Adamson University
College of Engineering
Computer Engineering Department

Linear Algebra

Laboratory Activity No. 5

Multi-dimensional Vectors

Submitted by:

Surio, Rovil Jr., M.

Instructor:

Engr. Dylan Josh D. Lopez

November 2, 2020

I. Objectives

This laboratory activity aims to implement the principles and techniques of representing multi-dimensional vectors or linear combinations in a 3-dimensional plane.

II. Methods

The practices of the activity include creating a system of a linear equation and representing it on a 3-dimensional plane. This teaches and implies the techniques for representing vectors using a different method of plotting in a 2D or 3D plane. The deliverables of the activity are to provide a system of a linear equation and scalar values for plotting on 2D and 3D planes. Moreover, it was achieved by creating an array of vectors and creating random values for scalars by using np.range() function that "return evenly spaced values within a given interval" [1]. The mathplotlib.pyplot.gca() function was used to get the current axes and project them in 3D [2]. For the creation of a figure object, the plt.figure() function was used [3] while plt.plot() was used to plot the vectors [4] and the plt.show() was used to show the plotted vectors on the plane [5].

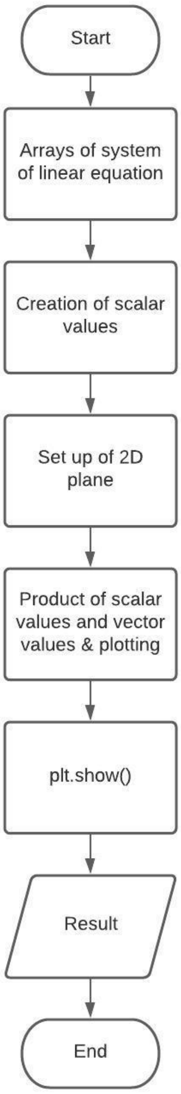

Start
Arrays of system of linear equation
Creation of scalar values
Set up of 2D plane
Product of scalar values and vector values & plotting
plt.show()
Result
End

Figure 1 The flowchart of task 1

Figure 1 shows the flow of the block of code created for plotting the system of a linear equation in a 2D plane.

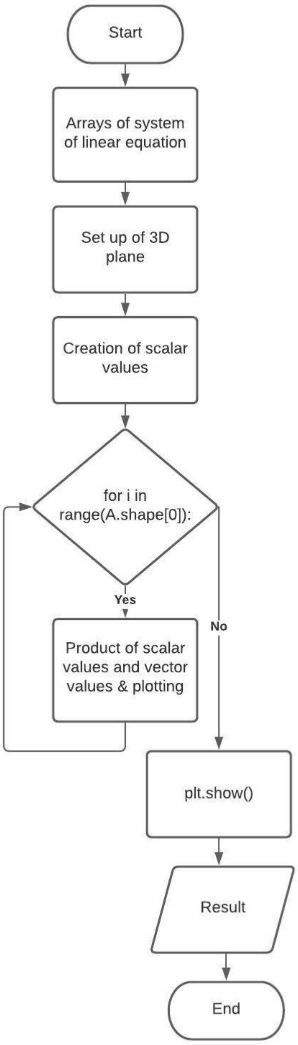
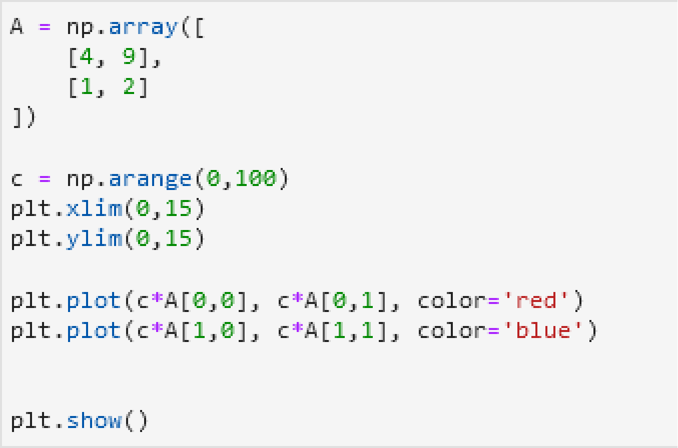

Start
Arrays of system of linear equation
Set up of 3D plane
Creation of scalar values
for i in range(A.shape[0]):
Yes
No
Product of scalar values and vector values & plotting
plt.show()
Result
End

Figure 2 The flowchart of task 2

Figure 2 shows the flow of the block of code created for plotting the system of a linear equation in a 3D plane.

III. Results

The results from the two tasks given to be familiar in representing multi-dimensional vectors and plotting system of linear equation in 2D and 3D-dimensional planes are presented in this section.



```
A = np.array([
    [4, 9],
    [1, 2]
])

c = np.arange(0,100)
plt.xlim(0,15)
plt.ylim(0,15)

plt.plot(c*A[0,0], c*A[0,1], color='red')
plt.plot(c*A[1,0], c*A[1,1], color='blue')


plt.show()
```

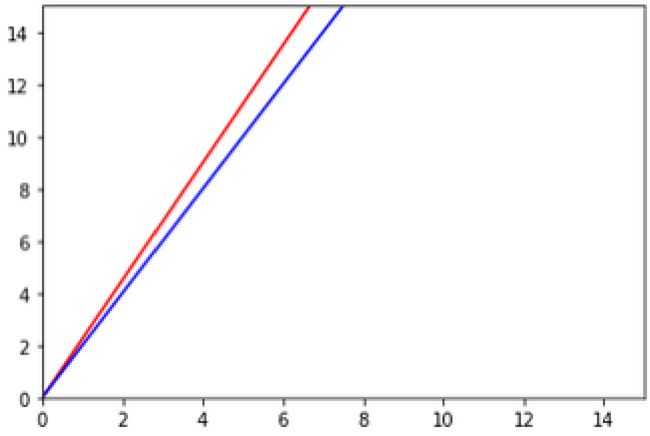
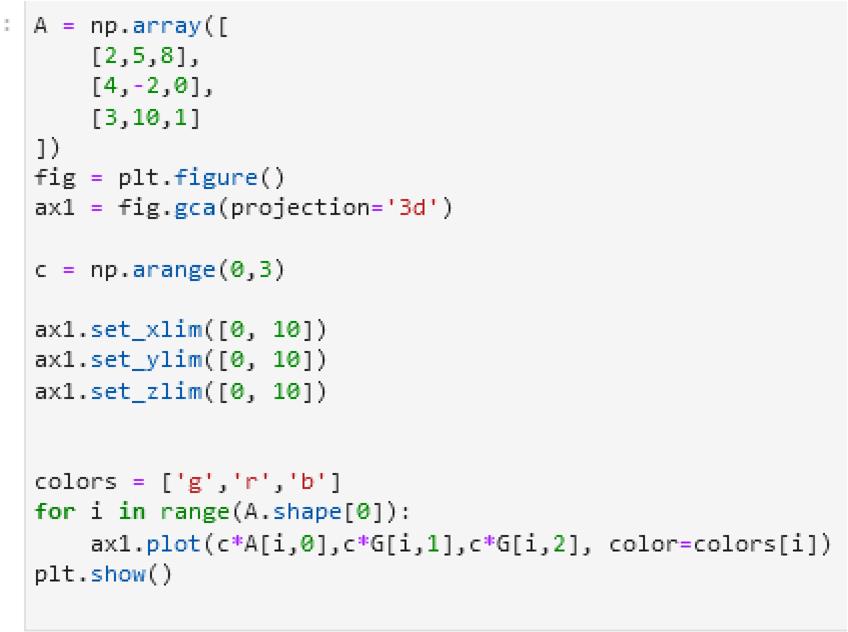


Figure 3 Task 1 code and output

Figure 3 shows the block of code created and its vector visualization output plotted in a 2D plane. The result represents the continuous line of the system of linear equations in the 2D plane. Each vector values were multiplied to the scalar values resulting in its scaled version or the continuous line shown in Figure 3.



```
A = np.array([
    [2,5,8],
    [4,-2,0],
    [3,10,1]
])
fig = plt.figure()
ax1 = fig.gca(projection='3d')

c = np.arange(0,3)

ax1.set_xlim([0, 10])
ax1.set_ylim([0, 10])
ax1.set_zlim([0, 10])


colors = ['g','r','b']
for i in range(A.shape[0]):
    ax1.plot(c*A[i,0],c*G[i,1],c*G[i,2], color=colors[i])
plt.show()
```

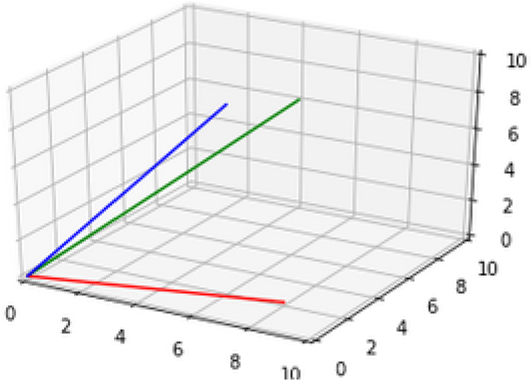


Figure 4 Task 2 code and output

Figure 4 shows the block of code created to plot the 3-dimensional vector on a 3D plane. Two objects were created which is the figure and the axis using the plt.figure() and the plt.gca() function. The projection of this was specified in 3D and the limit for x, y, and z was set afterward. Lastly, a loop statement was used in plotting the values. For the number of rows, it's going to plot with its corresponding column values which are also scaled.

According to [6], every data set has a general structure and is always characterized by the group of variables or the dimensions. It was stated in [6] that the variable in one-dimensional is usually time, while in two-dimensional, it can be data usually found in statistics like the number of financial transactions in a specific period and for the three-dimensional, it's just an extension of 2D therefore, an additional dimension can be added like for example in the financial transactions in a specific period, another dimension can be added like the customer type in observing the data.

Data can have more than three-dimension according to [7], for this particular example, it shows a relational database about the sales volumes for a printing company.

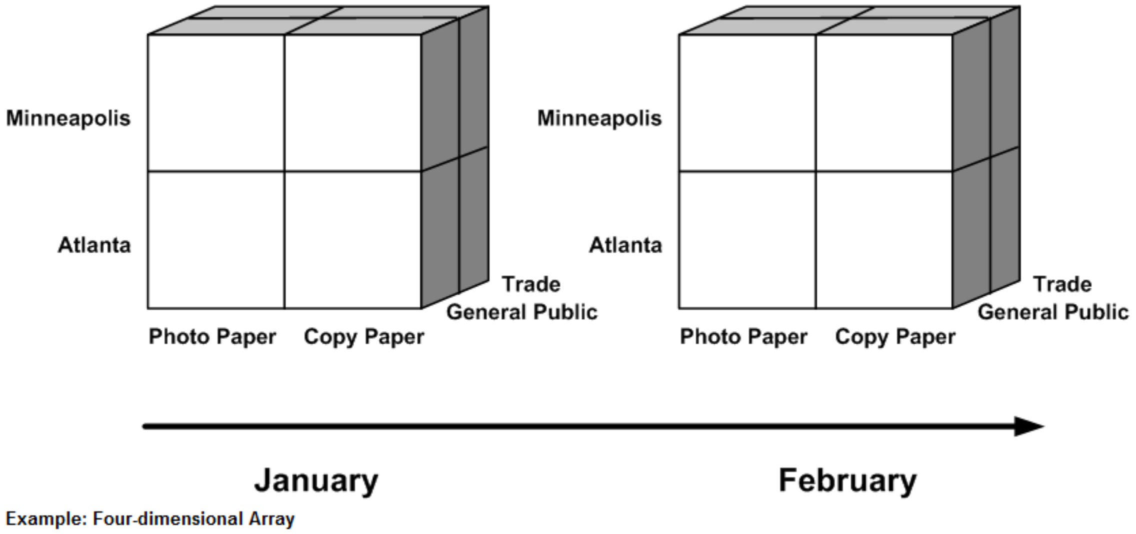


Figure 5 Four-dimensional data

Figure 5 shows the example of data with four-dimensional from [7], the following dimensions are store, item, customer type, and time.

IV. Conclusion

This laboratory implements the principles and techniques of representing multi-dimensional vectors. Through discussion and activities, I have familiarized myself with plotting a 3-dimensional vector and a system of linear equations in 2D and a 3D plane. Moreover, by answering the discussion question, I was introduced to the real-world application of multi-dimensional vectors and this learning is what I achieved the most because it's helpful information to completely understand why I am doing this. I am looking forward to the next application of multi-dimensional vectors since what I have learned so far from this laboratory is only the techniques for visualizing the 3-dimensional vectors in a 3D plane. Lastly, the concepts of visualizing and representing vectors in n-dimensions for business applications can be applied in compiling records to be observed for a certain period like for example, the number of items sold in a store during week 1 and from what type of transaction was done [7].

References

[1]"numpy.arange — NumPy v1.19 Manual", *Numpy.org*, 2020. [Online]. Available: https://numpy.org/doc/stable/reference/generated/numpy.arange.html. [Accessed: 01- Nov- 2020].

[2]"matplotlib.pyplot.gca — Matplotlib 3.3.2 documentation", *Matplotlib.org*, 2020. [Online]. Available: https://matplotlib.org/3.3.2/api/_as_gen/matplotlib.pyplot.gca.html. [Accessed: 01- Nov- 2020].

[3]"matplotlib.pyplot.figure — Matplotlib 3.3.2 documentation", *Matplotlib.org*, 2020. [Online]. Available: https://matplotlib.org/3.3.2/api/_as_gen/matplotlib.pyplot.figure.html. [Accessed: 01- Nov- 2020].

[4]"matplotlib.pyplot.plot — Matplotlib 3.3.2 documentation", *Matplotlib.org*, 2020. [Online]. Available: https://matplotlib.org/3.3.2/api/_as_gen/matplotlib.pyplot.plot.html. [Accessed: 02- Nov- 2020].

[5]"matplotlib.pyplot.show — Matplotlib 3.3.2 documentation", *Matplotlib.org*, 2020. [Online]. Available: https://matplotlib.org/3.3.2/api/_as_gen/matplotlib.pyplot.show.html. [Accessed: 02- Nov- 2020].

[6]*Citeseerx.ist.psu.edu*, 2020. [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.83.3421&rep=rep1&type=pdf. [Accessed: 02- Nov- 2020].

[7]"Multidimensional Databases", *Docs.oracle.com*, 2020. [Online]. Available: https://docs.oracle.com/cd/E12478_01/rpas/pdf/150/html/classic_client_user_guide/basic_rpas_concepts/multidimensional_databases.htm. [Accessed: 02- Nov- 2020].

Appendix

This is the GitHub Repository https://github.com/RovilSurioJr/Laboratory-5