



**YTÜ**  
**ELEKTRİK-ELEKTRONİK**  
**FAKÜLTESİ**

Bilgisayar  
Mühendisliği  
Bölümü

# **SAYISAL ANALİZ**

## **Dönem Projesi**



**Dersi Veren Öğretim Üyesi**  
**Prof.Dr. Banu DİRİ**

<b>Numarası</b>	<b>Adı Soyadı</b>
1601113	MUHAMMED ALİ SOYLU

# PROJE KONU BAŞLIKLARI

1. Bisection
2. Regula-Falsi (Sunulan method)
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

Yukarıda yer alan konu başlıklarından **YEŞİL** ile işaretlenen başlıklar yapılabilmiş, **SARI** ile işaretlenen başlıklar ise yapılamamıştır.

## Regula Falsi (Sunulan method)

```
void regulaFalse(){
    float m,n,q;
    int flag=0;
    printf("\n*****REGULA FALSE YONTEMI ILE KOK BULMA*****n");
    fx_oku();
    printf("Lutfen koku icerisine(kiskaca) alan iki degeri giriniz :
");scanf("%f%f",&m,&n);
    printf("Epsilon = ");scanf("%f",&eps);
    q = (fx_hesapla(m)*n-m*fx_hesapla(n))/(fx_hesapla(m)-fx_hesapla(n));
    while(fabs(fx_hesapla(q))>=eps){
        if(fx_hesapla(m)*fx_hesapla(n)<0){
            q = (fx_hesapla(m)*n-m*fx_hesapla(n))/(fx_hesapla(m)-fx_hesapla(n));
            if(fx_hesapla(m)*fx_hesapla(q)>0){
                m = q;
            }
            else{
                n = q;
            }
        }
        else{
            else{

```

```

if(fx_hesapla(m)*fx_hesapla(n)==0){
    if(fx_hesapla(m)==0){
        m = q;
    }
    else{
        n = q;
    }
}
else{
    flag=1;
    printf("\nGirilen degerler koku icerisine almamaktadır.");
    eps=fx_hesapla(q)+1;
}
}
}
if(flag==0){
    printf("\nDenklemin koku : %f -----> 2 decimal point'li hali : %.2f\n",q,q);
}
printf("\nMenuye donmek icin lutfen bir tusa basiniz...\n\n");
getch();
system("cls");
menu();
}

```

```

D:\Dersler\Say\u00b2sal Analiz\2Homework\Proje2020\16011131.exe
*****REGULA FALSE YONTEMI ILE KOK BULMA*****n
f(x) = a*x^5 + b*x^4 + c*x^3 + d*x^2 + e*x^1 + f*x^0
a = 0
b = 0
c = 0
d = 1
e = 2
f = -15

Lutfen koku icerisine(kiskaca) alan iki degeri giriniz : 0
5
Epsilon = 0.001

Denklemin koku : 2.999938 -----> 2 decimal point'li hali : 3.00

Menuye donmek icin lutfen bir tusa basiniz...

```

# PROGRAM KODU

```
//Muhammed Ali SOYLU
```

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

```
void menu();
void fx_oku();float fx_hesapla(float x);float tureval(float x);
void grafik();void bisection();void regulaFalse();void newtonraphson();
void trapezIntegral();void simpsonIntegral();
void matrisInvers();
void gaussElimination();
void numerikTurev();
void gaussSeidal();
void GregoryNewton();
```

```
float xk,xk1,eps,a,b,c,d,e,f,A,B,C,D,E,F,x,sonuc,turev;//global deęişkenler
```

```
main(){
    printf("SAYISAL ANALIZ PROJE\n");
    printf("\nMuhammed Ali SOYLU\n");
    printf("\n16011131\n");
    printf("\nBaslamak icin lutfen herhangi bir tusa basiniz...");
    getch();
    system("cls");
    menu();
    return 0;
}
```

```
void menu(){
    char secim;
    printf("
                                ANA EKRAN
                                \n");
    printf("-----\n");

    printf("1Bisection(YARIYA BOLME)..... (1'ye basiniz) \n");
    printf("2Regula-Falsi..... (2'e basiniz)\n");
    printf("3Newton-Rapshon..... (3'e basiniz) \n");
    printf("4NxN lik bir matrisin tersi.....(4'e basiniz)\n");
    printf("5Gauss Eleminasyonu..... (5'ya basiniz) \n");
    printf("6Gauss Seidal.....(6'a basiniz)\n");
    printf("7Sayisal Turev..... (7'a basiniz) \n");
    printf("8Simpson yontemi..... (8'e basiniz)\n");
    printf("9Trapez yontemi..... (9'ye basiniz) \n");
    printf("10Gregory Newton Enterpolasyonu..... (10'a basiniz) \n\n");
}
```

```

        printf("(q'a basiniz) CIKIS\n");
printf("\nLutfen seciminizi girin: ");
secim = getche();
system("cls");
switch(secim){
    case '1':bisection();break;
    case '2':regulaFalse();break;
    case '3':newtonraphson();break;
    case '4':matrisInvers();break;
    case '5':gaussElimination();break;
    case '6':gaussSeidal();break;
    case '7':numerikTurev();break;
    case '8':simpsonIntegral();break;
    case '9':trapezIntegral();break;
    case '0':GregoryNewton();break;
    //case 'q':exit(0);break;
    default:printf("\nPlease select given number[1-9]");
    menu();
    break;
}
}

void fx_oku(){//modülerlik açısından her fonksiyon girişinde 5.dereceye kadar buradan okuyoruz
    printf("\nf(x) = a*x^5 + b*x^4 + c*x^3 + d*x^2 + e*x^1 + f*x^0\n");
    printf("a = ");scanf("%f",&a);//printf("\n");
    printf("b = ");scanf("%f",&b);//printf("\n");
    printf("c = ");scanf("%f",&c);//printf("\n");
    printf("d = ");scanf("%f",&d);//printf("\n");
    printf("e = ");scanf("%f",&e);//printf("\n");
    printf("f = ");scanf("%f",&f);printf("\n");
}

float fx_hesapla(float x){

    A = a * (x*x*x*x*x);
    B = b * (x*x*x*x);
    C = c * (x*x*x);
    D = d * (x*x);
    E = e * (x);
    return A + B + C + D + E + f;
}

float tureval(float x){
    return (fx_hesapla(x-0.001)-fx_hesapla(x))/(-0.001);
}

```

```

void grafik(){
    float deltax;
    printf("      *****GRAFIK YONTEMI ILE KOK BULMA*****\n");
    fx_oku();
    printf("x0 = ");
    scanf("%f",&xk);
    printf("Delta x = ");
    scanf("%f",&deltax);
    printf("Epsilon = ");
    scanf("%f",&eps);
    xk1 = xk + deltax;
    while(fabs(xk1-xk) >= eps){//bulunan bir sonraki deęerle bir nceki deęerin mutlak
deęerce farkı epsilon'dan <"kk" olunca duracaęız
        if(fx_hesapla(xk)<0){
            while(fx_hesapla(xk)<0){
                xk = xk + deltax;
            }
            xk1 = xk;
            xk = xk - deltax;
            deltax = deltax / 2;
        }
        else{
            while(fx_hesapla(xk)>0){
                xk = xk + deltax;
            }
            xk1 = xk;
            xk = xk - deltax;
            deltax = deltax / 2;
        }
    }
    printf("\nDenklemin koku: %f -----> 2 decimal point'li hali : %.2f\n",xk,xk);
    printf("\nAna ekrana donmek icin lutfen bir tusa basiniz...\n");
    getch();
    system("cls");
    menu();
}

```

```

void bisection(){
    float m,n,q;
    int flag=0;
    printf("\n*****BI-SECTION(YARIYA BOLME) YONTEMI ILE KOK
BULMA*****\n");
    fx_oku();
    printf("Lutfen koku icerisine(kiskaca) alan iki degeri giriniz : ");
    scanf("%f%f",&m,&n);
    printf("Epsilon = ");
    scanf("%f",&eps);

```

```
q = (m+n) / 2; //üst(a) ve alt(b) değerleri toplayıp 2 ye bölerek kök(c) değerini bulmaya başlanır
//printf("\ta\t\t\t\t\tf(a)\t\t\t\t\tb\t\t\t\t\tf(b)\t\t\t\t\tc\t\t\t\t\tf(c)\n-----\n");
//printf("\t%.2ft%.2ft%.2ft%.2ft%.2ft%.2ft\n",m,fx_hesapla(m),n,fx_hesapla(n),q,fx_hesapla(q));
while(fabs(fx_hesapla(q))>=eps){

    //printf("\t%.ft%.ft%.ft%.ft%.ft%.ft\n",m,fx_hesapla(m),n,fx_hesapla(n),q,fx_hesapla(q));

    if(fx_hesapla(m)*fx_hesapla(n)<0){
        q = (m+n) / 2;
        if(fx_hesapla(m)*fx_hesapla(q)>0){
            m = q;
        }
        else{
            n = q;
        }
    }
    else{
        if(fx_hesapla(m)*fx_hesapla(n)==0){
            if(fx_hesapla(m)==0){
                m = q;
            }
            else{
                n = q;
            }
        }
        else{
            flag=1;
            printf("\nGirilen degerler koku icerisine almamaktadır.\n");
            eps=fx_hesapla(q)+1;
        }
    }

    //printf("\t%.2ft%.2ft%.2ft%.2ft%.2ft%.2ft\n",m,fx_hesapla(m),n,fx_hesapla(n),q,fx_hesapla(q));

}

if(flag==0){
    printf("\nDenklemin koku : %f -----> 2 decimal point'li hali : %.2f\n",q,q);
}

printf("\nLutfen menuye donmek icin bir tusa basiniz...\n\n");
getch();
system("cls");
menu();
```

```

}

void regulaFalse(){
    float m,n,q;
    int flag=0;
    printf("\n*****REGULA FALSE YONTEMI ILE KOK
    BULMA*****n");
    fx_oku();
    printf("Lutfen koku icerisine(kiskaca) alan iki degeri giriniz : ");scanf("%f%f",&m,&n);
    printf("Epsilon = ");scanf("%f",&eps);
    q = (fx_hesapla(m)*n-m*fx_hesapla(n))/(fx_hesapla(m)-fx_hesapla(n));
    while(fabs(fx_hesapla(q))>=eps){
        if(fx_hesapla(m)*fx_hesapla(n)<0){
            q = (fx_hesapla(m)*n-m*fx_hesapla(n))/(fx_hesapla(m)-fx_hesapla(n));
            if(fx_hesapla(m)*fx_hesapla(q)>0){
                m = q;
            }
            else{
                n = q;
            }
        }
        else{
            if(fx_hesapla(m)*fx_hesapla(n)==0){
                if(fx_hesapla(m)==0){
                    m = q;
                }
                else{
                    n = q;
                }
            }
            else{
                flag=1;
                printf("\nGirilen degerler koku icerisine almamaktadir.");
                eps=fx_hesapla(q)+1;
            }
        }
    }
    if(flag==0){
        printf("\nDenklemin koku : %f -----> 2 decimal point'li hali : %.2f\n",q,q);
    }
    printf("\nMenuye donmek icin lutfen bir tusa basiniz...\n\n");
    getch();
    system("cls");
    menu();
}

void newtonraphson(){

```



```

printf("\n*****NEWTON RAPHSON YONTEMI ILE KOK BULMA*****\n");
fx_oku();
printf("x0 = ");scanf("%f",&xk);
printf("Epsilon = ");scanf("%f",&eps);
xk1=xk - (fx_hesapla(xk)/tureval(xk));
printf("\txk\t\t\t xk1\t\t\t | xk1-xk | \n-----\n");
while(fabs(xk1-xk) >= eps){
    printf("\t%f\t%f\t%f\n",xk,xk1,fabs(xk1-xk));
    //printf("\t%.2f\t%.2f\t%.2f\n",xk,xk1,fabs(xk1-xk));
    xk = xk1;
    xk1 = xk - (fx_hesapla(xk)/tureval(xk));
}
printf("\nDenklemin koku: %f -----> 2 decimal point'li hali %.2f\n",xk,xk);
printf("Ana ekrana donmek icin lutfen bir tusa basiniz...\n");
getch();
system("cls");
menu();
}

```

```

void matrisInvers(){
    int i,j;
    float A[4][4];
    float birim[3][3] = {{1,0,0},{0,1,0},{0,0,1}}; //Birim matris tanımlama
    printf("*****GIRILEN MATRISIN INVERSINI BULMA*****\n\n");
    //printf("\n\nMatrisin:\n");
    for(i=1;i<4;i++){
        for(j=1;j<4;j++){
            printf("Lutfen %d. satir %d. sutun elemanini giriniz A[%d][%d] = ",i,j,i,j);
            scanf("%f",&A[i][j]);
        }
    }
    printf("\nGirdiginiz matris :");
    for(i=1;i<4;i++){
        printf("\n");
        for(j=1;j<4;j++){
            printf("%10.2f",A[i][j]);
        }
    }

    birim[0][0] /= A[1][1]; birim[0][1] /= A[1][1]; birim[0][2] /= A[1][1];
    A[1][2]/=A[1][1]; A[1][3]/=A[1][1]; A[1][1]/=A[1][1];

    birim[1][0] -= A[2][1]*birim[0][0]; birim[1][1] -= A[2][1]*birim[0][1]; birim[1][2] -=
    A[2][1]*birim[0][2];
    A[2][2] -= A[2][1]*A[1][2]; A[2][3] -= A[2][1]*A[1][3]; A[2][1] -= A[2][1]*A[1][1];

```

```

    birim[2][0] -= A[3][1]*birim[0][0]; birim[2][1] -= A[3][1]*birim[0][1]; birim[2][2] -=
    A[3][1]*birim[0][2];
    A[3][2] -= A[3][1]*A[1][2]; A[3][3] -= A[3][1]*A[1][3]; A[3][1] -= A[3][1]*A[1][1];

    birim[1][0] /= A[2][2]; birim[1][1] /= A[2][2]; birim[1][2] /= A[2][2];
    A[2][1]/=A[2][2]; A[2][3]/=A[2][2]; A[2][2]/=A[2][2];

    birim[0][0] -= A[1][2]*birim[1][0]; birim[0][1] -= A[1][2]*birim[1][1]; birim[0][2] -=
    A[1][2]*birim[1][2];
    A[1][1] -= A[1][2]*A[2][1]; A[1][3] -= A[1][2]*A[2][3]; A[1][2] -= A[1][2]*A[2][2];

    birim[2][0] -= A[3][2]*birim[1][0]; birim[2][1] -= A[3][2]*birim[1][1]; birim[2][2] -=
    A[3][2]*birim[1][2];
    A[3][1] -= A[3][2]*A[2][1]; A[3][3] -= A[3][2]*A[2][3]; A[3][2] -= A[3][2]*A[2][2];

    birim[2][0] /= A[3][3]; birim[2][1] /= A[3][3]; birim[2][2] /= A[3][3];
    A[3][1]/=A[3][3]; A[3][2]/=A[3][3]; A[3][3]/=A[3][3];

    birim[0][0] -= A[1][3]*birim[2][0]; birim[0][1] -= A[1][3]*birim[2][1]; birim[0][2] -=
    A[1][3]*birim[2][2];
    A[1][1] -= A[1][3]*A[3][1]; A[1][2] -= A[1][3]*A[3][2]; A[1][3] -= A[1][3]*A[3][3];

    birim[1][0] -= A[2][3]*birim[2][0]; birim[1][1] -= A[2][3]*birim[2][1]; birim[1][2] -=
    A[2][3]*birim[2][2];
    A[2][1] -= A[2][3]*A[3][1]; A[2][2] -= A[2][3]*A[3][2]; A[2][3] -= A[2][3]*A[3][3];

    printf("\nGirilen matrisin tersi:");
    for(i=0;i<3;i++){
        printf("\n");
        for(j=0;j<3;j++){
            printf("%10.2f",birim[i][j]);
        }
    }
    printf("\nAna ekrana donmek icin lutfen bir tusa basiniz...\n");
    getch();
    system("cls");
    menu();
}

void gaussElimination(){
    float x1,x2,x3;
    float A[4][4];
    float X[4][2];

    printf("\n*****GAUSS ELEMINASYON YONTEMI ILE DOGRUSAL
    DENKLEM TAKIMI COZUMU*****\n");
    printf("\na1*x1 + b1*x2 + c1*x3 = d1\n");

```

```

printf("a2*x1 + b2*x2 + c2*x3 = d2\n");
printf("a3*x1 + b3*x2 + c3*x3 = d3\n");
printf("\na1 = ");scanf("%f",&A[1][1]);
printf("b1 = ");scanf("%f",&A[1][2]);
printf("c1 = ");scanf("%f",&A[1][3]);
printf("d1 = ");scanf("%f",&X[1][1]);
printf("\na2 = ");scanf("%f",&A[2][1]);
printf("b2 = ");scanf("%f",&A[2][2]);
printf("c2 = ");scanf("%f",&A[2][3]);
printf("d2 = ");scanf("%f",&X[2][1]);
printf("\na3 = ");scanf("%f",&A[3][1]);
printf("b3 = ");scanf("%f",&A[3][2]);
printf("c3 = ");scanf("%f",&A[3][3]);
printf("d3 = ");scanf("%f",&X[3][1]);

X[1][1] /= A[1][1];
A[1][2] /= A[1][1]; A[1][3] /=A [1][1]; A[1][1] /=A [1][1];

X[2][1] /= A[2][1];
A[2][2] /= A[2][1]; A[2][3] /=A [2][1]; A[2][1] /=A [2][1];

X[2][1] -= X[1][1];
A[2][1] -= A[1][1]; A[2][2] -= A[1][2]; A[2][3] -= A[1][3];

X[3][1] /= A[3][1];
A[3][2] /= A[3][1]; A[3][3] /=A [3][1]; A[3][1] /=A [3][1];

X[3][1] -= X[1][1];
A[3][1] -= A[1][1]; A[3][2] -= A[1][2]; A[3][3] -= A[1][3];

X[2][1] /= A[2][2];
A[2][1] /= A[2][2]; A[2][3] /=A [2][2]; A[2][2] /=A [2][2];

X[3][1] /= A[3][2];
A[3][1] /= A[3][2]; A[3][3] /=A [3][2]; A[3][2] /=A [3][2];

X[3][1] -= X[2][1];
A[3][1] -= A[2][1]; A[3][2] -= A[2][2]; A[3][3] -= A[2][3];

x3 = X[3][1]/A[3][3];
x2 = (X[2][1]-A[2][3]*x3) / A[2][2];
x1 = (X[1][1]-A[1][3]*x3-A[1][2]*x2) / A[1][1];
printf("\nx1 = %.2f\tx2 = %.2f\tx3 = %.2f\n",x1,x2,x3);
printf("\nAna Menuye donmek icin lutfen bir tusa basiniz...\n");
getch();
system("cls");
menu();

```

```
}
```

```
void trapezIntegral(){
    float I=0.0,ust,alt,deltax,i,n;
    printf("*****TRAPEZ YONTEMI ILE INTEGRAL ALMA*****");
    fx_oku();
    printf("Integralin ust degeri = ");
    scanf("%f",&ust);
    printf("Integralin alt degeri = ");
    scanf("%f",&alt);
    printf("Araligin kaca bolunecegini belirten n degeri = ");
    scanf("%f",&n);
    deltax = (ust-alt)/n;
    for(i = alt+deltax;i<ust;i+=deltax){
        I += fx_hesapla(i);
    }
    I = I + (fx_hesapla(alt) + fx_hesapla(ust))/2;
    I = I * deltax;
    printf("\nIntegralin degeri: %.2f\n",I);
    printf("\nAna menuye donmek icin lutfen bir tusa basiniz...\n\n");
    getch();
    system("cls");
    menu();
}
```

```
void simpsonIntegral(){
    float I=0.0,ust,alt,deltax,h,i,n;
    printf("*****SIMPSON YONTEMI ILE INTEGRAL ALMA*****\n");
    fx_oku();
    printf("Integralin ust degeri = ");scanf("%f",&ust);
    printf("Integralin alt degeri = ");scanf("%f",&alt);
    printf("Araligin kaca bolunecegini belirten n degerini girin (**Cift sayi olmalı**):");scanf("%f",&n);
    h = (ust-alt)/2;
    deltax = (ust-alt)/n;
    for(i = alt+deltax;i<ust;i+=deltax*2){
        I += 4* fx_hesapla(i);
        printf("\t%f",I);
    }
    for(i = alt+2*deltax;i<ust;i+=deltax*2){
        I += 2* fx_hesapla(i);
        printf("\n\t%f",I);
    }
    I = I + fx_hesapla(alt) + fx_hesapla(ust);
    I = I * (deltax/3);
    printf("\nIntegralin degeri : %.2f\n",I);
    printf("\nAna menuye donmek icin lutfen bir tusa basiniz...\n\n");
}
```

```

    getch();
    system("cls");
    menu();
}

void numerikTurev(){
    printf("*****NUMERIK YOLLA TUREV ALMA*****\n");
    float deltax,x0;
    fx_oku();
    printf("x0 = ");scanf("%f",&x0);
    printf("\ndelta x = ");scanf("%f",&deltax);
    float ileri_fark_turev = ( fx_hesapla(x0+deltax) - fx_hesapla(x0) ) / deltax;//böyle de
hesaplanabilir
    float geri_fark_turev = ( fx_hesapla(x0) - fx_hesapla(x0-deltax) ) / deltax;
    //float turev = ( ileri_fark_turev + geri_fark_turev ) / (2*deltax);//küçük bir bug var
    float turev = ( fx_hesapla(x0+deltax) - fx_hesapla(x0-deltax) ) / (2*deltax);
    printf("\nNumerik Turev(Merkezi Fark Turev) = %f\n",turev);//printf("\n\tİleri Fark
Turev = %f\n\tGeri Fark Turev = %f\n",ileri_fark_turev,geri_fark_turev);
    printf("\nAna menuye donmek icin lutfen bir tusa basiniz...\n");
    getch();
    system("cls");
    menu();
}

void gaussSeidal(){
    int n , i, j, r, mxit;
    float x[100][100], a[100], ae, max, t, s, e;
    printf("Denklem sayisini giriniz:\n");
    scanf_s("%d", &n);

    for (i = 0;i < n;i++) a[i] = 0;
    puts(" Denklem elemanlarini girin: \n");
    for (i = 0;i < n;i++)
    {
        for (j = 0;j < n + 1;j++)
        {
            scanf_s("%f", &x[i][j]);
        }
    }
    printf(" İzin verilen hatayi ve maksimum yineleme sayisini girin: ");
    scanf_s("%f%d", &ae, &mxit);
    printf("iterasyon \tx[1]\tx[2]\n");
    for (r = 1;r <= mxit;r++)
    {
        max = 0;
        for (i = 0;i < n;i++)
        {

```

```

        s = 0;
        for (j = 0; j < n; j++)
            if (j != i) s += x[i][j] * a[j];
        t = (x[i][n] - s) / x[i][i];
        e = fabs(a[i] - t);
        a[i] = t;
    }
    printf(" %5d\t", r);
    for (i = 0; i < n; i++)
        printf(" %9.4f\t", a[i]);
    printf("\n");
    if (max < ae)
    {
        printf(" iterasyon %3d \n", r);
        for (i = 0; i < n; i++)
            printf("a[%3d]=%7.4f\n", i + 1, a[i]);
    }
}

    printf("\nAna menuye donmek icin lutfen bir tusa basiniz...\n");
    getch();
    system("cls");
    menu();
}

void GregoryNewton(){
int i, j, m;
    float current_row, matrix[50][50], results[50];

    printf("Matrisin Boyutu : ");
    scanf_s("%d", &m);

    for (i = 0; i < m; i++) {
        for (j = 0; j < m; j++) {
            printf("Matris[%d][%d] : ", i, j);
            scanf_s("%f", &matrix[i][j]);
        }
    }
    for (i = 0; i < m; i++) {
        printf("Sonuc[%d] : ", i);
        scanf_s("%f", &results[i]);
    }
    for (i = 0; i < m; i++) {
        for (j = 0; j < m; j++) {

```

```
        matrix[i][j] = matrix[i][j] / matrix[i][i];
    }
    results[i] = results[i] / matrix[i][i];
}
printf("\n\nMatris :\n");
for (i = 0; i < m; i++) {
    for (j = 0; j < m; j++) {
        printf("%.3f ", matrix[i][j]);
    }
    printf("| %f\n", results[i]);
}
}
```