Lineer Denklem Sistemlerinin Çözümü

n bilinmeyenli n denklemden oluşan bir sistem;

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

Matrisel formda ifade edilirse;

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}}_{[A]} \cdot \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}}_{[X]} = \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}}_{[B]}$$

Burada;

A : Katsayılar matrisi,

B : Sabitler matrisi,

X : Bilinmeyenler matrisidir.

Bu denklem sisteminde çözümün olabilmesi için det(A)≠0 olmalıdır.

Lineer denklem sistemlerinin çözüm yöntemleri iki alt başlıkta toplanabilir;

1. Doğrudan Yöntemler (Ters Matris, Cramer, Gauss Eleminasyon, Gauss-Jordan Yöntemleri),
2. Sayısal Yöntemler (Jakobi, Gauss Siedell Yöntemleri)

1. Doğrudan Yöntemler

Ters Matris Yöntemi

$$[A]\cdot[X]=[B]\rightarrow[A]^{-1}\cdot[A]\cdot[X]=[A]^{-1}\cdot[B]\rightarrow[X]=[A]^{-1}\cdot[B]$$
$$[R]=[A]^{-1}\cdot[B]\rightarrow[X]=[R]$$

Örnek :

$\begin{matrix}5x_1+4x_2=14\\2x_1+3x_2=7\end{matrix}$ denklem sisteminin çözüm kümesini bulunuz.

Cramer Yöntemi

Yukarda verilen denklem sisteminin cramer yöntemi ile çözümü;

$$|A| = \Delta = \begin{bmatrix} a_{11} & a_{12} & \cdot\cdot & a_{1n} \\ a_{21} & a_{22} & \cdot\cdot & a_{2n} \\ \cdot & \cdot & \cdot\cdot & \cdot \\ \cdot & \cdot & \cdot\cdot & \cdot \\ a_{n1} & a_{n2} & \cdot\cdot & a_{nn} \end{bmatrix}$$

$$\Delta_1 = \begin{bmatrix} b_1 & a_{12} & \cdot\cdot & a_{1n} \\ b_2 & a_{22} & \cdot\cdot & a_{2n} \\ \cdot & \cdot & \cdot\cdot & \cdot \\ \cdot & \cdot & \cdot\cdot & \cdot \\ b_n & a_{n2} & \cdot\cdot & a_{nn} \end{bmatrix}$$

$$\Delta_2 = \begin{bmatrix} a_{11} & b_1 & \cdot\cdot & a_{1n} \\ a_{21} & b_2 & \cdot\cdot & a_{2n} \\ \cdot & \cdot & \cdot\cdot & \cdot \\ \cdot & \cdot & \cdot\cdot & \cdot \\ a_{n1} & b_n & \cdot\cdot & a_{nn} \end{bmatrix}$$

$$\vdots$$

$$\Delta_n = \begin{bmatrix} a_{11} & a_{12} & \cdot\cdot & b_1 \\ a_{21} & a_{22} & \cdot\cdot & b_2 \\ \cdot & \cdot & \cdot\cdot & \cdot \\ \cdot & \cdot & \cdot\cdot & \cdot \\ a_{n1} & a_{n2} & \cdot\cdot & b_n \end{bmatrix}$$

$\Longrightarrow$

$$x_1 = \frac{\Delta_1}{\Delta}$$

$$x_2 = \frac{\Delta_2}{\Delta}$$

$$\vdots$$

$$x_n = \frac{\Delta_n}{\Delta}$$

Not : Bu iki yöntem denklem sayısı 3 ve 3 ‘ten az olan sistemlere uygundur.

Örnek :

$$\begin{aligned} 5x_1 + 4x_2 &= 14 \\ 2x_1 + 3x_2 &= 7 \end{aligned}$$ denklem sisteminin çözüm kümesini bulunuz.

Gauss Eleminasyon Yöntemi

Çok eski bir yöntem olmasına rağmen bir çok popüler yazılım paketinde doğrusal denklemlerin çözüm yöntemi olarak kullanılmaktadır.

Yöntemin gerçekleştirilmesi iki aşamadan oluşur;

1. Bilinmeyenlerin elenmesi;

Ana köşegen altındaki elemanlar, elementer satır işlemleri ile sıfır yapılır.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & | & b_1 \\ a_{21} & a_{22} & a_{23} & | & b_2 \\ a_{31} & a_{32} & a_{33} & | & b_3 \end{bmatrix} \rightarrow \begin{bmatrix} a_{11} & a_{12} & a_{1n} & | & b_1 \\ 0 & a_{22}' & a_{23}' & | & b_2' \\ 0 & 0 & a_{33}' & | & b_3' \end{bmatrix}$$

2. Geriye doğru çözüm kümesinin bulunması;

$$\begin{bmatrix} a_{11} & a_{12} & a_{1n} \\ 0 & a_{22}' & a_{23}' \\ 0 & 0 & a_{33}' \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2' \\ b_3' \end{bmatrix}$$

$$x_3 = \frac{b_3'}{a_{33}'}\ ,\ x_2 = \frac{b_2' - a_{23}' \cdot x_3}{a_{22}'}\ ,\ x_1 = \frac{b_1' - a_{12} \cdot x_2 - a_{13} \cdot x_3}{a_{11}}$$

Örnek :

$x_1 + 2x_2 + x_3 = 8$
$2x_1 - x_2 + x_3 = 3$ denklem sisteminin çözüm kümesini bulunuz.
$3x_1 + 3x_2 - 2x_3 = 3$

```c
#include<stdio.h>
int main(void)
{
	void backsubs(float [][10],float [], int);
	float a[10][10],b[10],tem=0,temp=0,temp1=0,temp2=0,temp4=0,temp5=0;
	int n=0,m=0,i=0,j=0,p=0,q=0;
	printf("Kare Matrisin Boyutu :");
	scanf("%d",&n);
	for(i=0;i<n;i++)
	{
		for(j=0;j<n;j++)
		{
			printf("A[%d,%d] :",i,j); scanf("%f",&a[i][j]);
		}
	}
	printf("\nEnSabitler Matrisi\n");
	for(i=0;i<n;i++)
	{
			printf("B[%d] :",i,j); scanf("%f",&b[i]);
	}
	for(i=0;i<n;i++)
	{
		temp=a[i][i];
		if(temp<0)
		temp=temp*(-1);
		p=i;
		for(j=i+1;j<n;j++)
		{
			if(a[j][i]<0)
				tem=a[j][i]*(-1);
			else
				tem=a[j][i];
			if(temp<0)
				temp=temp*(-1);
			if(tem>temp)
			{
				p=j; temp=a[j][i];
			}
		}
		//Satir degisimleri
		for(j=0;j<n;j++)
		{
			temp1=a[i][j]; a[i][j]=a[p][j]; a[p][j]=temp1;
		}
		 temp2=b[i];
		 b[i]=b[p];
		 b[p]=temp2;
		//Kosegen haric elemanlarin sifirlanmasi
		temp4=a[i][i];
		for(q=i+1;q<n;q++)
		{
			temp5=a[q][i];
			for(j=0;j<n;j++)
			{
			a[q][j]=a[q][j]-((temp5/temp4)*a[i][j]);
			}
			b[q]=b[q]-(temp5/temp4*b[i]);
		}
	}
	backsubs(a,b,n);
	return 0;
}
```

```c
void backsubs(float a[][10],float b[], int n)
{
	int i=0,j=0;
	for(i=n-1;i>=0;i--)
	{
		for(j=n-1;j>i;j--)
		{
			b[i]=b[i]-a[i][j]*b[j];
		}
		b[i]=b[i]/a[i][i];
		printf("x%d = %f\n",i+1,b[i]);
	}
}
```

Gauss-Jordan Yöntemi

Gauss yönteminin farklı bir durumudur. Ana köşegen hariç diğer elemanlar, elementer satır işlemleri ile sıfırlanır. Dolayısıyla, çözümü bulmak için geriye doğru çözümün bulunması adımını içermez.

$$\begin{bmatrix} a_{11} & 0 & 0 \\ 0 & a_{22}' & 0 \\ 0 & 0 & a_{33}' \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1' \\ b_2' \\ b_3' \end{bmatrix} \qquad x_3 = \frac{b_3'}{a_{33}'},\ x_2 = \frac{b_2'}{a_{22}'},\ x_1 = \frac{b_1'}{a_{11}}$$

Ayrıca, bütün satırlar pivot elemanlara bölünerek normalize edilebilir.

Örnek :

$x_1 + 2x_2 + x_3 = 8$
$2x_1 - x_2 + x_3 = 3$ denklem sisteminin çözüm kümesini bulunuz.
$3x_1 + 3x_2 - 2x_3 = 3$

2. İteratif Yöntemler

Gauss-Jakobi Yöntemi

n bilinmeyenli denklem sistemi,

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

Şeklinde verilmiş olsun. Bu sistemş Gauss-Jakobi yöntemi ile çözebilmek için aşağıdaki forma dönüştürülür.

$$\frac{1}{a_{11}}\cdot\left[a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n\right] = \frac{1}{a_{11}}\cdot b_1$$
$$\frac{1}{a_{22}}\cdot\left[a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n\right] = \frac{1}{a_{22}}\cdot b_2$$
$$\vdots$$
$$\frac{1}{a_{nn}}\cdot\left[a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n\right] = \frac{1}{a_{nn}}\cdot b_n$$

$$\Longrightarrow$$

$$x_1 = \frac{b_1}{a_{11}} - \left(\frac{a_{12}}{a_{11}}x_2 + \cdots + \frac{a_{1n}}{a_{11}}x_n\right)$$
$$x_2 = \frac{b_2}{a_{22}} - \left(\frac{a_{21}}{a_{22}}x_1 + \cdots + \frac{a_{2n}}{a_{22}}x_n\right)$$
$$\vdots$$
$$x_n = \frac{b_n}{a_{nn}} - \left(\frac{a_{n1}}{a_{nn}}x_1 + \cdots + \frac{a_{nn-1}}{a_{nn}}x_{n-1}\right)$$

Dönüştürülen sistemin çözüm kümesine yakınsayabilmesi için, yakınsama koşulunu sağlaması gerekir.

$$|a_{ii}| \geq \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|$$

her satır için köşegen elemanı, diğer elemanların toplamından büyük olmalıdır. Eşit olma durumunda yakınsama çok yavaş olur.

İterasyon için;

$$x_1^{k+1} = \frac{b_1}{a_{11}} - \left(\frac{a_{12}}{a_{11}}x_2^k + \cdots + \frac{a_{1n}}{a_{11}}x_n^k\right)$$
$$x_2^{k+1} = \frac{b_2}{a_{22}} - \left(\frac{a_{21}}{a_{22}}x_1^k + \cdots + \frac{a_{2n}}{a_{22}}x_n^k\right)$$
$$\vdots$$
$$x_n^{k+1} = \frac{b_n}{a_{nn}} - \left(\frac{a_{n1}}{a_{nn}}x_1^k + \cdots + \frac{a_{nn-1}}{a_{nn}}x_{n-1}^k\right)$$

Genel iterasyon formulu;

$$x_i^{k+1} = \frac{1}{a_{ii}}\cdot\left(b_i - \sum_{\substack{j=1 \\ i\neq j}}^{n} a_{ij}x_j^k\right)$$

Matrisel formda yazılacak olursa;

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}^{k+1} = \begin{bmatrix} \frac{b_1}{a_{11}} \\ \frac{b_2}{a_{22}} \\ \vdots \\ \frac{b_n}{a_{nn}} \end{bmatrix} - \begin{bmatrix} 0 & \frac{a_{12}}{a_{11}} & \cdots & \frac{a_{1n}}{a_{11}} \\ \frac{a_{21}}{a_{22}} & 0 & \cdots & \frac{a_{2n}}{a_{22}} \\ \vdots & & & \\ \frac{a_{n1}}{a_{nn}} & \frac{a_{n2}}{a_{nn}} & \cdots & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}^{k}$$

$$[X]^{k+1} = [B'] - [A'][X]^k$$

Örnek :

$$5x_1 - 2x_2 + x_3 = 4$$
$$x_1 + 4x_2 - 2x_3 = 3$$
$$x_1 + 2x_2 + 4x_3 = 17$$

Denklem sisteminin çözüm kümesini (0,0,0) başlangıç değerlerini kullanarak bulunuz.

Gauss-Seidel Yöntemi

Bu yöntem Jakobi yönteminin geliştirilmiş halidir. Yakınsaması çok daha hızlıdır. Seidel yönteminde, k+1 iterasyonunda bulunan x_i^{k+1} sonuçları, j=i+1,...n olmak üzere x_j^{k+1} sonuçlarının bulunmasında kullanılır.

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

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

.

.

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

Genel iterasyon formulu;

$$x_i^{k+1} = \frac{b_i}{a_{ii}} - \sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^{k+1} - \sum_{j=i+1}^{n} \frac{a_{ij}}{a_{ii}} x_j^k$$

İterasyona başlanmadan yakınsama koşuluna dikkat edilmelidir.

```
#include<stdio.h>
int main(void)
{
	float a[10][10],b[10],x[10],y[10];
	int n=0,m=0,i=0,j=0;
	printf("Kare Matrisin Boyutu : ");
	scanf("%d",&n);
	for(i=0;i<n;i++)
	{
		for(j=0;j<n;j++)
		{
			printf("A[%d,%d] :",i,j);
			scanf("%f",&a[i][j]);
		}
	}
	printf("\nSabitler Matrisi\n");
	for(i=0;i<n;i++)
	{
			printf("B[%d] :",i,j);
			scanf("%f",&b[i]);
	}
	printf("Baslangic Degerleri\n");
	for(i=0;i<n;i++)
	{
		printf("x[%d]_0 :",i);
		scanf("%f",&x[i]);
	}
	printf("\nIterasyon sayisi : " );
	scanf("%d",&m);
```

```
while(m>0)
	{
		for(i=0;i<n;i++)
		{
			y[i]=(b[i]/a[i][i]);
			for(j=0;j<n;j++)
			{
			if(j==i)

			continue;
		y[i]=y[i]-((a[i][j]/a[i][i])*x[j]);
				x[i]=y[i];
			}
		printf("x%d = %f	",i+1,y[i]);
		}
		printf("\n\n");
		m--;
	}
	return 0;
}
```

Örnek :

$$5x_1 - 2x_2 + x_3 = 4$$
$$x_1 + 4x_2 - 2x_3 = 3$$
$$x_1 + 2x_2 + 4x_3 = 17$$

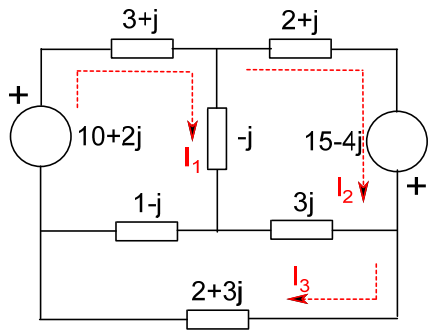
Denklem sisteminin çözüm kümesini (0,0,0) başlangıç değerlerini kullanarak bulunuz.

Örnek :

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

Denklem sisteminin çözüm kümesini (0,0,0) başlangıç değerlerini kullanarak bulunuz.

Ödev :



Yandaki devrede I_1 , I_2 ve I_3 akımlarını genelleştirilmiş matris formatı ile, her iki iterasyon yöntemini kullanarak bulunuz.