$Ax = b$ sistemi için yinelemeli(iteratif) yöntemler

Prof. Dr. Erhan Coşkun

Karadeniz Teknik Üniversitesi

Eylül 2020

Yinelemeli yöntemler

Bu bölümde

- herhangi bir başlangıç değer ile önceden bilinmeyen sayıda yinelemeli işlem sonucunda yaklaşık çözümü elde eden ve *yinelemeli(iteratif) yöntemler* olarak bilinen yöntemlerden

Yinelemeli yöntemler

Bu bölümde

- herhangi bir başlangıç değer ile önceden bilinmeyen sayıda yinelemeli işlem sonucunda yaklaşık çözümü elde eden ve *yinelemeli(iteratif) yöntemler* olarak bilinen yöntemlerden
  - *Gauss-Jacobi* ve

Yinelemeli yöntemler

Bu bölümde

- herhangi bir başlangıç değer ile önceden bilinmeyen sayıda yinelemeli işlem sonucunda yaklaşık çözümü elde eden ve *yinelemeli(iteratif) yöntemler* olarak bilinen yöntemlerden
  - *Gauss-Jacobi* ve
  - *Gauss-Seidel* yöntemlerini inceliyoruz.

Yinelemeli yöntemler

- Yinelemeli yöntemler, $A\mathbf{x} = \mathbf{b}$ denklem sisteminin çözümünü belirleme problemini $f(\mathbf{x}) = A\mathbf{x} - \mathbf{b}$ olarak tanımlanan f fonksiyonunun sıfıryerini belirleme problemine dönüştürürler.

Yinelemeli yöntemler

- Yinelemeli yöntemler, $A\mathbf{x} = \mathbf{b}$ denklem sisteminin çözümünü belirleme problemini $f(\mathbf{x}) = A\mathbf{x} - \mathbf{b}$ olarak tanımlanan f fonksiyonunun sıfıryerini belirleme problemine dönüştürürler.
- f fonksiyonunun sıfıryerini belirlemek için ise önceki bölümde incelediğimiz sabit nokta iterasyon yöntemi uygulanır.

Yinelemeli yöntemler

- Yinelemeli yöntemler, $A\mathbf{x} = \mathbf{b}$ denklem sisteminin çözümünü belirleme problemini $f(\mathbf{x}) = A\mathbf{x} - \mathbf{b}$ olarak tanımlanan f fonksiyonunun sıfıryerini belirleme problemine dönüştürürler.
- f fonksiyonunun sıfıryerini belirlemek için ise önceki bölümde incelediğimiz sabit nokta iterasyon yöntemi uygulanır.
- Bu amaçla akla gelen klasik iki yöntem Gauss-Jacobi ve Gauss-Seidel yöntemleridir. Öncelikle Gauss-Jacobi yöntemini inceleyelim:

Gauss-Jacobi yöntemi

Yöntemi 3×3 lük bir sistem üzerinde inceleyelim.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

ve

$$\mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

olmak üzere $A\mathbf{x} = \mathbf{b}$ denklem sistemini gözönüne alalım.

Gauss-Jacobi yöntemi

Matrisin köşegen üzerindeki elemanlarının sıfırdan farklı olduğunu kabul edelim. Daha açık olarak sistemi

$$\begin{aligned} a_{11}x + a_{12}y + a_{13}z &= b_1 \\ a_{21}x + a_{22}y + a_{23}z &= b_2 \\ a_{31}x + a_{32}y + a_{33}z &= b_3 \end{aligned}$$

olarak yazalım. Birinci denklemde x değişkenini, ikinciden y yi ve üçüncüden de z yi yanlız bırakarak $\mathbf{x} = g(\mathbf{x})$ biçiminde

$$\begin{aligned} x &= \frac{1}{a_{11}}(b_1 - a_{12}y - a_{13}z) \\ y &= \frac{1}{a_{22}}(b_2 - a_{21}x - a_{23}z) \\ z &= \frac{1}{a_{33}}(b_3 - a_{31}x - a_{32}y) \end{aligned}$$

sabit nokta belirleme problemini elde ederiz.

Burada $\mathbf{x}^{(1)}$ sitemin çözümü için tahmindir. Bileşen bazında yazmak gerekirse

$$\mathbf{x}^{(1)} = [x_1, y_1, z_1]^T$$

başlangıç noktası ile

$$\begin{aligned}
x_{k+1} &= \frac{1}{a_{11}}(b_1 - a_{12}y_k - a_{13}z_k) \\
y_{k+1} &= \frac{1}{a_{22}}(b_2 - a_{21}x_k - a_{23}z_k) \\
z_{k+1} &= \frac{1}{a_{33}}(b_3 - a_{31}x_k - a_{32}y_k)
\end{aligned} \tag{1}$$

$k = 1, 2, \cdots$ elde edilir.

Örnek 1.

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

denklem sistemi versilsin. Sistemin çözümü için

$$\mathbf{x}^{(1)} = [x_1, y_1, z_1] = [1\ 1\ 1]^T$$

alarak Gauss-Jacobi yöntemiyle $x^{(2)}, x^{(3)}$ *yaklaşımlarını belirleyiniz.*

Gauss-Jacobi yöntemi

Yukarıda belirtilen prosedürü takip ederek,

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

elde ederiz. O halde Gauss-Jacobi iterasyonlarını

$$\begin{aligned} x_{k+1} &= \frac{1}{3}(2 + y_k + z_k) \\ y_{k+1} &= \frac{1}{4}(-1 - x_k - z_k) \\ z_{k+1} &= \frac{1}{3}(8 - x_k + y_k) \end{aligned}$$

olarak tanımlarız.

Başlangıç tahminini kullanmak suretiyle

$$\begin{aligned} x_2 &= \frac{1}{3}(2 + y_1 + z_1) = \frac{4}{3} \\ y_2 &= \frac{1}{4}(-1 - x_1 - z_1) = -\frac{3}{4} \\ z_2 &= \frac{1}{3}(8 - x_1 + y_1) = \frac{8}{3} \end{aligned}$$

elde ederiz.
O halde $\mathbf{x}^{(2)} = [4/3 - 3/4 \ 8/3]^T$ olarak elde edilir. Virgülden sonra dört basamağa kadar yuvarlatılarak sunulan yaklaşımlar aşağıdaki gibidir.

Sonuçlandırma kriteri olarak

$$||\mathbf{x}^{(n+1)} - \mathbf{x}^{(n)}||_2 < 10^{-4}$$

kriterini kullanıyoruz.

$$\begin{array}{ccccccc} x^{(1)} & & x^{(2)} & & x^{(3)} & & x^{(17)} \\ \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, & & \begin{bmatrix} 1.3333 \\ -0.7500 \\ 2.6667 \end{bmatrix}, & & \begin{bmatrix} 1.3056 \\ -1.2500 \\ 1.9722 \end{bmatrix} & ,\cdots, & \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix} \end{array}$$

Yukarıda elde edilen yaklaşımların $[1 \ -1 \ 2]^T$ gerçek çözümüne yakınsadığına dikkat edelim.

Örnek 2.

Yukarıdaki sistemin iki satırını yerdeğiştirerek elde edilen

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

sistem için yine aynı başlangıç değeri ile $x^{(2)}, x^{(3)}, x^{(4)}$ yaklaşımlarını Gauss-Jacobi yöntemi yardımıyla hesaplayınız.

Verilen sistemi

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

olarak yazmak suretiyle,

$$\begin{aligned} x_{k+1} &= -1 - 4y_k - z_k \\ y_{k+1} &= 2 + 3x_k - z_k \\ z_{k+1} &= \frac{1}{3}(8 - x_k + y_k) \end{aligned}$$

iterasyonunu tanımlayalım.

Bu durumda Gauss-Jacobi yöntemi ile

$$\overset{\mathbf{x}^{(1)}}{\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}}, \quad \overset{\mathbf{x}^{(2)}}{\begin{bmatrix} -6 \\ 0 \\ 2.6667 \end{bmatrix}}, \quad \overset{\mathbf{x}^{(3)}}{\begin{bmatrix} -3.6667 \\ -22.6667 \\ 4.6667 \end{bmatrix}}, \quad \overset{\mathbf{x}^{(4)}}{\begin{bmatrix} 85 \\ -17.6667 \\ -3.6667 \end{bmatrix}}, \quad \cdots$$

ıraksayan iterasyon yaklaşımlarını elde ederiz.

Gauss-Jacobi yöntemi

- Örnek 1 için uygulanan yöntem yakınsak sonuç verirken aynı denklem sisteminin satırlarının yerdeğiştirilmesi ile elde edilen Örnek 2 için ıraksak bir yaklaşım elde edilmiştir. O halde akla gelen soru A matrisi üzerindeki hangi kısıtlama altında Gauss-Jacobi iterasyonları keyfi başlangıç noktası için yakınsar?

Gauss-Jacobi yöntemi

- Örnek 1 için uygulanan yöntem yakınsak sonuç verirken aynı denklem sisteminin satırlarının yerdeğiştirilmesi ile elde edilen Örnek 2 için ıraksak bir yaklaşım elde edilmiştir. O halde akla gelen soru A matrisi üzerindeki hangi kısıtlama altında Gauss-Jacobi iterasyonları keyfi başlangıç noktası için yakınsar?
- Bunun için aşağıda verilecek olan Teoremde ifade ve ispat edildiği üzere yeter şart, A matrisinin köşegen baskın olması yani her bir köşegen üzerindeki elemanın mutlak değerce aynı satırdaki diğer elemanların mutlak değerlerinin toplamından büyük olmasıdır.

- Örnek 1 için uygulanan yöntem yakınsak sonuç verirken aynı denklem sisteminin satırlarının yerdeğiştirilmesi ile elde edilen Örnek 2 için ıraksak bir yaklaşım elde edilmiştir. O halde akla gelen soru A matrisi üzerindeki hangi kısıtlama altında Gauss-Jacobi iterasyonları keyfi başlangıç noktası için yakınsar?
- Bunun için aşağıda verilecek olan Teoremde ifade ve ispat edildiği üzere yeter şart, A matrisinin köşegen baskın olması yani her bir köşegen üzerindeki elemanın mutlak değerce aynı satırdaki diğer elemanların mutlak değerlerinin toplamından büyük olmasıdır.
- Eğer A matrisi köşegen baskın bir matris ise, Gauss-Jocobi iterasyonları keyfi başlangıç noktası için yakınsar. Örnek 1 deki katsayı matrisi köşegen baskındır:

$$3 > |-1| + |-1| \ , 4 > 1 + 1, 3 > 1 + |-1|$$

dir. Fakat Örnek **??** daki matris köşen baskın değildir.
Gauss-Jacobi yöntemini vektör-matris notasyonuyla da ifade edebiliriz:
A matrisi alt üçgensel, köşegen ve üst üçgensel olmak üzere üç matrisin toplamı olarak yazılmak suretiyle

$$\begin{aligned} A &= \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 0 \\ a_{21} & 0 & 0 \\ a_{31} & a_{32} & 0 \end{bmatrix} + \begin{bmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{bmatrix} + \begin{bmatrix} 0 & a_{12} & a_{13} \\ 0 & 0 & a_{23} \\ 0 & 0 & 0 \end{bmatrix} \\ &= L + D + U \end{aligned}$$

yazılmak suretiyle

$$A\mathbf{x} = \mathbf{b}$$

veya

$$\mathbf{x} = g(\mathbf{x}) = D^{-1}\left(b - (L+U)\mathbf{x}\right)$$

olarak ifade edilen $\mathbf{x}$ sabit noktasını belirleme problemine dönüştürülebilir. Böylece Gauss-Jacobi yöntemi g nin sabit noktasını belirlemek amacıyla oluşturulan

$$\mathbf{x}^{(k+1)} = g(\mathbf{x}^{(k)}) = D^{-1}\left(b - (L+U)\mathbf{x}^{(k)}\right), k = 1, 2, \cdots \tag{2}$$

iterasyondan oluşur.
(2) ile tanımlanan vektör tabanlı Gauss-Jacobi yöntemine ait Algoritma aşağıda verilmektedir.

Gauss-Jacobi yöntemi

1. Girdi: A, b, $X0$

Gauss-Jacobi yöntemi

1. Girdi: A, b, $X0$
2. $fark = 1$, $sayac = 0$, $eps = 1e - 4$, döngü değişkenleri başlangıç değerleri

Gauss-Jacobi yöntemi

1. Girdi: $A, b, X0$
2. $fark = 1$, $sayac = 0$, $eps = 1e - 4$, döngü değişkenleri başlangıç değerleri
3. $max\_sayac = 50$, maksimum iterasyon sayısı,

Gauss-Jacobi yöntemi

1. Girdi: $A, b, X0$
2. $fark = 1$, $sayac = 0$, $eps = 1e - 4$, döngü değişkenleri başlangıç değerleri
3. $max\_sayac = 50$, maksimum iterasyon sayısı,
4. D: A nın köşegen elemanlarını içeren köşegen matris

Gauss-Jacobi yöntemi

1. Girdi: $A, b, X0$
2. $fark = 1$, $sayac = 0$, $eps = 1e - 4$, döngü değişkenleri başlangıç değerleri
3. $max\_sayac = 50$, maksimum iterasyon sayısı,
4. D: A nın köşegen elemanlarını içeren köşegen matris
5. LaU: A nın köşegen dışındaki elemanlarını içeren alt matris

Gauss-Jacobi yöntemi

1. Girdi: $A, b, X0$
2. $fark = 1$, $sayac = 0$, $eps = 1e - 4$, döngü değişkenleri başlangıç değerleri
3. $max\_sayac = 50$, maksimum iterasyon sayısı,
4. D: A nın köşegen elemanlarını içeren köşegen matris
5. LaU: A nın köşegen dışındaki elemanlarını içeren alt matris
6. $Td = D$ nin tersi

Gauss-Jacobi yöntemi

1. Girdi: $A, b, X0$
2. $fark = 1$, $sayac = 0$, $eps = 1e-4$, döngü değişkenleri başlangıç değerleri
3. $max\_sayac = 50$, maksimum iterasyon sayısı,
4. D: A nın köşegen elemanlarını içeren köşegen matris
5. LaU: A nın köşegen dışındaki elemanlarını içeren alt matris
6. $Td = D$ nin tersi
7. $fark > eps$ ve $sayac < max\_n$ olduğu sürece

Gauss-Jacobi yöntemi

1. Girdi: $A, b, X0$
2. $fark = 1$, $sayac = 0$, $eps = 1e - 4$, döngü değişkenleri başlangıç değerleri
3. $max\_sayac = 50$, maksimum iterasyon sayısı,
4. D: A nın köşegen elemanlarını içeren köşegen matris
5. LaU: A nın köşegen dışındaki elemanlarını içeren alt matris
6. $Td = D$ nin tersi
7. $fark > eps$ ve $sayac < max\_n$ olduğu sürece
    1. $X1 = TD * (b - LaU * X0)$

Gauss-Jacobi yöntemi

1. Girdi: $A, b, X0$
2. $fark = 1$, $sayac = 0$, $eps = 1e - 4$, döngü değişkenleri başlangıç değerleri
3. $max\_sayac = 50$, maksimum iterasyon sayısı,
4. D: A nın köşegen elemanlarını içeren köşegen matris
5. LaU: A nın köşegen dışındaki elemanlarını içeren alt matris
6. $Td = D$ nin tersi
7. $fark > eps$ ve $sayac < max\_n$ olduğu sürece
    1. $X1 = TD * (b - LaU * X0)$
    2. $fark = norm((X1 - X0), inf)$; maksimum hata

Gauss-Jacobi yöntemi

1. Girdi: $A, b, X0$
2. $fark = 1$, $sayac = 0$, $eps = 1e-4$, döngü değişkenleri başlangıç değerleri
3. $max\_sayac = 50$, maksimum iterasyon sayısı,
4. D: A nın köşegen elemanlarını içeren köşegen matris
5. LaU: A nın köşegen dışındaki elemanlarını içeren alt matris
6. $Td = D$ nin tersi
7. $fark > eps$ ve $sayac < max\_n$ olduğu sürece
    1. $X1 = TD * (b - LaU * X0)$
    2. $fark = norm((X1 - X0), inf)$; maksimum hata
    3. $X0 = X1;$

Gauss-Jacobi yöntemi

1. Girdi: $A, b, X0$
2. $fark = 1$, $sayac = 0$, $eps = 1e-4$, döngü değişkenleri başlangıç değerleri
3. $max\_sayac = 50$, maksimum iterasyon sayısı,
4. D: A nın köşegen elemanlarını içeren köşegen matris
5. LaU: A nın köşegen dışındaki elemanlarını içeren alt matris
6. $Td = D$ nin tersi
7. $fark > eps$ ve $sayac < max\_n$ olduğu sürece
    1. $X1 = TD * (b - LaU * X0)$
    2. $fark = norm((X1 - X0), inf)$; maksimum hata
    3. $X0 = X1$;
    4. $sayac = sayac + 1$;

Gauss-Jacobi yöntemi

1. Girdi: $A, b, X0$
2. $fark = 1$, $sayac = 0$, $eps = 1e-4$, döngü değişkenleri başlangıç değerleri
3. $max\_sayac = 50$, maksimum iterasyon sayısı,
4. D: A nın köşegen elemanlarını içeren köşegen matris
5. LaU: A nın köşegen dışındaki elemanlarını içeren alt matris
6. $Td = D$ nin tersi
7. $fark > eps$ ve $sayac < max\_n$ olduğu sürece
    1. $X1 = TD * (b - LaU * X0)$
    2. $fark = norm((X1 - X0), inf)$; maksimum hata
    3. $X0 = X1;$
    4. $sayac = sayac + 1;$
8. eğer $sayac = max\_sayac$ ise iterasyonun yakınsamadığı mesajını ilet

Gauss-Jacobi yöntemi

1. Girdi: $A, b, X0$
2. $fark = 1$, $sayac = 0$, $eps = 1e - 4$, döngü değişkenleri başlangıç değerleri
3. $max\_sayac = 50$, maksimum iterasyon sayısı,
4. D: A nın köşegen elemanlarını içeren köşegen matris
5. LaU: A nın köşegen dışındaki elemanlarını içeren alt matris
6. $Td = D$ nin tersi
7. $fark > eps$ ve $sayac < max\_n$ olduğu sürece
    1. $X1 = TD * (b - LaU * X0)$
    2. $fark = norm((X1 - X0), inf)$; maksimum hata
    3. $X0 = X1$;
    4. $sayac = sayac + 1$;
8. eğer $sayac = max\_sayac$ ise iterasyonun yakınsamadığı mesajını ilet
9. çıktı:$X1$

Gauss-Jacobi yöntemi kodu

1. $function\ X1 = gaussjacobi(A, b, X0)$

Gauss-Jacobi yöntemi kodu

1. $function\ X1 = gaussjacobi(A, b, X0)$
2. $fark = 1;$

Gauss-Jacobi yöntemi kodu

1. $function\ X1 = gaussjacobi(A, b, X0)$
2. $fark = 1;$
3. $\max sayac = 50;$

Gauss-Jacobi yöntemi kodu

1. $function\ X1 = gaussjacobi(A, b, X0)$
2. $fark = 1;$
3. $\max sayac = 50;$
4. $sayac = 0;$

Gauss-Jacobi yöntemi kodu

1. $function\ X1 = gaussjacobi(A, b, X0)$
2. $fark = 1;$
3. $\max sayac = 50;$
4. $sayac = 0;$
5. $eps = 1e - 4;$

Gauss-Jacobi yöntemi kodu

1. $function\ X1 = gaussjacobi(A, b, X0)$
2. $fark = 1;$
3. $\max sayac = 50;$
4. $sayac = 0;$
5. $eps = 1e - 4;$
6. $D = diag(diag(A));$

Gauss-Jacobi yöntemi kodu

```
function X1 = gaussjacobi(A, b, X0)
fark = 1;
max sayac = 50;
sayac = 0;
eps = 1e - 4;
D = diag(diag(A));
LaU = A - D;
```

Gauss-Jacobi yöntemi kodu

$function\ X1 = gaussjacobi(A, b, X0)$

$fark = 1;$

$\max sayac = 50;$

$sayac = 0;$

$eps = 1e - 4;$

$D = diag(diag(A));$

$LaU = A - D;$

$TD = inv(D);$

Gauss-Jacobi yöntemi kodu

1. $function\ X1 = gaussjacobi(A, b, X0)$
2. $fark = 1;$
3. $\max sayac = 50;$
4. $sayac = 0;$
5. $eps = 1e - 4;$
6. $D = diag(diag(A));$
7. $LaU = A - D;$
8. $TD = inv(D);$
9. $while\ \ fark > eps\&sayac < \max sayac$

Gauss-Jacobi yöntemi kodu

```
function X1 = gaussjacobi(A, b, X0)
fark = 1;
max sayac = 50;
sayac = 0;
eps = 1e - 4;
D = diag(diag(A));
LaU = A - D;
TD = inv(D);
while fark > eps&sayac < max sayac
    X1 = TD * (b - LaU * X0)
```

Gauss-Jacobi yöntemi kodu

```
function X1 = gaussjacobi(A, b, X0)
fark = 1;
max sayac = 50;
sayac = 0;
eps = 1e - 4;
D = diag(diag(A));
LaU = A - D;
TD = inv(D);
while fark > eps&sayac < max sayac
    X1 = TD * (b - LaU * X0)
    fark = norm(X1 - X0, inf); X0 = X1;
```

Gauss-Jacobi yöntemi kodu

```
function X1 = gaussjacobi(A, b, X0)
fark = 1;
max sayac = 50;
sayac = 0;
eps = 1e − 4;
D = diag(diag(A));
LaU = A − D;
TD = inv(D);
while fark > eps&sayac < max sayac
    X1 = TD ∗ (b − LaU ∗ X0)
    fark = norm(X1 − X0, inf); X0 = X1;
    sayac = sayac + 1;
```

Gauss-Jacobi yöntemi kodu

```
function X1 = gaussjacobi(A, b, X0)
fark = 1;
max sayac = 50;
sayac = 0;
eps = 1e - 4;
D = diag(diag(A));
LaU = A - D;
TD = inv(D);
while fark > eps&sayac < max sayac
    X1 = TD * (b - LaU * X0)
    fark = norm(X1 - X0, inf); X0 = X1;
    sayac = sayac + 1;
end
```

Gauss-Jacobi yöntemi kodu

1. $function\ X1 = gaussjacobi(A, b, X0)$
2. $fark = 1;$
3. $\max sayac = 50;$
4. $sayac = 0;$
5. $eps = 1e - 4;$
6. $D = diag(diag(A));$
7. $LaU = A - D;$
8. $TD = inv(D);$
9. $while\ \ fark > eps \& sayac < \max sayac$
10. $\quad X1 = TD * (b - LaU * X0)$
11. $\quad fark = norm(X1 - X0, inf); X0 = X1;$
12. $\quad sayac = sayac + 1;$
13. end
14. $if\ \ sayac == \max sayac\ \ error('iterasyon\ \ yakinsamadi');$

Gauss-Jacobi yöntemi kodu

```
function X1 = gaussjacobi(A, b, X0)
fark = 1;
max sayac = 50;
sayac = 0;
eps = 1e - 4;
D = diag(diag(A));
LaU = A - D;
TD = inv(D);
while fark > eps&sayac < max sayac
    X1 = TD * (b - LaU * X0)
    fark = norm(X1 - X0, inf); X0 = X1;
    sayac = sayac + 1;
end
if sayac == max sayac error('iterasyon yakinsamadi');
end
```

Gauss-Seidel yöntemi

- Gauss-Seidel yöntemi, iterasyonların yakınsayacağı varsayımı ile x_{k+1} yaklaşımının gerçek değere x_k dan daha yakın olduğunu ve benzer biçimde y_{k+1} yaklaşımının gerçek değere y_k dan daha yakın olduğunu kabul ederek ilgili iterasyonlarda bu güncel değerlerin kullanılması esasına dayınır.

Gauss-Seidel yöntemi

- Gauss-Seidel yöntemi, iterasyonların yakınsayacağı varsayımı ile x_{k+1} yaklaşımının gerçek değere x_k dan daha yakın olduğunu ve benzer biçimde y_{k+1} yaklaşımının gerçek değere y_k dan daha yakın olduğunu kabul ederek ilgili iterasyonlarda bu güncel değerlerin kullanılması esasına dayınır.
- Örnek 1 daki Gauss-Jacobi yerine

$$
\begin{aligned}
x_{k+1} &= \frac{1}{3}(2 + y_k + z_k) \\
y_{k+1} &= \frac{1}{4}(-1 - x_{k+1} - z_k) \\
z_{k+1} &= \frac{1}{3}(8 - x_{k+1} + y_{k+1})
\end{aligned}
$$

iterasyonu kullanılır.

- Buna göre Örnek 1 deki aynı başlangıç değeri ve aynı sonuçlandırma kriteri için

$$\begin{array}{ccccccc} \mathbf{x}^{(1)} & & \mathbf{x}^{(2)} & & \mathbf{x}^{(3)} & & \mathbf{x}^{(10)} \\ \left[\begin{array}{c} 1 \\ 1 \\ 1 \end{array}\right], & & \left[\begin{array}{c} 1.3333 \\ -0.8333 \\ 1.9444 \end{array}\right], & & \left[\begin{array}{c} 1.0370 \\ -0.9954 \\ 1.9892 \end{array}\right] & ,\cdots & , \left[\begin{array}{c} 1 \\ -1 \\ 2 \end{array}\right] \end{array}$$

elde ederiz.

Gauss-Seidel yöntemi

- Örneğimiz için Gauss-Seidel iterasyonları 10 adımda yakınsarken, Gauss-Jacobi 17 adımda yakınsamıştır.

Gauss-Seidel yöntemi

- Örneğimiz için Gauss-Seidel iterasyonları 10 adımda yakınsarken, Gauss-Jacobi 17 adımda yakınsamıştır.
- Gauss-Jacobi yöntemine benzer, ancak $A\mathbf{x} = \mathbf{b}$ sistemi

$$(L + D)\mathbf{x} = b - U\mathbf{x}$$

biçiminde yazılmak suretiyle

$$\mathbf{x} = g(\mathbf{x}) = (L + D)^{-1}(\mathbf{b} - U\mathbf{x})$$

olarak ifade edilen $\mathbf{x}$ sabit noktasını belirleme problemine dönüştürülebilir.

Gauss-Seidel yöntemi

- Örneğimiz için Gauss-Seidel iterasyonları 10 adımda yakınsarken, Gauss-Jacobi 17 adımda yakınsamıştır.
- Gauss-Jacobi yöntemine benzer, ancak $A\mathbf{x} = \mathbf{b}$ sistemi

$$(L + D)\mathbf{x} = b - U\mathbf{x}$$

biçiminde yazılmak suretiyle

$$\mathbf{x} = g(\mathbf{x}) = (L + D)^{-1}(\mathbf{b} - U\mathbf{x})$$

olarak ifade edilen $\mathbf{x}$ sabit noktasını belirleme problemine dönüştürülebilir.
- Böylece Gauss-Seidel yöntemi g nin sabit noktasını belirlemek amacıyla

$$\mathbf{x}^{(k+1)} = g(\mathbf{x}^{(k)}) = (L + D)^{-1}\left(b - U\mathbf{x}^{(k)}\right), k = 1, 2, \cdots \tag{3}$$

biçiminde oluşturulan iterasyondur.

Vektör tabanlı Gauss-Seidel yöntemine ait algoritma ve ilgili program Gauss-Jacobi yöntemine benzer biçimde geliştirilebilir(Alıştırma 8).

Teorem 1.

Eğer A matrisi satırca köşegen baskın, yani

$$|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}|, i = 1, 2, \cdots, m$$

ise veya sütunca köşegen baskın, yani

$$|a_{ii}| > \sum_{j=1, j\neq i}^{m} |a_{ji}|, i = 1, 2, \cdots, n$$

ise bu durumda hem Gauss-Jacobi ve hem de Gauss-Seidel yöntemleri ile oluşturulan iterasyon dizileri ilgili denklem sisteminin çözümüne yakınsarlar.

İspat A matrisinin satırca köşegen baskın olduğunu ve Gauss-Jacobi yönteminin yakınsak olduğunu gösterelim. (2) iterasyonunu

$$\mathbf{x}^{(k+1)} = C\mathbf{x}^{(k)} + F \tag{4}$$

biçiminde yazalım. Burada

$$C = -D^{-1}(L+U), F = D^{-1}b$$

dir.

$\mathbf{x}$, verilen denklem sisteminin çözümü olsun. Bu taktirde

$$A\mathbf{x} = (L + D + U)\mathbf{x} = b$$

den

$$D\mathbf{x} = b - (L + U)\mathbf{x}$$

veya

$$\mathbf{x} = D^{-1}b - D^{-1}(L + U)\mathbf{x}$$

veya

$$\mathbf{x} = C\mathbf{x} + F \tag{5}$$

elde ederiz.

$$E^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}$$

ile $k-$ıncı adımdaki iterasyon yaklaşım hatasını gösterelim. (4) ve (5) nin taraf tarafa farkını alarak, $k-$ıncı adımdaki hata vektörü için

$$\begin{aligned} E^{(k+1)} &= CE^{(k)} \\ &= C^2 E^{(k-1)} \\ &\vdots \\ &= C^k E \end{aligned}$$

elde ederiz. A köşegen baskın bir matris olduğu için $||C||_\infty < 1$ dir(Alıştırma 4). O halde

$$||E^{(k+1)}||_\infty = ||C^k E||_\infty \leq ||C^k||_\infty ||E||_\infty \leq ||C||_\infty^k ||E||_\infty$$

ve

$$\lim_{k\to\infty} ||E^{(k+1)}||_\infty = 0 \Rightarrow \lim_{k\to\infty} \mathbf{x}^{(k)} = \mathbf{x}$$

elde ederiz.

Uyarı 1.

Gauss-Jacobi veya Gauss-Seidel iterasyonun yakınsaması için köşegen baskınlık kriteri sadece yeter şarttır, fakat gerek şart değildir.

Örnek 3.

$$A = \begin{bmatrix} 4 & 1 & 3.2 \\ 2 & 5 & 2 \\ 1 & 2 & 3 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}, \mathbf{x}^{(1)} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

için Gauss-Jacobi yöntemi 100 *adımda*

$$\mathbf{x} = [0.7042, 0.1112, 0.0301]^T$$

yaklaşık çözümüne yakınsar.
Yakınsama oranı olarak aşağıda tanımlanan hata oranı

$$\frac{||A\mathbf{x}^{(k+1)} - \mathbf{b}||_2}{||A\mathbf{x}^{(k)} - \mathbf{b}||_2} \approx 0.9456$$

dir. Bu oran $I - D^{-1}A$ *nın mutlak değerce en büyük özdeğerinin mutlak değerine eşittir.*

Örnek 4.

Gauss-Seidel yöntemi ise 10 *adımda*

$$\frac{||A\mathbf{x}^{(k+1)} - \mathbf{b}||_2}{||A\mathbf{x}^{(k)} - \mathbf{b}||_2} \approx 0.3138$$

oranı ile

$$\mathbf{x} = [0.7011, 0.1087, 0.0272]^T$$

çözümüne yakınsar. Buna rağmen A matrisi satırca veya sütunca köşegen baskın değildir.

Örnek 5.

Bazen Gauss-Seidel yakınsak iterasyon ürettiği halde, Gauss-Jacobi ıraksayabilir:

$$A = \begin{bmatrix} 4 & 1 & 4 \\ 2 & 5 & 2 \\ 1 & 2 & 3 \end{bmatrix}$$

matrisi ve Örnek 3 deki b ve $x^{(1)}$ *vektörleri için Gauss-Seidel yöntemi* 11 *adımda*

$$\mathbf{x} = [0.6944, 0.1111, 0.0278]^T$$

çözümüne 0.3333 *hata oranı ile yakınsar. Ancak Gauss-Jacobi yöntemi yakınsamaz.*

Alıştırmalar

$$A = \begin{bmatrix} 3 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & -1 & 5 \end{bmatrix} \text{ matrisi ve } b = \begin{bmatrix} 3 \\ -1 \\ 7 \end{bmatrix}$$

vektörü verilsin.

1. $\mathbf{x}^{(1)} = [1\ 0\ 0]^T$ başlangıç tahmini için $\mathbf{x}^{(2)}, \mathbf{x}^{(3)}$ ve $\mathbf{x}^{(4)}$ yaklaşımlarını Gauss-Jacobi yöntemiyle hesaplayınız.

2 Soru 1 i Gauss-Seidel yöntemi ile tekrarlayınız.

3 Herhangi $A_{3\times3}$ matrisi ve $\mathbf{b}_{3\times1}$ vektörü için (1) iterasyonu ve (**??**) iterasyonunun denk olduğunu gözlemleyiniz.

4 A = L + D + U köşegen baskın bir matris ve $C = -D^{-1}(L+U)$ ise $||C||_\infty < 1$ olduğunu gösteriniz.

Alıştırmalar

5 (Bilgisayar Uygulamaları) Soru 1 de elde ettiğiniz yaklaşımları Gauss_Jacobi programı yardımıyla da elde ediniz.

6 Örnek 1,2 ve **??** de elde edilen sonuçları Gauss_Jacobi programı yardımıyla da kontrol ediniz.

7 İteratif yöntemlerde $A\mathbf{x} = \mathbf{b}$ sistemi, uygun bir tersinir B matrisi için

$$B\mathbf{x} + (A - B)\mathbf{x} = b$$

olarak yazılır ve

$$BX^{(k+1)} = (B - A)X^{(k)} + b$$

iterasyonu tanımlanır. Özel olarak

$$A = L + D + U$$

olmak üzere

- $B = D$ olması durumunda elde edilen yöntemin Gauss-Jacobi yöntemi ve
- $B = L + D$ olması durumunda elde edilen yöntemin ise Gauss-Seidel yöntemi olduğunu gözlemleyiniz.

8 Gauss_Jacobi programını düzenleyerek Gauss_Seidel programını geliştiriniz. Elde ettiğiniz program yardımıyla Örnek 3 deki yaklaşımları elde ediniz. Ayrıca Gauss-Seidel yöntemi için teorem1 i ispat ediniz.

9 Verilen bir üç köşegenli simetrik $A_{n\times n}$ matrisi için $[L, U] = lumat(A)$ komutuyla A $=LU$ ayrışımını, ayrışım elemanlarını A nın elemanları cinsinden açıkça belirlemek suretiyle hesaplayacak uygun bir algoritma ve MATLAB/Octave programı hazırlayınız.

10 (Proje) Thomas Algoritması: Aşağıdaki gibi tanımlanan A matrisi ve $\mathbf{d}$ vektörü verilsin

$$A = \begin{bmatrix} b_1 & c_1 & 0 & \cdots & 0 \\ a_2 & b_2 & c_2 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & a_{n-1} & b_{n-1} & c_{n-1} \\ 0 & 0 & \cdots & a_n & b_n \end{bmatrix}, \mathbf{d} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix}$$

$[A|\mathbf{d}]$ ekli matrisini elemanter satır işlemleri yardımıyla $[A'|\mathbf{d}']$ ekli sistemine dönüştürünüz. Burada $a_1 = 0, c_n = 0$ olmak üzere ,

Alıstırmalar

$$A' = \begin{bmatrix} b'_1 & c_1 & 0 & \cdots & 0 \\ 0 & b'_2 & c_2 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & b'_{n-1} & c_{n-1} \\ 0 & 0 & \cdots & 0 & b'_n \end{bmatrix}, \mathbf{d}' = \begin{bmatrix} d'_1 \\ d'_2 \\ \vdots \\ d'_n \end{bmatrix}$$

nın bileşenleri

$$\begin{aligned} b'_1 &= b_1 \\ d'_1 &= d_1 \\ b'_i &= b_i - \frac{a_i}{b'_{i-1}} c_{i-1}, \\ d'_i &= d_i - \frac{a_i}{b'_{i-1}} d'_{i-1}, i = 2, 3, \cdots, n \end{aligned}$$

olarak tanımlanır.

Alıştırmalar

- $A'\mathbf{x} = \mathbf{d}'$ sistemini çözerek

$$\begin{aligned} x_n &= d_n' / b_n', \\ x_i &= (d_n' - c_n x_{n+1}) / b_n', i = n-1, n-2, \cdots, 1 \end{aligned}$$

olduğunu gösteriniz.

Alıstırmalar

- $A'\mathbf{x} = \mathbf{d}'$ sistemini çözerek

$$\begin{aligned} x_n &= d'_n / b'_n, \\ x_i &= (d'_n - c_n x_{n+1}) / b'_n, \; i = n-1, n-2, \cdots, 1 \end{aligned}$$

olduğunu gösteriniz.

- Yukarıda elde ettiğiniz algoritmayı $X =$thomas(a,b,c,d) komutu ile aynı sayıda bileşene sahip a,b,c,d vektörleri için çalıştıran MATLAB/Octave programı hazırlayınız.

11 (Proje: Modifiye Thomas)A matrisini

$$\mathrm{A} = \mathrm{Ls} + \mathrm{TD} + \mathrm{Us} = \mathrm{Ls} + \begin{bmatrix} b_1 & c_1 & 0 & \cdots & 0 \\ a_2 & b_2 & c_2 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & a_{n-1} & b_{n-1} & c_{n-1} \\ 0 & 0 & \cdots & a_n & b_n \end{bmatrix} + \mathrm{Us}$$

olarak yazınız.Burada Ls üç köşenli kısmın aşağısında yer alan elemanları içeren alt üçgensel matris ve Us ise üç köşegenli kısmın yukarısında kalan kısmı içeren üst üçgensel matristir. $A\mathbf{x} = \mathbf{b}$ sistemini

$$TD\mathbf{x} + (Ls + Us)\mathbf{x} = \mathbf{b}$$

olarak yazarak

$$TDX^{(k+1)} = b - (Ls + Us)X^{(k)}, k = 0, 1, \cdots$$

iterasyonunu tanımlayalım.

Alıştırmalar

- Yöntemin her adımda Thomas algoritmasını kullanmak suretiyle üç kögenli lineer sistemi çözmesi gerektiğine dikkat edelim.

Alıstırmalar

- Yöntemin her adımda Thomas algoritmasını kullanmak suretiyle üç kögenli lineer sistemi çözmesi gerektiğine dikkat edelim.
- Yukarıda tanımlanan algoritmayı $X =$mthomas$(A, \mathbf{b}, X0)$ komutuyla uygulayan bir MATLAB/Octave programı hazırlayınız.

Alıştırmalar

- Yöntemin her adımda Thomas algoritmasını kullanmak suretiyle üç kögenli lineer sistemi çözmesi gerektiğine dikkat edelim.
- Yukarıda tanımlanan algoritmayı $X =$mthomas$(A, \mathbf{b}, X0)$ komutuyla uygulayan bir MATLAB/Octave programı hazırlayınız.
- Yöntemin yakınsama hızını Gauss-Jacobi ve Gauss-Seidel ile karşılaştırınız.

Kaynaklar

- Atkinson, K. An Introduction to Numerical Analysis, John Wiley & Sons, 1988.
- Coşkun, E. MATLAB/Octave ile Sayısal Hesaplama ve Kodlama(URL:erhancoskun.com.tr).
- Kincaid, D., Cheney, W., Numerical Analysis, Brooks/Cole, 1991.
- LAPACK, Linear Algebra Package,(URL:netlib.org)
- OCTAVE, GNU özgür yazılım(*URL:OCTAVE.sourceforge.net*).
- Press, H. W. ve ark., Numerical Recipes in C, Cambridge University Press, 1988.
- Stoer, J., Bulirsh, R., Introduction to Numerical Analysis, Springer-Verlag, 1976.
- Strang, G., Introduction to Applied Mathematics, Wellesley-Cambridge, 1986.