MATGEO Presentation: 4.7.52

Subhodeep Chakraborty
ee25btech11055,
IIT Hyderabad.

October 1, 2025

1. Problem
2. Solution
    - Plot
3. C Code
4. Python Code
    - Using shared objects
    - Plot
    - In pure Python
    - Plot

Problem Statement

If the points $(1, 1, p)$ and $(-3, 0, 1)$ be equidistant from the plane $\mathbf{r} \cdot \left(3\hat{\imath} + 4\hat{\jmath} - 12\hat{k}\right) + 13 = 0$, then find the value of p.

Given data points

Given:

$$\mathbf{A} = \begin{pmatrix} 1 \\ 1 \\ p \end{pmatrix} \tag{3.1}$$

$$\mathbf{B} = \begin{pmatrix} -3 \\ 0 \\ 1 \end{pmatrix} \tag{3.2}$$

$$\mathbf{S} : \begin{pmatrix} 3 & 4 & -12 \end{pmatrix} \cdot \mathbf{r} = -13 \tag{3.3}$$

Formulae

We know distance of point $\mathbf{P}$ from plane $\mathbf{n}^\top \mathbf{x} = c$ is given by:

$$d = \frac{|\mathbf{n}^\top \mathbf{P} - c|}{\|\mathbf{n}\|} \tag{3.4}$$

Solving

Thus

$$\frac{|\mathbf{n}^\top \mathbf{A} - c|}{\|\mathbf{n}\|} = \frac{|\mathbf{n}^\top \mathbf{B} - c|}{\|\mathbf{n}\|} \tag{3.5}$$

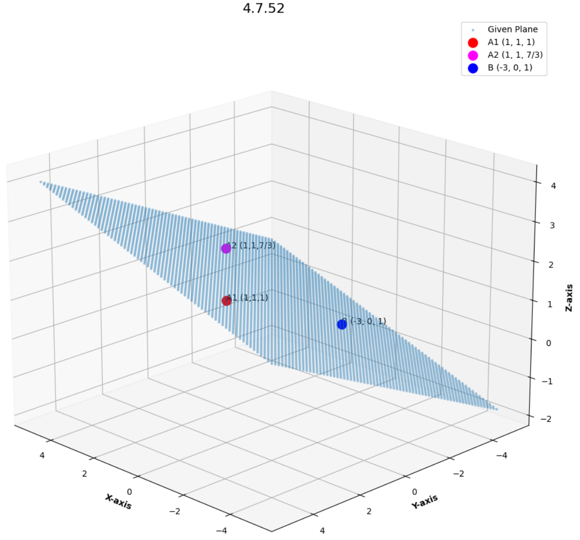
$$\mathbf{n}^\top \mathbf{A} = \mathbf{n}^\top \mathbf{B} \text{ OR } \mathbf{n}^\top \mathbf{A} = 2c - \mathbf{n}^\top \mathbf{B} \tag{3.6}$$

Substituting values

$$7 - 12p = -21 \textbf{ OR } 7 - 12p = -26 + 21 \tag{3.7}$$

$$p = 7/3 \textbf{ OR } p = 1 \tag{3.8}$$

Plot



C code for generating points on plane

```
void generate_plane_points(
    // Output params
    double* x_coords, double* y_coords, double* z_coords,
    // Grid params
    double x_min, double x_max, int x_steps,
    double y_min, double y_max, int y_steps,
    // Plane stuff
    double n1, double n2, double n3, double c) {

    double x_step_val = (x_max - x_min) / (x_steps - 1);
    double y_step_val = (y_max - y_min) / (y_steps - 1);
    int index = 0;
```

C code for generating points on a plane

```
for (int i = 0; i < x_steps; i++) {
    for (int j = 0; j < y_steps; j++) {
        double current_x = x_min + i * x_step_val;
        double current_y = y_min + j * y_step_val;
        double current_z;
        // Vertical plane check
        if ((c < 1e-9)&&(c > -1e-9)) {
            current_z = 0.0;
        } else {
            current_z = (-n1 * current_x - n2 * current_y + c) /
                n3;
        }
        x_coords[index] = current_x;
        y_coords[index] = current_y;
        z_coords[index] = current_z;
        index++;
    }
```

Python code for plotting using C

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

lib = ctypes.CDLL("./plane.so")
lib.generate_plane_points.argtypes = [
    ctypes.POINTER(ctypes.c_double),
    ctypes.POINTER(ctypes.c_double),
    ctypes.POINTER(ctypes.c_double),
    ctypes.c_double, ctypes.c_double, ctypes.c_int,
    ctypes.c_double, ctypes.c_double, ctypes.c_int,
    ctypes.c_double,
    ctypes.c_double,
    ctypes.c_double,
    ctypes.c_double,
]
lib.generate_plane_points.restype = None
```

```
A1 = np.array([1, 1, 1]).T
A2 = np.array([1, 1, 7 / 3]).T
B = np.array([-3, 0, 1]).T
n = np.array([3, 4, -12]).T
c = -13
x_steps, y_steps = 70, 70
total_points = x_steps * y_steps
x_plane = np.zeros(total_points, dtype=np.double)
y_plane = np.zeros(total_points, dtype=np.double)
z_plane = np.zeros(total_points, dtype=np.double)
lib.generate_plane_points(
    x_plane.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
    y_plane.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
    z_plane.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
    -5.0, 5.0, x_steps,
    -5.0, 5.0, y_steps,
    n[0], n[1], n[2],
    c,
)
```

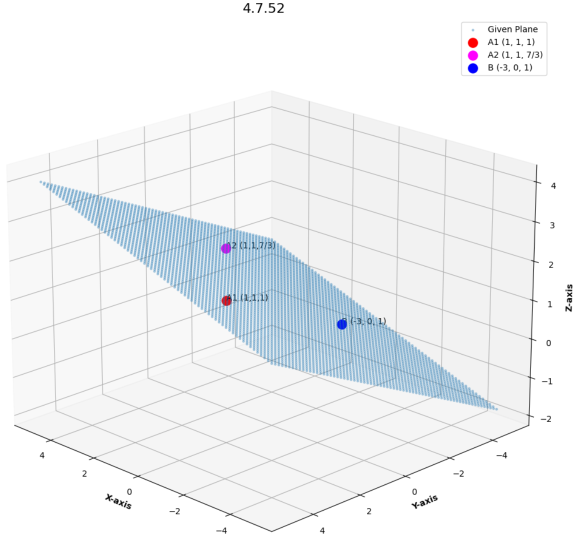
```
fig = plt.figure(figsize=(8, 6))
ax = fig.add_subplot(111, projection="3d")
t_values = np.linspace(0, 1, 100)
line_points_x, line_points_y, line_points_z = [], [], []
for t in t_values:
    result_arr = DoubleArray3()

    get_point(o, a, t, result_arr)

    line_points_x.append(result_arr[0])
    line_points_y.append(result_arr[1])
    line_points_z.append(result_arr[2])
ax.plot(
    line_points_x,
    line_points_y,
    line_points_z,
    color="red",
    label="a",
)
```

```
ax.set_xlabel("X-axis", fontweight="bold")
ax.set_ylabel("Y-axis", fontweight="bold")
ax.set_zlabel("Z-axis", fontweight="bold")
ax.set_title("4.7.52", fontsize=16)
ax.view_init(elev=20, azim=135)
ax.legend()
plt.tight_layout()
plt.savefig("../figs/plot.png")
plt.show()
```

Plot



Pure Python code

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

n = np.array([3, 4, -12]).T
d = -13
a = np.array([-3, 0, 1]).T
b = np.array([1, 1, 1]).T
c = np.array([1, 1, 7 / 3]).T

fig = plt.figure(figsize=(8, 8))
ax = fig.add_subplot(111, projection="3d")



x, y = np.meshgrid(range(10), range(10))
z = -(n[0] * x + n[1] * y - d) / n[2]
ax.plot_surface(x, y, z, alpha=0.7, color="gray")
```

Pure Python code

```
ax.scatter(a[0], a[1], a[2], color="red", label="a")
ax.scatter(b[0], b[1], b[2], color="blue", label="b")
ax.scatter(c[0], c[1], c[2], color="green", label="c")
ax.text(a[0], a[1], a[2], "Given Point")
ax.text(b[0], b[1], b[2], "Point 1")
ax.text(c[0], c[1], c[2], "Point 2")
ax.set_xlabel("X-axis")
ax.set_ylabel("Y-axis")
ax.set_zlabel("Z-axis")
ax.set_title("2.9.6")
ax.set_xlim([-5, 10])
ax.set_ylim([-5, 10])
ax.set_zlim([-5, 10])
ax.legend()
ax.grid(True)
plt.savefig("../figs/python.png")
plt.show()
```

Plot