2.10.73

Navya Priya - EE25BTECH11045

October 1,2025

Question

Let $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ be unit vectors. Suppose that $\mathbf{A} \cdot \mathbf{B} = \mathbf{A} \cdot \mathbf{C} = 0$, and that the angle between $\mathbf{B}$ and $\mathbf{C}$ is $\frac{\pi}{6}$. Then $\mathbf{A} = \pm 2(\mathbf{B} \times \mathbf{C})$

Theoretical Solution

Let us solve the given equation theoretically and then verify the solution computationally.

Since $\mathbf{A} \cdot \mathbf{B} = \mathbf{A} \cdot \mathbf{C} = 0$, it follows that $\mathbf{A}$ is perpendicular to both $\mathbf{B}$ and $\mathbf{C}$. Therefore A is parallel(or anti-parallel) to the cross product of $\mathbf{B}$ and $\mathbf{C}$.

$$\mathbf{A} = \lambda(\mathbf{B} \times \mathbf{C}) \tag{1}$$

From the given question,

$$\mathbf{B}^\top \mathbf{C} = \cos\left(\frac{\pi}{6}\right) \tag{2}$$

We know that,

$$\left(\mathbf{B}^\top \mathbf{C}\right)^2 + ||\mathbf{B} \times \mathbf{C}||^2 = ||\mathbf{B}||^2 ||\mathbf{C}||^2 \tag{3}$$

Theoretical Solution

$$\implies ||\mathbf{B} \times \mathbf{C}||^2 = \frac{1}{4} \tag{4}$$

$$\implies ||\mathbf{B} \times \mathbf{C}|| = \frac{1}{2} \tag{5}$$

As **A** is a unit vector, from(1)

$$||\mathbf{A}|| = ||\lambda(\mathbf{B} \times \mathbf{C})|| \tag{6}$$

$$1 = |\lambda|\frac{1}{2} \tag{7}$$

Hence

$$\lambda = \pm 2 \tag{8}$$

$$\therefore \mathbf{A} = \pm 2(\mathbf{B} \times \mathbf{C}) \tag{9}$$

C code

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

int main() {
    // Define B and C as given
    double B[3] = {0.5, sqrt(3)/2, 0};
    double C[3] = {0.5, -sqrt(3)/2, 0};
    double cross[3], A1[3], A2[3];

    // Cross product B x C
    cross[0] = B[1]*C[2] - B[2]*C[1];
    cross[1] = B[2]*C[0] - B[0]*C[2];
    cross[2] = B[0]*C[1] - B[1]*C[0];

    // A = 2(B x C)
    for (int i=0; i<3; i++) {
        A1[i] = 2 * cross[i];
        A2[i] = -2 * cross[i];
    }
```

C code

```
    // Print results
    printf("Vector B = (%.2f, %.2f, %.2f)\n", B[0], B[1], B[2]);
    printf("Vector C = (%.2f, %.2f, %.2f)\n", C[0], C[1], C[2]);
    printf("Cross Product (B x C) = (%.2f, %.2f, %.2f)\n", cross
        [0], cross[1], cross[2]);
    printf("A1 = +2(B x C) = (%.2f, %.2f, %.2f)\n", A1[0], A1[1],
         A1[2]);
    printf("A2 = -2(B x C) = (%.2f, %.2f, %.2f)\n", A2[0], A2[1],
         A2[2]);

    return 0;
}
```

CallC.py

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

# Load shared library
lib = ctypes.CDLL("./vectors.so") # use "vectors.dll" on Windows

# Define argument and return types
lib.compute_vectors.argtypes = [
    np.ctypeslib.ndpointer(dtype=np.float64, ndim=1, flags="
        C_CONTIGUOUS"),
    np.ctypeslib.ndpointer(dtype=np.float64, ndim=1, flags="
        C_CONTIGUOUS"),
    np.ctypeslib.ndpointer(dtype=np.float64, ndim=1, flags="
        C_CONTIGUOUS"),
    np.ctypeslib.ndpointer(dtype=np.float64, ndim=1, flags="
        C_CONTIGUOUS"),
]
lib.compute_vectors.restype = None
```

```
# Input vectors
B = np.array([0.5, np.sqrt(3)/2, 0.0], dtype=np.float64)
C = np.array([0.5, -np.sqrt(3)/2, 0.0], dtype=np.float64)

# Output arrays
A1 = np.zeros(3, dtype=np.float64)
A2 = np.zeros(3, dtype=np.float64)

# Call C function
lib.compute_vectors(B, C, A1, A2)

# --- Plot ---
fig = plt.figure()
ax = fig.add_subplot(111, projection="3d")
```

```
vectors = {"B": B, "C": C, "A1": A1, "A2": A2}
colors = {"B": "r", "C": "g", "A1": "b", "A2": "m"}

for name, vec in vectors.items():
    ax.quiver(0,0,0, vec[0], vec[1], vec[2], color=colors[name],
        label=name)
    ax.text(vec[0], vec[1], vec[2], f"{name}{tuple(vec.round(2))}
        ")

ax.set_xlabel("X")
ax.set_ylabel("Y")
ax.set_zlabel("Z")
ax.set_title("Vectors from C + Python (ctypes)")
ax.legend()
plt.show()
```

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

# Define vectors
B = np.array([1, 0, 0])
C = np.array([np.cos(np.pi/6), np.sin(np.pi/6), 0])
cross_BC = np.cross(B, C)
A1 = 2 * cross_BC
A2 = -2 * cross_BC

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

# Function to draw vectors
def draw_vector(ax, vec, color, label):
    ax.quiver(0, 0, 0, vec[0], vec[1], vec[2], color=color, label
        =label)
```

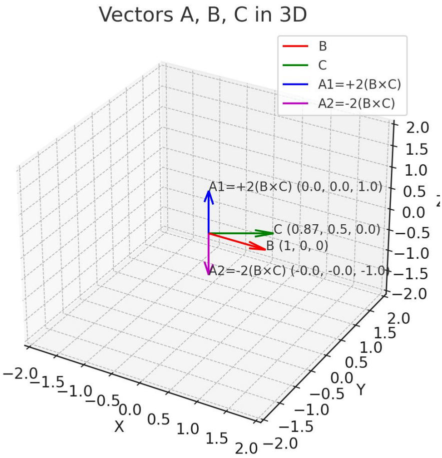
```
# Draw vectors
draw_vector(ax, B, 'r', 'B')
draw_vector(ax, C, 'g', 'C')
draw_vector(ax, A1, 'b', 'A = +2(BC)')
draw_vector(ax, A2, 'm', 'A = -2(BC)')

# Axes settings
ax.set_xlim([-2, 2])
ax.set_ylim([-2, 2])
ax.set_zlim([-2, 2])
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.legend()
ax.set_title("Vectors A, B, C in 3D")

plt.show()
```

Plot

To verify the solution computationally let us assume the vectors $\mathbf{B}$ and $\mathbf{C}$ as

$$\mathbf{B} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \text{ and } \mathbf{C} = \begin{pmatrix} \frac{\sqrt{3}}{2} \\ \frac{1}{2} \\ 0 \end{pmatrix}$$

Plot


Vectors A, B, C in 3D
B
C
A1=+2(B×C)
A2=-2(B×C)
A1=+2(B×C) (0.0, 0.0, 1.0)
C (0.87, 0.5, 0.0)
B (1, 0, 0)
A2=-2(B×C) (-0.0, -0.0, -1.0)
X
Y