2.10.15

RATHLAVATH JEEVAN -AI25BTECH11026

September 16, 2025

Question

The number of vectors of unit length perpendicular to vectors

$$\mathbf{a} = (1, 1, 0) \quad \text{and} \quad \mathbf{b} = (0, 1, 1) \tag{1}$$

is

$$\text{(a) one} \quad \text{(b) two} \quad \text{(c) three} \quad \text{(d) infinite} \quad \text{(e) None of these} \tag{2}$$

Theoretical Solution

Solution:
Given Let

$$\mathbf{a} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}. \tag{3}$$

A vector $\mathbf{x}$ perpendicular to both $\mathbf{a}$ and $\mathbf{b}$ satisfies

$$\mathbf{a}^\top \mathbf{x} = 0 \quad \Rightarrow \quad x_1 + x_2 = 0, \tag{4}$$

$$\mathbf{b}^\top \mathbf{x} = 0 \quad \Rightarrow \quad x_2 + x_3 = 0. \tag{5}$$

Theoretical Solution

Solution:

$$x_1 = -x_2, \qquad x_3 = -x_2 \;\Rightarrow\; \mathbf{x} = x_2 \begin{pmatrix} -1 \\ 1 \\ -1 \end{pmatrix}. \tag{6}$$

Thus a direction vector is

$$\mathbf{n} = \begin{pmatrix} -1 \\ 1 \\ -1 \end{pmatrix}, \qquad \|\mathbf{n}\| = \sqrt{3}. \tag{7}$$

Hence the *unit* vectors perpendicular to both **a** and **b** are

$$\mathbf{u} = \pm \frac{1}{\sqrt{3}} \begin{pmatrix} -1 \\ 1 \\ -1 \end{pmatrix}. \tag{8}$$

Therefore, the number of such unit vectors is $\boxed{2}$.

C Code

```c
#include <stdio.h>
#include <math.h>

typedef struct { double x, y, z; } Vec;

Vec cross(Vec a, Vec b) {
    Vec c = {
        a.y*b.z - a.z*b.y,
        a.z*b.x - a.x*b.z,
        a.x*b.y - a.y*b.x
    };
    return c;
}

double norm(Vec v) {
    return sqrt(v.x*v.x + v.y*v.y + v.z*v.z);
}
```

C Code

```c
Vec scale(Vec v, double s) {
    Vec r = { v.x * s, v.y * s, v.z * s };
    return r;
}

int main(void) {
    // Given vectors
    Vec a = {1, 1, 0};
    Vec b = {0, 1, 1};

    // Vector perpendicular to both is a * b
    Vec n = cross(a, b);
    double m = norm(n);

    // If cross product is zero, vectors are parallel ->
        infinitely many unit normals
    const double EPS = 1e-12;
```

C Code

```c
    if (m < EPS) {
        printf(Number of unit vectors perpendicular to both:
            infinite\n);
        return 0;
        }

    // Two unit vectors:+- (a * b) / ||a * b||
    Vec u = scale(n, 1.0 / m);
    Vec v = scale(u, -1.0);

    printf(Number of unit vectors perpendicular to both: 2\n);
    printf(u1 = (%.6f, %.6f, %.6f)\n, u.x, u.y, u.z);
    printf(u2 = (%.6f, %.6f, %.6f)\n, v.x, v.y, v.z);

    return 0;
}
```

Python Code

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

# Given vectors
a = np.array([1, 1, 0])
b = np.array([0, 1, 1])

# Cross product gives a vector perpendicular to both
v = np.cross(a, b)
v = v / np.linalg.norm(v) # Unit vector

# The two perpendicular unit vectors are +-v
v1 = v
v2 = -v
```

Python Code

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

# Plot given vectors
ax.quiver(0, 0, 0, a[0], a[1], a[2], color='r', label='a = (1,1,0)')
ax.quiver(0, 0, 0, b[0], b[1], b[2], color='g', label='b = (0,1,1)')

# Plot perpendicular unit vectors
ax.quiver(0, 0, 0, v1[0], v1[1], v1[2], color='b', label='Unit perp vector +v')
ax.quiver(0, 0, 0, v2[0], v2[1], v2[2], color='orange', label='Unit perp vector -v')
```

Python Code

```python
# Set labels
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_title('Unit vectors perpendicular to a and b')
ax.legend()

# Save image
plt.savefig(perpendicular_vectors.png)
plt.show()
```

Plot

beamer/figs/matg5.jpeg