

Muhammet Kayra Bulut-20011901

1. Bisection
2. Regula Falsi
3. Newton-Raphson
4. $N \times N$ 'lik bir matrisin tersi
5. Gauss Eliminasyon
6. Gauss Seidal
7. Sayısal Türev(merkezi,ilerş,geri)
8. Simpson Yöntemi
9. Trapez Yöntemi

NOT: Bütün Yöntemler yapılmıştır. Sunumda Gauss Seidel ve Merkezi Farkla Türev yöntemleri gösterilmiştir. Gösterilen yöntemlerin kodları ve çalıştırıldığında alınan ekran görüntüleri aşağıdadır.

```
#include <stdio.h>

#include <math.h>

int main(void) {
    int i, a[10], n;
    float e, l, k, kucuk=0.0, buyuk=0.0;
    printf("Lutfen en buyuk ussu giriniz");
    scanf("%d", &n);
    for(i=0; i<=n; i++){
        printf("Lutfen ussu %d olan degiskenin katsayisini giriniz", i);
        scanf("%d", &a[i]);
    }
    printf("Kac noktasindaki turev");
    scanf("%f", &k);
    l=k+0.0001;
    k-=0.0001;
    for(i=0; i<=n; i++){
        kucuk+=pow(k, i)*a[i];
        buyuk+=pow(l, i)*a[i];
    }
    e=(kucuk-buyuk)/(k-l);
    printf("%f", e);
    return 0;}
```



C:\Users\kayra\OneDrive\Masa³st³\programlar\c\Say²sal Analiz\A\Ortalama Turev.

Lutfen en buyuk ussu giriniz2

Lutfen ussu 0 olan degiskenin katsayisini giriniz-15

Lutfen ussu 1 olan degiskenin katsayisini giriniz2

Lutfen ussu 2 olan degiskenin katsayisini giriniz1

Kac noktasindaki turev2

6.003578

Process exited after 17.21 seconds with return value 0

Press any key to continue . . .

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

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

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

```
#define SIZE 10
```

```
int main(void)
```

```
{
```

```
    int i,j,n,m;
```

```
    float a[SIZE][SIZE],b[SIZE],c[SIZE],e;
```

```
    printf("Lutfen degisken sayisini giriniz: ");
```

```
        scanf("%d", &n);
```

```
    printf("Lutfen denklem sayisini giriniz: ");
```

```
    scanf("%d", &m);
```

```
    printf("Lutfen epsilone degerini giriniz: ");
```

```
    scanf("%f", &e);
```

```
    printf("Lutfen denklem ve degiskenleri siralanmis sekilde giriniz:\n");
```

```
    for(j=1;j<=m;j++){
```

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

```
            printf("Lutfen %d. denklemin %d. degiskenini giriniz :  
",j,i);
```

```
            scanf("%f",&a[j][i]);
```

```
        }
```

```
    printf("Lutfen %d. denklemin sonucunu giriniz: \n",j);
```

```
        scanf("%f",&a[j][n+1]);
```

```

    }
    for(i=1;i<=n;i++){
        printf("Lutfen %d. degisken icin baslangic degeri giriniz: \n",i);
        scanf("%f",&b[i]);
    }
    c[1]=b[1]+2*e;
    while(c[1]-b[1]>e || b[1]-c[1]>e || c[2]-b[2]>e || b[2]-c[2]>e || c[3]-
b[3]>e || b[3]-c[3]>e || c[4]-b[4]>e || b[4]-c[4]>e){
        for(i=1;i<=m;i++){
            c[i]=b[i];
            b[i]=a[i][n+1];
            for(j=1;j<=n;j++){
                if(i!=j)
                    b[i]-=a[i][j]*b[j];
            }
            b[i]/=a[i][i];
            printf("%d %f\t",i,b[i]);
        }
        printf("\n");
    }
    for(i=1;i<=n;i++){

printf("%d. degisken = %f\n",i,b[i]);
    }

    return 0;}

```

C:\Users\kayra\OneDrive\Masaüstü\programlar\c\Sayısal Analiz\Gauss seidel.exe

```
Lutfen degisken sayisini giriniz: 3
Lutfen denklem sayisini giriniz: 3
Lutfen epsilone degerini giriniz: 0.001
Lutfen denklem ve degiskenleri siralanmis sekilde giriniz:
Lutfen 1. denklemin 1. degiskenini giriniz : 3
Lutfen 1. denklemin 2. degiskenini giriniz : 1
Lutfen 1. denklemin 3. degiskenini giriniz : 1
Lutfen 1. denklemin sonucunu giriniz:
5
Lutfen 2. denklemin 1. degiskenini giriniz : 1
Lutfen 2. denklemin 2. degiskenini giriniz : -2
Lutfen 2. denklemin 3. degiskenini giriniz : 1
Lutfen 2. denklemin sonucunu giriniz:
0
Lutfen 3. denklemin 1. degiskenini giriniz : 1
Lutfen 3. denklemin 2. degiskenini giriniz : -1
Lutfen 3. denklemin 3. degiskenini giriniz : -1
Lutfen 3. denklemin sonucunu giriniz:
-1
Lutfen 1. degisken icin baslangic degeri giriniz:
0.7
Lutfen 2. degisken icin baslangic degeri giriniz:
0.7
Lutfen 3. degisken icin baslangic degeri giriniz:
0.7
1 1.200000      2 0.950000      3 1.250000
1 0.933333      2 1.091667      3 0.841667
1 1.022222      2 0.931944      3 1.090278
1 0.992593      2 1.041435      3 0.951157
1 1.002469      2 0.976813      3 1.025656
1 0.999177      2 1.012416      3 0.986761
1 1.000274      2 0.993517      3 1.006757
1 0.999909      2 1.003333      3 0.996576
1 1.000031      2 0.998303      3 1.001727
1 0.999990      2 1.000859      3 0.999131
1 1.000003      2 0.999567      3 1.000436
1 0.999999      2 1.000217      3 0.999781
1. degisken = 0.999999
2. degisken = 1.000217
3. degisken = 0.999781
```

```
-----
Process exited after 38.86 seconds with return value 0
Press any key to continue . . .
```