2.10.60

Sai Sreevallabh - EE25BTECH11031

September 28, 2025

Question

Find the ratio in which the Y-axis divides the line segment joining the points $(5, -6)$ and $(-1, -4)$. Also find the point of intersection.

1. $\mathbf{i} - 3\mathbf{j} + 3\mathbf{k}$
2. $-3\mathbf{i} - 3\mathbf{j} - \mathbf{k}$
3. $3\mathbf{i} - \mathbf{j} + 3\mathbf{k}$
4. $\mathbf{i} + 3\mathbf{j} - 3\mathbf{k}$

Theoretical Solution

Given vectors:

$$\mathbf{a} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} , \mathbf{b} = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} , \mathbf{c} = \begin{pmatrix} 1 \\ -1 \\ -1 \end{pmatrix} \tag{1}$$

Given that $\mathbf{v}$ is in the plane of $\mathbf{a}$ and $\mathbf{b}$, we can represent it as

$$\mathbf{v} = \alpha\mathbf{a} + \beta\mathbf{b} \tag{2}$$

Theoretical Solution

Given that the projection of $\mathbf{v}$ on $\mathbf{c}$ is $\frac{1}{\sqrt{3}}$,

$$\frac{\mathbf{v}^\top \mathbf{c}}{\|\mathbf{c}\|} = \frac{1}{\sqrt{3}} \tag{3}$$

$\because \|c\| = \sqrt{3}$

$$\mathbf{v}^\top \mathbf{c} = 1 \tag{4}$$

$$\alpha \mathbf{a}^\top \mathbf{c} + \beta \mathbf{b}^\top \mathbf{c} = 1 \tag{5}$$

Theoretical Solution

Substituting the values of **a**, **b** and **c**, we get

$$\beta - \alpha = 1 \tag{6}$$
$$\beta = \alpha + 1 \tag{7}$$

Consequently,

$$\mathbf{v} = \alpha\mathbf{a} + (\alpha + 1)\,\mathbf{b} \tag{8}$$
$$\mathbf{v} = \alpha \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + (\alpha + 1) \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} \tag{9}$$
$$\mathbf{v} = \begin{pmatrix} 2\alpha + 1 \\ -1 \\ 2\alpha + 1 \end{pmatrix} \tag{10}$$

Theoretical Solution

This is the general expression for the vector $\mathbf{v}$. Out of the given options, only option 3 i.e. $3\mathbf{i} - \mathbf{j} + 3\mathbf{k}$ satisfies the general expression (with $\alpha = 1$).
$\therefore \mathbf{v} = 3\mathbf{i} - \mathbf{j} + 3\mathbf{k}$

C Code - Finding the Plane

```c
#include <stdio.h>

void plane_from_vectors(double a[3], double b[3], double
    plane[4]) {

   double nx = a[1]*b[2] - a[2]*b[1];
   double ny = a[2]*b[0] - a[0]*b[2];
   double nz = a[0]*b[1] - a[1]*b[0];

   plane[0] = nx;
   plane[1] = ny;
   plane[2] = nz;
   plane[3] = 0.0;
}
```

Python Code - Using Shared Object

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


c_lib = ctypes.CDLL("./code.so")


c_lib.plane_from_vectors.argtypes = [ctypes.c_double*3,
                                     ctypes.c_double*3,
                                     ctypes.c_double*4]


a = (ctypes.c_double*3)(1.0, 1.0, 1.0)
b = (ctypes.c_double*3)(1.0, -1.0, 1.0)
plane = (ctypes.c_double*4)(0.0, 0.0, 0.0, 0.0)
```

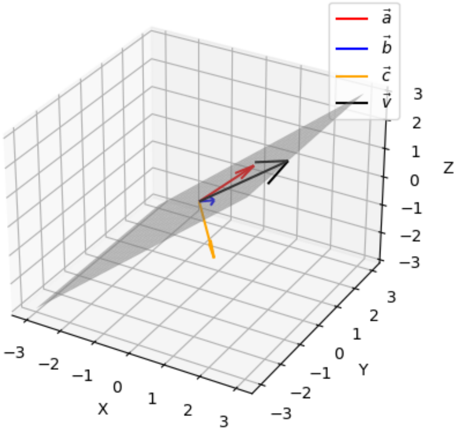
Python Code - Using Shared Object

```
c_lib.plane_from_vectors(a,b,plane)

c = np.array([1.0,-1.0,-1.0])
v = np.array([3.0, -1.0, 3.0])

A, B, C, D = plane
xx, yy = np.meshgrid(np.linspace(-3, 3, 20), np.linspace
    (-3, 3, 20))

zz = (-A*xx - B*yy - D)/C

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot_surface(xx, yy, zz, alpha=0.5, color='grey')
```

Python Code - Using Shared Object

```
ax.quiver(0,0,0, a[0],a[1],a[2], color="red", label=r"$\vec{a}$")
ax.quiver(0,0,0, b[0],b[1],b[2], color="blue", label=r"$\vec{b}$")
ax.quiver(0,0,0, c[0],c[1],c[2], color="orange", label=r"$\vec{c}$")
ax.quiver(0,0,0, v[0],v[1],v[2], color="black", label=r"$\vec{v}$")

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

Plot-Using Both C and Python



Python Code

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

u = np.array([1, 1, 1])
v = np.array([1, -1, 1])
w = np.array([3, -1, 3])
x = np.array([1, -1, -1])

uv = v - u
uw = w - u
normal = np.cross(uv, uw)
a, b, c = normal
d = 0
```

Python Code

```
xx, yy = np.meshgrid(
    np.linspace(-3, 3, 10),
    np.linspace(-3, 3, 10)
)
zz = (-a * xx - b * yy - d) / c

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

ax.plot_surface(xx, yy, zz, alpha=0.5, color='grey')
```

Python Code

```
origin = np.zeros(3)

for vec, color, label in zip([u, v, w, x], ['r', 'g', 'b'
    , 'orange'], [r'$\vec{a}$', r'$\vec{b}$', r'$\vec{v}$
    ', r'$\vec{c}$']):
    ax.quiver(*origin, *vec, color=color, label=label)


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

Plot-Using Python only