

**T.C.
BAHÇEŞEHİR ÜNİVERSİTESİ**

**EŞLENİK GRADYAN METODU VE
UYGULAMALARI**

Yüksek Lisans Tezi

BİRSEN ÖZKAN AYHAN

İSTANBUL, 2015

T.C.
BAHÇEŞEHİR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
UYGULAMALI MATEMATİK

EŞLENİK GRADYAN METODU VE
UYGULAMALARI

Yüksek Lisans Tezi

BİRSEN ÖZKAN AYHAN

Tez Danışmanı : Doç. Dr. Ersin ÖZUĞURLU

İSTANBUL, 2015

T.C.
BAHÇEŞEHİR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
UYGULAMALI MATEMATİK YÜKSEK LİSANS PROGRAMI

Tezin Adı : EŞLENİK GRADYAN METODU VE
UYGULAMALARI
Öğrencinin Adı Soyadı : Birsen ÖZKAN AYHAN
Tez Savunma Tarihi : 27.08.2015

Bu tezin Yüksek Lisans tezi olarak gerekli şartları yerine getirmiş olduğu Fen Bilimleri Enstitüsü tarafından onaylanmıştır.

Doç. Dr. Nafiz ARICA
Enstitü Müdürü

Bu tezin Yüksek Lisans tezi olarak gerekli şartları yerine getirmiş olduğunu onaylarım.
Prof. Dr. Canan ÇELİK KARAASLANLI
Program Koordinatörü

Bu tez tarafımızca okunmuş, nitelik ve içerik açısından bir Yüksek Lisans tezi olarak yeterli bulunmuştur.

Jüri Üyeleri

İmzalar

Tez Danışmanı

Doç. Dr. Ersin ÖZUGURLU

Üye

Prof. Dr. Mehmet AHLATÇIOĞLU

Üye

Doç. Dr. Maksat ASHYRALİYEV

.....

.....

.....

TEŐEKKÜR

Bu tezi hazırlarken yardımlarını ve desteęini benden hiç bir zaman esirgemeyen saygıdeęer hocam Doç. Dr. Ersin ÖZUĞURLU' ya sonsuz teşekkürlerimi bir borç bilirim.

Hayatım boyunca yanımda olan sevgili aileme ve sonsuz destekleri için sevgili eşim Caner'e çok teşekkür ederim.

İstanbul, 2015

Birsen ÖZKAN AYHAN



ÖZET

EŞLENİK GRADYAN METODU VE UYGULAMALARI

ÖZKAN AYHAN, Birsen

Uygulamalı Matematik

Tez Danışmanı: Doç. Dr. Ersin Özüğurlu

Ağustos, 2015, 60 sayfa

Bu çalışmada katsayılar matrisi simetrik ve pozitif tanımlı olan lineer denklem sistemlerini çözmek için bir grup ardışık metot ele alınmıştır. Bu metotlardan özellikle eşlenik gradyan metodu incelenmiştir. Eşlenik gradyan metodunu Jacobi, Gaus-Seidel, S.O.R. (successive over relaxation-ardışık aşırı rahatlamış) gibi ardışık yöntemlerle hem işlem sayısı hem de adım sayısı açısından karşılaştırılmıştır. Ardışık metotların hangi koşullarda yakınsadığı analizi de gösterilmiştir. Uygulama olarak da kimyasal reaksiyon ve malzeme-denge denklemlerini içeren bir lineer denklem sistemi ele alınıp yukarıda bahsi geçen ardışık metotlarla çözülmüştür.

Anahtar Kelimeler:

Jacobi, Eşlenik Gradyan Metodu, Koşul Sayısı, Lineer Denklem Sistemi, Pozitif Tanımlı Matris.

ABSTRACT

CONJUGATE GRADIENT METHOD AND ITS APPLICATIONS

ÖZKAN AYHAN, Birsen

Applied Mathematics Graduate Program
Supervisor: Assoc. Prof. Dr. Ersin Özüğurlu

Ağustos, 2015, 60 pages

In this thesis, some iterative methods for the solution of linear equation systems where coefficient matrix is positive definite and symmetric are investigated. Conjugate Gradient Method, which is one of these methods, is particularly discussed. Conjugate Gradient Method is compared with Jacobi, Gauss-Seidel, S.O.R (Successive Over Relaxation), etc. methods in terms of number of both operation and step. Analysis of how iterative methods converge is also investigated. A linear equation system which includes chemical reactions and material-balance equations has been solved with methods emphasized here as an application.

Keywords:

Jacobi, Conjugate Gradient Method, Condition Number, Linear Equations System, Positive Definite Matrix

İÇİNDEKİLER

TABLolar.....	vi
ŞEKİLLER.....	vii
KISALTMALAR.....	viii
1. GİRİŞ.....	1
2. LİTERATÜR ÖZETİ.....	2
3. GENEL TANIMLAR.....	3
3.1 MATRİSLERİN VE VEKTÖRLERİN NOrMLARI.....	3
3.2 LİNEER DENKLEMLERİ ÇÖZMEK İÇİN ARDIŞIK METODLAR.....	8
3.2.1 Jacobi Yöntemi.....	9
3.2.2 Gauss- Seidel Yöntemi.....	10
3.2.3 S.O.R. Yöntemi.....	11
3.3 HATA SINIRLARI VE ARDIŞIK GELİŞTİRME.....	17
4. MATERYAL VE METOT.....	23
4.1 EN DİK İNİŞ METODU (THE METHOD OF STEEPEST DESCENT)..	23
4.2 EŞLENİK GRADYAN METODU.....	27
4.3 EŞLENİK GRADYAN METODU İÇİN YAKINSAMA.....	31
4.4 ÖN KOŞULLU EŞLENİK GRADYAN METODU (PRECONDITIONED CONJUGATE GRADIENT METODU).....	34
5. BULGULAR VE TARTIŞMA.....	36
5.1 ARDIŞIK METODLARIN UYGULAMALARI.....	36
6. SONUÇ.....	45
7. KAYNAKÇA.....	46
EKLER	
Ek 1: Ön Koşullu Eşlenik Gradyan Metodu İçin MATLAB Komutu.....	50

TABLÖLAR

Tablo 5.1: Her reaktör için hacim ve hız sabiti deęerleri.....	37
Tablo 5.2: Bařlangıç deęeri 0.6 için çözüml.....	39
Tablo 5.3: Bařlangıç deęeri 100 için çözüml.....	40
Tablo 5.4: Jacobi, Gauss- Seidel, S.O.R., EGM ve ÖEGM' nin karşılařtırılması I.	43
Tablo 5.5: Jacobi, Gauss- Seidel, S.O.R., EGM ve ÖEGM' nin karşılařtırılması II.	44



ŞEKİLLER

Şekil 3.1: Değeri 1' en küçük l_2 normlu vektörler.....	4
Şekil 3.2: Değeri 1' den küçük l_∞ normlu vektörler.....	4
Şekil 3.3: l_∞ normlu vektörler.	6
Şekil 3.4: l_2 normlu vektörler.	7
Şekil 3.5: l_1 ve l_2 durumu.	18
Şekil 4.1: r^k vektörünün gösterimi.....	26
Şekil 5.1: Sürekli karıştırılan kapalı havuz reaktörleri dizisi.....	36



KISALTMALAR

DİY : Dik İniş Yöntemi

EGM : Eşlenik Gradyan Metot

MH : Mutlak Hata

ÖEGM : Ön Koşullu Eşlenik Gradyan Metot



1. GİRİŞ

Mühendislik problemlerinin yanı sıra iş, ekonomi ve sosyal bilimler problemlerinin çözümünde de sayısal yöntemler kullanılır. Bu sayısal yöntemler önce bir lineer denklem sistemine indirgenir sonra o sisteme en uygun olan çözüm yöntemiyle çözülür. Bu uygun olma koşulu; indirgenmiş olan matrisin özelliklerine ve sistemin girdilerine göre değişir.

Bu çalışmada katsayılar matrisi, simetrik ve pozitif tanımlı olan lineer denklem sistemleri ele alındı. Bu denklem sistemlerini çözmek için bir çok sayısal metot olmasına rağmen çalışmada iteratif metotlardan bazıları seçildi ve bu metotlar Eşlenik Gradyan Metot (EGM) sonuçları ile karşılaştırıldı.

Öncelikle ardışık metotlardan Jacobi, Gauss-Seidel, S.O.R. (successive over relaxation - ardışık aşırı rahatlamış) gibi metotlar açıklanıp ve bunlarla ilgili teoremler verildi; sonra da bir kimyasal reaktör problemi ele alınıp, bahsi geçen ardışık yöntemlerle çözüm yapıp bu sonuçlar EGM ile karşılaştırıldı.

2. LİTERATÜR ÖZETİ

Eşlenik Gradyan Metodu (EGM) ilk kez 1952 yılında M.R. HESTENES ve E. STIEFEL tarafından yayınlanmıştır. Bu metot mühendislik ve sağlık alanlarında çeşitli problemleri çözmek için kullanılmıştır. Ayrıca EGM 20. Yüzyılın en iyi algoritmalarından biri olduğu bilinmektedir.

Eşlenik Gradyan Metodu sadece lineer denklem sisteminin katsayılar matrisi A simetrik ve pozitif tanımlı olması durumunda kullanılır. Eğer A matrisi simetrik değilse $Ax = b$ denklem sistemi A^T (A 'nın transpozesi) ile çarpılarak $A^T Ax = A^T b$ haline getirilir, burada $A^T A$ katsayılar matrisi simetrik ve pozitif tanımlıdır. Başka bir deyişle, çoğu sistem için uygulanabilirliği olan yeni ve hızlı bir metottur.

3. GENEL TANIMLAR

3.1 MATRİSLERİN VE VEKTÖRLERİN NORMLARI

$f(\mathbf{x})=0$ biçimindeki denklemlerin köklerini bulmak için ardışık teknikler vardır. Önce, başlangıç değeri ele alınır ve bir sonraki adımda bulunan değer, bir önceki değere ne kadar yakın olduğu belirlenir. Lineer denklem sistemlerini çözmeye kullanılan ardışık metotları ele almak için öncelikle bir dizi vektörün sistemin çözümüne yakınsayıp yakınsamadığını belirlemek için n-boyutlu sütun vektörleri arasındaki mesafeyi ölçmeye ihtiyacımız var. Bu aslında lineer denklem sistemlerini doğrudan çözerken de gerekli bir ölçümdür.

\mathbb{R}^n ile bileşenleri reel sayılar olan bütün n-boyutlu sütun vektörlerinin bir kümesini temsil edelim. \mathbb{R}^n 'de uzaklığı tanımlamak için bir norm notasyonuna ihtiyacımız var.

TANIM 3.1: \mathbb{R}^n 'de vektör normu, $\| \cdot \|$, \mathbb{R}^n 'den $\mathbb{R}^+ \cup \{0\}$ 'a aşağıdaki özelliklere sahip bir fonksiyondur;

- (i) $\| \mathbf{x} \| \geq 0$, tüm $\mathbf{x} \in \mathbb{R}^n$ için,
- (ii) $\| \mathbf{x} \| = 0$ sadece ve sadece $\mathbf{x} = 0$,
- (iii) $\| a\mathbf{x} \| = |a| \| \mathbf{x} \|$ her $a \in \mathbb{R}$ ve $\mathbf{x} \in \mathbb{R}^n$
- (iv) $\| \mathbf{x} + \mathbf{y} \| \leq \| \mathbf{x} \| + \| \mathbf{y} \|$ her $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$

\mathbb{R}^n 'deki vektörler sütun vektörleri ve bir vektör bileşenleri ile gösteriliyorsa transpoze notasyonu kullanmak daha uygundur. Örnek olarak, vektör;

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$\mathbf{x} = (x_1, x_2, \dots, x_n)^t$ yazılacaktır. Sadece \mathbb{R}^n 'de belirli iki norma ihtiyaç duyulacaktır.

TANIM 3.2: $\mathbf{x} = (x_1, x_2, \dots, x_n)^t$ vektörü için l_2 ve l_∞ normları şöyle tanımlanır:

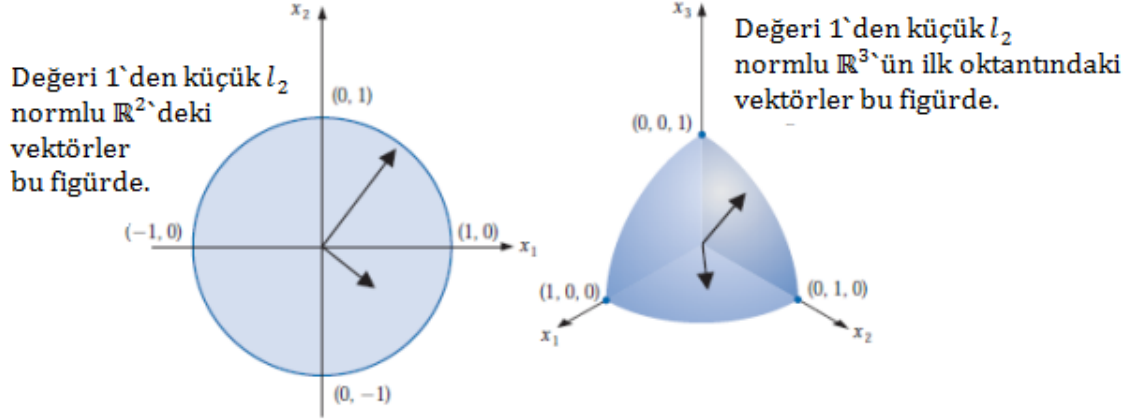
$$\| \mathbf{x} \|_2 = \left\{ \sum_{i=1}^n x_i^2 \right\}^{\frac{1}{2}} \quad \text{ve} \quad \| \mathbf{x} \|_\infty = \max_{1 \leq i \leq n} |x_i|$$

Bu normların her biri n=1 olması durumunda mutlak değerine indirgenir. Norm l_2 , \mathbf{x} vektörünün Öklid normu olarak adlandırılır çünkü \mathbf{x} 'in \mathbb{R}, \mathbb{R}^2 veya \mathbb{R}^3 'de olması durumunda orijinden uzaklığın genel nosyonunu gösterir. Örneğin,

$x = (x_1, x_2, \dots, x_n)^t$ vektörünün l_2 normu $(0,0,0)$ ve (x_1, x_2, x_3) noktalarını birleştiren düz çizginin uzunluğunu verir.

Şekil 3.1: Değeri 1' den küçük l_2 normlu vektörler.

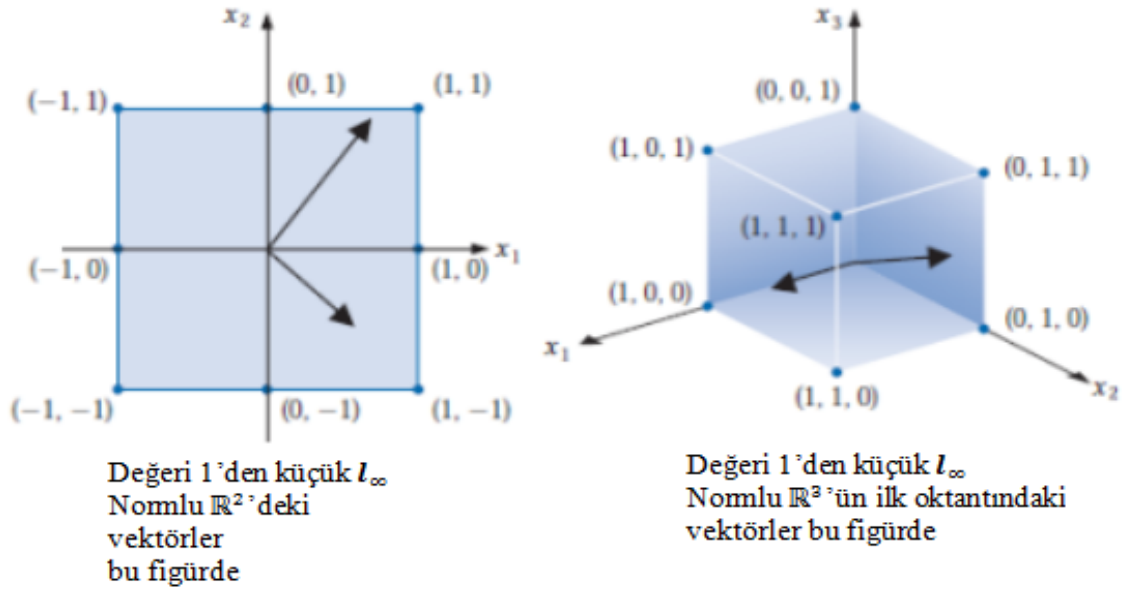
□



Kaynak: Burden ve Faires (2011, s. 433).

Şekil 3.2: Değeri 1' den küçük l_∞ normlu vektörler.

□



Kaynak: Burden ve Faires (2011, s. 433).

TEOREM 3.3: (Cauchy-Bunyakovsky-Schwarz Inequality for Sums) : \mathbb{R}^n 'de her $\mathbf{x} = (x_1, x_2, \dots, x_n)^t$ ve $\mathbf{y} = (y_1, y_2, \dots, y_n)^t$ için,

$$\mathbf{x}^t \mathbf{y} = \sum_{i=1}^n x_i y_i \leq \left\{ \sum_{i=1}^n x_i^2 \right\}^{\frac{1}{2}} \left\{ \sum_{i=1}^n y_i^2 \right\}^{\frac{1}{2}} = \|\mathbf{x}\|_2 \cdot \|\mathbf{y}\|_2$$

TANIM 3.4: Eğer $\mathbf{x} = (x_1, x_2, \dots, x_n)^t$ ve $\mathbf{y} = (y_1, y_2, \dots, y_n)^t$ \mathbb{R}^n 'de vektörler, \mathbf{x} ve \mathbf{y} arasındaki l_2 ve l_∞ mesafeleri şöyle tanımlanır:

$$\|\mathbf{x} - \mathbf{y}\|_2 = \left\{ \sum_{i=1}^n (x_i - y_i)^2 \right\}^{\frac{1}{2}} \quad \text{ve} \quad \|\mathbf{x} - \mathbf{y}\|_\infty = \max_{1 \leq i \leq n} |x_i - y_i|$$

TANIM 3.5: Verilen her $\varepsilon > 0$ için $N(\varepsilon)$ tam sayısı mevcut ve aşağıdaki eşitsizlik

$$\|\mathbf{x} - \mathbf{x}^{(k)}\| < \varepsilon, \quad \text{her } k \geq N(\varepsilon) \text{ için}$$

sağlanıyor ise bu takdirde \mathbb{R}^n 'deki vektörlerin $\{\mathbf{x}^{(k)}\}_{k=1}^\infty$ sıralısı $\|\cdot\|$ normuna göre \mathbf{x} 'e yakınsadığı söylenir.

TEOREM 3.6: $\{\mathbf{x}^{(k)}\}$ vektörü sıralısı l_∞ normuna göre \mathbb{R}^n 'de \mathbf{x} 'e yakınsar sadece ve sadece her $i = 1, 2, \dots, n$ için $\lim_{k \rightarrow \infty} x_i^{(k)} = x_i$.

TEOREM 3.7: Her $\mathbf{x} \in \mathbb{R}^n$ için,

$$\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_2 \leq \sqrt{n} \|\mathbf{x}\|_\infty$$

TANIM 3.8: $n \times n$ matris seti üzerinde bir matris normu bir reel değerli fonksiyondur, bu set üzerinde tanımlanan $\|\cdot\|$, bütün $n \times n$ boyutlu A ve B matrisleri ve bütün reel sayılar a için :

- (i) $\|A\| \geq 0$;
- (ii) $\|A\| = 0$, sadece ve sadece $A=0$ ise yani bütün girdileri 0 olan matris;
- (iii) $\|aA\| = |a| \|A\|$;
- (iv) $\|A + B\| \leq \|A\| + \|B\|$.

$n \times n$ boyutlu A ve B matrisleri, bu matris normuna göre, aralarındaki uzaklık $\|A - B\|$ dir. Matris normlarının çeşitli yollarla elde edilebilmesine rağmen, sıklıkla düşünülen normlar l_2 ve l_∞ vektör normlarının doğal sonuçlarıdır.

TEOREM 3.9: $\|\cdot\|$ \mathbb{R}^n 'de bir vektör normu ise $\|A\| = \max_{\|x\|=1} \|Ax\|$ bir matris normudur.

Bir matrisin normu, vektör normları ile tanımlanan matris normları ya da vektör normu ile ilişkilendirilen matris normu (induced) olarak adlandırılır. Bu yazıda bütün matrisler aksi belirtilmediği sürece matris normu olarak düşünülecektir.

Her $z \neq 0$, $x = z/\|z\|$ vektörü birim vektördür. Buradan,

$$\max_{\|x\|=1} \|Ax\| = \max_{z \neq 0} \left\| A \left(\frac{z}{\|z\|} \right) \right\| = \max_{z \neq 0} \frac{\|Az\|}{\|z\|}$$

ve alternatif olarak şöyle yazılabilir:

$$\|A\| = \max_{z \neq 0} \frac{\|Az\|}{\|z\|}$$

SONUÇ 3.10: Her $z \neq 0$ vektörü, A matrisi ve herhangi bir norm için $\|\cdot\|$, $\|Az\| \leq \|A\| \cdot \|z\|$ bulunur. Norm altında bir matrise verilen ölçüm, bu norma göre matrisin birim vektörü nasıl genişlettiğini gösterir. Maksimum uzama matrisin normudur. Burada düşünülecek matris normları şu formlara sahiptir;

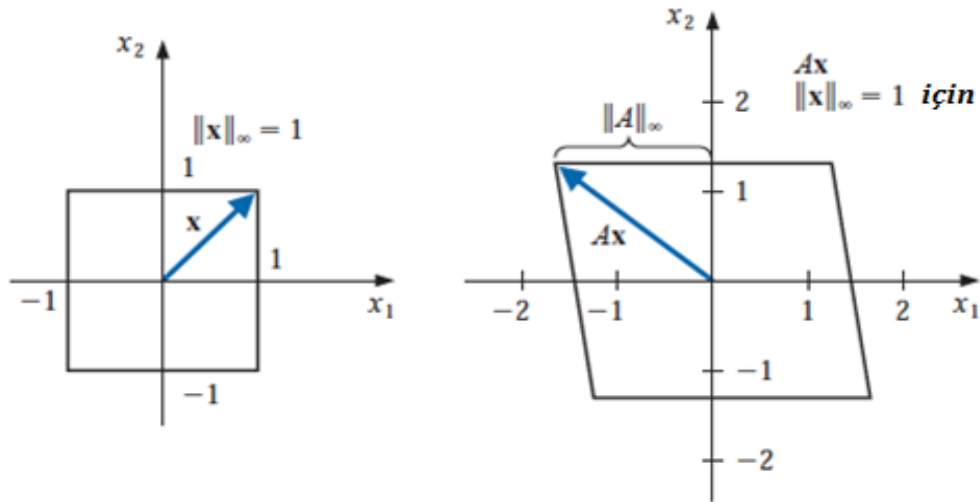
$$\|A\|_{\infty} = \max_{\|x\|_{\infty}=1} \|Ax\|_{\infty} \quad l_{\infty} \text{ normu}$$

$$\|A\|_2 = \max_{\|x\|_2=1} \|Ax\|_2 \quad l_2 \text{ normu}$$

$n = 2$ için bu normların gösterimi şekil 3.3 ve 3.4'te verilmiştir.

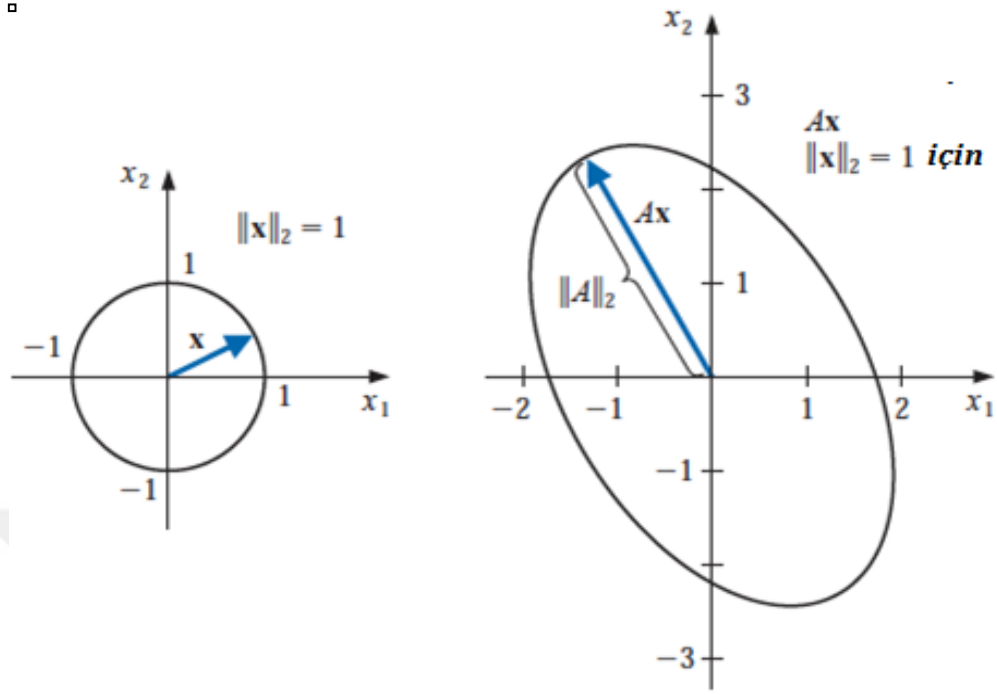
Şekil 3.3: l_{∞} normlu vektörler.

□



Kaynak: Burden ve Faires (2011, s. 439).

Şekil 3.4: l_2 normlu vektörler.



Kaynak: Burden ve Faires (2011, s. 439).

Bir matrisin l_∞ normu matrisin girdilerinden kolayca hesaplanabilir.

TEOREM 3.11: Eğer $A = (a_{ij})$ bir $n \times n$ matris ise

$$\|A\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}|$$

TANIM 3.12: Bir A matrisinin saçılım yarıçapı $\rho(A)$ şöyle tanımlanır:

$$\rho(A) = \max_{1 \leq i \leq n} |\lambda_i|, \quad \lambda_i, A \text{ vektörünün özdeğeridir}$$

($\lambda = \alpha + \beta i$ için, $|\lambda| = (\alpha^2 + \beta^2)^{\frac{1}{2}}$).

TEOREM 3.13: Eğer A bir $n \times n$ matris ise

(i) $\|A\|_2 = [\rho(A^t A)]^{1/2}$

(ii) $\rho(A) \leq \|A\|$, herhangi bir norm $\|\cdot\|$ için

Bu teoremin ispatı için Ortega (1972, s. 21).

Ardışık matris tekniklerini çalışmada matrisin kuvvetlerinin küçüldüğünü bilmek özellikle önemlidir (matrisin bütün elemanlarının sifira yaklaştığı zaman). Bu tip matrisleri yakınsak olarak adlandıracağız.

TANIM 3.14: $n \times n$ A matrisi yakınsak matris olarak tanımlanır eğer

$$\lim_{k \rightarrow \infty} (A^k)_{ij} = 0, \quad \text{her } i = 1, 2, \dots, n \text{ ve } j = 1, 2, \dots, n \text{ için.}$$

TEOREM 3.15: Aşağıdaki ifadeler denktir:

- (i) A yakınsak bir matristir.
- (ii) $\lim_{n \rightarrow \infty} \|A^n\| = 0$, bazı normlar için.
- (iii) $\lim_{n \rightarrow \infty} \|A^n\| = 0$, bütün normlar için.
- (iv) $\rho(A) < 1$
- (v) $\lim_{n \rightarrow \infty} A^n x = 0$, her x için.

Bu teoremin ispatı için Issacson ve Keller (1966, s. 14).

3.2 LİNEER DENKLEMLERİ ÇÖZMEK İÇİN ARDIŞIK METODLAR

Bu bölümde Jacobi ve Gauss-Seidel gibi 18.yy' ın sonlarına kadar dayanan klasik ardışık metotları ele alacağız. Küçük boyuttaki lineer denklem sistemlerini çözmek için ardışık metotlar tercih edilmez. Bunlar yerine doğrudan metot olan Gauss yok etme metodu kullanılır. Sıfırı çok olan büyük sistemler için ardışık teknikleri kullanmak hem bilgisayar belleği hem de işlem sayısı açısından daha verimlidir. Bu tip sistemler genellikle kısmi diferansiyel denklemlerin ve sınır değer problemlerin nümerik çözümünde, ayrıca devre analizinde sıkça karşımıza çıkar.

$n \times n$ lineer denklem sistemi $Ax = b$ yi çözmek için bir ardışık teknik başlangıç yaklaşım vektörü $x^{(0)}$ ile başlar ve x e yakınsayan bir dizi $(x^{(k)})_{k=0}^{\infty}$ vektörleri üretilir. Ardışık teknikler $Ax = b$ sistemini $x = Tx + c$ sistemine dönüştüren bir işlev içerir, burada T sabit bir matris ve c sabit bir vektördür. Başlangıç vektörü $x^{(0)}$ seçildikten sonra yaklaşık çözüm vektörleri dizisi aşağıdaki formülle oluşturulur:

$$x^{(k)} = Tx^{(k-1)} + c, \quad \text{her } k \geq 1 \text{ için}$$

Bu sonuç, bilinen sabit-nokta (fixed-point) iterasyonunu anımsatır.

3.2.1 Jacobi Yöntemi

$A\mathbf{x} = \mathbf{b}$ lineer denklem sistemini çözmek için kullanılan en basit yöntem Jacobi Metodudur. Bu metodun basit oluşu hem iyidir hem de kötü. İyidir çünkü, anlama açısından diğerlerine nazaran basittir ve ardışık metotlar için iyi bir başlangıçtır. Kötüdür çünkü, pratikte (gelişmiş paralel hesaplama ile tekrar düşünülmüş olsa da) tipik olarak kullanılmaz. Halen, daha karmaşık ardışık yöntemlerin anlaşılmasında başlangıç noktası olarak iyidir.

Öncelikle $A\mathbf{x} = \mathbf{b}$ denklem sistemini daha açık haliyle yazalım;

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \dots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \dots & + & a_{2n}x_n & = & b_2 \\ \vdots & & \vdots & & \ddots & & \vdots & & \vdots \\ a_{n1}x_1 & + & a_{n2}x_2 & + & \dots & + & a_{nn}x_n & = & b_n \end{array}$$

x için tahmini vektörü şöyle alalım;

$$\mathbf{x}^{(k)} = (x_1^{(k)}, x_2^{(k)}, x_3^{(k)}, \dots, x_n^{(k)}).$$

Jacobi metodunun stratejisi, yeni değeri $x_1^{(k+1)}$ bulmak için, ilk eşitliği ve şimdiki değerleri, $x_2^{(k)}, x_3^{(k)}, \dots, x_n^{(k)}$ kullanmaktır ve benzer olarak, i . eşitliği ve diğer değişkenlerin eski değerlerini kullanarak yeni değer $x_i^{(k)}$, y_i bulmaktır. Şöyle ki; verilen şimdiki $\mathbf{x}^{(k)} = (x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)})$ değerlerini kullanarak,

$$\begin{array}{ccccccccc} a_{11}x_1^{(k+1)} & + & a_{12}x_2^{(k)} & + & \dots & + & a_{1n}x_n^{(k)} & = & b_1 \\ a_{21}x_1^{(k)} & + & a_{22}x_2^{(k+1)} & + & \dots & + & a_{2n}x_n^{(k)} & = & b_2 \\ \vdots & & \vdots & & \ddots & & \vdots & & \vdots \\ a_{n1}x_1^{(k)} & + & a_{n2}x_2^{(k)} & + & \dots & + & a_{nn}x_n^{(k+1)} & = & b_n \end{array}$$

sistemi elde edilir ve bu sistemden de,

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

denklemlerine ulaşılır.

İçinde,

$$\mathbf{x}^{(k+1)} = (x_1^{(k+1)}, x_2^{(k+1)}, \dots, x_n^{(k+1)})$$

için çözerek yeni değerleri buluruz.

Bu sistem şöyle de yazılabilir;

$$\begin{bmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & a_{nn} \end{bmatrix} \begin{bmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ \vdots \\ x_n^{(k+1)} \end{bmatrix} + \begin{bmatrix} 0 & a_{12} & \dots & a_{1n} \\ a_{21} & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{(n-1)n} \\ a_{n1} & \dots & a_{nn-1} & 0 \end{bmatrix} \begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ \vdots \\ x_n^{(k)} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

alt indis i , $x_i^{(k)}$ 'deki,

$$\mathbf{x}^{(k)} = (x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)})$$

vektörün i . elemanını gösterir ve üst indis k ise kaçınıcı iterasyon olduğunu gösterir.

Eğer, A matrisini sırasıyla D , L ve U olarak köşgensel, alt üçgensel ve üst üçgensel için yazarsak;

$$D = \begin{bmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & a_{nn} \end{bmatrix}, L = \begin{bmatrix} 0 & 0 & \dots & 0 \\ a_{21} & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ a_{n1} & \dots & a_{nn-1} & 0 \end{bmatrix}$$

$$U = \begin{bmatrix} 0 & a_{12} & \dots & a_{1n} \\ 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{n-1n} \\ 0 & \dots & 0 & 0 \end{bmatrix}$$

elde ederiz. Böylece Jacobi Metodu matris-vektör notasyonu da şöyle yazılabilir;

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

öyleyse,

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

olur. Bu da $\mathbf{x} = T\mathbf{x} + \mathbf{c}$ formunda yazılabilir; burada $T = D^{-1}(-L - U)$ ve $\mathbf{c} = D^{-1}\mathbf{b}$ dir.

3.2.2 Gauss- Seidel Yöntemi

Jacobi Metodunu bir adım daha ileri götürelim. $\mathbf{x} = (x_1, x_2, \dots, x_n)$ gerçek çözüm olduğunda, eğer $x_1^{(k+1)}$, x_1 'in gerçek değerine $x_1^{(k)}$ 'dan daha iyi bir tahmin ise, o zaman, $x_2^{(k+1)}, \dots, x_n^{(k+1)}$ 'yi bulmakta kullanmak için, (eski değer $x_1^{(k)}$ 'dan ziyade) yeni $x_1^{(k+1)}$ değerini bulduğumuzda daha anlamlı olurdu. Öyleyse, $x_1^{(k+1)}$, Jacobi Metodundaki gibi bulunur, ama $x_2^{(k+1)}$ bulunurken, $x_1^{(k)}$ 'nin eski değeri ve $x_3^{(k)}, \dots, x_n^{(k)}$ eski

değerlerini kullanmak yerine $x_1^{(k+1)}$ ve $x_3^{(k)}, \dots, x_n^{(k)}$ eski değerlerini kullanırız ve benzer şekilde $x_3^{(k+1)}, \dots, x_n^{(k+1)}$ için de uygulanır.

$$\begin{array}{ccccccccc} a_{11}x_1^{(k+1)} & + & a_{12}x_2^{(k)} & + & \dots & + & a_{1n}x_n^{(k)} & = & b_1 \\ a_{21}x_1^{(k+1)} & + & a_{22}x_2^{(k+1)} & + & \dots & + & a_{2n}x_n^{(k)} & = & b_2 \\ \vdots & & \vdots & & \ddots & & \vdots & & \vdots \\ a_{n1}x_1^{(k+1)} & + & a_{n2}x_2^{(k+1)} & + & \dots & + & a_{nn}x_n^{(k+1)} & = & b_n \end{array}$$

Lineer denklem sisteminden

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

yazılabilir.

Bunu aynı zamanda şöyle de yazabiliriz:

$$\begin{bmatrix} a_{11} & 0 & \dots & 0 \\ a_{21} & a_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ a_{n1} & \dots & a_{n(n-1)} & a_{nn} \end{bmatrix} \begin{bmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ \vdots \\ x_n^{(k+1)} \end{bmatrix} + \begin{bmatrix} 0 & a_{12} & \dots & a_{1n} \\ 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{(n-1)n} \\ 0 & \dots & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ \vdots \\ x_n^{(k)} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Yani,

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

buradan

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

denkleme ulaşılır. Bu formülde $\mathbf{x} = T\mathbf{x} + \mathbf{c}$ formunda yazılabilir; burada

$T = -(L + D)^{-1}U$ ve $\mathbf{c} = (L + D)^{-1}\mathbf{b}$ dir.

3.2.3 S.O.R. Yöntemi

Üçüncü bir yöntem olan S.O.R Metodu, Gauss-Seidel Metodunun geliştirilmiş ve geliştirilmiş bir versiyonudur.

$\mathbf{x}^{(k)}$, dan $\mathbf{x}^{(k+1)}$ 'i bulmakta kullanılan her ardışık yöntem için özel bir yönde $\mathbf{x}^{(k)}$, dan $\mathbf{x}^{(k+1)}$ 'e kesin bir miktar kaydırırız. Bu yön, $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + (\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})$ olduğu için, $\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$ vektörüdür. Farz edelim ki, $\mathbf{x}^{(k)}$, dan $\mathbf{x}^{(k+1)}$ 'e olan yön bizi \mathbf{x} 'in

gerçek çözümüne daha fazla yaklaşıyorsa o zaman, $\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$ yönünde hareket etmek daha anlamlıdır.

Şimdi Gauss-Seidel metodundan nasıl S.O.R metoduna geçilir ona bakalım. Öncelikle Gauss-Seidel metodunu bir daha yazalım:

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

buradan

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

elde edilir. Bu denklemin her tarafından $\mathbf{x}^{(k)}$ çıkaralım:

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

Şimdi Gauss-Seidel düzeltmesi olarak $(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})_{GS}$ düşünelim. Eğer ki standart Gauss-Seidel düzeltmesinin ötesine giderse, yukarıda tavsiye edildiği üzere, gerçek çözüme çözüm tahminleri sıralılarının $\mathbf{x}^{(k)} \rightarrow \mathbf{x}$ yakınsaması sıkça daha hızlıdır. SOR metodu prensibi şu iterasyonla başlar,

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \omega(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})_{GS}$$

daha önce bulduğumuz,

$$(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})_{GS} = D^{-1}[b - L\mathbf{x}^{(k+1)} - D\mathbf{x}^{(k)} - U\mathbf{x}^{(k)}]$$

burada genel olarak $1 < \omega < 2$. Farkına varalım ki eğer $\omega = 1$ ise bu, Gauss-Seidel Metodudur.

Detaylı yazarsak, SOR Metodu,

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

her iki tarafı D matrisi ile çarpabiliriz ve her iki tarafı ω 'ya bölüp yeniden yazarsak,

$$\frac{1}{\omega} D\mathbf{x}^{(k+1)} = \frac{1}{\omega} D\mathbf{x}^{(k)} + [b - L\mathbf{x}^{(k+1)} - D\mathbf{x}^{(k)} - U\mathbf{x}^{(k)}]$$

sol tarafta $\mathbf{x}^{(k+1)}$ terimlerini toplarsak,

$$\begin{aligned} \left(L + \frac{1}{\omega} D\right) \mathbf{x}^{(k+1)} &= \frac{1}{\omega} D\mathbf{x}^{(k)} + [b - D\mathbf{x}^{(k)} - U\mathbf{x}^{(k)}] \\ &= \left(\frac{1}{\omega} D - D - U\right) \mathbf{x}^{(k)} + b \end{aligned}$$

$\mathbf{x}^{(k+1)}$ için çözdüğümüzde şunu elde ederiz,

$$\mathbf{x}^{(k+1)} = \left(L + \frac{1}{\omega} D\right)^{-1} \left[\left(\frac{1}{\omega} D - D - U\right) \mathbf{x}^{(k)} + b\right]$$

Farkına varalım ki SOR metodu aynı zamanda $\mathbf{x} = T\mathbf{x} + \mathbf{c}$ formundadır. SOR Metodu yakınsamasını hesaplayana T matrisinin iterasyonu,

$$\left(L + \frac{1}{\omega}D\right)^{-1} \left(\frac{1}{\omega}D - D - U\right)$$

öyleyse optimum yakınsama, ω 'nın minimize edecek bir değerinin seçilmesiyle olur,

$$\left\| \left(L + \frac{1}{\omega}D\right)^{-1} \left(\frac{1}{\omega}D - D - U\right) \right\|$$

Pratikte bu iterasyonları hesaplamak için bilgisayar kullanırız.

Genel iterasyon tekniklerinin yakınsaklığını çalışmak için aşağıdaki formülü ele alacağız:

$$\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}, \quad \text{her } k \geq 1 \text{ için}$$

ve $\mathbf{x}^{(0)}$ keyfi başlangıç vektörü olmak üzere.

YARDIMCI TEOREM 3.16: Eğer saçılım yarıçapı $\rho(T) < 1$ 'i sağlıyorsa o zaman $(I - T)^{-1}$ mevcuttur ve

$$(I - T)^{-1} = I + T + T^2 + \dots = \sum_{j=0}^{\infty} T^j$$

olur. Çünkü $(I - T)\mathbf{x} = (1 - \lambda)\mathbf{x}$ olduğunda $T\mathbf{x} = \lambda\mathbf{x}$ tam olarak doğrudur.

$(I - T)$ 'nin özdeğeri tam olarak $(1 - \lambda)$ olduğunda T 'nin özdeğeri λ olur. Ancak $|\lambda| \leq \rho(T) < 1$ ise, $\lambda = 1$ T 'nin bir özdeğeri değildir, ve 0, $I - T$ 'nin özdeğeri olamaz.

Böylece, $(I - T)^{-1}$ mevcut olur.

$S_m = I + T + T^2 + \dots + T^m$ olsun.

$$(I - T)S_m = (I + T + T^2 + \dots + T^m) - (T + T^2 + \dots + T^{m+1}) = I - T^{m+1},$$

ve T yakınsak olduğundan,

$$\lim_{m \rightarrow \infty} (I - T)S_m = \lim_{m \rightarrow \infty} (I - T^{m+1}) = I.$$

Böylece,

$$(I - T)^{-1} = \lim_{m \rightarrow \infty} S_m = I + T + T^2 + \dots = \sum_{j=0}^{\infty} T^j \text{ olur.}$$

TEOREM 3.17: Her $\mathbf{x}^{(0)} \in \mathbb{R}^n$ için, $(\mathbf{x}^{(k)})_{k=0}^{\infty}$ şöyle tanımlanır:

$$\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}, \quad \text{her } k \geq 1 \text{ için}$$

sadece ve sadece $\rho(T) < 1$ olduğunda $\mathbf{x} = T\mathbf{x} + \mathbf{c}$ tek çözümüne yakınsar.

SONUÇ 3.18: Her matris normu için $\|T\| < 1$ ve c verilen bir vektör ise,

$$\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + c$$

ile tanımlı $(\mathbf{x}^{(k)})_{k=0}^{\infty}$ kümesi, her $\mathbf{x}^{(0)} \in \mathbb{R}^n$ için, $\mathbf{x} = T\mathbf{x} + c$ ile $\mathbf{x} \in \mathbb{R}^n$ 'de bir vektöre yakınsar ve aşağıdaki hata sınırları geçerli olur:

- (i) $\|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \|T\|^k \|\mathbf{x}^{(0)} - \mathbf{x}\|$
- (ii) $\|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \frac{\|T\|^k}{1-\|T\|} \|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\|$

TEOREM 3.19: Eğer A köşegenel tam baskın (strictly diagonal dominant) bir matris ise, $\mathbf{x}^{(0)}$ 'ın herhangi bir çözümü için, Jacobi ve Gauss-Siedel metodlarının her ikisi de,

$A\mathbf{x} = \mathbf{b}$ 'ın tek çözümüne yakınsayan $(\mathbf{x}^{(k)})_{k=0}^{\infty}$ kümesini verir.

(ispat için Young (1971, s. 120-127)).

TEOREM 3.20 (Stein-Rosenberg): Eğer $a_{ij} \leq 0$ ise, her $i \neq j$ ve $a_{ii} > 0$ için, her $i = 1, 2, \dots, n$ için, aşağıdaki ifadelerden sadece ve sadece bir tanesi geçerlidir:

- (i) $0 \leq \rho(T_g) < \rho(T_j) < 1$
- (ii) $1 < \rho(T_j) < \rho(T_g)$
- (iii) $\rho(T_j) = \rho(T_g) = 0$
- (iv) $\rho(T_j) = \rho(T_g) = 1$

TANIM 3.21: $\tilde{\mathbf{x}} \in \mathbb{R}^n$ 'in, $A\mathbf{x} = \mathbf{b}$ ile tanımlanan lineer sisteminin çözümüne bir yaklaşım olduğunu varsayalım. $\tilde{\mathbf{x}}$ için bu sisteme göre kalıntı vektörü $\mathbf{r} = \mathbf{b} - A\tilde{\mathbf{x}}$ olur. Jacobi ve Gauss-Seidel metodlarındaki prosedürler gibi, kalıntı vektörü, çözüm vektörüne yaklaşık bir bileşenin her hesaplamasıyla ilişkilidir. Asıl amaç, kalıntı vektörünün sıfıra yakınsamasına neden olacak bir tahmin dizisi oluşturmaktır. Şu varsayımda bulunalım:

$$\mathbf{r}_i^{(k)} = (\mathbf{r}_{1i}^{(k)}, \mathbf{r}_{2i}^{(k)}, \dots, \mathbf{r}_{ni}^{(k)})^t$$

tahmini çözüm vektörü, aşağıda tanımlı $\mathbf{x}_i^{(k)}$ 'ya karşılık gelen Gauss-Seidel metodu için kalıntı vektörü gösterir.

$$\mathbf{x}_i^{(k)} = (\mathbf{x}_1^{(k)}, \mathbf{x}_2^{(k)}, \dots, \mathbf{x}_{i-1}^{(k)}, \mathbf{x}_i^{(k-1)}, \dots, \mathbf{x}_n^{(k-1)})^t$$

$r_i^{(k)}$ 'nin m . bileşeni,

$$\mathbf{r}_{mi}^{(k)} = \mathbf{b}_m - \sum_{j=1}^{i-1} a_{mj} \mathbf{x}_j^{(k)} - \sum_{j=i}^n a_{mj} \mathbf{x}_j^{(k-1)} \quad (3.1)$$

veya eşit olarak,

$$\mathbf{r}_{mi}^{(k)} = \mathbf{b}_m - \sum_{j=1}^{i-1} a_{mj} \mathbf{x}_j^{(k)} - \sum_{j=i+1}^n a_{mj} \mathbf{x}_j^{(k-1)} - a_{mi} \mathbf{x}_i^{(k-1)}$$

her $m = 1, 2, \dots, n$ için.

Özel olarak, $\mathbf{r}_i^{(k)}$ 'nin i . bileşeni,

$$\mathbf{r}_{ii}^{(k)} = \mathbf{b}_i - \sum_{j=1}^{i-1} a_{ij} \mathbf{x}_j^{(k)} - \sum_{j=i+1}^n a_{ij} \mathbf{x}_j^{(k-1)} - a_{ii} \mathbf{x}_i^{(k-1)}$$

öyleyse,

$$a_{ii} \mathbf{x}_i^{(k-1)} + \mathbf{r}_{ii}^{(k)} = \mathbf{b}_i - \sum_{j=1}^{i-1} a_{ij} \mathbf{x}_j^{(k)} - \sum_{j=i+1}^n a_{ij} \mathbf{x}_j^{(k-1)} \quad (3.2)$$

olur. Hatırlayalım, ancak, Gauss-Seidel metodunda, $\mathbf{x}_i^{(k)}$,

$$\mathbf{x}_i^{(k)} = \frac{1}{a_{ii}} \left[\mathbf{b}_i - \sum_{j=1}^{i-1} a_{ij} \mathbf{x}_j^{(k)} - \sum_{j=i+1}^n a_{ij} \mathbf{x}_j^{(k-1)} \right] \quad (3.3)$$

böylece 3.2 şöyle yazılabilir,

$$a_{ii} \mathbf{x}_i^{(k-1)} + \mathbf{r}_{ii}^{(k)} = a_{ii} \mathbf{x}_i^{(k)}.$$

Sonuç olarak, Gauss-Seidel metodu aşağıdaki eşitliği sağlaması için $\mathbf{x}_i^{(k)}$ 'yi seçerek karakterize edilebilir:

$$\mathbf{x}_i^{(k)} = \mathbf{x}_i^{(k-1)} + \frac{\mathbf{r}_{ii}^{(k)}}{a_{ii}}. \quad (3.4)$$

Kalıntı vektörleri ve Gauss-Seidel metotları arasında bir başka bağlantı kurabiliriz.

$\mathbf{x}_{i+1}^{(k)} = (\mathbf{x}_1^{(k)}, \dots, \mathbf{x}_i^{(k)}, \mathbf{x}_{i+1}^{(k-1)}, \dots, \mathbf{x}_n^{(k-1)})^t$ vektörü ile ilişkili $\mathbf{r}_{i+1}^{(k)}$ kalıntı vektörünü

düşünelim. 3.1 kullanılarak $\mathbf{r}_{i+1}^{(k)}$ 'in i . bileşeni,

$$\mathbf{r}_{i,i+1}^{(k)} = \mathbf{b}_i - \sum_{j=1}^i a_{ij} \mathbf{x}_j^{(k)} - \sum_{j=i+1}^n a_{ij} \mathbf{x}_j^{(k-1)}$$

$$= \mathbf{b}_i - \sum_{j=1}^{i-1} a_{ij} \mathbf{x}_j^{(k)} - \sum_{j=i+1}^n a_{ij} \mathbf{x}_j^{(k-1)} - a_{ii} \mathbf{x}_i^{(k)}.$$

$\mathbf{x}_i^{(k)}$, nin tanımlandığı denklem 3.3'teki mantıkla, $\mathbf{r}_{i,i+1}^{(k)} = 0$. Bir anlamda Gauss-Seidel tekniği, $\mathbf{r}_{i+1}^{(k)}$ 'in i . bileşeni sıfır olacak şekilde $\mathbf{x}_{i+1}^{(k)}$ 'yi seçerek karakterize edilir.

Kalıntı vektörün bir koordinatı sıfır olacak şekilde $\mathbf{x}_{i+1}^{(k)}$ 'yi seçmek, $\mathbf{x}_{i+1}^{(k)}$ vektörünün normunu azaltmak için çok da verimli bir yöntem değildir. Denklem 3.4'de verilen Gauss-Seidel metodunu modifiye edersek:

$$\mathbf{x}_i^{(k)} = \mathbf{x}_i^{(k-1)} + \omega \frac{\mathbf{r}_{ii}^{(k)}}{a_{ii}}. \quad (3.5)$$

ω 'nın belirli pozitif seçimleri için kalıntı vektörün normunu azaltabiliriz ve daha da hızlı yakınsama elde ederiz.

Denklem 3.5' i de kapsayan metotlar rahatlatma metotları (relaxation methods) olarak adlandırılır. ω 'nın $0 < \omega < 1$ aralığı için, bu prosedürlere alt-rahatlatma metotları (under-relaxation) denir. $1 < \omega$ aralığı ile ilgileneceğiz ve buna da üst-rahatlatma metotları (over-relaxation methods) denir. Bu yöntemler, Gauss-Seidel tekniğiyle yakınsatılan sistemlerin yakınsamasını hızlandırmak için kullanılırlar. Bu metotlar S.O.R. (Art arda aşırı rahatlamış) olarak kısaltılırlar ve bazı kısmi diferansiyel eşitliklerin nümerik çözümlerinin olduğu lineer sistemlerin çözümlerinde özel olarak kullanılırdılar.

SOR metodunun avantajlarından bahsetmeden önce hesaplama amaçlı, denklem 3.2' den 3.5' e formüle edebiliriz,

$$\mathbf{x}_i^{(k)} = (1 - \omega) \mathbf{x}_i^{(k-1)} + \frac{\omega}{a_{ii}} \left[\mathbf{b}_i - \sum_{j=1}^{i-1} a_{ij} \mathbf{x}_j^{(k)} - \sum_{j=i+1}^n a_{ij} \mathbf{x}_j^{(k-1)} \right]$$

S.O.R. metodunun matris formunu hesaplamak için yukarıdakini tekrar yazarsak,

$$a_{ii}^{(k)} + \omega \sum_{j=1}^{i-1} a_{ij} \mathbf{x}_j^{(k)} = (1 - \omega) a_{ii} \mathbf{x}_i^{(k-1)} - \omega \sum_{j=i+1}^n a_{ij} \mathbf{x}_j^{(k-1)} + \omega \mathbf{b}_i$$

Vektör formunda,

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

Böylece,

$$\mathbf{x}^k = (D - \omega L)^{-1}[(1 - \omega)D + \omega U]\mathbf{x}^{(k-1)} + \omega(D - \omega L)^{-1}\mathbf{b} \quad (3.6)$$

Burada, $T_\omega = (D - \omega L)^{-1}[(1 - \omega)D + \omega U]$ ve $c_\omega = \omega(D - \omega L)^{-1}\mathbf{b}$ olarak belirlersek,

$$\mathbf{x}^k = T_\omega \mathbf{x}^{(k-1)} + c_\omega \quad (3.7)$$

TEOREM 3.22 (Kahan): Eğer $a_{ii} \neq 0$, her $i = 1, 2, \dots, n$, o zaman $\rho(T_\omega) \geq |\omega - 1|$ olur. Bu, S.O.R. metodu sadece $0 < \omega < 2$ olursa yakınsar. (ispatı için Ortega, 1972, s. 123-133).

TEOREM 3.23 (Ostrowski-Reich): Eğer A pozitif tanımlı bir matris ve $0 < \omega < 2$ ise S.O.R. metodu tahmini her başlangıç vektörü seçimi $\mathbf{x}^{(0)}$ için yakınsar.

TEOREM 3.24: Eğer A pozitif tanımlı ve tridiagonal ise, $\rho(T_g) = [\rho(T_j)]^2 < 1$ olur, ve S.O.R. metodu için ω 'nın optimum seçimi,

$$\omega = \frac{2}{1 + \sqrt{1 - [\rho(T_j)]^2}}$$

ω 'nın bu seçimi ile, $\rho(T_\omega) = \omega - 1$ elde ederiz.

3.3 HATA SINIRLARI VE ARDIŞIK GELİŞTİRME

Eğer $\tilde{\mathbf{x}}$, $A\mathbf{x} = \mathbf{b}$ 'nin \mathbf{x} çözümüne yaklaşık bir vektör ve kalan vektörü $\mathbf{r} = \mathbf{b} - A\tilde{\mathbf{x}}$ \mathbf{r} 'nin normu küçük ise $\|\mathbf{x} - \tilde{\mathbf{x}}\|$ 'nin de küçük olduğunu öngörmek mantıklıdır. Ama bazı durumlarda bu özellik işe yaramaz. Özellikle; koşul sayısı büyük olan matrislerde bu beklentide olmamız yanlış olur.

ÖRNEK 1: $A\mathbf{x} = \mathbf{b}$ lineer sistemi şöyle verilmiştir:

$$\begin{bmatrix} 1 & 2 \\ 1.0001 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 3.0001 \end{bmatrix}$$

ve $\mathbf{x} = (-1, 1)^t$ tek çözümüne sahiptir. $\tilde{\mathbf{x}} = (3, -0.0001)^t$ zayıf tahmini için kalıntı vektörü hesaplayınız.

ÇÖZÜM: Elimizde,

$$\mathbf{r} = \mathbf{b} - A\tilde{\mathbf{x}} = \begin{bmatrix} 3 \\ 3.0001 \end{bmatrix} - \begin{bmatrix} 1 & 2 \\ 1.0001 & 2 \end{bmatrix} \begin{bmatrix} 3 \\ -0.0001 \end{bmatrix} = \begin{bmatrix} 0.0002 \\ 0 \end{bmatrix}$$

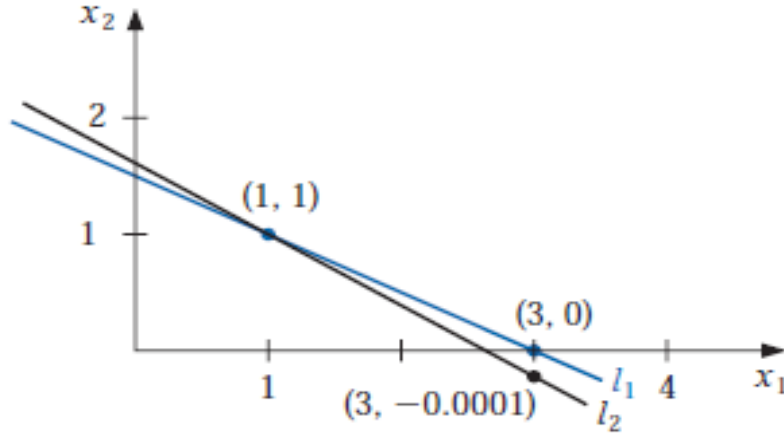
var. Öyleyse $\|\mathbf{r}\|_\infty = 0.0002$ 'dir. Kalıntı vektörün normu küçük olmasına rağmen, $\mathbf{x} = (-1,1)^t$ tahmini açıkça görülür ki biraz zayıftır, aslında, $\|\mathbf{x} - \tilde{\mathbf{x}}\|_\infty = 2$ 'dir. ■

Örnek 1'deki zorluk, sistemin çözümü doğruların kesim noktası olduğu vurgulanarak basitçe açıklanmıştır:

$$l_1: x_1 + 2x_2 = 3 \quad \text{ve} \quad l_2: 1.0001x_1 + 2x_2 = 3.0001.$$

$(3,-0.0001)$ noktası l_2 üzerindedir ve doğrular neredeyse paraleldir. Bu gösterir ki, $(1,1)$ kesim noktalı sistemin çözümünden çok farklı olsa da $(3,-0.0001)$ noktası aynı zamanda l_2 üzerindedir. (Şekil 3.5'e bakınız.).

Şekil 3.5: l_1 ve l_2 durumu.



Kaynak: Burden ve Faires (2011, s. 470).

Örnek 1, ortaya çıkabilecek, aslında çıkacak, olan zorlukları göstermek için tam olarak hazırlanmıştır. Doğrular yakın olarak çakışmış olmasaydı, daha doğru bir tahmin göstermek adına küçük bir kalıntı vektör beklerdik.

Genel durumda, problem olduğunda işaret vermek için sistem geometrisine güvenemeyiz. Ancak, A matrisinin ve tersinin normunu düşünerek bu bilgiyi elde edebiliriz.

TEOREM 3.25: $A\mathbf{x} = \mathbf{b}$ ' nin çözümüne $\tilde{\mathbf{x}}$ bir tahmin olsun, burada A tekil olmayan bir matris, ve \mathbf{r} , $\tilde{\mathbf{x}}$ için kalıntı vektör. O zaman her hangi bir norm için,

$$\|\mathbf{x} - \tilde{\mathbf{x}}\| \leq \|\mathbf{r}\| \cdot \|A^{-1}\|$$

ve eğer $\mathbf{x} \neq 0$ ve $\mathbf{b} \neq 0$ ise,

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \leq \|A\| \cdot \|A^{-1}\| \frac{\|\mathbf{r}\|}{\|\mathbf{b}\|} \quad (3.8)$$

İSPAT: $\mathbf{r} = \mathbf{b} - A\tilde{\mathbf{x}} = A\mathbf{x} - A\tilde{\mathbf{x}}$ ve A tekil değilse (nonsingular), $\mathbf{x} - \tilde{\mathbf{x}} = A^{-1}\mathbf{r}$ elde ederiz.

$$\|\mathbf{x} - \tilde{\mathbf{x}}\| = \|A^{-1}\mathbf{r}\| \leq \|A^{-1}\| \cdot \|\mathbf{r}\|.$$

Dahası, $\mathbf{b} = A\mathbf{x}$ olduğu için, $\|\mathbf{b}\| \leq \|A\| \cdot \|\mathbf{x}\|$ elde ederiz.

Böylece, $1/\|\mathbf{x}\| \leq \|A\|/\|\mathbf{b}\|$ ve

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \leq \frac{\|A\| \cdot \|A^{-1}\|}{\|A\|} \|\mathbf{r}\|.$$

TANIM 3.26: $\|\cdot\|$ normuna göre tekil olmayan A matrisinin koşul sayısı;

$$K(A) = \|A\| \cdot \|A^{-1}\|.$$

Bu notasyon ile,

$$\|\mathbf{x} - \tilde{\mathbf{x}}\| \leq K(A) \frac{\|\mathbf{r}\|}{\|A\|}$$

olur ve

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \leq K(A) \frac{\|\mathbf{r}\|}{\|\mathbf{b}\|}.$$

olur.

Her tekil olmayan A matrisi ve norm $\|\cdot\|$ için,

$$1 = \|\mathbf{I}\| = \|A \cdot A^{-1}\| \leq \|A\| \cdot \|A^{-1}\| = K(A).$$

Bir A matrisi, eğer $K(A)$ 1'e yakınsa iyi koşullandırılmış (well-conditioned) ve $K(A)$ 1'den çok çok büyük ise kötü koşullandırılmış (ill-conditioned) demektir. Durumlandırma bu anlamda, küçük kalıntı vektörlerin nispeten doğru yaklaşık çözümlerin belirttiği göreceli güvenliği kasteder.

ÖRNEK 2: $A = \begin{bmatrix} 1 & 2 \\ 1.0002 & 2 \end{bmatrix}$ matrisi için koşul sayısı nedir?

ÇÖZÜM: Örnek 1’ de tam çözüm $(1,1)^t$ olmasına rağmen yaklaşık çözümümüz $(3, -0.0001)^t$ idi. Buna karşılık kalıntı vektörümüzün büyüklüğü çok küçüktür. Burada A ’nın koşul sayısının büyük olmasını beklemeliyiz. $\|A\|_\infty = \max\{|1| + |2|, |1.001| + |2|\} = 3.0001$, görüldüğü gibi büyük olarak düşünemeyiz. Ancak,

$$A^{-1} = \begin{bmatrix} -10000 & 10000 \\ 5000.5 & -5000 \end{bmatrix},$$

Böylece,

$$\|A\|_\infty = 20000,$$

Ve sonsuz norm için, $K(A) = (20000)(3.0001) = 60002$. Bu örnekte koşul sayısının büyüklüğü, yaklaşıklık kalıntısına bağlı olarak aceleci doğruluk kararları vermekten bizi kesinlikle alıkoymalıdır.

Bir matrisin koşul sayısı tamamen matrisin ve tersinin normalarına bağlı olsa da, tersinin hesaplanması yuvarlama hatasına ve hesaplamanın yapıldığı doğruluğa bağlıdır. Eğer işlemler t kadar basamak doğrulukta aritmetik içeriyorsa, A matrisi için yaklaşık koşul sayısı, t -basamaklı aritmetik kullanılarak yapılan, matrisin normu kadar A ’nın tersine yaklaşıklığın normudur. Aslında bu koşul sayısı A ’nın tersini hesaplamada kullanılan metoda da bağlıdır. Ek olarak, tersi bulmak için kullanılan hesaplama sayısı nedeniyle, direkt olarak tersi hesaplamadan koşul sayısını tahmin etmemiz gerekir. $Ax = b$ lineer sisteminin yaklaşık çözümünün t -basamaklı aritmetik ve Gauss eliminasyonu kullanılarak hesaplandığını varsayarsak, \tilde{x} yaklaşıklığı için r kalıntı vektörünün,

$$\|r\| \approx 10^{-t} \|A\| \cdot \|\tilde{x}\|. \quad (3.9)$$

olduğu gösterilebilir.

TEOREM 3.27: $A \in \mathbb{R}^{m \times n}$ nin saklanan kısmı \tilde{A} ve $\tilde{A} = A + E$ dir. $E \in \mathbb{R}^{m \times n}$ dir ve

$$\|E\|_1 \leq \text{eps} \|A\|_1$$

burada eps makine duyarlılık sayısıdır.

İSPAT: Eğer $\tilde{A} = (\tilde{a}_{ij})$ ve

$$\tilde{a}_{ij} = fl(a_{ij}) = a_{ij}(1 + \epsilon_{ij})$$

buradaki $|\epsilon_{ij}| \leq \text{eps}$. Buradan

$$\begin{aligned}\|E\|_1 &= \|\tilde{A} - A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^m |\tilde{a}_{ij} - a_{ij}| \\ &\leq \max_{1 \leq j \leq n} \sum_{i=1}^m |\tilde{a}_{ij} \epsilon_{ij}| \leq eps \max_{1 \leq j \leq n} \sum_{i=1}^m |\tilde{a}_{ij}| = eps \|A\|_1\end{aligned}$$

Bu yaklaşımdan, t -basamaklı aritmetikte etkin bir koşul sayısı tahmini, A matrisini tersleme gereği duymadan elde edilebilir. Aslında bu yaklaşıklık, Gauss Eliminasyon tekniğindeki bütün aritmetik işlemler t -basamaklı aritmetik kullanılarak yapıldığını varsayar ama kalıntıyı hesaplamada gerekli işlemler çift duyarlı aritmetikle ($2t$ -digit) yapılır. Bu teknik çokça sayısal efor gerektirmez ve kalıntı hesaplarken oluşan neredeyse eşit sayıların çıkarmalarından kaynaklı doğruluk kaybının çoğunu elimine eder.

t -basamaklı koşul sayısı $K(A)$ 'nin yaklaşıklığı aşağıdaki lineer sistemden gelir,

$$A\mathbf{y} = \mathbf{r}.$$

Gauss eliminasyon metodu için çarpanlar halihazırda hesaplandığı için bu sisteme çözüm zaten tahmin edilebilir. Öyleyse A matrisi P^tLU formunda faktörlenebilir, aslında $\tilde{\mathbf{y}}$, $A\mathbf{y} = \mathbf{r}$ 'nin yaklaşık çözümü,

$$\tilde{\mathbf{y}} \approx A^{-1}\mathbf{r} = A^{-1}(\mathbf{b} - A\tilde{\mathbf{x}}) = A^{-1}\mathbf{b} - A^{-1}\tilde{\mathbf{x}} = \mathbf{x} - \tilde{\mathbf{x}} \quad (3.10)$$

sağlar ve

$$\mathbf{x} = \tilde{\mathbf{x}} + \tilde{\mathbf{y}}$$

olur.

Öyleyse $\tilde{\mathbf{y}}$, \mathbf{x} çözümünü, $\tilde{\mathbf{x}}$ orijinal sisteme yaklaştırdığında üretilen hatanın bir tahminidir. 3.9 ve 3.10 denklemleri kullanarak,

$$\|\tilde{\mathbf{y}}\| \approx \|\mathbf{x} - \tilde{\mathbf{x}}\| = \|A^{-1}\mathbf{r}\| \leq \|A^{-1}\| \cdot \|\mathbf{r}\| \approx \|A^{-1}\| (10^{-t} \|A\| \cdot \|\tilde{\mathbf{x}}\|) = 10^{-t} \|\tilde{\mathbf{x}}\| K(A).$$

Bu, Gauss Eliminasyon ve t -basamak tipli aritmetik kullanılarak $A\mathbf{x} = \mathbf{b}$ sisteminin çözümünü içeren koşul sayısı için aşağıda tanımlı bir yaklaşım verir,

$$K(A) \approx \frac{\|\tilde{\mathbf{y}}\|}{\|\tilde{\mathbf{x}}\|} 10^{-t} \quad (3.11)$$

$A\mathbf{x} = \mathbf{b}$ lineer sisteminde A ve \mathbf{b} tam olarak temsil edilebilir. Gerçekte, a_{ij} ve b_j girdileri, δa_{ij} ve δb_j kadar değiştirilecek ya da bozularak $A\mathbf{x} = \mathbf{b}$ olması durumunda çözülecek aşağıdaki lineer sisteme neden olacaktır,

$$(A + \delta A)\mathbf{x} = \mathbf{b} + \delta \mathbf{b}.$$

Normalde, $\|\delta\mathbf{A}\|$ ve $\|\delta\mathbf{b}\|$ küçükse (10^{-t} derecesinde), t -basamaklı aritmetik, paralel olarak $\|\mathbf{x} - \tilde{\mathbf{x}}\|$ küçük olacak şekilde bir $\tilde{\mathbf{x}}$ çözümü verecektir. Ancak, kötü durumda, \mathbf{A} ve \mathbf{b} tam olarak temsil edilse de yuvarlama hataları, $\|\mathbf{x} - \tilde{\mathbf{x}}\|$ 'nin çok büyük olmasına neden olabilir. Takip eden teorem lineer sistem bozunmalarını bir matrisin koşul sayısı ile ilişkilendirir.

TEOREM 3.28: Varsayalım ki \mathbf{A} tekil olmayan bir matris ve

$$\|\delta\mathbf{A}\| < \frac{1}{\|\mathbf{A}^{-1}\|}$$

olsun.

$(\mathbf{A} + \delta\mathbf{A})\mathbf{x} = \mathbf{b} + \delta\mathbf{b}$ 'ye çözüm $\tilde{\mathbf{x}}$, $\mathbf{A}\mathbf{x} = \mathbf{b}$ 'nin çözümü \mathbf{x} 'e şu hata tahmini ile yaklaşır.

Bu takdirde;

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \leq \frac{K(\mathbf{A})\|\mathbf{A}\|}{\|\mathbf{A}\| - K(\mathbf{A})\|\delta\mathbf{A}\|} \left(\frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|} + \frac{\|\delta\mathbf{A}\|}{\|\mathbf{A}\|} \right) \quad (3.12)$$

eşitsizliği sağlanır.

4. MATERYAL VE METOT

4.1 EN DİK İNİŞ METODU (THE METHOD OF STEEPEST DESCENT)

$$A\vec{x} = b \quad (4.1)$$

lineer denklem sistemini ele alalım; burada A simetrik ve pozitif tanımlı $n \times n$ boyutunda bir matristir.

$\forall \vec{x}, \vec{x} \neq \vec{0}, \vec{x}^T A \vec{x} > 0 \Rightarrow A$ pozitif sabit bir matristir.

Aşağıdaki şekilde tanımlı bir fonksiyon ele alalım;

$$F(\mathbf{y}) = \frac{1}{2}(\mathbf{y} - \mathbf{x}, A(\mathbf{y} - \mathbf{x})) \quad (4.2)$$

Burada \mathbf{x} (4.1)'in çözümü ve $(.,.)$ ise \mathbb{R}^n 'de iç çarpımdır.

(Hatırlatma: İç Çarpım; $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ olmak üzere ;

$(\mathbf{x}, \mathbf{y}) = x_1 y_1 + x_2 y_2 + \dots + x_n y_n$ şeklinde tanımlanan fonksiyona iç çarpım denir.)

Açıkça görülür ki F fonksiyonunun \mathbf{x} ' e eşit olan \mathbf{y} 'de tek bir minimumu vardır, o da (4.1)' in çözümüdür.

$$E(\mathbf{y}) = F(\mathbf{y}) - F(\mathbf{0}) = \frac{1}{2}(\mathbf{y}, A\mathbf{y}) - (\mathbf{y}, \mathbf{b}) \quad (4.3)$$

Bu eşitlik (4.1)'in çözümünde tek bir minimuma sahiptir. En dik iniş ve eşlenik gradyan metotlarının her ikisi de iteratif metotlardır. Bu metotlar E'nin değerini $E(\mathbf{y})$ 'nin en küçük veya küçüğe yakın değeri için \mathbf{y} vektörü elde edilene kadar her adımda düşürür. Bir çok uygulamada $E(\mathbf{y})$ sistemin enerjisi gibi önem miktarını temsil eder. Bu durumlarda (4.1)'in çözümü minimum enerji durumudur. $E(\mathbf{y})$ fonksiyonunun gradyanı,

$$G(\mathbf{y}) = A\mathbf{y} - \mathbf{b} = -\mathbf{r} \quad (4.4)$$

vektörüdür. Burada \mathbf{r} kalıntı olarak adlandırılır.

Bir fonksiyonunun gradyanı en dik çıkış yönünü işaret ettiği için, fonksiyonun değerini düşürmek için gradyanın karşı yönünde gitmek avantajlıdır. Bu da en dik iniş yönüdür. En dik iniş metodu, ilk değer vektörü \mathbf{x}^0 'dan başlayarak aşağıdaki gibi verilir:

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k \cdot \mathbf{r}^k \quad (4.5)$$

Burada,

$$\mathbf{r}^k = \mathbf{b} - A\mathbf{x}^k$$

ve α_k bir parametredir.

Buradaki küçük Roma harfleri vektörleri gösterecek ve üst sembollere sahip olacak. Yunan harfleri ise sayısal değerleri verecek ve alt sembollere sahip olacak. \mathbf{v} vektörünün normu $|\mathbf{v}|$ ile gösterilecek ve burada $|\mathbf{v}| = (\mathbf{v}, \mathbf{v})^{\frac{1}{2}}$.

(4.5)'teki α_k parametresi $E(\mathbf{x}^{k+1})$ minimum olacak şekilde seçilecek.

$$\begin{aligned} E(\mathbf{x}^{k+1}) &= E(\mathbf{x}^k + \alpha_k \mathbf{r}^k) \\ &= \frac{1}{2}(\mathbf{x}^k, A\mathbf{x}^k) + \alpha_k(\mathbf{r}^k, A\mathbf{x}^k) + \frac{1}{2}\alpha_k^2(\mathbf{r}^k, A\mathbf{r}^k) - (\mathbf{x}^k, \mathbf{b}) - \alpha_k(\mathbf{r}^k, \mathbf{b}) \\ &= E(\mathbf{x}^k) - \alpha_k(\mathbf{r}^k, \mathbf{r}^k) + \frac{1}{2}\alpha_k^2(\mathbf{r}^k, A\mathbf{r}^k). \end{aligned}$$

$\partial E / \partial \alpha_k = 0$ alırsak α_k şöyle olur,

$$\alpha_k = \frac{(\mathbf{r}^k, \mathbf{r}^k)}{(\mathbf{r}^k, A\mathbf{r}^k)} = \frac{|\mathbf{r}^k|^2}{(\mathbf{r}^k, A\mathbf{r}^k)} \quad (4.6)$$

(\mathbf{x}^{k+1}) minimum olduğu değerdir. (4.5)'ten,

$$\mathbf{r}^{k+1} = \mathbf{r}^k - \alpha_k A\mathbf{r}^k$$

ve (4.6)'dan,

$$(\mathbf{r}^{k+1}, \mathbf{r}^k) = (\mathbf{r}^k, \mathbf{r}^k) - \alpha_k(\mathbf{r}^k, A\mathbf{r}^k) = 0 \quad (4.7)$$

ardışık kalıntıların ortogonal olduğunu gösterir. α_k 'nin optimum seçimi için elimizde,

$$E(\mathbf{x}^{k+1}) = E(\mathbf{x}^k) - \frac{1}{2} \frac{|\mathbf{r}^k|^4}{(\mathbf{r}^k, A\mathbf{r}^k)} \quad (4.8)$$

bulunur ve k artarken $E(\mathbf{x}^k)$ azalır. $E(\mathbf{x}^k)$ ve \mathbf{r}^k 'nin tanımdan şunun farkına varılmalıdır;

$$E(\mathbf{x}^k) = \frac{1}{2}(A^{-1}\mathbf{r}^k, \mathbf{r}^k) - F(0)$$

ve böylece (4.8) şuna eşit olduğundan,

$$(A^{-1}\mathbf{r}^{k+1}, \mathbf{r}^{k+1}) = (A^{-1}\mathbf{r}^k, \mathbf{r}^k) - \frac{|\mathbf{r}^k|^4}{(\mathbf{r}^k, A\mathbf{r}^k)} \quad (4.9)$$

olur.

Şimdi bu formülleri en dik iniş için bir araya getirirsek,

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k \mathbf{r}^k \quad (4.10a)$$

$$\mathbf{r}^{k+1} = \mathbf{r}^k - \alpha_k A\mathbf{r}^k \quad (4.10b)$$

$$\alpha_k = \frac{|\mathbf{r}^k|^2}{(\mathbf{r}^k, A\mathbf{r}^k)} \quad (4.10c)$$

elde edilir.

Dikkat edelim ki A matrisini bellekte saklamaya gerek yok. Her bir adım için sadece bir matris çarpımına ihtiyacımız var. Verilen bir \mathbf{r} vektörü için sadece $A\mathbf{r}$ ' yi üretmeye ihtiyacımız var. (4.10b) formülü $\mathbf{r}^k = \mathbf{b} - A\mathbf{x}^k$ yerine kullanılmalıdır. \mathbf{b} ve $A\mathbf{x}^k$ vektörleri k ' nın büyük değerleri için büyüklükleri birbirine yakın olacağından hesaplamalarda sonlu duyarlılıklı bilgisayarlar kullanıldığında digit kaybına yol açacağından (4.10b) formülü bu problemi ortadan kaldırır.

Dik iniş metodu A matrisinin simetrik ve pozitif tanımlı olması durumunda geçerli olmasına rağmen A' nın simetrik olmadığı durumda da (4.10) algoritmasını kullanabiliriz. Aşağıdaki teorem bu metodun yakınsayacağı koşulları verir.

TEOREM 4.1.1: A matrisi pozitif tanımlı bir matris ise öyle ki $A^T A^{-1}$ de pozitif tanımlıdır, bu takdirde (4.10) ile verilen algoritma her başlangıç \mathbf{x}^0 vektörü için (4.1)'in tek bir çözümüne yakınsar.

İSPAT: İlk olarak dikkat edelim ki eğer A pozitif tanımlı ise A^{-1} de pozitif tanımlıdır. Eğer $A^T A^{-1}$ pozitif tanımlı ise bu takdirde C_0 ve C_1 sabitleri vardır ve bunlar aşağıdaki eşitsizlikleri sağlar:

$$C_0(\mathbf{x}, A^{-1}\mathbf{x}) \leq (\mathbf{x}, A^T A^{-1}\mathbf{x}) \quad (4.11)$$

$$C_1(\mathbf{x}, A\mathbf{x}) \leq (\mathbf{x}, \mathbf{x}) \quad (4.12)$$

Şimdi $(\mathbf{r}^{k+1}, A^{-1}\mathbf{r}^{k+1})$ denkleminde bir bakalım. Buradaki \mathbf{r}^{k+1} 'in \mathbf{r}^k 'ya bağlı olduğunu (4.10b) ' den biliyoruz ve $\mathbf{r}^0 = \mathbf{b} - A\mathbf{x}^0$ dir.

$$\begin{aligned} (\mathbf{r}^{k+1}, A^{-1}\mathbf{r}^{k+1}) &= (\mathbf{r}^k, A^{-1}\mathbf{r}^k) - \alpha_k(\mathbf{r}^k, \mathbf{r}^k) - \alpha_k(A\mathbf{r}^k, A^{-1}\mathbf{r}^k) + \alpha_k^2(\mathbf{r}^k, A\mathbf{r}^k) \\ &= (\mathbf{r}^k, A^{-1}\mathbf{r}^k) - \alpha_k(A\mathbf{r}^k, A^T A^{-1}\mathbf{r}^k) \end{aligned}$$

ifadesini α_k ' nın denklemdir. (4.12) ' yi kullanarak ,

$$\alpha_k = \frac{(\mathbf{r}^k, \mathbf{r}^k)}{(\mathbf{r}^k, A\mathbf{r}^k)} \geq C_1$$

bulunur. (4.11) 'i kullanarak

$$(\mathbf{r}^{k+1}, A^{-1}\mathbf{r}^{k+1}) \leq (\mathbf{r}^k, A^{-1}\mathbf{r}^k)(1 - C_0 C_1), k \geq 0 \quad (4.13)$$

elde edilir. Burada $1 - C_0 C_1 > 0$ dır. Çünkü; (4.13) denkleminde, A^{-1} pozitif tanımlı olduğu için $(\mathbf{r}^{k+1}, A^{-1}\mathbf{r}^{k+1}) > 0$ ve $(\mathbf{r}^k, A^{-1}\mathbf{r}^k) > 0$ dır. Bu da

demektir ki; $C_0C_1 < 1$, C_0 ve C_1 'in tanımlarından dolayı aynı zamanda $C_0C_1 > 0$ 'dır ve sonuç olarak;

$$\lim_{k \rightarrow \infty} (1 - C_0C_1)^k = 0$$

Ardışık olarak gösterilebilir ki;

$$(\mathbf{r}^k, A^{-1}\mathbf{r}^k) \leq (\mathbf{r}^0, A^{-1}\mathbf{r}^0)(1 - C_0C_1)^k$$

ve eşitsizliğin her iki tarafından $k \rightarrow \infty$ için limit alınır; $(\mathbf{r}^k, A^{-1}\mathbf{r}^k)$ 'nin sifıra gittiği görülür.

Ancak A^{-1} matrisi pozitif tanımlı bir matris olduğundan bu da bize \mathbf{r}^k 'nin sifıra yakınsadığını söyler.

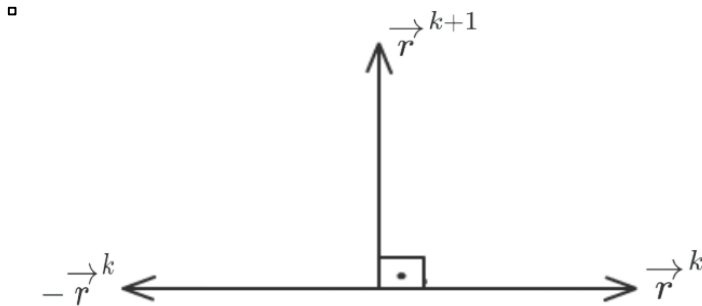
$$\mathbf{x}^k = A^{-1}(\mathbf{b} - \mathbf{r}^k)$$

olduğundan dolayı, \mathbf{x}^k , $A^{-1}\mathbf{b}$ 'ye yakınsar, bu da (4.1) denkleminin tek bir çözümünü verir, yani $\mathbf{x} = A^{-1}\mathbf{b}$ dir.

SONUÇ 4.1.2: A matrisi simetrik ve pozitif tanımlı ise Dik İnişöntemi (DİY) yöntemi yakınsar. (4.13) eşitsizliği şunu söyler; eğer C_0C_1 ifadesi 1'e yakınsa DİY hızlı yakınsar. Ancak, eğer kalan vektörler salınım yaparlarsa DİY oldukça yavaş olabilir.

k	$\mathbf{r}^k = \mathbf{b} - A\mathbf{x}^k$
1	(-0.001,-0.002)
2	(0,001,0.02)

Şekil 4.1: \mathbf{r}^k vektörünün gösterimi.



DİY yavaş olduğu için bunu hızlandırmak mümkündür. Bu hızlandırma yöntemlerinden bir tanesi Eşlenik Gradyan Metodudur (EGM) (Conjugate Gradient Method). Bundan

sonraki bölümlerde DIY yönlendirmek için önce EGM'yi sonra da ön koşullu minimum kalan metodlarını (Preconditioned Minimum Residual) ele alacağız.

4.2 EŞLENİK GRADYAN METODU (THE CONJUGATE GRADIENT METHOD)

Bu bölümde en dik iniş yönteminin hızlandırılmış hali olan eşlenik gradyan metodunu ele alacağız.

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k [\mathbf{r}^k + \gamma^k (\mathbf{x}^k - \mathbf{x}^{k-1})]$$

α_k ve γ^k sayısal parametrelerdir.

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k \mathbf{p}^k$$

formülünü kullanarak yukarı denklemi yeniden yazarsak,

$$\begin{aligned} \mathbf{p}^k &= \mathbf{r}^k + \gamma_k (\mathbf{x}^k - \mathbf{x}^{k-1}) = \mathbf{r}^k + \gamma_k \alpha_{k-1} \mathbf{p}^{k-1} \\ &= \mathbf{r}^k + \beta_{k-1} \mathbf{p}^{k-1} \end{aligned}$$

elde edilir. Şimdi bu formülleri toplarsak;

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k \mathbf{p}^k \quad (4.14a)$$

$$\mathbf{r}^{k+1} = \mathbf{r}^k - \alpha_k A \mathbf{p}^k \quad (4.14b)$$

$$\mathbf{p}^{k+1} = \mathbf{r}^{k+1} + \beta_k \mathbf{p}^k \quad (4.14c)$$

buradaki α_k ve β_k tespit edilecek olan parametreler, \mathbf{p}^k vektörü ise k. adımdaki arama doğrultusudur.

Şimdi α_k ve β_k parametrelerini hesaplamak isteyelim ve de (4.14) mümkün olduğunca hızlı yakınsayacak şekilde \mathbf{p}^0 'ın ne olması gerektiğini hesaplayalım. En dik iniş yöntemiyle beraber, \mathbf{x}^{k+1} ve α_k değerlerini $E(\mathbf{x}^{k+1})$ yı minimum yapacak şekilde seçme niyetindeyiz. Başlangıç için \mathbf{p}^0 in bilindiğini farz edelim.

$$\begin{aligned} E(\mathbf{x}^{k+1}) &= \frac{1}{2} (\mathbf{x}^k, A \mathbf{x}^k) + \alpha_k (\mathbf{p}^k, A \mathbf{x}^k) - \frac{1}{2} \alpha_k^2 (\mathbf{p}^k, A \mathbf{p}^k) - (\mathbf{x}^k, \mathbf{b}) - \alpha_k (\mathbf{p}^k, \mathbf{b}) \\ &= E(\mathbf{x}^k) - \alpha_k (\mathbf{p}^k, \mathbf{r}^k) + \frac{1}{2} \alpha_k^2 (\mathbf{p}^k, A \mathbf{p}^k) \end{aligned}$$

$E(\mathbf{x}^{k+1})$ ' in α_k 'ya göre türevi alınırsa,

$$\frac{\partial E}{\partial \alpha_k} = -(\mathbf{p}^k, \mathbf{r}^k) + \alpha_k (\mathbf{p}^k, A \mathbf{p}^k) = 0$$

denkleminde

$$\alpha_k = \frac{(\mathbf{p}^k, \mathbf{r}^k)}{(\mathbf{p}^k, A\mathbf{p}^k)} \quad (4.15)$$

α_k 'nin en uygun değeri elde edilmiş olur. Bu α_k değerini kullanarak,

$$E(\mathbf{x}^{k+1}) = E(\mathbf{x}^k) - \frac{1}{2} \frac{(\mathbf{p}^k, \mathbf{r}^k)^2}{(\mathbf{p}^k, A\mathbf{p}^k)}$$

elde edilir.

$k = 0$ için \mathbf{p}^0 'ı keyfi seçebiliriz. Ancak, \mathbf{p}^0 yerine \mathbf{r}^0 da seçilebilir. Bu seçim dik iniş yöntemine eşdeğerdir. Yukarıdaki formülden anlaşılıyor ki $E(\mathbf{x}^1) < E(\mathbf{x}^0)$ olacaktır. Buradan sonra $\mathbf{p}^0 = \mathbf{r}^0$ kabul edeceğiz ve bu seçimin diğer avantajlarını ileride göreceğiz.

(4.15) ve (4.14b) denklemlerini kullanarak

$$(\mathbf{p}^k, \mathbf{r}^{k+1}) = (\mathbf{p}^k, \mathbf{r}^k) - \alpha_k (\mathbf{p}^k, A\mathbf{p}^k) = 0 \quad (4.16)$$

burada (4.14c) kullanılırsa

$$\begin{aligned} (\mathbf{p}^{k+1}, \mathbf{r}^{k+1}) &= (\mathbf{r}^{k+1}, \mathbf{r}^{k+1}) + \beta_k (\mathbf{p}^k, \mathbf{r}^{k+1}) \\ &= |\mathbf{r}^{k+1}|^2 \end{aligned}$$

her $k \geq 0$ için yazılabilir.

Daha sonra bizim seçimimizle \mathbf{p}^0

$$(\mathbf{p}^k, \mathbf{r}^k) = (\mathbf{r}^k, \mathbf{r}^k) = |\mathbf{r}^k|^2 ; k \geq 0 \quad (4.17)$$

bulunur. Bu yöntem eşlenik gradyan metot analizinde tekrar tekrar kullanılacaktır.

(4.17) ve (4.15)'den;

$$\alpha_k = \frac{\|\mathbf{r}^k\|^2}{(\mathbf{p}^k, A\mathbf{p}^k)} \quad (4.18)$$

bu da α_k 'yi hesaplamak için çok uygun bir formüldür. O zaman formülü tamamlarsak;

$$E(\mathbf{x}^{k+1}) = E(\mathbf{x}^k) - \frac{1}{2} \frac{\|\mathbf{r}^k\|^4}{(\mathbf{p}^k, A\mathbf{p}^k)}$$

elde edilir.

Bu formül gösteriyor ki \mathbf{p}^k , $(\mathbf{p}^k, A\mathbf{p}^k)$ minimum olacak şekilde seçilmelidir çünkü bu verilen \mathbf{x}^k değeri için $E(\mathbf{x}^{k+1})$ minimize edilecektir. (4.14c) formülü yardımıyla görüyoruz ki verilen \mathbf{p}^{k-1} değeri için $(\mathbf{p}^k, A\mathbf{p}^k)$ 'yi minimize etmek için β_{k-1} seçilmelidir. Elimizde;

$$(\mathbf{p}^k, A\mathbf{p}^k) = (\mathbf{r}^k, A\mathbf{r}^k) + 2\beta_{k-1}(\mathbf{r}^k, A\mathbf{p}^{k-1}) + \beta_{k-1}^2(\mathbf{p}^{k-1}, A\mathbf{p}^{k-1})$$

Böylece β_{k-1} 'in optimum seçimi,

$$\beta_{k-1} = -\frac{(\mathbf{r}^k, A\mathbf{p}^{k-1})}{(\mathbf{p}^{k-1}, A\mathbf{p}^{k-1})}, \quad k \geq 1 \text{ için}$$

veya eşit olarak,

$$\beta_k = -\frac{(\mathbf{r}^{k+1}, A\mathbf{p}^k)}{(\mathbf{p}^k, A\mathbf{p}^k)}, \quad k \geq 0 \text{ için} \quad (4.19)$$

Bu formülden ilk çıkarım, bu formülü β_k için (4.14c) ile kullanırsak elimizde,

$$(\mathbf{p}^{k+1}, A\mathbf{p}^k) = (\mathbf{r}^{k+1}, A\mathbf{p}^k) + \beta_k(\mathbf{p}^k, A\mathbf{p}^k) = 0$$

Buradan şu önemli sonucu elde ederiz;

$$(\mathbf{p}^{k+1}, A\mathbf{p}^k) = 0, \quad k \geq 0 \text{ için} \quad (4.20)$$

ardışık arama yönünde eşlenik olduğunu söyleriz. (4.20)'yi (4.14c) ile kullanırsak şunu elde ederiz;

$$\begin{aligned} (\mathbf{p}^k, A\mathbf{p}^k) &= (\mathbf{r}^k, A\mathbf{p}^k) + \beta_{k-1}(\mathbf{p}^{k-1}, A\mathbf{p}^k) \\ &= (\mathbf{r}^k, A\mathbf{p}^k) \end{aligned}$$

(4.14b)' yi (4.18) ile kullanırsak aşağıdakini elde ederiz,

$$\begin{aligned} (\mathbf{r}^{k+1}, \mathbf{r}^k) &= (\mathbf{r}^k, \mathbf{r}^k) - \alpha_k(A\mathbf{p}^k, \mathbf{r}^k) \\ &= (\mathbf{r}^k, \mathbf{r}^k) - \alpha_k(\mathbf{p}^k, A\mathbf{p}^k) \\ &= 0 \end{aligned} \quad (4.21)$$

Şimdi, β_k için (4.19)' dan daha uygun bir formül elde ederiz. Öncelikle (4.14b) ve (4.21) ile,

$$\begin{aligned} (\mathbf{r}^{k+1}, \mathbf{r}^{k+1}) &= (\mathbf{r}^{k+1}, \mathbf{r}^k) - \alpha_k(\mathbf{r}^{k+1}, A\mathbf{p}^k) \\ &= -\alpha_k(\mathbf{r}^{k+1}, A\mathbf{p}^k) \end{aligned}$$

öyleyse (4.19) ile β_k için formülümüz,

$$\beta_k = \frac{1}{\alpha_k} \frac{|\mathbf{r}^{k+1}|^2}{(\mathbf{p}^k, A\mathbf{p}^k)} = \frac{|\mathbf{r}^{k+1}|^2}{|\mathbf{r}^k|^2}.$$

Eşlenik Gradyan Metodu için şimdi formülleri bir araya getirelim:

$$\mathbf{p}^0 = \mathbf{r}^0 = \mathbf{b} - A\mathbf{x}^0 \quad (4.22a)$$

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k A\mathbf{p}^k \quad (4.22b)$$

$$\mathbf{r}^{k+1} = \mathbf{r}^k - \alpha_k A\mathbf{p}^k \quad (4.22c)$$

$$\mathbf{p}^{k+1} = \mathbf{r}^{k+1} + \beta_k \mathbf{p}^k \quad (4.22d)$$

$$\alpha_k = \frac{|r^k|^2}{(p^k, Ap^k)} \quad (4.22e)$$

$$\beta_k = \frac{|r^{k+1}|^2}{|r^k|^2} \quad (4.22f)$$

(4.22) formüllerinden şunu görürüz: eğer β_k küçükse, şöyle ki $|r^{k+1}|, |r^k|$ 'den daha küçükse, o zaman p^{k+1} doğal olarak r^{k+1} 'dir ve Eşlenik Gradyan Metodu, En Dik İniş Metoduna yakındır. Eğer $|r^{k+1}|, |r^k|$ 'den çok daha küçük değilse, o zaman yeni arama yönü, p^{k+1} , lokal en dik iniş yönü r^{k+1} 'e yakın olmayacaktır. Farkına varalım ki, (4.22c)'de tanımlanan r^k vektörleri, k 'nın her değeri için kalıntı $b - Ax^k$ 'ya eşittir. Eşlenik Gradyan Metodu için arama yönleri ve kalıntılar için çok ilginç ve önemli bir sonucu ispatlayalım:

TEOREM 4.2.1: (4.22) Eşlenik Gradyan Metodu için kalıntılar ve arama yönleri aşağıdaki ilişkileri sağlar,

$$(r^k, r^j) = (p^k, Ap^j) = 0, \quad k \neq j \text{ için} \quad (4.23)$$

İSPAT: Bu sonucu tümevarım ile ispatlayalım. İlk önce (4.21) ve (4.20) ile,

$$(r^0, r^1) = 0$$

ve

$$(p^0, Ap^1) = 0$$

olur.

Sonra farz edelim ki,

$$(r^l, r^j) = (p^l, Ap^j) = 0, \quad 0 \leq j < l \leq k \text{ için}$$

olsun. $0 \leq j < l \leq k + 1$ ile bütün j ve l 'ler için bunun geçerli olduğunu da göstermek isteyelim. İlk olarak, (4.21) ve (4.20)'den, j eşittir k ve l eşittir $k + 1$ ile bu durumu alalım:

$$(r^{k+1}, r^k) = (p^{k+1}, Ap^k) = 0.$$

Şimdi farz edelim ki j, k 'dan daha küçük. (4.22c) ve (4.22d)'den (p^l, Ap^j) ve (p^l, Ap^{j-1}) giriş hipotezimizden sıfır olduğu için elimizde,

$$\begin{aligned} (r^{k+1}, r^j) &= (r^k, r^j) - \alpha_k (Ap^k, r^j) \\ &= -\alpha_k (Ap^k, p^j - \beta_{j-1} p^{j-1}) \\ &= 0 \end{aligned}$$

olur. Ayrıca ispatlanan sonuçtan, j, k 'dan daha küçük olduğu için,

$$\begin{aligned} (\mathbf{p}^{k+1}, A\mathbf{p}^j) &= (\mathbf{r}^{k+1}, A\mathbf{p}^j) + \beta_k(\mathbf{p}^k, A\mathbf{p}^j) \\ &= (\mathbf{r}^{k+1}, \mathbf{r}^j - \mathbf{r}^{j+1})\alpha_j^{-1} \\ &= 0 \end{aligned}$$

olur. Bu teoremin ispatını tamamlar.

Teorem 4.2.1 doğrudan şu sonuca varır:

SONUÇ 4.2.2: Eğer A matrisi $K \times K$ boyutunda simetrik pozitif sabit bir matris ise, o zaman Eşlenik Gradyan algoritması K adımında yakınsar.

İSPAT: Teorem 4.2.1'de bütün kalıntılar (4.23) ile karşılıklı olarak ortogondur. O zaman, $k = 0, \dots, K - 1$ için $\mathbf{r}^k, \mathbf{r}^{k+1}$ 'ya ortogondur. Uzayın boyutu K olduğu için \mathbf{r}^K sıfır olmalıdır ve böylece metod K adımda yakınsamalıdır.

4.3 EŞLENİK GRADYAN METODU İÇİN YAKINSAMA

Teorem 4.2.1 gösteriyor ki $K \times K$ boyutunda bir A matrisi için Eşlenik Gradyan Metodu en fazla K adımda yakınsar. Fakat, şimdi metodun yakınsama oranı üzerine metodun sıkça K 'dan daha az adımda yakınsadığını gösteren bir ispat yapacağız. Teorem 4.4.1'deki A matrisi, özdeğerleri $[a, b]$, ($a > 0$) aralığında olan simetrik pozitif tanımlı bir matris ise bu takdirde Eşlenik Gradyan Metodu için hata vektörü \mathbf{e}^k aşağıdaki eşitsizliği sağlar:

$$(\mathbf{e}^k, A\mathbf{e}^k)^{\frac{1}{2}} \leq 2 \left(\frac{\sqrt{b} - \sqrt{a}}{\sqrt{b} + \sqrt{a}} \right)^k (\mathbf{e}^0, A\mathbf{e}^0)^{\frac{1}{2}} \quad (4.24)$$

İSPAT: Önce (4.22b) ve (4.22c) denklemlerini hatırlayalım;

$$(4.22b) \text{ denklemini } \mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k \mathbf{p}^k$$

$$(4.22c) \text{ denklemini } \mathbf{r}^{k+1} = \mathbf{r}^k - \alpha_k A\mathbf{p}^k \text{ idi.}$$

Eşlenik Gradyan Metodu'nda k adımdan sonra kalan vektör $\{A^j \mathbf{r}^0\}$, $j = 0, \dots, k$ vektör kümesinin bir lineer kombinasyonu olarak ifade edilebilir. Bu gözlemi,

$$\mathbf{r}^k = R_k(A)\mathbf{r}^0 \quad (4.25)$$

olarak yazabiliriz. Burada $R_k(A)$, k 'nci dereceden A 'ya göre bir polinom ifade eder.

$\mathbf{r}^k = R_k(A)\mathbf{r}^0$ eşitliğinden,

$$R_k(0) = 1 \quad (4.26)$$

$k \in \mathbb{Z}^+$ için, alalım. Hata vektörü e^k , Eşlenik Gradyan Metodu'ndaki k'inci adımda kalan vektörü ile ilişkisini şöyle yazabiliriz:

$$r^k = Ae^k \quad (4.27)$$

A matrisi ile $R_k(A)$ ifadesi değişmeli olduğu için yani $AR_k(A) = R_k(A)A$,

$$A(e^k - R_k(A)e^0) = 0$$

$$r^k = Ae^k$$

$$Ae^k = R_k(A)Ae^0$$

$$= AR_k(A)e^0$$

ve A matrisinin tersi olduğu (tekel olmadığı) için,

$$e^k = R_k(A)e^0 \quad (4.28)$$

Şimdi Teorem 4.2.1'i kullanarak;

$$(e^k, Ae^k) = Q_k(A)e^0, Ae^k \quad (4.29)$$

eşitliğini elde etmeye çalışacağız. Burada $Q_k(A)$ k. dereceden A'ya göre polinom olup, $Q_k(0) = 1$ kabul edeceğiz.

(4.4.5) ve Teorem 4.2.1'i kullanırsak, herhangi γ_j katsayıları için,

$$(e^k, Ae^k) = (e^k, r^k) = \left(e^k + \sum_{j=0}^{k-1} \gamma_j r^j, r^k \right)$$

olur.

(4.27) ve (4.28) denklemlerini kullanarak, γ_j katsayılarının uygun seçimi ve $Q_k(0) = 1$ koşulu ile,

$$e^k + \sum_{j=0}^{k-1} \gamma_j r^j = \left[R_k(A) + \sum_{j=0}^{k-1} \gamma_j R_j(A) \right] e^0 = Q_k(A)e^0$$

Bu da (4.29) denklemini verir.

Hatırlatma: Cauchy-Schwarz Eşitsizliği

$$|(a, b)|^2 \leq \|a\| \|b\|$$

$$|(a, b)| \leq \|a\|^{\frac{1}{2}} \|b\|^{\frac{1}{2}}$$

Bir iç çarpım uzayının elemanları x ve y ise, bu durumda;

$$|(x, y)| \leq (x, x)^{\frac{1}{2}} (y, y)^{\frac{1}{2}} \text{ veya } |(x, y)|^2 \leq (x, x)(y, y) \text{ 'dir.}$$

$(x, Ay) \leq (x, Ax)^{\frac{1}{2}}(y, Ay)^{\frac{1}{2}}$ 'dir. (ispat için $(\alpha x - \beta y, A(\alpha x - \beta y))$ alınıp Cauchy-Schwarz eşitsizliği uygulanır.)

Pozitif tanımlı matrisler için Cauchy-Schwarz eşitsizliğini kullanırsak,

$$\begin{aligned} (e^k, Ae^k) &= (Q_k(A)e^0, Ae^k) \leq (Q_k(A)e^0, Q_k(A)e^0)^{\frac{1}{2}}(Ae^k, Ae^k)^{\frac{1}{2}} \\ (e^k, Ae^k) &\leq (Q_k(A)e^0, AQ_k(A)e^0) \end{aligned} \quad (4.30)$$

Bu eşitsizlikte $Q_k(A)$ 'yı öyle seçelim ki (4.30)'nin sağ tarafındaki ifade mümkün olduğu kadar küçük olsun. Saçılım Tasvir Teoremini (Spectral Mapping Theorem) kullanarak (4.30)'nin sağ tarafının minimum değişimini bulmaya çalışalım.

Hatırlatma: Saçılım Tasvir Teoremi (Spectral Mapping Theorem);

f , bir A matrisinin özdeğerlerini de içeren bir küme üzerinde analitik ise bu takdirde,

$$f(\Lambda(A)) = \Lambda(f(A)).$$

Eğer λ , A 'nın özdeğeri ise $f(\lambda)$ da $f(A)$ 'nın bir özdeğeridir. A matrisinin özdeğerleri $[a, b]$ aralığında olduğu için,

$$(Q_k(A)e^0, AQ_k(A)e^0) \leq \max_{a \leq \lambda \leq b} |Q_k(\lambda)|^2 (e^0, Ae^0) \quad (4.31)$$

$Q_k(\lambda)$ 'yı öyle seçeceğiz ki λ için $Q_k(\lambda)$ $[a, b]$ 'de çok küçük olacak.

Dik polinom özelliklerini kullanarak $Q_k(\lambda)$ polinomunu şöyle seçelim:

$$Q_k(\lambda) = \frac{T_k\left(\frac{b+a-2\lambda}{b-a}\right)}{T_k\left(\frac{b+a}{b-a}\right)},$$

Burada $T_k(x)$ k . dereceden Chebyshev Polinomu olup şöyle tanımlanır:

$$T_k(x) = \begin{cases} \cos(k \cos^{-1}(x)) & \text{eğer } |x| \leq 1 \\ [sgn(x)]^k \cosh(k \cosh^{-1}(x)) & \text{eğer } |x| \geq 1 \end{cases} \quad (4.32)$$

$\lambda \in [a, b]$ için $a > 0$ olduğundan $\left|\frac{b+a-2\lambda}{b-a}\right| < 1$ 'dir.

Ayrıca $|T_k(x)| \leq 1$, $x \in [-1, 1]$ için. Bundan dolayı:

$$\max_{a \leq \lambda \leq b} |Q_k(\lambda)| \leq \left[T_k\left(\frac{b+a}{b-a}\right) \right]^{-1} = \left[\cosh\left(k \cosh^{-1}\left(\frac{b+a}{b-a}\right)\right) \right]^{-1}$$

k arttıkça $\cosh\left(k \cosh^{-1}\left(\frac{b+a}{b-a}\right)\right)$ artar dolayısıyla (e^k, Ae^k) azalır.

$$\frac{b+a}{b-a} = \cosh c = \frac{e^c + e^{-c}}{2}$$

diyelim. e^c için çözersek,

$$e^c = \frac{\sqrt{b} + \sqrt{a}}{\sqrt{b} - \sqrt{a}}$$

bulunur.

$$\begin{aligned} \cosh kc &= \frac{e^{kc} + e^{-kc}}{2} = \frac{1}{2} \left(\frac{\sqrt{b} + \sqrt{a}}{\sqrt{b} - \sqrt{a}} \right)^k \left[1 + \left(\frac{\sqrt{b} - \sqrt{a}}{\sqrt{b} + \sqrt{a}} \right)^{2k} \right] \\ &\geq \frac{1}{2} \left(\frac{\sqrt{b} + \sqrt{a}}{\sqrt{b} - \sqrt{a}} \right)^k \end{aligned} \quad (4.33)$$

Bundan dolayı,

$$\max_{a \leq \lambda \leq b} |Q_k(\lambda)| \leq 2 \left(\frac{\sqrt{b} - \sqrt{a}}{\sqrt{b} + \sqrt{a}} \right)^k$$

olur. ■

Teorem (4.3.1) gösterir ki A matrisinin özdeğerleri $a/b \approx 1$ için Eşlenik Gradyan Metodu daha hızlı yakınsar.

4.4 ÖN KOŞULLU EŞLENİK GRADYAN METODU (PRECONDITIONED CONJUGATE GRADIENT METODU)

Eşlenik Gradyan Metodu'nun daha hızlandırılmıştır tekniğine Ön Koşullu Eşlenik Gradyan Metodu diyeceğiz. Bu metodun temel fikri,

$$Ax = \mathbf{b} \quad (4.33)$$

denklem sistemini,

$$B^{-1}AB^{-T}(B^T \mathbf{x}) = B^{-1}\mathbf{b} \quad (4.34)$$

denklem sistemi ile yer değiştirmektedir.

Burada $B^{-1}AB^{-T}$ matrisi Eşlenik Gradyan Metodu daha hızlı yakınsayacak şekilde inşa edilir. B matrisi öyle seçilir ki $B^{-1}\mathbf{y}$ ve $B^{-T}\mathbf{y}$ işlemleri kolay bir şekilde yapılır. $B^{-1}AB^{-T}$ matrisinin özdeğerleri A matrisinin özdeğerlerinden daha fazla kümelenmiş olsun. A matrisi simetrik ve pozitif tanımlı olduğu için öyle bir C matrisi vardır ki $A = CC^T$ ve B matrisi de C'ye yakın olacak şekilde seçilir. Not olarak şunu da söyleyelim ki A simetrik pozitif tanımlı ise $B^{-1}AB^{-T}$ de simetrik pozitif tanımlıdır.

Farz edelim ki;

$$\tilde{A}\tilde{\mathbf{x}} = \tilde{\mathbf{b}} \quad (4.35)$$

sistemine Eşlenik Gradyan Metodunu uygulayalım. Burada,

$$\tilde{A} = B^{-1}AB^{-T}, \quad \tilde{\mathbf{x}} = B^T \mathbf{x}, \quad \tilde{\mathbf{b}} = B^{-1}\mathbf{b}$$

(4.22)'dan,

$$\begin{aligned} \tilde{\mathbf{x}}^{k+1} &= \tilde{\mathbf{x}}^k + \alpha_k \tilde{\mathbf{p}}^k \\ \tilde{\mathbf{r}}^{k+1} &= \tilde{\mathbf{r}}^k - \alpha_k \tilde{A} \tilde{\mathbf{p}}^k \\ \tilde{\mathbf{p}}^{k+1} &= \tilde{\mathbf{r}}^{k+1} + \beta_k \tilde{\mathbf{p}}^k \end{aligned} \quad (4.36)$$

Burada,

$$\alpha_k = \frac{|\tilde{\mathbf{r}}^k|^2}{(\tilde{\mathbf{p}}^k, \tilde{A} \tilde{\mathbf{p}}^k)} \quad \text{ve} \quad \beta_k = \frac{|\tilde{\mathbf{r}}^{k+1}|^2}{|\tilde{\mathbf{r}}^k|^2}$$

Şimdi (4.36)'deki denklemleri $\tilde{\mathbf{x}}$ yerine \mathbf{x} cinsinden yazalım,

$$\mathbf{x}^k = B^{-T} \tilde{\mathbf{x}}, \quad \mathbf{p}^k = B^{-1} \tilde{\mathbf{p}}^k, \quad \mathbf{r}^k = B \tilde{\mathbf{r}}^k$$

olup, (4.36) denklemleri,

$$\begin{aligned} \mathbf{x}^{k+1} &= \mathbf{x}^k + \alpha_k \mathbf{p}^k \\ \mathbf{r}^{k+1} &= \mathbf{r}^k - \alpha_k A \mathbf{p}^k \\ \mathbf{p}^{k+1} &= M^{-1} \mathbf{r}^{k+1} + \beta_k \mathbf{p}^k \end{aligned} \quad (4.37)$$

Burada $M = BB^T$ 'dir ve

$$\alpha_k = \frac{(\mathbf{r}^k, M^{-1} \mathbf{r}^k)}{(\mathbf{p}^k, A \mathbf{p}^k)}, \quad \beta_k = \frac{(\mathbf{r}^{k+1}, M^{-1} \mathbf{r}^{k+1})}{(\mathbf{r}^k, A \mathbf{r}^k)}.$$

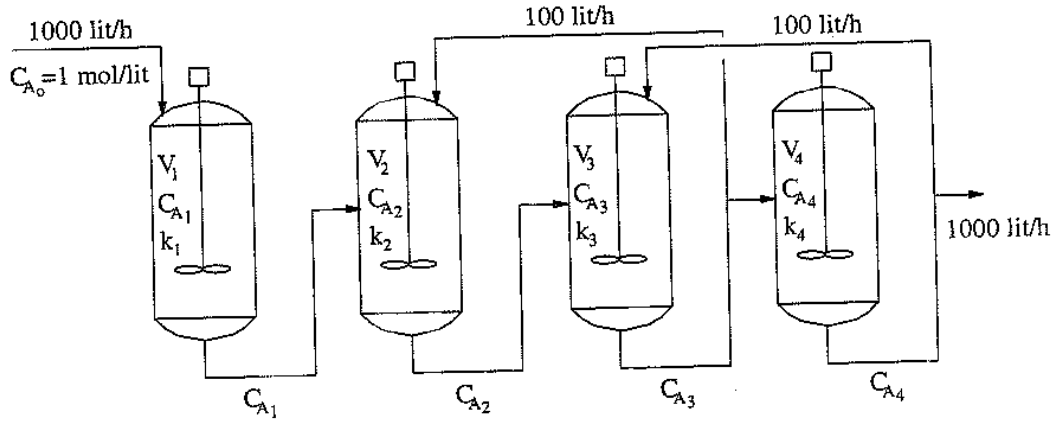
Ön koşullu olmasının etkisi arama doğrultusu \mathbf{p}^{k+1} ve α_k, β_k tanımlarında değişikliğe sebep olmuştur. Metodun etkili olması için, $\mathbf{r} = M\mathbf{z} = BB^T \mathbf{z}$ denklemi \mathbf{z} 'ye göre kolay çözülebilmelidir. B için genelde $B = \tilde{L}$ alınır. Burada \tilde{L} , A matrisinin alt üçgensel çarpanına yaklaşımdır. Öyle ki, $A = \tilde{L}\tilde{L}^T + N$. Burada N küçük bir matristir.

5. BULGULAR VE TARTIŞMA

5.1 ARDIŞIK METODLARIN UYGULAMALARI

Problem: Bir kimyasal reaksiyon Şekil 5.1'deki gibi düzenlenmiş dört adet sürekli karıştırılan kapalı havuz reaktörler dizisi içinde yer alsın.

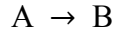
Şekil 5.1: Sürekli karıştırılan kapalı havuz reaktörleri dizisi.



Kaynak: Constantinides ve Mostoufi (1999, s. 114).

Kimyasal reaksiyon tipi,

$$k_i$$



birinci dereceden geri dönüşümü olmayan bir tepkimedir.

Her reaktör için sıcaklık koşuluna bağlı olan hız sabiti k_i değeri farklıdır. Ayrıca, her bir reaktörün V_i hacmi farklıdır. k_i ve V_i değerleri Tablo 5.1 de verilmektedir. Bu sistem için aşağıdaki varsayımlar kabul edilmiştir:

1. Sistem zamanla değişmemektedir.
2. Reaktörler sıvı fazdadır.
3. Sıvının yoğunluğu veya hacmi değişmemektedir.
4. Her bir reaktördeki A bileşeninin kaybolma oranı aşağıdaki formülle verilmektedir.

$$R_i = V_i k_i c_{A_i} \text{ mol/h}$$

Tablo 5.1: Her reaktör için hacim ve hız sabiti değerleri.

<i>Reaktör</i>	$V_i(L)$	$k_i(h^{-1})$	<i>Reaktör</i>	$V_i(L)$	$k_i(h^{-1})$
1	1000	0.1	3	100	0.4
2	1500	0.2	4	500	0.3

Kaynak: Constantinides ve Mostoufi (1999, s. 114).

Aşağıdaki soruları cevaplandıracağız:

- Her bir reaktör için malzeme-denge denklemini kurun. Bu denklemlerin tipini belirleyiniz.
- Her bir reaktörden çıkan konsantrasyon miktarını (c_{A_i}) miktarını çözmek için en iyi hangi metodu önerirsiniz?
- a şikkındaki denklemleri çözmek ve her bir reaktörden çıkan konsantrasyonu bulmak için MATLAB kodunu yazınız.
- Problemi Jacobi, Gauss-Seidel, S.O.R., Eşlenik Gradyan ve Ön Koşullu Eşlenik Gradyan metotlarını kullanarak sonuçları yorumlayın.

Çözüm Metodu:

- Her bir reaktör için genel zamana bağlı durumu gösteren malzeme dengesi şöyledir:

$$\text{Girdi} = \text{Çıktı} + \text{Reaksiyon Sebebiyle Kaybolma} + \text{Birikim}$$

Sistem durgun durumda olduğu için birikim terimini sıfırdır; bundan dolayı, malzeme-denge denklemi şu duruma indirgenir:

$$\text{Girdi} = \text{Çıktı} + \text{Reaksiyon Sebebiyle Kaybolma}$$

Bu denge formülü her bir reaktöre uygulandığında aşağıdaki denklemler elde edilir.

$$(1000)(1) = 1000 c_{A_1} + V_1 k_1 c_{A_1}$$

$$1000 c_{A_1} + 100 c_{A_3} = 1100 c_{A_2} + V_2 k_2 c_{A_2}$$

$$1100 c_{A_2} + 100 c_{A_4} = 1200 c_{A_3} + V_3 k_3 c_{A_3}$$

$$1100 c_{A_3} = 1100 c_{A_4} + V_4 k_4 c_{A_4}$$

k_i ve V_i değerlerini yerine koyup bu denklemleri düzenlersek;

$$\begin{aligned}
1100 c_{A_1} &= 1000 \\
1000 c_{A_1} - 1400 c_{A_2} + 100 c_{A_3} &= 0 \\
1100 c_{A_2} - 1240 c_{A_3} + 100 c_{A_4} &= 0 \\
1100 c_{A_3} - 1250 c_{A_4} &= 0
\end{aligned} \tag{5.1}$$

Elde edilen denklem sistemleri lineerdir. Görülüyor ki; bu denklem sistemi köşegensel dominanttır.

b. Jakobi Metoduyla karşılaştırıldığında bu köşegensel dominant denklem sistemi için Gauss- Seidel Metodu daha iyi bir çözüm olarak görünmesine rağmen; MATLAB daki işlemler matrisler üzerine olduğu için matris formundaki Jacobi Metodu, Gauss- Seidel Metodundan oldukça hızlıdır.

c. Bir sonraki bölümde Jakobi Metodu açıklanacaktır. Jakobi Algoritmasını başlatmak için c_{A_1} den c_{A_4} e kadarki bilinmeyenlerin başlangıç değerine ihtiyaç duyulur. Bu denklem sistemi köşegensel dominant olduğu bilinmeyenler için herhangi bir başlangıç değeri çözümü verecektir. Buna karşın; $c_{A_0} = 1.0$ olduğu gerçeğinden yola çıkarak başlangıç değerlerinin dört bilinmeyen için 0.6 almamız uygundur. Jacobi Metodunun yakınsadığını test etmek için iki farklı durum ele alacağız. İlk durum için başlangıç değerleri olarak 0.6 ; ikinci durumda da 100 alacağız.

Program Tanımı: Jacobi İterasyon Metodu ile lineer denklem sistemlerini çözmek için adı Jacobi.m olan bir MATLAB fonksiyonu yazdık. Bu fonksiyon için girdileri; katsayı matrisi, sağ taraf sabit vektörü ve bütün bilinmeyenler için başlangıç değerlerini içeren bir vektör alındı. Yakınsama kriteri olarak

$\left| x_j^{(k)} - x_j^{(k-1)} \right| < 10^{-6}$ alındı. Programdaki bir sonraki adım değiştirilmiş katsayı matrisi $(D^{-1} A - I)$ ve değiştirilmiş sabit vektör olarak da $((D^{-1} b))$ inşa edildi.

Durum 1: Başlangıç değeri 0.6 için;

Tablo 5.2: Başlangıç değeri 0.6 için çözüm.

İterasyon Sayısı	c_{A_1}	c_{A_2}	c_{A_3}	c_{A_4}
1	0.909091	0.471429	0.580645	0.528
2	0.909091	0.690825	0.460783	0.510968
3	0.909091	0.682264	0.654036	0.405489
4	0.909091	0.696068	0.637935	0.575552
5	0.909091	0.694917	0.663895	0.561383
6	0.909091	0.696772	0.661732	0.584227
7	0.909091	0.696617	0.665219	0.582324
8	0.909091	0.696866	0.664928	0.585393
9	0.909091	0.696846	0.665397	0.585137
10	0.909091	0.696879	0.665358	0.585549
11	0.909091	0.696876	0.665421	0.585515
12	0.909091	0.696881	0.665416	0.58557
13	0.909091	0.69688	0.665424	0.585566
14	0.909091	0.696881	0.665423	0.585573
15	0.909091	0.696881	0.665424	0.585573
16	0.909091	0.696881	0.665424	0.585574

$$c_{A_1} = 0.9091$$

$$c_{A_2} = 0.6969$$

$$c_{A_3} = 0.6654$$

$$c_{A_4} = 0.5856$$

sonuçları elde edildi.

Durum 2: Başlangıç değeri 100 için;

Tablo 5.3:Başlangıç değeri 100 için çözüm.

İterasyon Sayısı	c_{A_1}	c_{A_2}	c_{A_3}	c_{A_4}
1	0.909091	78.5714	96.7742	88
2	0.909091	7.56179	76.7972	85.1613
3	0.909091	6.13487	13.5759	67.5816
4	0.909091	1.61906	10.8923	11.9468
5	0.909091	1.42738	2.39971	9.58527
6	0.909091	0.820759	2.03923	2.11175
7	0.909091	0.79501	0.898394	1.79452
8	0.909091	0.713522	0.84997	0.790587
9	0.909091	0.710063	0.69672	0.747973
10	0.909091	0.699116	0.690215	0.613113
11	0.909091	0.698652	0.669628	0.607389
12	0.909091	0.697181	0.668755	0.589273
13	0.909091	0.697119	0.665989	0.588504
14	0.909091	0.696921	0.665872	0.586071
15	0.909091	0.696913	0.6655	0.585967
16	0.909091	0.696886	0.665485	0.58564
17	0.909091	0.696885	0.665435	0.585626
18	0.909091	0.696882	0.665433	0.585583
19	0.909091	0.696882	0.665426	0.585581
20	0.909091	0.696881	0.665426	0.585575
21	0.909091	0.696881	0.665425	0.585575

$$c_{A_1} = 0.9091$$

$$c_{A_2} = 0.6969$$

$$c_{A_3} = 0.6654$$

$$c_{A_4} = 0.5856$$

sonuçları elde edildi.

Sonuçların Tartışılması : İlk durumda c_{A_1} den c_{A_4} e kadarki bilinmeyenlerin başlangıç değeri olarak 0.6 aldık ve Jacobi Metodu çözüme 16 adımda yakınsadı ve yakınsama kriteri 10^{-6} idi. İkinci durumda başlangıç değeri olarak 100 alındı. Bu sefer birinci durumdaki çözüme 21 adımda ulaşıldı.

d. Aşağıdaki tablodan da görüleceği üzere ön koşullu eşlenik gradyan metodu(ÖEGM) en tutarlı yaklaşık çözümü en az iterasyon sayısı ile vermektedir. Ancak şunu belirtmek gerekir ki ÖEGM için matrisin simetrik olması gerekmektedir. Halbuki ele aldığımız A matrisi simetrik değildir. Bundan dolayı $Ax=b$ denklemi yerine $A^T Ax = A^T b$ denklemini ele alarak simetrik bir katsayı matrisi elde edilir.

ÖEGM daha çok sıfırı bol ve pozitif tanımlı büyük matrislere uygulanır. Böyle sistemler adi diferansiyel denklemlerin sınır değer problemlerinde karşımıza çıkar. Bu sistemlerde önkoşul matrisi A matrisinin Choleski çarpanlarına ayırma metodundaki L matrisine tekabül eder. Eşlenik Gradyan Metodu için daha fazla bilgiye Kelley' in Iterative Method For Linear and Nonlinear Equations adlı kitabından ulaşılabilir.

Şimdi (5.1) denkleminin katsayılar matrisi ele alalım:

$$A = \begin{bmatrix} 1100 & 0 & 0 & 0 \\ 1000 & -1400 & 100 & 0 \\ 0 & 1100 & -1240 & 100 \\ 0 & 0 & 1100 & -1250 \end{bmatrix}$$

Koşul sayısı $K_\infty(A) = Cond(A) = \|A\|_\infty \|A^{-1}\|_\infty$ idi. $K_\infty(A) = 4.9024$ ve başlangıç değeri için 0.6 alalım.

i. Jacobi Çözümünü (c) şıkkında çözmüştük.

ii. Gauss-Seidel Çözümü:

Gauss-Seidel için iterasyon sayısı=6

$$c_{A_1} = 0.9091$$

$$c_{A_2} = 0.6969$$

$$c_{A_3} = 0.6654$$

$$c_{A_4} = 0.5856$$

∞ -norma göre mutlak hata=1.304436e-07

iii. S.O.R Çözümü:

S.O.R. Çözümü için iterasyon Sayısı=13

$$c_{A_1} = 0.9091$$

$$c_{A_2} = 0.6969$$

$$c_{A_3} = 0.6654$$

$$c_{A_4} = 0.5856$$

∞ -norma göre mutlak hata= 5.4417e-07

Optimum w parametre değeri: 1.25

iv. Eşlenik Gradyan Çözümü:

Eşlenik Gradyan için iterasyon sayısı:13

$$c_{A_1} = 0.90909091$$

$$c_{A_2} = 0.69688100$$

$$c_{A_3} = 0.66542454$$

$$c_{A_4} = 0.58557367$$

∞ -norma göre mutlak hata= 0.00000

v. Ön koşullu Eşlenik Gradyan Çözümü:

Ön Koşullu Eşlenik Gradyan kullanabilmek için A matrisinin simetrik ve pozitif tanımlı olması gerekir.

Ön koşullu Eşlenik Gradyan iterasyon sayısı: 4

$$c_{A_1} = 0.90909091$$

$$c_{A_2} = 0.69688100$$

$$c_{A_3} = 0.66542454$$

$$c_{A_4} = 0.58557367$$

∞ -norma göre mutlak hata= 4.79873e-10

Tablo 5.4: Jacobi, Gauss- Seidel, S.O.R., EGM ve ÖEGM' nin karşılaştırılması I.

Metot	İterasyon sayısı	c_{A_1}	c_{A_2}	c_{A_3}	c_{A_4}	∞ -norma göre M.H.
Jacobi	16	0.9091	0.6969	0.6654	0.5856	0.00000
Gauss-Seidel	6	0.9091	0.6969	0.6654	0.5856	1.304436e-07
S.O.R	13	0.9091	0.6969	0.6654	0.5856	5.4417e-07
EGM	13	0.90909091	0.69688100	0.66542454	0.58557367	0.00000
ÖEGM	4	0.90909091	0.69688097	0.66542454	0.58557360	4.79873e-10

Çarpıcı bir örnek teşkil etmesi amacıyla simetrik, pozitif tanımlı ancak kötü koşullu yani koşul sayısı çok yüksek bir matrisi içeren aşağıdaki lineer denklem sistemini ele alacağız.

$$\mathbf{A} = \begin{bmatrix} 0.2 & 0.1 & 1 & 1 & 0 \\ 0.1 & 4 & -1 & 1 & -1 \\ 1 & -1 & 60 & 0 & -2 \\ 1 & 1 & 0 & 8 & 4 \\ 0 & -1 & -2 & 4 & 700 \end{bmatrix} \text{ ve } \mathbf{b} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}$$

A katsayılar matrisi ve b sonuç matrisi olmak üzere $Ax=b$ lineer denklem sistemini çözersek çözüm:

$$x^* = (7.859713071, 0.4229264082, -0.07359223906, -0.5406430164, 0.01062616286)^t$$

Koşul sayısı= $K_\infty(A) = Cond(A) = \|A\|_\infty \|A^{-1}\|_\infty$ idi.

A matrisi için $K_\infty(A) = 13961.71$ Tolerans değerini 0.01 alalım ve Jacobi, Gauss-Seidel, S.O.R. için ($\omega = 1.25$)iteratif metotlarından ve de Eşlenik Gradyan metodu elde edilen sonuçları karşılaştıracamız.

Tablo 5.5: Jacobi, Gauss- Seidel, S.O.R., EGM ve ÖEGM' nin karşılaştırılması II.

<i>Metot</i>	<i>İterasyon Sayısı</i>	$x^{(k)}$	$\ x^* - x^{(k)}\ _{\infty}$
Jacobi	49	(7.86277141, 0.42320802, -0.07348669, -0.53975964, 0.01062847) ^t	0.00305834
Gauss-Seidel	15	(7.83525748, 0.42257868, -0.07319124, -0.53753055, 0.01060903) ^t	0.02445559
S.O.R. ($\omega = 1.25$)	7	(7.85152706, 0.42277371, -0.07348303, -0.53978369, 0.01062286) ^t	0.00818607
Eşlenik Gradyan	5	(7.85341523, 0.42298677, -0.07347963, -0.53987920, 0.008628916) ^t	0.00629785
Önkoşullu Eşlenik Gradyan	4	(7.85968827, 0.42288329, -0.07359878, -0.54063200, 0.01064344) ^t	0.00009312

Kaynak: Burden ve Faires (2011, s. 477).

6. SONUÇ

Bu çalışmada katsayılar matrisi simetrik ve pozitif tanımlı olan lineer denklem sistemlerini çözmek için bir grup ardışık metot ve Eşlenik Gradyan Metot (EGM) ele alınmıştır. Bu metotlardan özellikle EGM incelenmiştir. Eşlenik Gradyan metodu; Jacobi, Gauss-Seidel, S.O.R. (successive over relaxation-ardışık aşırı rahatlamış) gibi ardışık yöntemlerle hem işlem sayısı hem de adım sayısı açısından karşılaştırılmıştır. Bir kimyasal reaksiyon problemini ele alıp, bu problem için malzeme-denge denklemleri elde edildi. Bu elde edilen denklemler bir lineer denklem sistemi haline indirildi. Çözüm olarak Jacobi , Gauss- Seidel, S.O.R. ardışık metotlar uygulanıp; EGM ve Ön Koşullu Eşlenik Gradyan Metodu (ÖEGM) karşılaştırılmıştır. Lineer denklem sisteminin katsayılar matrisi A 'nın simetrik ve pozitif tanımlı olması durumunda Ön Koşullu Eşlenik Gradyan Metodunun diğer yöntemlerden daha az hata ile daha az adımda sonuca vardığı gösterilmiştir.

7. KAYNAKÇA

Kitaplar

- Amirali G., Duru H., Nümerik Analiz, Pegem Yayıncılık, 2002.
- Bayram M., Nümerik Analiz, Sürat Üniveriste Yayınları, 2002.
- Burden, R.L., Faires, J.D., Numerical Analysis, Brooks Cole, California, 2000.
- Charles F. Van Loan, Introduction To Scientific Computing A Matrix-Vektor Approach Using Matlab, Prentice-Hall, NewJersey, 2000.
- Constantinides, A., Mostoufi, Navid, Numerical Methods For Chemical Engineers With MATLAB Applications, Prentice Hall, NewJersey, 1999.
- Epperson, J., Introduction To Numerical Analysis, John Wiley And Sons, 2001.
- Kelly, C.T., Iterative Methods For Linear And Nonlinear Equations, SIAM Philadelphia, 1995
- Kreyszig, E., Introductory Functional Analysis With Applications, John Wiley And Sons, 1987.
- Musayev, B., Alp, M., Fonksiyonel Analiz, Tek Ağaç Eylül Yayınevi, 1999.
- Ortega, J. M. Numerical Analysis; A Second Course, Academic Press, New York, 1972,
- Strikwerda, John C., Finite Difference Schemes And Partial Differential Equations, Wadsworth & Brooks/Cole Advanced Books, California, 1989.
- Şuhubi, E., Fonksiyonel Analiz , İTÜ Vakfı Yayınları, 2001.
- Yıldız, A., Eröz, M., Fonksiyonel Analiz, Saü Yayınları, 2009.
- Yosida, K., Functional Analysis, Springer-Verlag, 1968.
- Young, D. M., Iterative Solution Of Large Lineer Systems, Acedemic Press, New York, 1971.

Sürelî Yayınlar

Cipra, B. A., The Best of the 20th Century: Editors Name Top 10 Algorithms SIAM News, Volume 33, Number 4.

Çelik, N., Bilişim Dergi (iübük elektronik dergisi), Haziran 2009, sayı 3.

Hestenes, M. L., Stiefel, E., 1952. Methods of Conjugate Gradients for Solving Linear Systems. J. of Research of the National Bureau of Standards 49 (6).



Diğer Yayınlar

Fekete, Imre, Numerical Methods For The Linear Algebraic Systems With M-matrices, Yüksek Lisans Tezi. Budapest: Universitas Budapestinensis De Rolando Eötvös Nominata, 2010.

Topçu, A., Eskişehir Osmangazi Üniversitesi, Bilgisayar Destekli Nümerik Analiz Ders notları, 2014

<http://www.maa.org/press/periodicals/loci/joma/iterative-methods-for-solving-iaxi-ibi-exercises-part-1-jacobi-and-gauss-seidel-methods>

<http://www.maa.org/press/periodicals/loci/joma/iterative-methods-for-solving-iaxi-ibi-analysis-of-jacobi-and-gauss-seidel-methods>

<http://www.maa.org/press/periodicals/loci/joma/iterative-methods-for-solving-iaxi-ibi-the-sor-method>

EKLER



Ek 1: Ön Koşullu Eşlenik Gradyan Metodu İçin MATLAB Komutu
[X, flag, resres, iter] = pcg($A^T * A, A^T * b, 1e - 6, 100$)

% Burada

% Tolerans: $1e - 6$

% Maksimum İterasyon Sayısı: 100.

