Linear Algebra

Laboratory Activity No. 2

Intro to Vectors And Numpy

|  |  |
| --- | --- |
| *Submitted by:* | *Instructor:* |
| Casiño, Christian Carlos H. | Engr. Dylan Josh D. Lopez |

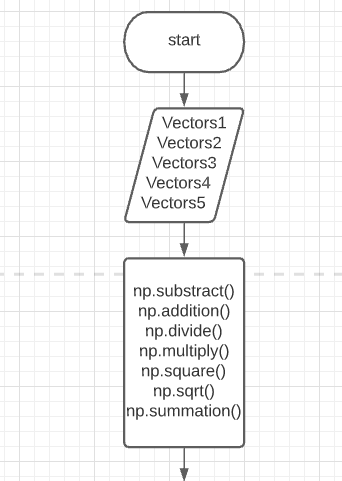
February 19, 2021

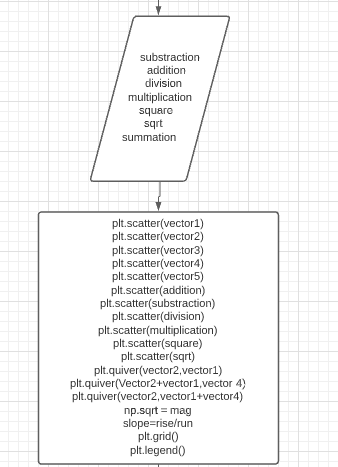
# Objectives

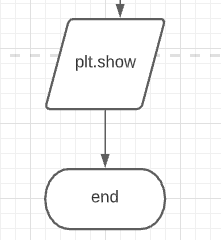
This laboratory activity aims to be familiar with the libraries in Python for numerical and scientific programming, visualize vectors through Python programming, and perform simple vector operations through code. This laboratory activity aims to create tasks with the use of the libraries Numpy and Matplotlib, the vectors will perform the following operations in any combination and any number of trials: addition, subtraction, multiplication, division, squaring, square root, and summation.

# Methods

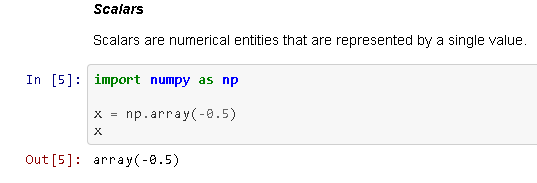
First, the programmer should draw a flow chart as a reference in our program which will teach us the flow of the program. Secondly, the programmer should discuss and review how to manipulate the vectors, arithmetic, scalars, libraries, scaling, and formulas for filling up the flow of the program. The deliverance of this activity is to help the programmer to familiarize the use of numpy and matplotlib libraries.





Figure 1.0 flowchart by Casiño

# Results

Figure 1.1 Scalars by Dylan Josh Domingo Lopez

In this figure we can learn about scalars. Scalars can be determined by its singular represented value. Additional information, the data that will be stored by the scalar variable can be of the different type like string, character, floating point, a large group of strings.

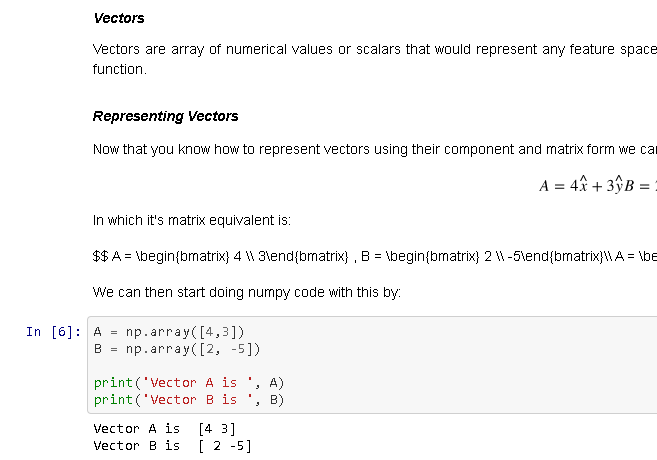


Figure 1.2 Vectors by Dylan Josh Domingo Lopez

In this Figure, Vectors are arrays of numerical values or with the help of scalars it can represent any point of space. Vector in the numpy library are the components, which make ordinary numbers combine together . The idea is that when a vector is in a list of numbers, we can use vector algebra as operations performed on the numbers in the list. In other words vector is the numpy 1-D array but may become 2-D depending in the given vectors.

.

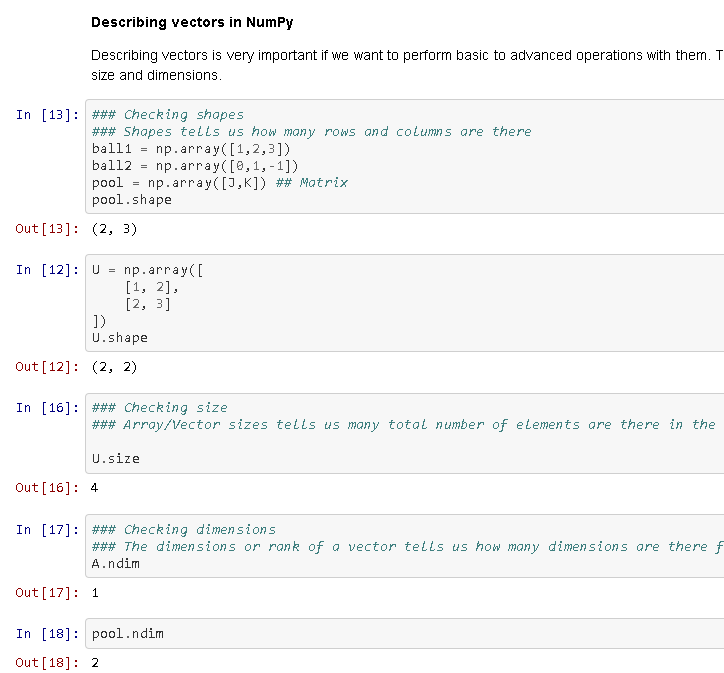


Figure 1.3 Describing vectors in Numpy by Dylan Josh Domingo Lopez

In this figure, vectors can be describe by its shape, size or dimension using Numpy. Using this feature will be valuable to check the shape and size of an array both for further calculations and simply as a guide checking after some operation because when two vectors have unequal set of elements and forming them to a operation, it would say that it is in error.

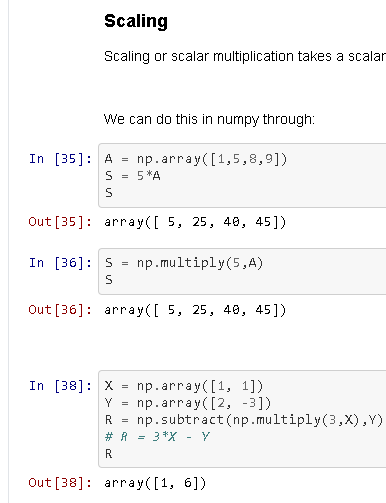


Figure 1.4 Scaling by Dylan Josh Domingo Lopez

In this figure, scaling or scalar multiplication can help us with the arithmetic of the vectors. The simple operations using the symbol operations has the same approach with matrix operation for vectors. Each operations with the library numpy is explained in the table.

|  |  |
| --- | --- |
| Np.add() | This function is used to perform element wise matrix addition. |
| Np.substract() | This function is used to perform element wise matrix subtraction. |
| Np.divide() | This function is used to perform element wise matrix division. |
| Np.multiply() | This function is used to perform element wise matrix multiplication. |
| Np.dot | This function is used to compute the matrix multiplication, rather than element wise multiplication. |
| Np.sqrt() | This function is used to compute the square root of each element of matrix. |
| Np.sum() | This function is used to add all the elements in matrix. Optional “axis” argument computes the column sum if axis is 0 and row sum if axis is 1. |
| Np.t() | This argument is used to transpose the specified matrix. |

Table 1 Matrix Operations by https://www.geeksforgeeks.org/matrix-manipulation-python/

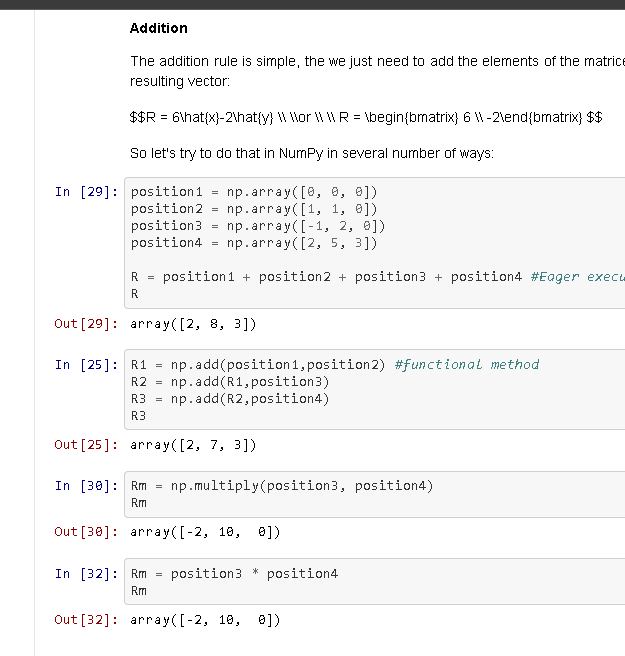
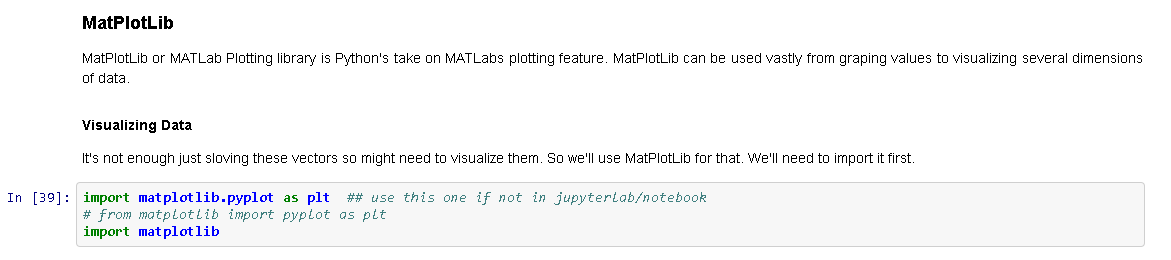
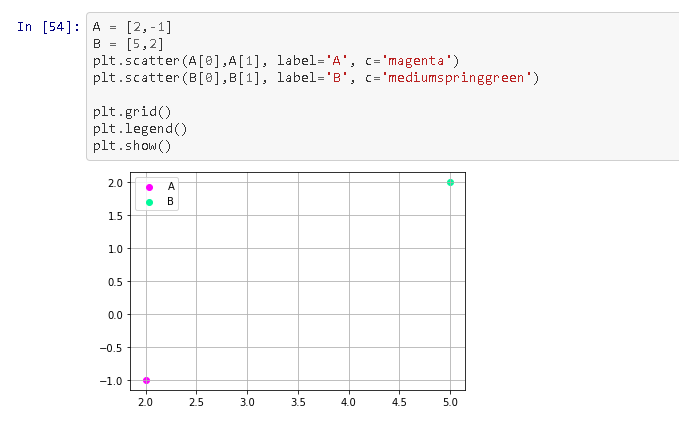


Figure 1.5 Addition Rule by Dylan Josh Domingo Lopez

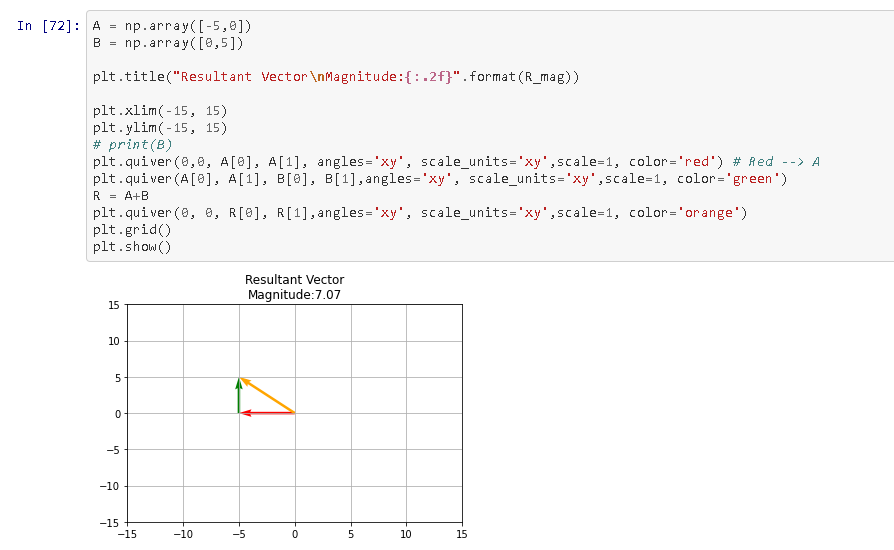
In this figure, we can figure out how we can add, subtract, multiply and divide elements of the matrices using the library of Numpy and also how to manually add vectors without the use of Numpy.

Figure 1.6 Matplotlib by Dylan Josh Domingo Lopez

In this figure, we can familiarize ourselves in the library of Matplotlib. This helps to plot the vectors we can obtain in the graph visualizing it in x and y dimensions.

Figure 1.7 Visualization by Dylan Josh Domingo Lopez

In this figure, We will learn how to plot the vectors given in A and B. Using plt.scatter we can declare the first and second element of the vector and label it with whatever we want, and declare a color so we can differentiate it with another vector.

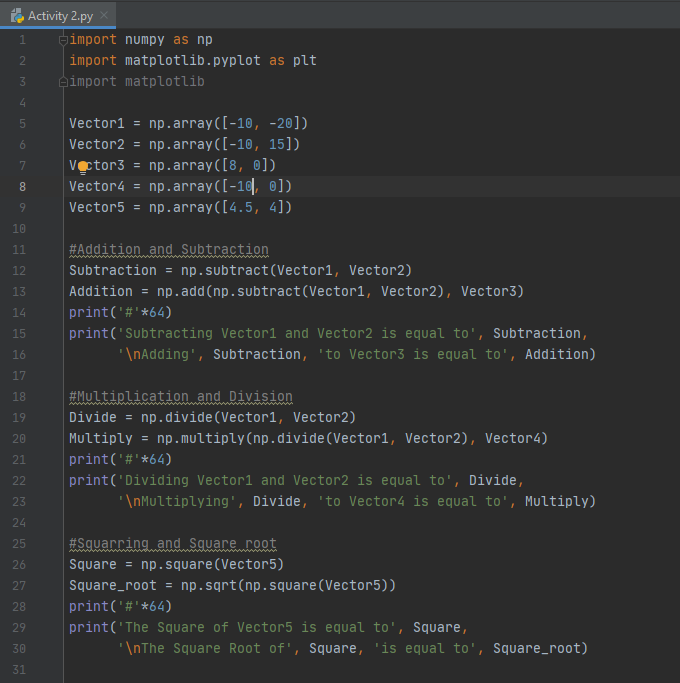
Figure 1.8 Resultant Vector by Dylan Josh Domingo Lopez

In this figure, we can identify the resultant vector using the Pythagorean theorem.

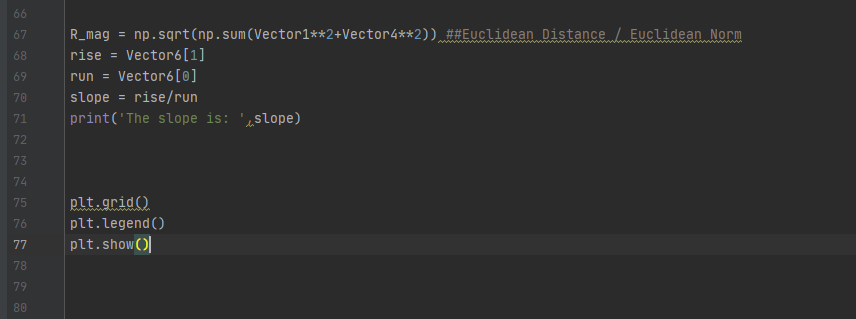


Figure 1.9 Slope by Dylan Josh Domingo Lopez

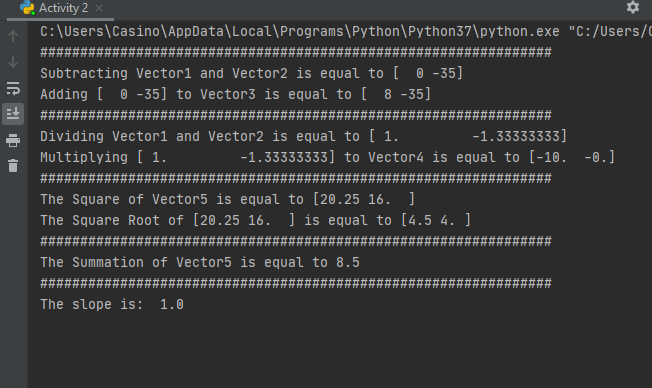
In this figure, we can determine the slope with the use of the rise and run formula.

Figure 2.1 Codes by Casiño

In this figure, the programmer have declared different vectors with their own respective value. With the help of the vectors that we have, the programmer will use them to portray how to add, subtract, multiply, divide, square, square root, and summation. Addition and Subtraction will add and subtract arrays and requires two array with the same shape and dimension. Multiplication and Division will multiply and divide arrays and also requires two arrays to return a product or quotient. Squaring and Square root requires only one array, this will return as a square of the array and root of the array. Summation will give the sum of arrays over a specified axis.

Figure 2.2 Codes by Casiño

This figure shows the plotting points of each vector and results in the graph. It also shows the resultant vector and the slope. Plt.scatter is used to pinpoint any given of x and y which is equal to index 0 and index 1 inside of a vector. Plt.quiver is used to determine where the arrow will go with a argument of based on my observation (x1,y1) for its origin and (x2,y2) for its destination.

Figure 3.1 Output by Casiño

This is the overall output printed(the graph is in the next figure). Results are presented here based on the codes in Figures 2.1 and 2.2.

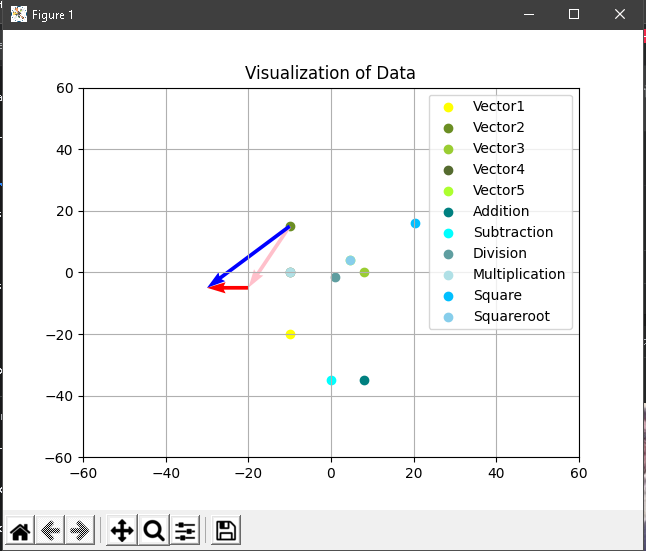


Figure 3.2 Output by Casiño

In this figure, the graph shows all the plots of the vectors and their result based on figure 3.1. The resultant vector is also portrayed here.

# Conclusion

The programmer have realized that NumPy is best used in numerical problems and with the combination of Matplotlib, it helps portray the plots of each vector and it also lets you perform a whole lot of complex scientific calculations on these arrays. The programmer also concluded that the library numpy and the python syntax “list” is almost the same but with a much better and wider area for numerical improvements. To the idea that if we are given a large amount. These two combined make a wide range of methods in computation and graphical efficiency. As a programmer, these libraries best help in statistics because NumPy helps in assigning and computes values for getting the results in each vector operations then proceeding with the use of matplotlib for graphing data of the vectors and plotting all the points and quivers with its respective labels.

**References**

[1] Dyjdlopez, “dyjdlopez/linearAlgebra2021,” *GitHub*. [Online]. Available: https://github.com/dyjdlopez/linearAlgebra2021. [Accessed: 20-Feb-2021].

[2] “numpy.sum() in Python,” *GeeksforGeeks*, 03-Apr-2020. [Online]. Available: https://www.geeksforgeeks.org/numpy-sum-in-python/. [Accessed: 20-Feb-2021].

[3] “numpy.addition() in Python,” *GeeksforGeeks*, 03-Apr-2020. [Online]. Available: https://www.geeksforgeeks.org/numpy-addition-in-python/. [Accessed: 20-Feb-2021].

[4] “numpy.subtraction() in Python,” *GeeksforGeeks*, 03-Apr-2020. [Online]. Available: https://www.geeksforgeeks.org/numpy-subtraction-in-python/. [Accessed: 20-Feb-2021].

[5] “numpy.multiplication() in Python,” *GeeksforGeeks*, 03-Apr-2020. [Online]. Available: https://www.geeksforgeeks.org/numpy-multiplication-in-python/. [Accessed: 20-Feb-2021].

[6] “numpy.division() in Python,” *GeeksforGeeks*, 03-Apr-2020. [Online]. Available: https://www.geeksforgeeks.org/numpy-division-in-python/. [Accessed: 20-Feb-2021].

[7] “numpy.square() in Python,” *GeeksforGeeks*, 03-Apr-2020. [Online]. Available: https://www.geeksforgeeks.org/numpy-square-in-python/. [Accessed: 20-Feb-2021].

[8] “numpy.squareroot in Python,” *GeeksforGeeks*, 03-Apr-2020. [Online]. Available: https://www.geeksforgeeks.org/numpy-squareroot-in-python/. [Accessed: 20-Feb-2021].