

Question-2.4.16

EE25BTECH11022 - sankeerthan

Question

show that the line through the points $(1, -1, 2)$, $(3, 4, -2)$ is perpendicular to the line through the points $(0, 3, 2)$ and $(3, 5, 6)$

Solution

$$\text{let } \mathbf{A} = (1, -1, 2), \mathbf{B} = (3, 4, -2), \mathbf{C} = (0, 3, 2), \mathbf{D} = (3, 5, 6) \quad (1)$$

Direction vector of line joining points **A** and **B** is

$$\mathbf{B} - \mathbf{A} = \begin{pmatrix} 3 - 1 \\ 4 - (-1) \\ -2 - 2 \end{pmatrix} \quad (2)$$

$$= \begin{pmatrix} 2 \\ 5 \\ -4 \end{pmatrix} \quad (3)$$

$$(4)$$

Direction vector of line joining points **C** and **D** is

Solution

$$\mathbf{C} - \mathbf{D} = \begin{pmatrix} 3 - 0 \\ 5 - 3 \\ 6 - 2 \end{pmatrix} \quad (5)$$

$$= \begin{pmatrix} 3 \\ 2 \\ 4 \end{pmatrix} \quad (6)$$

$$(\mathbf{B} - \mathbf{A})^T (\mathbf{D} - \mathbf{C}) = (2 \ 5 \ -4) \begin{pmatrix} 3 \\ 2 \\ 4 \end{pmatrix} \quad (7)$$

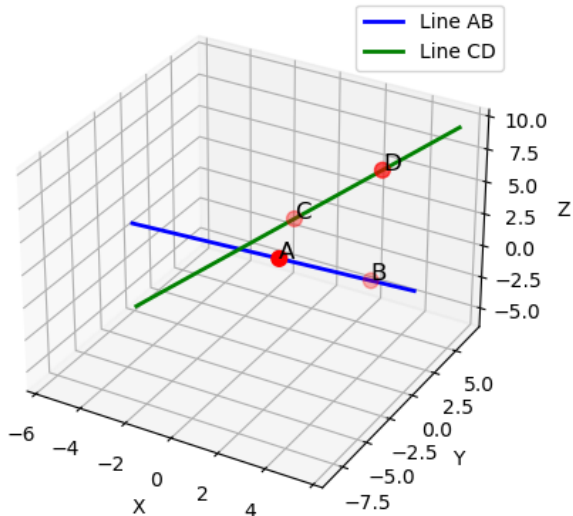
$$= 2(3) + 5(2) + (-4)(4) \quad (8)$$

$$= 6 + 10 - 16 = 0 \quad (9)$$

$$(\mathbf{B} - \mathbf{A})^T (\mathbf{D} - \mathbf{C}) = 0 \quad (10)$$

Therefore, the lines joining points **A,B** and **C,D** are perpendicular

Perpendicular Lines



```
// code.c
#define NUM_POINTS 4

void get_points(double *arr) {
    double points[NUM_POINTS][3] = {
        {1, -1, 2},
        {3, 4, -2},
        {0, 3, 2},
        {3, 5, 6}
    };
    for (int i = 0; i < NUM_POINTS; ++i)
        for (int j = 0; j < 3; ++j)
            arr[i*3 + j] = points[i][j];
}
```

```
void get_dir_vectors(double *v1, double *v2) {  
    // v1: AB; v2: CD  
    v1[0] = 3 - 1;  
    v1[1] = 4 - (-1);  
    v1[2] = -2 - 2;  
    v2[0] = 3 - 0;  
    v2[1] = 5 - 3;  
    v2[2] = 6 - 2;  
}  
  
int check_perpendicular() {  
    double v1[3], v2[3];  
    get_dir_vectors(v1, v2);  
    double dot = v1[0]*v2[0] + v1[1]*v2[1] + v1[2]*v2[2];  
    return dot == 0 ? 1 : 0;  
}
```

```
# Code by GVV Sharma  
# Modified for Problem Solution  
# Released under GNU GPL  
# Calculating area enclosed between curves  
# call.py  
import ctypes  
  
lib = ctypes.CDLL('./libcode.so')  
  
# Perpendicularity check from C  
result = lib.check_perpendicular()
```



```
print("The lines are perpendicular." if result else "The lines  
are not perpendicular.")  
  
# Extract and print the points  
arr = (ctypes.c_double * 12)()  
lib.get_points(arr)  
for i in range(4):  
    print(f"Point {chr(65 + i)}: ({arr[i*3]}, {arr[i*3 + 1]}, {  
        arr[i*3 + 2]})")
```

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

# Load .so and extract points
lib = ctypes.CDLL('./libcode.so')
arr = (ctypes.c_double * 12)()
lib.get_points(arr)
pts = np.array(arr).reshape((4,3))
A, B, C, D = pts

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

```
ax.plot([A[0], B[0]], [A[1], B[1]], [A[2], B[2]], color='blue',
        label='Line AB')
ax.plot([C[0], D[0]], [C[1], D[1]], [C[2], D[2]], color='green',
        label='Line CD')

# Plot and annotate points
ax.scatter(pts[:,0], pts[:,1], pts[:,2], color='red', s=70)
for i, label in enumerate(['A', 'B', 'C', 'D']):
    ax.text(pts[i,0], pts[i,1], pts[i,2], label, size=12)

ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_title('3D Plot of Two Perpendicular Lines')
ax.legend()
plt.show()
```