1.11.5

AI25BTECH11003 - Bhavesh Gaikwad

August 26,2025

Question

The scalar product of vector $\overrightarrow{a} = \hat{i} + \hat{j} + \hat{k}$ with a unit vector along the sum of the vectors $\overrightarrow{b} = 2\hat{i} + 4\hat{j} - 5\hat{k}$ and $\overrightarrow{c} = \lambda\hat{i} + 2\hat{j} + 3\hat{k}$ is equal to 1. Find the value of λ and hence find the unit vector along $\overrightarrow{b} + \overrightarrow{c}$.

Theoretical Solution

Given: $\mathbf{a} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 2 \\ 4 \\ -5 \end{pmatrix}, \mathbf{c} = \begin{pmatrix} \lambda \\ 2 \\ 3 \end{pmatrix}.$

Let $\mathbf{u}$ be the unit vector along $\mathbf{b} + \mathbf{c}$.

$$\mathbf{b} + \mathbf{c} = \begin{pmatrix} 2+\lambda \\ 4+2 \\ -5+3 \end{pmatrix} = \begin{pmatrix} 2+\lambda \\ 6 \\ -2 \end{pmatrix}.$$

$$\|\mathbf{b} + \mathbf{c}\| = \sqrt{(2+\lambda)^2 + 6^2 + (-2)^2} = \sqrt{\lambda^2 + 4\lambda + 44}.$$

$$\mathbf{u} = \frac{\mathbf{b} + \mathbf{c}}{\|\mathbf{b} + \mathbf{c}\|} = \frac{1}{\sqrt{\lambda^2 + 4\lambda + 44}} \begin{pmatrix} 2+\lambda \\ 6 \\ -2 \end{pmatrix}.$$

Theoretical Solution

Given condition: $\mathbf{a} \cdot \mathbf{u} = 1$.

$$\mathbf{a} \cdot \mathbf{u} = \frac{\mathbf{a} \cdot (\mathbf{b} + \mathbf{c})}{\|\mathbf{b} + \mathbf{c}\|} = \frac{\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 2 + \lambda \\ 6 \\ -2 \end{pmatrix}}{\sqrt{\lambda^2 + 4\lambda + 44}} = \frac{\lambda + 6}{\sqrt{\lambda^2 + 4\lambda + 44}} = 1.$$

$$\Rightarrow \; (\lambda+6)^2 = \lambda^2+4\lambda+44 \implies \lambda^2+12\lambda+36 = \lambda^2+4\lambda+44 \implies 8\lambda = 8$$

$$\implies \boxed{\lambda = 1}$$

Now, with

$$\lambda = 1: \quad \mathbf{b} + \mathbf{c} = \begin{pmatrix} 3 \\ 6 \\ -2 \end{pmatrix}, \quad \|\mathbf{b} + \mathbf{c}\| = \sqrt{3^2 + 6^2 + (-2)^2} = \sqrt{49} = 7.$$

Theoretical Solution

Unit vector along, $\mathbf{b}+\mathbf{c}$ is: $$\frac{1}{7}\begin{pmatrix}3\\6\\-2\end{pmatrix}=\frac{1}{7}\begin{pmatrix}3\\6\\-2\end{pmatrix}.$$

$$\boxed{\lambda=1} \quad \textit{and} \quad \boxed{\text{Unit vector along } \mathbf{b}+\mathbf{c}=\frac{1}{7}\begin{pmatrix}3\\6\\-2\end{pmatrix}}. \tag{1}$$

C Code

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

#define DIM 3


void add(const double u[DIM], const double v[DIM], double out[DIM
    ]) {
    for (int i = 0; i < DIM; ++i) out[i] = u[i] + v[i];
}

double dot(const double u[DIM], const double v[DIM]) {
    double s = 0.0;
    for (int i = 0; i < DIM; ++i) s += u[i] * v[i];
    return s;
}
```

C Code

```c
double norm(const double u[DIM]) {
    return sqrt(dot(u, u));
}

void scale(const double u[DIM], double s, double out[DIM]) {
    for (int i = 0; i < DIM; ++i) out[i] = s * u[i];
}

void normalize(const double u[DIM], double out[DIM]) {
    double n = norm(u);
    if (n == 0.0) {

        for (int i = 0; i < DIM; ++i) out[i] = 0.0;
    } else {
        scale(u, 1.0 / n, out);
    }
}
```

C Code

```c
int main(void) {

    const double a[DIM] = {1.0, 1.0, 1.0};
    const double b[DIM] = {2.0, 4.0, -5.0};


    const double lambda = 1.0;
    const double c[DIM] = {lambda, 2.0, 3.0};


    double s[DIM];
    add(b, c, s); // s = (3, 6, -2)


    double uhat[DIM];
    normalize(s, uhat); // uhat = (3/7, 6/7, -2/7)
```

C Code

```c

    printf("lambda = %.0f\n", lambda);
    printf("b + c = (%.2f, %.2f, %.2f)\n", s[0], s[1], s[2]);
    printf("||b + c|| = %.2f\n", norm(s));
    printf("Unit vector along (b + c) is: (%.2f, %.2f, %.2f)\n",
        uhat[0], uhat[1], uhat[2]);


    double check = dot(a, uhat);
    printf("Verification a * u = %.2f\n", check);

    return 0;
}
```

Python Code

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

# Define vectors
a = np.array([1, 1, 1])
b = np.array([2, 4, -5])
lambda_val = 1
c = np.array([lambda_val, 2, 3])

# b + c and its unit vector
bc = b + c
bc_unit = bc / np.linalg.norm(bc)
```

Python Code

```
# Set up 3D plot
fig = plt.figure(figsize=(7, 7))
ax = fig.add_subplot(111, projection='3d')
ax.set_xlim([-1, 7])
ax.set_ylim([-1, 7])
ax.set_zlim([-6, 4])

# Origin
origin = np.zeros(3)

def plot_vec(ax, v, color, label):
    ax.quiver(*origin, *v, color=color, arrow_length_ratio=0.1,
        linewidth=2)
    ax.text(*(v*1.12), label, color=color, fontsize=13)
```

Python Code

```
plot_vec(ax, a, 'blue', 'a')
plot_vec(ax, b, 'orange', 'b')
plot_vec(ax, c, 'green', 'c')
plot_vec(ax, bc_unit, 'red', '(b+c)/|b+c|')

# Labels and style
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_title('Vectors a, b, c and unit vector along (b+c)')
plt.tight_layout()

# Save the figure
plt.savefig('fig1.png')
plt.close()
```

Vector Representation