3.3.15

Shivam Sawarkar
AI25BTECH11031

September 29, 2025

Question

Find the value of x such that the points $A(3, 2, 1)$, $B(4, x, 5)$, $C(4, 2, -2)$ and $D(6, 5, -1)$ are coplanar.

Solution

Let the plane (not passing through the origin) be given by

$$\mathbf{n}^\top \mathbf{x} = 1, \qquad \mathbf{n} = \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix} \tag{1}$$

Since the points

$$\mathbf{A} = \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} 4 \\ 2 \\ -2 \end{pmatrix} \quad \mathbf{C} = \begin{pmatrix} 6 \\ 5 \\ -1 \end{pmatrix} \tag{2}$$

lie on the plane, they satisfy

$$\mathbf{n}^\top \mathbf{A} = 1 \tag{3}$$

$$\mathbf{n}^\top \mathbf{B} = 1 \tag{4}$$

$$\mathbf{n}^\top \mathbf{C} = 1 \tag{5}$$

Solution

$$\begin{pmatrix} 3 & 2 & 1 \\ 4 & 2 & -2 \\ 6 & 5 & -1 \end{pmatrix} \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}. \tag{6}$$

Thus

$$n_1 = \frac{9}{16}, \qquad n_2 = -\frac{7}{16}, \qquad n_3 = \frac{3}{16}. \tag{7}$$

Now require B to lie on the same plane:

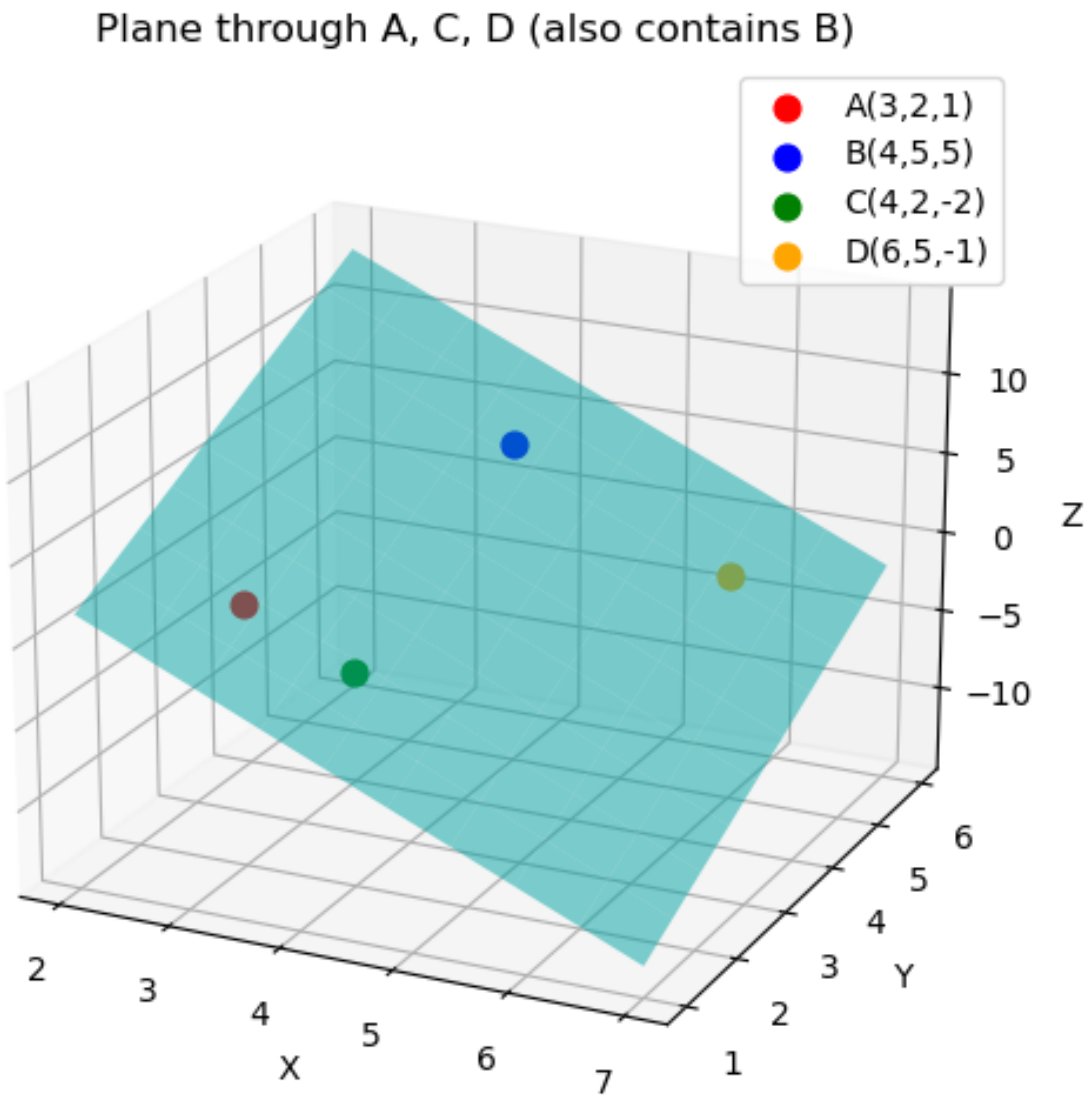
$$\mathbf{n}^\top \mathbf{B} = 1 \tag{8}$$

$$\begin{pmatrix} \frac{9}{16} & \frac{-7}{16} & \frac{3}{16} \end{pmatrix} \begin{pmatrix} 4 \\ x \\ 5 \end{pmatrix} = 1 \tag{9}$$

$$\frac{36}{16} - \frac{7}{16}x + \frac{15}{16} = 1 \tag{10}$$

$$x = 5 \tag{11}$$

Plot


Plane through A, C, D (also contains B)
A(3,2,1)
B(4,5,5)
C(4,2,-2)
D(6,5,-1)
10
5
0
−5
−10
Z
2
3
4
5
6
7
X
1
2
3
4
5
6
Y

C Code

```
#ifndef COPLANAR_H
#define COPLANAR_H

#include <stdio.h>

typedef struct {
    double x;
    double y;
    double z;
} Point;

// Function to compute vector b-a
Point vector(Point a, Point b) {
    Point res = {b.x - a.x, b.y - a.y, b.z - a.z};
    return res;
}
```

C Code

```
// Cross product
Point cross(Point u, Point v) {
    Point res = {
        u.y * v.z - u.z * v.y,
        u.z * v.x - u.x * v.z,
        u.x * v.y - u.y * v.x
    };
    return res;
}

// Dot product
double dot(Point u, Point v) {
    return u.x * v.x + u.y * v.y + u.z * v.z;
}
```

C Code

```
// Function to compute value of x for coplanarity
double solve_for_x(Point A, Point C, Point D) {
    // AC and AD vectors
    Point AC = vector(A, C);
    Point AD = vector(A, D);
    // Cross product AC × AD
    Point cross_prod = cross(AC, AD);
    // Scalar triple product condition:
    // (1, x-2, 4)  (cross_prod) = 0
    // => coeff_x * x + constant = 0
    double coeff_x = cross_prod.y;
    double constant = cross_prod.x + (-2)*cross_prod.y + 4*cro
    return -constant / coeff_x;
}
#endif
```

C Code

```
#include "solution.h"

int main() {
    Point A, C, D;
    // Input A, C, D
    printf("Enter coordinates of A (x y z): ");
    scanf("%lf %lf %lf", &A.x, &A.y, &A.z);
    printf("Enter coordinates of C (x y z): ");
    scanf("%lf %lf %lf", &C.x, &C.y, &C.z);
    printf("Enter coordinates of D (x y z): ");
    scanf("%lf %lf %lf", &D.x, &D.y, &D.z);
    // Solve for x
    double x = solve_for_x(A, C, D);
    printf("The value of x such that A, B, C, D are coplanar i
    return 0;
}
```

Python Code

```
import numpy as np

def solve_for_x(A, C, D):
    # Convert to numpy arrays
    A, C, D = np.array(A), np.array(C), np.array(D)

    # Step 1: compute vectors AC and AD
    AC = C - A
    AD = D - A

    # Step 2: normal vector n = AC × AD
    n = np.cross(AC, AD)

    # Step 3: vector (B-A) = (1, x-2, 4)
    coeff_x = n[1]
    constant = n[0]*1 + n[1]*(-2) + n[2]*4
```

Python Code

```
    x = -constant / coeff_x
    return x

if __name__ == "__main__":
    # Take inputs
    A = list(map(float, input("Enter coordinates of A (x y z):
    C = list(map(float, input("Enter coordinates of C (x y z):
    D = list(map(float, input("Enter coordinates of D (x y z):

    x_value = solve_for_x(A, C, D)
    print("The value of x such that A, B, C, D are coplanar
    is:", round(x_value, 2))
```

Python + C Code

```
import ctypes

# Load the shared library
lib = ctypes.CDLL("./solution.so")

# Define the Point struct in Python
class Point(ctypes.Structure):
    _fields_ = [("x", ctypes.c_double),
                ("y", ctypes.c_double),
                ("z", ctypes.c_double)]

# Tell ctypes about the function signature
lib.solve_for_x.argtypes = [Point, Point, Point]
lib.solve_for_x.restype = ctypes.c_double
```

Python + C Code

```
if __name__ == "__main__":
    A_vals = list(map(float, input("Enter coordinates of A
    (x y z): ").split()))
    C_vals = list(map(float, input("Enter coordinates of C
    (x y z): ").split()))
    D_vals = list(map(float, input("Enter coordinates of D
    (x y z): ").split()))

    A = Point(*A_vals)
    C = Point(*C_vals)
    D = Point(*D_vals)

    x_value = lib.solve_for_x(A, C, D)

    print("The value of x such that A, B, C, D are coplanar
    is:", round(x_value, 2))
```