Adamson University
College of Engineering
Computer Engineering Department

Linear Algebra

Laboratory Activity No. 3

Linear Combination and Vector Spaces

Submitted by:
Columba, Lorenzo Miguel L.

Instructor:
Engr. Dylan Josh D. Lopez

October 29, 2020

I. Objectives

This laboratory activity aims to be familiarized in implementing linear combinations in two dimensional plane and visualization of spans of vectors in Python. Also to perform vector field operations using scientific programming.

II. Methods

The practices in this activity are to try different linear combinations using different scalar values and creating different spans of vectors. For task 1 and task 2, I used the following functions such numpy as np is for working with arrays. [1], matplotlib.pyplot as plt is the important function in matplot library which displays the plotting of the two-dimensional data [2]. I used np.array() used to create a grid of values, of all shape, and size of the arrays along each dimension in task 1 and task 2 problem which declaring the scalar values of the vector. [3] Then I used np.arrange() in each task to define the interval of values in an array where it starts into a specific range and stops. Also, it gives the interval of steps of the range [4]. After that I used plt.scatter() to plot each value in the data set that represented as dot [5], I add size for the plot to see how thick or thin the dot is to determine the size of arrays [6]. Also, I add alpha is to set the transparency of the plot in the plane in task 2 [7]. Then in task 1 and task 2 I used plt.xlim() to determine the x-axis limits [8] and plt.ylim() used to determine the y-axis limits. [9]. After that, the function plt.axvline() in both tasks was used to add a vertical line which represents as x-axis that sets the x position in the coordinates of a vertical line. [10] and plt.axhine(), used to add horizontal line in the axis which represents as the y axis that sets y position in the coordinates of a horizontal line [11]. For task 2 I used np.meshgrid() to create a rectangular grid that represente two dimensional arrays of X and coordinates of all points. [12]. To add title to the grid I used plt.title() [13]. Then for the figures, I used plt.legend() to describe elements in the graph [14]. For both tasks, I used plt.grid() to set the visibility of the grid inside the figures [15]. Then to display the plots of task 1 and task 2 I used plt.show()[16].

III. Results

The task 1, the linear equation is $X = c * \begin{bmatrix} 1 \\ 3 \end{bmatrix}, Y = c * \begin{bmatrix} 5 \\ 7 \end{bmatrix}$ while the vector form of the linear combination is $vector_x = \lfloor {}^{1} 3 \rfloor, vector_y = \lceil 7^{5} \rceil$. First factor I do is to import numpy as np.matplotlib.pyplot as plt, and %matplot inline. Then I declare the values of the x and y vector the use of np.array(). I used the c variable which means as a regular followed through the function np.arrange() which plays the ranges and steps of scalar values. Then plt.scatter() collectively with the parameters of consistent times the first vectors withinside the index zero to 1, in addition to the y vector. Then I introduced color and label for the vector x and y. However, I upload a marker that is the fashion of the plotted values for the vector y to look the distinction among the x and y vectors withinside the graph. To position the cartesian coordinates I used plt.axhline for the y-axis and plt.avline for the x-axis as a vertical line. Then I introduced a few colours for every line. The feature plt.identify is to position a few identities withinside the graph to suggest what sort of graph is withinside the end result and I alter its font length to 12. After that plt.legend is used to explain all of the figures declared withinside the factors which I placed on the decrease right. To finalize the end result I used the grid to make it greater presentable and plt.display used to show all data. As the end result of the vectors, it's miles linearly impartial as it has 2 exclusive scalar values however at some point they intersect which presentations as a one-dimensional space. See figure 1.1 for the code snippet. 1.2 for the flow chart and 1.three for the output of the code.

```
[6]  import numpy as np
     import matplotlib.pyplot as plt
     %matplotlib inline
```

```
[22] X = np.array([1,3])
     Y = np.array([5,7])
```

```
[23] c = np.arange(-10,10,0.25)
     plt.scatter(c*X[0],c*X[1],color = "red", label = "vector X")
     plt.scatter(c*Y[0],c*Y[1],color = "green", marker = "*", label = "vector Y")
     plt.xlim(-10,10)
     plt.ylim(-10,10)

     plt.axhline(y=0, color='blue')
     plt.axvline(x=0, color='blue')

     plt.title("Linear combination using different scalar values ", fontsize = 12)
     plt.legend(loc='lower right')

     plt.grid()
     plt.show()
```

Figure 1.1: Code snippet

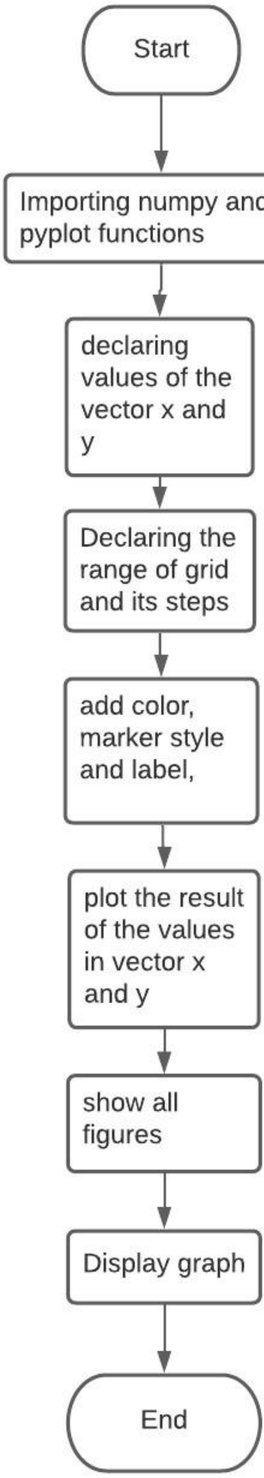

Figure 1.2: Flow chart of task 1 and 2

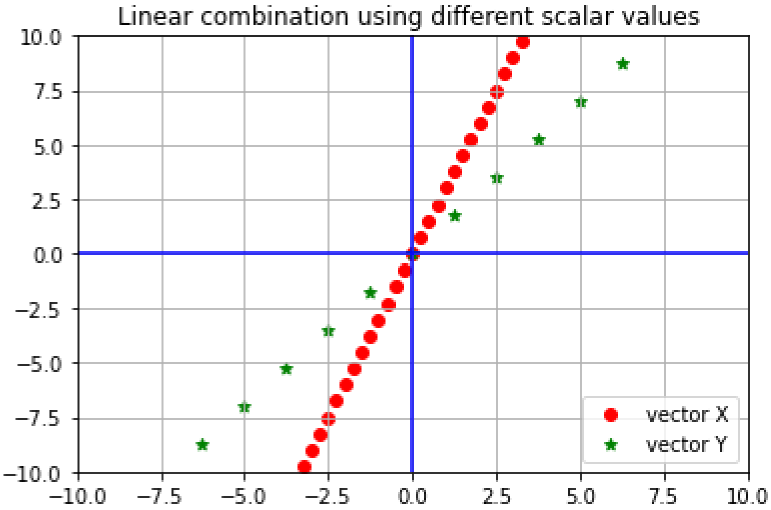

Figure 1.3: Output of the code.

\

The task 2, I need to create three different spans of the vector with using the two dimensional space with a span of plane.The first linear equation for this problem $S = \{(C_1 \times [10\ 7]) + (C_2 \times [3\ \ \text{-}8])\}$ second trial of linear equation is $S = \{(C_1 \times [11\ 21]) + (C_2 \times [-16\ 18])\}$ and for the last trial equation $S = \{(C_1 \times [3\ \ 5]) + (C_2 \times [1\ \text{-}21])\}$ while the vector form of the first linear combination is $vector_x$ = [10 7]+ $vector_y$ = [-3 8], second trial equation as $vector_x$ =[11 21], $vector_y$ = [-16 18] and the last vector equation is $vector_x$= [3 5]In the first trial of span, I used np.array to put the values of the vectors [8] , $vector_y$ = [1 -21]in x and y, followed by the R = np.arrange() meaning R is the rank that provides an estimate of the number of linearly independent rows or columns of a full matrix [17], also it has a range of -10 to 10 with the step of 1. In the code c_1 and c_2 as constants that is equivalent to the np.meshgrid() will display the span of planes in the graph because it is a rank of 2. For the spanRx is the sum of the constants times the values of the vector x is the index [0] added by the values of vector y in the index of [0]. This span will show the values of each vector that will be displayed in a graph. For the spanRy which is the same process as spanRx but the difference is indexing values which is [1]. The plt.scatter function with its parameters of spanRx, spanRy, s =5 meaning the size of which points in the graph, alpha = 0.75 as the thickness of the points, I add colors to each point as magenta, the marker represented as X which means it will display a figure point x in the graph and I label it as a point of values. Then the other process is the same from task 1. However, in this block of code I added a label title for the y and x-axis as plt.xlabel () and plt.ylabel(). Then same as the task 1, I used legends to describe its figures. Lastly for displaying the plots of the values I used the function plt.show().

See figure 2.1 for the code snippet, 2.2 for flow chart, 2.3 for the output of the the first trial, second output 2.4 and lastly for the trial of the third output .2.5

```
vectorX = np.array([10, 7])
vectorY = np.array([3,-8])

R = np.arange(-10,10,1)
c1, c2 = np.meshgrid(R,R)

spanRx = c1*vectorX[0] + c2*vectorY[0]
spanRy = c1*vectorX[1] + c2*vectorY[1]

plt.scatter(spanRx, spanRy, s=5, alpha=0.75, color = "magenta", marker = "X", label = "point values")

plt.axhline(y=0, color='blue')
plt.axvline(x=0, color='blue')

plt.title(" First trial of span of 2 Vectors ", fontsize = 12)
plt.xlabel('x-axis', fontsize=12)
plt.ylabel('y-axis', fontsize=12)
plt.legend(loc='upper right')

plt.grid()
plt.show()
```

Figure 2.1 : Code snippet

Figure 2.2 Flow chart

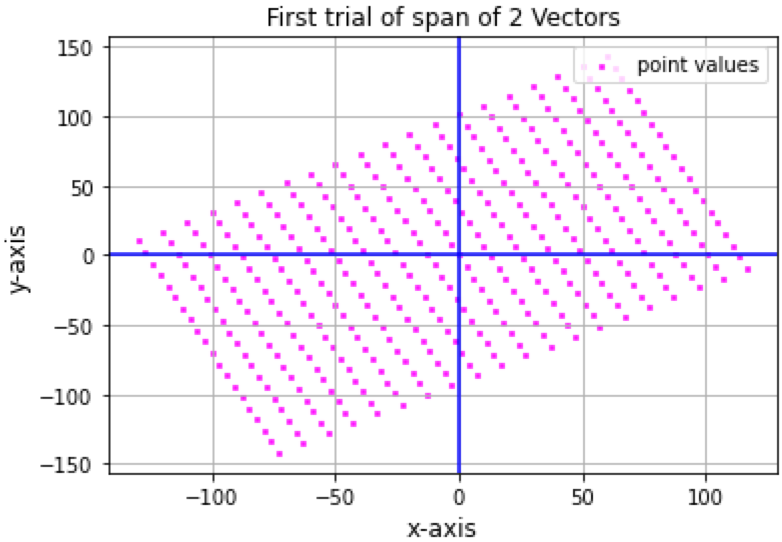

Figure 2.3: Output of the code

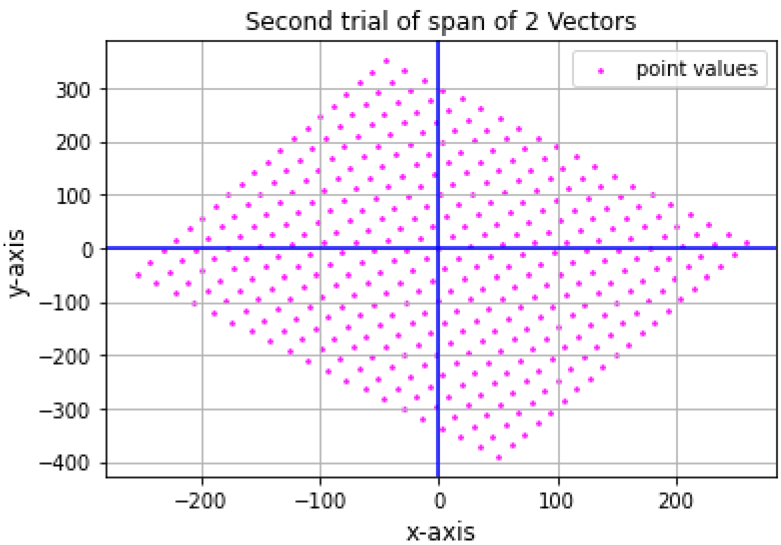

Figure 2.4: output of second trial .

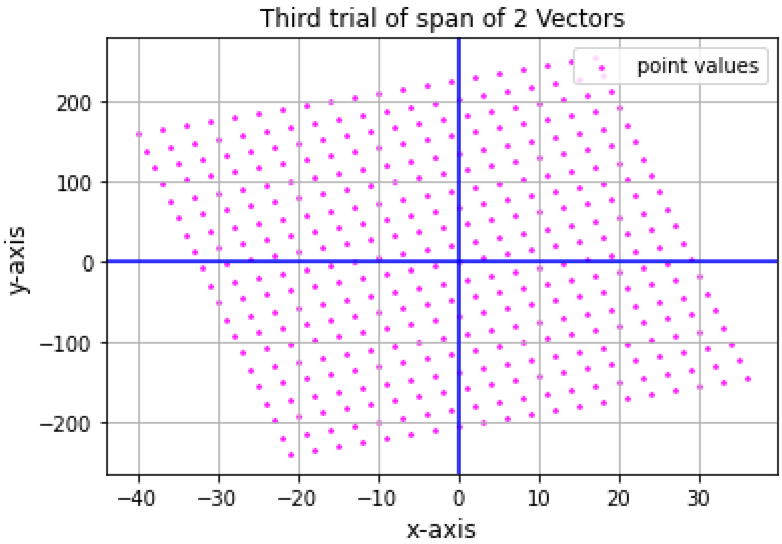

Figure 2.5: Output of third trial

In the form of the vector in ℝ = 3 or Rank 3 could be 3dimensional as it has 3 pivots withinside the rows or columns of matrices. While the form of a vector in ℝ = four or Rank four could be four-dimensional as it has 4 pivots withinside the rows or columns of matrices. However, If a span of the vector is 3 however the pivots are best two, therefore, it might be a span of two-dimensional planes of rank three. This kind of rank is linearly impartial as it doesn’t show one line that's known as linearly dependent. The function of unit vectors is generally used to offer a path with a scalar coefficient. The relation of those unit vectors to linear aggregate is to feature and multiply all of the feasible scalars with the regular time period and getting its result alongside the x and y axis.

IV. Conclusion

Creating this application of the laboratory is about how you'll visualize a set of vectors this is increased into constant values. I discovered that during plotting a scatter factor size, its opacity, form, variety, and steps are wanted with a view to completely describe and visualize a fixed of vectors right into a graph. Also, this laboratory offers me a new approach in plotting a graph with a unique style of factors that I used the marker to provide detail about the point, not simply the simplest dot however it could be a form of a square, triangle, star, etc. Also, the variety could be the limiter in which the x and y values could be plotted and knowledge the correlation of traces which if one line is displayed in a plot with the aid of using the sum of 2 vectors it might be linearly based which means they have the equal values or they have got the equal set of plotted values withinside the graph it could overlap however nonetheless it shows one line. However, on the subject of getting a span of an aircraft which means the pivots of the set of vectors are in a 2-dimensional area. It shows a brand new set of scattering factors that are known as linearly independent. They aren't correlated to every different as it does show a one-dimensional area.For me, this idea of linear aggregate may be implemented withinside the experience of plotting a fixed of vectors that locates a populace increase of a bacterial how does it multiply, what's the time c programming language of a bacterial to unfold faster, etc. Besides, It may be implemented to the examination of quantum mechanics that lets in a linear vector area that has homes that ought to be observed together with linear superposition, scalar multiplication over an area and the set ought to incorporate a 0 vector. [18] . Learning the combination of linear is good and not that hard in the use of programming language.

References

[1] Tutorialspoint, 2020. *About " NumPy Matplotlib*." Available:

https://www.tutorialspoint.com/numpy/numpy_matplotlib.htm Date Accessed: 10/23/2020

[2] The Matplotlib development team.Pyplot, 2020. *About "Pyplot tutorial.*" Available: https://matplotlib.org/users/pyplot_tutorial.html#:~:text=pyplot%20is%20a%20collection%20of,the%20pl ot%20with%20labels%2C%20etc. Date Accessed: 10/23/2020

[3] The SciPy community, 2020. *About "numpy.array*" Available: https://numpy.org/doc/stable/reference/generated/numpy.array.html Date Accessed: 10/23/2020

[4] Stojiljković, 2020. *About "*NumPy arange(): How to Use np.arange()" Available: https://realpython.com/how-to-use-numpy-arange/ Date Accessed: 10/23/2020

[5] The Matplotlib development team.Pyplot, 2020. *About "matplotlib.pyplot.scatter"* Available: https://matplotlib.org/3.3.2/api/_as_gen/matplotlib.pyplot.scatter.html. Date Accessed: 10/23/2020

[6] , Netlib, 2020. *About "MATLAB Function Reference.* Available: http://www.ece.northwestern.edu/localapps/matlabhelp/techdoc/ref/size.html#:~:text=size%20(MATLAB%20Functions)&text=d%20%3D%20si ze(X)%20returns,X%20specified%20by%20scalar%20dim%20. Date Accessed: 10/23/2020

[7], The MathWorks, Inc., 2020. *About "Alpha"* Available:

https://www.mathworks.com/help/matlab/ref/alpha.html#:~:text=The%20alpha%20function%20sets%20the,property%20of%20the%20scatter%20object Date Accessed: 10/23/2020

[8] The MathWorks, Inc., 2020. *About "xlim*" Available: https://www.mathworks.com/help/matlab/ref/xlim.html#:~:text=xlim%20auto%20sets%20an%20automati c,the%20axes%20to%20'auto'%20. Date Accessed: 10/23/2020

[9] The MathWorks, Inc.,2020. *About "ylim*" Available: https://www.mathworks.com/help/matlab/ref/ylim.html#:~:text=ylim%20auto%20sets%20an%20automati c,the%20axes%20to%20'auto'%20. Date Accessed: 10/23/2020

[10] GeeksforGeeks,2020. *About"Matplotlib.axes.Axes.axvline() in Python)"* Available: https://www.geeksforgeeks.org/matplotlib-axes-axes-axvline-in-

python/#:~:text=axvline()%20function%20in%20axes,vertical%20line%20across%20the%20axis.&text=Parameters%3A%20This%20method%20accept%20the,with%20default%20value%20of%200. Date Accessed: 10/23/2020

[11] GeeksforGeeks , 2020. *About "matplotlib.pyplot.axhline() in Python"* Available: https://www.geeksforgeeks.org/matplotlib-pyplot-axhline-inpython/#:~:text=The%20axhline()%20function%20in,horizontal%20line%20across%20the%20axis.&text =Parameters%3A%20This%20method%20accept%20the,coordinates%20of%20the%20horizontal%20line Date Accessed: 10/23/2020

[12]] GeeksforGeeks , 2020. *About "Numpy Meshgrid function"* Available: https://www.geeksforgeeks.org/numpy-meshgrid-function/ Date Accessed: 10/23/2020

[13] The Matplotlib development team, 2020. *About "*matplotlib.pyplot.title*"* Available: https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.title.html Date Accessed: 10/23/2020

[14] GeeksforGeeks ,2020. *About "Matplotlib.pyplot.legend() in Python."* Available: https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.title.html Date Accessed: 10/23/2020

[15] Tutorialspoin,2020. *About" Matplotlib – Grids"* Available: https://www.tutorialspoint.com/matplotlib/matplotlib_grids.htm. Date Accessed: 10/23/2020

[16] VanderPlas,2016. *About"Visualization with Matplotlib."*Available: https://jakevdp.github.io/PythonDataScienceHandbook/04.00-introduction-to-matplotlib.html. Date Accessed: 10/23/2020

[17] The Society for Industrial and Applied Mathematics,1999. *About "Rank."* Available: http://www.ece.northwestern.edu/localapps/matlabhelp/techdoc/ref/rank.html#:~:text=rank%20(MATLAB%20Functions)&text=The%20rank%20function%20provides%20an,))*norm(A)*eps%20. Date Accessed: 10/23/2020

[18] Gosain, 2018. *About "Concept of linear algebra and linear combinations*". Available:

https://www.quora.com/Why-is-the-concept-of-linear-combination-very-important-in-mathematics-Isthere-a-similar-generalisation-to-degrees-higher-then-1-such-as-quadratic-combination-etc. Date Accessed: 10\24\2020