

# SAYISAL ANALİZ DÖNEM PROJESİ

**DERSİN HOCASI : PROF. DR. BANU DİRİ**

**HAZIRLAYAN : ACHMET VELİ AMET**

**ÖĞRENCİ NO : 19011912**

**PROJEDE KODLADIĞIM ALGORİTMALAR :**

- 1. BİSECTION.**
- 2. REGULA FALSİ**
- 3. NEMTON RAPSHON**
- 4.  $N \times N$  LİK MATRİSİN TERSİNİN ALINMASI**
- 5. GAUSS ELEMINASYON**
- 6. GAUSS SEİDAL**
- 7. SAYISAL TÜREV ( MERKEZİ, İLERİ VE GERİ )**
- 8. SİMPSON YÖNTEMİ**
- 9. TRAPEZ YÖNTEMİ**
- 10. GREGORY NEWTON ENTERPOLASYONU**

**Proje kapsamında istenen algoritmaların TAMAMI C dilinde kodlanmıştır.**

**Proje sunumu GREGORY NEWTON ENTERPOLASYONU ve . SİMPSON YÖNTEMİ ile gerçekleştirilmiştir. ( KODLARDAKİ HATALAR DÜZELTİLİP RAPORA EKLENMİŞTİR)**

# SIMPSON YÖNTEMİ

## Programın C kodu:

```
#include <stdio.h>

#include <math.h>

#define max 20

float F( float x,float matris[2][max], int n );

int main(){

    float N;

    float A,B,D;

    float matris[2][max];

    float H;

    float sum=0.0,sum2 = 0.0;

    int i,n;

    float integral;

    float fonk_a,fonk_b;

    printf("\n giriceginiz fonksiyonun terim sayisini giriniz : ");

    scanf(" %d",&n);

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

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

        printf("\nfonksiyonun %d. teriminin katsayisini gir : ",i+1);

        scanf(" %f",&matris[0][i]);

        printf("\nfonksiyonun %d. teriminin derecesini gir : ",i+1);

        scanf(" %f",&matris[1][i]);
```

```

}

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

printf("\n\tgirmis oldugunuz fonksiyon : ");

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

    printf(" %.0fX^%.0f ",matris[0][i],matris[1][i]);

}

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

printf("\n\n\n");


printf("\n\ndeger araligini girin : \n");

scanf(" %f %f",&A,&B);

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

printf("\n N  degerini girin (N degeri ciftsayi olmalidir) : ");

scanf(" %f",&N);

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


fonk_a=F(A,matris,n);

fonk_b=F(B,matris,n);



$$H = (B-A)/N ;$$


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

printf("\n\nh degeri = %.4f olarak hesaplandi.",H);

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

printf("\n\n\n");


    for(i=1; i<N; i++)

{

    if(i%2==1)

    {

```

```

        D=A;

        D= D+i*H;

        sum = sum + F(D,matris,n) ;

    }

    else if(i%2==0)
    {

        D=A;

        D= D+i*H;


        sum2 = sum2 + F(D,matris,n);

    }

}

integral = (H/3.0)*( (fonk_a + fonk_b)+      (4*sum) + (2*sum2) ) ;


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

    printf(" %.0fX^%.0f ",matris[0][i],matris[1][i]);

}

printf(" fonksiyonunu ");


printf("integral degeri %.4f olarak hesaplandi.",integral);

return 0;


}

```

```
float F( float x,float matrix[2][max],int n ){  
    float fonk =0.000000;  
  
    int i;  
    for(i=0;i<n;i++){  
  
        fonk = fonk + (matrix[0][i]*pow(x,matrix[1][i]));  
    }  
  
    return fonk;  
}
```

## PROGRAMIN ÇIKTISI :

```
C:\Users\monster pc\Desktop\say2sal analiz proje\simpson_y+ntemi.exe

giriceginiz fonksiyonun terim sayisini giriniz : 3
-----

fonksiyonun 1. teriminin katsayisini gir : 1
fonksiyonun 1. teriminin derecesini gir : 2
fonksiyonun 2. teriminin katsayisini gir : 2
fonksiyonun 2. teriminin derecesini gir : 1
fonksiyonun 3. teriminin katsayisini gir : 3
fonksiyonun 3. teriminin derecesini gir : 0
-----

      girmis oldugunuz fonksiyon :   1X^2   2X^1   3X^0
-----

deger araligini girin :
1
2
-----

N degerini girin (N degeri ciftsayi olmalidir) : 8
-----

-----

h degeri = 0.1250 olarak hesaplandi.
-----

1X^2   2X^1   3X^0   fonksiyonunu integral degeri 8.3333 olarak hesaplandi.
-----
```

# GREGORY NEWTON INTERPOLASYONU

## Programın C kodu:

```
#include <stdio.h>

#include <math.h>

# define max 50

float xfaktoriyel(float x,int y,float matris[max][2* max]);

int faktoriyel(int y);

float F(float x,int derece,float matris[max][2*max],float h);

float mut(float x);

int main (){

    int i,j,k=1,n;

    float h ,x;

    float matris[max][2*max]={0.000000};

    int flag=0,sutun,derece;

    float hata = 0.1;

    printf("kac adet deger gireceginizi giriniz : ");

    scanf("%d",&n);

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

    printf("\nX degerlerini ve onalara karsilik gelen Y degerlerini giriniz : \n");
```

```

for(i=0;i<n;i++)
{
    printf("\nX%d, Y%d : \n",i,i);
    scanf(" %f %f",&matris[i][0],&matris[i][1]);
}
h= matris[1][0]-matris[0][0];
    printf("\n-----\n");
printf("\nx:\t\tf(x)\t\t1f(x)\t\t2f(c)\t\t3f(x)...\n");
j=1;
do{
    flag=0;
    j++;
    for(i=0;i<n-k;i++)
    {
        matris[i][j] = matris[i+1][j-1] - matris[i][j-1];
        if(i>0 && mut(matris[i][j] - matris[i-1][j]) < hata)
        {
            flag++;
        }
    }
    k++;
    if(flag == (n-(j))){
        sutun=j;
        derece=j-1;
    }

}while(flag != (n-(j))) ;

```



```

for(i=0;i<n;i++){
    for(j=0;j<sutun+1;j++){
        printf(" %f \t",matris[i][j]);
    }
    printf("\n");
}
printf("\nderece:%d",derece);
x=1;
while(x!=0.1){
    printf("\n-----\n");
    printf("\n hesaplamak istediginiz yeni degeri giriniz :");
    scanf("%f",&x);
    if(x==0.00001){
        return 0 ;
    }

    printf("\n %f degeri %f olarak hesaplandi ..",x,F(x,derece,matris,h));
}

return 0;
}

```

```

float F(float x,int derece,float matris[max][2*max],float h){

    int i;

    float toplam=0;

```

```

        if(derece>1){
            for(i=2;i<derece+1;i++){
                toplam =toplam +
                ((xfaktoriyel(x,i,matris)/(faktoriyel(i)*pow(h,i)))*matris[0][i+1] );
            }
        }
        toplam= matris[0][1] + (x-matris[0][0])*matris[0][2]/h + toplam ;
        return toplam;
    }
    int faktoriyel(int y){
        int i;
        int fakt=1;
        if(y== 1 || y==0){
            return fakt;
        }
        for(i=2;i<y+1;i++){
            fakt=fakt*i;
        }
        return fakt;
    }
    float xfaktoriyel(float x,int y,float matris[max][2* max]){
        int i;
        float fakt=x-matris[0][0];
        for(i=1;i<y;i++){
            fakt=fakt*(x-matris[i][0]);
        }
        return fakt;
    }
    float mut(float x){
        if(x<0){

```

```
        x=x*(-1);  
    }  
    return x;  
}
```

## PROGRAMIN ÇIKTISI :

```
C:\Users\monster pc\Desktop\say2sal analiz proje\enterpolasyon.exe
kac adet deger gireceginizi giriniz : 5

-----
X degerlerini ve onalara karsilik gelen Y degerlerini giriniz :

X0, Y0 :
0
0

X1, Y1 :
1
4

X2, Y2 :
2
10

X3, Y3 :
3
18

X4, Y4 :
4
28

-----
x:          f(x)          d1f(x)          d2f(c)          d3f(x)...
0.000000    0.000000    4.000000    2.000000
1.000000    4.000000    6.000000    2.000000
2.000000    10.000000   8.000000    2.000000
3.000000    18.000000   10.000000   0.000000
4.000000    28.000000   0.000000    0.000000

derece:2

-----
hesaplamak istediginiz yeni degeri giriniz :1.5

1.500000 degeri 6.750000 olarak hesaplandi ..

-----
hesaplamak istediginiz yeni degeri giriniz :
```