

## 2.8.9

J.NAVYASRI- EE25BTECH11028

september 2025

# Question

Let  $\mathbf{a}, \mathbf{b}, \mathbf{c}$  be three vectors such that  $|\mathbf{a}| = 3$ ,  $|\mathbf{b}| = 4$ ,  $|\mathbf{c}| = 5$ , and each one of them is perpendicular to the sum of the other two. Find  $|\mathbf{a} + \mathbf{b} + \mathbf{c}|$ .

# Solution:

From the identity:

$$a^\top(b + c) = 0, \quad (1)$$

we expand:

$$a^\top b + a^\top c = 0. \quad (2)$$

Similarly, from the symmetry of dot products:

$$b^\top c + b^\top a = 0, \quad (3)$$

$$c^\top a + c^\top b = 0. \quad (4)$$

# Solution:

Let

$$x = a^\top b, \quad y = b^\top c, \quad z = c^\top a. \quad (5)$$

Then equations (2), (3), and (4) become:

$$x + z = 0, \quad (6)$$

$$x + y = 0, \quad (7)$$

$$y + z = 0. \quad (8)$$

From equation (7):

$$y = -x,$$

# Solution:

and from equation (6):

$$z = -x.$$

Substitute into equation (8):

$$y + z = -x + (-x) = -2x = 0 \Rightarrow x = 0.$$

Therefore:

$$x = y = z = 0, \quad (9)$$

so  $a, b, c$  are **pairwise orthogonal**.

The **Gram matrix** of  $(a, b, c)$  is:

$$G = \begin{bmatrix} a^\top a & a^\top b & a^\top c \\ b^\top a & b^\top b & b^\top c \\ c^\top a & c^\top b & c^\top c \end{bmatrix} = \begin{bmatrix} \|a\|^2 & 0 & 0 \\ 0 & \|b\|^2 & 0 \\ 0 & 0 & \|c\|^2 \end{bmatrix} = \begin{pmatrix} 9 & 0 & 0 \\ 0 & 16 & 0 \\ 0 & 0 & 25 \end{pmatrix}. \quad (10)$$

# Solution:

Let

$$u = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

Then

$$\|a + b + c\|^2 = (a + b + c)^\top (a + b + c) = u^\top Gu. \quad (11)$$

Now compute:

$$\begin{aligned} u^\top Gu &= \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 9 & 0 & 0 \\ 0 & 16 & 0 \\ 0 & 0 & 25 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \\ &= 9 + 16 + 25 = 50. \end{aligned} \quad (12)$$

Therefore:

$$\|a + b + c\| = \sqrt{50} = 5\sqrt{2}. \quad (13)$$

**Final Answer:**

$$5\sqrt{2}$$

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

# Define mutually perpendicular vectors
a = np.array([3, 0, 0])
b = np.array([0, 4, 0])
c = np.array([0, 0, 5])
s = a + b + c # resultant (3,4,5)

fig = plt.figure(figsize=(8, 6))
ax = fig.add_subplot(111, projection='3d')
```

```
# Plot main vectors
ax.quiver(0, 0, 0, *a, color='r', linewidth=2,
          arrow_length_ratio=0.08, normalize=False, label='a (3)')
ax.quiver(0, 0, 0, *b, color='g', linewidth=2,
          arrow_length_ratio=0.08, normalize=False, label='b (4)')
ax.quiver(0, 0, 0, *c, color='b', linewidth=2,
          arrow_length_ratio=0.08, normalize=False, label='c (5)')

# Plot resultant
ax.quiver(0, 0, 0, *s, color='m', linewidth=2,
          arrow_length_ratio=0.05, normalize=False,
          label='a+b+c')
```



```
# Axis limits
ax.set_xlim(0, 8)
ax.set_ylim(0, 8)
ax.set_zlim(0, 8)

# Axis labels
ax.set_xlabel("X-axis")
ax.set_ylabel("Y-axis")
ax.set_zlabel("Z-axis")
ax.set_title("Mutually Perpendicular Vectors and their Resultant")

ax.legend()
plt.show()
```

# C Code

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

int main() {
    // Given magnitudes
    int a = 3, b = 4, c = 5;
    // Since a, b, c are mutually perpendicular (proved in
        solution),
    //  $|a + b + c|^2 = |a|^2 + |b|^2 + |c|^2$ 
    int sum_sq = a*a + b*b + c*c;

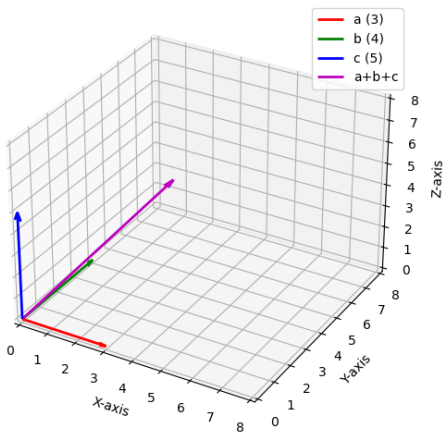
    double magnitude = sqrt(sum_sq);

    // Print result
    printf("The magnitude  $|a + b + c| = %.2f$ \n", magnitude);

    return 0;
}
```

# Plot-Using by Python

Mutually Perpendicular Vectors and their Resultant



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

# Load the compiled C library
lib = ctypes.CDLL("./vector_calc.so") # use "vector_calc.dll" on
Windows

# Call the C function
lib.vector_magnitude.restype = ctypes.c_double
magnitude = lib.vector_magnitude()
print("Result from C code |a+b+c| =", magnitude)
```

# Python and C Code

```
# ---- Plotting in Python ----
a = np.array([3, 0, 0])
b = np.array([0, 4, 0])
c = np.array([0, 0, 5])
resultant = a + b + c

fig = plt.figure(figsize=(8, 6))
ax = fig.add_subplot(111, projection="3d")

# plot vectors
origin = np.array([0, 0, 0])
ax.quiver(*origin, *a, color='r', label='a (3)')
ax.quiver(*origin, *b, color='g', label='b (4)')
ax.quiver(*origin, *c, color='b', label='c (5)')
ax.quiver(*origin, *resultant, color='m', label='a+b+c')
```

```
ax.set_xlim([0, 8])
ax.set_ylim([0, 8])
ax.set_zlim([0, 8])

ax.set_xlabel("X axis")
ax.set_ylabel("Y axis")
ax.set_zlabel("Z axis")
ax.set_title("C code calculation + Python plot")

ax.legend()
plt.show()
```

# Plot-Using by both C and Python

C code calculation + Python plot

