4.3.16

J.NAVYASRI- EE25BTECH11028

september 2025

Find the equation of the plane through the points

$$(2, 1, 0), \quad (3, -2, -2), \quad (3, 1, 7).$$

Soultion:

Given three points:

$$P_1(2, 1, 0), \quad P_2(3, -2, -2), \quad P_3(3, 1, 7)$$

The direction vectors are:

$$\left(v\right)_1 = \left(P_2 - P_1\right) = (3 - 2, \ -2 - 1, \ -2 - 0) = (1, -3, -2) \quad (1)$$

$$\left(v\right)_2 = \left(P_3 - P_1\right) = (3 - 2, \ 1 - 1, \ 7 - 0) = (1, 0, 7) \quad (2)$$

The normal vector to the plane is given by the cross product:

$$\left(n\right) = \left(v\right)_1 \times \left(v\right)_2 \quad (3)$$

Soultion:

$$\left(n\right) = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 1 & -3 & -2 \\ 1 & 0 & 7 \end{vmatrix} \tag{4}$$

$$\left(n\right) = (-21, -9, 3) = (-7, -3, 1) \tag{5}$$

Hence, the equation of the plane is:

$$-7(x - 2) - 3(y - 1) + (z - 0) = 0 \tag{6}$$

Simplifying:

$$-7x + 14 - 3y + 3 + z = 0 \tag{7}$$

$$7x + 3y - z - 17 = 0 \tag{8}$$

Final answer:

$$\boxed{7x + 3y - z - 17 = 0}$$

Python Code

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

# Given points
P1 = np.array([2, 1, 0])
P2 = np.array([3, -2, -2])
P3 = np.array([3, 1, 7])
```

Python Code

```python
# Create a meshgrid for x, y
xx, yy = np.meshgrid(range(-2, 6), range(-3, 6))

# Equation of plane: 7x + 3y - z - 17 = 0 => z = 7x + 3y - 17
zz = 7*xx + 3*yy - 17

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

# Plot the surface (plane)
ax.plot_surface(xx, yy, zz, alpha=0.5, color='cyan')

# Plot points
ax.scatter(*P1, color='r', s=50, label='P1 (2,1,0)')
ax.scatter(*P2, color='g', s=50, label='P2 (3,-2,-2)')
ax.scatter(*P3, color='b', s=50, label='P3 (3,1,7)')
```
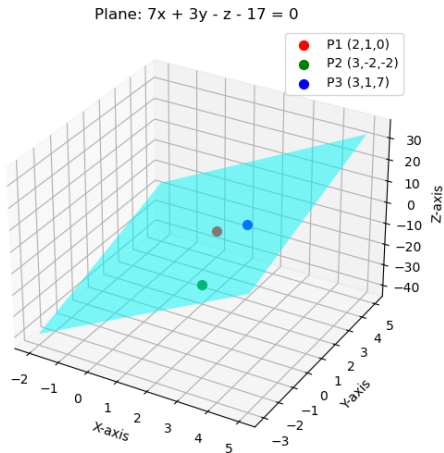
Python Code

```python
# Labels
ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.set_zlabel('Z-axis')
ax.set_title("Plane: 7x + 3y - z - 17 = 0")

ax.legend()
plt.savefig("fig6.png")
plt.show()
```

Plot-Using by Python

C Code

```c
#include <stdio.h>

// Function to compute gcd
int gcd(int a, int b) {
    if (b == 0) return a > 0 ? a : -a;
    return gcd(b, a % b);
}

// Function to compute gcd of 4 numbers
int gcd4(int a, int b, int c, int d) {
    int g = gcd(a, b);
    g = gcd(g, c);
    g = gcd(g, d);
    return g;
}
```

C Code

```c
int main() {
    // Given three points
    int x1 = 2, y1 = 1, z1 = 0;
    int x2 = 3, y2 = -2, z2 = -2;
    int x3 = 3, y3 = 1, z3 = 7;
```

C Code

```
// Direction vectors
    int v1x = x2 - x1, v1y = y2 - y1, v1z = z2 - z1;
    int v2x = x3 - x1, v2y = y3 - y1, v2z = z3 - z1;

    // Cross product  normal vector (a, b, c)
    int a = v1y * v2z - v1z * v2y;
    int b = v1z * v2x - v1x * v2z;
    int c = v1x * v2y - v1y * v2x;

    // Constant term d
    int d = -(a * x1 + b * y1 + c * z1);
```

C Code

```c
// Simplify using gcd
    int g = gcd4(a, b, c, d);
    a /= g; b /= g; c /= g; d /= g;

    // Make leading coefficient positive
    if (a < 0) {
        a = -a; b = -b; c = -c; d = -d;
    }

    // Final output
    printf("The equation of plane is: %dx + %dy + %dz + %d = 0\n"
        , a, b, c, d);

    return 0;
}
```

Python and C Code

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

# Load the compiled C library
lib = ctypes.CDLL("./plane.so")

# Define the cross_product function signature
lib.cross_product.argtypes = [ctypes.POINTER(ctypes.c_double),
                              ctypes.POINTER(ctypes.c_double),
                              ctypes.POINTER(ctypes.c_double)]

# Points
P1 = np.array([2,1,0], dtype=np.double)
P2 = np.array([3,-2,-2], dtype=np.double)
P3 = np.array([3,1,7], dtype=np.double)
```

Python and C Code

```python
# Direction vectors
v1 = P2 - P1
v2 = P3 - P1

# Prepare ctypes arrays
v1_c = (ctypes.c_double * 3)(*v1)
v2_c = (ctypes.c_double * 3)(*v2)
n_c = (ctypes.c_double * 3)()

# Call C function
lib.cross_product(v1_c, v2_c, n_c)

# Normal vector from C
n = np.array([n_c[0], n_c[1], n_c[2]])
print("Normal vector from C:", n)

# Equation of plane: n(X - P1) = 0  ax+by+cz+d=0
a, b, c = n
d = -(a*P1[0] + b*P1[1] + c*P1[2])
```

Python and C Code

```python
print(f"Plane equation: {a}x + {b}y + {c}z + {d} = 0")

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

# Plot points
ax.scatter(*P1, color='red', s=50, label="P1(2,1,0)")
ax.scatter(*P2, color='blue', s=50, label="P2(3,-2,-2)")
ax.scatter(*P3, color='green', s=50, label="P3(3,1,7)")
```

Python and C Code

```python
# Create grid for plane
xx, yy = np.meshgrid(range(0,6), range(-3,3))
zz = (-a*xx - b*yy - d)/c

ax.plot_surface(xx, yy, zz, alpha=0.3, color='yellow')

ax.set_xlabel("X")
ax.set_ylabel("Y")
ax.set_zlabel("Z")
ax.legend()
plt.show()
```

Plot-Using by C and Python