

## 4.4.13

BALU-ai25btech11017

September 12, 2025

# Question

If  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$  are three non-coplanar vectors, then

$$\frac{\mathbf{A} \cdot (\mathbf{B} \times \mathbf{C})}{(\mathbf{C} \times \mathbf{A}) \cdot \mathbf{B}} + \frac{\mathbf{B} \cdot (\mathbf{A} \times \mathbf{C})}{\mathbf{C} \cdot (\mathbf{A} \times \mathbf{B})} = \quad (1)$$

# Theoretical Solution

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

According to the question,

Let us take three non coplanar vectors

$$\mathbf{A} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad \mathbf{C} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad (2)$$

$$\mathbf{B} \times \mathbf{C} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad \mathbf{A} \times \mathbf{C} = \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix} \quad \mathbf{C} \times \mathbf{A} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad \mathbf{A} \times \mathbf{B} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad (3)$$

$$\mathbf{A} \cdot (\mathbf{B} \times \mathbf{C}) = \mathbf{A}^T (\mathbf{B} \times \mathbf{C}) = 1 \quad (4)$$

$$(\mathbf{C} \times \mathbf{A}) \cdot \mathbf{B} = (\mathbf{C} \times \mathbf{A})^T \mathbf{B} = 1 \quad (5)$$

$$\mathbf{B} \cdot (\mathbf{A} \times \mathbf{C}) = \mathbf{B}^T (\mathbf{A} \times \mathbf{C}) = -1 \quad (6)$$

$$\mathbf{C} \cdot (\mathbf{A} \times \mathbf{B}) = \mathbf{C}^T (\mathbf{A} \times \mathbf{B}) = 1 \quad (7)$$

$$\frac{\mathbf{A} \cdot (\mathbf{B} \times \mathbf{C})}{(\mathbf{C} \times \mathbf{A}) \cdot \mathbf{B}} + \frac{\mathbf{B} \cdot (\mathbf{A} \times \mathbf{C})}{\mathbf{C} \cdot (\mathbf{A} \times \mathbf{B})} = \frac{1}{1} + \frac{-1}{1} = 1 - 1 = 0 \quad (8)$$

By verification method we showed the result is 0

```
#include <stdio.h>

// Function to compute cross product of two vectors
void crossProduct(double u[], double v[], double result[]) {
    result[0] = u[1]*v[2] - u[2]*v[1];
    result[1] = u[2]*v[0] - u[0]*v[2];
    result[2] = u[0]*v[1] - u[1]*v[0];
}

// Function to compute dot product of two vectors
double dotProduct(double u[], double v[]) {
    return u[0]*v[0] + u[1]*v[1] + u[2]*v[2];
}

int main() {
    // Define vectors A = i, B = j, C = k
    double A[3] = {1, 0, 0};
    double B[3] = {0, 1, 0};
    double C[3] = {0, 0, 1};
```

```
double BxC[3], CxA[3], AxC[3], AxB[3];
double numerator1, denominator1, numerator2, denominator2,
    result;

// Compute cross products
crossProduct(B, C, BxC);
crossProduct(C, A, CxA);
crossProduct(A, C, AxC);
crossProduct(A, B, AxB);

// Compute terms
numerator1 = dotProduct(A, BxC);
denominator1 = dotProduct(CxA, B);
numerator2 = dotProduct(B, AxC);
denominator2 = dotProduct(C, AxB);
```

```
// Final result
result = (numerator1 / denominator1) + (numerator2 /
    denominator2);

// Print results
printf("Numerator1 = %.2f, Denominator1 = %.2f\n", numerator1
    , denominator1);
printf("Numerator2 = %.2f, Denominator2 = %.2f\n", numerator2
    , denominator2);
printf("Final Result = %.2f\n", result);

return 0;
}
```