# EBMS00 İLERİ MÜHENDİSLİK MATEMATİĞİ
# FİNAL ÖDEVİ

ADI = Muharrem Osman

SOYADI = TOPAKKAYA

NO = 233720418

DANIŞMAN = Dr. Öğr. Üyesi Merve Bulut YILGÖR

# İLERİ MÜHENDİSLİK MATEMATİĞİ-FİNAL

Muharrem Osman TOPAKKAYA - 233720418

## Bilgisayar ve Elektrik Mühendisliğinde Lineer Cebirin Kullanım Alanları

### 1) Temel Kavramlar ve Tanımlar

Lineer cebir, vektörler, matrisler, determinantlar ve özdeğerler gibi matematiksel yapılarla ilgilenen bir matematik dalıdır. Bazı temel kavramlar,

* Vektörler: Bir yönü ve büyüklüğü olan matematiksel nesnelerdir.

$$v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

* Matrisler: Sayıların dikdörtgen tablosu şeklinde düzenlenmiş hali olup, lineer dönüşümleri temsil eder.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

* Determinantlar: Bir kare matrisin bir sayıya indirgenmiş hali olup, matrisin tersinin olup olmadığını belirlen

$$\det(A) = a_{11}a_{22} - a_{12}a_{21}$$

* Özdeğerler ve Özvektörler: Bir matrisin karakteristik özelliklerini belirleyen ve dönüşümlerini tanımlayan bileşenlerdir

$$Av = \lambda v$$

Burada $A$ matris, $v$ özvektör, $\lambda$ ise özdeğerdir

## 2) Lineer Cebirin Bilgisayar ve Elektrik Mühendisliğindeki Kullanım Alanları

### a) Grafik ve Görüntü İşleme:

* 2D ve 3D Grafiklerde Dönüşümler: Grafiklerdeki nesnelerin ölçeklenmesi, döndürülmesi ve yansıtılması gibi işlemler lineer cebir kullanılarak matris çarpımlarıyla gerçekleştirilir. Örneğin, bir nesnenin bir eksen etrafında döndürülmesi rotasyon matrisleri ile yapılır.

2D rotasyon matrisi:

$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

3D rotasyon matrisleri:

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

$$R_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix}$$

$$R_z(\gamma) = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

★ Görüntü Sıkıştırma Algoritmaları:

JPEG gibi görüntü sıkıştırma algoritmaları, görüntüleri daha küçük boyutlarda saklamak için lineer cebir tekniklerinden yararlanır. Bu algoritmalar, görüntüyü matris olarak temsil eder ve tekil değer ayrışımı (SVD) gibi yöntemlerle sıkıştırır.

$$A = U \Sigma V^T$$

Burada A görüntü matrisi, U ve V ortonormal matrisler, $\Sigma$ ise diyagonal matristir.

b) Makine Öğrenmesi ve Veri Madenciliği:

★ Veri Temsili ve Boyut İndirgeme:

Veriler, vektör uzayında temsil edilerek analiz edilir. Özellik seçimi ve boyut indirgeme teknikleri, verilerin daha yönetilebilir boyutlara indirgenmesini sağlar. Örneğin PCA (Principal Componet Analysis) verilerin boyutunu düşürerek bilgi kaybını minimize eder.

$$X' = XW$$

Burada X veri matrisi, W ise dönüşüm matrisidir.

★ Lineer Regresyon: Lineer regresyon modeli, verilerin doğrusal ilişkilerini analiz etmek için kullanılır.

$$y = X\beta + \epsilon$$

Burada y hedef değişken, X özellik matrisi, $\beta$ katsayılar ve $\epsilon$ hata terimidir.

c) Kriptografi:

Kod Çözme ve Şifreleme Algoritmaları:

Lineer cebir, kriptografik algortmaların temelinde yer alır. Örneğin, RSA algoritmasında büyük asal sayılar ve matris işlemleri kullanılarak şifreleme ve çözme işlemleri yapılır.

$$C = M^e \mod n$$
$$M = C^d \mod n$$

Burada C şifreli metin, M açık metin, e ve d şifreleme ve çözme anahtarlarıdır, n ise modül.

d) Sinyal İşleme:

Filtreleme ve Dönüşümler: Sinyal işleme, sinyallerin analiz edilmesi ve işlenmesini içerir. Fourier dönüşümü, sinyallerin frekans bileşenlerine ayrılmasını sağlar.

$$F(w) = \int_{-\infty}^{\infty} f(t) e^{-iwt} \, dt$$

Diskret Fourier dönüşümü (DFT):

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N}$$

## e) Devre Analizi:

### ✱ Dirençler, Kapasitörler ve İndüktörler:

Elektrik devrelerindeki bileşenlerin analizinde, lineer cebir kullanılarak devre denklemleri çözülür. Örneğin, Kirchhoff'un gerilim ve akım yasaları matris denklemleriyle ifade edilir.

Örnek devre için Kirchhoff denklemleri:

$$\begin{cases} V_1 - I_1 \cdot R_1 - (I_1 - I_2) \cdot R_2 = 0 \\ -(I_2 - I_1) \cdot R_2 + I_2 \cdot R_3 = 0 \end{cases}$$

Bu denklemleri matris formunda yazarsak;

$$\begin{bmatrix} R_1 + R_2 & -R_2 \\ -R_2 & R_2 + R_3 \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} V_1 \\ 0 \end{bmatrix}$$

## f) Kontrol Sistemleri:

### ✱ Durum Uzayı Modelleri:

Kontrol sistemlerinde, sistemlerin dinamik davranışları durum uzayı modelleri ile analiz edilir. Bu modeller diferansiyel denklemlerle ifade edilir ve matris biçiminde gösterilebilir.

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) + Du(t)$$

Burada $x(t)$ durum vektörü, $u(t)$ giriş vektörü, $y(t)$ çıkış vektörü, $A, B, C, D$ sistem matrisleridir.

g) Elektromanyetik Alan Teorisi:

* Maxwell Denklemleri:

Elektromanyetik alanların analizi için kullanılan Maxwell denklemleri, matris formunda gözdebilir. Bu denklemler, elektrik ve manyetik alanların davranışını tanımlar.

$$\nabla \times E = -\frac{\partial B}{\partial t}$$

$$\nabla \times H = J + \frac{\partial D}{\partial t}$$

h) Güç Sistemleri:

* Güç Akışı Analizleri:

Elektrik güç sistemlerinde, güç akışı analizleri lineer cebir kullanılarak yapılır. Bu analizler, enerji dağıtımını optimize etmek için kullanılır.

$$P = VI \cos(Q)$$

3) Lineer Cebir Yazılımları ve Araçları

Lineer cebir işlemleri ve analizleri için çeşitli yazılımlar ve kütüphaneler kullanılır:

MATLAB: Mühendislik ve bilimsel hesaplamalar için yaygın olarak kullanılan bir yazılımdır. MATLAB, vektör ve matris işlemleri için geniş bir fonksiyon yelpazesi sunar.

NumPy: Python programlama dilinde kullanılan bir kütüphanedir. NumPy, lineer cebir işlemleri için kullanışlı fonksiyonlar içerir.

4) Örnek Uygulamalar ve Problemler

* Makine Öğrenmesi ile PCA Kullanımı:

Verilen bir veri setinin boyutunu PCA kullanarak indirgeme ve analiz etme.

Python,

```
import numpy as np
from sklearn.decomposition import PCA

#Veri Seti
X=np.array([[2.5, 2.4], [0.5, 0.7], [2.2, 2.9], [1.9, 2.2
[3.1, 3.0], [2.3, 2.7],

#PCA ile boyut indirgeme
pca = PCA(n_components=1)
X_reduced = pca.fit_transform(X)
print("İndirgenmiş Veri:\n", X_reduced)
```

* Devre Analizi ile Matris Denklemi Çözümü;

Basit bir elektrik devresi için kirchhoff denklemlerinin matris çözüm

Matlab,

```
% Devre Parametreleri
R1=1; R2=2; R3=3;
V1=10;
% Matris Denklemi
A= [R1+R2, -R2; -R2, R2+R3];
B=[V1; 0];
% Akım çözümleri
I= inv(A)* B;
disp('Akımlar:');
disp(I);
```