

2.10.41

BEERAM MADHURI - EE25BTECH11012

September 2025

Question

Let the vectors \mathbf{a} , \mathbf{b} , \mathbf{c} and \mathbf{d} be such that $(\mathbf{a} \times \mathbf{b}) \times (\mathbf{c} \times \mathbf{d}) = \mathbf{0}$. Let A and B be planes determined by the pairs of vectors \mathbf{a} , \mathbf{b} and \mathbf{c} , \mathbf{d} respectively. Then the angle between A and B is

a) 0

b) $\frac{\pi}{4}$

c) $\frac{\pi}{3}$

d) $\frac{\pi}{2}$

$$(\mathbf{a} \times \mathbf{b}) \times (\mathbf{c} \times \mathbf{d}) = \mathbf{0}$$

plane A is determined by \mathbf{a}, \mathbf{b}

plane B is determined by \mathbf{c}, \mathbf{d}

finding Angle between Planes A and B

$$(\mathbf{a} \times \mathbf{b}) \times (\mathbf{c} \times \mathbf{d}) = 0 \quad (1)$$

\Rightarrow angle between $\mathbf{a} \times \mathbf{b}$ and $\mathbf{c} \times \mathbf{d}$ is 0

$$\therefore \mathbf{a} \times \mathbf{b} \parallel \mathbf{c} \times \mathbf{d} \quad (2)$$

Given that,
plane A is determined by **a, b**
plane B is determined by **c, d**

normals to planes A and B:

$$n_A = \mathbf{a} \times \mathbf{b} \quad (3)$$

$$n_B = \mathbf{c} \times \mathbf{d} \quad (4)$$

Angle between Planes A and B = Angle between Normals n_A and n_B

$$\mathbf{a} \times \mathbf{b} \parallel \mathbf{c} \times \mathbf{d} \quad (5)$$

$$\therefore n_A \parallel n_B \quad (6)$$

$$\therefore \text{planeA} \parallel \text{planeB} \quad (7)$$

Hence, Angle between the planes is 0
option (a)

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

#Example vectors that satisfy (ab)(cd)=0

a = np.array([1, 0, 0])
b = np.array([0, 1, 0])
c = np.array([2, 0, 0])
d = np.array([0, 2, 0])
```

```
#Normals of planes A and B

n1 = np.cross(a, b)
n2 = np.cross(c, d)

print("n1:", n1, " n2:", n2) # normals
print("Cross of normals:", np.cross(n1, n2)) # should be zero
```



```
#Mesh grid for plotting
xx, yy = np.meshgrid(np.linspace(-2, 2, 20), np.linspace(-2, 2,
    20))

def plane_z(normal, X, Y):
#  $n(x,y,z) = 0 \Rightarrow z = (-n_x X - n_y Y)/n_z$  if  $n_z \neq 0$ 
return (-normal[0]*X - normal[1]*Y)/normal[2] if normal[2] != 0
    else np.zeros_like(X)

z1 = plane_z(n1, xx, yy)
z2 = plane_z(n2, xx, yy)
```

```
fig = plt.figure(figsize=(9,7))
ax = fig.add_subplot(111, projection='3d')

#Plot planes

surf1 = ax.plot_surface(xx, yy, z1, color='cyan', alpha=0.5)
surf2 = ax.plot_surface(xx, yy, z2 + 0.1, color='magenta', alpha
    =0.5) # small offset for visibility
```

```
#Mark plane names
ax.text(0, 0, 0.05, "Plane A", color='blue', fontsize=12, ha='
center')
ax.text(0, 0, 0.15, "Plane B", color='purple', fontsize=12, ha='
center')

#Plot and label normals
origin = np.array([0, 0, 0])
ax.quiver(*origin, *n1, color='blue', length=1.0,
          arrow_length_ratio=0.1)
ax.quiver(*origin, *n2, color='red', length=1.0,
          arrow_length_ratio=0.1)
```

```
#Add text at the arrow tips for normals
ax.text((n11.1), "n", color='blue', fontsize=12)
ax.text((n21.1), "n", color='red', fontsize=12)
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_title('Planes A & B with Normals Marked')
#Make the axes equal for a better view
ax.set_box_aspect([1,1,0.5])
plt.show()
```

```
#include <stdio.h>
#include <math.h>
// cross product of two 3D vectors
void cross(double u[3], double v[3], double result[3]) {
    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];}
// dot product
double dot(double u[3], double v[3]) {
    return u[0]*v[0] + u[1]*v[1] + u[2]*v[2];
}
```

```
// magnitude of a 3D vector
double magnitude(double v[3]) {
    return sqrt(dot(v, v));
}

int main(void) {
    // Example vectors satisfying (ab)(cd)=0
    double a[3] = {1, 0, 0};
    double b[3] = {0, 1, 0};
    double c[3] = {2, 0, 0};
    double d[3] = {0, 2, 0};
}
```

```
double n1[3], n2[3];
cross(a, b, n1);
cross(c, d, n2);
double angle = acos(dot(n1, n2) / (magnitude(n1) * magnitude(
    n2)));
printf("Angle between the two planes (radians): %f\n", angle)
    ;
printf("Angle between the two planes (degrees): %f\n", angle
    * 180.0 / M_PI);
return 0;
}
```

```
import subprocess

# 1. Compile the C program
subprocess.run(["gcc", "planes.c", "-o", "planes"])

# 2. Run the compiled C program
result = subprocess.run(["./planes"], capture_output=True, text=
    True)

# 3. Print the output from the C program
print(result.stdout)
```


Planes A & B with Normals Parallel

