**LINEAR ALGEBRA AND ITS APPLICATION**

DEPARTMENT: COMPUTER SCIENCE

|  |  |  |
| --- | --- | --- |
|  | BAKARE ABDULLAHI | 180805015 |
|  | OLORUNFEMI-OJO DANIEL | 190805503 |
|  | ADEWUMI OLORUNJUWON | 180805080 |
|  | OLUWATOSIN ELIJAH | 180805007 |
|  | JACOB KOREDE | 180805010 |
|  | ANIMASHAUN TOBILOBA | 180805032 |
|  | AKINOLA ISAAC | 180805002 |
|  | BALOGUN OLASUNKANMI | 180805079 |
|  | IDOWU-KOYA IREWOLEDE | 180805064 |
|  | ADEKOGBE OLUWADAMILARE | 180805054 |
|  | ALADE MICHAEL | 180805001 |
|  | KADIRI MICHAEL | 180805036 |
|  | KASSIM STEPHEN | 180805051 |
|  | OMOLOJA ISHAQ | 180805028 |
|  | OYELEYE JOSHUA | 180805078 |
|  | MAKINDE OPEYEMI | 180805021 |
|  | ODULAJA OLUWADAMILARE | 180805052 |
|  | HASSAN AYOMIDE | 190805505 |
|  | WONAH GODWIN | 180805030 |
|  | SODIQ BAKARE | 180805027 |

Linear Algebra is the branch of mathematics concerning vector spaces and linear mapping between such spaces. It includes the study of lines, planes and subspaces but is also concerned with the properties common to all vector spaces. Linear Algebra is used in most sciences and fields of engineering because it allows modelling many natural phenomena and computing efficiently with such models. The topics of Linear Algebra and its implementation include:

**VECTORS & VECTOR SPACES**

A vector is an n-tuple of real numbers. For example, consider a vector that has three components:

The operations we can perform on vectors and include:

* Vector addition:
* Vector subtraction:
* Scalar multiplication:
* Norm(length):
* Dot product:

A **Vector Space** is a nonempty set V of objects, called vectors, on which are defined two operations, called addition and multiplication by scalars. The following axioms must hold for all and in the vector space and all scalars and .

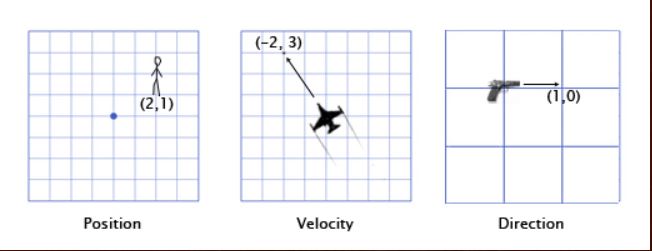
1. The sum of u and v denoted by , is in
2. (
3. There is a zero vector in such that
4. For each in , there is a scalar in such that
5. The scalar multiple of by , denoted by , is in

Given vectors and weights we can define the linear combination by:

If are in , then the set of all linear combinations of is denoted by Span and is called the subset of spanned by . So, Span is the collection of all vectors that can be written in the form with scalars.

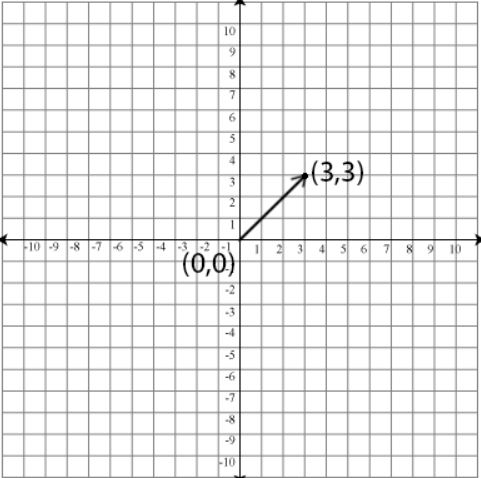
An indexed set of vectors in is said to be linearly independent if the vector equation has only the trivial solution. The set is said to be linearly dependent if there exist weights , not all zero, such that . This equation is called a linear dependence relation among .

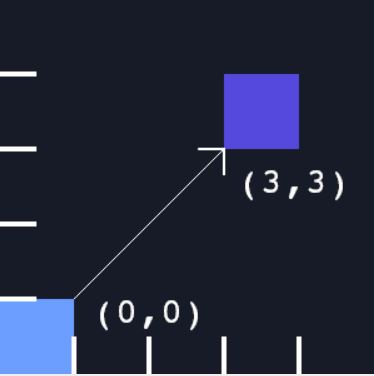
In game development, vectors are used to store positions, directions, and velocities.



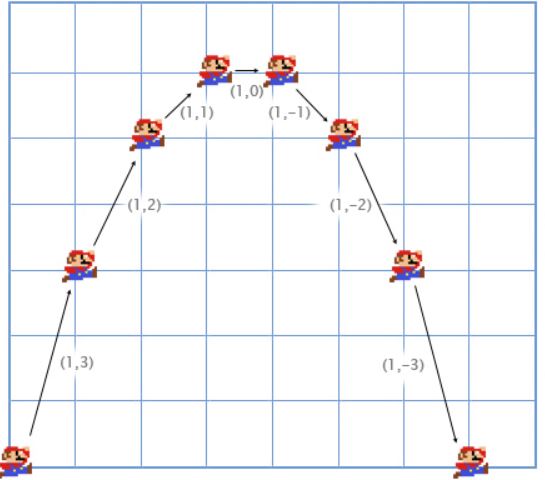
The position vector indicates that the man is standing two meters east of the origin, and one meter north. The velocity vector shows that in one minute, the plane moves three kilometers up, and two to the left. The direction vector tells us that the pistol is pointing to the right.

Say I am creating a 2D game, then a positional vector (0, 0) might indicate that a given object is at the center of the game world. Now, let’s say that I want to manipulate the vector by adding to its position. Adding (3, 3) yields



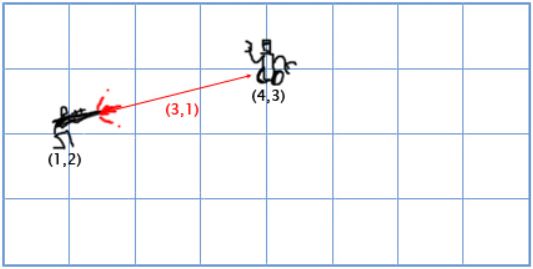


Addition and subtraction operations integrates physics into a game. Let's consider the example of Mario jumping. He starts at position (0,0). As he starts the jump, his velocity is (1,3). He is moving upwards quickly, but also to the right. His acceleration throughout is (0,-1), because gravity is pulling him downwards. Here is what his jump looks like over the course of seven more frames.



For the first frame, we add his velocity (1,3) to his position (0,0) to get his new position (1,3). Then, we add his acceleration (0,-1) to his velocity (1,3) to get his new velocity (1,2). We do it again for the second frame. We add his velocity (1,2) to his position (1,3) to get (2,5). Then, we add his acceleration (0,-1) to his velocity (1,2) to get (1,1).

Vector subtraction is useful for getting a vector that points from one position to another. For example, let's say the player is standing at (1,2) with a laser rifle, and an enemy robot is at (4,3). To get the vector that the laser must travel to hit the robot, you can subtract the player's position from the robot's position. This gives us:



**MATRIX**

A matrix is a rectangular array of real numbers with m rows and n columns. For example, a matrix looks like

Some matrix operations include:

* Addition:
* Subtraction:
* Product: The product of matrices and is another matrix given by the formula

=

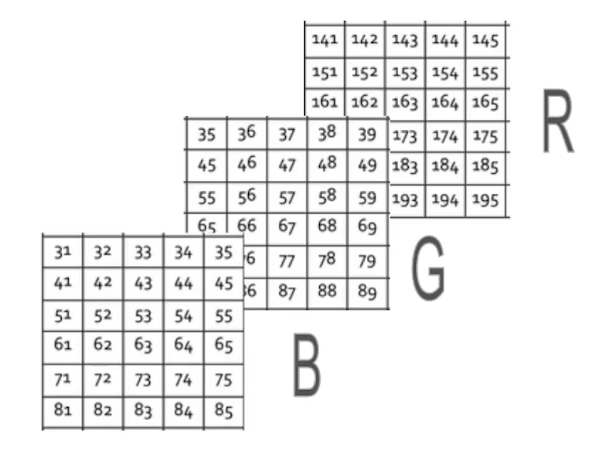
Note that matrix product is not commutative operation:

* Scalar product:
* Transpose:
* Trace:

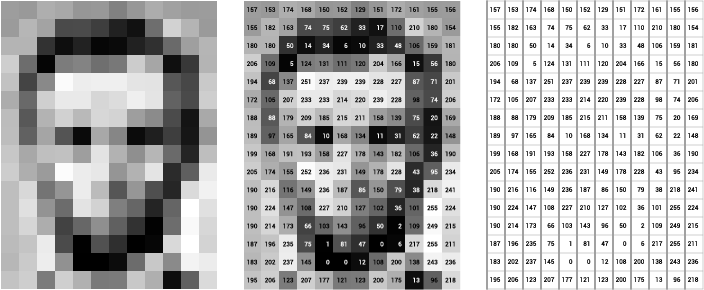
Matrices can be applied in data structures to represent information. This  
is sometimes not obvious as data can be complicated. Still, we usually get away with  
matrices. For example, if we have names linked to addresses and telephone numbers,  
we can make a spread sheet *A(i,j),* where *A(i,1)* is the name and *A(i,2)* is the  
address and *A(i,3)* is the telephone number. In a relational database, information  
is stored in tables as matrices containing records as rows and attributes as columns.

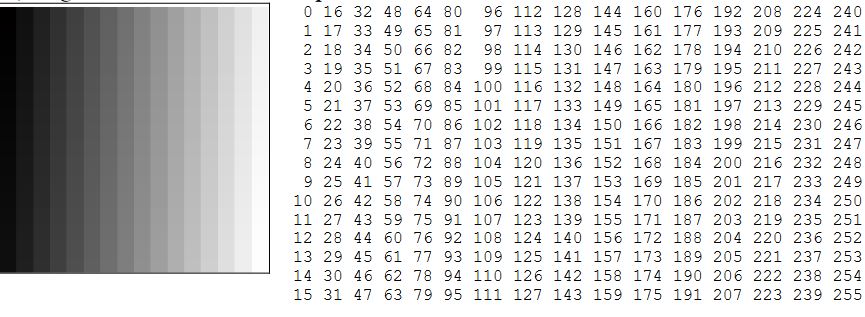
Multimedia like images, sound or movies are stored in matrix form. An image is an array of pixels, a sound is an array of amplitudes, a movie is given by an array of pictures and a sound. An image is a matrix where each entry *A(i, j)* contains three numbers *(r, g, b),* where r, g, b are red, green and blue color values. The elements of these matrices are integer numbers between 0 and 255, and they determine the intensity of the pixel with respect to the color of the matrix. Thus, in the RGB system, it is possible to represent 256 3 = 2 24 = 16777216 different colors.



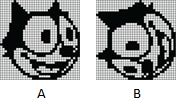


Grayscale images can also be represented by matrices. Each element of the matrix determines the intensity of the corresponding pixel. For convenience, most of the current digital files use integer numbers between 0 (to indicate black, the color of minimal intensity) and 255 (to indicate white, maximum intensity), giving a total of 256 = 2 8 different levels of gray.



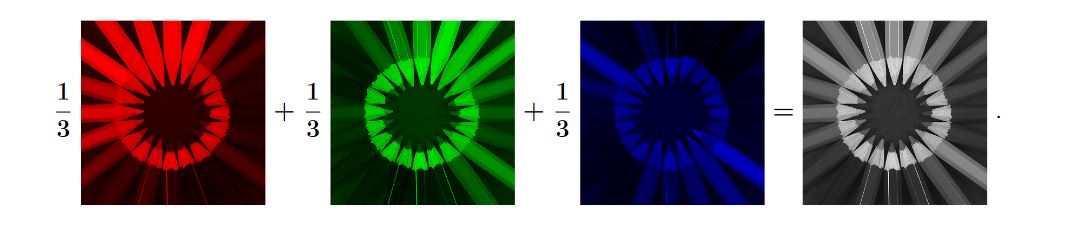


Since an image can be represented by matrices, we can perform some matrix operations on their elements affect the corresponding image. For example, if we consider the binary image A below as a matrix, say *A = ( ai,j ),* then the image *B* corresponds to the transposed matrix of *A*, that is, *B* = *( bi,j ) = ( aj,i) = AT .*



Finding the transpose reverses the matrix therefore, flipping the image.

Using matrix addition and scalar multiplication, if we take the standard arithmetic mean of the component matrices R, G and B from a color image A, we will get a grayscale version of the image (non-integer values are rounded to the nearest integer).



Another example is using the operation of multiplying by a scalar and sum of matrices, it is possible to create an image transition effect commonly used, for instance, in Power Point presentations and slide shows. More precisely, consider two grayscale images of the same size, represented by the matrices A and Z. For each scalar (real number) t in the interval [0, 1], define the matrix

Notice that M(0) = A, M(1) = Z and, for each t between 0 and 1, the elements of the matrix M(t) are between the elements of the matrices A and Z. Therefore, when t varies from 0 to 1, the matrix M(t) varies from A to Z. For the case of color images, the transformation above must be applied to the matrices R, G and B that compose each image.



**DETERMINANT**

The determinant of a matrix, denoted det(A) or |A|, is a special way to combine the entries of a matrix that serves to check if a matrix is invertible or not. The determinant formulas for 2 × 2 and 3 × 3 matrices are

and

The determinant of matrix provides a way to find the inverse of that matrix. Let ***A*** be an matrix. Then,

For example, find in the inverse of the matrix,

To get , find the transpose of the cofactor matrix of A.

Therefore, we can find the inverse of a matrix by multiplying the determinant of the matrix with the transpose of the cofactor of the matrix.

Determinant is also applied in solving a system of linear equations in the from ***AX = B*** where the solutions are given by ***X.* Cramer’s Rule** can be used to solve this system. Cramer’s rule gives a formula for the solutions of ***X*** in the special case that ***A*** is a square matrix. The rule does not apply if the system of equations has a different number of equations than variables. In other words, ***A*** is not a square matrix.

Hence, the solutions ***X*** to the system are given by ***X* = *A-1B***. Since we assume that ***A*−1** exists, we can use the formula for ***A*−1** given above. Substituting this formula into the equation for ***X***, we have

***A*** is an matrix. To solve the system ***AX = B*** for , Cramer’s rule says

where **Ai** is the matrix obtained by replacing the **ith** column of **A** with the column matrix

For example, find x, y & z if

To find x,

Therefore,

Similarly, to find *y*, we construct ***A2*** by replacing the second column of ***A*** with ***B***

Therefore,

By using determinants in Cramer’s rule, we find the solutions to the system of equation are

**LINEAR MAPPING**

A linear transformation is a map satisfying two properties

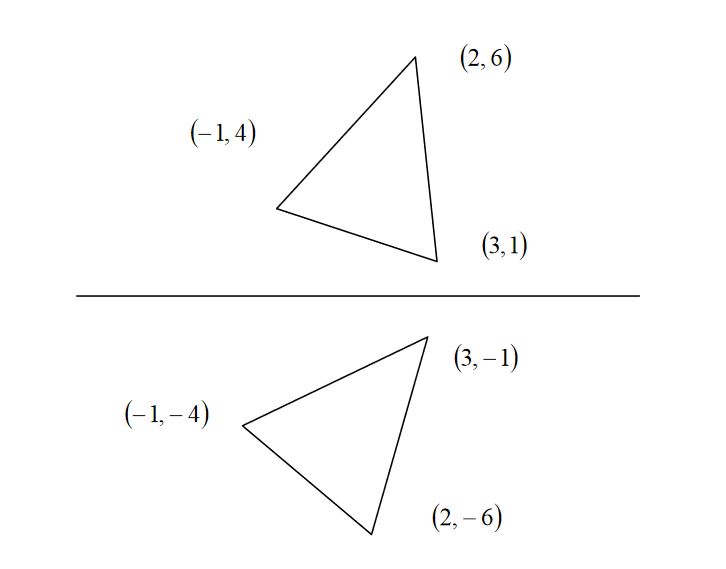
1. for all
2. for all and

The map defined by defines a Linear Transformation from . The Kernel and Image of the transformation are:

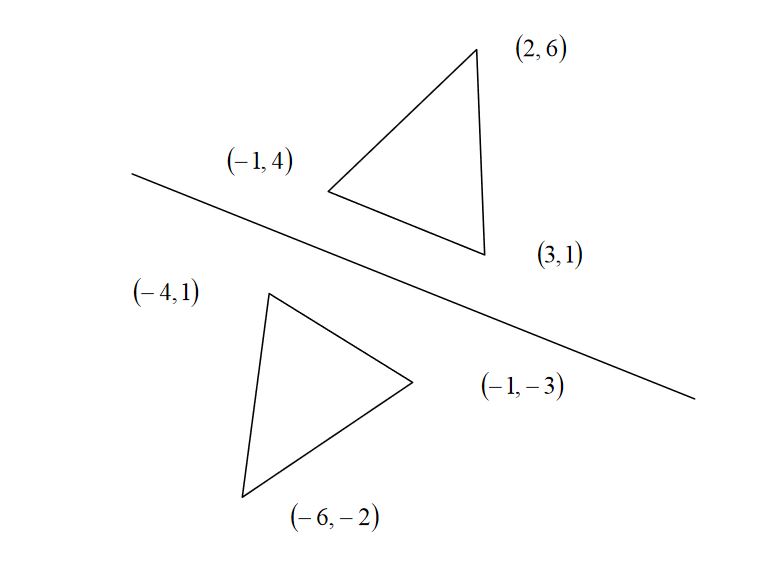
Linear mapping is applied in Computer Graphics. In Computer Graphics, matrices are used to represent many different types of data. Games that involve 2D or 3D graphics may rely on some matrix operations to display the game environment and characters in the game. Linear mapping (transformation) is used to reposition the graphics on the screen and change their size or orientation. We can have various types of transformation such as translation, scaling up or down, rotation, shearing, etc.

**Reflection:** is the mirror image of an original object. In other words, we can say that it is a rotation operation with 180 degrees. In reflection transformation, the size of the object does not change. For example, the reflection for the triangle with vertices (-1, 4), (3, 1), (2, 6) with respect to the x axis is

The plot is given below,



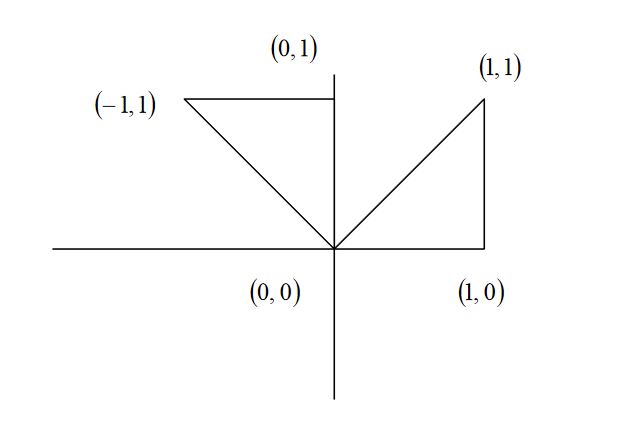
For reflection with respect to y = -x,



**Rotation:** In rotation, we rotate the object at angle θ

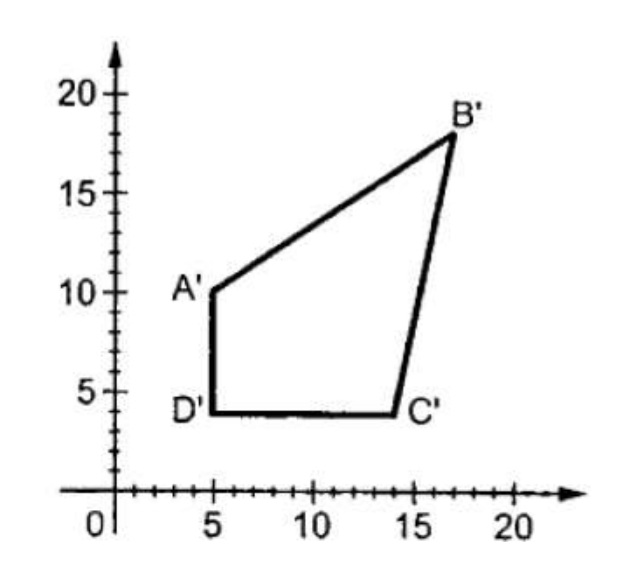
For example, as θ =

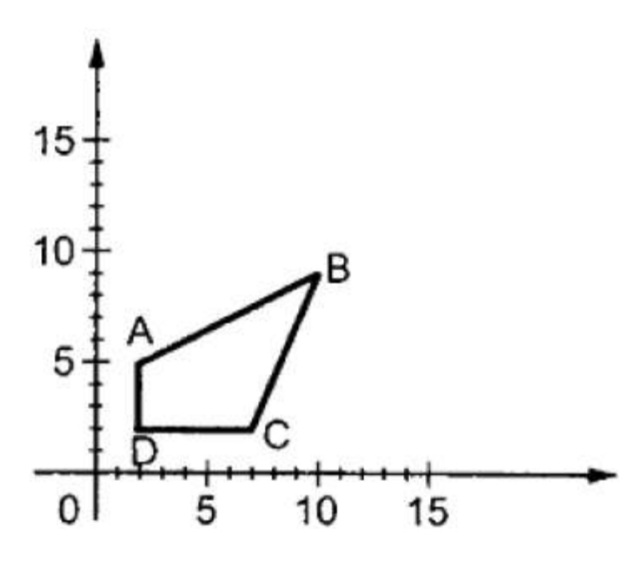
Thus, the rotation for the triangle with vertices (0, 0), (1, 0), (1, 1) is



**Scaling:** To change the size of an object, scaling transformation is used. In the scaling process, you either expand or compress the dimensions of the object. Scaling can be achieved by multiplying the original coordinates of the object with the scaling factor to get the desired result. Let us assume that the original coordinates are X, Y, their scaling factors are (Sx, Sy) and the produced coordinates are X`, Y`. This can be mathematically shown as

The scaling factor Sx, Sy scales the object in X and Y direction respectively. The above equations can also be represented in matrix form.





If we provide values less than 1 to the scaling factor S, then we can reduce the size of the object. If we provide values greater than 1, then we can increase the size of the object.

Linear mapping can also be employed in cryptography. Linear transformation can be used to encrypt and decrypt messages.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | B | … | X | Y | Z |
| 1 | 2 | … | 24 | 25 | 26 |

For example,

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| M | E | E | T | T | O | M | O | R | R | O | W |
| 13 | 5 | 5 | 20 | 20 | 15 | 13 | 15 | 18 | 18 | 15 | 23 |

The sequence: **13 5 5 20 20 15 13 15 18 18 15** **23** is the original code message. To encrypt the original code message, we can apply a transformation to the original code message. Let

where

Then, we break the original message into 4 vectors first.

and use the linear transformation to obtain the encrypted message code

Therefore, we can send the encrypted message code,

**38 28 15 105 70 50 97 64 51 117 79 61**

Suppose our friend wants to decrypt the encrypted message code. Our friend can find the inverse matrix of A first,

Then,

Thus, our friend can find the original message code

**13 5 5 20 20 15 13 15 18 18 15 23**

via the inverse of the matrix A.