Sayısal Analiz BLM1022 Gr-1

Proje Ödevi Raporu

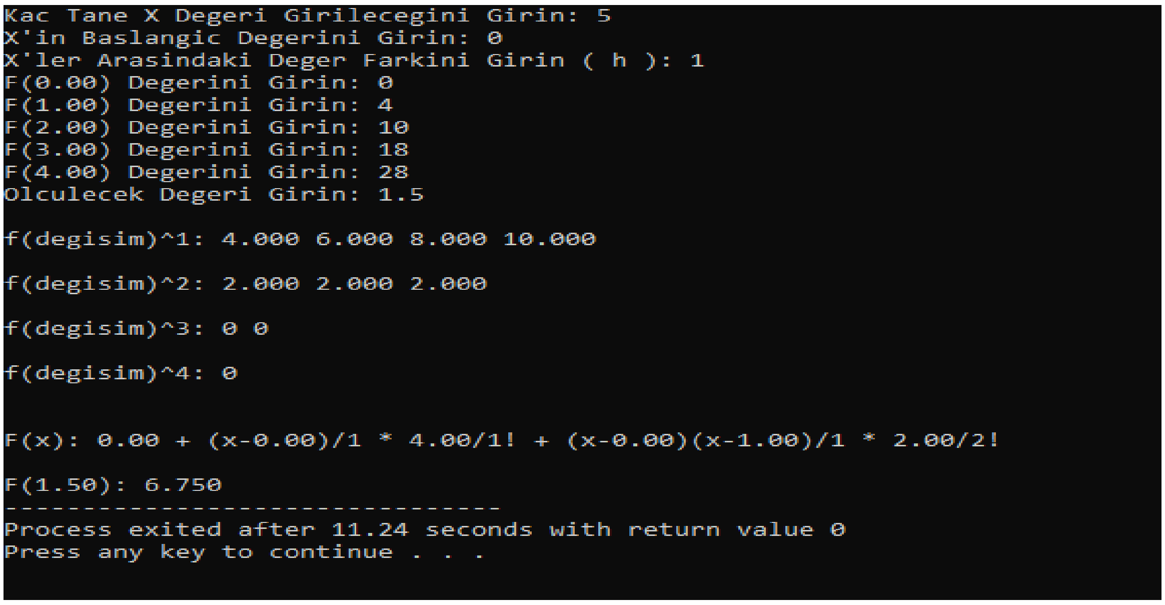
Muhammed Eren Şekkeli-20011019

Yapılan Projeler:

1. Bisection
2. Regula-Falsi
3. Newton-Rapshon
4. NxN'lik bir matrisin tersi
5. Gauss Eleminasyon
6. Gauss Seidal
7. Sayısal Türev (merkezi, ileri ve geri)
8. Simpson yöntemi
9. Trapez yöntemi
10. Değişken dönüşümsüz Gregory Newton Enterpolasyonu

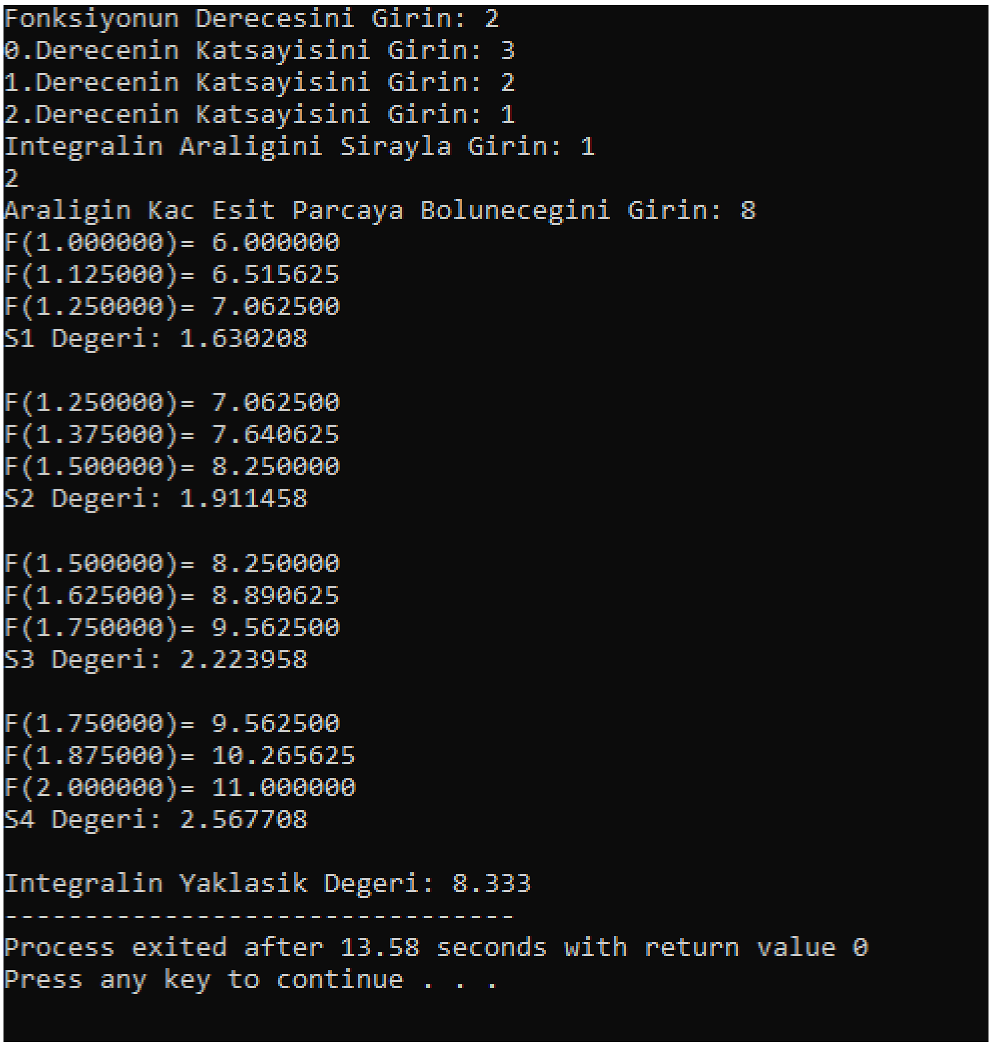
SUNUMU YAPILAN YÖNTEMLERİN EKRAN GÖRÜNTÜSÜ:

1-)Gregory-Newton:



```
Kac Tane X Degeri Girilecegini Girin: 5
X'in Baslangic Degerini Girin: 0
X'ler Arasindaki Deger Farkini Girin ( h ): 1
F(0.00) Degerini Girin: 0
F(1.00) Degerini Girin: 4
F(2.00) Degerini Girin: 10
F(3.00) Degerini Girin: 18
F(4.00) Degerini Girin: 28
Olculecek Degeri Girin: 1.5

f(degisim)^1: 4.000 6.000 8.000 10.000

f(degisim)^2: 2.000 2.000 2.000

f(degisim)^3: 0 0

f(degisim)^4: 0


F(x): 0.00 + (x-0.00)/1 * 4.00/1! + (x-0.00)(x-1.00)/1 * 2.00/2!

F(1.50): 6.750
--------------------------------
Process exited after 11.24 seconds with return value 0
Press any key to continue . . .
```

2-)Simpson 1/3 Kuralı



```
Fonksiyonun Derecesini Girin: 2
0.Derecenin Katsayisini Girin: 3
1.Derecenin Katsayisini Girin: 2
2.Derecenin Katsayisini Girin: 1
Integralin Araligini Sirayla Girin: 1
2
Araligin Kac Esit Parcaya Bolunecegini Girin: 8
F(1.000000)= 6.000000
F(1.125000)= 6.515625
F(1.250000)= 7.062500
S1 Degeri: 1.630208

F(1.250000)= 7.062500
F(1.375000)= 7.640625
F(1.500000)= 8.250000
S2 Degeri: 1.911458

F(1.500000)= 8.250000
F(1.625000)= 8.890625
F(1.750000)= 9.562500
S3 Degeri: 2.223958

F(1.750000)= 9.562500
F(1.875000)= 10.265625
F(2.000000)= 11.000000
S4 Degeri: 2.567708

Integralin Yaklasik Degeri: 8.333
--------------------------------
Process exited after 13.58 seconds with return value 0
Press any key to continue . . .
```

Not: **Sunumda gösterilen yöntemlerin kodları dahil olmak üzere tüm kodlar aşağıdadır.**

Bisection Yöntemi

```
#include <stdio.h>

#include <math.h>

        float a,b,c,d,hata,olchata,ite2=1,f1,f2,f3;

        int ite=0,der[25],n,i,j;


void func(float c){

        f3=0;

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

        d=1;

        for(j=1;j<=i;j++){

                d=d*c;

        }

        d=d*der[i];

        f3=f3 + d;

}

ite++;


}

void funca(float a){

        f1=0;

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

        d=1;

        for(j=1;j<=i;j++){

                d=d*a;

        }

        d=d*der[i];

        f1=f1 + d;

}


}
```

```
void funcb(float b){

            f2=0;

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

            d=1;

            for(j=1;j<=i;j++){

                        d=d*b;

            }

            d=d*der[i];

            f2=f2 + d;

}


}


int main(){


            printf("Fonksiyonun derecesini girin: "); scanf("%d",&n);


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

                        printf("%d. derecenin katsayisini girin: ", i); scanf("%d",&der[i]);


            }


            do{

                        if(f1!=0 || f2!=0){

                        printf("Girdiginiz kontrol degerler uygun degil \n\n");

                        }

            f1=0; f2=0;

printf("Kontrol edilecek ilk degeri girin: "); scanf("%f",&a);

printf("Kontrol edilecek ikinci degeri girin: "); scanf("%f",&b);


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

            d=1;

            for(j=1;j<=i;j++){
                d=d*a;
        }
        d=d*der[i];
        f1=f1 + d;
}

for(i=0;i<=n;i++){
        d=1;
        for(j=1;j<=i;j++){
                d=d*b;
        }
        d=d*der[i];
        f2=f2 + d;
}
}while(f1*f2>=0);



printf("Kabul edilebilecek hata degerini girin: "); scanf("%f",&hata);


olchata=(b-a)/2;


printf("ilk deger araligi: [%f,%f] \n F(%f)= %f \n F(%f)= %f \n\n",a,b,a,f1,b,f2);
while(olchata>hata){


        ite2*=2;
c=(a+b)/2;
func(c);
if(f1*f3<0){
        if(a>c){
                olchata=(a-c)/ite2;
                printf("%d. iterasyon: \n Aranan deger araligi: [%f,%f]  F(%f)= %f\n",ite,b,a,c,f3);
        }else{
                olchata=(c-a)/ite2;
                printf("%d. iterasyon: \n Aranan deger araligi: [%f,%f]  F(%f)= %f\n",ite,a,b,c,f3);

        }
        b=c;
}else if(f2*f3<0){
        if(b>c){
                olchata=(b-c)/ite2;
                        printf("%d. iterasyon: \n Aranan deger araligi: [%f,%f]  F(%f)= %f\n",ite,a,b,c,f3);
        }else{
                olchata=(c-b)/ite2;
                        printf("%d. iterasyon: \n Aranan deger araligi: [%f,%f]  F(%f)= %f\n",ite,b,a,c,f3);
                        }
                        a=c;
        }
}
printf("\n\n");
if(olchata<=hata){
        printf("Yapilan iterasyon sayisi: %d \n",ite);
        printf("Fonksiyonun yaklasik koku: %f \n", c);
        printf("Fonksiyon sonucu: %f \n", f3);
        printf("En son olculen hata degeri: %f \n", olchata*2);


}
        system("pause");
}
```

Regula-Falsi Yöntemi

```
#include <stdio.h>

#include <math.h>

        float a,b,c,d,hata,olchata,ite2=1,f1,f2,f3;

        int ite=0,der[25],n,i,j;


void func(float c){

        f3=0;

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

        d=1;

        for(j=1;j<=i;j++){

                d=d*c;

        }

        d=d*der[i];

        f3=f3 + d;

}

ite++;


}

void funca(float a){

        f1=0;

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

        d=1;

        for(j=1;j<=i;j++){

                d=d*a;

        }

        d=d*der[i];

        f1=f1 + d;

}


}

void funcb(float b){
```

```
            f2=0;
for(i=0;i<=n;i++){
            d=1;
            for(j=1;j<=i;j++){
                        d=d*b;
            }
            d=d*der[i];
            f2=f2 + d;
}

}

int main(){

            printf("Fonksiyonun derecesini girin: "); scanf("%d",&n);

            for(i=0; i<=n;i++){
                        printf("%d. derecenin katsayisini girin: ", i); scanf("%d",&der[i]);

            }

            do{
                        if(f1!=0 || f2!=0){
                        printf("Girdiginiz kontrol degerler uygun degil \n\n");
                        }

            f1=0; f2=0;
printf("Kontrol edilecek ilk degeri girin: "); scanf("%f",&a);
printf("Kontrol edilecek ikinci degeri girin: "); scanf("%f",&b);

for(i=0;i<=n;i++){
            d=1;
            for(j=1;j<=i;j++){
```

```
                        d=d*a;
            }
            d=d*der[i];
            f1=f1 + d;
}

for(i=0;i<=n;i++){
            d=1;
            for(j=1;j<=i;j++){
                        d=d*b;
            }
            d=d*der[i];
            f2=f2 + d;
}
}while(f1*f2>=0);

printf("Kabul edilebilecek hata degerini girin: "); scanf("%f",&hata);

olchata=(b-a)/2;

printf("ilk deger araligi: [%f,%f] \n F(%f)= %f \n F(%f)= %f \n\n",a,b,a,f1,b,f2);
while(olchata>hata){
            ite2*=2;
c=((b*f1)-(a*f2))/(f1-f2);
func(c);
if(f1*f3<0){
            if(a>c){
                        olchata=(a-c)/ite2;
                        printf("%d. iterasyon: \n Aranan deger araligi: [%f,%f]  F(%f)= %f\n",ite,b,a,c,f3);
            }else{
                        olchata=(c-a)/ite2;
                        printf("%d. iterasyon: \n Aranan deger araligi: [%f,%f]  F(%f)= %f\n",ite,a,b,c,f3);
            }
            b=c; funcb(b);
}else if(f2*f3<0){
            if(b>c){
                        olchata=(b-c)/ite2;
                        printf("%d. iterasyon: \n Aranan deger araligi: [%f,%f]  F(%f)= %f\n",ite,a,b,c,f3);
            }else{
                        olchata=(c-b)/ite2;
                        printf("%d. iterasyon: \n Aranan deger araligi: [%f,%f]  F(%f)= %f\n",ite,b,a,c,f3);
                                    }
                                    a=c; funca(a);
            }
}
printf("\n\n");
if(olchata<=hata){
            printf("Yapilan iterasyon sayisi: %d \n",ite);
            printf("Fonksiyonun yaklasik koku: %f \n", c);
            printf("Fonksiyon sonucu: %f \n", f3);
            printf("En son olculen hata degeri: %f \n", olchata*2);


}
            system("pause");
}
```

Newton-Raphson Yöntemi

```
#include <stdio.h>

#include <math.h>

        float a,b,c,d,hata,olchata,ite2=1,f1,f1t,f2t,f2,f3;

        int ite=0,der[25],derturev[25],n,i,j,alt=0,ust=0;


void func(float c){

        f3=0;

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

        d=1;

        for(j=1;j<=i;j++){

                d=d*c;

        }

        d=d*der[i];

        f3=f3 + d;

}


}

void funca(float a){

        f1=0;

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

        d=1;

        for(j=1;j<=i;j++){

                d=d*a;

        }

        d=d*der[i];

        f1=f1 + d;

}


}

void funcatur(float a){

        f1t=0;
```

```
            for(i=0;i<n;i++){
            d=1;
            for(j=1;j<=i;j++){
                        d=d*a;
            }
            d=d*derturev[i];
            f1t=f1t + d;
}
}
void funcb(float b){
            f2=0;
for(i=0;i<=n;i++){
            d=1;
            for(j=1;j<=i;j++){
                        d=d*b;
            }
            d=d*der[i];
            f2=f2 + d;
}


}
void funcbtur(float b){
            f2t=0;
            for(i=0;i<n;i++){
            d=1;
            for(j=1;j<=i;j++){
                        d=d*b;
            }
            d=d*derturev[i];
            f2t=f2t + d;
}
}
```

```
int main(){

        printf("Fonksiyonun derecesini girin: "); scanf("%d",&n);

        for(i=0; i<=n;i++){
                printf("%d. derecenin katsayisini girin: ", i); scanf("%d",&der[i]);

        }

        for(i=1; i<=n;i++){
                derturev[i-1]=der[i]*i;
        }

        f1=0; f2=0; f1t=0;
printf("Kontrol edilecek ilk degeri girin: "); scanf("%f",&a);
printf("Kontrol edilecek ikinci degeri girin: "); scanf("%f",&b);

for(i=0;i<=n;i++){
        d=1;
        for(j=1;j<=i;j++){
                d=d*a;
        }
        d=d*der[i];
        f1=f1 + d;
}

for(i=0;i<=n;i++){
        d=1;
        for(j=1;j<=i;j++){
                d=d*b;
        }
        d=d*der[i];
        f2=f2 + d;
```

```
}



printf("Kabul edilebilecek hata degerini girin: "); scanf("%f",&hata);


olchata=fabs(b-a);
funcatur(a);
c=a-(f1/f1t);
func(c);
if(fabs(f3)<fabs(f1)){
        while(olchata>hata){
                ite++; alt++;
                funca(a); funcatur(a);
        c=a-(f1/f1t);
        olchata=fabs(c-a);
        a=c; funca(a);
                printf("%d. iterasyon: \n Yeni deger : %f  F(%f)= %f\n",ite,c,c,f1);
        }
}else {
        while(olchata>hata){
        ite++; ust++;
         funcbtur(b);
        c=b-(f2/f2t);
        olchata=fabs(c-b);
        b=c; funcb(b);

                        printf("%d. iterasyon: \n Yeni deger : %f  F(%f)= %f\n",ite,c,c,f2);
                }
        }



printf("\n\n");
if(olchata<=hata){
        printf("Yapilan iterasyon sayisi: %d \n",ite);

        printf("Fonksiyonun yaklasik koku: %f \n", c);

        if(alt!=0){

                printf("Fonksiyon sonucu: %f \n", f1);

        }else{

                printf("Fonksiyon sonucu: %f \n", f2);

        }

        printf("En son olculen hata degeri: %f \n", olchata);


}

        system("pause");

}
```

NxN’lik Matrisin Tersi (Gaus-Jordan)

```
#include<stdio.h>
#define max_length 100

int main(){
                float matrix[max_length][max_length],c,tmp;
                int i,j,z,k,n,hata;

                printf("Matrisin Degerini Girin: ");  scanf("%d", &n);

                printf("Matrisin Satir ve Sutun Degerlerini Girin: \n");
                for(i=1;i<=n;i++){
                        for(j=1;j<=n;j++){
                                printf("matrix[%d][%d] = ",i,j);
                                scanf("%f", &matrix[i][j]);
                        }
                }

                for(i=1;i<=n;i++){
                        for(j=1;j<=n;j++){

                                if(i==j){
                                      matrix[i][j+n] = 1;
                                }else{
                                      matrix[i][j+n] = 0;
                                }
                        }
                }

                for(i=1;i<=n;i++){
                        if(matrix[i][i] == 0.0&&i+1<=n){
                                for(z=1;z<=2*n;z++){
```

```
                    tmp=matrix[i][z];
                            matrix[i][z]=matrix[i+1][z];
                            matrix[i+1][z]=tmp;
                    }
          }else if(matrix[i][i] == 0.0&&i+1>n){
                  for(z=1;z<=2*n;z++){
                    tmp=matrix[i][z];
                            matrix[i][z]=matrix[i-1][z];
                            matrix[i-1][z]=tmp;
                    }
          }
          for(j=1;j<=n;j++){
                    if(i!=j){
                              c = matrix[j][i]/matrix[i][i];
                              for(k=1;k<=2*n;k++){

                                     matrix[j][k] = matrix[j][k] - c*matrix[i][k];


                    }
          }
     }
}
     for(i=1;i<=n;i++){
          for(j=n+1;j<=2*n;j++){

                  matrix[i][j] = matrix[i][j]/matrix[i][i];
          }
     }

     printf("\nMatrisin Tersi: \n");
     for(i=1;i<=n;i++){
          for(j=n+1;j<=2*n;j++){
                  printf("%0.3f\t",matrix[i][j]);
```

```
                    }
                    printf("\n");
                }

                system("pause");
}
```

Gauss Eliminasyon Yöntemi

```
#include <stdio.h>
#define max_length 100

int main(){

        int n,i,j,k,z;
        float matrix[max_length][max_length],x[max_length],tmp,c;

        do{
                printf("Denklem Sisteminin Bilinmeyen Sayisini Girin: "); scanf("%d",&n);
        }while(n>max_length);

        for(i=1;i<=n;i++){
                for(j=1;j<=n+1;j++){
                        if(j<=n){
                        printf("%d.Denkleminin x%d kokunun kat sayisi: ",i,j); scanf("%f",&matrix[i][j]);
                }else{
                        printf("%d.Denklemin Sonucu: ",i); scanf("%f",&matrix[i][j]);
                        }
                }
        }

         for(i=1; i<=n; i++) {
                if(matrix[i][i] == 0.0&&i+1<=n){
                                 for(z=1;z<=2*n;z++){
                                 tmp=matrix[i][z];
                                        matrix[i][z]=matrix[i+1][z];
                                        matrix[i+1][z]=tmp;
                                 }
                           }else if(matrix[i][i] == 0.0&&i+1>n){
                                for(z=1;z<=2*n;z++){
```

```
                                                tmp=matrix[i][z];
                                                      matrix[i][z]=matrix[i-1][z];
                                                      matrix[i-1][z]=tmp;
                                                }
                                        }
      for(j=1; j<=n; j++){
        if(j>i){
          c=matrix[j][i]/matrix[i][i];
          for(k=1; k<=n+1; k++){
            matrix[j][k]=matrix[j][k]-c*matrix[i][k];
          }
          }
      }
    }
    x[n]=matrix[n][n+1]/matrix[n][n];
     for(i=n-1; i>=1; i--){
      tmp=0;
      for(j=i+1; j<=n; j++){
        tmp=tmp+matrix[i][j]*x[j];
      }
      x[i]=(matrix[i][n+1]-tmp)/matrix[i][i];
    }

    printf("\nKoklerin Degeri: \n");
    for(i=1;i<=n;i++){
      printf("\nx%d=%0.2f\t",i,x[i]);
    }
          return 0;
  }
```

Gauss Seidal Yöntemi

```
#include <stdio.h>

#include <math.h>

#define max_length 100

 int main(){


        int n,i,j,k,l,finish,ite=1;

        float
matrix[max_length][max_length],x[max_length],dx[max_length],tmp,hata,start[max_length],err,max=1,max
2=1;


        do{

                printf("Denklemin Bilinmeyen Sayisini Girin: "); scanf("%d",&n);

        }while(n>max_length);


        for(i=1;i<=n;i++){

                for(j=1;j<=n+1;j++){

                        if(j<=n){

                        printf("%d.Denkleminin x%d. kokunun kat sayisi: ",i,j); scanf("%f",&matrix[i][j]);

                }else{

                        printf("%d.Denklemin Sonucu: ",i); scanf("%f",&matrix[i][j]);

                        }

                }

        }

        for(i=1;i<=n;i++){

                printf("x%d Kokunun Baslangic Degerini Girin: ",i); scanf("%f",&start[i]);

        }


        printf("Kabul Edilebilecek Hata Degerini Girin: "); scanf("%f",&hata);


        for(i=1;i<=n;i++){

                max=max*matrix[i][i];

        }
```

```
max=fabs(max);

for(i=1;i<=n;i++){
    for(j=1;j<=n+1;j++){
        if(i!=j){
        for(k=1;k<=n;k++){
            if(k!=i && k!=j){
                for(l=1;l<=n+1;l++){
                tmp=matrix[j][l];
                matrix[j][l]=matrix[k][l];
                matrix[k][l]=tmp;

                }
                for(l=1;l<=n;l++){
                    max2=max2*matrix[l][l];
                    }
                        max2=fabs(max2);
                        if(max2>max){
                            max=max2; max2=1;
                        }else{
                for(l=1;l<=n+1;l++){
                tmp=matrix[k][l];
                matrix[k][l]=matrix[j][l];
                matrix[j][l]=tmp; max2=1;

                            }
                        }
                    }

            }
        }
    }
}
```

```
    for(i=1;i<=n;i++){
        x[i]=start[i];
    }
    while(finish!=n){
        if(ite==1){
            printf("Iterasyon: %d\n",ite);
            for(i=1;i<=n;i++){
                printf("\n x%d: %f  Degisim: -\n",i,x[i]);
            }
        }
        ite++; printf("Iterasyon: %d\n",ite);
    for(i=1;i<=n;i++){
    tmp=0;  err=x[i];
        for(j=1;j<=n;j++){
            if(i!=j){
            tmp=tmp+(matrix[i][j]*x[j]);
            }
        }
        x[i]=matrix[i][n+1]-tmp;
        x[i]=x[i]/matrix[i][i];
        dx[i]=err-x[i]; printf("\n x%d: %f  Degisim: %f\n",i,x[i],fabs(dx[i]));
    }
    i=1; finish=0;
    while(fabs(dx[i])<=hata &&i<=n){
        finish++; i++;
    }
}
    if(finish==n){
        printf("\n\n");
        printf("\n Yapilan Toplam iterasyon sayisi: %d\n",ite);
        for(i=1;i<=n;i++){
            printf("\nx%d Degeri: %0.2f\t\n",i,x[i]);
        }
        }


        system("pause");
}
```

Sayısal Türev (merkezi, ileri, geri)

```
#include <stdio.h>

#define max_length 100


        float value,f1,h,f2,result;

        int der[max_length],derturev[max_length],n,i,j;


        void geri(float value,float h){

                float d,val1=value;


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

                d=1;

                for(j=1;j<=i;j++){

                d=d*val1;

                        }

        d=d*der[i];

        f1=f1 + d;

                }


                val1=val1-h;


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

                d=1;

                for(j=1;j<=i;j++){

                d=d*val1;

                        }

        d=d*der[i];

        f2=f2 + d;

                }


        result=(f1-f2)/h;

        }
```

```
void ileri(float value,float h){
            float d,val2=value;


            for(i=0;i<=n;i++){
            d=1;
            for(j=1;j<=i;j++){
            d=d*val2;
                        }
d=d*der[i];
f1=f1 + d;
            }


            val2=val2+h;


            for(i=0;i<=n;i++){
            d=1;
            for(j=1;j<=i;j++){
            d=d*val2;
                        }
d=d*der[i];
f2=f2 + d;
            }


result=(f2-f1)/h;
}
void merkez(float value,float h){
            float d,val3=value;
            val3=val3+h;
            for(i=0;i<=n;i++){
            d=1;
            for(j=1;j<=i;j++){
            d=d*val3;
                        }
            d=d*der[i];
            f1=f1 + d;
                        }

                        val3=val3-(2*h);

                        for(i=0;i<=n;i++){
                        d=1;
                        for(j=1;j<=i;j++){
                        d=d*val3;
                                    }
            d=d*der[i];
            f2=f2 + d;
                        }

            result=(f1-f2)/(2*h);
            }

int main(){

            printf("Fonksiyonun derecesini girin: "); scanf("%d",&n);

            for(i=0; i<=n;i++){
                        printf("%d. derecenin katsayisini girin: ", i); scanf("%d",&der[i]);

            }
            printf("Hangi Degerin Turevi Alinsin: "); scanf("%f",&value);

            printf("Degisim Degerini Girin ( h ): "); scanf("%f",&h);

geri(value,h);
printf("\nGeri Fark Ile Turevin Sonucu: %0.3f\t\n",result);
result=0; f1=0; f2=0; ileri(value,h);
```

```
printf("Ileri Fark Ile Turevin Sonucu: %0.3f\t\n",result);

result=0; f1=0; f2=0; merkez(value,h);

printf("Merkezi Fark Ile Turevin Sonucu: %0.3f\t\n",result);


        system("pause");

}
```

Simpson Yöntemi

```
#include <stdio.h>

#include <math.h>

#define max_length 100


int der[max_length],n,test=0,test2=1;

float Y[max_length],S[max_length],value;


void func(float h){


        int i,j;

        float f1,f2,f3,d;


        Y[test]=0;

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

        d=1;

        for(j=1;j<=i;j++){

                d=d*value;

                }

        d=d*der[i];

        Y[test]=Y[test] + d;

        }

                f1=Y[test]; printf("F(%f)= %f\n",value,f1);

        test++; value=value+h;


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

        d=1;

        for(j=1;j<=i;j++){

                d=d*value;

                }

        d=d*der[i];

        Y[test]=Y[test] + d;
        }
                f2=Y[test]; printf("F(%f)= %f\n",value,f2);
        test++; value=value+h;
        for(i=0;i<=n;i++){
        d=1;
        for(j=1;j<=i;j++){
                d=d*value;
                }
        d=d*der[i];
        Y[test]=Y[test] + d;

        }
                f3=Y[test];  printf("F(%f)= %f\n",value,f3);
        S[test2]=(f1+4*f2+f3)*(h/3);
                printf("S%d Degeri: %f\n\n",test2,S[test2]);
                test2++;
}

int main(){
        float toplam,h,a,b;
        int i,j,k,bol;

        printf("Fonksiyonun Derecesini Girin: "); scanf("%d",&n);



for(i=0;i<=n;i++){
        printf("%d.Derecenin Katsayisini Girin: ",i); scanf("%d",&der[i]);
}

printf("Integralin Araligini Sirayla Girin: "); scanf("%f %f",&a,&b);

do{
        if(bol%2==1){
                printf("Aralik Degeri Cift Sayi Olmali!\n");
        }
        printf("Araligin Kac Esit Parcaya Bolunecegini Girin: "); scanf("%d",&bol);
}while(bol%2==1);

h=(b-a)/bol;
value=a;

while(test2<=bol/2){
        func(h);
}
for(i=1;i<=bol;i++){
        S[i]=fabs(S[i]);
        toplam=toplam+S[i];
}

printf("Integralin Yaklasik Degeri: %0.3f\t",toplam);

return 0;
}
```

Trapez Yöntemi

```
#include <stdio.h>
#define max_length 100

int der[max_length],n,test=0;
float Y[max_length],S[max_length];

void func(float value,float h){

        int i,j;
        float f1,f2,value2,d;
        value2=value; Y[test]=0;
        for(i=0;i<=n;i++){
        d=1;
        for(j=1;j<=i;j++){
                d=d*value2;
                }
        d=d*der[i];
        Y[test]=Y[test] + d;
        }
                f1=Y[test]; printf("F(%f)= %f\n",value2,f1);
        test++; value2=value+h;

        for(i=0;i<=n;i++){
        d=1;
        for(j=1;j<=i;j++){
                d=d*value2;
                }
        d=d*der[i];
        Y[test]=Y[test] + d;

        }
```

```
                    f2=Y[test]; printf("F(%f)= %f\n",value2,f2);
          S[test]=((f1+f2)/2)*h;
                    printf("S%d Alan Degeri: %f\n\n",test,S[test]);
}


int main(){
          float bol,toplam,value,h,a,b;
          int i,j,k;

          printf("Fonksiyonun Derecesini Girin: "); scanf("%d",&n);



for(i=0;i<=n;i++){
          printf("%d.Derecenin Katsayisini Girin: ",i); scanf("%d",&der[i]);
}

printf("Integralin Araligini Sirayla Girin: "); scanf("%f %f",&a,&b);

printf("Araligin Kac Esit Parcaya Bolunecegini Girin: "); scanf("%f",&bol);

h=(b-a)/bol;
value=a;

while(value<b){
          func(value,h);
          value=value+h;
}
for(i=1;i<=bol;i++){

          toplam=toplam+S[i];
}

printf("\nIntegralin Yaklasik Degeri: %f\n",toplam);
```

```
system("pause");

}
```

Gregory Newton Enterpolasyonu

```
#include <stdio.h>
#define max_length 100

int us(int h,int j){
        int i=0;
        if(i==j){
                return 1;
        }else{
                return (us(h,--j)*h);
        }
}

int main(){

        int i,j,n,k,n2,test,fac;
        float h,X[max_length],F[max_length],ileri_F[max_length][max_length],value,d,f1=0;

        do{
                printf("Kac Tane X Degeri Girilecegini Girin: "); scanf("%d",&n);
        }while(n>max_length);

        printf("X'in Baslangic Degerini Girin: "); scanf("%f",&X[0]);

        printf("X'ler Arasindaki Deger Farkini Girin ( h ): "); scanf("%f",&h);

        for(i=1;i<=n;i++){
                X[i]=X[i-1]+h;
        }
        for(i=0;i<n;i++){
                printf("F(%0.2f) Degerini Girin: ",X[i]); scanf("%f",&F[i]);
        }
```

```
printf("Olculecek Degeri Girin: "); scanf("%f",&value);


for(i=1;i<=n;i++){
            ileri_F[i][0]=F[i-1];
}


 k=1; n2=n-1;
while(n2>0){
test=0;
if(k==1){
            for(i=1;i<=n-1;i++){
            ileri_F[i][k]=F[i]-F[i-1];


            }
}else{
            n2--;
            for(i=1;i<=n-1;i++){
            ileri_F[i][k]=ileri_F[i+1][k-1]-ileri_F[i][k-1];


            }
}


            for(i=1;i<=n-1;i++){
                        if(ileri_F[i][k]==0){
                                    test++;
                        }
            }
            k++;
}
            n2=n-1;
for(j=1;j<=n-1;j++){
            printf("\nf(degisim)^%d: ",j);
```

```
                for(i=1;i<=n2;i++){

                        if(ileri_F[i][j]>0)

                        printf("%0.2f ",ileri_F[i][j]);

                        else if(ileri_F[i][j]<0)

                        printf("%0.2f ",ileri_F[i][j]);

                        else

                        printf("0 ");

                }

                printf("\n");

                n2--;

        }


        printf("\n\nF(x): ");

        for(i=0;i<=n-1;i++){

        d=1; fac=1; if(i!=0 && ileri_F[1][i]!=0)printf(" + ");

        for(j=1;j<=i;j++){

                d=d*(value-X[j-1]); fac=fac*j; if(ileri_F[1][i]!=0) printf("(x-%0.2f)",X[j-1]);

        }

        if(i==0){

                f1=f1+F[i]; printf("%0.2f",F[i]);

        }else{

                d=d*ileri_F[1][i]/(fac*us(h,j-1));

                        f1=f1 + d;  if(ileri_F[1][i]!=0)  printf("/%d * %0.2f/%d!",us(h,j-1),ileri_F[1][i],fac);

        }


}


printf("\n\nF(%0.2f): %0.3f",value,f1);


        return 0;

}
```