Karadeniz Fen Bilimleri Dergisi 8(1), 115-130, 2018

KFBD

Karadeniz Fen Bilimleri Dergisi
The Black Sea Journal of Sciences
ISSN (Online): 2564-7377

Araştırma Makalesi / Research Article

Mathematica ile Matris Cebiri

Halil MUTUK

Ondokuz Mayıs Üniversitesi, Fen-Edebiyat Fakültesi, Fizik Bölümü, Samsun, Türkiye

Geliş Tarihi: 03.05.2018
Kabul Tarihi: 29.05.2018

Sorumlu Yazar: halilmutuk@gmail.com

Özet

Bu çalışmada Mathematica programı kullanılarak matris cebirindeki işlemler bilgisayar ortamında yapılmıştır. Matrislerde toplama, çıkarma, çarpma, determinant alma ve ters bulma işlemleri ayrıntılı algoritmalarıyla birlikte verilmiştir.

Anahtar Kelimeler: Mathematica, Matris, Matris Cebiri

Matrix Algebra in Mathematica

Abstract

In this work we implemented matrix algebra into Mathematica. Summation, subtraction, multiplication, taking determinant and inverse of the matrices are shown detailed algorithms in Mathematica.

Keywords: Mathematica, Matrix, Matrix Algebra

1. Giriş

Matris en genel anlamda sayıların veya değişkenlerin dikdörtgen veya kare biçimindeki bir tablo şeklinde sıralanmasıdır. Bu sayılar bir problemde ele alınan parametreler de olabilir. Bir matris satır ve sütun sayısıyla temsil edilir. Matrisin matematiksel tanımı şu şekilde yapılır:

Tanım 1. $m, n, i, j > 0$ olmak üzere tüm a_{ij} reel sayılarından oluşan

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \ddots & \cdots \\ a_{m1} & a_{m2} & \cdots & \cdots & a_{mn} \end{bmatrix}_{m\times n} \tag{1}$$

dizilişe $m \times n$ boyutunda bir A matrisi denir. Bu matris kısaca $A = [a_{ij}]_{m\times n}$ şeklinde de gösterilebilir. Bu gösterimde a_{ij}, A matrisinin $i-nci$ satır ve $j-nci$ sütunundaki elemanıdır. Matristeki yatay sıralara matrisin satırı, düşey sıralara da matrisin sütunu denir. $m \neq n$ ise matrise dikdörtgen matris, $m = n$ ise matrise kare matris denir. Uygulamalarda kare matrisi daha geniş bir kullanım alanına sahiptir. Her ne kadar matris elemanlarını genelde reel sayılardan oluşsa da bazı durumlarda kompleks sayılardan oluşabilir. Ayrıca matrisin elemanları fonksiyon veya bizzat kendisi de bir matris olabilir.

Matris teorisi, hem lineer cebirin hem de matematiğin önemli bir alt dalıdır. Matrisler gerek matematik gerekse fen bilimleri içerisinde geniş kullanım ve uygulama sahasına sahiptir. Lineer denklem takımının matris gösterimi çözümün araştırılmasında büyük kolaylık sağlar. Örneğin elektrik devre analizinde Kirchhoff denklemlerini matrisler aracılığıyla çözmek doğrudan çözmeye göre daha caziptir. Benzer şekilde vektör cebirini de matris teorisi aracılığıyla çalışmak sonlu bir uzayda yer alan vektör ve operatörler matrislerle temsil edilebildiğinden mümkündür. Örneğin Heisenberg tarafından geliştirilen kuantum mekaniğinin matris formülasyonu özellikle parçacıkların saçılma genliğini hesaplarken oldukça kullanışlıdır. Yine mühendislikte yer alan dinamik sistemlerin analiz ve dizayn problemleri matrislerin kullanıldığı bir çalışma alanıdır.

Matrisler teorisinde matrisin toplama, çıkarma ve çarpma işlemleri tanımlıyken iki matrisin bölme işlemi tanımlı değildir. Daha doğru bir ifadeyle matrislerin bölünmesi iyi tanımlı değildir. Bunun altında yatan sebep matris çarpımının yer değiştirmemesidir (komütatif olmamasıdır). A ve B iki matris olmak üzere bölme işlemini şu şekilde gösterelim: $\frac{A}{B} = AB^{-1}$. Bu tanımda B^{-1}, B matrisinin tersidir. Burada her matrisin tersinin var olmayacağını da not etmek gerekir. B terslenebilir bir matris olduğunda $B^{-1}A = \frac{A}{B}$ olacağı söylenebilir fakat iki matrisin çarpımı genel anlamda sıra değiştirmezdir, yani $AB^{-1} \neq B^{-1}A$. Dolayısıyla tanımlanan iki bölme işleminde farklı sonuçlar çıkacaktır ki bu bölme işleminin tanımına uygun değildir.

Matrislerle ilgili Türkçe literatürde geniş ve kapsamlı ders kitapları bulmak mümkündür. Bunlardan bazıları bu çalışmanın kaynaklar kısmında belirtilmiştir (Hacısalihoğlu, 1999; Sabuncuoğlu, 2017; Yüce, 2015; Çalışkan, 2011; Çiftçi, 2015; Akın, 2007; Taşçı, 2006; Çallıalp ve Kuruoğlu, 1996). Biz bu çalışmada kısaca matrislerde toplama, çıkarma ve çarpma işlemlerini Mathematica programı yardımıyla tanımlayacağız.

Tanım 2. Matrislerin Toplanması ve Çıkarılması

A ve B aynı boyutlu iki matrisin toplamı karşılıklı elemanların toplanması veya çıkarılması ile yapılır. Sonuç matrisi C ile gösterilirse toplama işlemi şu şekilde yazılabilir

$$[c_{ij}]_{m\times n} = [a_{ij}]_{m\times n} \pm [b_{ij}]_{m\times n} \tag{2}$$

Farklı boyuttaki matrisle toplanamaz veya çıkarılamaz. Örneğin iki matrisin toplamı şu şekildedir (birinci satırda işlemlerin nasıl yapıldığı gösterilmiştir):

$$\begin{bmatrix} 1 & 7 & -1 \\ 2 & 2 & 2 \\ -3 & 3 & 4 \end{bmatrix} + \begin{bmatrix} 2 & -2 & 5 \\ 6 & -4 & 0 \\ -5 & 1 & -5 \end{bmatrix} = \begin{bmatrix} (1+2=)3 & (7-2=)5 & (-1+5=)2 \\ 8 & -2 & 2 \\ -8 & 4 & -1 \end{bmatrix}.$$

Tanım 3. Matrislerin Çarpımı

Matrislerle çarpma işleminin yapılabilmesi için birinci matrisin sütun sayısı ile ikinci matrisin satır sayıları birbirlerine eşit olmalıdır. A ve B iki matris olmak üzere çarpma işlemi ancak $[a_{ij}]_{m\times p}$ ile $[b_{ij}]_{p\times n}$ olduğunda mümkündür. Çarpımın gösterimi şu şekildedir:

$$[c_{ij}]_{m\times n} = a_{i1}b_{ij} + a_{i2}b_{2j} + \cdots + a_{ip}b_{pj} = \sum_{k=1}^{p} a_{ik}b_{kj} \tag{3}$$

Bir örnek olması açısından şu çarpma işlemini gösterelim:

$$\begin{bmatrix} 3 & 2 \\ 1 & 4 \\ -1 & -5 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 3 \\ -2 & 1 & -3 \end{bmatrix} = \begin{bmatrix} (3-4) & (6+2) & (9-6) \\ (1-8) & (2+4) & (3-12) \\ (-1+10) & (-2-5) & (-3+15) \end{bmatrix}$$

$$= \begin{bmatrix} 16 & 5 & 37 \\ 38 & 30 & 76 \\ -23 & -38 & -33 \end{bmatrix}.$$

Burada ispatını yapmadan göstereceğimiz, matrislerin bir takım cebirsel özellikleri vardır. Matrislerde toplama ve çıkarma değişme ve birleşme özelliklerine sahiptir:

$A + B = B + A$ (değişme) (4)

$A + (B + C) = (A + B) + C$ (birleşme) (5)

$k(A + B) = kA + kB$, k skaler bir sayı (6)

Çarpmanın ise şu özellikleri vardır:

$A(B + C) = AB + AC$ (soldan dağılma) (7)

$(A + B)C = AC + BC$ (sağdan dağılma) (8)

$A(BC) = (AB)C$ (birleşme) (9)

$AB \neq BA$ (değişme özelliği yoktur) (10)

$AB = 0$ olduğunda $A \neq 0$ veya $B \neq 0$ olabilir. (11)

$AB = AC$ olması $B = C$ olmasını gerektirmez. (12)

Şüphesiz matrislerle yapılabilecek işlemler burada gösterilen özelliklerle sınırlı değildir. Burada detaylıca göstermeyeceğimiz, matrisin transpozesi, determinantı, simetrik ve ters simetrik matrisler, matrisin izi vb. kavramlar lineer cebir kitaplarında bulunabileceği gibi sayısal çözümleme kitaplarında da bulunabilir (Tapramaz, 2002; Aktaş ve ark., 1991). Buradan da anlaşılacağı üzere matris işlemleri sayısal çözümlemenin de kapsamındadır.

Bu çalışmanın amacı matris cebirindeki işlemlerin bilgisayar ortamında yapılmasını olanak tanıyan algoritmaları vermektir. Bu algoritmalar Mathematica programlama dilinde yapılmıştır. Çalışmanın ikinci bölümünde sayısal çözümleme ile ilgili kısa bir giriş yapılmış ve daha sonra Mathematica programlama dilinin temel özelliklerinden bahsedilmiştir. Üçüncü bölümde ilgili algoritmalar verilmiş ve son bölümde ilgili algoritmalar hakkında bir sonuç verilmiştir.

2. Sayısal Çözümleme

Tarih boyunca bilim insanlarının doğanın ve hatta evrenin bir kısmını veya tamamını betimlemesiyle ortaya koyduğu model ve kuramlarla doğayı ve evreni anlama çabamız devam etmektedir. Matematik dilinde ortaya konulan bu model ve kuramların işleyişinde ortaya çıkabilecek bazı problemlerin çözümü en az orijinal problem kadar önemli bir problemdir. Hareketi anlamak adına Newton, türev ve integrali bir anlamda “icat ederek” kendi kuramının ortaya koyduğu problemleri çözmeyi amaçlamıştır. Türev ve integral aynı zamanda Newton’un çağdaşı olan Alman bilim insanı Leibniz tarafından da bulunmuştur. Türev ve integral olmasaydı hareket denklemlerinin çözümü neredeyse imkansızdı.

Bilimin genel anlamda yöntemi mevcut model ve kuramlar aracılığıyla problemler ortaya koymak ve onlara çözüm bulmaktır. Bu problemler bazen model ve/veya kuramlarda öngörülmemiş olabilir. Özellikle fen bilimleri ve mühendislik alanındaki genel yaklaşım, problemleri matematik dilinde ifade etmektir. Doğadaki veya evrendeki olguları modellemek en az onu çözmek kadar değerlidir. Bu çözüm her zaman tam ve doğru bir şekilde olmayabilir.

Sayısal çözümlemenin kapsamı ve sınırları kesin olarak belirlenememektedir. Matematiksel işlemlerin bulunmasına kadar sürdürülebilen bir başlangıcı olsa da sayısal çözümleme adı 1950’lerde ortaya çıkmıştır. Bu tarihlerde kurulan ABD Nümerik Analiz Enstitüsü’nün bu isimlendirmede pay sahibi olduğu söylenir (Çağal, 1989). Sayısal çözümlenin bir tanımı Aktaş ve ark. tarafından (1991) şu şekilde verilmiştir: Sayısal çözümlemenin amacı matematiksel modellerle ifade edilmiş çok çeşitli alanlara ait problemlerin çözümünü bilgisayar yardımıyla yaparak belirli bir hassaslığa sahip sonuç elde edebilmek için kullanılabilecek yöntemlerin bulunması, geliştirilmesi ve var olanların irdelenmesiyle en etkin olanının (olanlarının) saptanmasıdır.

Günümüzde bu konudaki genel yaklaşım, sayısal çözümlemenin bir disiplinler arası bilim olduğu yönündedir. Bundaki etken sayısal çözümlemenin hem bilgisayar bilimi ve teknolojisiyle hem de matematikle olan yakın ilgisidir. Sayısal çözümleme ile bilimsel, teknik ve hatta sosyal alandaki problemlerin çözümü yapılabilir.

Matematik diliyle ifade edilen problemlerin iki çözümü vardır: *analitik çözümler* ve *sayısal çözümler.* Analitik çözümler, tam çözümler olarak da bilinir. Analitik çözümler ilgili problemde var olan parametrelere ve değişkenlere bağlı bir ifade verdiğinden her zaman arzu edilen bir çözümdür çünkü parametreleri değiştirerek problemin incelenmesi ve olgunun anlamlanması kolaylaşır (Tapramaz, 2002). Örneğin kuantum mekaniğindeki bir olgu olan kuantum harmonik salınıcı analitik olarak çözülebilen nadir örneklerden biridir. Harmonik olmayan salınıcı modelleri belli yaklaşımlar altında harmonik salınıcıya benzetilerek çözüme sahip olmak istenir.

Sayısal çözümleme işte bu şekilde analitik olarak çözüme sahip olunamayan durumlarda kullanılır. Sayısal çözümleme ile elde edilen çözümler sayısal çözümlerdir ve çözümde yer alan parametreleri değiştirerek ancak sistemin bir kısmı açıklanabilir çünkü başlangıçta belirli bir kabul altında bu çözüme ulaşılmıştır.

Hangi tür problemlerin sayısal analiz kapsamına girdiği hususunda bir görüş birliği olmasa da analitik olarak çözümlenemeyen problemleri ele alması yaygın bir kabuldür. Bununla birlikte analitik olarak çözülebilen problemler de sayısal olarak çözümlenebilir. Analitik olarak çözümlenemeyen her problem sayısal olarak çözümlenebilir şeklindeki çıkarım yanlıştır. Örneğin birinci dereceden bir polinomun kökleri hem analitik hem de sayısal olarak bulunabilir: $p(x) = ax + b = 0 \Rightarrow x = -\frac{b}{a}$. İkinci dereceden bir polinomun kökleri de buna benzer şekilde bulunabilir fakat derecesi dörtten büyük gerçel katsayılı bütün polinomlar için köklerinin cebirsel yöntem veya kök alma yoluyla hesaplanmasını veren bir algoritma yoktur. Yani bu problem sayısal çözümleme ile de çözülemez. Bununla birlikte üçüncü ve dördüncü dereceli polinomların kökleri sayısal çözümleme ile bulunabilir.

2.1. Mathematica

Bilgisayar bilimleri ve teknolojisinin gelişmesiyle birlikte 1980'li yıllardan itibaren paket yazılımlar ortaya çıkmıştır. MATHEMATICA, MAPLE ve MATLAB bunlardan en popülerleridir. Bu paket yazılımlar programlama dillerinden biri veya birkaçı yardımıyla oluşturulmuştur. 1950'li yıllardan itibaren ortaya çıkan BASIC, FORTRAN, PASCAL, C, VISUAL BASIC ve son yıllarda popülerleşen PYTHON gibi programlama dilleri bu paket yazılımlarının temelini oluşturur. Bu programlama dilleri genel amaçlı olup sayısal hesaplar yapmak üzere de programlanabilirler. Mathematica, Maple ve Matlab hem sembolik hem de sayısal işlem yapabildikleri için özellikle karmaşık hesaplama gerektiren alanlarda kullanıcılarına büyük kolaylık sağlamıştır. Bu çalışmadaki algoritmalar Mathematica'da verilmiştir. Mathematica fizik alanında doktora derecesine sahip olan Stephen Wolfram tarafından ilk sürümü 1991 yılında yayımlanmış olan bir paket programdır. Mathematica matematiksel hesaplamalar yapan genel bir sistem olarak düşünülebilir. Mathematica'da hem sembolik hem de cebirsel işlem de yapılabilir. Buna ek olarak Mathematica bir programlama dili olarak da kullanılabilir. "Verilen işlem süreleri ve tezgah sayısı temelinde en iyi iş çizelgesi" ve "En küçük maliyetli, en büyük yakınlık değerli yerleşim planı belirleme" programları Mathematica ile yazılabilir (Çınar ve Çalışkan, 1995). Üstün bir hesaplama ortamı sağlamakla birlikte veri işleme ve grafik özellikleri son derece gelişmiştir.

Mathematica iki ana bölümden oluşur: matematiksel hesapları yapan kernel bölümü ve kullanıcı ile bilgisayarın iletişimini sağlayan front end bölümü (Çalışkan ve Çınar, 1995). Mathematica sahip olduğu kütüphane bakımından çok üstündür. Birçok matematiksel fonksiyon kütüphanesinde kullanıcılar için hazır olarak yer almaktadır. Mathematica sembolik bir yazılım olduğundan veri girişi çok kolaydır. Dolayısıyla yoğun hesaplamalar gerektiren işlemler için veri girişinde ve işlenmesinde zaman kaybını ortadan kaldırır.

Python ve C++ genel amaçlı programlama dilleridir. Matlab ve Mathematica daha çok sayısal hesaplamalarda kullanılan programlardır. Matlab, sayısal hesaplamalar ve grafik işlemleri için ayrı paketler sunabilmektedir. Örneğin adi diferansiyel denklemleri çözmek için Matlab'ın ODE Solver paketi kullanılmalıdır. Grafik ve görselleştirme işlemleri de ayrı paketler halinde sunulmaktadır. Mathematica'da böyle bir durum yoktur. Sembolik ve fonksiyonel bir programlama ara yüzü de sunduğundan son derece kullanıcı dostudur.

Mathematica'nın sahip olduğu yaklaşık 5000 yerleşik komut hesaplama işlemlerini kolaylaştırmaktadır. Bu yerleşik komutlar grafiklerin oluşturulması ve hesaplamaların tamamlanması için gereken hazır fonksiyonlardır. Teknik ve detaylı hesaplama gerektiren sinir ağları, makine öğrenmesi, görüntü işleme, veri işleme ve görselleştirme özelliklerini sunan üstün bir programlama aracıdır (Wolfram, 2018). Mathematica her türlü sembolik, sayısal ve grafiksel işlemleri yapabilen bir üründür.

Dilimizde Mathematica ile ilgili Türkçe kaynak ne yazık ki sınırlıdır (Çınar ve Çalışkan, 1995; Cesur, 2015; Sınıksıran ve Aktütün, 2009). Bu çalışmanın amaçlarından biri de Türkçemizde eksikliği bulunan bu alana bir katkı sağlamaktır.

3. Matris Cebiri

Bu bölümde matrislerde toplama, çıkarma, çarpma, determinant ve ters bulma işlemleri ile ilgili algoritmalar verilecektir.

3.1. Matrislerde Toplama

Tanım 2'den hatırlanacağı üzere matrislerin toplanması için boyutlarının aynı olması gerekmektedir. Şekil 1'de ilgili algoritma gösterilmektedir.

```
In[1]:= mattop[n1_, o_, p_] := Block[{top},
          k = Array[0 &, {n1, n1}];
          For[i = 1, i ≤ n1, i ++,
           For[j = 1, j ≤ n1, j ++,
            k[[i, j]] = o[[i, j]] + p[[i, j]]]];
          top = k];
        A = {{4, 1, 2, 1}, {1, 1, 1, 2}, {2, 1, 1, 1}, {1, 2, 1, 2}};
        B = {{2, 1, 2, 1}, {1, 1, 1, 2}, {2, 1, 1, 1}, {1, 2, 1, 2}};
        Print["A+B=", mattop[4, A, B] // MatrixForm]
```

$$A+B=\begin{pmatrix} 6 & 2 & 4 & 2 \\ 2 & 2 & 2 & 4 \\ 4 & 2 & 2 & 2 \\ 2 & 4 & 2 & 4 \end{pmatrix}$$

Şekil 1. Mathematica'da matris toplamını veren bir program

Bu programda A ve B matrislerinin boyutları 4 olarak alınmıştır. `mattop` adında matris boyutuna ve toplanacak matrislere göre bir fonksiyon tanımlanarak matris toplama işlemi gerçekleştirilmiştir. Boyut sayısı ve A ve B matrisleri değiştirilerek program yeniden kullanılabilir.

3.2. Matrislerde Çıkarma

Tanım 2'den hatırlanacağı üzere matrislerin çıkarılması için boyutlarının aynı olması gerekmektedir. Şekil 2'de ilgili algoritma gösterilmektedir.

```
In[1]:= matcikar[n1_, o_, p_] := Block[{fark},
          k = Array[0, {n1, n1}];
          For[i = 1, i ≤ n1, i ++,
           For[j = 1, j ≤ n1, j ++,
            k[[i, j]] = o[[i, j]] - p[[i, j]]]];
          fark = k];
        A = {{4, 1, 2, 1}, {1, 0, -3, 5}, {7, -8, 4, 6}, {1, -4, -5, 9}};
        B = {{3, 1, 4, 6}, {2, 1, -6, 2}, {4, 0, 3, 1}, {3, 5, -2, 2}};
        Print["A-B=", matcikar[4, A, B] // MatrixForm]
```

$$\text{A-B=}\begin{pmatrix} 1 & 0 & -2 & -5 \\ -1 & -1 & 3 & 3 \\ 3 & -8 & 1 & 5 \\ -2 & -9 & -3 & 7 \end{pmatrix}$$

Şekil 2. Mathematica'da matris çıkarılmasını veren bir program

Toplama işlemine benzer şekilde *A* ve *B* matrisleri ve boyutları girilerek çıkarma işlemi gerçekleştirilmiştir.

3.3. Matrislerde Çarpma

Tanım 3'ten hatırlanacağı üzere matrislerle çarpma işleminin yapılabilmesi için birinci matrisin sütun sayısı ile ikinci matrisin satır sayıları birbirlerine eşit olmalıdır. Şekil 3'te Mathematica'da çarpma işlemi gösterilmiştir.

```
In[1]:= matcarp[v_, y_, o_, p_] := Block[{tot},
          cm = Array[0 &, {v, y}];
          ust = Dimensions[o][[2]];
          For[im = 1, im ≤ v, im ++,
           For[jm = 1, jm ≤ y, jm ++,
            cm[[im, jm]] = 0;
            For[km = 1, km ≤ ust, km ++,
             cm[[im, jm]] = cm[[im, jm]] + o[[im, km]] * p[[km, jm]]]]];
          tot = cm];
        A = {{1, 2, -3, 5}, {2, -4, 7, 6}, {2, 1, 4, 0}, {-3, -2, -1, 2}};
        B = {{5, 2, 6, 1}, {4, 2, 1, -8}, {9, 7, 5, 6}, {-1, -2, -1, 6}};
        Print["A.B=", matcarp[4, 4, A, B] // MatrixForm]
```

$$\text{A.B=}\begin{pmatrix} -19 & -25 & -12 & -3 \\ 51 & 33 & 37 & 112 \\ 50 & 34 & 33 & 18 \\ -34 & -21 & -27 & 19 \end{pmatrix}$$

Şekil 3. Mathematica'da çarpma işlemini veren bir program

Bu programda 4×4 matrisler kullanılmıştır. Program matris boyutları değiştirilerek yeniden çalıştırılabilir.

3.4. Matrislerde Determinant

Tanım olarak bir kare matrisin determinantı şu şekilde gösterilir

$$|A| = \det A = \begin{vmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \ddots & \cdots \\ a_{m1} & a_{m2} & \cdots & \cdots & a_{mn} \end{vmatrix} \tag{13}$$

ve önceden belirlenmiş baz aritmetik işlemlerle aynı matrisin elemanları üzerindeki operasyonlarla bulunur (Çağal, 1989). Determinantın özellikleri şu şekilde sıralanabilir (Aktaş ve ark., 1991):

i. Eğer matrisin herhangi bir satırının veya sütununun bütün elemanları sıfırsa, bu matrisin determinantı da sıfırdır.

ii. Bir matrisin determinantı ile tersinin determinantı birbirine eşittir.

iii. Bir matrisin iki satırı veya iki sütunu yer değiştirirse determinantlarının değeri aynı kalır fakat işareti değişir.

iv. Eğer bir matrisin bir satırının veya sütununun bütün elemanları bir k sayısı ile çarpılırsa bu matrisin determinantı da k ile çarpılmış olur.

v. Eğer bir matrisin iki satırının veya sütununun elemanları birbirine belli bir oran içerisinde eşitse, bu matrisin determinantı sıfır olur.

vi. Eğer bir matrisin herhangi bir satırının (sütununun) bütün elemanları bir k sayısı ile çarpılıp, bir diğer satırın (sütunun) elemanları ile toplanırsa matrisin determinantı değişmez.

vii. Verilen A ve B gibi iki matrisin çarpımının determinantı, matrislerin determinantlarının çarpımına eşittir: $\det AB = \det A \det B$.

viii. Bir üst üçgen veya alt üçgen matrisin determinantı köşegenindeki elemanları çarpımına eşittir: $\det L = l_{11} l_{22} \cdots l_{nn}$, $\det U = u_{11} u_{22} \cdots u_{nn}$

Üst üçgen ve alt üçgen matris şu şekilde tanımlanır. Eğer bir kare matrisin elemanları $i > j$ için $a_{ij} = 0$ oluyorsa bu matrise üst üçgen matris denir ve şu şekilde gösterilir:

$$U = \begin{bmatrix} u_{11} & u_{12} & \cdots u_{1n} \\ 0 & u_{22} & \cdots u_{2n} \\ \vdots & \vdots & \ddots \ \vdots \\ 0 & 0 & \cdots u_{nn} \end{bmatrix} \tag{14}$$

Eğer bir kare matrisin elemanları $i < j$ için $a_{ij} = 0$ oluyorsa bu matrise alt üçgen matris denir ve şu şekilde gösterilir:

$$L = \begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix} \tag{15}$$

Bu çalışmada matrislerin determinantını bulmak için Gauss yok etme ve Chio yöntemi kullanılacaktır.

3.4.1. Gauss yok etme yöntemi ile matris determinantı

Bu yöntemde matrisin satırları ile yapılan işlemler ile matris üst üçgen matrise dönüştürülür ve sonra determinant hesaplanır. Üst üçgene dönüştürme işlemi boyutu 3×3 olan bir matris için şu şekilde yapılabilir:

$$|A| = \begin{vmatrix} a_{11} & a_{21} & a_{31} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = \begin{vmatrix} a_{11} & a_{21} & a_{31} \\ 0 & a'_{22} & a'_{23} \\ 0 & a'_{32} & a'_{33} \end{vmatrix} \begin{matrix} 2.satır - \frac{a_{21}}{a_{11}} \times 1.satır \\ 3.satır - \frac{a_{31}}{a_{11}} \times 1.satır \end{matrix} \tag{16}$$

Benzer şekilde işlemlere devam ettirilerek verilen matris üst üçgene dönüştürülür. Gauss yok etme yöntemi ile matris determinantını veren algoritma Şekil 4'te gösterilmektedir.

```
In[1]:= n = 4;
Array[A, {n, n}];
A = {{1, 4, 5, 2}, {-5, 0, 1, 9}, {2, 3, 5, -8}, {3, 1, 7, 6}};
Print["Verilen matris: A=", A // MatrixForm];
For[k = 1, k ≤ n - 1, k++,
  For[i = k + 1, i ≤ n, i ++,
   p = A[[i, k]] / A[[k, k]];
   For[j = 1, j ≤ n, j ++,
    A[[i, j]] = A[[i, j]] - p * A[[k, j]]]]];
det = 1;
For[i = 1, i ≤ n, i ++,
  det = det * A[[i, i]]];
Print["Üst üçgen matris: A'=", A // MatrixForm];
Print["det(A) = ", det]
```

Verilen matris: A= $\begin{pmatrix} 1 & 4 & 5 & 2 \\ -5 & 0 & 1 & 9 \\ 2 & 3 & 5 & -8 \\ 3 & 1 & 7 & 6 \end{pmatrix}$

Üst üçgen matris: A'= $\begin{pmatrix} 1 & 4 & 5 & 2 \\ 0 & 20 & 26 & 19 \\ 0 & 0 & \frac{3}{2} & -\frac{29}{4} \\ 0 & 0 & 0 & \frac{409}{10} \end{pmatrix}$

det(A) = 1227

Şekil 4. Gauss yok etme yöntemi ile determinant alma

3.4.2. Chio yöntemi ile matris determinantı

Bu yöntem determinantı alınacak matrisin boyutu en sonunda (2×2) olacak şekilde kalana kadar bir takım işlemler yapmaktan ibarettir. Boyutu $(n \times n)$ olan bir matris için şu şekilde yapılabilir (Aktaş ve ark., 1991):

$$|A| = \frac{1}{a_{11}^{n-2}} \left\| \begin{matrix} \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} & \cdots & \begin{vmatrix} a_{11} & a_{1n} \\ a_{21} & a_{2n} \end{vmatrix} \\ \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} & \cdots & \begin{vmatrix} a_{11} & a_{1n} \\ a_{31} & a_{3n} \end{vmatrix} \\ \vdots & \vdots & \cdots & \vdots \\ \begin{vmatrix} a_{11} & a_{13} \\ a_{n1} & a_{n2} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{n1} & a_{n3} \end{vmatrix} & \cdots & \begin{vmatrix} a_{11} & a_{1n} \\ a_{n1} & a_{nn} \end{vmatrix} \end{matrix} \right\| \tag{17}$$

Yani boyut her seferinde yukarıda gösterildiği gibi azaltılır ve işlemin sonunda (2×2)'lik determinant elde edilinceye kadar devam ettirilir. Örnek olarak şu matris ele alınırsa

$$|A| = \begin{vmatrix} 2 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \\ 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 2 \end{vmatrix} = \frac{1}{4^{4-2}} \begin{vmatrix} 1 & 0 & -3 \\ 0 & -2 & 0 \\ 3 & 0 & 3 \end{vmatrix} = \frac{1}{4} \frac{1}{1^{3-2}} \begin{vmatrix} -2 & 0 \\ 0 & -6 \end{vmatrix} = 3 \tag{18}$$

elde edilir. Şekil 5'te Gauss yok etme yönteminde kullanılan matrisin Chio yöntemiyle determinantının bulunması gösterilmiştir.

```
In[1]:= n = 4;
        Array[A, {n, n}];
        Array[B, {n, n}];
        A = {{1, 4, 5, 2}, {-5, 0, 1, 9}, {2, 3, 5, -8}, {3, 1, 7, 6}};
        B = A;
        Print["Verilen Matris: ", A // MatrixForm];
        determinant[a1_, a2_, a3_, a4_] := a1 * a4 - a2 * a3;
        m = n;
        s = 1;
        kat = 1;
            For[r = 1, r ≤ n - 2, r++,
                   k = A[[1, 1]];
                   carp = 1/k^(m-r-1);
                   kat = kat * carp;
                   For[i = 1, i ≤ m - r, i ++,
                         For[j = 1, j ≤ m - r, j ++,

            cijk = determinant[k, A[[1, j + 1]], A[[i + 1, 1]], A[[i + 1, j + 1]]];
                               B[[i, j]] = cijk]];
                   A = B;

          If[r == n - 2, det = kat * determinant[A[[1, 1]], A[[1, 2]], A[[2, 1]], A[[2, 2]]];
                            s = 0]];
        Print["Matrisin Determinantı= ", det]

        Verilen Matris: ( 1  4  5  2
                         -5  0  1  9
                          2  3  5 -8
                          3  1  7  6 )

        Matrisin Determinantı= 1227
```

Şekil 5. Chio yöntemi ile determinant alma

3.5. Matrislerde Ters Bulma

3.5.1. Gauss-Jordan yöntemi

Bu yöntemde amaç ardışık satır işlemleriyle $(n \times n)$ boyutlu bir matrisin tersini bulmaktır. Gauss-Jordan yok etme yöntemi ile bir matrisin tersini bulmak için verilen matris birim matrisle genişletilmiş bir şekilde yazılır:

$$(A|I) = \left(\begin{array}{cccc|cccc} a_{11} & a_{12} & \cdots & a_{1M} & 1 & 0 & 0 & 0 \\ a_{21} & a_{22} & \cdots & a_{2M} & 0 & 1 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{M1} & a_{M2} & \cdots & a_{MM} & 0 & 0 & 0 & 1 \end{array}\right) \tag{19}$$

Bu gösterimde soldaki matris üzerinde satır işlemleri uygulanarak birim matris haline getirilmeye çalışılır. Birim matris haline getirildiğinde sağda oluşan matris A matrisinin tersi olur.

Detaylı işlemleri göstermeden bir örnek vermek gerekirse A matrisi şu şekilde verilsin (Tapramaz, 2002):

$$A=\begin{bmatrix} 1 & 2 & -3 & 4 \\ -2 & 4 & 3 & 5 \\ 9 & 6 & 0 & 2 \\ 2 & -3 & 4 & -5 \end{bmatrix}.$$

Eklemeli matris biçiminde yazılırsa

$$\left(\begin{array}{cccc|cccc} 1 & 2 & -3 & 4 & 1 & 0 & 0 & 0 \\ -2 & 4 & 3 & 5 & 0 & 1 & 0 & 0 \\ 9 & 6 & 0 & 2 & 0 & 0 & 1 & 0 \\ 2 & -3 & 4 & -5 & 0 & 0 & 0 & 1 \end{array}\right)$$

$$\Rightarrow\left(\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & 0.284192 & -0.014210 & 0.026643 & 0.223801 \\ 0 & 1 & 0 & 0 & -0.609236 & -0.019538 & 0.161634 & -0.442274 \\ 0 & 0 & 1 & 0 & 0.087034 & 0.145648 & -0.0223091 & 0.206039 \\ 0 & 0 & 0 & 1 & 0.548845 & 0.122558 & -0.104796 & 0.319716 \end{array}\right)$$

olarak bulunur ki sağdaki bu matris A matrisinin tersidir. Ok işareti ile gösterilen kısımda matris üzerinde satır işlemleri yapılmıştır. Şekil 6'da Gauss-Jordan yok etme yönteminin programı verilmiştir.

```
In[1]:= nx = 4;
        ax = {{1, 2, -3, 4}, {-2, 4, 3, 5}, {9, 6, 0, 2}, {2, -3, 4, -5}};
        gaujor[n_, c_] := Module[{gj},
           b = IdentityMatrix[n];
           a = Array[0 &, {n, n}];
           a = c;
           For[i = 1, i ≤ n, i ++,
                p = a[[i, i]];
                For[j = 1, j ≤ n, j ++,
                     a[[i, j]] = a[[i, j]] / p;
                     b[[i, j]] = b[[i, j]] / p];
                    For[k = 1, k ≤ n, k ++,
                         If[k =!= i,
                          p = a[[k, i]];
                      For[j = 1, j ≤ n, j ++,
                           a[[k, j]] = a[[k, j]] - p * a[[i, j]];
                           b[[k, j]] = b[[k, j]] - p * b[[i, j]]]]]]];
           gj = b];
        matcarp[v_, y_, o_, p_] := Block[{tot},
           cm = Array[0 &, {v, y}];
           ust = Dimensions[o][[2]];
           For[im = 1, im ≤ v, im ++,
            For[jm = 1, jm ≤ y, jm ++,
             cm[[im, jm]] = 0;
             For[km = 1, km ≤ ust, km ++,
              cm[[im, jm]] = cm[[im, jm]] + o[[im, km]] * p[[km, jm]]]]];
           tot = cm];
        Print["A=", ax // MatrixForm, "; A^-1=", N[gaujor[nx, ax]] // MatrixForm]
        Print["A.A^-1=", matcarp[4, 4, ax, gaujor[nx, ax]] // MatrixForm]
```

$$A=\begin{pmatrix} 1 & 2 & -3 & 4 \\ -2 & 4 & 3 & 5 \\ 9 & 6 & 0 & 2 \\ 2 & -3 & 4 & -5 \end{pmatrix};\ A^{-1}=\begin{pmatrix} 0.284192 & -0.0142096 & 0.026643 & 0.223801 \\ -0.609236 & -0.0195382 & 0.161634 & -0.442274 \\ 0.0870337 & 0.145648 & -0.0230906 & 0.206039 \\ 0.548845 & 0.122558 & -0.104796 & 0.319716 \end{pmatrix}$$

$$A.A^{-1}=\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Şekil 6. Gauss-Jordan yöntemi ile matris tersi alma

4. Sonuç ve Tartışma

Bu çalışmanın amacı bilimsel hesaplamalarda çok sık kullanılan Mathematica paket programının bazı sayısal çözümleme işlemlerinde örnek kullanımlarının gösterilmesidir. Şüphesiz Mathematica gibi paket programlar matrislerin toplamını, çıkarmasını, çarpımı, determinantı ve tersinin bulunması sahip olduğu yerleşik işlemler sayesinde yapabilmektedir. Bu tip işlemler çözüm yöntemi hakkında bilgi sahibi olunmadan kullanılabilir ki paket programlar genellikle bu kesime hitap eder. Bu tip bir kullanım çözümleme teorileriyle ilgilenen bilim insanlarını pek tatmin etmeyebilir (Tapramaz, 2002). Paket programlarda matematiğe çok fazla hakim olmadan gerekli işlemleri bilgisayara yaptırabilirsiniz.

Bu çalışmada son derece etkin bir program olan Mathematica'nın programlama yönüne dikkat çekilmiştir. Buradaki amacımız Mathematica'nın programlamaya uygun bir dil olduğunu göstermekti. Çalışmanın hem lisans öğrencilerine hem de öğretim üyelerine yararlı olacağına inanıyoruz. Şüphesiz bu çalışmada elde edilen sonuçlar Matlab, Python, Fortran gibi başka dillerde de elde edilebilir. Mathematica'nın bu dillere göre avantajı sahip olduğu fonksiyon kütüphanesi ve üstün grafik özellikleridir. Sayısal çözümleme derslerinde Mathematica'nın kullanımı yararlı olacaktır.

Kaynaklar

Akın, Ö. (2007). Çeviri editörü. *Uygulamalı Lineer Cebir*. Palme Yayıncılık: Ankara.
Aktaş, Z., Öncül, H., Ural, S. (1984). *Sayısal Çözümleme.* Orta Doğu Teknik Üniversitesi Yayınları: Ankara.
Cesur, Y. (2015) *Diferansiyel Denklemler ve Mathematica,* Kişisel Yayın
Çağal, B. (1989). *Sayısal Analiz.* Seç Yayın Dağıtım: İstanbul.
Çallıalp, F. ve Kuruoğlu, N. (1996). *Lineer Cebir*. Ondokuz Mayıs Üniversitesi Yayınları: Samsun.
Çınar, M., Çalışkan, F. (1995). *Mathematica ile Programlama.* Beta Basım Yayım Dağıtım: İstanbul.
Çınar, M. (2000) *Mathematica 3.0 ve 4.0 Sürümü*, Seçkin Yayıncılık:Ankara
Çiftçi, S. (2015). *Lineer Cebir.* Dora Yayıncılık: Bursa.
Hacısalihoğlu, H., (1999), Çeviri editörü. *Matris işlemleri- Schaum's.* Ankara: Nobel Yayın Dağıtım.
Sabuncuoğlu, A., (2017). *Lineer Cebir-Mühendislik ve İstatistik Bölümleri İçin.* Ankara: Nobel Yayın Dağıtım
Sınıksıran, E. ve Aktütün, A. (2009) *Matematik ve İstatistik Uygulamalarıyla Mathematica,* Türkmen Kitapevi: İstanbul
Tapramaz, R. (2002). *Sayısal Çözümleme.* Literatür Yayıncılık: İstanbul.
Taşçı, D. (2006). *Lineer Cebir.* Gazi Kitabevi: Ankara.
Yüce, S. (2015). *Lineer Cebir.* Ankara: Pegem Akademi Yayıncılık.
Wolfram Co., www.wolfram.com (Erişim tarihi 28.05.2018)