

# Sayısal Analiz

# Projesi Raporu

YILDIZ TEKNİK ÜNİVERSİTESİ  
ELEKTRİK -ELEKTRONİK FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Recep KARAKAYA  
22011068 || recep.karakaya@std.yildiz.edu.tr

# İÇİNDEKİLER

- Ön Bilgi
- Ana Menü
- Desteklenen Fonksiyonlar
- Matris Girişi
- Bisection Yöntemi
- Regula-Falsi Yöntemi
- Newton-Rapshon Yöntemi
- NxN'lik bir matrisin tersi
- Gauus Eleminasyon
- Gauss Seidal yöntemleri
- Sayısal Türev (merkezi, ileri ve geri farklar opsiyonlu)
- Simpson yöntemi,
- Trapez yöntemi
- Değişken dönüşümsüz Gregory newton Enterpolasyonu

# Ön Bilgi

Program, 10 tane belirli işlemi yerine getirebilmek için tasarlanmıştır. Bu işlemler sırasıyla şöyledir:

1. Bisection yöntemi
2. Regula-Falsi yöntemi
3. Newton-Rapshon yöntemi
4. NxN'lik bir matrisin tersi
5. Gauss eliminasyon yöntemi
6. Gauss-Seidel yöntemi
7. Sayısal Türev
8. Simpson yöntemi
9. Trapez yöntemi
10. Değişken dönüşümsüz Gregory-Newton enterpolasyonu

Bu program yukarıdaki tüm işlemleri gerçekleştirir.

1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	1	1

## Ana Menü

Çalıştırılmak istenilen işlem program çalıştırıldıktan sonra numarası girilip gereken parametrelerin verilmesiyle çalışır. Numarası verilen metoda girdikten sonra metottaki fonksiyonlara yapar ve sonrasında programı sonlandırır.

```
hello/ world.  
Asagidaki metodlardan birini seciniz.  
1. Bisection yöntemi  
2. Regula-Falsi yöntemi  
3. Newton-Rapshon yöntemi  
4. NxN'lik bir matrisin tersi  
5. Gauus Eleminasyon Yontemi  
6. Gauss Seidal yöntemi  
7. Sayısal Türev (merkezi, ileri ve geri farklar opsiyonlu)  
8. Simpson yöntemi  
9. Trapez yöntemi  
10. Değişken dönüşümsüz Gregory newton Enterpolasyonu|
```

# Desteklenen Fonksiyonlar

Kök bulma yöntemleri (1, 2, 3), sayısal türev ve integral yöntemleri (7, 8, 9) yöntemleri için ilk istenilen parametre fonksiyondur. Fonksiyon, string olarak tek parça halinde direkt alınır ve polinom, üstel, logaritmik, trigonometrik ve ters trigonometrik fonksiyon tiplerini içerebilir.

String alınan diziyi ilk önce float dizisine çevirdim. Çevirme işlemi yaparken her operatöre ayrı kod verdim ve float dizisinde öyle tuttum. Örneğin, "+" işareti yerine "10001" gibi bir kod kullandım. Daha sonra bu diziyi shunting yard algoritması ile işlem önceliğine göre sıralayıp işlemin sonucunu buldum.

## ❖ Polinom $ax^b$

a: x'in katsayısı

b: x'in üstü

## ❖ Üstel $b^{f(x)}$

b: taban

f(x): üs fonksiyonu

## ❖ Logaritmik

$\log_b(f(x))$

b: logaritmanın tabanı

f(x): logaritması alınacak fonksiyon

## ❖ Trigonometrik

$\text{trig}(f(x))$

f(x): trigonometrik değeri alınacak fonksiyon

## ❖ Ters Trigonometrik

$\text{arctrig}(f(x))$

f(x): ters trigonometrik değeri alınacak fonksiyon

# Matris Girişi

Kullanıcıdan N değeri alındıktan sonra N değerine göre matrisin elemanları girilir.Sonrasında matrisi ekrana yazdırır.

```
1 1 elemanı giriniz.1
1 2 elemanı giriniz.2
1 3 elemanı giriniz.3
2 1 elemanı giriniz.4
2 2 elemanı giriniz.5
2 3 elemanı giriniz.6
3 1 elemanı giriniz.7
3 2 elemanı giriniz.8
3 3 elemanı giriniz.12
1.00 2.00 3.00
4.00 5.00 6.00
7.00 8.00 12.00
```

## Bisection Metodu

Parametreler:

- Fonksiyon
- Kökü arasına alan iki değer
- Epsilon
- Maksimum iterasyon

```
x^3-7x^2+14x-6
Koku icene alan x1 ve x2 degerlerini giriniz.
0 1
Epsilon degeri nedir ?
0.01
Kac iterasyon olsun ?
20
1 . iterasyon
ust = 1.000000 ==>> f(ust) = 2.000000
orta = 0.500000 ==>> f(orta) = -0.625000
alt = 0.000000 ==>> f(alt) = -6.000000
Fark => 0.250000

2 . iterasyon
ust = 1.000000 ==>> f(ust) = 2.000000
orta = 0.750000 ==>> f(orta) = 0.984375
alt = 0.500000 ==>> f(alt) = -0.625000
Fark => 0.125000

3 . iterasyon
ust = 0.750000 ==>> f(ust) = 0.984375
orta = 0.625000 ==>> f(orta) = 0.259766
alt = 0.500000 ==>> f(alt) = -0.625000
Fark => 0.062500

4 . iterasyon
ust = 0.625000 ==>> f(ust) = 0.259766
orta = 0.562500 ==>> f(orta) = -0.161865
alt = 0.500000 ==>> f(alt) = -0.625000
Fark => 0.031250
```

```
5 . iterasyon
ust = 0.625000 ==>> f(ust) = 0.259766
orta = 0.593750 ==>> f(orta) = 0.054047
alt = 0.562500 ==>> f(alt) = -0.161865
Fark => 0.015625
```

```
6 . iterasyon
ust = 0.593750 ==>> f(ust) = 0.054047
orta = 0.578125 ==>> f(orta) = -0.052624
alt = 0.562500 ==>> f(alt) = -0.161865
Fark => 0.007812
```

```
Epsilon degeri farktan daha buyuk.
Kok yaklasik olarak : 0.578125 dir.
f(0.578125)nin degeri -0.052624 dir.
Program ended with exit code: 0
```

## Regula – Falsi Metodu

Parametreler:

- Fonksiyon
- Kökü arasına alan iki değer
- Epsilon
- Maksimum iterasyon

Fonksiyonu giriniz.

'^' üs operatörü, '\_' taban operatörü olarak kullaniniz.

$\cos(x) \cdot x^2 - \log_{10}(10^4)$

Koku icene alan x1 ve x2 degerlerini giriniz.

1 2.5

Epsilon degeri nedir ?

0.001

Kac iterasyon olsun ?

20

1 . iterasyon

ust = 2.500000 ==>> f(ust) = 2.244051

orta = 1.858134 ==>> f(orta) = -0.549154

alt = 1.000000 ==>> f(alt) = -3.000152

Fark => 0.320933

2 . iterasyon

ust = 2.500000 ==>> f(ust) = 2.244051

orta = 1.984327 ==>> f(orta) = -0.064807

alt = 1.858134 ==>> f(alt) = -0.549154

Fark => 0.257836

```
3 . iterasyon
ust = 2.500000 ==>> f(ust) = 2.244051
orta = 1.998802 ==>> f(orta) = -0.007223
alt = 1.984327 ==>> f(alt) = -0.064807
Fark => 0.250599

4 . iterasyon
ust = 2.500000 ==>> f(ust) = 2.244051
orta = 2.000410 ==>> f(orta) = -0.000800
alt = 1.998802 ==>> f(alt) = -0.007223
Fark => 0.249795

5 . iterasyon
ust = 2.500000 ==>> f(ust) = 2.244051
orta = 2.000588 ==>> f(orta) = -0.000088
alt = 2.000410 ==>> f(alt) = -0.000800
Fark => 0.249706

6 . iterasyon
ust = 2.500000 ==>> f(ust) = 2.244051
orta = 2.000607 ==>> f(orta) = -0.000010
alt = 2.000588 ==>> f(alt) = -0.000088
Fark => 0.249696

7 . iterasyon
ust = 2.500000 ==>> f(ust) = 2.244051
orta = 2.000609 ==>> f(orta) = -0.000001
alt = 2.000607 ==>> f(alt) = -0.000010
Fark => 0.249695

8 . iterasyon
ust = 2.500000 ==>> f(ust) = 2.244051
orta = 2.000610 ==>> f(orta) = 0.000000
alt = 2.000609 ==>> f(alt) = -0.000001
Fark => 0.000000
```

Epsilon degeri farktan daha büyük.  
Kok yaklasik olarak : 2.000610 dir.  
f(2.000610)nin degeri 0.000000 dir.  
Program ended with exit code: 0

# Newton-Rapshon Metodu

Parametreler:

- Fonksiyon
- Başlangıç değeri
- Epsilon
- Maksimum iterasyon

```
x^3-7x^2+14x-6
x1 degerini giriniz.
0
Epsilon degeri nedir ?
0.00001
Kac iterasyon olsun ?
20
f(0.00000) in degeri -6.000000 dir.
1 . iterasyon
f(0.433894) in degeri -1.161649 dir.
  Fark => 0.433894
2 . iterasyon
f(0.577197) in degeri -0.059041 dir.
  Fark => 0.143303
3 . iterasyon
f(0.586721) in degeri 0.006378 dir.
  Fark => 0.009524
4 . iterasyon
f(0.585885) in degeri 0.000675 dir.
  Fark => 0.000836
5 . iterasyon
f(0.585791) in degeri 0.000031 dir.
  Fark => 0.000094
6 . iterasyon
f(0.585786) in degeri -0.000004 dir.
  Fark => 0.000005
Epsilon degeri farktan daha buyuk.
Kok yaklasik olarak : 0.585786 dir.
f(0.585786)nin degeri -0.000004 dir.
Program ended with exit code: 0
```



# NxN'lik Bir Matrisin Tersini

## Parametreler:

- Matrisin boyutu
- Matrisin elemanları

Programımda ilk başta verilen matrisin kofaktör matrisi bulur . Sonrasında bu kofaktör matrisin transpozmesini alır ve bunu verilen matrisin determinantına bölerek istenilen ters matrisi elde eder .

## Örnek

$$A = \begin{bmatrix} 5 & 2 & -4 \\ 1 & 4 & 2 \\ 2 & 3 & 6 \end{bmatrix} \quad A^{-1} = \begin{bmatrix} 0,16 & -0,23 & 0,19 \\ -0,02 & 0,36 & -0,13 \\ -0,05 & -0,11 & 0,17 \end{bmatrix}$$

```
Kare matrisin N boyutunu giriniz: 3
```

```
Matrisin elemanlarını giriniz:
```

```
1 1 elemanı giriniz.5
```

```
1 2 elemanı giriniz.2
```

```
1 3 elemanı giriniz.-4
```

```
2 1 elemanı giriniz.1
```

```
2 2 elemanı giriniz.4
```

```
2 3 elemanı giriniz.2
```

```
3 1 elemanı giriniz.2
```

```
3 2 elemanı giriniz.3
```

```
3 3 elemanı giriniz.6
```

```
5.00 2.00 -4.00
```

```
1.00 4.00 2.00
```

```
2.00 3.00 6.00
```

```
Kofaktör matrisi:
```

```
18.00 -24.00 20.00
```

```
-2.00 38.00 -14.00
```

```
-5.00 -11.00 18.00
```

```
Kofaktör matrisinin transpozmesi:
```

```
18.00 -2.00 -5.00
```

```
-24.00 38.00 -11.00
```

```
20.00 -14.00 18.00
```

```
Matrisin tersi:
```

```
0.17 -0.23 0.19
```

```
-0.02 0.36 -0.13
```

```
-0.05 -0.10 0.17
```

```
Matrisin determinanti: 106.00
```

```
Program ended with exit code: 0
```

# Gauss Eliminasyon Yöntemi

Parametreler:

- Bilinmeyen sayısı
- Denklemlerin katsayıları ve sonuçları

Örnek:

$$\begin{aligned}3,6 x + 2,4 y - 1,8 z &= 6,3 \\4,2 x - 5,8 y + 2,1 z &= 7,5 \\0,8 x + 3,5 y + 6,5 z &= 3,7\end{aligned}$$

```
Bilinmeyen sayisini giriniz3
Denklem katsayilarini sirasiyla girin:
Denklem 1 katsayilari: 3.6 2.4 -1.8 6.3
Denklem 2 katsayilari: 4.2 -5.8 2.1 7.5
Denklem 3 katsayilari: 0.8 3.5 6.5 3.7

Denklem sisteminin cozumleri:
x1 = 1.8108
x2 = 0.1201
x3 = 0.2817
Program ended with exit code: 0
```

# Gauss-Seidal Yöntemi

Parametreler:

- Bilinmeyen sayısı
- Denklemlerin katsayıları ve sonuçları
- Epsilon
- Maksimum iterasyon
- Başlangıç Değerleri

Örnek:

$$\begin{aligned}-x + 4y - 3z &= -8 \\3x + y - 2z &= 9 \\x - y + 4z &= 1\end{aligned}$$

Gauss Seidal yontemine hosgeldiniz!  
Bu yontem 3 e 3 bir matris icin uygulanir.  
1. denklemi giriniz (katsayılar ve sabit terim): -1 4 -3 -8  
2. denklemi giriniz (katsayılar ve sabit terim): 3 1 -2 9  
3. denklemi giriniz (katsayılar ve sabit terim): 1 -1 4 1

Başlangıç değerlerini giriniz:  
x1: 1  
x2: 1  
x3: 1

İterasyon sayısını giriniz: 20

Epsilon giriniz: 0.0001

1.iterasyon  
x1: 3.333333  
x2: 1.000000  
x3: 1.000000  
Aradaki fark degeri: 2.333333dir.  
x1: 3.333333  
x2: -0.416667  
x3: 1.000000  
Aradaki fark degeri: 1.416667dir.  
x1: 3.333333  
x2: -0.416667  
x3: -0.687500  
Aradaki fark degeri: 1.687500dir.

4.iterasyon  
x1: 2.979462  
x2: -1.904261  
x3: -0.982937  
Aradaki fark degeri: 0.048027dir.  
x1: 2.979462  
x2: -1.992337  
x3: -0.982937  
Aradaki fark degeri: 0.088077dir.  
x1: 2.979462  
x2: -1.992337  
x3: -0.992950  
Aradaki fark degeri: 0.010013dir.

5.iterasyon  
x1: 3.002146  
x2: -1.992337  
x3: -0.992950  
Aradaki fark degeri: 0.022684dir.  
x1: 3.002146  
x2: -1.994176  
x3: -0.992950  
Aradaki fark degeri: 0.001838dir.  
x1: 3.002146  
x2: -1.994176  
x3: -0.999080  
Aradaki fark degeri: 0.006131dir.

2.iterasyon  
x1: 2.680556  
x2: -0.416667  
x3: -0.687500  
Aradaki fark degeri: 0.652778dir.  
x1: 2.680556  
x2: -1.845486  
x3: -0.687500  
Aradaki fark degeri: 1.428819dir.  
x1: 2.680556  
x2: -1.845486  
x3: -0.881510  
Aradaki fark degeri: 0.194010dir.

3.iterasyon  
x1: 3.027488  
x2: -1.845486  
x3: -0.881510  
Aradaki fark degeri: 0.346933dir.  
x1: 3.027488  
x2: -1.904261  
x3: -0.881510  
Aradaki fark degeri: 0.058774dir.  
x1: 3.027488  
x2: -1.904261  
x3: -0.982937  
Aradaki fark degeri: 0.101427dir.

6.iterasyon  
x1: 2.998672  
x2: -1.994176  
x3: -0.999080  
Aradaki fark degeri: 0.003474dir.  
x1: 2.998672  
x2: -1.999642  
x3: -0.999080  
Aradaki fark degeri: 0.005466dir.  
x1: 2.998672  
x2: -1.999642  
x3: -0.999579  
Aradaki fark degeri: 0.000498dir.

7.iterasyon  
x1: 3.000162  
x2: -1.999642  
x3: -0.999579  
Aradaki fark degeri: 0.001490dir.  
x1: 3.000162  
x2: -1.999643  
x3: -0.999579  
Aradaki fark degeri: 0.000001dir.  
x1: 3.000162  
x2: -1.999643  
x3: -0.999951  
Aradaki fark degeri: 0.000373dir.

```

8.iterasyon
x1: 2.999913
x2: -1.999643
x3: -0.999951
Aradaki fark degeri: 0.000248dir.
x1: 2.999913
x2: -1.999985
x3: -0.999951
Aradaki fark degeri: 0.000342dir.
x1: 2.999913
x2: -1.999985
x3: -0.999975
Aradaki fark degeri: 0.000023dir.

9.iterasyon
x1: 3.000012
x2: -1.999985
x3: -0.999975
Aradaki fark degeri: 0.000098dir.
x1: 3.000012
x2: -1.999978
x3: -0.999975
Aradaki fark degeri: 0.000007dir.
x1: 3.000012
x2: -1.999978
x3: -0.999997
Aradaki fark degeri: 0.000023dir.
Aradaki fark degeri epsilondan daha kucuk
Çözümler:
x1 = 3.000012
x2 = -1.999978
x3 = -0.999997
Program ended with exit code: 0

```

## Sayısal Türev Yöntemleri

Parametreler:

- Fonksiyon
- Türevinin bulunması istenen nokta
- Fark(h)
- Türevin yöntemi

```

7
Fonksiyonu giriniz.
'^' üs operatörü, '_' taban operatörü olarak kullaniniz.
log_e(x)
Hangi nokta için turev almak istedinizi ve h degerini giriniz
5 0.1
Asagidakilerden birini seciniz
1.İleri Farkli Sayisal Turev
2.Geri Farkli Sayisal Turev
3.Merkezi Farkli Sayisal Turev
3
Merkezi Farkli Sayisal Turev
f'(5.000000) in degeri 0.002000dir.
Program ended with exit code: 0

```

# Simpson Yöntemleri

Hem simpson 1/3 kuralı hem de simpson 3/8 kuralına göre sonuç alınabilir.

## Parametreler:

- Kullanılması istenen simpson kuralı
- İntegralinin bulunması istenen fonksiyon
- Alt ve üst sınır
- Aralığı kaç bölseğini belirten n değeri

## Simpson 3/8 Metodu

```
8
Fonksiyonu giriniz.
'^' üs operatörü, '_' taban operatörü olarak kullaniniz.
1/(1+x^4)
Asagidakilerden birini seciniz
1.1/3Simpson Metodu
2.3/8Simpson Metodu
2
n ikinin kati olmasi lazim
Hangi noktalar icin Simpson metodunu yapmak istedinizi ve n degerini giriniz
0 6 6
Verdiginiz fonksiyonun alani Simpson 3/8 e gore 1.017489 dir.
Program ended with exit code: 0
```

## Simpson 1/8 Metodu

```
8
Fonksiyonu giriniz.
'^' üs operatörü, '_' taban operatörü olarak kullaniniz.
x^3+2x^2-x-2
Asagidakilerden birini seciniz
1.1/3Simpson Metodu
2.3/8Simpson Metodu1
n ikinin kati olmasi lazim
Hangi noktalar icin Simpson metodunu yapmak istedinizi ve n degerini giriniz
-2 -1 4
Verdiginiz fonksiyonun alani Simpson 1/3 e gore 0.416667 dir.
Program ended with exit code: 0
```

# Trapez Yöntemi

## Parametreler:

- İntegralinin bulunması istenen fonksiyon
- Alt ve üst sınır
- Aralığı kaç bölseğini belirten n değeri

```
9
Fonksiyonu giriniz.
'^' üs operatörü, '_' taban operatörü olarak kullanınız.
1/(1+x^2)
Hangi noktalar için Trapez metodunu yapmak istedinizi ve n degerini giriniz
0 1 9
Verdiginiz fonksiyonun alani Trapez Yontemine gore 0.784884 dir.
Program ended with exit code: 0
```

# Gregory-Newton Enterpolasyonu

## Parametreler:

- Nokta sayısı
- Noktaların kendisi ve görüntüleri
- Sonucunun bulunması istenen değer

```
10
Gregory Newton Enterpolasyonuna Hosgeldiniz!
Kac tane nokta girmek istiyorsunuz?
x1 koordinatini ve y1 degerieni giriniz0 -4
x2 koordinatini ve y2 degerieni giriniz1 -2
x3 koordinatini ve y3 degerieni giriniz2 14
x4 koordinatini ve y4 degerieni giriniz3 62
x5 koordinatini ve y5 degerieni giriniz4 160
x6 koordinatini ve y6 degerieni giriniz5 326
x7 koordinatini ve y7 degerieni giriniz6 578
Hesaplanmasini istediginiz x degerini giriniz.
6
istediginiz 6.000000 degeeri icin sonuc : 578.00 dir.
Program ended with exit code: 0
```