Adamson University
College of Engineering
Computer Engineering Department

Linear Algebra

Laboratory Activity No. 2

Plotting Vectors using Numpy and MatPlotLib

Submitted by:

Canoza, Cherrylyn S.

Instructor:

Engr. Dylan Josh D. Lopez

October 08, 2020

I. Objectives

This laboratory activity aims to implement the principles and techniques of using Python in solving and plotting vectors using Numpy and MatPlotLib.

II. Methods

This activity intends analyze and understand plotting of vectors using of importing of the Numpy and Matplotlib in python as it encodes the program. Using the Numpy library, it can able to perform solutions, equations and solving Mathematical problems in programming Python using array and different function. Matplotlib is another library used to input functions that will visualize the vectors into arrows and actual plot of it.

The deliverables of this activities are the visual representation of the vectors as well as the analization of the problem before it plots its outcome. It gives a result of solving the vector using the functions in numpy as well as the actual plotting of vectors which involves arrows and graphs using the functions of matplotlib.

III. Results

Part I : Skyline Eagle Fly

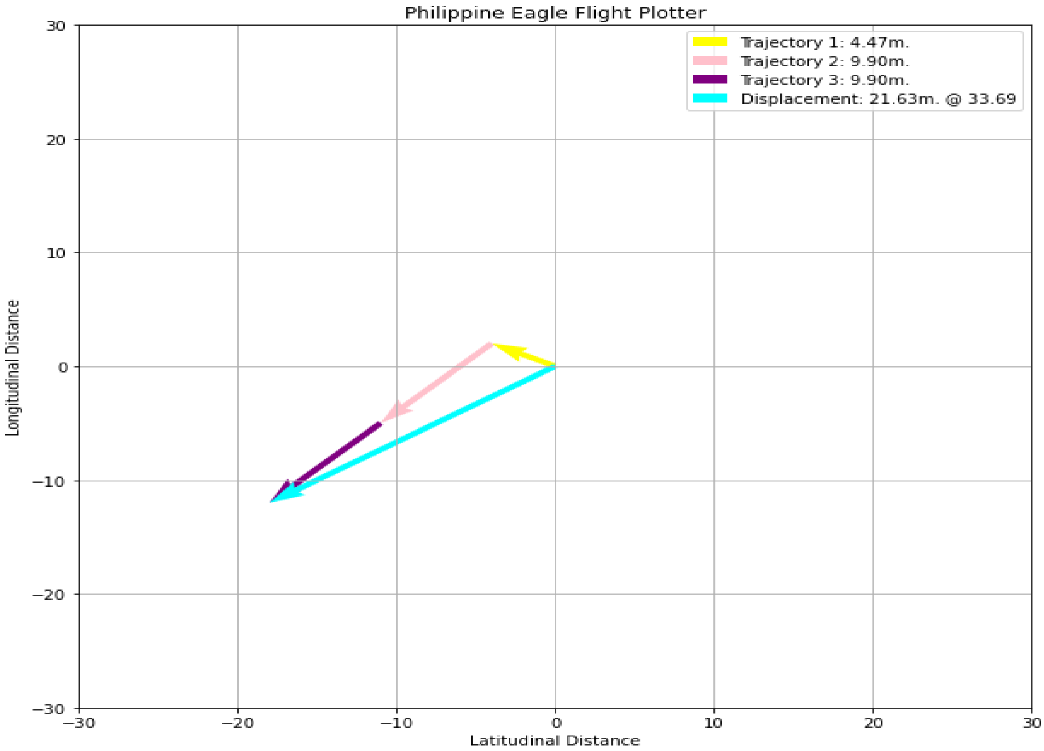


Figure 1 Exercise 1

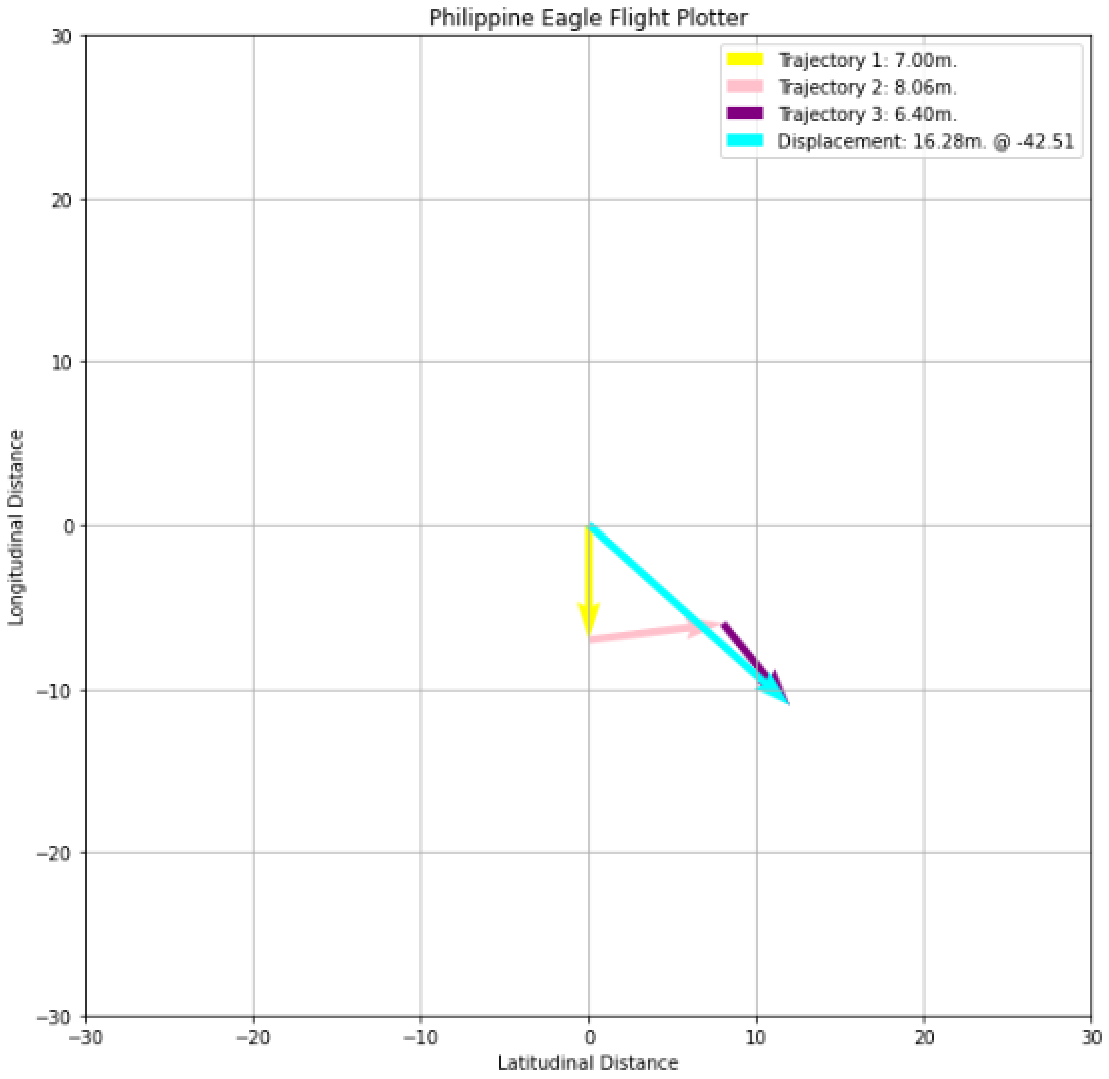


Figure 2 Exercise 1

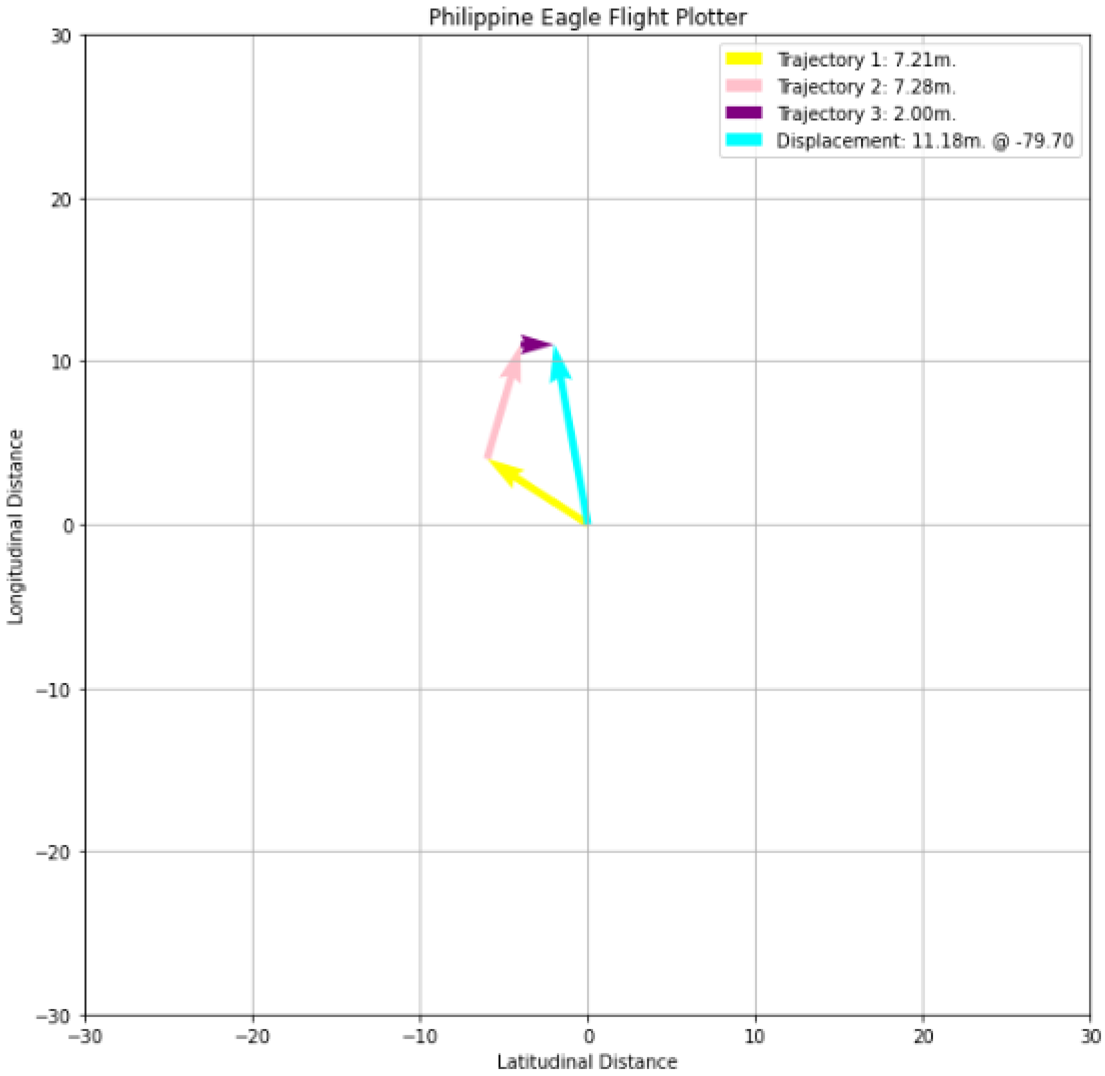


Figure 3 Exercise 1

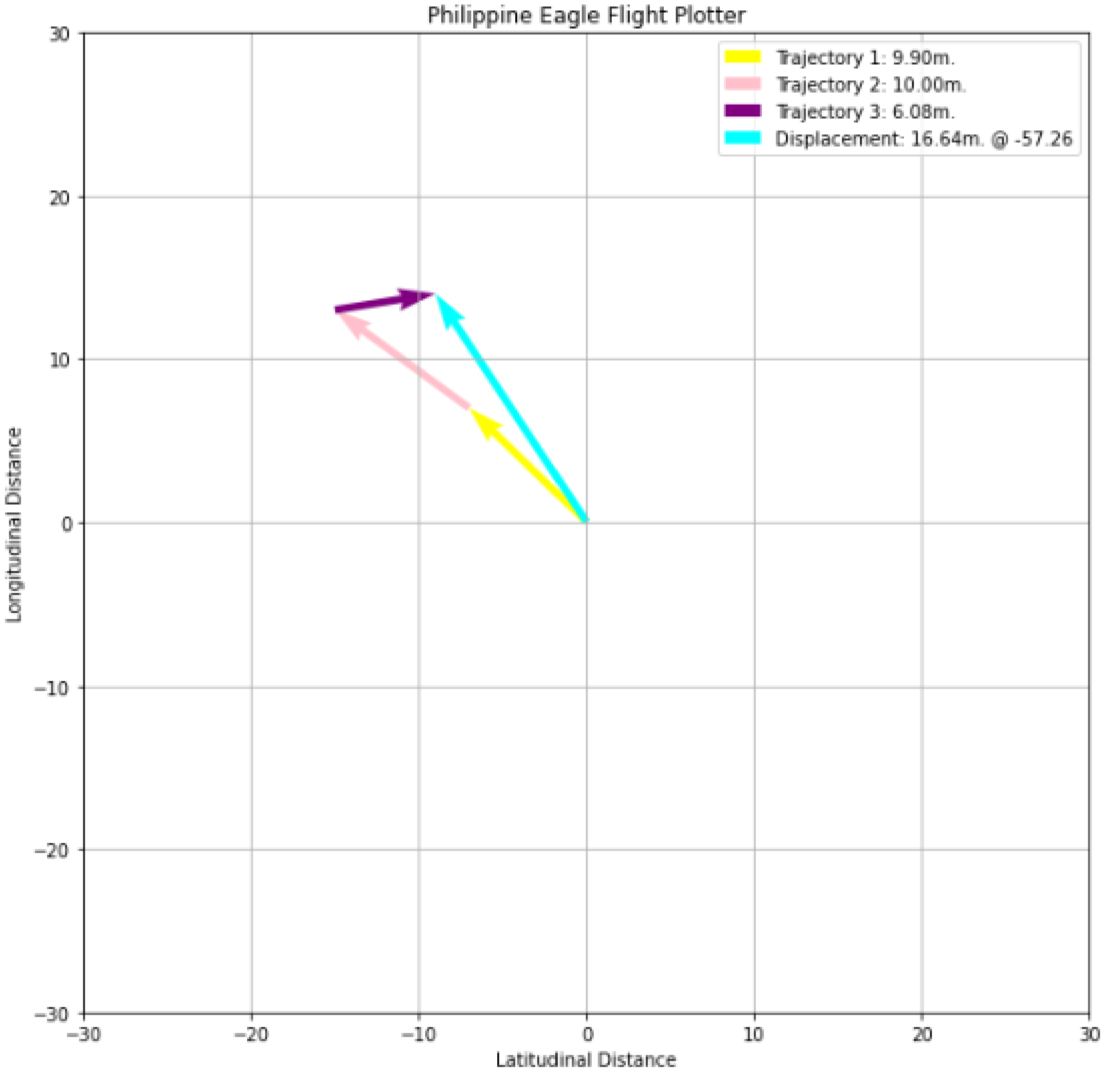


Figure 4 Exercise 1

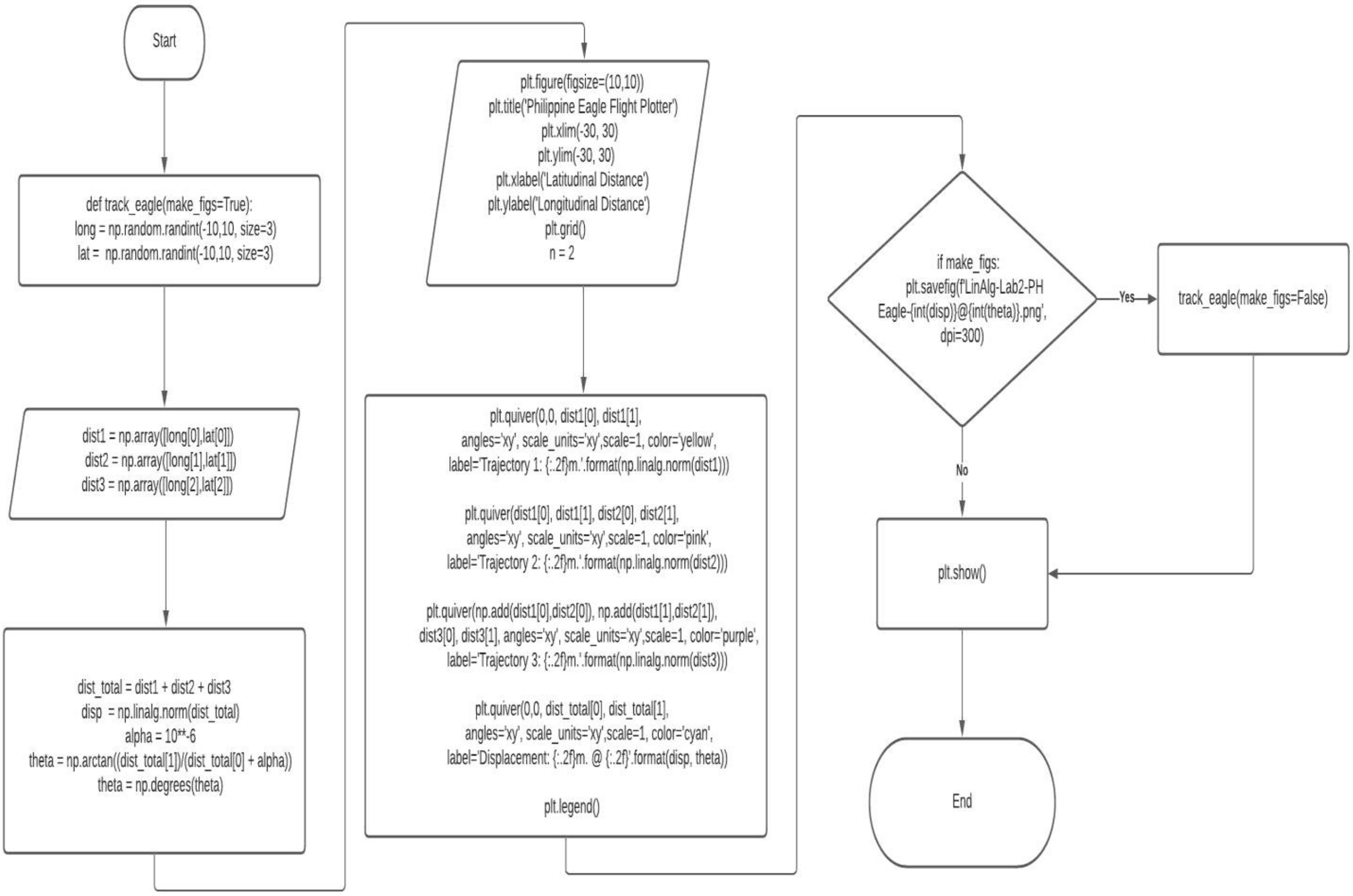
1. What is the use of the variable α?

The use of the alpha here is not actually used since its equivalent is 0.000001 and if ever it is added in the equation, it will give no difference or super minimal difference. The exact equation in getting the angles is just using y value divided by the x value and if the given value of alpha is added to the x value then it will only give zeroes, it makes no sense and if you analyze it in real equation, it can be rounded off and will only give the same value as what is x value have.

2. Kindly explain the process of plotting the eagle flight vectors.

Using the answers of the vectors above which the angle, the positions and displacements are defined or already have a value, it can now be plotted using the plt.quiver function that allows make visual representation of vectors in programming, it creates arrows and lines to visualize the plot of the vector. There are four lines need to be plotted which are the trajectory one, two, three and the total displacement that will connect these three trajectories. The plt.quiver allows to input values inside of it that will represent the lines and arrows, the colors should differs so that the arrows are visible as the different trajectories are plotted. Trajectory one has an input of 0,0, dist1[0], dist1[1], this is where the eagle first landed, followed by the second trajectory that uses dist1[0], dist1[1], dist2[0], dist2[1], then last trajectory has an input of adding the distances that is np.add(dist1[0],dist2[0]), np.add(dist1[1],dist2[1]), dist3[0], dist3[1]. To connect these three trajectory, we have the input of the total displacement which is using the computed total distance, 0,0, dist_total[0], dist_total[1]. With this, we can able to plot the eagle's flight vectors if it satisfies the condition below calling the function which will be like this, if make_figs: plt.savefig(f'LinAlg-Lab2-PH Eagle-{int(disp)}@{int(theta)}.png', dpi=300), then it will print out the eagle's flight vector successfully.

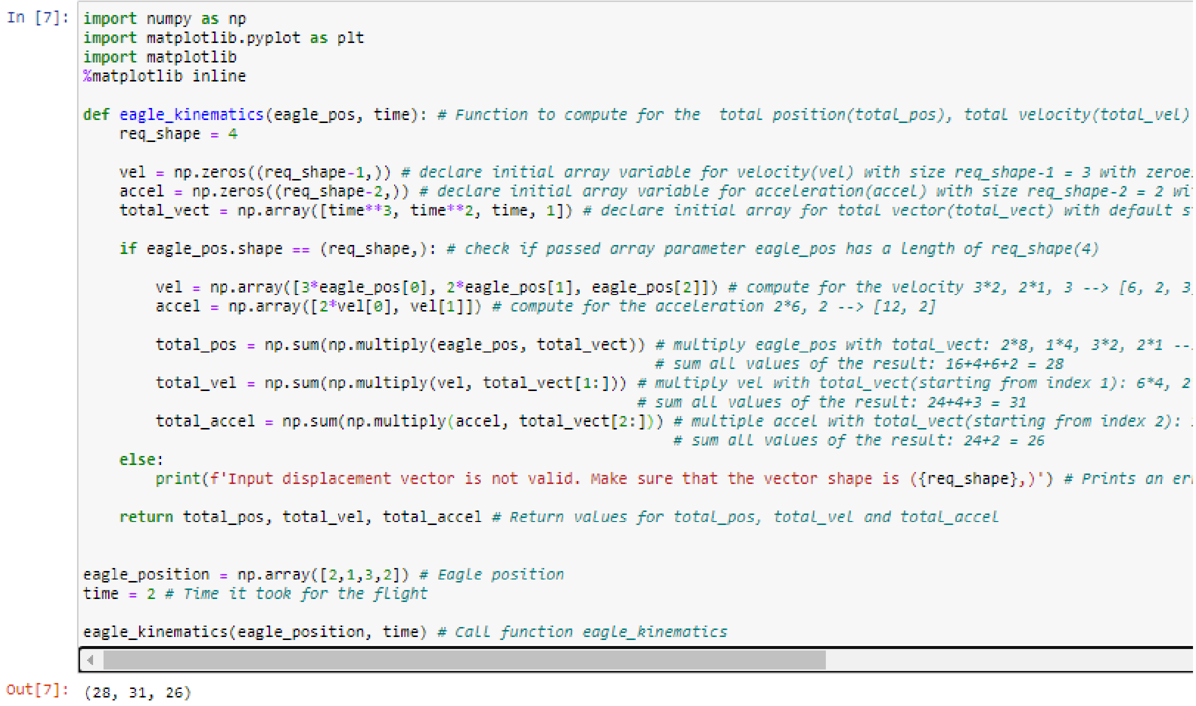
3. Provide a flow chart describing the 'track_eagle' function. Please use LucidCharts or similar apps in making charts.



```
def track_eagle(make_figs=True):
long = np.random.randint(-10,10, size=3)
lat =  np.random.randint(-10,10, size=3)

dist1 = np.array([long[0],lat[0]])
dist2 = np.array([long[1],lat[1]])
dist3 = np.array([long[2],lat[2]])

dist_total = dist1 + dist2 + dist3
disp  = np.linalg.norm(dist_total)
alpha = 10**-6
theta = np.arctan((dist_total[1])/(dist_total[0] + alpha))
theta = np.degrees(theta)

plt.figure(figsize=(10,10))
plt.title('Philippine Eagle Flight Plotter')
plt.xlim(-30, 30)
plt.ylim(-30, 30)
plt.xlabel('Latitudinal Distance')
plt.ylabel('Longitudinal Distance')
plt.grid()
n = 2

plt.quiver(0,0, dist1[0], dist1[1],
angles='xy', scale_units='xy',scale=1, color='yellow',
label='Trajectory 1: {:.2f}m.'.format(np.linalg.norm(dist1)))

plt.quiver(dist1[0], dist1[1], dist2[0], dist2[1],
angles='xy', scale_units='xy',scale=1, color='pink',
label='Trajectory 2: {:.2f}m.'.format(np.linalg.norm(dist2)))

plt.quiver(np.add(dist1[0],dist2[0]), np.add(dist1[1],dist2[1]),
dist3[0], dist3[1], angles='xy', scale_units='xy',scale=1, color='purple',
label='Trajectory 3: {:.2f}m.'.format(np.linalg.norm(dist3)))

plt.quiver(0,0, dist_total[0], dist_total[1],
angles='xy', scale_units='xy',scale=1, color='cyan',
label='Displacement: {:.2f}m. @ {:.2f}'.format(disp, theta))

plt.legend()

if make_figs:
plt.savefig(f'LinAlg-Lab2-PH
Eagle-{int(disp)}@{int(theta)}.png',
dpi=300)

track_eagle(make_figs=False)

plt.show()
```

Figure 1 Exercise 2

Part II : Reverse Engineering



```
In [7]: import numpy as np
        import matplotlib.pyplot as plt
        import matplotlib
        %matplotlib inline

        def eagle_kinematics(eagle_pos, time): # Function to compute for the  total position(total_pos), total velocity(total_vel)
            req_shape = 4

            vel = np.zeros((req_shape-1,)) # declare initial array variable for velocity(vel) with size req_shape-1 = 3 with zeroe
            accel = np.zeros((req_shape-2,)) # declare initial array variable for acceleration(accel) with size req_shape-2 = 2 wi
            total_vect = np.array([time**3, time**2, time, 1]) # declare initial array for total vector(total_vect) with default s

            if eagle_pos.shape == (req_shape,): # check if passed array parameter eagle_pos has a length of req_shape(4)

                vel = np.array([3*eagle_pos[0], 2*eagle_pos[1], eagle_pos[2]]) # compute for the velocity 3*2, 2*1, 3 --> [6, 2, 3
                accel = np.array([2*vel[0], vel[1]]) # compute for the acceleration 2*6, 2 --> [12, 2]

                total_pos = np.sum(np.multiply(eagle_pos, total_vect)) # multiply eagle_pos with total_vect: 2*8, 1*4, 3*2, 2*1 --
                                                                       # sum all values of the result: 16+4+6+2 = 28
                total_vel = np.sum(np.multiply(vel, total_vect[1:])) # multiply vel with total_vect(starting from index 1): 6*4, 2
                                                                     # sum all values of the result: 24+4+3 = 31
                total_accel = np.sum(np.multiply(accel, total_vect[2:])) # multiple accel with total_vect(starting from index 2):
                                                                         # sum all values of the result: 24+2 = 26
            else:
                print(f'Input displacement vector is not valid. Make sure that the vector shape is ({req_shape},)') # Prints an er

            return total_pos, total_vel, total_accel # Return values for total_pos, total_vel and total_accel


        eagle_position = np.array([2,1,3,2]) # Eagle position
        time = 2 # Time it took for the flight

        eagle_kinematics(eagle_position, time) # Call function eagle_kinematics

Out[7]: (28, 31, 26)
```

Figure 2 Exercise 2

1. What is the function for? Explain the inputs, process, functions used, and the outputs.

   The declared function is used to compute for the total position or its variable name is ‘total_pos’, the total velocity or ‘total_vel’ and the total acceleration or ‘totatl_accel’ of the eagle as it travels and fly in the sky. The input needed is the value of velocity, acceleration and total vector so it needs to declare initial array of them in order to call them later for the operations and getting the desired outcome. Using if statement, it has check if passed array verifies the condition above and if that happens, operations will follow. The method of getting the velocity, acceleration, total position, total velocity, and total acceleration which is required and to that, you have to call the values of the function in order to solve for the needed method. If the input didn’t satisfy the condition then it will fall to an error or print out saying that displacement vector is not valid.

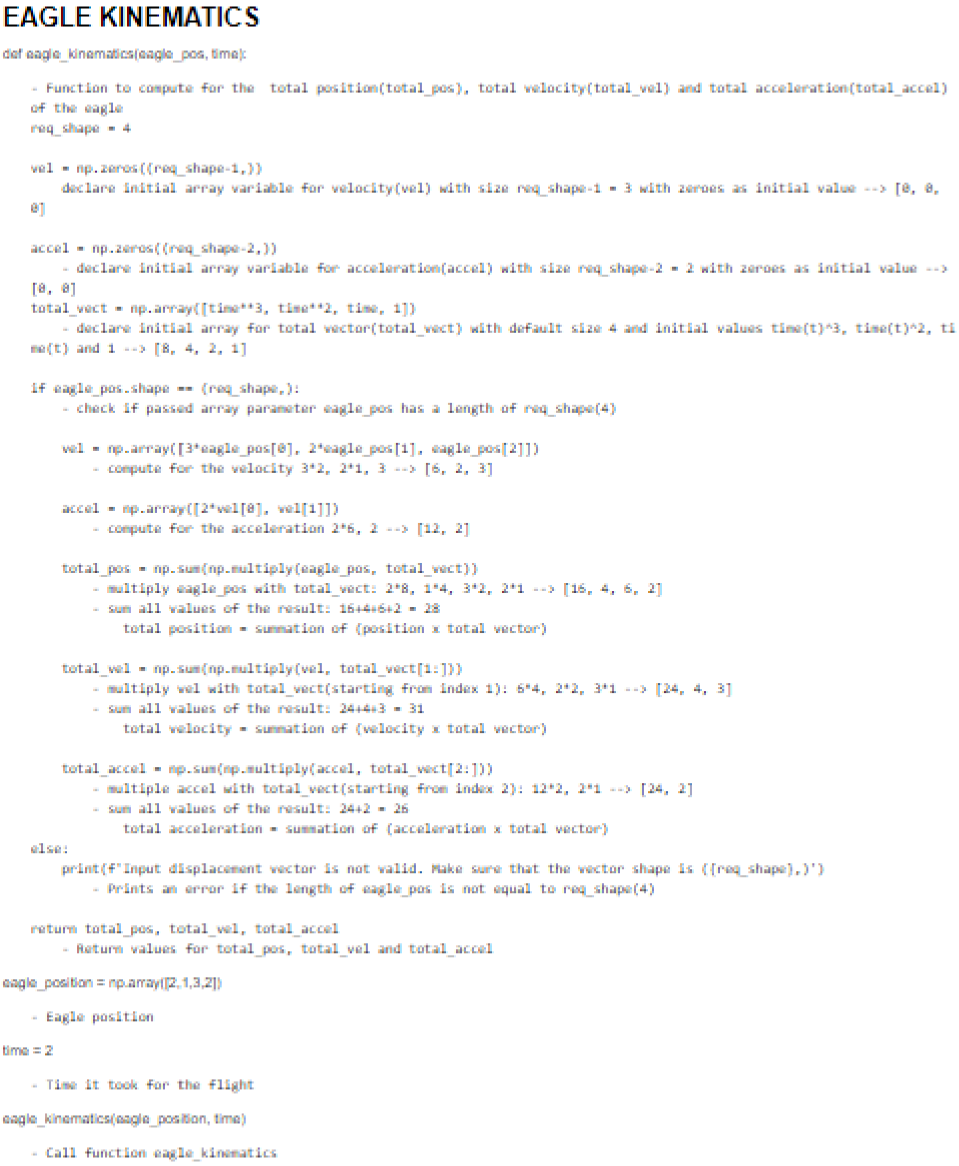
2. Kindly provide equations for s_t,v_t, and a_t basing on their code.

$$total\ position\ or\ S_t = \sum (position\ x\ total\ vector)$$

$$total\ velocity\ or\ V_t = \sum (velocity\ x\ total\ vector)$$

$$total\ acceleration\ or\ A_t = \sum (acceleration\ x\ total\ vector)$$

3. Re-type the 'eagle_kinematics' function in your jupyter notebook. Write a brief description of the function before the code cell in a markdown. Include the equations from (2.).



EAGLE KINEMATICS

```
def eagle_kinematics(eagle_pos, time):

    - Function to compute for the  total position(total_pos), total velocity(total_vel) and total acceleration(total_accel)
    of the eagle
    req_shape = 4

    vel = np.zeros((req_shape-1,))
        declare initial array variable for velocity(vel) with size req_shape-1 = 3 with zeroes as initial value --> [0, 0,
    0]

    accel = np.zeros((req_shape-2,))
        - declare initial array variable for acceleration(accel) with size req_shape-2 = 2 with zeroes as initial value -->
    [0, 0]
    total_vect = np.array([time**3, time**2, time, 1])
        - declare initial array for total vector(total_vect) with default size 4 and initial values time(t)^3, time(t)^2, ti
    me(t) and 1 --> [8, 4, 2, 1]

    if eagle_pos.shape == (req_shape,):
        - check if passed array parameter eagle_pos has a length of req_shape(4)

        vel = np.array([3*eagle_pos[0], 2*eagle_pos[1], eagle_pos[2]])
            - compute for the velocity 3*2, 2*1, 3 --> [6, 2, 3]

        accel = np.array([2*vel[0], vel[1]])
            - compute for the acceleration 2*6, 2 --> [12, 2]

        total_pos = np.sum(np.multiply(eagle_pos, total_vect))
            - multiply eagle_pos with total_vect: 2*8, 1*4, 3*2, 2*1 --> [16, 4, 6, 2]
            - sum all values of the result: 16+4+6+2 = 28
              total position = summation of (position x total vector)

        total_vel = np.sum(np.multiply(vel, total_vect[1:]))
            - multiply vel with total_vect(starting from index 1): 6*4, 2*2, 3*1 --> [24, 4, 3]
            - sum all values of the result: 24+4+3 = 31
              total velocity = summation of (velocity x total vector)

        total_accel = np.sum(np.multiply(accel, total_vect[2:]))
            - multiple accel with total_vect(starting from index 2): 12*2, 2*1 --> [24, 2]
            - sum all values of the result: 24+2 = 26
              total acceleration = summation of (acceleration x total vector)
    else:
        print(f'Input displacement vector is not valid. Make sure that the vector shape is ({req_shape},)')
            - Prints an error if the length of eagle_pos is not equal to req_shape(4)

    return total_pos, total_vel, total_accel
        - Return values for total_pos, total_vel and total_accel

eagle_position = np.array([2,1,3,2])

    - Eagle position

time = 2

    - Time it took for the flight

eagle_kinematics(eagle_position, time)

    - Call function eagle_kinematics
```

Part II : Bebang’s Online Business

1. What is the relationship between the FB post reach and the profit?

    It foresees the development of profit as it is shown on the social media of Facebook as the post reach different people or became viral. Thus, it is directly proportional as it shown on the efficiency of the post in the profit of Bebang per week.

2. Why is profit in the y-axis and FB post reach on the x-axis and not the other way around?

    The outcome of the graph will not be understandable and will not see the relationship of the profit and FB Post Reach Increment. Also, since the FB post will only continuously increase while profit is not yet defined if it will grow continuously, the graphing will not be consistent and will not give an outcome that is logical and reasonable as based on what it will print.

3. Kindly explain the process of plotting the post efficiency vectors.

    The same as plotting in exercise one which is the eagle plotting but in here, there are five to be plotted. When the computation above is processed, it will not process the plotting in which it is calling the array of reach and profit in a given week. On the week one plot, it started with 0, 0 then called the week one reach and its profit that week. In week two, it calls the reach and profit of week one and week two, then in week three, it added the reach of week one and two, added the profits of week one and two the called the profit and reach of week three. Next is the week for which added the reach of week one, two and three, added also the profits of week one, two and three then called the reach and profit for week four. Lastly, for the efficiency plot, it started with 0,0 then called the computed values total week reach and total week profits. The use of declaring different colors per plot will help to see the actual plot of the vectors and the plot will only run if it satisfies the condition of ‘if make_figs: plt.savefig(f'LinAlg-Lab2-Bebang Post Eff-{int(week_performance)}@{int(reach_gradient)}.png', dpi=300)’ and if not satisfied, it will print out ‘Run at End of Month to generate report. Requires 4 weeks of input for both profit and reach’.

4. Provide a necessary equation that are representative to finding the performance$_{\text{week}}$ and reach$_{\text{gradient}}$

$$perfomance_{week} = \sqrt{(weektotal_{reach})^2 + (weektotal_{profit})^2}$$

$$reach_{gradient} = \arctan\left(\frac{weektotal_{profit}}{weektotal_{reach} + \alpha}\right)$$

5. Try to alter the 'reach' and 'profit' values to display different scenarios. Save at least 3 scenarios then append and discuss the figures in your report.

Scenario one: Higher efficiency, sustained profitability

```
week1 = np.array((reach[0], profit[1]))
week2 = np.array((reach[1], profit[1]))
week3 = np.array((reach[2], profit[1]))
week4 = np.array((reach[3], profit[1]))
```

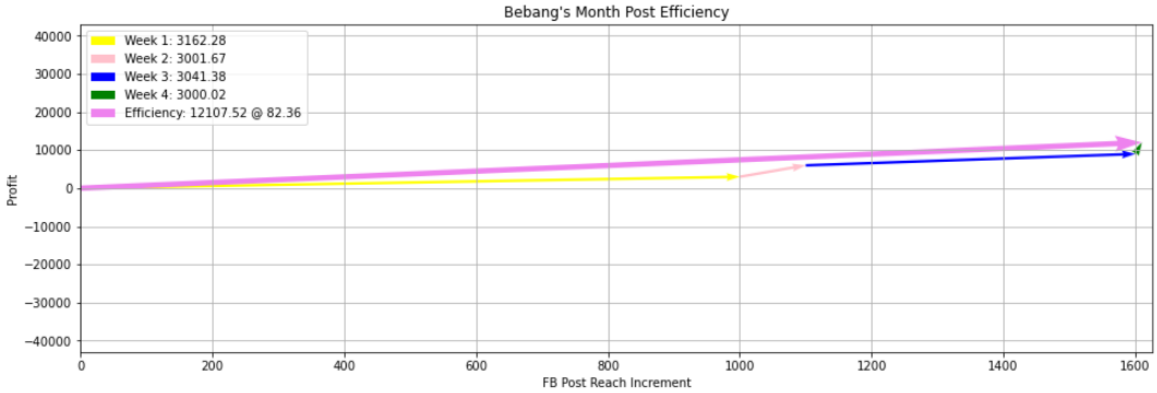


Figure 1 Exercise 3

Scenario two: Higher efficiency, sustained profitability

```
week1 = np.array((reach[0], profit[2]))
week2 = np.array((reach[1], profit[2]))
week3 = np.array((reach[2], profit[2]))
week4 = np.array((reach[3], profit[2]))
```

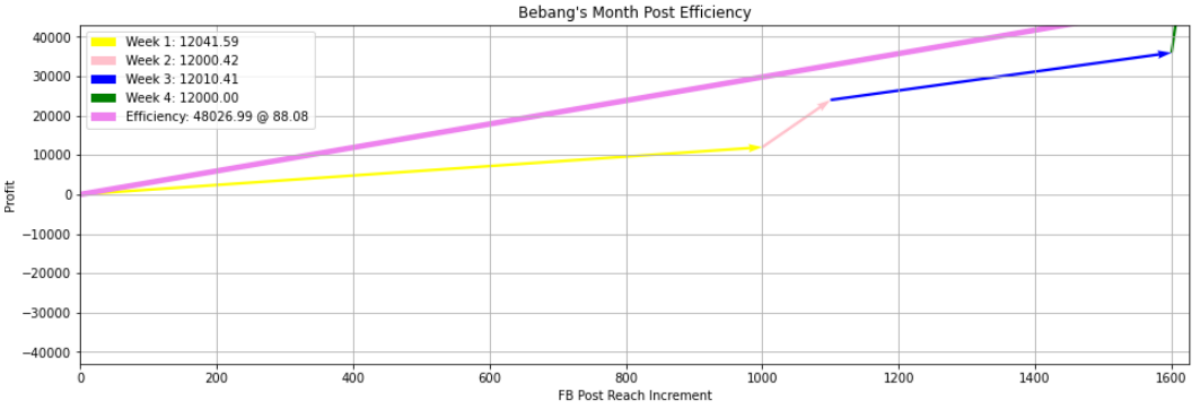


Figure 2 Exercise 3

Scenario three: Higher efficiency, sustained profitability

```
week1 = np.array((reach[0], profit[3]))
week2 = np.array((reach[1], profit[3]))
week3 = np.array((reach[2], profit[3]))
week4 = np.array((reach[3], profit[3]))
```

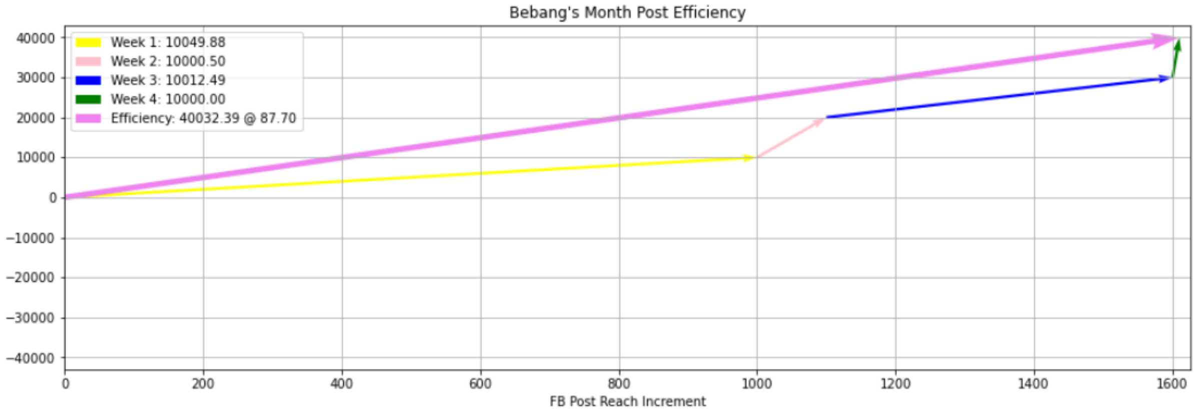


Figure 3 Exercise 3

6. What can be observed and concluded with the behavior of the vectors with respect to post efficiency?

As shown in the graph, it is profit and FB post reach is directly proportional by which, as the FB post continue to be known in the people, there will be more customers and with more customers, it means it more profit income. This shows the efficiency of FB posting and profitability since the chosen market platform is Facebook, people who seen the post will not decrease but as it continuously viral to the people then products will be sold but if the post will be kind of not known then the profitability of the product will only get slow increase.

IV. Conclusion

1. Enumerate and briefly discuss the functions you have used in the laboratory exercise, please cite their usage using their respective documentations. (min of 200 words)

Creating a vector in a programming is somewhat complex if being imagined, it takes a lot of Mathematical equations and formula in order to achieve it. Linear Algebra introduced the use of Numpy library which is used in programming to input formulas that will run the correct answers. In this laboratory, importing numpy as np is used to use different functions of numpy like adding, multiplying and different other complex operations and equations. The Matplotlib library also is helpful in order to visualize the plot that is connected to solving the vectors to see the actual output of it. Matplotlib has a function of quiver that is used in plotting arrows using the desired input inside of it in order to visualize the arrows like its scale, colors and the value of it. Also, the exercises used different functions also based on the given scenario. The functions that was needed to be defined in the exercise one in order to make the whole process is the track_eagle which is used on the first part of coding in order to declare the methods that will be done inside of it and it so it will be done the whole process of solving and plotting. In the exercise two which needed a reverse engineering, it defines the function eagle_kinematics and has values that need input which are the eagle_pos and time in order to process all the methods also after defining and so the it can do the solving part and will print out the desired outcome. Lastly, in the exercise three, it defines the function of month_profit_trace that also has values that needs input which are the profit and reach except for make_figs which declares its method, after this function, it will now also do the process and run the inputted numpy, matplot and different solving of the exercise. Overall, the function is important to make the codes run and do the desired method by calling them and their uses. Successful calling of function will have a successful running of program and output will be visualized.

2. How do vectors relate to real-life values? (min of 50 words)

Vectors are always connected to the life of the people and the world. It gives visualization to the different phenomenon in our environment or even with the activities done by the people. Life is just like a vector, it has distance which is the lifespan of the person, displacement can be considered as your achievements and success of your life and the angle are your turns, hindrances, stepping stones as you walk through your journey to get into your final outcome as a human being.

3. Kindly give other examples of how vectors are used or other real-life situations that can be modeled using vectors? (min of 100 words and do proper citation)

The use of vectors is wide and it is actually experienced in everyday activities of life. One example of this is riding a car going to your school, your starting point would be at your house. Suppose that you have a sister/brother who goes to different school that is nearer than your school, then your first point would be your sister/brother's school and suppose that your mom is going to mall then you have to drop her there then that's your next point and lastly is your school which will the last point of your ride. That's the distance you have traveled for that day but if in case you're about to be late and there's a train right in front of your subdivision and dropping off in front of your school, taking that train would be the fastest ride you can get to your school and that will serve as your displacement from house to school. This scenario is happening all the time in many people's life that is why vectors are really part of life which may not be realize but it does happen all the time.

Code Link

https://github.com/cherrylyncanoza/Linear-Algebra

References

[1] D.J.D. Lopez. "Adamson University Computer Engineering Department Honor Code," AdU-CpE Departmental Policies, 2020.