MATGEO Presentation: 1.10.17

Soma Harsha Vardhan Reddy
ee25btech11054,
IIT Hyderabad.

September 13, 2025

Problem Statement

Find the unit vector in the direction of the sum of the vectors,
$\mathbf{a} = 2\mathbf{i} + 2\mathbf{j} - 5\mathbf{k}$ and $\mathbf{b} = 2\mathbf{i} + \mathbf{j} + 3\mathbf{k}$

Sum of vectors

Given the vectors $\mathbf{a}$ and $\mathbf{b}$

$$\mathbf{a} = \begin{pmatrix} 2 \\ 2 \\ -5 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix} \tag{3.1}$$

$$\mathbf{P} = \mathbf{a} + \mathbf{b} \tag{3.2}$$

$$\mathbf{P} = \begin{pmatrix} 4 \\ 3 \\ -2 \end{pmatrix} \tag{3.3}$$

Formula

The formula for finding unit vector along a given vector we use

$$\mathbf{p} = \frac{\mathbf{P}}{\|\mathbf{P}\|} \tag{3.4}$$

Formula

$$\|\mathbf{P}\|^2 = P^{\mathsf{T}} P = \begin{pmatrix} 4 & 3 & -2 \end{pmatrix} \begin{pmatrix} 4 \\ 3 \\ -2 \end{pmatrix} = 29 \tag{3.5}$$

Unit vector

Using (3.4)

$$\mathbf{p} = \frac{1}{\sqrt{29}} \begin{pmatrix} 4 \\ 3 \\ -2 \end{pmatrix} \tag{3.6}$$

$$\mathbf{p} = \begin{pmatrix} \frac{4}{\sqrt{29}} \\ \frac{3}{\sqrt{29}} \\ \frac{-2}{\sqrt{29}} \end{pmatrix} \tag{3.7}$$

C code

```c
#include <stdio.h>

void generate_points(double vector[3], int n, double *points) {
    for (int i = 0; i < n; i++) {
        double t = (double)i / (n - 1); // from 0 to 1
        points[3*i + 0] = t * vector[0];
        points[3*i + 1] = t * vector[1];
        points[3*i + 2] = t * vector[2];
    }
}
```

Python code for plotting using C

```
import ctypes
import numpy as np
import matplotlib.pyplot as plt

lib = ctypes.CDLL("./vector.so")
lib.generate_points.argtypes = [
    ctypes.POINTER(ctypes.c_double),
    ctypes.c_int,
    ctypes.POINTER(ctypes.c_double)
]

def get_points(vector, n=20):
    vec = (ctypes.c_double * 3)(*vector)
    points = np.zeros((n, 3), dtype=np.float64)
    lib.generate_points(vec, n, points.ctypes.data_as(ctypes.POINTER(
        ctypes.c_double)))
    return points
```

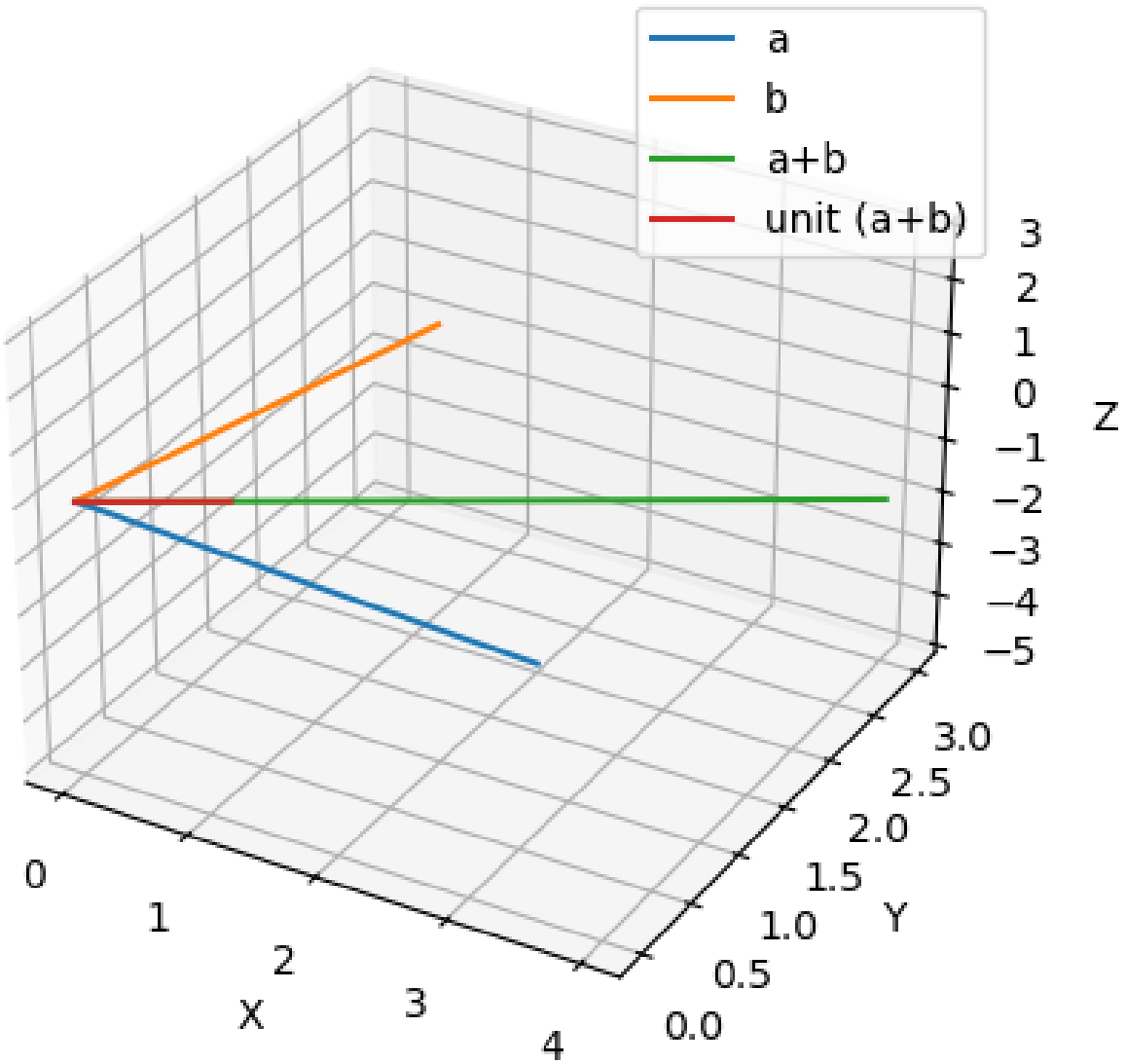
Python code for plotting using C

```
a = np.array([2, 2, −5], dtype=np.float64)
b = np.array([2, 1, 3], dtype=np.float64)
s = a + b
unit_s = s / np.linalg.norm(s)



pa = get_points(a)
pb = get_points(b)
ps = get_points(s)
pu = get_points(unit_s)



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

Python code for plotting using C

```
ax.plot(pa[:,0], pa[:,1], pa[:,2], label='a')
ax.plot(pb[:,0], pb[:,1], pb[:,2], label='b')
ax.plot(ps[:,0], ps[:,1], ps[:,2], label='a+b')
ax.plot(pu[:,0], pu[:,1], pu[:,2], label='unit (a+b)')

ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.legend()
plt.savefig("../figs/plot.png")
plt.show()
```

Plot



Pure Python Code for Plotting

```
import numpy as np
import matplotlib.pyplot as plt


a = np.array([2, 2, -5])
b = np.array([2, 1, 3])


c = a + b


c_norm = np.linalg.norm(c)
unit_c = c / c_norm
```

Pure Python Code for Plotting

```
print("Vector a:", a)
print("Vector b:", b)
print("a + b:", c)
print("Unit vector in direction of (a+b):", unit_c)


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


origin = np.zeros(3)
```

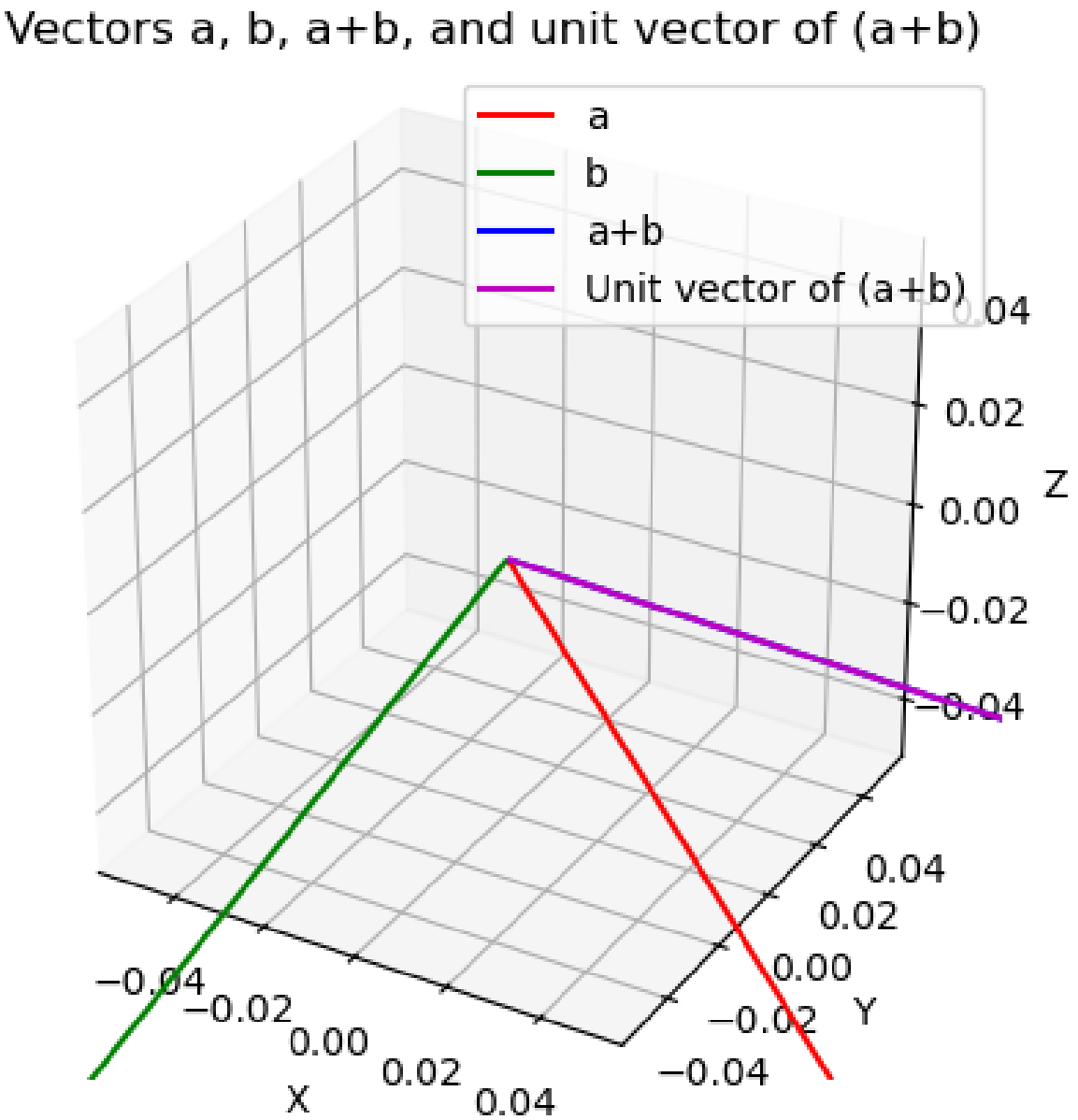
Pure Python Code for Plotting

```
ax.quiver(*origin, *a, color='r', label='a', arrow_length_ratio=0.1)
ax.quiver(*origin, *b, color='g', label='b', arrow_length_ratio=0.1)
ax.quiver(*origin, *c, color='b', label='a+b', arrow_length_ratio=0.1)
ax.quiver(*origin, *unit_c, color='m', label='Unit vector of (a+b)',
    arrow_length_ratio=0.2)


ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_title('Vectors a, b, a+b, and unit vector of (a+b)')
ax.legend()
```

Pure Python Code for Plotting

```
ax.set_box_aspect([1,1,1])
plt.savefig("../figs/plot2.png")
plt.show()
```

Plot


Vectors a, b, a+b, and unit vector of (a+b)
a
b
a+b
Unit vector of (a+b)
0.04
0.02
0.00
−0.02
−0.04
Z
−0.04
−0.02
0.00
0.02
0.04
X
0.04
0.02
0.00
−0.02
−0.04
Y