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 $|\mathbf{a} + \mathbf{b} + \mathbf{c}|$.

Theoretical solution

Let

$$|\mathbf{a}| = 3, \quad |\mathbf{b}| = 4, \quad |\mathbf{c}| = 5 \tag{1}$$

Since each vector is perpendicular to the sum of the other two, we have:

$$\mathbf{a} \cdot (\mathbf{b} + \mathbf{c}) = 0, \quad \mathbf{b} \cdot (\mathbf{c} + \mathbf{a}) = 0, \quad \mathbf{c} \cdot (\mathbf{a} + \mathbf{b}) = 0 \tag{2}$$

Introduce notation:

$$s = \mathbf{a} \cdot \mathbf{b}, \quad t = \mathbf{b} \cdot \mathbf{c}, \quad u = \mathbf{c} \cdot \mathbf{a} \tag{3}$$

From (2), the equations become:

$$s + u = 0, \quad t + s = 0, \quad u + t = 0 \tag{4}$$

Theoretical solution

From the first equation,

$$u = -s \tag{5}$$

From the second equation,

$$t = -s \tag{6}$$

Substitute (5) and (6) into the third equation:

$$(-s) + (-s) = -2s = 0 \quad \Rightarrow \quad s = 0 \tag{7}$$

Hence,

$$s = t = u = 0 \tag{8}$$

This shows that $\mathbf{a}, \mathbf{b}, \mathbf{c}$ are mutually perpendicular.
Now,

$$\|\mathbf{a} + \mathbf{b} + \mathbf{c}\|^2 = \|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 + \|\mathbf{c}\|^2 + 2(s + t + u) \tag{9}$$

Theoretical solution

Substitute values from (1) and (8):

$$= 3^2 + 4^2 + 5^2 + 2(0 + 0 + 0) \tag{10}$$

$$= 9 + 16 + 25 = 50 \tag{11}$$

Therefore,

$$\|\mathbf{a} + \mathbf{b} + \mathbf{c}\| = \sqrt{50} = 5\sqrt{2} \tag{12}$$

Final Answer:

$$\boxed{5\sqrt{2}}$$

Python Code

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

# Define vectors
a = np.array([3, 0, 0]) # vector a
b = np.array([0, 4, 0]) # vector b
c = np.array([0, 0, 5]) # vector c

# Origin
O = np.array([0, 0, 0])

# Resultant
R = a + b + c
```

Python Code

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

# Plot coordinate axes
ax.quiver(0, 0, 0, 6, 0, 0, arrow_length_ratio=0.1, color='k')
ax.quiver(0, 0, 0, 0, 6, 0, arrow_length_ratio=0.1, color='k')
ax.quiver(0, 0, 0, 0, 0, 6, arrow_length_ratio=0.1, color='k')
```

Python Code

```
# Plot vectors a, b, c
ax.quiver(*O, *a, color='black', linewidth=2)
ax.text(*a, r'$\vec{a}$', color='black')

ax.quiver(*O, *b, color='black', linewidth=2)
ax.text(*b, r'$\vec{b}$', color='black')

ax.quiver(*O, *c, color='black', linewidth=2)
ax.text(*c, r'$\vec{c}$', color='black')

# Plot resultant
ax.quiver(*O, *R, color='red', linewidth=2)
ax.text(*R, r'$\vec{a}+\vec{b}+\vec{c}$', color='red')
```

Python Code

```
# Draw dashed parallelepiped edges
ax.plot([a[0], R[0]], [a[1], R[1]], [a[2], R[2]], 'k--')
ax.plot([b[0], R[0]], [b[1], R[1]], [b[2], R[2]], 'k--')
ax.plot([c[0], R[0]], [c[1], R[1]], [c[2], R[2]], 'k--')

# Labels
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('z')
ax.set_title("Fig. 4")

# Set limits
ax.set_xlim([0, 6])
ax.set_ylim([0, 6])
ax.set_zlim([0, 6])

plt.show()
```

C Code

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

int main() {
    // Given magnitudes
    double a = 3.0;
    double b = 4.0;
    double c = 5.0;

    // Since a, b, c are mutually perpendicular:
    double sumSq = a*a + b*b + c*c;

    // Magnitude of (a+b+c)
    double result = sqrt(sumSq);

    printf("||a + b + c|| = sqrt(%.0f) = %.4f\n", sumSq, result);

    return 0;
}
```

```python
import numpy as np

# Part 1: Input vectors
a = np.array([3.0, 0, 0]) # Along X-axis
b = np.array([0, 4.0, 0]) # Along Y-axis
c = np.array([0, 0, 5.0]) # Along Z-axis

print("---- Part 1: Input Vectors ----")
print("a =", a)
print("b =", b)
print("c =", c)
```

```python
import numpy as np

# Part 2: Resultant vector calculation
a = np.array([3.0, 0, 0])
b = np.array([0, 4.0, 0])
c = np.array([0, 0, 5.0])

res = a + b + c
sumSq = np.dot(a, a) + np.dot(b, b) + np.dot(c, c)
result = np.linalg.norm(res)

print("\n---- Part 2: Resultant ----")
print(f"||a + b + c|| = sqrt({sumSq:.0f}) = {result:.4f}")
```

Python and C Code

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

# Part 3: Plot vectors
a = np.array([3.0, 0, 0])
b = np.array([0, 4.0, 0])
c = np.array([0, 0, 5.0])
res = a + b + c

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

ax.quiver(0, 0, 0, a[0], a[1], a[2], color='r', label='a (3)')
ax.quiver(0, 0, 0, b[0], b[1], b[2], color='g', label='b (4)')
ax.quiver(0, 0, 0, c[0], c[1], c[2], color='b', label='c (5)')
ax.quiver(0, 0, 0, res[0], res[1], res[2], color='k', linewidth
    =2, label='a+b+c')
```

Python and C Code

```
import matplotlib.pyplot as plt

# Part 4: Plot setup and show
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

ax.set_xlim([0, 6])
ax.set_ylim([0, 6])
ax.set_zlim([0, 6])
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_title('Vector Addition: a + b + c')
ax.legend()

plt.show()
```

Graphical Representation:

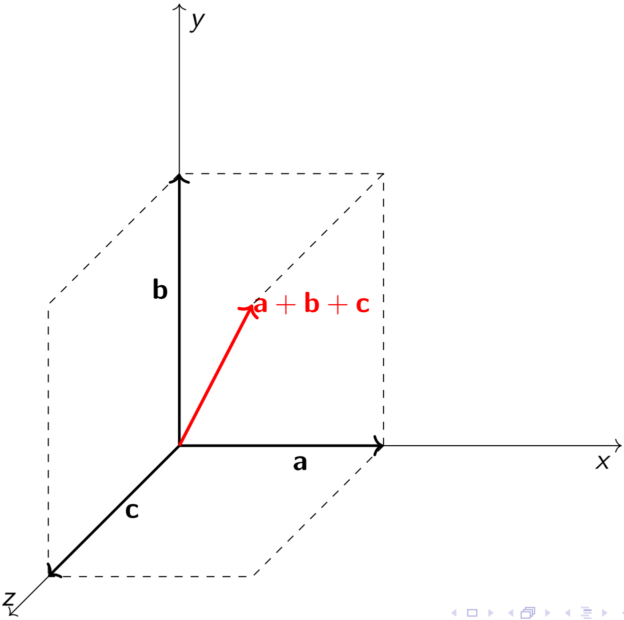

Fig. 4