4.3.17

Jnanesh Sathisha karmar - EE25BTECH11029

September 18,2025

Question

A plane passes through the points $(2, 0, 0)$, $(0, 3, 0)$ and $(0, 0, 4)$. The equation of the plane is _____

Equation

Given details

$$\mathbf{A} = \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} 0 \\ 3 \\ 0 \end{pmatrix} \quad \mathbf{C} = \begin{pmatrix} 0 \\ 0 \\ 4 \end{pmatrix} \tag{1}$$

Theoretical Solution

The points for plane for 3 given points is:

$$\mathbf{n}^\top x = c \tag{2}$$

to find $\mathbf{n}$ by performing Gaussian elimination on the augmented matrix:

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} & \mathbf{C} \end{pmatrix}^\top \mathbf{n} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \tag{3}$$

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

$$\left(\begin{array}{ccc|c} 2 & 0 & 0 & 1 \\ 0 & 3 & 0 & 1 \\ 0 & 0 & 4 & 1 \end{array}\right) \xrightarrow[R_2 \leftarrow R_2/3 \\ R_3 \leftarrow R_3/4]{R_1 \leftarrow R_1/2} \left(\begin{array}{ccc|c} 1 & 0 & 0 & \frac{1}{2} \\ 0 & 1 & 0 & \frac{1}{3} \\ 0 & 0 & 1 & \frac{1}{4} \end{array}\right) \tag{5}$$

Theoretical Solution

This gives the solultion:

$$\mathbf{n} = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{3} \\ \frac{1}{4} \end{pmatrix} \tag{6}$$

Therefore the equation of plane is:

$$\begin{pmatrix} \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \end{pmatrix}^{\top} \mathbf{x} = 1 \tag{7}$$

$$\begin{pmatrix} 6 & 4 & 3 \end{pmatrix}^{\top} = 12 \tag{8}$$

C Code (1) - Function to store the points

```c
#include <stdlib.h>


float* generate_plane_points(float x_min, float x_max, float
    y_min, float y_max, int num_steps) {
    if (num_steps <= 1) {
        return NULL;
    }

    int total_points = num_steps * num_steps;
    float* points = (float*)malloc(total_points * 3 * sizeof(
        float));
    if (points == NULL) {
        return NULL;
    }
```

C Code (1) - Function to store the points

```
    float x_step_size = (x_max - x_min) / (num_steps - 1);
    float y_step_size = (y_max - y_min) / (num_steps - 1);

    int index = 0;

    for (int i = 0; i < num_steps; i++) {
        float x = x_min + i * x_step_size;

        for (int j = 0; j < num_steps; j++) {
            float y = y_min + j * y_step_size;


            float z = 4.0f - 2.0f * x - (4.0f / 3.0f) * y;
```

C Code (1) - Function to store the points

```
            points[index++] = x;
            points[index++] = y;
            points[index++] = z;
        }
    }

    return points;
}


void free_points(float* points) {
    if (points != NULL) {
        free(points);
    }
}
```

Python Code - Using Shared Object

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

lib = ctypes.CDLL("./plane.so")

lib.generate_plane_points.argtypes = [
    ctypes.c_float, ctypes.c_float,
    ctypes.c_float, ctypes.c_float,
    ctypes.c_int
]
lib.generate_plane_points.restype = ctypes.POINTER(ctypes.c_float
    )

lib.free_points.argtypes = [ctypes.POINTER(ctypes.c_float)]
lib.free_points.restype = None
NUM_STEPS = 50
total_points = NUM_STEPS * NUM_STEPS
points_ptr = None
```

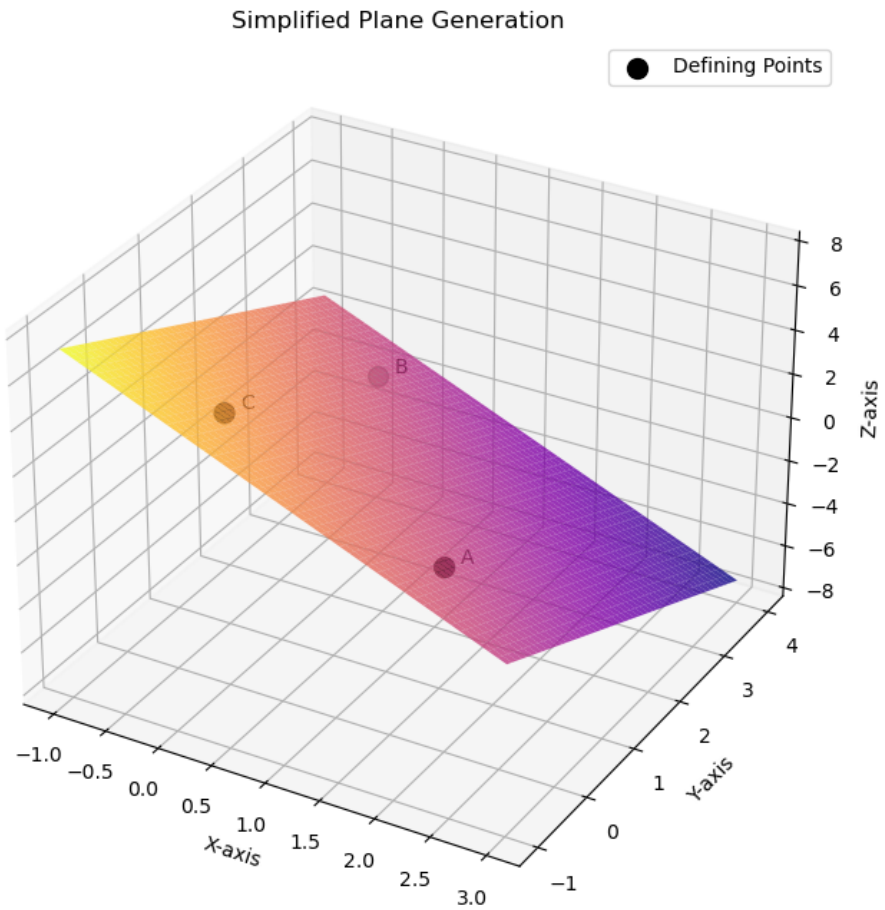
Python Code - Using Shared Object

```
try:
    points_ptr = lib.generate_plane_points(-1.0, 3.0,-1.0, 4.0,
        NUM_STEPS)
    if not points_ptr:
        raise MemoryError("C function failed to allocate memory."
            )

    points_np = np.ctypeslib.as_array(points_ptr, shape=(
        total_points, 3))
    points_data = np.copy(points_np)
finally:
    if points_ptr:
        lib.free_points(points_ptr)
X = points_data[:, 0].reshape(NUM_STEPS, NUM_STEPS)
Y = points_data[:, 1].reshape(NUM_STEPS, NUM_STEPS)
Z = points_data[:, 2].reshape(NUM_STEPS, NUM_STEPS)
```

Python Code - Using Shared Object

```
fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(projection='3d')


ax.plot_surface(X, Y, Z, cmap='plasma', alpha=0.8)
ax.scatter([2, 0, 0], [0, 3, 0], [0, 0, 4], color='black', s=100,
    label='Defining Points')
ax.text(2.1,0.1,0.1,"A",color="black")
ax.text(0.1,3.1,0.1,"B",color="black")
ax.text(0.1,0.1,4.1,"C",color="black")
ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.set_zlabel('Z-axis')
ax.set_title('Simplified Plane Generation')
ax.legend()
plt.savefig("./figs/plane.png")
subprocess.run(shlex.split('termux-open ../figs/parallelogram.png
    '))
```

Plot-Using Both C and Python



Python Code

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

def plot_plane_from_points():
    p1 = np.array([2, 0, 0])
    p2 = np.array([0, 3, 0])
    p3 = np.array([0, 0, 4])

    x_range = np.linspace(-1, 4, 20)
    y_range = np.linspace(-1, 4, 20)
    X, Y = np.meshgrid(x_range, y_range)

    Z = 4 - 2 * X - (4 / 3) * Y

    fig = plt.figure(figsize=(10, 8))
    ax = fig.add_subplot(projection='3d')
```

Python Code

```
    ax.plot_surface(X, Y, Z, alpha=0.7, cmap='plasma', edgecolor=
        'none')

    ax.scatter(p1[0], p1[1], p1[2], color='red', s=120, label='
        (2,0,0)', depthshade=False)
    ax.scatter(p2[0], p2[1], p2[2], color='red', s=120, label='
        (0,3,0)', depthshade=False)
    ax.scatter(p3[0], p3[1], p3[2], color='red', s=120, label='
        (0,0,4)', depthshade=False)
    ax.text(2.1,0.1,0.1,"A",color="black")
    ax.text(0.1,3.1,0.1,"B",color="black")
    ax.text(0.1,0.1,4.1,"C",color="black")
```

Python Code

```
    ax.set_xlabel('X-axis')
    ax.set_ylabel('Y-axis')
    ax.set_zlabel('Z-axis')
    ax.set_title('Plane Passing Through Three Points')
    ax.legend()
    ax.set_xlim([-1, 4])
    ax.set_ylim([-1, 4])
    ax.set_zlim([-1, 6])
    plt.savefig("./figs/plane2.png")
    plt.show()
    subprocess.run(shlex.split('termux-open ../figs/parallelogram
        .png'))

if __name__ == '__main__':
    plot_plane_from_points()
```

Plot-Using only Python