Journal of Pure and Applied Mathematics: Advances and Applications
Volume 27, Number 1, 2024, Pages 1-23
Available at http://scientificadvances.co.in
DOI: http://dx.doi.org/10.18642/jpamaa_7100122300

SOLVING MANY COMPLICATED REAL-LIFE PROBLEMS USING LINEAR ALGEBRA

HENA RANI BISWAS and SAIDA ISLAM NOURIN

Department of Mathematics
University of Barishal
Barishal-8200
Bangladesh
e-mail: biswas.hena@yahoo.com

Abstract

In this paper, we explore the wide-ranging applications of linear algebra in solving complex problems across various industries and disciplines. More specially, we will focus on the phenomena of solving many complicated problems by using it. To perform such an analysis, we will use several critical concepts of linear algebra, including the following: matrices, vector spaces, difference equations, eigenvalues and eigenvectors, etc. In this sense, some practical applications related to computer graphics, geometry, areas, and volumes are presented, along with some problems connected to sports and investments.

2020 Mathematics Subject Classification: 15A18, 15A29.
Keywords and phrases: linear algebra, matrix, determinant, analytic geometry, computer graphics, GeoGebra, eigenvalue, eigenvector.
Received April 13, 2024

1. Introduction

Linear algebra is a subject essential to learning since linear models based on linear algebra are the base models in statistical learning. The origins of linear algebra in the West can be traced to Rene Descartes in 1637, who developed the idea of coordinates using a geometric method that is now known as Cartesian geometry [4].

The first contributions date back to 1843, when Irish scientist William Rowan Hamilton created quaternions and coined the term "vector." When imaginary units $\hat{\imath}$, $\hat{j}$ and $\hat{k}$ are added to real numbers, the result is an extension known as a quaternion. The term matrix was finally introduced in 1848 by English Mathematician James Joseph Sylvester [4]. Important ideas in linear algebra include vectors, matrices, systems of linear equations, vector spaces, eigenvalues, eigenvectors, and more. Numerous fields of study, including probability theory, statistics, engineering, economics, and statistics, use matrices. Similar to eigenvalues and eigenvectors, which are strong mathematical tools with a wide range of engineering applications, vector spaces are widely used in common situations, namely wherever functions with values in some fields are involved. Engineers can learn a great deal about system behaviour, stability analysis, optimization, and problem-solving from these concepts, which range from structural analysis and control systems to image processing, power systems, and fluid dynamics.

Additionally, new opportunities for engineering courses have arisen as a result of the advancement of technology and digital resources, where computers are used to solve mathematically challenging issues, and visualizations and simulations play a major role [3]. Additionally, these advancements have altered the environment for instruction and learning. Increasingly realistic virtual simulations of very complicated engineering challenges can be solved thanks to new technology. Innovative techniques used at different colleges are characterized by students actively learning in real-life activities and practicing self-regulation on personalized learning routes.

The purpose of this paper is to introduce some interesting real-world applications of linear algebra. Here we introduce how linear algebra is used for image processing and data transmission. One common application of matrices in image processing is image transformation and matrices offer an ordered and structured means of representing data, which facilitates processing and the extraction of significant insights. Here discussed the powerful statistical method of principal component analysis (PCA) using linear algebra. Linear algebra has an important application in network and graph theory and also for finding the shortest path as well as cryptography which we also exercise here.

2. Results and Discussion

Linear algebra is a fundamental tool in mathematics and statistics. Numerous academic disciplines can benefit from its application, such as science, business studies, economics, photoshop, photo editing, and three-dimensional gaming. It is widely used in the engineering domains. In data science and machine learning, it is very helpful. In this chapter, we will examine some important real-world uses of linear algebra.

One of the novel aspects of the instructional approaches employed is the application of linear algebra and geometry problems to real-world scenarios [9, 10]. A few illustrations are provided. In this section, we will discuss application of some problems of linear algebra in real life.

2.1. Application of linear algebra in image processing

In a digital image processing system, a pixel is the smallest unit that has image control. The quality of the camera determines how many pixels are in a photograph. A photograph with more pixels than any other has higher quality than any other picture with fewer pixels. It was expressed in megapixels by several cameras. A camera with a higher megapixel count will generally be of higher quality, which more than others can provide us with a clear image [20].

Color bleending produces the coloration of pixels. Three colors are based which stand for RGB, or red, green, and blue. This implies that we can produce $256 \times 256 \times 256$ distinct colors using the RGB method. For computer displays, the RGB scheme is utilized, whereas other colors are used for printing.

A digital image is nothing more than a collection of single-color pixels when we zoom in or examine it closely. Which appear to be square.

Then, a numerical value can be used to represent those colors. Which is equivalent to a square image composed of one million pixels where a 1000×1000 matrix could be used to represent 1000 on each edge, where each pixel's color values are entered [19].

Example 2.1.1. We can multiply an image by a constant. Image addition and subtraction are also possible. Here is an example for multiplication by a constant. By applying **MATLAB** code, we get the image.

[Program 1]

Figure 2.1.1. Original image.

Figure 2.1.2. Multiplication of image by 1.5.

Here all coefficients of the matrix are multiplied by 1.5 and the second image is lighter than the original image. Thus, if we multiply an image by a constant then this process changes the contrast of the image.

2.2. Linear algebra in data science

Data science is the study of data to obtain significant commercial insights. It is a branch of computer engineering, mathematics, statistics, and artificial intelligence that analyzes data by taking computation and interpretation into account. Here, we go over practical algorithms for working with eigenvalues and eigenvectors. We'll talk about how linear algebra is used in principal component analysis (dimension reduction).

2.2.1. Principal component analysis

Principal component analysis (PCA) was first introduced by British Mathematician Karl Pearson in 1901. This approach is widely used to analyze large datasets with many dimensions. By gathering as much data as possible, we can improve the data's ability to explain patterns and enable the visualization of multidimensional data. It is a statistical method for lowering a data set's dimensionality. By linear transformation, it changes the data into a new coordinate system where each variation can be explained by a smaller number of dimensions than the original data [8].

The mean of m variables is $\overline{M} = \frac{1}{n}(\overline{a_1} + \overline{a_2} + \ldots \overline{a_n})$.

Here the center of mass is the origin and $\overline{a_\imath}$ are the vectors. Let A be the $m \times n$ matrix $A = |\overline{a_1} - \overline{M}| \ldots |\overline{a_n} - \overline{M}|$ and define the covariance matrix $B = \frac{1}{n-1} AA^T$, where B is symmetric.

Suppose $\overline{a_1} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$, $\overline{a_2} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$, $\overline{a_3} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix}$, $\overline{M} = \begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ M_4 \end{bmatrix}$, so that

$$A = \begin{bmatrix} x_1 - M_1 & y_1 - M_1 & z_1 - M_1 \\ x_2 - M_2 & y_2 - M_2 & z_2 - M_2 \\ x_3 - M_3 & y_3 - M_3 & z_3 - M_3 \\ x_4 - M_4 & y_4 - M_4 & z_4 - M_4 \end{bmatrix}.$$

Now $B_{11} = \frac{1}{3-1}((x_1 - M_1)^2 + (y_1 - M_1)^2 + (z_1 - M_1)^2)$,

$B_{21} = \frac{1}{3-1}((x_1 - M_1)(x_2 - M_2) + (y_1 - M_1)(y_2 - M_2) + (z_1 - M_1)(z_2 - M_2))$, which is the covariance of first and second variables. Here the i-th entry of the diagonal of B_{ii} is the variance of i-th variable and the ij-th entry with $i \neq j$ is the covariance between the i-th and j-th variables.

2.2.2. Dimension reduction

The most common method of dimension reduction is PCA. Reducing dimensionality is a technique for making a model less complex. Which includes feature extraction and feature selection as its two primary dimensionality reduction categories. A subset of the original feature is chosen through feature selection, and information is extracted from the feature to create a new feature subspace. Now we discuss in short how PCA performs on data.

(i) Calculate the mean of the n-dimensional data.

(ii) Find eigenvalues of the matrix B and also eigenvectors.

(iii) If a small number of eigenvalues is bigger than others, then the dimensionality reduction is possible. Then check which variables are much more important than others and which factors have the same or opposite sign than the others [8].

Example 2.2.2.1. Apply PCA to decrease the dimension from two to one given the following data.

Table 2.2.2.1. Data set for x and y variables

Feature	Ex.1	Ex. 2	Ex. 3	Ex. 4
x	5	4	8	9
y	6	2	3	11

Step 1: Dataset: There are $n = 2$ features & samples, $N = 4$.

Step 2: Computation mean of variables:

$$\bar{x} = \frac{5 + 4 + 8 + 9}{4} = 6.5,$$

$$\bar{y} = \frac{6 + 2 + 3 + 11}{4} = 5.5.$$

Step 3: Computation of covariance matrix:

Ordered pairs are: $(x, x), (x, y), (y, x), (y, y)$.

$$\Rightarrow \text{Cov}(x, x) = \frac{1}{N-1}\sum_{k=1}^{N}(x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j)$$

$$\Rightarrow \frac{1}{4-1}((5-6.5)^2 + (4-6.5)^2 + (8-6.5)^2 + (9-6.5)^2) = 5.67.$$

Similarly, $\text{Cov}(x, y) = \frac{1}{4-1}((5-6.5)(6-5.5) + (4-6.5)(2-5.5) + (8-6.5)(3-5.5) + (9-6.5)(11-5.5)) = 6.$

$\Rightarrow \text{Cov}(y, x) = 6$ and $\text{Cov}(y, y) = 16.33$.

So covariance matrix $S = \begin{bmatrix} 5.67 & 6 \\ 6 & 16.33 \end{bmatrix}$.

Step 4: Find Eigenvalue, Eigenvector, Normalized Eigenvector: Eigenvalue of S, $\lambda_1 = 19.0256$ and $\lambda_2 = 2.9744$. Here $\lambda_1 > \lambda_2$. So, the first principal component is $\lambda_1 = 19.0256$. Now the eigenvector of λ_1 is $\boldsymbol{v_1} = \begin{bmatrix} 6 \\ -13.3556 \end{bmatrix}$. Normalize eigenvector $\boldsymbol{v_1}$,

$$\boldsymbol{e_1} = \begin{bmatrix} \frac{6}{\sqrt{(6)^2 + (-13.3556)^2}} \\ \frac{-13.3556}{\sqrt{(6)^2 + (-13.3556)^2}} \end{bmatrix} = \begin{bmatrix} 0.4098 \\ -0.9122 \end{bmatrix}.$$

For λ_2, $\boldsymbol{e_2} = \begin{bmatrix} 0.9122 \\ 0.4098 \end{bmatrix}$.

Step 5: Derive new dataset:

$$P_{11} = \boldsymbol{e_1}^T \begin{bmatrix} 5 - 6.5 \\ 6 - 5.5 \end{bmatrix} = -1.0708,\ P_{12} = 2.1682,\ P_{13} = 2.8952,$$

$$P_{14} = -\ 3.9926.$$

PC1	− 1.0708	2.1682	2.8952	− 3.9926

Now we have the reduced dataset with the dimension one. The blue line indicates the new axes and the blue dots are the new dataset.

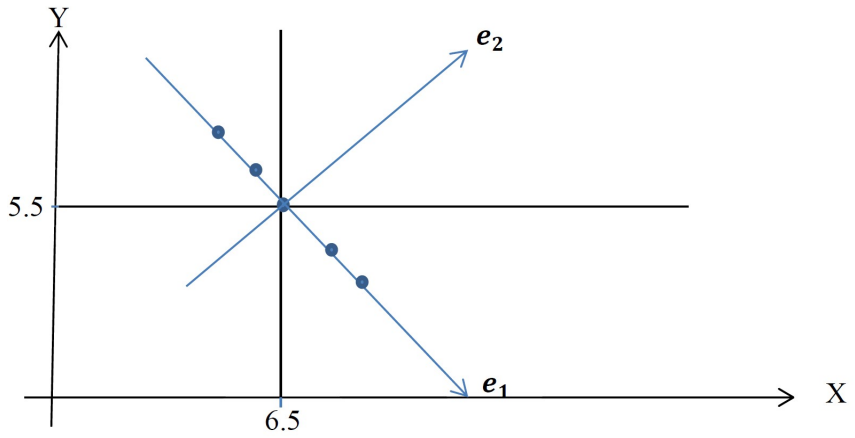

Figure 2.2.2.2. New axes by using PCA.

2.3. Application of linear algebra in networks, graph theory & finding shortest path

Graphs can represent a lot of things. People and who are their friends, connections on a dating app, networks of cities and how they are connected to websites and how they link to each other, and so on. There are many useful ways to represent a graph [1]. This graph consists of a non-empty set of vertices and multiple edges. We can easily understand a graph where it is connected and where it is disconnected by network matrices. Network matrices show how objects in a system are related to another which is more structured and can be easier to read. This matrix

is known as an adjacency matrix [16]. An adjacency matrix is a square matrix that is used to describe a finite graph in graph theory and computer science. The matrix's entries show whether or not pairs of vertices in the graph are adjacent. The adjacency matrix in the particular situation of a finite simple graph is a matrix with zeros on its diagonal. We can understand it by some examples**:**

Example 2.3.1. Consider a big city which has four towns named (A, B, C, D). Some of the local train information for these towns is shown in the graph:

A train runs from A to B and another train runs from B to A. Similarly for A to C and C to A, B to C and C to D, and D to C. Notice that there is no train from C to B. By using matrix, we can easily arrange this information and understand it:

	A	B	C	D
A	0	1	1	1
B	1	0	1	0
C	1	0	0	1
D	0	0	1	0

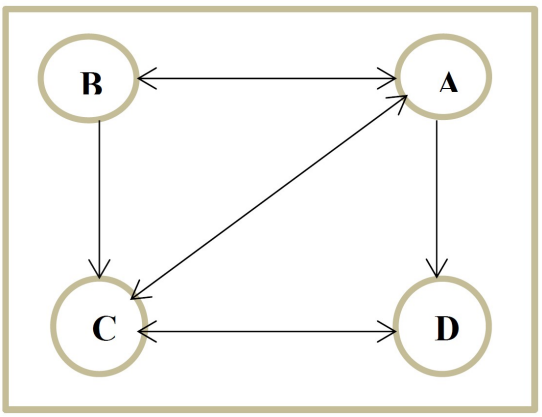

Figure 2.3.1. A simple directed graph for train schedule.

Where if the train comes from one to another station, then the value is 1, otherwise the value is 0.

Example 2.3.2. Matrix is very useful for telecom networking. Let us consider four cities (A, B, C, D). People in these cities use four types of

networks. These are Grameenphone (G), Banglalink (B), Robi (R), Teletalk (T). 1000 customers are using network G, 200 are using B, and 100 are using T in city A. In city B the network G, B, R, and T users are respectively (500, 0, 300, 100), in C (0, 400, 0, 50), and in D (0, 0, 300, 200). We can arrange this easily by a graph:

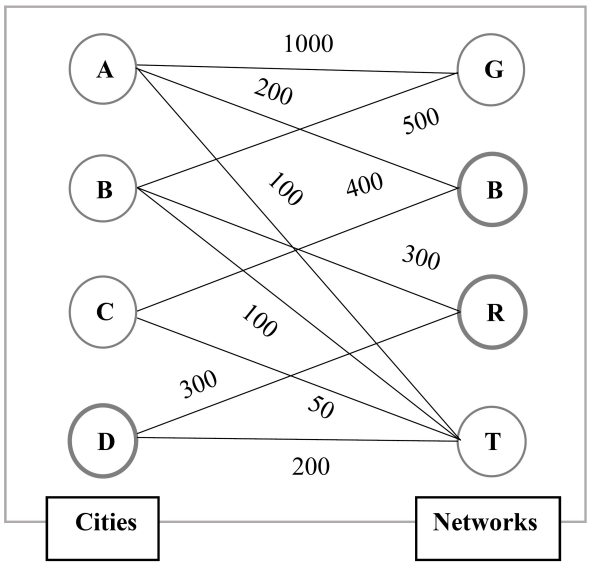

Figure 2.3.2. Weighted graph modelling a telecom networking.

In matrix form:
$$\begin{array}{c} \\ \mathbf{A} \\ \mathbf{B} \\ \mathbf{C} \\ \mathbf{D} \end{array} \begin{bmatrix} 1000 & 200 & 0 & 100 \\ 500 & 0 & 300 & 100 \\ 0 & 400 & 0 & 50 \\ 0 & 0 & 300 & 200 \end{bmatrix}.$$

This short example shows how a matrix facilitates this network which is used by billions of customers. All this data is managed by the companies' network software. Their software is arranged through a matrix like this. That's why a matrix is a way of collecting information. It is a compact way to collect information.

2.3.1. Shortest path

The shortest path problem, as defined in graph theory, is the task of determining a path that minimizes the total weight of all the edges that connect two vertices or nodes in a graph. A weighted graph can be used to solve this issue. Weighted graphs are graphs in which each edge is assigned a number. Computer networks are modeled using weighted graphs. Weighted graphs can be used to analyze communication costs, computer response times over these lines, and computer distance. This graph has both directed and undirected options [16].

By using **MATLAB**, we can easily find the shortest path and draw this graph. For this, we have to know about sparse matrix. This matrix is a special case of a matrix in which the number of zero elements is much higher than the number of non-zero elements.

For example, suppose we find the shortest path between node-1 to node-4 and draw the weighted graph. Which is defined by a 4 by 4 matrix, $A = \begin{bmatrix} 0 & 20 & 0 & 50 \\ 10 & 0 & 0 & 10 \\ 0 & 60 & 0 & 50 \\ 0 & 0 & 30 & 0 \end{bmatrix}$. This can be directed and also undirected. The 2D array representation of the sparse matrix is given in the following table.

Table 2.3.1. Array representation of sparse matrix

Row	1	1	2	2	3	3	4
Column	2	4	1	4	2	4	3
Value	20	50	10	10	60	50	30

Here is how to use **MATLAB** to draw the graph and find the shortest path:

For directed graph:

[Program 2]

Then we get the weighted graph as the output and the shortest path is from node-1 to node-2 and then node-4 with distance $20 + 10 = 30$.

For undirected graph:

[Program 3]

Then we get the weighted graph as the output and the shortest path is node-1, node-2 and node-4 with distance $30 + 10 = 40$. The graph for both cases,

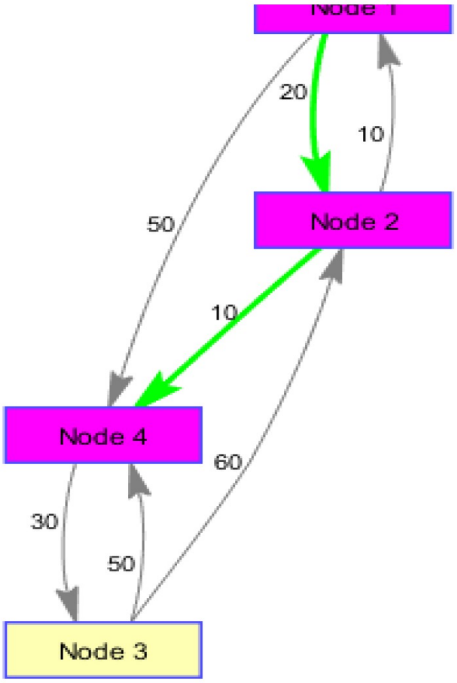

Figure 2.3.1. A directed graph of shortest path.

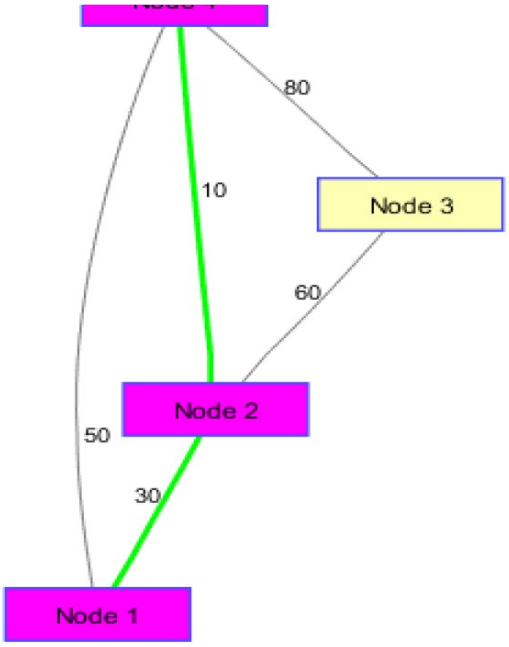

Figure 2.3.2. An undirected graph of shortest path.

Similarly finding shortest paths is important for solving many problems in our daily life. This method is used in different networks. Google map direction is the best example of finding the shortest path between two points. And obviously, the matrix has made this task easier for us. By using matrices, we can easily find the shortest path which is very helpful in our daily life.

2.4. Application of linear algebra in cryptography

The process of hiding information so that only the recipients can decipher and read it is known as cryptography. For thousands of years, people have utilized the art of cryptography to encode messages. Today, blank cards, computer passwords, and e-commerce all use cryptography. The foundation of modern cryptography is mathematical theory. The study of encoding and decoding private messages is known as cryptography. Here we will introduce the study of cryptography and focus on linear algebra-based cipher.

Caesar cipher

One of the most straightforward and well-known encryption techniques is the Caesar cipher. This alternative and representational cipher substitutes a different letter from an alphabet that occupies a set number of positions in the alphabet for each letter in the plaintext of the secret message. Julius Caesar employed this technique in his private correspondence. This technique bears his name, Caesar code [18].

While exchanging information in this way, the person receiving this information, will not be able to decipher it. This special form is called encryption. The act of extracting the original information from the encrypted information is called decryption.

In Caesar's cipher system, a number is given along with a text called a key. For example, if the value of k is 2 then ABC will be written as CDE [3].

Plain text:

A	B	C	D	E	F	G	H	I	J	K	L	M	N

O	P	Q	R	S	T	U	V	W	X	Y	Z

Cipher text:

C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S

T	U	V	W	X	Y	Z	A	B

Using this cipher text "BARISHAL UNIVERSITY" will be "CBSJTIBM VOJWFSJUZ"

Hill Cipher:

The Hill Cipher is a linear algebra-based polygraphic substitution cipher. Where each letter is represented by a number modulo 26. Here to encrypt a message each block of n letters is multiplied by an invertible n by n matrix, against modulus 26. Each block is multiplied by the encryption matrix's inverse to decode the message. The matrix used for encryption is the cipher key and should be chosen randomly from the set of invertible n by n matrices (modulo 26).

A	B	C	D	E	F	G	H	I	J	K	L	M
	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

This computation used in Hill Cipher is based on linear algebra techniques. Before encryption and decryption, it is important to recognize that the above alphabet is a linear space.

(i) The alphabet has a zero element. Here the zero element is "*A*". The numerical value $x + A = x$.

(ii) The alphabet is closed under modulo addition. For two letters x and y, $x + y = z$, where z is the reminder from dividing the sum of x and y by the size of the alphabet.

(iii) The alphabet is closed under modulo scalar multiplication. For two letters x and y, $xy = z$, where z is the reminder of the product of x and y by the size of the alphabet.

Encryption with the Hill Cipher:

First, we know the alphabet is in linear space. We can perform a linear transformation on it. Encrypting text using the hill cipher is executed by breaking a plaintext into n blocks, where the blocks are column vectors, and multiplying these vectors by $n \times n$ matrix. Where the matrix is invertible. That means the determinant of the matrix cannot be zero. Determinants must be relatively prime with the size of the alphabet. The encryption matrix must be invertible because its inverse is used for the decryption matrix [3].

Example 2.4.1.

As an example, to encrypt the plaintext "BARISHAL UNIVERSITY" with $n = 3$. The process is as follows:

(1) Choose a 3×3 matrix. The key is "BEAUTYFUL" so,

$$k = \begin{bmatrix} \mathrm{B} & \mathrm{E} & \mathrm{A} \\ \mathrm{U} & \mathrm{T} & \mathrm{Y} \\ \mathrm{F} & \mathrm{U} & \mathrm{L} \end{bmatrix} = \begin{bmatrix} 1 & 4 & 0 \\ 20 & 19 & 24 \\ 5 & 20 & 11 \end{bmatrix}.$$

The determinant of $k = -671 \neq 0$ and which is relatively prime with 26.

(2) Split the plain text into the block of size 3 (ignoring space). If the plain text's length isn't evenly divisible by 3, add a predetermined character to the end of the string until the text's length is divisible by 3.

$$\begin{bmatrix} \mathrm{B} \\ \mathrm{A} \\ \mathrm{R} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 17 \end{bmatrix}, \begin{bmatrix} \mathrm{I} \\ \mathrm{S} \\ \mathrm{H} \end{bmatrix} = \begin{bmatrix} 8 \\ 18 \\ 7 \end{bmatrix}, \begin{bmatrix} \mathrm{A} \\ \mathrm{L} \\ \mathrm{U} \end{bmatrix} = \begin{bmatrix} 0 \\ 11 \\ 20 \end{bmatrix}, \begin{bmatrix} \mathrm{N} \\ \mathrm{I} \\ \mathrm{V} \end{bmatrix} = \begin{bmatrix} 13 \\ 8 \\ 21 \end{bmatrix}, \begin{bmatrix} \mathrm{E} \\ \mathrm{R} \\ \mathrm{S} \end{bmatrix} = \begin{bmatrix} 4 \\ 17 \\ 18 \end{bmatrix}, \begin{bmatrix} \mathrm{I} \\ \mathrm{T} \\ \mathrm{Y} \end{bmatrix} = \begin{bmatrix} 8 \\ 19 \\ 24 \end{bmatrix}.$$

(3) Apply the formula $c = kp$, where p is the plain text, c is the cipher text, and k is the key.

$$\begin{bmatrix} 1 & 4 & 0 \\ 20 & 19 & 24 \\ 5 & 20 & 11 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \\ 17 \end{bmatrix} = \begin{bmatrix} 1 \\ 428 \\ 192 \end{bmatrix} = \begin{bmatrix} 1 \\ 12 \\ 10 \end{bmatrix} \pmod{26},$$

$$\begin{bmatrix} 1 & 4 & 0 \\ 20 & 19 & 24 \\ 5 & 20 & 11 \end{bmatrix}\begin{bmatrix} 8 \\ 18 \\ 7 \end{bmatrix} = \begin{bmatrix} 80 \\ 670 \\ 477 \end{bmatrix} = \begin{bmatrix} 2 \\ 20 \\ 9 \end{bmatrix} \pmod{26},$$

$$\begin{bmatrix} 1 & 4 & 0 \\ 20 & 19 & 24 \\ 5 & 20 & 11 \end{bmatrix}\begin{bmatrix} 0 \\ 11 \\ 20 \end{bmatrix} = \begin{bmatrix} 44 \\ 689 \\ 440 \end{bmatrix} = \begin{bmatrix} 18 \\ 13 \\ 24 \end{bmatrix} \pmod{26},$$

$$\begin{bmatrix} 1 & 4 & 0 \\ 20 & 19 & 24 \\ 5 & 20 & 11 \end{bmatrix}\begin{bmatrix} 13 \\ 8 \\ 21 \end{bmatrix} = \begin{bmatrix} 45 \\ 916 \\ 456 \end{bmatrix} = \begin{bmatrix} 19 \\ 6 \\ 14 \end{bmatrix} \pmod{26},$$

$$\begin{bmatrix} 1 & 4 & 0 \\ 20 & 19 & 24 \\ 5 & 20 & 11 \end{bmatrix}\begin{bmatrix} 4 \\ 17 \\ 18 \end{bmatrix} = \begin{bmatrix} 72 \\ 835 \\ 558 \end{bmatrix} = \begin{bmatrix} 20 \\ 3 \\ 12 \end{bmatrix} \pmod{26},$$

$$\begin{bmatrix} 1 & 4 & 0 \\ 20 & 19 & 24 \\ 5 & 20 & 11 \end{bmatrix}\begin{bmatrix} 8 \\ 19 \\ 24 \end{bmatrix} = \begin{bmatrix} 84 \\ 1097 \\ 684 \end{bmatrix} = \begin{bmatrix} 6 \\ 5 \\ 8 \end{bmatrix} \pmod{26}.$$

(4) Convert each of the matrices to their alphabetical value:

$$\begin{bmatrix} 1 \\ 12 \\ 10 \end{bmatrix} = \begin{bmatrix} \mathrm{B} \\ \mathrm{M} \\ \mathrm{K} \end{bmatrix}, \begin{bmatrix} 2 \\ 20 \\ 9 \end{bmatrix} = \begin{bmatrix} \mathrm{C} \\ \mathrm{U} \\ \mathrm{J} \end{bmatrix}, \begin{bmatrix} 18 \\ 13 \\ 24 \end{bmatrix} = \begin{bmatrix} \mathrm{S} \\ \mathrm{N} \\ \mathrm{Y} \end{bmatrix}, \begin{bmatrix} 19 \\ 6 \\ 14 \end{bmatrix} = \begin{bmatrix} \mathrm{T} \\ \mathrm{G} \\ \mathrm{O} \end{bmatrix}, \begin{bmatrix} 20 \\ 3 \\ 12 \end{bmatrix} = \begin{bmatrix} \mathrm{U} \\ \mathrm{D} \\ \mathrm{M} \end{bmatrix}, \begin{bmatrix} 6 \\ 5 \\ 8 \end{bmatrix} = \begin{bmatrix} \mathrm{G} \\ \mathrm{F} \\ \mathrm{I} \end{bmatrix}.$$

Cipher text: BMKCUJSNYTGOUDMGFI

Decryption with the Hill Cipher:

Here we are interested in how a party receiving a secret message encoded by the Hill Cipher can decode it into the Plaintext:

(1) Find k^{-1},

$$d = \det\begin{bmatrix} 1 & 4 & 0 \\ 20 & 19 & 24 \\ 5 & 20 & 11 \end{bmatrix} = -671$$

$$-671 \pmod{26} = (26 \times 26 - 671) \pmod{26} = 5 \pmod{26}$$

$$5 \times d^{-1} \equiv 1 \pmod{26} = d^{-1} = 21$$

$$Adj\ k = \begin{bmatrix} -271 & -44 & 96 \\ -100 & 11 & -24 \\ 305 & 0 & -61 \end{bmatrix} = \begin{bmatrix} 15 & 8 & 96 \\ 4 & 11 & 2 \\ 305 & 0 & 17 \end{bmatrix}$$ (by removing negative sign)

$$= k^{-1} = 21\begin{bmatrix} 15 & 8 & 96 \\ 4 & 11 & 2 \\ 305 & 0 & 17 \end{bmatrix} = \begin{bmatrix} 315 & 168 & 2016 \\ 84 & 231 & 42 \\ 6405 & 0 & 357 \end{bmatrix}$$

$$\pmod{26} = \begin{bmatrix} 3 & 12 & 14 \\ 6 & 23 & 16 \\ 9 & 0 & 19 \end{bmatrix}.$$

(2) Using the formula $p = k^{-1}c$

$$\begin{bmatrix} 3 & 12 & 14 \\ 6 & 23 & 16 \\ 9 & 0 & 19 \end{bmatrix}\begin{bmatrix} 1 \\ 12 \\ 10 \end{bmatrix} = \begin{bmatrix} 287 \\ 442 \\ 199 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 17 \end{bmatrix} \pmod{26} = \begin{bmatrix} \mathrm{B} \\ \mathrm{A} \\ \mathrm{R} \end{bmatrix},$$

$$\begin{bmatrix} 3 & 12 & 14 \\ 6 & 23 & 16 \\ 9 & 0 & 19 \end{bmatrix}\begin{bmatrix} 2 \\ 20 \\ 9 \end{bmatrix} = \begin{bmatrix} 372 \\ 616 \\ 189 \end{bmatrix} = \begin{bmatrix} 8 \\ 18 \\ 7 \end{bmatrix} \pmod{26} = \begin{bmatrix} \mathrm{I} \\ \mathrm{S} \\ \mathrm{H} \end{bmatrix},$$

$$\begin{bmatrix}3 & 12 & 14\\ 6 & 23 & 16\\ 9 & 0 & 19\end{bmatrix}\begin{bmatrix}18\\ 13\\ 24\end{bmatrix}=\begin{bmatrix}546\\ 791\\ 618\end{bmatrix}=\begin{bmatrix}0\\ 11\\ 20\end{bmatrix}(\text{mod } 26)=\begin{bmatrix}\text{A}\\ \text{L}\\ \text{U}\end{bmatrix},$$

$$\begin{bmatrix}3 & 12 & 14\\ 6 & 23 & 16\\ 9 & 0 & 19\end{bmatrix}\begin{bmatrix}19\\ 6\\ 14\end{bmatrix}=\begin{bmatrix}325\\ 476\\ 437\end{bmatrix}=\begin{bmatrix}13\\ 8\\ 21\end{bmatrix}(\text{mod } 26)=\begin{bmatrix}\text{N}\\ \text{I}\\ \text{V}\end{bmatrix},$$

$$\begin{bmatrix}3 & 12 & 14\\ 6 & 23 & 16\\ 9 & 0 & 19\end{bmatrix}\begin{bmatrix}20\\ 3\\ 12\end{bmatrix}=\begin{bmatrix}264\\ 381\\ 408\end{bmatrix}=\begin{bmatrix}4\\ 17\\ 18\end{bmatrix}(\text{mod } 26)=\begin{bmatrix}\text{E}\\ \text{R}\\ \text{S}\end{bmatrix},$$

$$\begin{bmatrix}3 & 12 & 14\\ 6 & 23 & 16\\ 9 & 0 & 19\end{bmatrix}\begin{bmatrix}6\\ 5\\ 8\end{bmatrix}=\begin{bmatrix}190\\ 276\\ 206\end{bmatrix}=\begin{bmatrix}8\\ 19\\ 24\end{bmatrix}(\text{mod } 26)=\begin{bmatrix}\text{I}\\ \text{T}\\ \text{Y}\end{bmatrix}.$$

So, the original plaintext: BARISHALUNIVERSITY.

Where the ciphertext has been decrypted into the original plaintext.

Block Cipher:

Block ciphers use a cryptographic key and algorithm to encrypt data in blocks to create ciphertext. This requires an initialization vector that is added to the input plaintext to increase the key space of the cipher. Block ciphers only encrypt messages that are the same size as their block length [18]. Blocks are encrypted.

***Electronic Codeback Mode (ECB):** This mode is used to electronically code messages in plaintext form. It doesn't add any randomness to the key stream. Here for 8 bytes plaintext, we need only 8 bytes key [18].

***Cipher Block Chaining Mode (CBC):** This mode is a method of encrypting data that ensures that each block of plaintext is combined with the previous ciphertext block before being encrypted. Here each plain text block is XORed with the previous cipher text block before being encrypted with the cipher algorithm. A bitwise XOR iterates through binary strings, comparing each significant digit and returning 1 if there is an odd number of 1's between the two and 0 otherwise [18].

Example 2.4.2. We want to encrypt the plaintext p_1 = "BARISHALU" and ciphertext string c_0 = XYXSCYVTX. Then the integer form is p_1 = 2 1 18 9 19 8 1 12 21 and c_0 = 24 25 24 29 3 25 22 20 24.

In this cipher, the largest value we need to represent in binary digits is 26. Each binary digit represents powers of 2. That means $n < 2^x$, where $x = \log_2 x$. It is clear that only 5 bits to represent 1-26 in binary, determining each number representation is a simple greedy algorithm, which subtracts each power of 2 from the desired number. If the subtraction is greater than or equal to zero then the bit is 1, otherwise 0.

c_0	11000	11001	11000	10011	00011	11001	10110	10100	11000

XOR

p_1	00010	00001	10010	01001	10011	01000	00001	01100	10101

Then we get after calculation

p_1'	11010	11000	01010	11010	10000	10001	10111	11000	01101

So the decimal digits are, p_1' = 26 24 10 26 16 17 23 24 13.

Now take 3×4 plaintext matrix $c_1 = k.p_1 = \begin{bmatrix} 3 & 3 & 4 \\ 0 & 1 & 1 \\ 4 & 3 & 4 \end{bmatrix} \begin{bmatrix} 26 & 24 & 10 \\ 26 & 16 & 17 \\ 23 & 24 & 13 \end{bmatrix} =$

$\begin{bmatrix} 248 & 216 & 133 \\ 49 & 40 & 30 \\ 274 & 240 & 143 \end{bmatrix}$. An inverse function is used to decrypt the text. The recipients first apply the inverse matrix key to the cipher text c_1 to get p_1 back.

$$p_1' = k^{-1}.\, c_1 = \begin{bmatrix} -1 & 0 & 1 \\ -4 & 4 & 3 \\ 4 & -3 & -3 \end{bmatrix} \begin{bmatrix} 248 & 216 & 133 \\ 49 & 40 & 30 \\ 274 & 240 & 143 \end{bmatrix} = \begin{bmatrix} 26 & 24 & 10 \\ 26 & 16 & 17 \\ 23 & 24 & 13 \end{bmatrix}.$$

We then perform a bitwise XOR on c_0 and p_1'

c_0	11000	11001	11000	10011	00011	11001	10110	10100	11000

XOR

p_1'	11010	11000	01010	11010	10000	10001	10111	11000	01101

and get back

p_1	00010	00001	10010	01001	10011	01000	00001	01100	10101

which is, 2 1 18 9 19 8 1 12 21.

If two people want to communicate securely then this is easy to do. This technique ensures the security of to transfer of private data. It has four major goals: confidentiality, integrity, authentication, and non-repudiation. Collectively these benefits allow companies to conduct business in the digital era with complete confidence.

3. Conclusion

In this paper, we have demonstrated the importance of linear algebra in the real world by introducing some well-known components by exhibiting some major applications. We expressed how linear algebra can be used in Image Processing, Cryptography, and also some other parts of sciences and showed how they affect our lives. The advantage of our representation is finding simple solutions to complex problems through linear algebra. The future motive of this project is to solve abstruse problems by combining all of the above with more novelties and to promote the use of linear algebra in science and our living.

References

[1] B. Andrásfai, Graph Theory: Flows, Matrices, CRC Press, 1991.

[2] R. A. Barnett and M. R. Ziegler, Linear Algebra: An Introduction with Applications, 1987.

[3] R. Doyle, Hill's Cipher: Linear Algebra in Cryptography. Consulta: marzo-2017.

[4] S. A. Ferrovial, Ferrovial, Transport Company, 2019.

[5] G. N. Hartman, Fundamentals of Matrix Algebra, APEX Calculus, 2011.

[6] C. Hsu and J. Wang, Applied linear algebra methods for data science, In Proceedings of the 2nd International Conference on Computing and Big Data, (2019), 17-21.

DOI: https://doi.org/10.1145/3366650.3366668

[7] O. G. Hurtado, W. G. Thiriata and W. P. Casallas, Some applications of linear algebra to genetics, Journal of Language and Linguistic Studies 18(4) (2022), 1143-1149.

[8] J. Jauregui, Principal Component Analysis with Linear Algebra, Philadelphia: Penn Arts & Sciences, 2012.

[9] J. Kirkham, Math 308 Project Autumn, 2001.

[10] N. M. Kumari, H. U. Kavya and Y. N Krupashree, Application of linear algebra in genetics, Journal of Harmonized Research in Applied Science 6(4) (2018), 236-240.

DOI: https://doi.org/10.30876/JOHR.6.4.2018.236-240

[11] D. C. Lay, Linear Algebra and its Applications, Pearson Education India, 2003.

[12] J. Ludwig, Image Convolution, Portland State University, 2013.

[13] J. Machado, Linear Algebra Application: Google page rank Algorithm, University of North Carolina at Greensboro.

[14] A. Moshrefi, S. Rogers, M. Wolf and J. Zambrano, Applications of Linear Algebra: Genetics, California State, Long Beach, (2012), 20.

[15] R. Preetha, Matrices in Data Science, Bannari Amman Institute of Technology, 2023.

[16] K. H. Rosen, Discrete Mathematics and its Applications, The McGraw Hill, 2007.

■