4.8.27

Shivam Sawarkar
AI25BTECH11031

September 29, 2025

Question

Find the equation of the plane passing through $(-1, 3, 2)$ and perpendicular to the planes $x + 2y + 3z = 5$ and $3x + 3y + z = 0$.

Solution

Normals of the given planes are

$$\mathbf{n_1} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}, \quad \mathbf{n_2} = \begin{pmatrix} 3 \\ 3 \\ 1 \end{pmatrix}. \tag{1}$$

Let the required plane have normal vector $\mathbf{n}$
Since it is perpendicular to both given planes:

$$\mathbf{n_1}^\top \mathbf{n} = 0, \quad \mathbf{n_2}^\top \mathbf{n} = 0. \tag{2}$$

That is,

$$\begin{pmatrix} \mathbf{n_1} & \mathbf{n_2} \end{pmatrix}^\top \mathbf{n} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{3}$$

$$\begin{pmatrix} 1 & 2 & 3 \\ 3 & 3 & 1 \end{pmatrix} \mathbf{n} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{4}$$

Solution

Let

$$\mathbf{n} = \begin{pmatrix} a \\ b \\ c \end{pmatrix} \tag{5}$$

$$a + 2b + 3c = 0, \tag{6}$$

$$3a + 3b + c = 0. \tag{7}$$

From these, we get

$$\mathbf{n} = t \begin{pmatrix} 7 \\ -8 \\ 3 \end{pmatrix}, \quad t \in \mathbb{R}, \quad t \neq 0 \tag{8}$$

Equation of plain is

$$\mathbf{n}^\top \mathbf{x} = 1 \tag{9}$$

Solution

since point $\mathbf{p} = \begin{pmatrix} -1 \\ 3 \\ 2 \end{pmatrix}$ lies on the plain

$$\mathbf{n}^\top \mathbf{p} = 1 \tag{10}$$

Substituting,

$$\begin{pmatrix} 7 & -8 & 3 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 7 & -8 & 3 \end{pmatrix} \begin{pmatrix} -1 \\ 3 \\ 2 \end{pmatrix}. \tag{11}$$

Solution

Substituting,

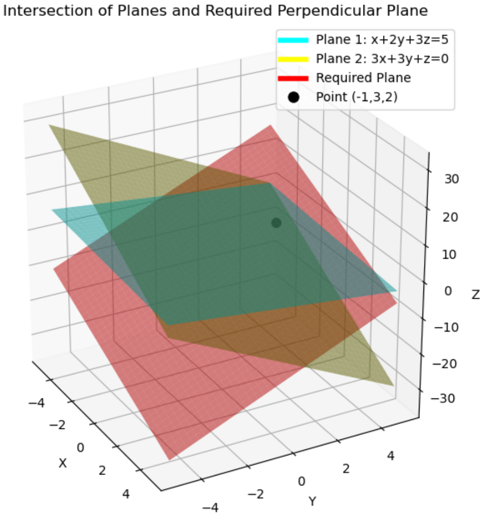
$$\begin{pmatrix} 7 & -8 & 3 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 7 & -8 & 3 \end{pmatrix} \begin{pmatrix} -1 \\ 3 \\ 2 \end{pmatrix}. \tag{12}$$

$$\begin{pmatrix} 7 & -8 & 3 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = -25 \tag{13}$$

$$\frac{-1}{25} \begin{pmatrix} 7 & -8 & 3 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = 1 \tag{14}$$

$$\mathbf{n} = \frac{-1}{25} \begin{pmatrix} 7 \\ -8 \\ 3 \end{pmatrix} \tag{15}$$

Plot


Intersection of Planes and Required Perpendicular Plane
Plane 1: x+2y+3z=5
Plane 2: 3x+3y+z=0
Required Plane
Point (-1,3,2)
30
20
10
0
−10
−20
−30
Z
−4
−2
0
2
4
X
−4
−2
0
2
4
Y

C Code

```
#ifndef PLANE_H
#define PLANE_H

#include <stdio.h>

// Function to compute cross product of two vectors
void cross_product(double a1, double b1, double c1,
                   double a2, double b2, double c2,
                   double *rx, double *ry, double *rz) {
    *rx = b1*c2 - c1*b2;
    *ry = c1*a2 - a1*c2;
    *rz = a1*b2 - b1*a2;
}
```

C Code

```
// Function to compute dot product of two vectors
double dot_product(double a1, double b1, double c1,
                   double a2, double b2, double c2) {
  return a1*a2 + b1*b2 + c1*c2;
}

// Function to print plane equation given normal and point
void plane_equation(double nx, double ny, double nz,
                    double x0, double y0, double z0) {
  double rhs = dot_product(nx, ny, nz, x0, y0, z0);
  printf("\nEquation of required plane: %.2lf*x + %.2lf*y + %.
         nx, ny, nz, rhs);
}

#endif
```

C Code

```
#include "solution.h"

int main() {
    double x0, y0, z0;
    printf("Enter point (x0 y0 z0): ");
    scanf("%lf %lf %lf", &x0, &y0, &z0);
    double a1, b1, c1, d1;
    printf("Enter coefficients of plane1 (a1 b1 c1 d1): ");
    scanf("%lf %lf %lf %lf", &a1, &b1, &c1, &d1);
    double a2, b2, c2, d2;
    printf("Enter coefficients of plane2 (a2 b2 c2 d2): ");
    scanf("%lf %lf %lf %lf", &a2, &b2, &c2, &d2);

    // Normal vectors of plane1 and plane2
    double n1x = a1, n1y = b1, n1z = c1;
    double n2x = a2, n2y = b2, n2z = c2;
```

C Code

```
    // Compute required normal (cross product)
    double nx, ny, nz;
    cross_product(n1x, n1y, n1z, n2x, n2y, n2z, &nx, &ny, &nz

    // Print required plane
    plane_equation(nx, ny, nz, x0, y0, z0);

    return 0;
}
```

Python Code

```
import numpy as np

def cross_product(v1, v2):
    """Return cross product of two vectors"""
    return np.cross(v1, v2)

def dot_product(v1, v2):
    """Return dot product of two vectors"""
    return np.dot(v1, v2)

def plane_equation(normal, point):
    """Given normal vector and point, return plane equation co
    d = dot_product(normal, point)
    return (*normal, d)
```

Python Code

```
def main():
    # Input point
    x0, y0, z0 = map(float, input("Enter point (x0 y0 z0): ").

    # Input first plane coefficients
    a1, b1, c1, d1 = map(float, input("Enter coefficients of p

    # Input second plane coefficients
    a2, b2, c2, d2 = map(float, input("Enter coefficients of p

    # Normals of plane1 and plane2
    n1 = np.array([a1, b1, c1])
    n2 = np.array([a2, b2, c2])

    # Required normal vector is cross product
    n_required = cross_product(n1, n2)
```

Python Code

```
    # Given point
    p = np.array([x0, y0, z0])

    # Plane equation: n·x = d
    a, b, c, d = plane_equation(n_required, p)

    print(f"\nRequired plane equation:")
    print(f"{a:.2f}x + {b:.2f}y + {c:.2f}z = {d:.2f}")

if __name__ == "__main__":
    main()
```

Python + C Code

```
import ctypes
import numpy as np

plane_lib = ctypes.CDLL("./solution.so")

plane_lib.cross_product.argtypes = [
    ctypes.c_double, ctypes.c_double, ctypes.c_double,
    ctypes.c_double, ctypes.c_double, ctypes.c_double,
    ctypes.POINTER(ctypes.c_double), ctypes.POINTER(ctypes.c_d
]
plane_lib.cross_product.restype = None

plane_lib.dot_product.argtypes = [
    ctypes.c_double, ctypes.c_double, ctypes.c_double,
    ctypes.c_double, ctypes.c_double, ctypes.c_double
]
```

Python + C Code

```
plane_lib.dot_product.restype = ctypes.c_double

plane_lib.plane_equation.argtypes = [
    ctypes.c_double, ctypes.c_double, ctypes.c_double,
    ctypes.c_double, ctypes.c_double, ctypes.c_double
]
plane_lib.plane_equation.restype = None


def main():
    x0, y0, z0 = map(float, input("Enter point (x0 y0 z0): ").
    a1, b1, c1, d1 = map(float, input("Enter coefficients of p
    a2, b2, c2, d2 = map(float, input("Enter coefficients of p
```

Python + C Code

```
n1x, n1y, n1z = a1, b1, c1
n2x, n2y, n2z = a2, b2, c2

nx = ctypes.c_double()
ny = ctypes.c_double()
nz = ctypes.c_double()
plane_lib.cross_product(n1x, n1y, n1z, n2x, n2y, n2z,
ctypes.byref(nx), ctypes.byref(ny), ctypes.byref(nz))

rhs = plane_lib.dot_product(nx.value, ny.value,
nz.value, x0, y0, z0)

print(f"\nRequired plane equation:")
print(f"{nx.value:.2f}x + {ny.value:.2f}y +
{nz.value:.2f}z = {rhs:.2f}")
```