

-> Sunduklarım; Newton Raphson ve Trapez, ss'leri kodlarının altında

9/10 yapıldı, gregory newton yetişmedi.

## 1- BISECTION

## 2- REGULA FALSI

## 3- NEWTON RAPHSON

```
#include <stdio.h>
#include <math.h>
#define SIZE 100

float hesaplaTurev(float, float *, int);
float hesaplaDenklem(float, float *, int);

int main(){
    float x0, x1, epsilon, a[SIZE], turev, denk, fark;
    int i,n;
    float *ptr;
    printf("NEWTON-RAPHSON YONTEMI\n");

    printf("Lutfen gireceginiz denklemin derecesini belirtiniz:");
    scanf("%d", &n);

    for(i=0; i<=n; i++){
        printf("Lutfen %d.terimin katsayisini giriniz:",i);
        scanf("%f", &a[i]);
    }

    printf("Lutfen baslangic degerini giriniz:");
    scanf("%f", &x0);
    printf("Lutfen hassasiyeti giriniz:");
    scanf("%f", &epsilon);
    printf("\n");
    ptr=a;

    turev=hesaplaTurev(x0, ptr, n);
    denk=hesaplaDenklem(x0, ptr, n);

    x1 = x0-(denk/turev);
    fark = fabs(x0-x1);
    printf("x(k)          x(k+1)\n");
    printf("%f          %f\n", x0, x1);

    while(fark>epsilon){
        x0=x1;
        turev=hesaplaTurev(x0, ptr, n);
        denk=hesaplaDenklem(x0, ptr, n);
        x1=x0-(denk/turev); // Newton-Raphson metodu kısmı
        printf("%f          %f\n", x0, x1);
        fark=fabs(x0-x1);
    }
    printf("\n");
    printf("Sonuc yaklasik olarak : %.3f", x1);
}

// Türevin alındığı fonksiyon
float hesaplaTurev(float x0, float *p, int n){
    int i;
    float h=0.1;
```

```

float toplam1=0, toplam2=0, temp;
float x=x0+h, y=x0-h, sonuc;
for(i=0; i<=n; i++){
    temp=p[i]*pow(x,i);
    toplam1+=temp;
}
temp=0;
for(i=0; i<=n; i++){
    temp=p[i]*pow(y,i);
    toplam2+=temp;
}
sonuc=(toplam1-toplam2)/(2*h);
return sonuc;
}
// Denklemin hesaplandığı fonksiyon
float hesaplaDenklem(float x0, float *p, int n){
    int i;
    float temp, sum=0;
    for(i=0; i<=n; i++){
        temp=p[i]*pow(x0,i);
        sum+=temp;
    }
    return sum;
}

```

```

C:\Users\monster\Desktop\Say2sal Analiz\Proje\Nonlinear-OpenMethods\NewtonRaphson.exe
NEWTON-RAPHSON YONTEMI
Lutfen gireceginiz denklemin derecesini belirtiniz:2
Lutfen 0.terimin katsayisini giriniz:-15
Lutfen 1.terimin katsayisini giriniz:2
Lutfen 2.terimin katsayisini giriniz:1
Lutfen baslangic degerini giriniz:0
Lutfen hassasiyeti giriniz:0.001

x(k)          x(k+1)
0.000000      7.500007
7.500007      4.191175
4.191175      3.136663
3.136663      3.002257
3.002257      3.000001
3.000001      3.000000

Sonuc yaklasik olarak : 3.000
-----
Process exited after 9.373 seconds with return value 29
Press any key to continue . . .

```

4- NxN Matris tersi ( Gauss Jordan )

5- Gauss Eliminasyon

6- Gauss Seidal

7- Sayısal Türev

8- SIMPSON Yöntemi

9- TRAPEZ

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#define SIZE 100

float integralhesapla(float, float, float *, float, int);
float integralhesapla2(float, float, float *, float, int);

int main(){

    int i,x;
    float bas, bitis, *ar, n, integral;
    printf("TRAPEZ YONTEMIYLE INTEGRALDE ALAN\n");
    printf("Lutfen fonksiyonunuzun derecesini giriniz:");
    scanf("%d", &x);

    ar=(float*)malloc(sizeof(int*)*x);

    for(i=0; i<=x; i++){
        printf("%d. terimin katsayisini giriniz:", i);
        scanf("%f", &ar[i]);
    }

    printf("Lutfen integralin baslangic degerini giriniz:");
    scanf("%f", &bas);
    printf("Lutfen integralin bitis degerini giriniz:");
    scanf("%f", &bitis);
    printf("Integrali hesaplamak icin alan kac dikdortgene ayrilsin? ");
    scanf("%f", &n);
    if(bas<0 && bitis >0){
        integral=integralhesapla2(bas, bitis, ar, n, x);
        printf("Sonucunuz: %.3f", integral);
    }
    else{
        integral=integralhesapla(bas,bitis,ar,n,x);
        printf("Sonucunuz: %.3f", integral);
    }
}

// INTEGRALIN HESAPLANDIGI YER
float integralhesapla(float bas, float bitis, float *ar, float n, int x){
    int j, k=0;
    float i, dizi[SIZE], sonuc, temp, toplam=0, esassonuc;
    float h=(bitis-bas)/n;
    for(i=bas; i<=bitis; i=i+h){
        temp=0;
        sonuc=0;
```

```

        for(j=0;j<=x;j++){
            temp=ar[j]*pow(i,j);
            sonuc+=temp;
        }
        dizi[k]=sonuc;
        k++;
    }
    k--;
    for(j=1;j<k;j++){
        toplam+=dizi[j];
    }
    esassonuc=h*((dizi[0]+dizi[k])/2 + toplam);
    return esassonuc;
}

```

// PARCALI ALANLAR ICIN INTEGRAL HESABI//

```

float integralhesapla2(float bas, float bitis, float *ar, float n, int x){
    int j,k=0,t=0;
    float i, dizi1[SIZE], dizi2[SIZE],sonuc,temp,toplam1=0,toplam2=0,esassonuc;
    float T1,T2;
    float h1=(0-bas)/n, h2=(bitis-0)/n;

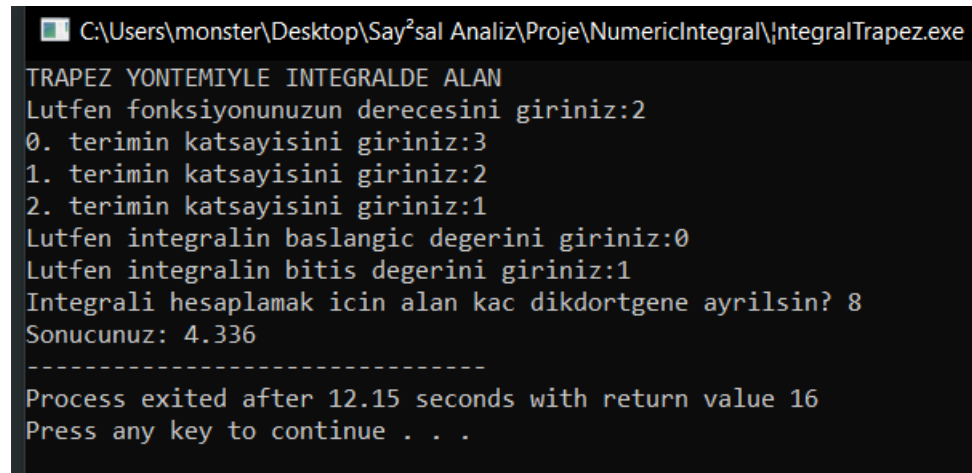
    // bas degerinden 0'a kadar olan kisim //
    for(i=bas;i<=0;i=i+h1){
        temp=0;
        sonuc=0;
        for(j=0;j<=x;j++){
            temp=ar[j]*pow(i,j);
            sonuc+=temp;
        }
        if(x%2==1){
            sonuc*=-1;
        }
        dizi1[k]=sonuc;
        k++;
    }
    k--;

    // 0'dan bitis degerine kadar olan kisim //
    for(i=0;i<=bitis;i=i+h2){
        temp=0;
        sonuc=0;
        for(j=0;j<=x;j++){
            temp=ar[j]*pow(i,j);
            sonuc+=temp;
        }
        dizi2[t]=sonuc;
        t++;
    }
    t--;

    for(j=1;j<k;j++){
        toplam1+=dizi1[j];
    }
    for(j=1;j<t;j++){
        toplam2+=dizi2[j];
    }
    // TRAPEZ KISMI //
    T1=h1*((dizi1[0]+dizi1[k])/2 + toplam1);
    T2=h2*((dizi2[0]+dizi2[k])/2 + toplam2);
    printf("%f\n",T1);
    printf("%f\n",T2);
    esassonuc=T1+T2;
}

```

```
// SONUC //  
return esassonuc;  
}
```



```
C:\Users\monster\Desktop\Say^2sal Analiz\Proje\NumericIntegral\integralTrapez.exe  
TRAPEZ YONTEMIYLE INTEGRALDE ALAN  
Lutfen fonksiyonunuzun derecesini giriniz:2  
0. terimin katsayisini giriniz:3  
1. terimin katsayisini giriniz:2  
2. terimin katsayisini giriniz:1  
Lutfen integralin baslangic degerini giriniz:0  
Lutfen integralin bitis degerini giriniz:1  
Integrali hesaplamak icin alan kac dikdortgene ayrilsin? 8  
Sonucunuz: 4.336  
-----  
Process exited after 12.15 seconds with return value 16  
Press any key to continue . . .
```