

# Kare Olmayan Matrisin Sözde Tersini Bulma

Ata GÜLALAN - Oğuzhan TÜRKER

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

xavaneo@gmail.com oguzturker8@gmail.com

## Özet

Uygulama kullanıcı tarafından seçilen bir kare olmayan matrisin sözde tersini almaktadır. Sözde tersi alma işleminde “Moore-Penrose Inverse” yöntemini kullanmaktadır.

## 1. Giriş

Uygulama, öncelikle kullanıcıya işlem görececek matrisin kendiliğinden mi yoksa kullanıcı tarafından mı oluşturulacağını sorar.

Eğer kullanıcı rastgele matris seçeneğini seçerse matris [1-5] boyut ve [1.0-9.0] sayı arasından rastgele oluşturulur. Matrisin boyutu rastgele atandığında kare matris olamaz.

Oluşturulan matrisin sözde tersi bulunur.

Kullanıcı eğer matris özelliklerini kendisi girmek isterse diğer seçenek ile uygulamaya devam eder. Matris boyutları ve elemanları girildikten sonra matrisin sözde tersi hesaplanır.

Uygulama kullandığı algoritmanın adımlarını arayüzde aşama aşama kullanıcıya göstermektedir. Kullanıcı algoritma basamaklarını tek tek inceleyebilme şansına sahiptir.

## 2. Temel Bilgiler

Proje gelişiminde;

Programlama dili olarak “C#”, tümleşik geliştirme ortamı olarak “Visual Studio 2017” ve “Visual Studio 2015” kullanılmıştır.

Uygulama içeriği olarak “Windows Presentation Foundation (WPF)” kullanılmıştır.

## 3. Tasarım

Proje aşağıdaki başlıklar altında geliştirilmiştir.

### 3.1 Yazılım Tasarımı

Projenin yazılım aşaması bu başlık altında bulunan konular tarafınca geliştirilmiştir.

### 3.1.1 Algoritma

Matrisin sözde tersini bulmak için “Moore-Penrose Inverse” metodunu kullandık.

Burada sözde tersi bulmak için;

- Öncelikle matrisin boyutları bulunur.
- Satır sütundan büyükse \* işleminde  $A^t A$  kullanılır.
- Satır sütundan küçükse \* işleminde  $A A^t$  kullanılır.
- Verilen matrisin transpozu alınır.
- Matrisin transpozu ile kendisi çarpılır.  $[*]$
- Bu çarpımın  $-1$ . kuvveti alınır.
- Alınan kuvvetin sonucu ile transpoz  $A$  çarpılır.

Sonuç bize sözde matris olan  $A^+$  'yı vermektedir.

### 3.1.2 Arayüz

**void tekerlegiEngelle** : Scrollviewer elemanı, aşamalar arası geçişte kullanılıyor. Bu yüzden mousewheel engellenmeli ki, kullanıcı aşamalar arasında bilinçsizce dolaşmasın.

**void Window\_SizeChanged** : Pencere yeniden boyutlandırıldığında aşama değişmesini engelle ve tüm aşamaların yüksekliğini pencere boyutu ile eşitle.

**sayiGirisKontrolu** : Girilen değerlerin sayı olup olmadığını kontrol et. Eğer sayı ise, bu sayının diğer girdi ile aynı olup olmadığını kontrol et. Kullanıcının kare matris oluşturmasını engelle.

**void otomatikOlusturButon\_Click** : Rastgele iki sayı üret, eğer aynı sayılar geldi ise ikinci sayıyı aynı olmayana kadar yeniden üret.

**degistirButon\_Click** : Satır ve sütun boyutlarını ters çevirir.

**olusturButon\_Click** : asama1Bitti() fonksiyonunu çağırır.

**asama1Bitti** : Scrollviewer'ı Aşama 2'nin olduğu yere kaydırır ve asama2Basla() fonksiyonunu çağırır.

**void MorNChanged** : Matrisin boyutlarından herhangi birisi girilmemiş ise oluştur butonunu tıklanamaz yapar. Eğer ikisi de girilmişse oluştur butonu aktif hale gelir.

**geriDonButon\_Click** : Aşama 1'e geri döndürür.  
**asama2Basla** : Aşama 2'yi başlatır. Aşama 1'de girilen matrise göre, matrisin elemanlarını girmeyi sağlayan inputları gizler/gösterir. Gizlenen elemanlar yüzünden kayma olmaması için oluşturulan matrisi ortalara. Geri gelip tekrar ilerlendiğinde eleman karmaşası olmaması için tüm elemanların içeriğini sıfırlar.  
**void otomatikMatrisElemanlari\_Click** : Tüm elemanlara rastgele veri atar.  
**void CheckForFullMatris** : Matris elemanlarının kullanılabilir olanları içerisinde gezer ve şu işlemleri yapar:  
1- Dolu mu diye kontrol eder.  
2- Double olup olmadığını kontrol eder.  
Kullanılabilir elemanların hepsi bu şartları sağlıyorsa devam edilebilir. Devam et butonunu aktif hale getirir.  
**void matrisTemizle\_Click** : Girilen matrisin elemanlarını temizler.  
**void asama2Bitti** : Scrollviewer'ı Aşama 3'ün olduğu yere kaydırır ve asama3Basla() fonksiyonunu çağırır.  
**void asama2DevamEt\_Click** : asama2Bitti(); fonksiyonunu çağırır.

**asama3Basla** : Geri döndüğünde toplama ve çarpma sayılarında karmaşa olmaması için bu değişkenleri sıfırlar. Bir önceki aşamada girilen elemanları iki boyutlu bir matrise atar. Bu matrisin sözde tersini alır ve kullanıcı arayüzüne yazdırır. Ters alınma işlemi, [3.1.3 Hesaplama] bölümünde gösterilmiştir.  
**geriDonButon2\_Click** : Aşama 2'ye geri döndürür.  
**asama3Bitti** : Scrollviewer'ı Aşama 4'ün olduğu yere kaydırır ve asama4Basla() fonksiyonunu çağırır.  
**asama3DevamEt\_Click** : asama3Bitti() fonksiyonunu çağırır.  
**void gosterMatris** : Matrisi ekranda gösterir. Matriste gizlenen elemanlar yüzünden kayma olmaması için oluşturulan matrisi ortalara.

**aGoster\_Click** : A Matrisini ekrana yazdırır ve Aşama 3'ün açıklamasını değiştirir.  
**aTGoster\_Click** :  $A^T$  Matrisini ekrana yazdırır ve Aşama 3'ün açıklamasını değiştirir.  
**aTaGoster\_Click** :  $M > N$  ve  $M < N$  kontrolü yapılır.  $M > N$  ise  $A^T A$ ,  $M < N$  ise  $A A^T$  matrisi ekranda gösterilir ve Aşama 3'ün açıklamasını değiştirir.  
**aTaM1Goster\_Click** :  $M > N$  ve  $M < N$  kontrolü yapılır.  $M > N$  ise  $(A^T A)^{-1}$ ,  $M < N$  ise  $(A A^T)^{-1}$  matrisi ekranda gösterilir ve Aşama 3'ün açıklamasını değiştirir.  
**aPGoster\_Click** : Tüm hesaplamalardan sonra çıkan matrisi, yani girilen matrisin sözde tersini yazdırır.

**asama4Basla** : Toplama ve çarpma işlemini ekrana yazdırır.  
**geriDonButon3\_Click** : Aşama 3'e geri döndürür.  
**bastanBasla\_Click** : Aşama 1'e geri döndürür.

### 3.1.3 Hesaplama

**double[,] pseudoinverse** : Matrisin sözde tersi alınma işlemi yapılır.

**double[,] transpose** : Parametre olarak alınan matrisin transpozunu alır ve döndürür.

**double[,] matris\_carpimi** : Girilen iki matrisin çarpımını bulur ve yeni bir matris olarak döndürür.

**double[,] kuvvet\_al** : Parametre olarak alınan matrisi, alınan değere göre kuvvetini alır.

### 3.1.4 Karşılaşılan Sorunlar ve Çözüm Yöntemleri

Programın ana ekranını tasarladıktan sonra, matrisin sözde tersini almak için Singular Value Decomposition (SVD) kullanmaya karar verdik, ancak bu algoritmayı koda uyarlarken bazı sıkıntılarla karşılaştık. Bu yüzden nispeten daha kolay bir metod olan ve matrisin sözde tersine çok yakın bir değer bulan farklı bir algoritmayı, Moore-Penrose Inverse (MPI) kullandık.

## 4. Kazanımlar

### 4.1 Algoritmalar

Programda kullandığımız algoritmanın mantığını ve neden böyle bir yönteme ihtiyaç duyulduğunu öğrendik. Üstelik bu projede kullanılabilecek diğer algoritmaları da tanımış olduk. Öğrendiğimiz algoritmalar: Singular Value Decomposition (SVD), Least Squares Method (LSM), Moore-Penrose Inverse (MPI), Gauss-Seidel Method (GSM).

Bu algoritmalarından Moore-Penrose Inverse (MPI) ve Gauss-Seidel Method (GSM) kullandık.

Projeyi daha önce çok az tecrübemiz olan bir dil olan C# ile yaptık. Sonucunda C#'da programlama yeteneğimizi bir üst düzeye taşıdık.

### 4.2 Matris Yapısı ve Matris İşlemleri

C#'da matrislerin nasıl işlendiğini, matrislerin birbirleri ile çarpılmasını, matrisin bir değer ile kuvvetinin alınması işlemlerinin her bir elemana göre nasıl yapılacağını öğrendik. Gauss-Seidel ve Moore-Penrose algoritmalarının koda nasıl uyarlanabileceğini öğrendik.

## 5. Tasarım Aşaması

Tasarım kısmında kullanıcı deneyimine ekstra özen gösterdik. Tüm uygulama tek bir pencere içerisinde aşamalı olarak çalışıyor. Kullanıcı, bu pencereyi yeniden boyutlandırabilme özelliğine sahip. Her bir aşamada sadece gerekli bilgiler alınıyor ve kullanıcının talep ettiği halde matris otomatik tanımlanıp otomatik veri atayabiliyor. Her aşamanın kendine özgü bir yapısı var. Kısaca bahsetmek gerekirse aşamalar şu şekilde:

**Aşama 1:** Kullanıcı matris boyutlarını tanımlar

**Aşama 2:** Kullanıcı matris içeriğini tanımlar.

**Aşama 3:** Kullanıcı tanımladığı matrisin sözde tersi alınırken her bir aşamasını inceleme şansına sahip olur.

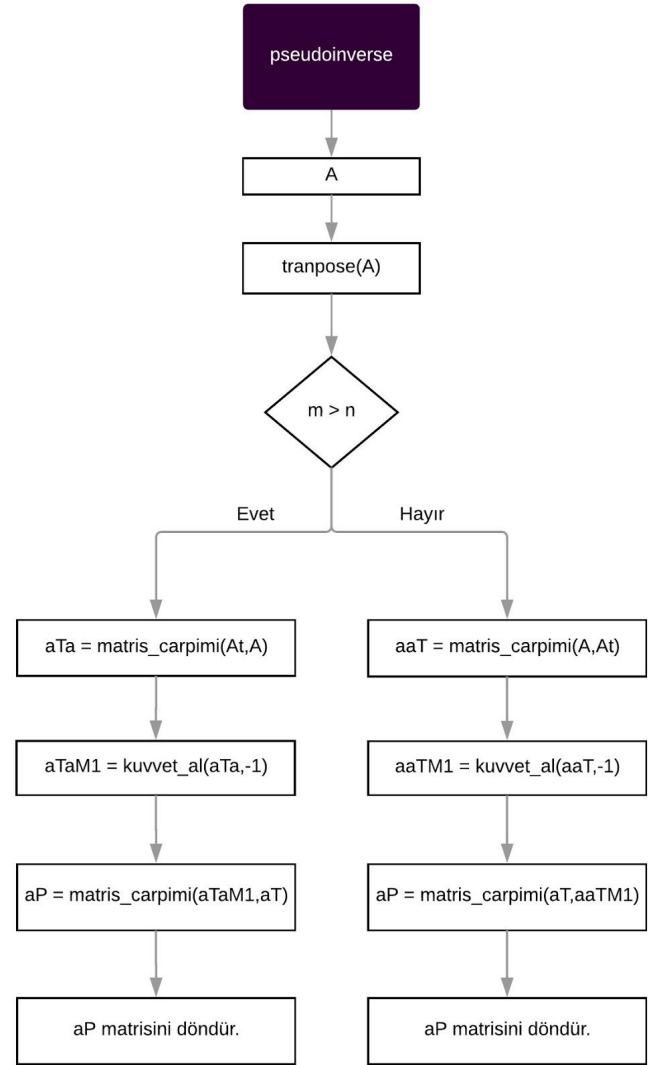
**Aşama 4:** Matrisin sözde tersini alırken yapılan işlemler gösterilir.

Bu aşamaların yanında, kullanıcı deneyimini arttırmak için yaptığımız birkaç özellik:

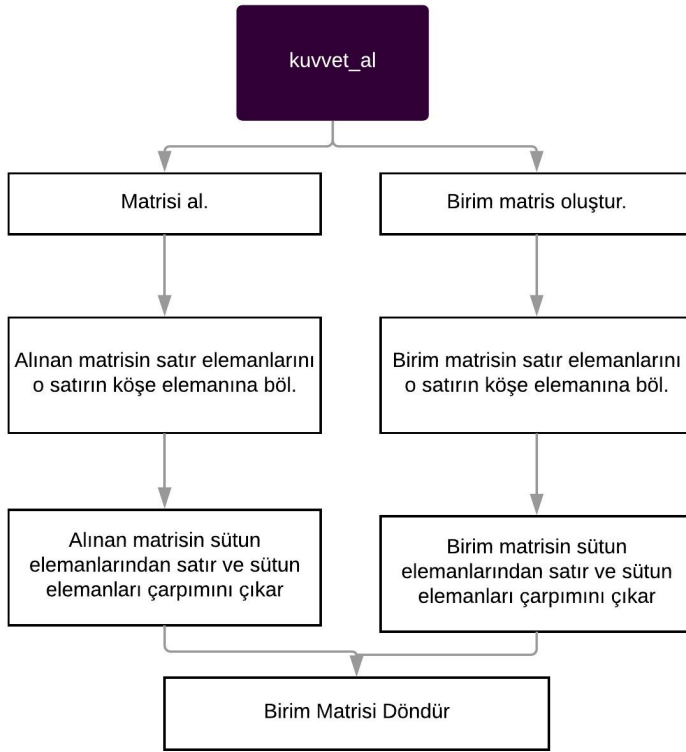
- Kullanıcı “TAB” tuşunu kutucuklar arasında geçiş yapmak için kullanabilir. Her doldurulabilir butonun TabIndex’i el ile ayarlandı.
- Kullanıcı matris boyutlarını girerken aynı sayıyı giremez. Kullanıcının aynı boyutu girmesi, daha girerken engellenir.
- Kullanıcı satır-sütun boyutlarını ters çevirebilir. Bu sayede 3x4 tanımlaması gerekirken 4x3 tanımladığında tek tık ile istenen matris boyutuna geri dönülebilir.
- Kullanıcı her bir aşamada geri gelme şansına sahiptir. Yanlış bir veri girildiği anlaşırsa bir önceki aşamaya dönüp girdiği veriyi düzenleyebilir.
- Matris boyutu girilen matrisin değerine göre kullanıcıya gösterilir. Yanlış matris boyutu girme şansı yoktur. Verilen matris boyutlarında herhangi bir yerde boşluk veya yanlış karakter var ise matris oluşturulması engellenir.
- Matris tersi alındıktan sonra, Aşama 3’te matrisin sözde tersinin hangi formüller ile alındığı kullanıcıya açıkça bildirilir. Bu şekilde, matris’in sözde tersini almayı bilmeyen bir kişi, yapılan işlemleri sırasıyla yaparak matrisin sözde tersini almayı öğrenebilir.

Kullanıcı deneyiminin yanında, kullanıcı arayüzüne de ekstra özen gösterdik. Minimalizm akımını örnek alan arayüzümüzde herhangi bir border (çerçeve) kullanmadık. Bunun yerine, her girdi yapılan (input veya button) elemana gölge efekti verdik. Bu sayede kullanıcıyı herhangi bir kutu içine hapsedmek yerine açık havada bisiklet sürüyormuş gibi hissetmesini sağladık.

## 6. Akış Şeması



Şekil 1 - (pseudoinverse fonksiyonu akış şeması)



őekil 2 - (kuvvet\_al fonksiyonu akış őeması)

## 7. Kaynakça

### Moore-Penrose Inverse

[1][https://en.wikipedia.org/wiki/Moore%E2%80%93Penrose\\_inverse](https://en.wikipedia.org/wiki/Moore%E2%80%93Penrose_inverse)

Eriřim Tarihi: 26.11.2018

### Last Square Method (LSM)

[2]<https://www.youtube.com/watch?v=vGowBXcur1k>

Eriřim Tarihi: 25.11.2018

### Singular Value Decomposition (SVD) - Stanford University

[3]<https://www.youtube.com/watch?v=P5mlg91as1c>

Eriřim Tarihi: 23.11.2018

### Singular Value Decomposition (SVD) - Massachusetts Institute of Technology

[4]<https://www.youtube.com/watch?v=mBcLRGuAFUk>

[5]<https://www.youtube.com/watch?v=cOUTpqIX-Xs>

[6]<https://www.youtube.com/watch?v=N74V4CBO0sk>

Eriřim Tarihi: 23.11.2018

### Gauss-Seidel Method

[7]<http://bilgisayarkavramlari.sadievrenseker.com/2008/11/19/matrisin-tersinin-alinmasi-mantrix-inverse>

Eriřim Tarihi: 26.11.2018

### Ascii Art Generator

[8]<http://patorjk.com/software/taag/#p=display&f=Big>

Eriřim Tarihi: 22.11.2018