

## ÖZET

### ULV AYRIŞIMI VE UYGULAMALARI

KOCAOĞLU, Selçuk

Kırıkkale Üniversitesi

Fen Bilimleri Enstitüsü

Matematik Ana Bilim Dalı, Yüksek Lisans Tezi

Danışman: Yrd. Doç. Dr. Hasan ERBAY

Haziran 2006, 32 sayfa

Ortogonal dönüşüm tabanlı matris algoritmaları, matris hesaplamalarında önemli bir rol oynar. Bunun nedenlerinden bazıları: (1) Ortogonal dönüşümler sayısal olarak kararlıdır ki, matrisin sayısal rankı istenildiğinde özellikle önemlidir, (2) Bu dönüşümler 2-normu korur, böylece problemleri sadeleştirmede kullanılırlar, (3) Ortogonal dönüşüm tabanlı matris ayrışmaları genellikle önemli derecede az işlemle veri ekleme/çıkarma yapan güvenilir usullerdir, (4) Bu ayrışmalar noise supression teknikleri ve diğer sinyal işleme uygulamalarında önemli bir rol oynayan matris üzerinde tanımlanmış belirli altuzaylar hakkında önemli bilgiler verir.

Zaman değişimli problemlerde, çok fazla veri ekleme/çıkarma işleminden sonra tekil değer ayrışımını (SVD) hesaplamak alternatif ayrışmaları çekici kılan hesapsal olarak pahalı bir süreçtir. Bu tezde SVD'ye alternatif olan ULVD üzerine odaklanacağız.

Bu tezde bařlangıçtan ULVD'yi hesaplayan özyineli bir algoritma sunacađız. Bu algoritma mevcut algoritmaların tersine matrisin rankının küçük veya büyük olmasını ayırmaz. Algoritmamızın diđer tüm SVD algoritmalarından daha hızlı olduğunu gösterilmiştir.

**Anahtar Kelimeler:** Matris hesaplaması, Matris ayrışımı, Tekil deđer ayrışımı, SVD, ULV ayrışımı, ULVD.

## ABSTRACT

### ULV DECOMPOSITION AND ITS APPLICATIONS

KOCAOĞLU, Selçuk

Kırıkkale University

Graduate School Of Natural and Applied Sciences

Department of Mathematics, M. Sc. Thesis

Supervisor : Asst. Prof. Dr. Hasan ERBAY

June 2006, 32 pages

Matrix algorithms based on orthogonal transformations play an important role in matrix computations. Some of the reasons for this are: (1) Orthogonal transformations are numerically stable, which is particularly important when the numerical rank of a matrix is an issue, (2) These transformations preserve the two-norm, thus, can be used to simplify problems, (3) Matrix decompositions based on orthogonal transformations are often easy to update/downdate in a reliable fashion with considerable less computation, (4) These decompositions can yield information about certain subspaces defined on the matrix which play an essential role in noise suppression techniques and other signal processing applications.

For the time varying applications, computing the singular value decomposition (SVD) after each update/downdate is computationally expensive

process which makes alternative decompositions attractive. This thesis focuses on ULV decomposition (ULVD), an alternative to the SVD.

In this thesis we present a recursive algorithm that computes ULVD from scratch. Unlike current algorithms it does not distinguish whether the matrix is high or low rank. It is shown that our algorithm is faster than all stable SVD algorithms.

**Key Words:** Matrix computation, Matrix decomposition, Singular value decomposition, SVD, ULV decomposition, ULVD.

## İÇİNDEKİLER

ÖZET .....	i
ABSTRACT .....	iii
İÇİNDEKİLER .....	v
ÇİZELGELER DİZİNİ .....	vii
ŞEKİLLER DİZİNİ .....	viii
1. GİRİŞ .....	1
1.1. Genel Bakış.....	2
1.2. Literatürün Gözden Geçirilmesi.....	2
1.3. Çalışmanın Amacı.....	3
2. MATERYAL VE YÖNTEM .....	4
2.1. Lineer cebirin gözden geçirilmesi.....	4
2.1.1. Vektörler ve matrisler.....	4
2.1.2. Matrisin Altuzayları.....	6
2.2. Normlar.....	8
2.2.1. Vektör Normları.....	8
2.2.2. Matris Normları.....	9
2.3. Ortogonal Dönüşümler.....	12
2.3.1. Givens Dönüşümü.....	15
2.3.2. Householder Dönüşümü.....	17
3. ARAŞTIRMA BULGULARI .....	21
3.1. Rankı Veren QR Ayrışımı.....	21
3.2. Tam Ortogonal Ayrışımalar.....	22
3.2.1. Tekil Değer Ayrışımı.....	22

3.3. Sayısal Rank ve Sayısal Tekil Altuzaylar.....	25
4. TARTIŞMA VE SONUÇ .....	27
4.1. Özyineli ULVD Algoritması.....	27
KAYNAKLAR .....	34

## ÇİZELGELER DİZİNİ

ÇİZELGE	
2.1. Givens Algoritması .....	17
2.2. Householder Algoritması .....	20
4.3. ULVD Algoritması .....	28
4.4. Özyineli ULVD Algoritması .....	31

## ŞEKİLLER DİZİNİ

ŞEKİL	
2.1. Düzlem Döndürmesi .....	16
2.2. Householder Yansıması .....	19



## 1. GİRİŞ

Ortogonal dönüşümlere dayanan matris algoritması, matris hesaplamalarında önemli bir rol oynar. Bunun için birçok sebep vardır; birincisi, ortogonal dönüşümler sayısal olarak kararlıdır ki bu matrisin sayısal rankı konu olduğunda gerçekten önemlidir. İkincisi, bu dönüşümler  $2$ -normu korur, böylece bu, problemi basitleştirmede kullanılabilir. Üçüncüsü, ortogonal dönüşümlere temel olan matris ayrışmalarının, oldukça az işlem ile güvenilir bir şekilde satır eklenmesi/çıkarılması genellikle kolaydır. Son olarak bu ayrışmalar, noise supression tekniklerde ve diğer sinyal işleme uygulamalarında zaruri bir rol oynayan matris üzerinde tanımlanmış bazı altuzaylar hakkında bilgi verebilir.

Genellikle kullanılan ortogonal dönüşümler, tekil değer ayrışmalarıdır (SVD) [8], [3], [12]. SVD güvenli bir biçimde bir matrisin içindeki near-rank hatalarını arar ve bütün gerekli altuzay bilgilerini verir. Çünkü SVD algoritması gerçekten güvenilir ve sayısal olarak kararlıdır, en küçük kareler ve toplam en küçük kareler değerler arası tayininde oldukça yaygın uygulamalar olarak kullanılır [13], [14].

SVD çok kuvvetli bir araç olmakla beraber, bazı sakıncaları vardır. Örneğin eski verinin silindiği ve/vaya yeni verinin eklendiği zaman değişimli problemler için, SVD ayrışmayı hesapsal olarak talep eder. Bu, alternatif çözümleri çekici hale getirir.

Alternatiflerden bir tanesi rankı ortaya çıkaran QR (RRQR) ayrışımıdır [15], [16]. QR kolay değiştirilebilen bir ayrışımıdır. Bununla birlikte, kirlilik altuzayı temel gerektiren uygulamalar için daha az kullanışlıdır.

Rankı ortaya çıkaran çift yönlü ortogonal ayrışmalar URVD ve ULVD, sayısal sıralamalar ve arzu edilen altuzay için güvenilir hesaplamalar sağlayan diğer umut verici alternatiflerdir [11]. Bu ayrışmalar ilk olarak Faddeev, Kublanowskaya ve Faddeeva [17] ve Hanson ve Lawson [18] tarafından tartışılmıştır. Bir zamanlar tamamen unutulmuş olsalar da, URVD ve ULVD algoritmalarını ilk olarak ortaya çıkaran Stewart tarafından araştırmacıların dikkatine sunulmuştur [9], [10]. Bu tezde ULVD üzerine yoğunlaşılacaktır.

## **1-1 Genel Bakış**

Tez boyunca iki ULVD ilişkili algoritma anlatılacaktır. Bu problemlerde, veri matrislerinin sayısal rankı ve küçük tekil değerlerle ilişkilendirilmiş altuzaylar için temeller aranmıştır. Bu altuzaylar, noise altuzay olarak ta adlandırılır. İlk olarak başlangıçtan ULVD'yi hesaplayan bir tekrarlamalı algoritma önerilir. Şimdiki algoritmalarından farklı olarak, anlatılacak olan algoritma, matrisin yüksek veya düşük ranklı olup olmadığını ayırt etmez. İlk olarak başlangıç matrisini, ortogonal matrisleri kullanarak bidiagonalleştirir. Sonrasında bidiagonal matrisi iki hemen hemen eşit bloğa böler. Daha sonra her bloğun ULVD'sini hesaplayarak, blokları birleştirir. Bu, verilecek algoritmanın diğer tüm SVD algoritmalarından daha hızlı olduğunu göstermiştir.

## **1-2 Literatürün Gözden Geçirilmesi**

ULV ayrışımı, SVD gibi, aynı rank bilgisini ve yaklaşık aynı altuzay bilgisini sağlar. Oldukça etkili ve kesin biçimde değiştirilebilir. Stewart bir ULVD

algoritması önermiştir [9]. Stewart'ın algoritması matrisi ortogonal dönüşümlerden üçgensel formlara dönüştürür. Bundan sonra tekil değerleri çözümlene başarlınca kadar tek tek kazıyarak yukarı doğru taşır. Bu algoritma yüksek ranklı matrisler için daha uygundur. Sonrasında Fierro ve Hansen [19] başka bir ULVD algoritması oluşturmuşlardır. Stewart'ın algoritmasının tersine en büyüğüyle başlayarak daha büyük değerleri tek tek kazıyarak üçgensel matriste yukarı doğru taşır.

### **1-3 Çalışmanın Amacı**

Tezin kalan kısmı şu şekilde yapılandırılmıştır. İkinci bölümdeki lineer cebir ve nümerik analizin anlatımının ardından üçüncü bölümde rankı veren ortogonal ayrışmaları tartışacağız. Daha sonra ULV ayrışımı ile ilgili algoritmalar vereceğiz ve SVD algoritmasından daha hızlı olduğunu göstereceğiz.

## 2. MATERYAL VE YÖNTEM

Bu ünite temel notasyonlara ve cebirsel kavramlara ayrılmıştır. İleriki bölümlerde bu incelemede sıklıkla kullanılacak olan lineer cebir ve nümerik analizin temel kavramlarından kısaca bahsedilecektir.

### 2.1 Lineer cebirin gözden geçirilmesi

#### 2.1.1 Vektörler ve matrisler

**Tanım 1:** Bir reel  $n$ -vektör  $x$ ,  $i=1,2,\dots,n$  için  $x_i \in \mathbb{R}$  olmak üzere,  $n$  tane reel sayının

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad (1.1)$$

formunda sıralanmasıdır.

Skalar  $x_i$  ler  $x$ 'in bileşenleri olarak adlandırılır. Tüm reel  $n$  vektörlerinin kümesi  $\mathbb{R}^n$  olarak gösterilir. Tez boyunca vektörler kalın, küçük harfle gösterilecektir.

**Tanım 2:** Tüm elemanları sıfır olan vektöre *sıfır vektörü* denir ve boyutundan bağımsız olarak  $0$ 'la gösterilir.  $i$ -inci bileşeni 1 olan ve diğer bileşenleri 0 olan vektöre  $i$ -inci *birim vektör* adı verilir ve  $e_i$  şeklinde gösterilir.

**Tanım 3:** Bir  $m \times n$  reel  $A$  matrisi  $m$  satırlı,  $n$  sütunlu dikdörtgensel reel sayı

dizisidir. Yani

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \quad (1.2)$$

$i = 1, 2, \dots, m$  ve  $j = 1, 2, \dots, n$  için  $a_{ij} \in \mathbb{R}$  biçimindedir. Skalar  $a_{ij}$ ,  $\mathbf{A}$ 'nın elemanlarıdır.  $m \times n$  reel matrisinin tüm kümeleri  $\mathbb{R}^{m \times n}$  şeklinde gösterilir. Tez boyunca matrisler büyük harfle gösterilecektir.

Verilen bir  $m \times n$  matrisi için  $\mathbf{a}_i$  vektörü  $\mathbf{A}$ 'nın  $i$ -inci sütunu olmak üzere genellikle  $\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n)$  yazılımını kullanılacaktır. Bununla birlikte  $a_{ij}$   $i = 1, 2, \dots, m$ ,  $j = 1, 2, \dots, n$ ,  $\mathbf{A}$  matrisinin elemanları olmak üzere çoğunlukla  $\mathbf{A} = (a_{ij})$  notasyonu kullanılacaktır.

Matrisler ve alt matrisler için MATLAB işaretleme sistemini de kullanacağız. Örneğin  $\mathbf{A}$ 'nın  $i$ 'den  $j$ 'ye kadar satırlarını,  $k$ 'dan  $l$ 'ye kadar sütunlarını alıp elde edeceğimiz alt matrisi  $\mathbf{A}(i:j, k:l)$  şeklinde gösterilecek.  $\mathbf{A}(:, k:l)$  şeklindeki gösterim ise  $\mathbf{A}$  matrisinin  $k$  dan  $l$  ye tüm sütunlarını işaret edecek. Benzer bir şekilde  $\mathbf{A}(i: j, :)$   $\mathbf{A}$  matrisinin  $i$ 'den  $j$ ' ye tüm satırlarını gösterecek.

Bu tez boyunca dikkatimizi yalnız reel matrisler üzerinde toplayacağız. Bu bölümün devamında matrislerin diğer çeşitleri hakkında kısaca bilgi vermektir.

**Tanım 4:** Bir  $m \times n$  boyutlu  $\mathbf{A}$  matrisi  $m = n$  ise *karedir*. Bu durumda  $\mathbf{A}$  matrisi için *n-inci derecedendir* denir.

**Tanım 5:** Eğer bir matrisin tüm girdileri sıfır ise *sıfır matrisi* olarak adlandırılır ve boyutuna bakılmaksızın  $\mathbf{0}$  ile gösterilir.

**Tanım 6:** Bir kare  $I = (l_{ij})$  matrisi

$$l_{ij} = \begin{cases} 1 & \text{eğer } i = j \\ 0 & \text{eğer } i \neq j \end{cases}$$

şeklinde tanımlanıyorsa *birim matris* adını alır.

**Tanım 7:**  $A$   $m \times n$  boyutlu bir matris olsun.  $A$ 'nın devriği, elemanları  $b_{ij} = a_{ji}$   $i = 1, 2, \dots, n$ , ve  $j = 1, 2, \dots, m$  şeklinde verilen  $n \times m$  boyutlu  $B$  matrisidir ve

$$B = A^T$$

şeklinde yazılır.

### 2.1.2 Matrisin Altuzayları

**Tanım 8:** Bir vektörler kümesi olan, içinde vektör toplaması, skalar vektör çarpımı gibi işlemlerin tanımlandığı  $V$  vektör uzayını ele alalım. Eğer  $V$ 'nin alt kümesi olan  $W$ ,  $V$ 'nin yapabildiği vektör işlemlerini kullanabilen bir vektör uzayı ise  $W$ 'ye  $V$ 'nin *altuzayı* denir.

Verilen bir  $A \in \mathbb{R}^{m \times n}$  matrisinin en önemli iki alt uzayı, değer ve çekirdek alt uzaylarıdır.

**Tanım 9:** Bir  $A$  matrisinin *değer altuzayı* aşağıdaki şekilde tanımlanır.

$$\text{range}(A) = \{y = Ax \mid x \in \mathbb{R}^n\} \quad (1.3)$$

**Tanım 10:** Bir  $A$  matrisi için  $Ax = 0$  eşitliğinin çözüm kümesi  $A$ 'nın *çekirdek alt uzayını* verir ve

$$\text{ker}(A) = \{x \in \mathbb{R}^n \mid Ax = 0\} \quad (1.4)$$

şekilde gösterilir.

**Tanım 11:** Eğer  $\sum_{i=1}^n \alpha_i x_i = 0$  ,  $\alpha_1 = \alpha_2 = \dots = \alpha_n = 0$  olmasını gerektiriyor ise

$\{x_1, x_2, \dots, x_n\} \in \mathbb{R}^m$  vektör kümesi *lineer bağımsızdır* denir.  $\mathbf{V}$  vektör uzayının, azami sayıda lineer bağımsız vektörlerinin kümesine  $\mathbf{V}$  'nin bir *bazı* denir.  $\mathbf{V}$  'nin bazı içindeki vektör sayısına  $\mathbf{V}$  'nin *boyu* denir ve  $\dim(\mathbf{V})$  şeklinde gösterilir.

**Tanım 12:**  $\mathbf{A} \in \mathbb{R}^{m \times n}$  matrisinin rankı

$$\text{rank}(\mathbf{A}) = \dim(\text{range}(\mathbf{A})) \quad (1.5)$$

olarak tanımlanır ve eğer  $\text{rank}(\mathbf{A}) = \min(m, n)$  ise  $\mathbf{A}$  için *tam dereceli* denir. Aksi halde  $\mathbf{A}$  *eksik derecelidir*.

**Tanım 13:** Eğer  $\mathbf{A}$  kare ve tam dereceli ise  $\mathbf{A}$  *tekil olmayan matristir* denir.

**Teorem 2.1:**  $\mathbf{A}$  kare ve tekil olmayan bir matris olsun.  $\mathbf{A}$  'nın *tersi* olarak adlandırılan

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I} \quad (1.6)$$

şeklinde, benzersiz bir  $\mathbf{A}^{-1}$  matrisi vardır.

**İspat :** Bir  $\mathbf{A}$  kare matrisinin  $\mathbf{B}$  ve  $\mathbf{C}$  gibi iki tane tersinin olduğunu varsayalım. Bu durumda ters matrisin tanımından dolayı  $\mathbf{A}$  matrisinin tersi  $\mathbf{B}$  ise

$$\mathbf{AB} = \mathbf{BA} = \mathbf{I} \quad (1.7)$$

ve eğer  $\mathbf{A}$  matrisinin tersi  $\mathbf{C}$  ise benzer düşünceyle

$$\mathbf{AC} = \mathbf{CA} = \mathbf{I} \quad (1.8)$$

yazarız. Eğer  $\mathbf{B} = \mathbf{C}$  olduğunu gösterirsek ispatımız tamamlanmış olur. Bunun için

(1.7) ve (1.8) ifadelerini ve matrisin çarpma işlemine göre birleşme özelliğini göz

önüne alarak

$$\mathbf{B} = \mathbf{I}\mathbf{B} = (\mathbf{C}\mathbf{A})\mathbf{B} = \mathbf{C}(\mathbf{A}\mathbf{B}) = \mathbf{C}\mathbf{I} = \mathbf{C}$$

eşitliğin elde ederiz. Bu da istenendir.  $\square$

**Tanım 14:**  $\mathbf{A}$ 'nın sözde tersi

$$\mathbf{A}^\dagger = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \quad (1.9)$$

şeklinde tanımlanır.

## 2.2 Normlar

Normlar sayısal problemlerin çözümlerindeki hataları ve hassaslıkları ölçmek için kullanılırlar. Bunlar sayısal metotların yakınsamalarını oluşturmakta ve yakınsamaların oranlarını tahmin etmekte kısmen kullanılırlar. Normlar, kesin değerlerin reel çizgi üzerindeki uzaklığını belirlemeye yararlar.

### 2.2.1 Vektör Normları

Bir (vektör) norm  $\|\cdot\|: \mathbb{R}^n \rightarrow \mathbb{R}$  şeklinde bir fonksiyondur ve herhangi

$\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$  ve  $\alpha \in \mathbb{R}$  için:

1.  $\|\mathbf{x}\| \geq 0$ ,  $\mathbf{x} = 0 \Leftrightarrow \|\mathbf{x}\| = 0$ ,
2.  $\|\alpha\mathbf{x}\| = |\alpha| \|\mathbf{x}\|$ ,
3.  $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$ , (üçgen eşitsizliği)

durumunu sağlar.

Verilen bir  $\mathbf{x} \in \mathbb{R}^n$  vektörü için birkaç önemli norm aşağıdaki gibidir.



- *Euclid normu*

$$\| \mathbf{x} \|_2 = \left( \sum_{i=1}^n |x_i|^2 \right)^{1/2} \quad (2.10)$$

biçimindedir ve *2-norm* olarak ta bilinir.

- *$\infty$  - norm*

$$\| \mathbf{x} \|_{\infty} = \max_i |x_i| \quad (2.11)$$

- *1 - norm*

$$\| \mathbf{x} \|_1 = \sum_{i=1}^n |x_i|. \quad (2.12)$$

## 2.2.2 Matris Normları

Vektör normlarına benzer şekilde matris normları da  $\| \cdot \|: \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$  şeklinde ve herhangi  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}$  ve  $\alpha \in \mathbb{R}$  için aşağıdaki koşulları sağlayan bir fonksiyondur.

1.  $\| \mathbf{A} \| \geq 0$ , ve  $\| \mathbf{A} \| = 0 \Leftrightarrow \mathbf{A} = \mathbf{0}$ ,
2.  $\| \alpha \mathbf{A} \| = |\alpha| \| \mathbf{A} \|$ ,
3.  $\| \mathbf{A} + \mathbf{B} \| \leq \| \mathbf{A} \| + \| \mathbf{B} \|$ .

Önemli birkaç matris normu şu şekildedir.

- *1-norm: Sütun normu* da denir.

$$\| \mathbf{A} \|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^m |a_{ij}|.$$

- *$\infty$ -norm: Satır normu* da denir.

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

- 2-norm : *Euclid normu* da denir.

$$\|\mathbf{A}\|_2 = \left( \sum_{i,j=1}^n |a_{ij}|^2 \right)^{1/2}.$$

- *Frobenius norm*:

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2}.$$

*Frobenius norm* ve 2-norm , altçarpım özelliği olarak bilinen

$$\|\mathbf{AB}\| \leq \|\mathbf{A}\| \|\mathbf{B}\| \quad (2.13)$$

eşitsizliğini sağlar.

**Tanım 15:** Eğer keyfi bir  $\mathbf{x}$  vektörü için

$$\|\mathbf{Ax}\| \leq \|\mathbf{A}\| \|\mathbf{x}\| \quad (2.14)$$

eşitsizliği geçerli ise,  $\|\mathbf{A}\|$  matris normuna  $\|\mathbf{x}\|$  vektör normu ile *uygundur* denir.

**Teorem 2.2:**  $\|\mathbf{A}\|_2 = \left( \sum_{i,j=1}^n |a_{ij}|^2 \right)^{1/2}$

matris normu ile

$$\|\mathbf{x}\|_2 = \left( \sum_{i=1}^n |x_i|^2 \right)^{1/2}$$

vektör normu uygundur.

**İspat:** Cauchy-Schwarz eşitsizliğini göz önüne alarak

$$\|\mathbf{Ax}\|_2^2 = \sum_{i=1}^n \left| \sum_{j=1}^n a_{ij} x_j \right|^2$$

$$\begin{aligned}
&\leq \sum_{i=1}^n \left( \sum_{j=1}^n |a_{ij}|^2 \right) \left( \sum_{j=1}^n |x_j|^2 \right) \\
&= \left( \sum_{i,j=1}^n |a_{ij}|^2 \right) \left( \sum_{j=1}^n |x_j|^2 \right) \\
&= \|\mathbf{Ax}\|_2^2 \|\mathbf{x}\|_2^2
\end{aligned}$$

yazarız. Buradan

$$\|\mathbf{Ax}\|_2 \leq \|\mathbf{A}\|_2 \|\mathbf{x}\|_2 \quad (2.15)$$

olduğu görülür.  $\square$

**Tanım 16:** Belirli bir vektör normu için buna uygun matris normları arasında en küçüğüne *matrisin sınır normu* veya *en küçük üst sınırı* denir ve  $eküs(\mathbf{A})$  ile gösterilir.

Bir  $\mathbf{A}$  matrisinin  $\|\mathbf{x}\|$  vektör normuna ait sınır normu veya  $eküs(\mathbf{A})$  denilince

$$\|\mathbf{Ax}\| \leq \alpha \|\mathbf{x}\| \quad (2.16)$$

eşitsizliğini sağlayan bir  $\alpha$  sayısı anlaşılır ki; bu bütün vektörler için

$$eküs(\mathbf{A}) = \min \alpha \quad (2.17)$$

şeklinde tanımlanır. (2.14)ün sonucundan

$$\|\mathbf{A}\| = \sup_{\mathbf{x} \neq 0} \frac{\|\mathbf{Ax}\|}{\|\mathbf{x}\|} \quad (2.18)$$

olarak yazılır. Eğer  $\mathbf{x} \neq 0$  olmak üzere

$$\mathbf{z} = \frac{\mathbf{x}}{\|\mathbf{x}\|}$$

alınırsa  $\|\mathbf{z}\| = 1$  olacağından

$$\|\mathbf{A}\| = \sup_{\|z\|=1} \|\mathbf{A}z\|$$

yazılır.

O halde

$$\|\mathbf{A}\|_2 = \sup_{x \neq 0} \frac{\|\mathbf{A}x\|_2}{\|x\|_2} \quad (2.19)$$

matris  $2-normunu$  tarif eder.

### 2.3 Ortogonal Dönüşümler

Bölüm (2.2.1) ve (2.2.2) de tartışıldığı gibi ortogonal dönüşümlerin normları koruması, onları, en küçük kareler problemlerinin ve toplam en küçük kareler problemlerinin çözümleri için güçlü bir araç yapar.

**Tanım 17:**  $n$  boyutlu  $\mathbf{Q}$  kare matrisi için

$$\mathbf{Q}^T \mathbf{Q} = \mathbf{I}_n \quad (3.20)$$

ise  $\mathbf{Q}$  *ortogondir* denir. Bu durumun  $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}_n$  eşitliğini de sağladığı aşikardır.

Bir vektörün ortogonal matrisle çarpılmasına *ortogonal dönüşüm* denir.

**Teorem 2.3:** Ortogonal dönüşümler  $2-normu$  korur.

**İspat:**  $\mathbf{Q}$  ortogonal matris,  $x$  vektör ve  $\mathbf{A}$  bir matris olmak üzere

$$\|\mathbf{Q}x\|_2^2 = \langle \mathbf{Q}x, \mathbf{Q}x \rangle = x^T \mathbf{Q}^T \mathbf{Q} x = x^T x = \|x\|_2^2 \quad (3.21)$$

eşitliğinden vektörün ortogonal dönüşümü için eşitliğin sağlandığı görülür.

Öte yandan  $\mathbf{A}$  matrisinin  $2-normu$

$$\|\mathbf{A}\|_2 = \max_{\|x\|_2=1} \|\mathbf{A}x\|$$

dir. Şimdi  $\|\mathbf{QA}\|_2$  normunu bulalım. Tanımdan

$$\|\mathbf{QA}\|_2 = \max_{\|x\|_2=1} \|\mathbf{QA}x\|$$

dir.  $y = \mathbf{A}x$  aldığımızda

$$\|\mathbf{QA}\|_2 = \max_{y \in \mathbb{R}^n} \|\mathbf{Q}y\|_2$$

elde ederiz. (3.21) den  $\|\mathbf{Q}y\|_2 = \|y\|_2$  olduğundan

$$\begin{aligned} \|\mathbf{QA}\|_2 &= \max_{\|x\|=1} \|\mathbf{A}x\| \\ &= \|\mathbf{A}\|_2 \end{aligned}$$

elde ederiz.  $\square$

**Teorem 2.4:**  $\mathbf{Q}$ ,  $n$  boyutlu bir ortogonal matris ise

$$\mathbf{Q}^T = \mathbf{Q}^{-1} \tag{3.22}$$

dir.

**İspat:**  $\mathbf{Q}$  ortogonal olduğundan  $\mathbf{Q}^T \mathbf{Q} = \mathbf{I}_n$  özelliğindedir. Eşitliğin iki tarafını

sağdan  $\mathbf{Q}^{-1}$  ile çarpıldığında

$$\mathbf{Q}^T \mathbf{Q} \mathbf{Q}^{-1} = \mathbf{I}_n \mathbf{Q}^{-1}$$

bulunur.  $\mathbf{Q} \mathbf{Q}^{-1} = \mathbf{I}_n$  eşitliğin solunda ve  $\mathbf{I}_n \mathbf{Q}^{-1} = \mathbf{Q}^{-1}$  eşitliğin sağında yerlerine

yazılırsa

$$\mathbf{Q}^T = \mathbf{Q}^{-1}$$

elde edilir.  $\square$

**Teorem 2.5:**  $\mathbf{Q}$  ortogonal ise

$$\|\mathbf{Q}\|_2 = 1 \quad (3.23)$$

dir.

**İspat:** Matris normu tanımından

$$\|\mathbf{Q}\|_2 = \sup_{x \neq 0} \frac{\|\mathbf{Q}x\|_2}{\|x\|_2} = \sup_{x \neq 0} \frac{\|x\|_2}{\|x\|_2} = 1$$

bulunur.  $\square$

**Teorem 2.6:**  $2 \times 2$  boyutlu  $\mathbf{Q}$  matrisi ortogonal ise  $[0, 2\pi]$  aralığındaki benzersiz bir  $\theta$  açısı için

$$\mathbf{Q} = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \quad (3.24)$$

şeklindedir.

**İspat:**  $2 \times 2$  boyutlu  $\mathbf{Q}$  matrisi ortogonal olduğundan  $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}_2$  ve  $\Delta = \det \mathbf{Q} = 1$

dir.  $\mathbf{Q} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$  alırsak  $\mathbf{Q}^T = \mathbf{Q}^{-1} = \frac{1}{\Delta} \text{adj} \mathbf{Q}$  eşitliğinden

$$\mathbf{Q} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} \quad (3.25)$$

elde edilir. Matris eşitliğinden  $a = d$ ,  $c = -b$  ve dolayısıyla  $\mathbf{Q} = \begin{pmatrix} a & b \\ -b & a \end{pmatrix}$  ve

$a^2 + b^2 = 1$  bulunur. Bu durumda  $(a, b)$  noktası birim çember üzerine düşer. Bu durum  $[0, 2\pi]$  aralığında benzersiz bir  $\theta$  açısı için  $a = \cos \theta$ ,  $b = \sin \theta$  şeklinde

yazılabileceğini gösterir. Değerler yerine yazıldığında  $\mathbf{Q} = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix}$  elde

edilir.  $\square$

Devam eden alt bölümlerde, bu tezde sıkça kullanılacak olan iki ortogonal

dönüşüm izah edilecektir: Givens ve Householder.

### 2.3.1 Givens Dönüşümü

Bir vektörün verilen bileşenini yok edecek bir ortogonal matris aranmaktadır. Böyle bir ortogonal dönüşüm, *Givens* olarak adlandırılan bir düzlem dönüşümdür. Bazı  $\theta$  için  $c = \cos \theta$  ve  $s = \sin \theta$  olduğunda,  $I_n$  birim matrisinin rankı 2 olan bir modifikasyonudur.(3.24)

$$J(i, k, \theta) = \begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ i \rightarrow 0 & \cdots & c & \cdots & s & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ j \rightarrow 0 & \cdots & -s & \cdots & c & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{pmatrix} \quad (3.26)$$

$x \in \mathbb{R}^n$  bir vektör iken  $J(i, k, \theta)$  ile çarpılmadan önce  $J(i, k, \theta) x = y$  olacak şekilde

$y \in \mathbb{R}^n$ ,  $y = x$   $i, j$  bileşenleri hariç ve

$$\begin{aligned} y_i &= cx_i + sx_j \\ y_j &= -sx_i + cx_j \end{aligned} \quad (3.27)$$

elde edilir.

Eğer

$$c = x_i/\gamma, \quad s = x_j/\gamma, \quad \gamma = \sqrt{(x_i^2 + x_j^2)}, \quad (3.28)$$

şeklinde düzenlenirse

$$\begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} x_i \\ x_j \end{pmatrix} = \begin{pmatrix} \gamma \\ 0 \end{pmatrix} \quad (3.29)$$

elde edilir.

GIVENS algoritması  $\alpha, \beta \in \mathbb{R}$  için  $c, s$  ve  $\gamma$  'yı

$$-s\alpha + c\beta = 0$$

olacak şekilde kurar.

Yukarıdaki gibi tanımladığımız Givens dönüşümünün çarpıldığı  $x$  vektörünü saat yönünde  $\theta$  kadar döndürdüğünü gözlemleyelim.

**Örnek 2.1:**  $x = \begin{pmatrix} 2 \\ 2\sqrt{3} \end{pmatrix}$  ve  $\theta = \frac{\pi}{6}$  radyan olsun. O halde ortogonal Givens matrisi

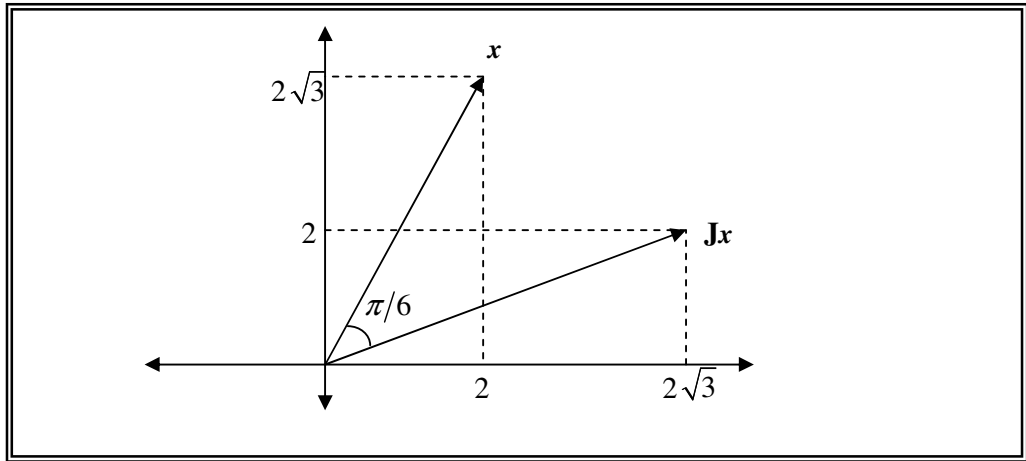
$$\mathbf{J} = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} = \frac{1}{2} \begin{pmatrix} \sqrt{3} & 1 \\ -1 & \sqrt{3} \end{pmatrix}$$

olacaktır. Buradan

$$\mathbf{J}x = \frac{1}{2} \begin{pmatrix} \sqrt{3} & 1 \\ -1 & \sqrt{3} \end{pmatrix} \begin{pmatrix} 2 \\ 2\sqrt{3} \end{pmatrix} = \begin{pmatrix} 2\sqrt{3} \\ 2 \end{pmatrix}$$

bulunur. Şekil (2.1) de görülecektir.

**Şekil 2.1: Düzlem Döndürmesi**



$x$  vektörü saat yönünde  $\pi/6$  radyan dönmüştür.



**Çizelge 2.1: Givens Algoritması**

**Girdi:**  $\alpha$  ve  $\beta$  skalarları

**Çıktı:**  $\begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \gamma \\ 0 \end{pmatrix}$  şeklindeki  $c, s, \gamma$  skalarları

$[c, s, \gamma] = GIVENS(\alpha, \beta)$

- (1) if ( $\beta = 0$ )
- (2)  $c = 1; s = 0; \gamma = \alpha;$
- (3) elseif ( $|\beta| > |\alpha|$ )
- (4)  $t = \alpha/\beta; tt = \sqrt{1+t^2};$
- (5)  $s = 1/tt; c = t s; \gamma = tt \beta;$
- (6) else
- (7)  $t = \beta/\alpha; tt = \sqrt{1+t^2};$
- (8)  $c = 1/tt; s = t c; \gamma = tt \alpha;$

### 2.3.2 Householder Dönüşümü

Genellikle  $\mathbf{H}$  şeklinde göstereceğimiz *Householder matrisi*,  $\mathbf{v}$  sıfır olmayan  $n$ -vektör olacak şekilde

$$\mathbf{H} = \mathbf{I}_n - \frac{1}{\mu} \mathbf{v} \mathbf{v}^T, \quad \mu = \frac{1}{2} \mathbf{v}^T \mathbf{v} \quad (3.30)$$

formundaki bir matristir.  $\mathbf{H}$  ile çarpma işlemine *Householder dönüşümü* denir.

Verilmiş bir  $\mathbf{x}$  vektörü için  $\mathbf{v}$  'yi

$$\mathbf{v} = \mathbf{x} \pm \sigma \mathbf{e}_1, \quad \sigma = \|\mathbf{x}\|_2 \quad (3.31)$$

şeklinde alırsak  $\mathbf{H}\mathbf{x} = \pm\sigma \mathbf{e}_1$  elde edilir.  $\mathbf{v}$  vektörünün işaretini seçiminde bir potansiyel tehlike vardır. Örneğin,  $\sigma_1 = \mathbf{x}^T \mathbf{e}_1$  alınırsa;

$$\mathbf{v}^T \mathbf{v} = (\mathbf{x} \pm \sigma \mathbf{e}_1)^T (\mathbf{x} \pm \sigma \mathbf{e}_1) = \sigma^2 \pm 2\sigma \alpha_1 + \sigma^2 = 2\sigma(\sigma \pm \alpha_1), \quad (3.32)$$

böylelikle  $\mu = \sigma(\sigma \pm \alpha_1)$  olur. Eğer  $\mathbf{x}$ ,  $\mathbf{e}_1$  çarpımına yakınsa  $\sigma \approx |\sigma_1|$  ve iptal olur.

Bu  $\mu$  içinde büyük bir bağıl hataya sebep olabilir. Bunu engelleyebilmek için

$$\mathbf{v} = \mathbf{x} + \text{sgn}(\alpha_1) \sigma \mathbf{e}_1, \quad \mu = \sigma(\sigma + |\alpha_1|) \quad (3.33)$$

alınır.

**Teorem 2.7:** Householder matrisi hem ortogonal hem de simetriktir. Yani  $\mathbf{H}$  bir Householder matrisi olmak üzere

$$\mathbf{H} = \mathbf{H}^T = \mathbf{H}^{-1} \quad (3.34)$$

dir.

**İspat:**  $\mathbf{H} = \mathbf{I}_n - \frac{2}{\mathbf{v}^T \mathbf{v}} (\mathbf{v}\mathbf{v}^T)$  matrisindeki  $\mathbf{v}^T \mathbf{v}$  ifadesinin reel sayı değeri olduğu

aşikardır. Öncelikle  $\mathbf{H}^T = \mathbf{H}$  olduğunu gösterelim.

$$\mathbf{H}^T = \left( \mathbf{I}_n - \frac{2}{\mathbf{v}^T \mathbf{v}} (\mathbf{v}\mathbf{v}^T) \right)^T = \mathbf{I}_n^T - \frac{2}{\mathbf{v}^T \mathbf{v}} (\mathbf{v}\mathbf{v}^T)^T$$

$\mathbf{I}_n^T = \mathbf{I}_n$  ve  $(\mathbf{v}\mathbf{v}^T)^T = (\mathbf{v}^T)^T \mathbf{v}^T = \mathbf{v}\mathbf{v}^T$  yerine yazıldığında

$$\mathbf{H}^T = \mathbf{I}_n - \frac{2}{\mathbf{v}^T \mathbf{v}} (\mathbf{v}\mathbf{v}^T) = \mathbf{H}$$

elde edilir.

Şimdi  $\mathbf{H}^T = \mathbf{H}^{-1}$  olduğunu gösterelim. Bunun için  $\mathbf{H}\mathbf{H}^T = \mathbf{I}_n$  olduğunu

göstermek yeterlidir.

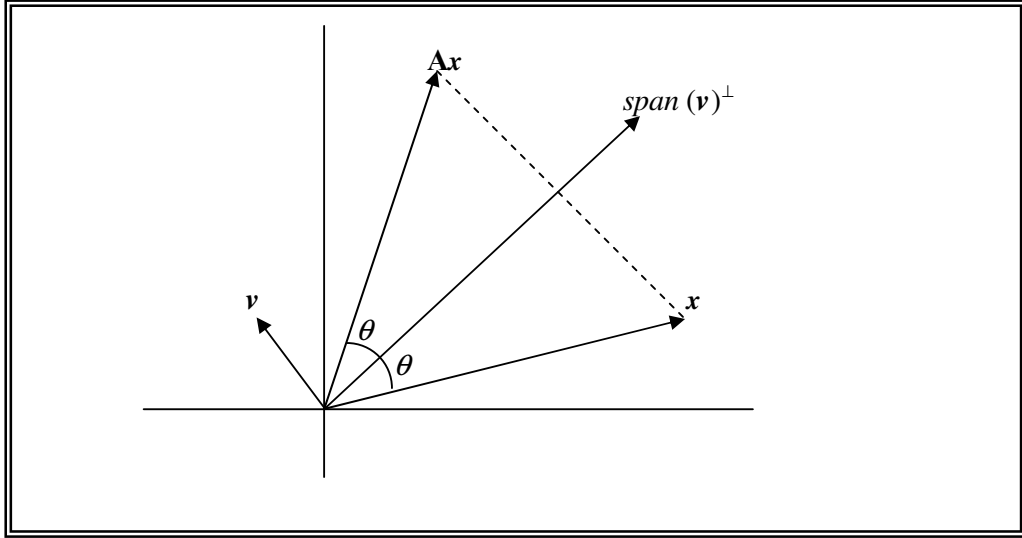
$$\begin{aligned} \mathbf{H}\mathbf{H}^T &= \left( \mathbf{I}_n - \frac{2}{\mathbf{v}^T \mathbf{v}} (\mathbf{v}\mathbf{v}^T) \right) \left( \mathbf{I}_n - \frac{2}{\mathbf{v}^T \mathbf{v}} (\mathbf{v}\mathbf{v}^T) \right) \\ &= \mathbf{I}_n - \frac{2}{\mathbf{v}^T \mathbf{v}} (\mathbf{v}\mathbf{v}^T) - \frac{2}{\mathbf{v}^T \mathbf{v}} (\mathbf{v}\mathbf{v}^T) + \frac{4}{(\mathbf{v}^T \mathbf{v})^2} (\mathbf{v}\mathbf{v}^T \mathbf{v}\mathbf{v}^T) \end{aligned}$$

burada  $\mathbf{v}\mathbf{v}^T \mathbf{v}\mathbf{v}^T = \mathbf{v}(\mathbf{v}^T \mathbf{v})\mathbf{v}^T = (\mathbf{v}^T \mathbf{v})\mathbf{v}\mathbf{v}^T$  değeri için

$$\mathbf{H}\mathbf{H}^T = \mathbf{I}_n - \frac{2}{\mathbf{v}^T \mathbf{v}} (\mathbf{v}\mathbf{v}^T) - \frac{2}{\mathbf{v}^T \mathbf{v}} (\mathbf{v}\mathbf{v}^T) + \frac{4}{\mathbf{v}^T \mathbf{v}} (\mathbf{v}\mathbf{v}^T) = \mathbf{I}_n$$

elde edilir.  $\square$

### Şekil 2.2: Householder Yansımaları



Givens dönüşümündeki transformasyon matrisi şekil (2.1)deki gibi

$$Q = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \text{ şeklinde ortogonal bir matrisle çarpılarak negatif yönde } \theta$$

açısı kadar döndürürken, Householder yansımada da  $v_1 = \cos \theta$  olarak düşünüldüğünde vektör,

$$H = \begin{pmatrix} 1-2v_1^2 & -2v_1v_2 \\ -2v_1v_2 & 1-2v_2^2 \end{pmatrix} = \begin{pmatrix} \cos(2\theta) & -\sin(2\theta) \\ -\sin(2\theta) & -\cos(2\theta) \end{pmatrix} = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ -\sin \alpha & -\cos \alpha \end{pmatrix}$$

matrisiyle çarpılacaktır. Householder dönüşümü şekil (2.2)de de görüldüğü gibi vektörü pozitif yönde  $2\theta$  kadar döndürmektedir.

Durum böyle iken Householder yansımasının yer yer Givens dönüşümünden kullanışlı olmasının sebebi, vektörün  $i$ -inci elemanından sonraki elemanların hepsini birden sıfırlıyor olmasıdır. Görüleceği gibi Givens dönüşümünde bu işlem her eleman için ayrı ayrı yapılmaktadır.

**Çizelge 2.2: Householder Algoritması**

<p><b>Girdi:</b> Bir <math>n</math>-vektör <math>x</math></p> <p><b>Çıktı:</b> Bir <math>n</math>-vektör <math>v</math></p> <p><math>[v, \mu, \hat{\sigma}] = \text{HOUSE}(x)</math></p> <p>(1) <math>\sigma = (x^T x)^{1/2};</math></p> <p>(2) <math>\alpha_1 = x^T e_1;</math></p> <p>(3) <math>\hat{\sigma} = -\text{sgn}(\alpha_1)\sigma;</math></p> <p>(4) <math>v = x - \hat{\sigma} e_1;</math></p> <p>(5) <math>\mu = \sigma(\sigma +  \alpha_1 );</math></p>
---

### 3. ARAŞTIRMA BULGULARI

#### 3.1 Rankı Veren QR Ayrışımı

Bir  $m \times n$   $\mathbf{A}$  matrisinin ( $m \geq n$ ) bir QR ayrışımı,  $\mathbf{Q} \in \mathbb{R}^{m \times m}$  ortogonal yani;  $\mathbf{Q}^T \mathbf{Q} = \mathbf{I}_m$  ve  $\mathbf{R} \in \mathbb{R}^{n \times n}$  üst üçgensel olacak şekilde

$$\mathbf{A} = \mathbf{Q} \begin{pmatrix} \mathbf{R} \\ 0 \end{pmatrix} \quad (1.1)$$

olarak verilmiştir.

$\mathbf{Q}$  ortogonal matrisini aşağıdaki biçimde parçalansın:

$$\mathbf{Q} = (\mathbf{Q}_1 \quad \mathbf{Q}_2), \quad \mathbf{Q}_1 \in \mathbb{R}^{m \times n}, \quad \mathbf{Q}_2 \in \mathbb{R}^{m \times (m-n)} \quad (1.2)$$

Bu durumda (1.1) ve (1.2) den

$$\mathbf{A} = \mathbf{Q}_1 \mathbf{R} \quad (1.3)$$

elde edilir.

$\mathbf{A} \in \mathbb{R}^{m \times n}$  matrisinin rankı  $n$  olduğu zaman  $\mathbf{Q}_1$  ortogonal matrisinin sütunları  $range(\mathbf{A})$ 'nın ortogonal bir tabanıdır. Bu tezde, bu durumla ilgilenilmeyecek olsa ta,  $rank(\mathbf{A}) = r < n$  olduğunda (1.1),  $\mathbf{R}_{11} \in \mathbb{R}^{r \times r}$  üst üçgensel ve sıfır olmayan köşegen girişli matris olmak üzere

$$\mathbf{A} = \begin{pmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ 0 & 0 \end{pmatrix} \quad (1.4)$$

şeklinde modifiye edilebilir. (1.4) formundaki gibi bir  $\mathbf{QR}$  ayrışımı rankı ortaya çıkaran QR (RRQR) ayrışımı olarak adlandırılır.

### 3.2 Tam Ortogonal Ayrışım

Sağdan ortogonal dönüşümleri kullanarak ayrışımın kalitesini artırmak mümkündür. Bu durumda  $\mathbf{U} \in \mathbb{R}^{m \times m}$  ve  $\mathbf{V} \in \mathbb{R}^{n \times n}$  ortogonal matrisleri ve  $\mathbf{T} \in \mathbb{R}^{r \times r}$  tekil olmayan matrisi köşegen olabilecek şekilde (alt üçgensel veya üst üçgensel matris) sıfır olmayan köşegen girişleri olan

$$\mathbf{A} = \mathbf{U} \begin{pmatrix} \mathbf{T} & 0 \\ 0 & 0 \end{pmatrix} \mathbf{V}^T \quad (2.5)$$

formunda bir ayrışım elde edilir. Devam eden alt bölümlerde köşegen form detaylı gösterilecektir.

#### 3.2.1 Tekil Değer Ayrışımı

(2.5) durumundaki ayrışım, orta matris köşegen matris olduğunda,  $\mathbf{A}$  matrisinin tekil değer ayrışımı (SVD: Singular Value Decomposition) olarak isimlendirilir. SVD matrisin rankını ortaya çıkarmak için en çok kullanılan tekniktir. Aynı zamanda en küçük kareler problemleri ve toplam en küçük kareler problemlerinin hesaplanmasında kullanışlıdır.  $\mathbf{U} \in \mathbb{R}^{m \times m}$  ve  $\mathbf{V} \in \mathbb{R}^{n \times n}$ 'nin her ikisi ortogonal ve  $\mathbf{\Sigma} \in \mathbb{R}^{n \times n}$  köşegen olacak şekilde, rankı  $r$  olan  $\mathbf{A}$  matrisinin SVD formu

$$\mathbf{A} = \mathbf{U} \begin{pmatrix} \mathbf{\Sigma} \\ 0 \end{pmatrix} \mathbf{V}^T \quad (2.6)$$

şeklinde.  $\mathbf{A}$ 'nın tekil değerleri olarak isimlendirilen  $\mathbf{\Sigma}$ 'nin köşegen elemanları  $\sigma_i$  ler

$$\sigma_1 \geq \sigma_2 \geq \dots \sigma_r > \sigma_{r+1} = \dots = \sigma_n = 0 \quad (2.7)$$

sıralamasına sahiptir.  $\mathbf{U}$  'nun ve  $\mathbf{V}$  'nin  $i$ -inci sütunları, sırasıyla  $i$ -inci sol ve sağ tekil vektörler olarak isimlendirilir.

Böylece (2.6)daki SVD ,

$$\sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^T \quad (2.8)$$

şeklinde genişletilebilir.

**Örnek 3.2:**  $\mathbf{A} = \begin{pmatrix} 1 & 2 \\ 2 & 0 \\ 0 & 2 \end{pmatrix}$  matrisinin tekil değer ayrışımını bulalım.

$$\mathbf{A}^T \mathbf{A} = \begin{pmatrix} 5 & 2 & 4 \\ 2 & 4 & 0 \\ 4 & 0 & 4 \end{pmatrix}$$

karakteristik polinomu

$$(\lambda - 5)(\lambda - 4)(\lambda - 4) - 20(\lambda - 4) = (\lambda - 4)(\lambda^2 - 9\lambda) = (\lambda - 9)(\lambda - 4)(\lambda - 0) \text{ dir.}$$

$\mathbf{AA}^T$  'un özdeğerleri

$$\lambda_1 = 9 , \lambda_2 = 4 \text{ ve } \lambda_3 = 0 \text{ dir.}$$

$\mathbf{A}$  matrisinin tekil değerleri

$$\sigma_1 = 3 \text{ ve } \sigma_2 = 2 \text{ dir.}$$

$\mathbf{A}$  'nın sol tekil vektörleri  $\mathbf{AA}^T$  'un özvektörleridir.

$$(\mathbf{9I} - \mathbf{AA}^T)\mathbf{u} = 0 \text{ çözümünde, } (5 \ 2 \ 4)^T \text{ un bütün katlarının } \mathbf{AA}^T \text{ un } \lambda_1 = 9$$

özdeğerleriyle ilişkilendirilmiş özvektörü olduğunu buluruz. Ve  $(\mathbf{4I} - \mathbf{AA}^T)\mathbf{u} = 0$  ın

çözümünde  $(0 \ 6 \ -3)^T$  ın katlarını içeren  $\mathbf{AA}^T$  un diğer özvektörünü buluruz.

İstedığımız temsilcileri Euclid normuyla

$$\mathbf{u}_1 = \frac{1}{3\sqrt{5}} \begin{pmatrix} 5 \\ 2 \\ 4 \end{pmatrix}, \mathbf{u}_2 = \frac{1}{3\sqrt{5}} \begin{pmatrix} 0 \\ 6 \\ -3 \end{pmatrix} \text{ ve } \mathbf{u}_3 = \frac{1}{3\sqrt{5}} \begin{pmatrix} -2\sqrt{5} \\ \sqrt{5} \\ 2\sqrt{5} \end{pmatrix}$$

şeklinde hesaplarız.

Bunlar  $\mathbf{A}$ 'nın sol tekil vektörleridir ve ortogonaldır.  $\mathbf{v}_1$  ve  $\mathbf{v}_2$  sağ özvektörleri

$\mathbf{A}^T \mathbf{A}$ 'nın özvektörlerini hesaplayarak bulabiliriz. Yine de  $\mathbf{v}_1$  ve  $\mathbf{v}_2$

$$\mathbf{v}_i = \frac{1}{\sigma_i} \mathbf{A}^T \mathbf{u}_i \quad i=1,2$$

formülüyle kolayca bulunur.

$$\mathbf{v}_1 = \frac{1}{3} \mathbf{A}^T \mathbf{u}_1 = \frac{1}{\sqrt{5}} \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

$$\mathbf{v}_2 = \frac{1}{2} \mathbf{A}^T \mathbf{u}_2 = \frac{1}{\sqrt{5}} \begin{pmatrix} 2 \\ -1 \end{pmatrix}.$$

$\mathbf{A}$ 'nın tekil değerlerini ve tekil vektörlerini elde etmiş olduk. Şimdi  $\mathbf{U}$ ,  $\Sigma$  ve

$\mathbf{V}$ 'yi kolayca kurabiliriz.

$$\mathbf{U} = (\mathbf{u}_1 \quad \mathbf{u}_2 \quad \mathbf{u}_3) = \frac{1}{3\sqrt{5}} \begin{pmatrix} 5 & 0 & -2\sqrt{5} \\ 2 & 6 & \sqrt{5} \\ 4 & -3 & 2\sqrt{5} \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 3 & 0 \\ 0 & 2 \\ 0 & 0 \end{pmatrix}$$

$$\mathbf{V} = (\mathbf{v}_1 \quad \mathbf{v}_2) = \frac{1}{\sqrt{5}} \begin{pmatrix} 1 & 2 \\ 2 & -1 \end{pmatrix}$$



$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \frac{1}{3\sqrt{5}} \begin{pmatrix} 5 & 0 & -2\sqrt{5} \\ 2 & 6 & \sqrt{5} \\ 4 & -3 & 2\sqrt{5} \end{pmatrix} \begin{pmatrix} 3 & 0 \\ 0 & 2 \\ 0 & 0 \end{pmatrix} \frac{1}{\sqrt{5}} \begin{pmatrix} 1 & 2 \\ 2 & -1 \end{pmatrix}^T$$

olduğu kolayca görülebilir.

### 3.3 Sayısal Rank ve Sayısal Tekil Altuzaylar

Eğer  $m \geq n$  olmak üzere bir  $\mathbf{A} \in \mathbb{R}^{m \times n}$  matrisinin  $r < n$  şeklinde bir matematiksel rankı varsa, yaklaşık matris  $\mathbf{A} + \mathbf{E}$ ,  $r$  den büyük  $n$  ye yakın bir matematiksel ranka sahiptir. Bu durum, yuvarlama hatalarını barındıran bu matrislerin rankını hesaplamamanın imkansız olduğunu anlatır. Fakat  $\mathbf{E}$ ,  $\mathbf{A}$  'dan yeterince küçükse rank eksikliğine oldukça kapalıdır ve sayısal rank eksikliği olarak göz önüne alınabilir.  $\mathbf{A}$  'nın tekil değerleri ve sayısal rankı arasındaki iyi bir tanım Stewart tarafından verilmiştir.

**Tanım 16:** Bir  $\mathbf{A}$  matrisi için

$$k = \min\{\text{rank}(\mathbf{B}) \mid \|\mathbf{A} - \mathbf{B}\|_2 \leq \varepsilon\} \quad (3.9)$$

ise  $k$  sayısal  $\varepsilon$ -rankı vardır denir.

Tanımdan dolayı  $\sigma_i$  ler  $\mathbf{A}$  'nın tekil değerleri olmak üzere  $\mathbf{A}$  matrisinin sayısal  $\varepsilon$ -rankı 'nın  $k$  olması için gerek ve yeter şart

$$\sigma_1 \geq \sigma_2 \geq \dots \sigma_k > \varepsilon \geq \sigma_{k+1} \geq \dots \sigma_n \quad (3.10)$$

dir.

Farzedelim ki (3.2.1)'deki  $\mathbf{A}$  matrisinin sayısal rankı  $k$  olsun.  $\mathbf{A}$  'nın (2.6)'da verilen tekil değer ayrışımı  $\mathbf{\Sigma}_1 = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_k)$  ve

$\Sigma_2 = \text{diag}(\sigma_{k+1}, \sigma_{k+2}, \dots, \sigma_n)$  köşegen matrisle olmak üzere

$$\mathbf{A} = (\mathbf{U}_1 \quad \mathbf{U}_2) \begin{pmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \\ 0 & 0 \end{pmatrix} (\mathbf{V}_1 \quad \mathbf{V}_2)^T \quad (3.11)$$

şeklindedir.

$\mathbf{A}_1 = \mathbf{U}_1 \Sigma_1 \mathbf{V}_1$  olarak , tanım (15)ten,

$$\|\mathbf{A} - \mathbf{A}_1\|_2 = \|\mathbf{A} \mathbf{V}_2\|_2 \leq \varepsilon, \quad \mathbf{V}_2 = (\mathbf{v}_{k+1}, \mathbf{v}_{k+2}, \dots, \mathbf{v}_n), \quad (3.12)$$

dir. Ayrıca,

$$\text{range}(\mathbf{V}_2) = \text{span}\{\mathbf{v}_{k+1}, \mathbf{v}_{k+2}, \dots, \mathbf{v}_n\} \quad (3.13)$$

$\mathbf{A}$  matrisinin sayısal çekirdek uzayı veya kirlilik uzayı denir.

## 4. TARTIŞMA VE SONUÇ

SVD matrislerin rankını ve alt uzay bilgilerini öğrenmek için kullanılan ana araçtır ama hesapsal olarak zordur ve yenilenmesi zordur. Bu nedenle ULVD, SVD'yi yaklaştırmak için kullanılır.

Bugünkü yöntemlerle, başlangıçtan ULVD'yi hesaplamak, SVD'yi hesaplamaktan çok daha az zaman alır. ULVD algoritmaları zayıf bir başlangıç olan  $A$ 'nın QR ayrışımı ile başlar ve durum tahmini, azaltma ve temizleme gibi basamaklarla devam eder.

Bu bölümde, rankı  $k$  olan  $m \times n$  boyutunda bir matris için,  $4mn^2 - \frac{4}{3}n^3 + \frac{98}{3}n^2 - 30n^{3/2} - \frac{12k+9}{2}n \log n + O(n)$  maliyetinin flopları, küçük flopları için, sabit SVD algoritmasının tüm uygulamaları yerine, özyineli ULVD (RULV) algoritması uygulanacaktır.

### 4.1 Özyineli ULVD Algoritması

Kolaylık olması amacıyla  $A$ ,  $n$  boyutlu bir kare matris olarak alınsın.

Çizelge (4.3)te verilen algoritmada, BIDIAGONAL yöntemi, bidiagonal  $\mathbf{B} \in \mathbb{R}^{n \times n}$  matrisini ve ortogonal  $\mathbf{U}^{(1)}, \mathbf{V}^{(1)} \in \mathbb{R}^{n \times n}$  matrislerini

$$\mathbf{A} = \mathbf{U}^{(1)} \mathbf{B} \mathbf{V}^{(1)T} \quad (1.1)$$

şekilde hesaplar.

Burada;

$$\mathbf{B} = \begin{pmatrix} \alpha_1 & & & & & \\ \phi_1 & \alpha_2 & & & & \\ & \phi_2 & \ddots & & & \\ & & \ddots & \alpha_{n-1} & & \\ & & & \phi_{n-1} & \alpha_n & \end{pmatrix} \quad (1.2)$$

dir.

**Çizelge 4.3:ULVD Algoritması**

<p><b>Girdi:</b> Bir <math>n \times n</math> <math>\mathbf{A}</math> matrisi</p> <p><b>Çıktı:</b> <math>\mathbf{A}</math>'nın sayısal rankı <math>k</math>, <math>\mathbf{C} \in \mathbb{R}^{n \times n}</math> alt üçgensel matrisi ve <math>\mathbf{U}, \mathbf{V} \in \mathbb{R}^{n \times n}</math> ortogonal matrisleri</p> <p><math>[k, \mathbf{U}, \mathbf{C}, \mathbf{V}] = \text{ULVD}(\mathbf{A})</math></p> <p>(1) <math>n = \text{size}(\mathbf{A});</math></p> <p>(2) <math>[\mathbf{U}_0, \mathbf{B}_0, \mathbf{V}_0] = \text{BIDIAGONALFORM}(\mathbf{A});</math></p> <p>(3) <math>[k, \mathbf{U}, \mathbf{C}, \mathbf{V}] = \text{RECVLVD}(n, \mathbf{U}_0, \mathbf{B}_0, \mathbf{V}_0, \epsilon);</math></p>
--

Bundan sonra, RECVLVD algoritması tekrarlayarak  $\mathbf{B}$ 'nin ULV ayrışımını hesaplar. Detaylar çizelge (4.4)te gösterilmiştir.

RECVLVD önce  $\mathbf{B}$ 'yi

$$\mathbf{B} = \begin{matrix} & & r & n-r \\ & r & & \\ & 1 & & \\ n-r-1 & & & \end{matrix} \begin{pmatrix} \mathbf{B}_1 & 0 \\ \phi_{r+1} e_r^T & \alpha_{r+1} e_1^T \\ 0 & \mathbf{B}_2 \end{pmatrix} \quad (1.3)$$

şeklinde parçalar. Burada  $\mathbf{B}_1$  ve  $\mathbf{B}_2$  yaklaşık olarak aynı boyuttadır. Bundan sonra

her  $\mathbf{B}_i$ ,  $i=1,2$  (1.3) deki gibi parçalanır ve bu işlem  $\mathbf{B}_i$ 'nin boyutu  $n^{1/2}$  den küçük yada eşit olana kadar devam eder. Bunun ardından her  $\mathbf{B}_i$ 'nin ULVD'si

$\varepsilon \|\mathbf{L}_1^{-1}\| \leq 1$ ,  $\varepsilon \|\mathbf{L}_2^{-1}\| \leq 1$  olmak üzere

$$\mathbf{B}_1 = \mathbf{U}_1^{(1)} \begin{pmatrix} \mathbf{L}_1 & 0 \\ \mathbf{F}_1 & \mathbf{G}_1 \end{pmatrix} \mathbf{V}_1^{(1)T} \quad (1.4)$$

$$\mathbf{B}_2 = \mathbf{U}_2^{(1)} \begin{pmatrix} \mathbf{L}_2 & 0 \\ \mathbf{F}_2 & \mathbf{G}_2 \end{pmatrix} \mathbf{V}_2^{(1)T} \quad (1.4)$$

olarak hesaplanır. Temel durumda SVD  $\mathbf{B}_i$ 'nin ULV ayrışımını elde etmek için kullanılır. Bu yöntem matrisin SVD'sini hesaplar.

MERGE, (1.4) deki  $\mathbf{B}_1$  ve  $\mathbf{B}_2$  matrislerini çizelge (4.4)te verilen algoritmayı elde etmek için birleştirir.

$$\begin{pmatrix} \mathbf{L}_1 & 0 & 0 & 0 \\ \mathbf{F}_1 & \mathbf{G}_1 & 0 & 0 \\ \mathbf{z}_1^T & \mathbf{z}_2^T & \mathbf{z}_3^T & \mathbf{z}_4^T \\ 0 & 0 & \mathbf{L}_2 & 0 \\ 0 & 0 & \mathbf{F}_2 & \mathbf{G}_2 \end{pmatrix} = \mathbf{U}^{(2)T} \mathbf{B} \mathbf{V}^{(2)}, \quad \mathbf{z} = \begin{pmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \mathbf{z}_3 \\ \mathbf{z}_4 \end{pmatrix} \quad (1.5)$$

Burada  $\mathbf{U}^{(2)}, \mathbf{V}^{(2)} \in \mathbb{R}^{n \times n}$  ortogonal matrislerdir. Bu aşağıdaki gibi gösterilebilir.

$$\mathbf{z} = \begin{pmatrix} \alpha \mathbf{V}_1^{(1)}(r,:) \\ \beta \mathbf{V}_2^{(1)}(1,:) \end{pmatrix} \quad (1.6)$$

Daha sonra birleştirilmiş matris REORDER algoritması ile  $\mathbf{U}^{(3)}, \mathbf{V}^{(3)} \in \mathbb{R}^{n \times n}$

ortogonal matrisler iken

$$\begin{pmatrix} \mathbf{L}_1 & 0 & 0 & 0 \\ 0 & \mathbf{L}_2 & 0 & 0 \\ \mathbf{z}_1^T & \mathbf{z}_3^T & \mathbf{z}_2^T & \mathbf{z}_4^T \\ \mathbf{F}_1 & 0 & \mathbf{G}_1 & 0 \\ 0 & \mathbf{F}_2 & 0 & \mathbf{G}_2 \end{pmatrix} = \mathbf{U}^{(3)T} \begin{pmatrix} \mathbf{L}_1 & 0 & 0 & 0 \\ \mathbf{F}_1 & \mathbf{G}_1 & 0 & 0 \\ \mathbf{z}_1^T & \mathbf{z}_2^T & \mathbf{z}_3^T & \mathbf{z}_4^T \\ 0 & 0 & \mathbf{L}_2 & 0 \\ 0 & 0 & \mathbf{F}_2 & \mathbf{G}_2 \end{pmatrix} \mathbf{V}^{(3)} \quad (1.7)$$

elde etmek için yeniden düzenlenir. Bu durumda algoritmanın geri kalanı SHAVE ve UPDATE algoritmalarıyla elde edilen  $\mathbf{z}^T$  gibi bir satırı toplamaya benzer.

SHAVE algoritması  $\mathbf{U}^{(4)}, \mathbf{V}^{(4)} \in \mathbb{R}^{n \times n}$  ortogonal matrislerini aşağıdaki gibi hesaplar.

$$\begin{pmatrix} \bar{\mathbf{L}} & 0 \\ \bar{\mathbf{z}}^{-T} & \zeta \mathbf{e}_1^T \\ \bar{\mathbf{F}} & \bar{\mathbf{G}} \end{pmatrix} = \mathbf{U}^{(4)T} \begin{pmatrix} \mathbf{L}_1 & 0 & 0 & 0 \\ 0 & \mathbf{L}_2 & 0 & 0 \\ \mathbf{z}_1^T & \mathbf{z}_3^T & \mathbf{z}_2^T & \mathbf{z}_4^T \\ \mathbf{F}_1 & 0 & \mathbf{G}_1 & 0 \\ 0 & \mathbf{F}_2 & 0 & \mathbf{G}_2 \end{pmatrix} \mathbf{V}^{(4)} \quad (1.8)$$

Burada  $\bar{\mathbf{L}} = \begin{pmatrix} \mathbf{L}_1 & 0 \\ 0 & \mathbf{L}_2 \end{pmatrix}$ ,  $\bar{\mathbf{z}}^{-T} = (\mathbf{z}_1^T \mathbf{z}_3^T)$ ,  $\bar{\mathbf{F}} = \begin{pmatrix} \mathbf{F}_1 & 0 \\ 0 & \mathbf{F}_2 \end{pmatrix}$ ,  $\bar{\mathbf{G}}$  alt üçgensel matris ve

$\zeta = \|\mathbf{z}_2^T \mathbf{z}_4^T\|$  dir.

Sonuç olarak UPDATE yöntemi, (1.8)deki matrisi  $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{n \times n}$  ortogonal matrislerini hesaplamak için

$$\begin{pmatrix} \mathbf{L} & 0 \\ \mathbf{F} & \mathbf{G} \end{pmatrix} = \mathbf{U}^T \begin{pmatrix} \bar{\mathbf{L}} & 0 \\ \bar{\mathbf{z}}^{-T} & \zeta \mathbf{e}_1^T \\ \bar{\mathbf{F}} & \bar{\mathbf{G}} \end{pmatrix} \mathbf{V} \quad (1.9)$$

şeklinde günceller.

Çizelge (4.3)te verilen algoritmanın karmaşasının “flop” un toplamalar ve çarpımlar tanımını kullanarak hesabı

#### Çizelge 4.4: Özyineli ULVD Algoritması

**Girdi:**  $A$  'nın boyutu  $N$ ,  $B \in \mathbb{R}^{n \times n}$  bidiagonal matrisi,  $A = UB V^T$

olacak şekilde  $U \in \mathbb{R}^{n \times n}$ ,  $V \in \mathbb{R}^{n \times n}$  ortogonal matrisleri ve  $\varepsilon$  hata öntahmini.

**Çıktı:**  $C \in \mathbb{R}^{n \times n}$  alt üçgensel matrisi,  $B$  'nin sayısal rankı  $k$  ve  $B = UC V^T$

olacak şekilde  $U \in \mathbb{R}^{n \times n}$ ,  $V \in \mathbb{R}^{n \times n}$  ortogonal matrisleri.

$[k, U, C, V] = \text{RECVLVD}(N, U, B, V, \varepsilon)$

(1)  $[m, n] = \text{size}(B)$ ;

(2)  $r = \text{floor}(m/2)$ ;

(3) if  $(r > n^{1/2})$

(4)  $B_1 = B(1:r, 1:r+1)$ ;  $B_2 = B(r+2:m, r+2:n)$ ;

(5)  $U_1 = U(:, 1:r)$ ;  $U_2(:, r+1:n)$ ;

(6)  $V_1 = V(:, 1:r)$ ;  $V_2(:, r+1:n)$ ;

(7)  $[k_1, U_1, B_1, V_1] = \text{RECVLVD}(N, U_1, B_1, V_1, \varepsilon)$ ;

(8)  $[k_2, U_2, B_2, V_2] = \text{RECVLVD}(N, U_2, B_2, V_2, \varepsilon)$ ;

(9)  $[U, B, V] = \text{MERGE}(m, n, r, B(r+1, r+1), B(r+1, r+2),$

$U_1, B_1, V_1, U_2, B_2, V_2)$ ;

(10)  $[k, U, B, V] = \text{REORDER}(r, k_1, k_2, U, B, V)$ ;

(11)  $[U, B, V] = \text{SHAVE}(k, U, B, V)$ ;

(12)  $[k, U, B, V] = \text{UPDATE}(k, U, B, V, \varepsilon)$

(13) else

(14)  $[k, U, B, V] = \text{SVD}(B, \varepsilon)$ ;

(15) endif

$$4mn^2 - \frac{4}{3}n^3 + \frac{98}{3}n^2 - 30n^{3/2} - \frac{12k+9}{2}n \log n + O(n) \quad (1.10)$$

dır. Bu durum önerme (4.2) de gösterilecektir.

Çizelge (4.4)te verilen algoritmanın ilk önce problemi her seviyede daha küçük boyutlu iki alt probleme böldüğü görülür. Şimdiki önerme seviyelerin toplam sayısını verir.

**Önerme 4.1:** Çizelge (4.4)te verilen algoritmanın terminolojisini varsayarsak, RULV'nin  $l$  seviyelerinin sayısı

$$l = \frac{1}{2}[\log_2 n] \quad (1.11)$$

ile verilir.

Seviyelerin sayısını hesapladıktan sonra artık RULV algoritmasının hesap karmaşasının yukarıda söz edildiği gibi olduğu gösterilebilir.

**Önerme 4.2:**  $m \times n$  boyutlu, rankı  $k$  olan  $\mathbf{A}$  matrisi, çizelge (4.3)te verilen algoritmaya uygulansın.  $\mathbf{A}$ 'nın ULV ayrışımını elde etmek için

$$4mn^2 - \frac{4}{3}n^3 + \frac{98}{3}n^2 - 30n^{3/2} - \frac{12k+9}{2}n \log n + O(n)$$

verir.

**İspat:**  $\mathbf{A}$  matrisinin çizelge (4.3)te verilen algoritmanın uygulandığı seviyelerinin sayısını  $l$  alalım. Çizelge (4.4)te verilen algoritmanın 14. basamağındaki bütün SVD lerinin maliyeti

$$2^l \times (\text{boyutu} \leq \frac{n}{2^l} \text{ olan bir matrisin her SVD'sinin maliyeti}) \quad (1.12)$$

şeklinde verilir.



Bundan başka,  $i \leq l-1$  için  $i$ -inci seviyede boyutu  $\frac{n}{2^i}$  den küçük olan bir matrisin

$2^i$  veri eklemesi vardır. Bu nedenle veri eklemeler, çizelge (4.4)te verilen algoritmanın 12. basamağına gelindiğinde bütün veri eklemeleri

$\frac{5}{2}[n_0^2 - n_0(2k_0 - 1) + (k_0^2 - k_0)]$  maliyetli,  $n_0$  boyutlu, rankı  $k_0$  olan bir matris için

$$\sum_{i=0}^{l-1} 2^i \times (\text{boyutu} \leq \frac{n}{2^i} \text{ olan bir matrisin veri ekleme maliyeti}) \quad (1.13)$$

ekler.  $l = \frac{1}{2}[\log_2 n]$  verilerek, (1.12) ve (1.13)ye bidiagonalleştirme basamağının

maliyeti eklenerek ispat tamamlanır.  $\square$

## KAYNAKLAR

1. Watkins D.S., Fundamentals of Matrix Computations, John Wiley & Sons, (1991).
2. Taşcı D., Lineer Cebir, Sel-Ün Vakfı Yayınları, (2001).
3. Golub G.H. ve Van Loan C.F., Matrix Computations Third Edition, John Hopkins University, (1996).
4. Lipschutz S., Schaum's Outline Of Theory And Problems Of Linear Algebra Si(metric) Edition, McGraw-Hill International Book Company, (1968).
5. Bayram M., Fen Ve Mühendislik İçin Nümerik Analiz, Aktif Yayınevi, (2002).
6. Björck A., Numerical Methods for Least Squares Problems, Society For Industrial And Applied Mathematics (1996).
7. Erbay H., Modifying Rank-Revealing Decomposition, Doktora Tezi, The Pennsylvania State University, The Graduate School, (2000).
8. Biglieri E. ve Yao K., Some Properties Of Singular Value Decomposition And Their Applications To Digital Signal Processing, Signal Processing, 18:277-289, (1989).
9. Stewart G.W., An Updating Algorithm For Subspace Tracking, Signal Processing, 40:1535-1541, (1992).
10. Stewart G.W., Updating A Rank-Revealing ULV Decomposition, Society For Industrial And Applied Mathematics, (1993).
11. Stewart G.W., UTV Decompositions. Numerical Analysis, Pitman Research Notes in Mathematical Science, (1994).
12. Stewart G.W., Matrix Algorithms. Volume 1: Basic Decompositions. Society

For Industrial And Applied Mathematics Philadelphia, (1998).

13. Rahman M.A. ve Yu K., Total Least Squares Approach for Frequency Estimation Using Linear Prediction, *ASSP*, 35:1142-54, (1987).
14. Scharf L.L., The SVD And Reduced Rank Signal Processing, *Signal Proc.*, 25:113-133, (1991).
15. Chan T.F., Rank Revealing Qr Factorizations, *Linear Algebra Applications*, 88:549-568, (1987).
16. Foster L., Rank And Null Space Calculations Using Matrix Decomposition Without Column Interchanges, *Linear Algebra Applications*, 17:883-893, (1980).
17. Faddeev D.K., Kublanovskaya V.N. ve Faddeeva V.N., Solutions Of Linear Algebraic Systems With Rectangular Matrices, *Proc. Steklov Inst. Math.*, 96: 93-111, (1968).
18. Hanson R.J. ve Lawson C.L., Extensions and Applications of The Householder Algorithm for Solving Linear Least Squares Problems, *Math. Comp.*, 23:787-812, (1969).
19. Fierro R.D. ve Hansen P.C., Low Rank Revealing Utv Decompositions, *Numerical Algorithms*, 15:37-55, (1997).