2.7.3

Shivam Sawarkar
AI25BTECH11031

September 13, 2025

Question (2.7.3)

If $\mathbf{a}$ and $\mathbf{b}$ are two vectors such that $\mathbf{a} = \hat{i} - \hat{j} + \hat{k}$, $\mathbf{b} = 2\hat{i} - \hat{j} - 3\hat{k}$, then find the vector $\mathbf{c}$, given that $\mathbf{a} \times \mathbf{c} = \mathbf{b}$, $\mathbf{a} \cdot \mathbf{c} = 4$.

Solution

$$\mathbf{a} = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} 2 \\ -1 \\ -3 \end{pmatrix} \quad \mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} \tag{1}$$

$$\mathbf{a} \times \mathbf{c} = \mathbf{b} \tag{2}$$

$$\begin{pmatrix} -c_2 - c_3 \\ c_1 - c_3 \\ c_1 + c_2 \end{pmatrix} = \begin{pmatrix} 2 \\ -1 \\ -3 \end{pmatrix} \tag{3}$$

$$\implies \begin{cases} -c_2 - c_3 = 2, \\ c_1 - c_3 = -1, \\ c_1 + c_2 = -3. \end{cases} \tag{4}$$

Solution

From the second and third equations:

$$c_1 = c_3 - 1, \qquad c_2 = -2 - c_3. \tag{5}$$

$$\implies \mathbf{c} = \begin{pmatrix} c_3 - 1 \\ -2 - c_3 \\ c_3 \end{pmatrix}, \quad c_3 \in \mathbb{R}. \tag{6}$$

Solution

Now apply the dot product condition:

$$\mathbf{a}^T\mathbf{c} = \begin{pmatrix} 1 & -1 & 1 \end{pmatrix} \begin{pmatrix} c_3 - 1 \\ -2 - c_3 \\ c_3 \end{pmatrix} = 3c_3 + 1. \tag{7}$$

$$3c_3 + 1 = 4 \quad \implies \quad c_3 = 1. \tag{8}$$

$$\therefore \quad \mathbf{c} = \begin{pmatrix} 0 \\ -3 \\ 1 \end{pmatrix} \tag{9}$$

Plot

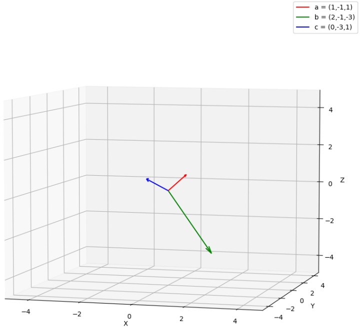

Figure:

Python Code

```
import numpy as np

# Input vectors
a = np.array(list(map(int, input("Enter vector a (3 integers
separated by space): ").split())))
b = np.array(list(map(int, input("Enter vector b (3 integers
separated by space): ").split())))
dot_val = int(input("Enter value of a · c: "))

# Coefficient matrix (augmented form)
# Let c = (x,y,z)
# Equations:
#   a2*z - a3*y = b1
#   a3*x - a1*z = b2
#   a1*y - a2*x = b3
#   a1*x + a2*y + a3*z = dot_val
```

Python Code

```
A = np.array([
    [0, -a[2], a[1]],      # from a2*z - a3*y = b1
    [a[2], 0, -a[0]],      # from a3*x - a1*z = b2
    [-a[1], a[0], 0],      # from a1*y - a2*x = b3
    [a[0], a[1], a[2]]     # dot product
], dtype=float)

rhs = np.array([b[0], b[1], b[2], dot_val], dtype=float)

# Solve least squares (in case overdetermined)
c, residuals, rank, s = np.linalg.lstsq(A, rhs, rcond=None)

print("Vector c =", np.round(c, 4))
```