2*tpl) ;
printf ("\nSonuc: %lf\n", sonuc) ;
return 0;
}
```

Newton Interpolasyon

```
#include <stdio.h>
int main() {
    int datasayi,i,j;
    float xdeger[10],ydeger[15],ortakf,fx,p,k=1,m;
    printf("-----NEWTON INTERPOLASYON-----\n\n");
    printf("Incelenecek olan data miktarini giriniz(max 10): ");
    scanf("%d",&datasayi);
    printf("\n---X DEGERLERINI GIRINIZ---\n\n");
    for(i=0;i<datasayi;i++){
        printf("%d. x degeri: ",i+1);
        scanf("%f",&xdeger[i]);
    }
    printf("\n\n---Y DEGERLERINI GIRINIZ---\n\n");
    for(i=0;i<datasayi;i++){
        printf("%d. Y degeri: ",i+1);
        scanf("%f",&ydeger[i]);
    }
    printf("\n\nBulunmasi istenen f(x) degeri x: ");
    scanf("%f",&fx);
    ortakf=xdeger[1]-xdeger[0];    //x degerlerinin ortak
farkların bulunması
    p=(fx-xdeger[0])/ortakf;    // p formulu: p=((x-x0)/h))
    m=ydeger[0]; // y'nin ilk değeri atanır
    for(i=0;i<(datasayi-1);i++){
        for(j=0;j<(datasayi-i-1);j++){
            ydeger[j]=ydeger[j+1]-ydeger[j];
        }
        k=k*(p-(i+1)+1)/(i+1);
        m=m+k*ydeger[0];
    }
    printf("\n\nf(%f) = %f",fx,m);
    return 0;
}
```