Adamson University
College of Engineering
Computer Engineering Department

Linear Algebra

Laboratory Activity No. 5

Multidimensional Vectors

Submitted by:
Chipongian, John Patrick Ryan J

Instructor:
Engr. Dylan Josh D. Lopez

.

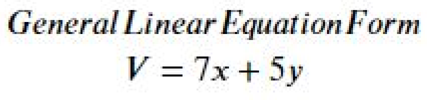
November 6, 2020

I. Objectives

This laboratory activity aims to implement the principles and techniques of representing linear combinations in 3-dimensional plane and be able to visualize vectors in three dimensions.

II. Methods

The practices that were used in completing this activity are numpy and matplotlib libraries. The function that were used from numpy is np.array() while the functions that were used in matplotlib library are plt.xlim(), plt.ylim(), plt.quiver()/ax1.quiver(), plt.show, plt.figure(), ax1.set_xlim(), ax1.set_ylim(), ax1.set_zlim(). Another function was used in completing the exercises given, which is the reshape() function which is a function of numpy. All of these functions have their own function in implementing the task given.



General Linear Equation Form

$$V = 7x + 5y$$

Figure 1

The first part of activity was to implement a ℝ=1 vector and to visualize it in a 2-dimensional plane. The vector that would be visualized has a general linear equation of V=7x + 5y.

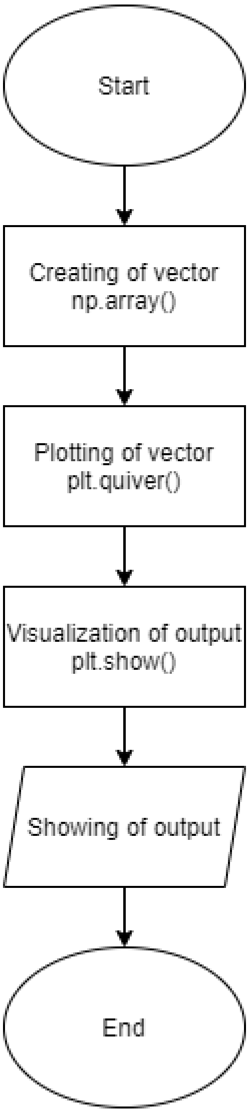


Figure 2

General Linear Equation Form

$$Z = \begin{cases} 5x + 8y + 4z \\ 7x + 2y + 9z \end{cases}$$

Figure 3

The second part of the activity was to implement a ℝ=2 vector vector and to visualize it in a 3-dimensional plane. The vector that would be visualized has a general linear equation of Z = 5x + 8y + 4z ; 7x + 2y + 9z.

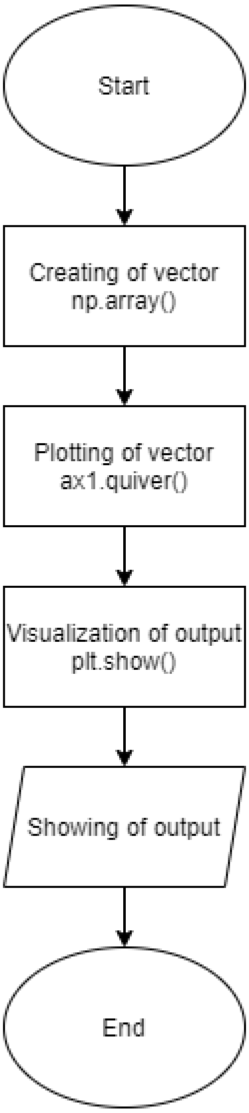
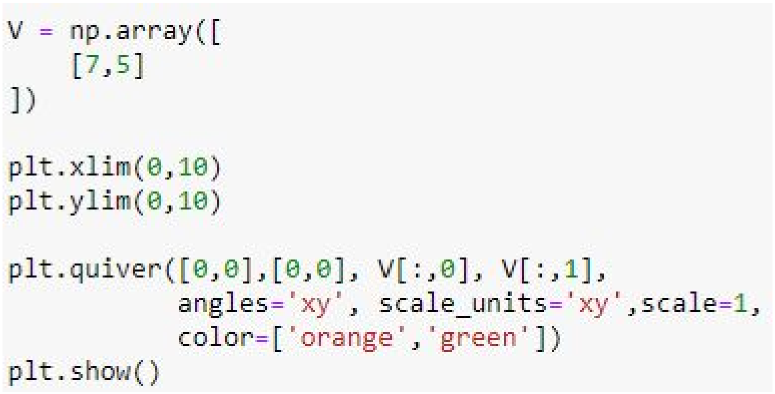


Figure 4

III. Results



```
V = np.array([
    [7,5]
])

plt.xlim(0,10)
plt.ylim(0,10)

plt.quiver([0,0],[0,0], V[:,0], V[:,1],
           angles='xy', scale_units='xy',scale=1,
           color=['orange','green'])
plt.show()
```

Figure 5

Figure 5 is the codes that implements the task given in task 1 of the activity. Firstly, the vector is created using np.array(). Functions [1] plt.xlim() and [2] plt.ylim() sets the limits of x and y axes. Function [3] plt.quiver() plots the arrow in the graph, it dictates its origin, endpoint, color and size. Lastly, [4] plt.show() displays all of the figures.

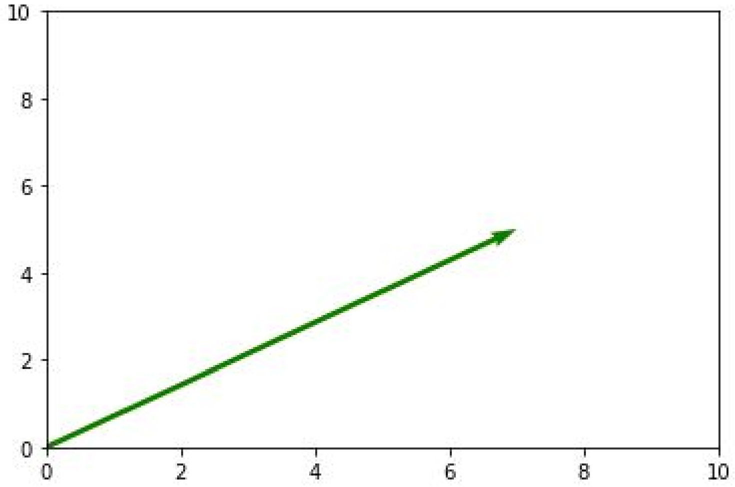
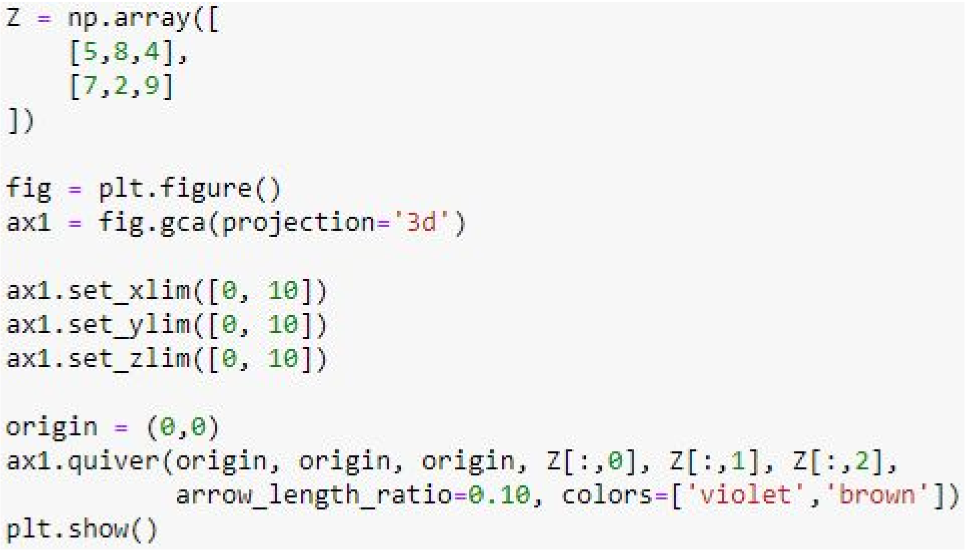


Figure 6

Figure 6 shows the visualization of the vector V= 7x + 5y in the task 1 of the lab activity.



```
Z = np.array([
    [5,8,4],
    [7,2,9]
])

fig = plt.figure()
ax1 = fig.gca(projection='3d')

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

origin = (0,0)
ax1.quiver(origin, origin, origin, Z[:,0], Z[:,1], Z[:,2],
           arrow_length_ratio=0.10, colors=['violet','brown'])
plt.show()
```

Figure 7

Figure 7 shows the code used to implement the second task of the activity. A vector of ℝ=2 is firstly created using np.array(). Function [5] plt.figure() then creates a new figure where the vector would be plotted. [6] As the figure would be plotted in 3-dimensions, fig.gca() sets the figure into 3-dimension. Functions [7] set_xlim(), set_ylim(), and set_zlim() sets the limit for x, y and z axes. Then, [7] quiver() is used to plot the vector in 3-dimensions. And lastly, [4] plt.show() shows all of the created figures.

[8] Many more data types can be plotted in two dimensional and three dimensional planes, such as different types of graphs and charts. An example of a graph is waveform

graph which only plots single-valued functions with points evenly distributed along the x-axis.

It is possible for a data to have more than three dimensions, however, [9] even after using different functions that are applicable for two dimensions or three dimensions, the visualization of it is still not possible but there are still different methods that can visualize a part of that data.

IV. Conclusion

In conclusion, we would be able to visualize vectors of different ℝ into two dimensional and three dimensional planes. And by using the function reshape() we would be able to change the rank of dimension of a vector.

Visualization and representation of vectors to any number of dimensions can be applied into business. An example of application of this is making different graphs and charts in presenting a relation of something so that you can present it in a more easier way.

References

[1] GeeksforGeeks. 2020. *Matplotlib.Pyplot.Xlim() In Python - Geeksforgeeks*. [online] Available at: <https://www.geeksforgeeks.org/matplotlib-pyplot-xlim-in-python/#:~:text=The%20xlim()%20function%20in,limits%20of%20the%20current%20axes.&text=Parameters%3A%20This%20method%20accept%20the,set%20the%20xlim%20to%20right.> [Accessed 6 November 2020].

[2] GeeksforGeeks. 2020. *Matplotlib.Pyplot.Ylim() In Python - Geeksforgeeks*. [online] Available at: <https://www.geeksforgeeks.org/matplotlib-pyplot-ylim-in-python/> [Accessed 6 November 2020].

[3] Matplotlib.org. 2020. *Matplotlib.Pyplot.Quiver — Matplotlib 3.3.2 Documentation*. [online] Available at: <https://matplotlib.org/3.3.2/api/_as_gen/matplotlib.pyplot.quiver.html> [Accessed 6 November 2020].

[4] GeeksforGeeks. 2020. *Matplotlib.Pyplot.Show() In Python - Geeksforgeeks*. [online] Available at: <https://www.geeksforgeeks.org/matplotlib-pyplot-show-in-python/> [Accessed 6 November 2020].

[5] GeeksforGeeks. 2020. *Matplotlib.Pyplot.Figure() In Python - Geeksforgeeks*. [online] Available at: <https://www.geeksforgeeks.org/matplotlib-pyplot-figure-in-python/> [Accessed 6 November 2020].

[6] Sodocumentation.net. 2020. *Matplotlib - Three-Dimensional Plots | Matplotlib Tutorial*. [online] Available at: <https://sodocumentation.net/matplotlib/topic/1880/three-dimensional-plots> [Accessed 6 November 2020].

[7] Matplotlib.org. 2020. *Mplot3d API — Matplotlib 2.0.2 Documentation*. [online] Available at: <https://matplotlib.org/mpl_toolkits/mplot3d/api.html> [Accessed 6 November 2020].

[8] Zone.ni.com. 2020. *Types Of Graphs And Charts - Flexrio Help - National Instruments*. [online] Available at: <https://zone.ni.com/reference/en-XX/help/372614J-01/lvconcepts/types_of_graphs_and_charts/> [Accessed 6 November 2020].

[9] Morgan, A., 2019. *Visualizing Beyond 3 Dimensions*. [online] Towards Data Science. Available at: <https://towardsdatascience.com/visualizing-beyond-3-dimensions-67531b431119> [Accessed 6 November 2020].