CENG 235 ALGORİTMALARLA SAYISAL ÇÖZÜMLEME

Doç. Dr. Tufan TURACI
tturaci@pau.edu.tr

- Pamukkale Üniversitesi
- Mühendislik Fakültesi
- Bilgisayar Mühendisliği Bölümü

- Hafta 9

9. Hafta Konular

- **Lineer Denklem Sistemlerinin Çözüm Yöntemleri**

  Sayısal Çözümler

  --- Jacobi İterasyon Yöntemi

  --- Gauss-Sidel İterasyon Yöntemi

Lineer Denklem Sistemleri

n bilinmeyenli n denklemli bir lineer denklem sistemi aşağıdaki şekilde açık formda ifade edilir:

$$
\begin{gathered}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\
\vdots \\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n
\end{gathered}
$$

Burada: $a_{11}, a_{12}, \dots, a_{nn}$ ‘ ler sistemin katsayılar olarak ifade edilir.

$x_1, x_2, \dots, x_\mathrm{n}$‘ ler bilinmeyenler olarak ifade edilir.

$b_1, b_2, \dots, b_n$‘ ler sağ taraf sabitleri olarak ifade edilir.

Lineer denklem sistemlerini çözmek için var olan iki yöntem olan Jacobi iterasyonu ve Gauss-Seidel İterasyon yöntemlerini geçmeden önce Köşegen Baskın Matris tanımını vereceğiz.

Köşegen Baskın Matris (Diagonally Dominant Matrix):

*n*n* boyutlu bir A matrisi her bir satır için aşağıdaki şartı sağlıyorsa

$$|a_{kk}| > |a_{k1}| + |a_{k2}| + \cdots + |a_{kn}|, (k = 1,2, \dots, n)$$

, bu durumda A matrisine köşegen baskın matris denir.

Örnek: Aşağıdaki iki matrisin köşegen baskın matris olup olmadığını kontrol ediniz.

$$A = \begin{bmatrix} 3 & 1 & -1 \\ 2 & -5 & 2 \\ 1 & 6 & 8 \end{bmatrix} \Rightarrow \text{A matrisi köşegen baskın matristir.}$$

$$|3| > |1| + |-1| \Rightarrow 3 > 2$$

$$|-5| > |2| + |2| \Rightarrow 5 > 4$$

$$|8| > |1| + |6| \Rightarrow 8 > 7$$

$$B = \begin{bmatrix} 3 & 2 & 6 \\ 1 & 8 & 1 \\ 9 & 2 & -2 \end{bmatrix} \Rightarrow$$ B matrisi köşegen baskın matris değildir.

$|3| > |2| + |6| \Rightarrow 3 > 8$ ✗

$|8| > |1| + |1| \Rightarrow 8 > 2$

$|-2| > |9| + |2| \Rightarrow 2 > 11$ ✗

Jacobi İterasyon Yöntemi

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n \end{aligned} \qquad (1)$$

n denklemli n bilinmeyenli (1) Lineer denklem sistemi verilsin.

Burada:

$$x_1 = \frac{b_1 - (a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n)}{a_{11}}$$

$$x_2 = \frac{b_2 - (a_{21}x_1 + a_{23}x_3 + \cdots + a_{2n}x_n)}{a_{22}}$$

$$\vdots \qquad \vdots$$

$$x_n = \frac{b_n - (a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{n(n-1)}x_{n-1})}{a_{nn}}$$

eşitlikleri elde edilir.

Jacobi iterasyon yönteminde bilinmeyenlere bir başlangıç değeri verilir.

$$x_1^0 = 0 \quad , x_2^0 = 0 , \ldots , x_n^0 = 0$$ olsun.

Yeni $x_1^1, x_2^1, \ldots, x_n^1$ değerleri bu başlangıç değerine göre hesaplanır.

Burada:

$$x_1^1 = \frac{b_1 - (a_{12} x_2^0 + a_{13} x_3^0 + \cdots + a_{1n} x_n^0)}{a_{11}}$$

$$x_2^1 = \frac{b_2 - (a_{21} x_1^0 + a_{23} x_3^0 + \cdots + a_{2n} x_n^0)}{a_{22}}$$

$$\vdots$$

$$x_n^1 = \frac{b_n - (a_{n1} x_1^0 + a_{n2} x_2^0 + \cdots + a_{n(n-1)} x_{n-1}^0)}{a_{nn}}$$

elde edilir.

iterasyon bu şekilde devam eder.

Her iterasyon sonunda hata toleransına ulaşıp ulaşılmadığı kontrol edilir.

Her x_i için $|x_i^{k+1} - x_i^k| < \varepsilon$, $i=1,2,\ldots,n$

kontrol edilir.

Tüm bilinmeyenler için bu sağlanırsa iterasyon durdurulur.

Jacobi iterasyonunun yakınsama şartı:

Teorem 1: A matrisi köşegen baskın bir matris olmak üzere $AX = B$ sisteminin seçilen her hangi bir P_0 noktası için Jacobi iterasyonu tek bir çözüme yakınsar.

Örnek:

$$8x_1 + 2x_2 + 3x_3 = 30$$
$$x_1 - 9x_2 + 2x_3 = 1$$
$$2x_1 + 3x_2 + 6x_3 = 31$$

lineer denklem sistemi verilsin. $x_1^0 = 0, x_2^0 = 0, x_3^0 = 0$ $(0,0,0)$ başlangıç noktasından başlayarak Jacobi iterasyonu ile $\varepsilon = 0{,}0001$ hata ve 4 ondalık kullanarak sistemi çözünüz.

Katsayılar matrisi: köşegen baskın mı?

$$\begin{bmatrix} 8 & 2 & 3 \\ 1 & -9 & 2 \\ 2 & 3 & 6 \end{bmatrix} \begin{array}{l} \longrightarrow |8| > |2| + |3| \; ✓ \\ \longrightarrow |-9| > |1| + |2| \; ✓ \\ \longrightarrow |6| > |2| + |3| \; ✓ \end{array}$$

Köşegen baskın matristir.

$$x_1 = \frac{30 - (2x_2 + 3x_3)}{8} \Rightarrow x_1^{n+1} = \frac{30 - (2x_2^{n} + 3x_3^{n})}{8}$$

$$x_2 = \frac{1 - (x_1 + 2x_3)}{-9} \Rightarrow x_2^{n+1} = \frac{1 - (x_1^{n} + 2x_3^{n})}{-9}$$

$$x_3 = \frac{31 - (2x_1 + 3x_2)}{6} \Rightarrow x_3^{n+1} = \frac{31 - (2x_1^{n} + 3x_2^{n})}{6}$$

iterasyon formülleri elde edilir.

$x_1^0 = 0$, $x_2^0 = 0$, $x_3^0 = 0$ başlangıç noktasını

alarak $x_1^1 = 3.7500$, $x_2^1 = -0.1111$, $x_3^1 = 5.1667$

elde edilir.

$$x_1 = \frac{30 - (2.0 + 3.0)}{8} = 3.7500$$

$$x_2 = \frac{1 - (0 + 2.0)}{-9} = -0.1111$$

$$x_3 = \frac{31 - (2.0 + 3.0)}{6} = 5.1667$$

$$|x_1^1 - x_1^0| = 3.7500 > \varepsilon = 10^{-4}$$
$$|x_2^1 - x_2^0| = 0.1111 > \varepsilon = 10^{-4}$$
$$|x_3^1 - x_3^0| = 5.166 > \varepsilon = 10^{-4}$$

$$x_1^{(2)} = \frac{30 - (2.(-0{,}1111) + 3.(5.1667))}{8} = 1.8403$$

$$x_2^{(2)} = \frac{1 - (3.7500 + 2.(5.1667))}{-9} = 1.4537$$

$$x_3^{(2)} = \frac{31 - (2.(3.7500) + 3.(-0.1111))}{6} = 3.9722$$

$$|x_1^2 - x_1^1| > \varepsilon = 10^{-4}$$

$$|x_2^2 - x_2^1| > \varepsilon = 10^{-4}$$

$$|x_3^2 - x_3^1| > \varepsilon = 10^{-4}$$

iterasyona devam edilir...

12. iterasyon sonucunda

$x_1 = 2,0000$

$x_2 = 1,0000$

$x_3 = 4,0000$

elde edilir.

C kodu:

```
#include<stdio.h>
#include<conio.h>
#include<locale.h>
#include<math.h>
#define hata 0.0001

float f1(float a,float b,float c)
{return (30-(2*b+3*c))/8;}

float f2(float a,float b,float c)
{return (1-(a+2*c))/-9;}

float f3(float a,float b,float c)
{return (31-(2*a+3*b))/6;}
```

```
int main()
{setlocale(LC_ALL, "Turkish");
float x1=0,x2=0,x3=0,temp1,temp2,temp3; int i=0;
printf("Yönteme başladığımız nokta= (%.4f,%.4f, %.4f)\n",x1,x2,x3);
do
{temp1=x1;
 temp2=x2;
 temp3=x3;
 x1=f1(temp1,temp2,temp3);
 x2=f2(temp1,temp2,temp3);
 x3=f3(temp1,temp2,temp3);
 i++;
printf("%d. adımda yaklaşık değer= (%.4f,%.4f,%.4f )\n",i,x1,x2,x3);
}while ((fabs(x1-temp1)>hata)||(fabs(x2-temp2)>hata)||(fabs(x3-temp3)>hata));
printf("yaklaşık kök x1=%.4f\n",x1);
printf("yaklaşık kök x2=%.4f\n",x2);
printf("yaklaşık kök x3=%.4f\n",x3);
getch ();
return 0;
}
```

Ekran Çıktısı:

```
Yönteme başladığımız nokta= (0,0000,0,0000, 0,0000)
1. adımda yaklaşık değer= (3,7500,-0,1111,5,1667 )
2. adımda yaklaşık değer= (1,8403,1,4537,3,9722 )
3. adımda yaklaşık değer= (1,8970,0,9761,3,8264 )
4. adımda yaklaşık değer= (2,0711,0,9500,4,0463 )
5. adımda yaklaşık değer= (1,9951,1,0182,4,0013 )
6. adımda yaklaşık değer= (1,9950,0,9998,3,9925 )
7. adımda yaklaşık değer= (2,0029,0,9978,4,0018 )
8. adımda yaklaşık değer= (1,9999,1,0007,4,0002 )
9. adımda yaklaşık değer= (1,9998,1,0000,3,9997 )
10. adımda yaklaşık değer= (2,0001,0,9999,4,0001 )
11. adımda yaklaşık değer= (2,0000,1,0000,4,0000 )
12. adımda yaklaşık değer= (2,0000,1,0000,4,0000 )
yaklaşık kök x1=2,0000
yaklaşık kök x2=1,0000
yaklaşık kök x3=4,0000


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

Örnek:

$$
\begin{aligned}
4x_1 + x_2 + x_3 &= 1 \\
x_1 + 4x_2 + x_4 &= 2 \\
x_1 + 4x_3 + x_4 &= 0 \\
x_2 + x_3 + 4x_4 &= 1
\end{aligned}
$$

lineer denklem sistemini

$x_1^0 = 0$, $x_2^0 = 0$, $x_3^0 = 0$, $x_4^0 = 0$ başlangıç

değerlerini kullanarak $\varepsilon = 10^{-4}$ hata ve

4 ondalık kullanarak Jacobi iterasyonu

ile bulunuz.

Katsayılar Matrisi:

$$\begin{bmatrix} 4 & 1 & 1 & 0 \\ 1 & 4 & 0 & 1 \\ 1 & 0 & 4 & 1 \\ 0 & 1 & 1 & 4 \end{bmatrix} \Rightarrow \text{köşegen baskın matris.}$$

Çözüme yakınsar ---

Çözüm:

```
Yönteme başladığımız nokta= (0,0000,  0,0000,  0,0000,  0,0000)
1. adımda yaklaşık değer= (0,2500,  0,5000,  -0,0000,  0,2500 )
2. adımda yaklaşık değer= (0,1250,  0,3750,  -0,1250,  0,1250 )
3. adımda yaklaşık değer= (0,1875,  0,4375,  -0,0625,  0,1875 )
4. adımda yaklaşık değer= (0,1563,  0,4063,  -0,0938,  0,1563 )
5. adımda yaklaşık değer= (0,1719,  0,4219,  -0,0781,  0,1719 )
6. adımda yaklaşık değer= (0,1641,  0,4141,  -0,0859,  0,1641 )
7. adımda yaklaşık değer= (0,1680,  0,4180,  -0,0820,  0,1680 )
8. adımda yaklaşık değer= (0,1660,  0,4160,  -0,0840,  0,1660 )
9. adımda yaklaşık değer= (0,1670,  0,4170,  -0,0830,  0,1670 )
10. adımda yaklaşık değer= (0,1665,  0,4165,  -0,0835,  0,1665 )
11. adımda yaklaşık değer= (0,1667,  0,4167,  -0,0833,  0,1667 )
12. adımda yaklaşık değer= (0,1666,  0,4166,  -0,0834,  0,1666 )
13. adımda yaklaşık değer= (0,1667,  0,4167,  -0,0833,  0,1667 )
yaklaşık kök x1=0,1667
yaklaşık kök x2=0,4167
yaklaşık kök x3=-0,0833
yaklaşık kök x4=0,1667


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

Ödev:

$$4x - y + z = 7$$
$$4x - 8y + z = -21$$
$$-2x + y + 5z = 15$$

lineer denklem sistemini

$x_1^0 = 1, \; x_2^0 = 2, \; x_3^0 = 2$ başlangıç

değerlerini kullanarak $\varepsilon = 10^{-6}$ hata ve

6 ondalık kullanarak Jacobi iterasyonu

ile bulunuz.

Çözüm:

```
Yönteme başladığımız nokta= (1,000000,2,000000, 2,000000)
1. adımda yaklaşık değer= (1,750000,3,375000,3,000000 )
2. adımda yaklaşık değer= (1,843750,3,875000,3,025000 )
3. adımda yaklaşık değer= (1,962500,3,925000,2,962500 )
4. adımda yaklaşık değer= (1,990625,3,976563,3,000000 )
5. adımda yaklaşık değer= (1,994141,3,995312,3,000938 )
6. adımda yaklaşık değer= (1,998594,3,997188,2,998594 )
7. adımda yaklaşık değer= (1,999648,3,999121,3,000000 )
8. adımda yaklaşık değer= (1,999780,3,999824,3,000035 )
9. adımda yaklaşık değer= (1,999947,3,999894,2,999947 )
10. adımda yaklaşık değer= (1,999987,3,999967,3,000000 )
11. adımda yaklaşık değer= (1,999992,3,999993,3,000001 )
12. adımda yaklaşık değer= (1,999998,3,999996,2,999998 )
13. adımda yaklaşık değer= (2,000000,3,999999,3,000000 )
14. adımda yaklaşık değer= (2,000000,4,000000,3,000000 )
yaklaşık kök x1=2,000000
yaklaşık kök x2=4,000000
yaklaşık kök x3=3,000000


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

Gauss- Seidel İterasyon Yöntemi

$$\begin{array}{c}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\
\vdots \\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n
\end{array} \qquad (1)$$

n denklemli *n* bilinmeyenli (1) Lineer denklem sistemi verilsin.

Gauss-Seidel iterasyon yöntemi, Jacobi iterasyon yöntemine benzer.
Fakat, burada $x_k^{(i)}$'lerin bir önceki iterasyondaki değerleri değil, yeni elde edilen değerleri kullanılır.

$x_1^0 = 0, \quad x_2^0 = 0, \quad ---, \quad x_n^0 = 0$ olsun.

Yeni $x_1^1, x_2^1, ---, x_n^1$ değerleri bu başlangıç değerine göre hesaplanır.

Burada:

$$x_1^1 = \frac{b_1 - (a_{12} x_2^0 + a_{13} x_3^0 + --- + a_{1n} x_n^0)}{a_{11}}$$

$$x_2^1 = \frac{b_2 - (a_{21} x_1^1 + a_{23} x_3^0 + --- + a_{2n} x_n^0)}{a_{22}}$$

$$\vdots$$

$$x_n^1 = \frac{b_n - (a_{n1} x_1^1 + a_{n2} x_2^1 + --- + a_{n(n-1)} x_{n-1}^0)}{a_{nn}}$$

elde edilir.

Gauss-Seidel iterasyonunun yakınsama şartı:

Yakınsama şartı Jacobi iterasyonundaki gibi aşağıdaki teorem ile ifade edilir.

Teorem 2: A matrisi köşegen baskın bir matris olmak üzere $AX = B$ sisteminin seçilen her hangi bir P_0 vektörü için Gauss-Seidel iterasyonu tek bir çözüme yakınsar.

Örnek:

$$8x_1 + 2x_2 + 3x_3 = 30$$
$$x_1 - 9x_2 + 2x_3 = 1$$
$$2x_1 + 3x_2 + 6x_3 = 31$$

lineer denklem sistemi verilsin.

$x_1^0 = 0, x_2^0 = 0, x_3^0 = 0$ $(0,0,0)$ başlangıç noktasından başlayarak Gauss-Seidel yöntemi ile $\varepsilon = 0.001$ hata ve 4 ondalık kullanarak sistemi çözünüz.

Katsayılar matrisi köşegen baskın'dır.

Tek bir çözüme yakınsar.

$x_1^{(0)} = 0$, $x_2^{(0)} = 0$, $x_3^{(0)} = 0$ başlangıç değerleri.

$$x_1 = \frac{30 - (2x_2 + 3x_3)}{8} \Rightarrow x_1^{n+1} = \frac{30 - (2x_2^n + 3x_3^n)}{8}$$

$$x_2 = \frac{1 - (x_1 + 2x_3)}{-9} \Rightarrow x_2^{n+1} = \frac{1 - (x_1^n + 2x_3^n)}{-9}$$

$$x_3 = \frac{31 - (2x_1 + 3x_2)}{6} \Rightarrow x_3^{n+1} = \frac{31 - (2x_1^n + 3x_2^n)}{6}$$

iterasyon formülleri elde edilir.

$x_1^0 = 0,\ x_2^0 = 0,\ x_3^0 = 0$ başlangıç noktasını

alarak $x_1^1 = 1.7500,\ x_2^1 = 0,3056,\ x_3^1 = 3,7639$

elde edilir.

$$x_1 = \frac{10 - (2.0 + 3.0)}{8} = 1.7500$$

$$x_2 = \frac{1 - (3.7500 + 2.0)}{-9} = 0.3056$$

$$x_3 = \frac{31 - (2.(1.7500) + 3.(0.3056))}{6} = 3.7639$$

$$|x_1^1 - x_1^0| = 1.7500 > \varepsilon = 10^{-4}$$
$$|x_2^1 - x_2^0| = 0.3056 > \varepsilon = 10^{-4}$$
$$|x_3^1 - x_3^0| = 3.7639 > \varepsilon = 10^{-4}$$

7. iterasyon sonucunda

$x_1 = 2,0000$

$x_2 = 1,0000$

$x_3 = 4,0000$

elde edilir.

C kodu:

```
#include<stdio.h>
#include<conio.h>
#include<locale.h>
#include<math.h>
#define hata 0.0001

float f1(float a,float b,float c)
{return (30-(2*b+3*c))/8;}

float f2(float a,float b,float c)
{return (1-(a+2*c))/-9;}

float f3(float a,float b,float c)
{return (31-(2*a+3*b))/6;}
```

```
int main()
{setlocale(LC_ALL, "Turkish");
float x1=0,x2=0,x3=0,temp1,temp2,temp3; int i=0;
printf("Yönteme başladığımız nokta= (%.4f,%.4f, %.4f)\n",x1,x2,x3);
do
{temp1=x1;
 temp2=x2;
 temp3=x3;
  x1=f1(temp1,temp2,temp3);
  x2=f2(x1,temp2,temp3);
  x3=f3(x1,x2,temp3);
  i++;
printf("%d. adımda yaklaşık değer= (%.4f,%.4f,%.4f )\n",i,x1,x2,x3);
}while ((fabs(x1-temp1)>hata)||(fabs(x2-temp2)>hata)||(fabs(x3-temp3)>hata));

printf("yaklaşık kök x1=%.4f\n",x1);
printf("yaklaşık kök x2=%.4f\n",x2);
printf("yaklaşık kök x3=%.4f\n",x3);
getch ();
return 0;
}
```

Ekran Çıktısı:

```
Yönteme başladığımız nokta= (0,0000,0,0000, 0,0000)
1. adımda yaklaşık değer= (3,7500,0,3056,3,7639 )
2. adımda yaklaşık değer= (2,2622,0,9767,3,9243 )
3. adımda yaklaşık değer= (2,0342,0,9870,3,9951 )
4. adımda yaklaşık değer= (2,0051,0,9995,3,9986 )
5. adımda yaklaşık değer= (2,0007,0,9998,3,9999 )
6. adımda yaklaşık değer= (2,0001,1,0000,4,0000 )
7. adımda yaklaşık değer= (2,0000,1,0000,4,0000 )
yaklaşık kök x1=2,0000
yaklaşık kök x2=1,0000
yaklaşık kök x3=4,0000


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

Örnek:

$$
\begin{aligned}
4x_1 + x_2 + x_3 &= 1 \\
x_1 + 4x_2 + x_4 &= 2 \\
x_1 + 4x_3 + x_4 &= 0 \\
x_2 + x_3 + 4x_4 &= 1
\end{aligned}
$$

lineer denklem sistemini

$x_1^0 = 0$, $x_2^0 = 0$, $x_3^0 = 0$, $x_4^0 = 0$ başlangıç değerlerini kullanarak $\varepsilon = 10^{-4}$ hata ve 4 ondalık kullanarak Gauss-Seidel iterasyonu ile bulunuz.

Katsayılar matrisi köşegen baskındır.

Çözüm:

```
Yönteme başladığımız nokta= (0,0000,  0,0000,  0,0000,  0,0000)
1. adımda yaklaşık değer= (0,2500,  0,4375,  -0,0625,  0,1563 )
2. adımda yaklaşık değer= (0,1563,  0,4219,  -0,0781,  0,1641 )
3. adımda yaklaşık değer= (0,1641,  0,4180,  -0,0820,  0,1660 )
4. adımda yaklaşık değer= (0,1660,  0,4170,  -0,0830,  0,1665 )
5. adımda yaklaşık değer= (0,1665,  0,4167,  -0,0833,  0,1666 )
6. adımda yaklaşık değer= (0,1666,  0,4167,  -0,0833,  0,1667 )
7. adımda yaklaşık değer= (0,1667,  0,4167,  -0,0833,  0,1667 )
yaklaşık kök x1=0,1667
yaklaşık kök x2=0,4167
yaklaşık kök x3=-0,0833
yaklaşık kök x4=0,1667


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

Ödev:

$$4x - y + z = 7$$
$$4x - 8y + z = -21$$
$$-2x + y + 5z = 15$$

lineer denklem sistemini

$x_1^0 = 1, \; x_2^0 = 2, \; x_3^0 = 2$ başlangıç

değerlerini kullanarak $\varepsilon = 10^{-6}$ hata ve

6 ondalık kullanarak Gauss-Seidel iterasyonu

ile bulunuz.

Çözüm:

```
Yönteme başladığımız nokta= (1,000000,2,000000, 2,000000)
1. adımda yaklaşık değer= (1,750000,3,750000,2,950000 )
2. adımda yaklaşık değer= (1,950000,3,968750,2,986250 )
3. adımda yaklaşık değer= (1,995625,3,996094,2,999031 )
4. adımda yaklaşık değer= (1,999266,3,999512,2,999804 )
5. adımda yaklaşık değer= (1,999927,3,999939,2,999983 )
6. adımda yaklaşık değer= (1,999989,3,999992,2,999997 )
7. adımda yaklaşık değer= (1,999999,3,999999,3,000000 )
8. adımda yaklaşık değer= (2,000000,4,000000,3,000000 )
yaklaşık kök x1=2,000000
yaklaşık kök x2=4,000000
yaklaşık kök x3=3,000000


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

İterasyon Yöntemlerinin Avantajları

- $AX=B$ denklem sistemi direkt yöntemler ile çözülemeyecek kadar büyük ise ve A seyrek bir matris ise iterasyon yöntemleri ile çözülebilir.
- A seyrek matris ise çok az bellek gerektirir. Çünkü, çok az dört işlem yapılır ve çok daha az hesap süresi gereklidir.

- Optimizasyon gibi tekrarlanan problemlerin çözümüne uygundur. Çünkü bir önceki çözüm bir sonraki çözümün başlangıç vektörü olarak alınabilir, çözüm çok çabuk yakınsar.
- İterasyon metotlarında $AX=B$ denklem sisteminin A katsayılar matrisi değişikliğe uğramaz. Buna karşın; direkt metotlarda A nın elemanları değişir, sıfır elemanlar sıfırdan farklı olurlar. Seyrek matrisler giderek dolu matris olurlar.

- İterasyon metotlarında yuvarlama hataları direkt metotlara nazaran çok daha az sorun yaratır.

- Birkaç iterasyon adımı sonunda oldukça yaklaşık bir ara çözüm oluşur. Kabaca bir çözümün yeterli olduğu problem türlerinde birkaç iterasyon adımı yeterli olur. Direkt yöntemlerde ara çözüm bulmak mümkün değildir.

İterasyon Yöntemlerinin Dezavantajları

- Katsayılar matrisi tam dolu ise, yani seyrek değil ise, iterasyon metotları kullanmak uygun olamaz (çok fazla bellek gerektirir, çok fazla işlem yükü olur, biriken yuvarlama hataları vardır).

- İterasyonun kaç adımda sona ereceği, dolayısıyla hesap süresi önceden kestirilemez.

- Bazı özel durumlar hariç, her denklem sisteminde iterasyonun yakınsayacağı garantisi yoktur. Çözüm, yakınsayabilir yada ıraksayabilir. Aynı denklem sisteminin için iterasyon metotlarından biri yakınsarken bir diğeri ıraksayabilir. **Ancak; katsayılar matrisi kesin köşegen baskın ise JACOBI ve GAUSS-SEIDEL metodu mutlaka çözüme yakınsar**.

- Çözümün yakınsamaması durumunda iterasyon, teorik olarak, sonsuza kadar devam eder. Bu nedenle iterasyon sayısı maksimum iterasyon sayısı ile sınırlandırılmak zorundadır. Maksimum iterasyon sayısının ne olması gerektiğinin net bir cevabı yoktur. Çok küçük tutulursa doğru sonuca ulaşılamaz, çok büyük tutulursa gereksiz yere hesap süresi uzar.
- İterasyonun yakınsamaması durumunda sorunun kullanılan metottan mı yoksa denklem sisteminin yapısından mı kaynaklandığını belirlemek zordur. Örneğin denklem sistemi hatalı kurulmuş ise, katsayılar matrisinin determinantı sıfır ise çözüm yakınsamaz. İterasyon metotlarında determinant hesabı mümkün olmadığından hata kaynağı belirlenemez.

Çalışma Sorusu:

$$-2x + y + 5z = 15$$
$$4x - 8y + z = -21$$
$$4x - y + z = 7$$

lineer denklem sistemini

$x_1^0 = 1,\ x_2^0 = 2,\ x_3^0 = 2$ başlangıç

değerlerini kullanarak $\varepsilon = 10^{-3}$ hata ve

3 ondalık kullanarak Gauss-Seidel iterasyonu

ile bulunuz.

Katsayılar Matrisi köşegen baskın değildir !!!

Çözüm:

```
Yönteme başladığımız nokta= (1,000,2,000, 2,000)
1. adımda yaklaşık değer= (-1,500,2,125,15,125 )
2. adımda yaklaşık değer= (31,375,20,203,-98,297 )
3. adımda yaklaşık değer= (-243,141,-131,232,848,330 )
4. adımda yaklaşık değer= (2047,709,1132,521,-7051,315 )
5. adımda yaklaşık değer= (-17069,529,-9413,554,58871,563 )
6. adımda yaklaşık değer= (142464,625,78593,883,-491257,625 )
7. adımda yaklaşık değer= (-1188854,500,-655831,813,4099593,250 )
8. adımda yaklaşık değer= (9921060,000,5472981,500,-34211252,000 )
9. adımda yaklaşık değer= (-82791648,000,-45672228,000,285494368,000 )
10. adımda yaklaşık değer= (690899840,000,381136704,000,-2382462720,000 )
11. adımda yaklaşık değer= (-5765588480,000,-3180602112,000,19881752576,000 )
12. adımda yaklaşık değer= (48114081792,000,26542260224,000,-165914066944,000 )
13. adımda yaklaşık değer= (-401514037248,000,-221496279040,000,1384559935488,000 )
14. adımda yaklaşık değer= (3350651469824,000,1848395694080,000,-11554210709504,000 )
15. adımda yaklaşık değer= (-27961327616000,000,-15424939884544,000,96420371628032,000 )
16. adımda yaklaşık değer= (233338459652096,000,128721780473856,000,-804632058134528,000 )
17. adımda yaklaşık değer= (-1947219255099392,000,-1074188634816512,000,6714688519798784,000 )
18. adımda yaklaşık değer= (16249626512326656,000,8964149186920448,000,-56034355251773440,000 )
19. adımda yaklaşık değer= (-135603817025634300,000,-74806201308676096,000,467609066793861120,000 )
20. adımda yaklaşık değer= (1131619497610838000,000,624260852089880580,000,-3902217069633994800,000 )
21. adımda yaklaşık değer= (-9443411801363447800,000,-5209482793867804700,000,32564165511097614000,000 )
22. adımda yaklaşık değer= (78805666333496181000,000,43473352756123664000,000,-271749312577861060000,000 )
23. adımda yaklaşık değer= (-657636587474404770000,000,-362786975401621060000,000,2267759444864742200000,000 )
24. adımda yaklaşık değer= (5488004878170440300000,000,3027472228955824600000,000,-18924546720775983000000,000 )
25. adımda yaklaşık değer= (-45797631531886976000000,000,-25264383261615556000000,000,157926142865932350000000,000 )
26. adımda yaklaşık değer= (382183174541222350000000,000,210832359632452350000000,000,-1317900392575632600000000,000 )
```

Çözüme yakınsama yoktur, ıraksama vardır…

Kaynaklar

- Numerical Analysis, Richard L. Burden, Brooks/Cole Cengage Learning, Boston., 2009.
- Numerical Methods for Mathematics, Science, and Engineering, 2nd Edition, John H. Mathews, Prentice Hall International Edition, 1992.
- Nümerik Analiz, (Numerical Analysis, D. Kincaid, W. Cheney, 3rd ed.(2002)), Nuri Özalp, Elif Demirci, Gazi Kitabevi Yayınları, 2012.
- Sayısal Analiz ve Mühendislik Uygulamaları, İrfan Karagöz, Nobel Yayıncılık, 2011.
- Sayısal Çözümleme, Recep Tapramaz, Literatür yayıncılık, 2002.
- Bilgisayar Uygulamalı Sayısal Analiz Yöntemleri, Eyüp Sabri Türker, Engin Can, II. Baskı, Değişim Yayınları.