Adamson University
College of Engineering
Computer Engineering Department

Laboratory Activity No. # 4

VECTOR OPERATIONS

(Supplementary Activity)

Name: Manlulu, Emmanuel L.
Schedule: M 7:00 – 10:00
Submitted to: Engr. Maria Rizette Sayo

Objective

1. Recall knowledge on vector operations while being familiar with new operations such as products.
2. Solve vector operations using the given values.
3. Perform vector operations using Python.

Algorithm

1. Type the main title of this activity as "Vector Operations and Its Applications"
2. On your GitHub, create a repository name Linear Algebra 58013
3. On your Colab, name your activity as Python Exercise 4.ipynb and save a copy to your GitHub repository

Coding Activity 4

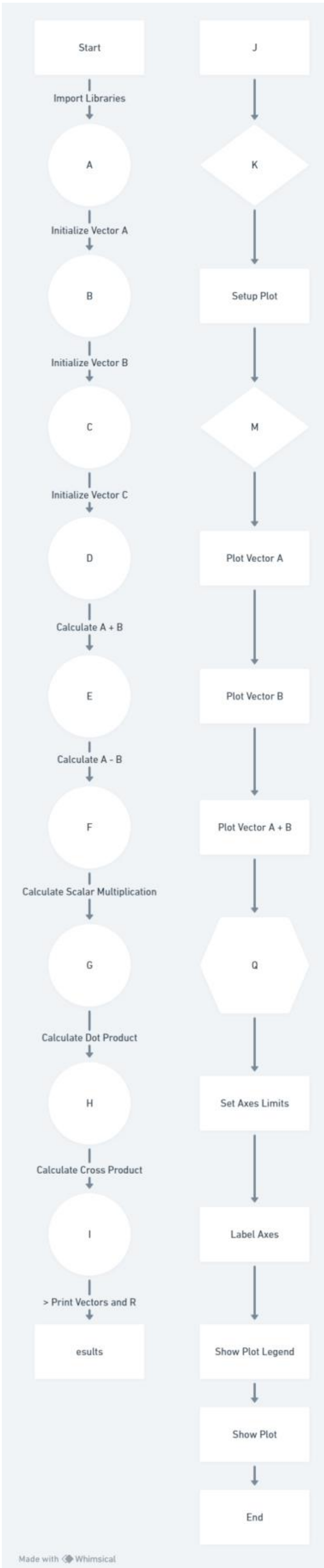
Code the following vector operation and solve them using the given vector values

$$A = \begin{bmatrix} -0.4 \\ 4.3 \\ -0.6 \end{bmatrix}, B = \begin{bmatrix} -0.20.21 \end{bmatrix}, C = \begin{bmatrix} -0.22.1 -1.5 \end{bmatrix}$$

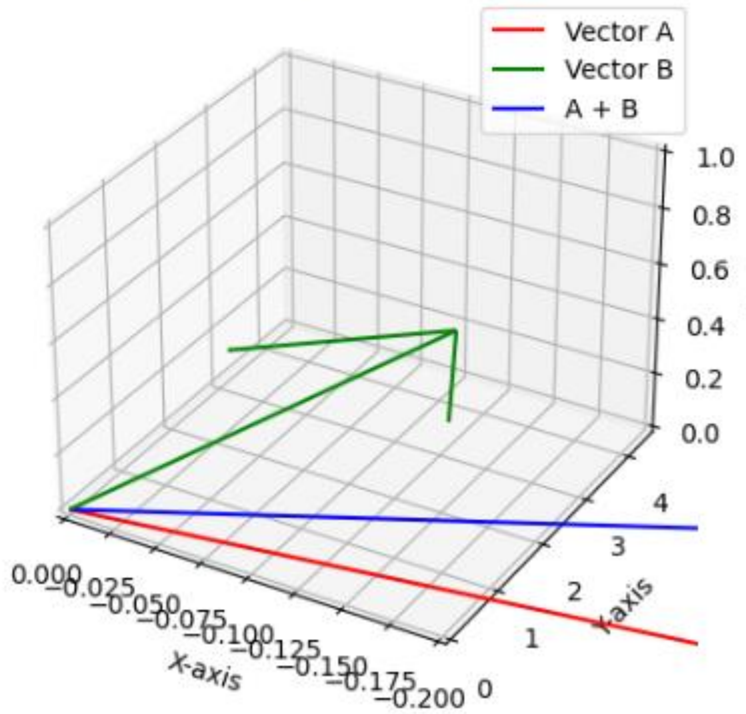
Create a program flowchart for your algorithm and explain it in your methodology. In your results, compare your answer to the expected output, visualize, and explain the resulting vector using a 3D plot.

Answer:

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

A = np.array([-0.4, 4.3, -0.6])
B = np.array([-0.2, 0.2, 1])
C = np.array([-0.2, 2.1, -1.5])

result_addition = A + B
result_subtraction = A - B
result_scalar_multiplication = 2 * A
result_dot_product = np.dot(B, C)
result_cross_product = np.cross(A, B)

print("Vector A:", A)
print("Vector B:", B)
print("Vector C:", C)

print("\nVector Addition (A + B):", result_addition)
print("Vector Subtraction (A - B):", result_subtraction)
print("Scalar Multiplication (2 * A):", result_scalar_multiplication)
print("Dot Product (B dot C):", result_dot_product)
print("Cross Product (A cross B):", result_cross_product)

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')


ax.quiver(0, 0, 0, A[0], A[1], A[2], color='r', label='Vector A')
ax.quiver(0, 0, 0, B[0], B[1], B[2], color='g', label='Vector B')
ax.quiver(0, 0, 0, result_addition[0], result_addition[1],
result_addition[2], color='b', label='A + B')

ax.set_xlim([0, max(A[0], B[0], result_addition[0])])
ax.set_ylim([0, max(A[1], B[1], result_addition[1])])
ax.set_zlim([0, max(A[2], B[2], result_addition[2])])

ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.set_zlabel('Z-axis')
ax.legend()

plt.show()
```


Start
Import Libraries
A
Initialize Vector A
B
Initialize Vector B
C
Initialize Vector C
D
Calculate A + B
E
Calculate A - B
F
Calculate Scalar Multiplication
G
Calculate Dot Product
H
Calculate Cross Product
I
> Print Vectors and R
esults
J
K
Setup Plot
M
Plot Vector A
Plot Vector B
Plot Vector A + B
Q
Set Axes Limits
Label Axes
Show Plot Legend
Show Plot
End
Made with Whimsical

```
Vector A: [-0.4  4.3 -0.6]
Vector B: [-0.2  0.2  1. ]
Vector C: [-0.2  2.1 -1.5]

Vector Addition (A + B): [-0.6  4.5  0.4]
Vector Subtraction (A - B): [-0.2  4.1 -1.6]
Scalar Multiplication (2 * A): [-0.8  8.6 -1.2]
Dot Product (B dot C): -1.04
Cross Product (A cross B): [4.42 0.52 0.78]
```



Explanation:
The Python program uses NumPy to perform vector operations on given vectors A, B, and C. It computes the addition, subtraction, scalar multiplication, dot product, and cross product of vectors A and B. The program then prints the original vectors and the results of each operation. Additionally, it creates a 3D plot using Matplotlib to visually represent the vectors A, B, and their addition (A + B). The 3D plot serves as a graphical interpretation of vector addition in three-dimensional space, providing a clear and intuitive representation of the resulting vector. Overall, the program is designed to demonstrate fundamental vector operations and visualize their outcomes using Python.

GitHub Permalink:
https://github.com/MNLLEMM/58013-Linear-Algebra/blob/fa644efbab5ba4450cfb4bead2ab84e69377b744/Python_Exercise_4.ipynb