2.10.85

Rushil Shanmukha Srinivas
EE25BTECH11057
Electrical Enggineering ,
IIT Hyderabad.

September 14, 2025

Problem Statement

Question : Let P be the plane 3x+2y+3z=16 and let S: $\alpha\hat{i} + \beta\hat{j} + \gamma\hat{k}$,where $\alpha + \beta + \gamma = 7$ and the distance of (α, β, γ) from the plane is $2/\sqrt{22}$ .Let $\mathbf{u}, \mathbf{v}, \mathbf{w}$ be three distinct vectors in S such that $|\mathbf{u} - \mathbf{v}| = |\mathbf{v} - \mathbf{w}| = |\mathbf{w} - \mathbf{u}|$. Let V be the volume of the parallelopiped determined by vectors $\mathbf{u}, \mathbf{v}, \mathbf{w}$. Then the value of (80/3)V is

Distance between point and plane,Line Equation

Solution :

$$P: \ \mathbf{n}^\top \mathbf{x} = c, \qquad \mathbf{n} = \begin{pmatrix} 3 \\ 2 \\ 3 \end{pmatrix} \ c = 16. \tag{3.1}$$

The distance of point $\mathbf{P_0} = \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix}$ from the plane is

$$\mathrm{dist}(\mathbf{P_0}, P) = \frac{|\mathbf{n}^\top \mathbf{P_0} - c|}{\|\mathbf{n}\|} \tag{3.2}$$

Given $\alpha + \beta + \gamma = 7$ and $\mathrm{dist}(\mathbf{P_0}, P) = \frac{2}{\sqrt{22}}$, we have

$$\frac{|3\alpha + 2\beta + 3\gamma - 16|}{\sqrt{22}} = \frac{2}{\sqrt{22}} \implies \mathbf{n}^\top \mathbf{P_0} = 18 \ \text{ or } \ \mathbf{n}^\top \mathbf{P_0} = 14. \tag{3.3}$$

Thus S lies on the intersections

$$\Pi: \ \mathbf{m}^\top \mathbf{x} = 7, \qquad \mathbf{m} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \tag{3.4}$$

with

$$P_+ : \mathbf{n}^\top \mathbf{x} = 18, \qquad P_- : \mathbf{n}^\top \mathbf{x} = 14. \tag{3.5}$$

textbf(i) Intersection of $\mathbf{m}^\top \mathbf{x} = 7$ and $\mathbf{n}^\top \mathbf{x} = 18$.
Write the augmented system in matrix form:

$$\left(\begin{array}{ccc|c} 1 & 1 & 1 & 7 \\ 3 & 2 & 3 & 18 \end{array}\right) \xrightarrow{R_2 \longrightarrow R_2 - 3R_1} \left(\begin{array}{ccc|c} 1 & 1 & 1 & 7 \\ 0 & -1 & 0 & -3 \end{array}\right). \tag{3.6}$$

From the second row we get $y = 3$. Substitute into $x + y + z = 7$:

$$x + 3 + z = 7 \quad \Longrightarrow \quad z = 4 - x. \tag{3.7}$$

So the line is

$$\mathbf{L_1} = \begin{pmatrix} 0 \\ 3 \\ 4 \end{pmatrix} + k_1 \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} = \mathbf{a} + k_1 \mathbf{d} \tag{3.8}$$

(ii) Intersection of $\mathbf{m}^\top \mathbf{x} = 7$ and $\mathbf{n}^\top \mathbf{x} = 14$.
The augmented matrix is

$$\left(\begin{array}{ccc|c} 1 & 1 & 1 & 7 \\ 3 & 2 & 3 & 14 \end{array}\right) \xrightarrow{R_2 \longrightarrow R_2 - 3R_1} \left(\begin{array}{ccc|c} 1 & 1 & 1 & 7 \\ 0 & -1 & 0 & -7 \end{array}\right). \tag{3.9}$$

So $y = 7$. From $x + y + z = 7$ we get

$$x + 7 + z = 7 \quad \Longrightarrow \quad z = -x. \tag{3.10}$$

$$\mathbf{L_2} = \begin{pmatrix} 0 \\ 7 \\ 0 \end{pmatrix} + k_2 \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} = \mathbf{b} + k_2 \mathbf{d} \tag{3.11}$$

hence the two lines are parallel.

Least Squares Solution

(iii) Perpendicular distance between the two parallel lines:
$\mathbf{P} = \mathbf{a} + p_1\mathbf{d}$ *and* $\mathbf{Q} = \mathbf{b} + p_2\mathbf{d}$ *be* 2*points on* $\mathbf{L_1}, \mathbf{L_2}$.

$$perpendicular\ distance = \operatorname{dist}(\mathbf{P}, \mathbf{Q}), \mathbf{M} = \begin{pmatrix} \mathbf{d} & \mathbf{d} \end{pmatrix} \tag{3.12}$$

$$\mathbf{M}^\top (a-b) + \mathbf{M}^\top \mathbf{M} \begin{pmatrix} p_1 \\ -p_2 \end{pmatrix} = 0, \tag{3.13}$$

$$\begin{pmatrix} \mathbf{d} \\ \mathbf{d} \end{pmatrix} (a-b) + \begin{pmatrix} \mathbf{d} \\ \mathbf{d} \end{pmatrix} \begin{pmatrix} \mathbf{d} & \mathbf{d} \end{pmatrix} \begin{pmatrix} p_1 \\ -p_2 \end{pmatrix} = 0 \tag{3.14}$$

On solving we get ,

$$p_1 - p_2 = 2\ take\ p_1 = 2\ and\ p_2 = 0 \tag{3.15}$$

Points are $\mathbf{P} = \begin{pmatrix} 2 \\ 3 \\ 2 \end{pmatrix}, \mathbf{Q} = \begin{pmatrix} 0 \\ 7 \\ 0 \end{pmatrix}, \mathbf{P} - \mathbf{Q} = \begin{pmatrix} 2 \\ -4 \\ 2 \end{pmatrix}$

$$Distance = D = \|\mathbf{P} - \mathbf{Q}\| = \sqrt{24} \tag{3.16}$$

Area and Volume

(iv) Area of the equilateral triangle formed by $\mathbf{u}, \mathbf{v}, \mathbf{w}$: As the two lines are parallel and let s = length of side of triangle

$$D = \frac{\sqrt{3}s}{2} \Longrightarrow s = 4\sqrt{2} \tag{3.17}$$

$$Area\ of\ the\ equilateral\ triangle = A = \frac{\sqrt{3}s^2}{4} = 8\sqrt{3} \tag{3.18}$$

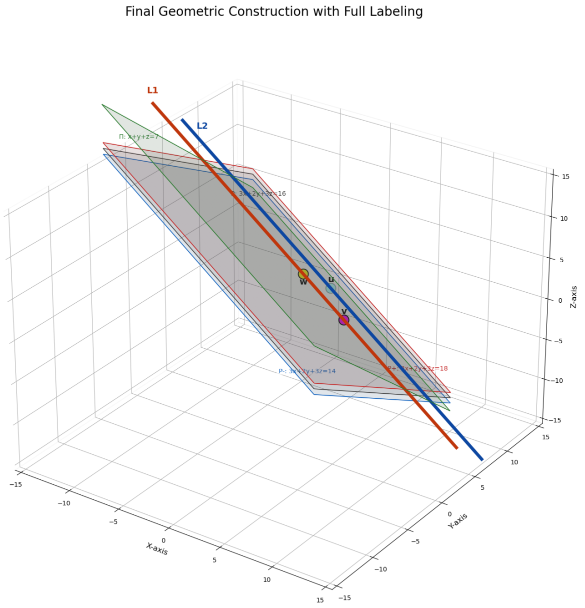
(v) Volume of the parallelepiped determined by three vectors $\mathbf{u}, \mathbf{v}, \mathbf{w}$:
$Volume\ of\ Parallelepiped = 6(Volume\ of\ Tetrahedron) = 2 \times base\ area \times height$

$$height = h = \frac{|\mathbf{m}^\top \mathbf{O} - c|}{\|\mathbf{m}\|} = \frac{|0-7|}{\sqrt{3}} = \frac{7}{\sqrt{3}} \tag{3.19}$$

$$Volume = 2 \times 8\sqrt{3} \times \frac{7}{\sqrt{3}} = 112 \tag{3.20}$$

$$\frac{80}{3}V = \frac{80}{3} \times 112 = \frac{8960}{3}. \tag{3.21}$$

Plots


Final Geometric Construction with Full Labeling
L1
L2
Π: x+y+z=7
P: 3x+2y+3z=16
P-: 3x+2y+3z=14
P+: 3x+2y+3z=18
w
u
v
X-axis
Y-axis
Z-axis
−15
−10
−5
0
5
10
15

C Code

```
#include <stdio.h>
#include <math.h>

// Function to compute determinant of 3x3 matrix
double determinant(double a[3][3]) {
    return a[0][0]*(a[1][1]*a[2][2] - a[1][2]*a[2][1])
        - a[0][1]*(a[1][0]*a[2][2] - a[1][2]*a[2][0])
        + a[0][2]*(a[1][0]*a[2][1] - a[1][1]*a[2][0]);}
int main() {
    // Two planes:
    // Plane P: 3x+2y+3z=16
    // Plane S: x+y+z=7
    // Solve intersection line between P and S
    // Choose basis for S: let
    // a = (1,-1,0), b = (1,0,-1)  both satisfy x+y+z=0
```

```
// So general point on S: (7,0,0) + s*a + t*b
// Constraint from P: 3x+2y+3z = 14 or 18 (distance condition)

// Pick "14" first.
// Solve quickly: Intersection point
double u[3] = {2,3,2};
double v[3] = {-2,3,6};
double w[3] = {-2,7,2};

// Put in matrix
double M[3][3] = {
    {u[0], v[0], w[0]},
    {u[1], v[1], w[1]},
    {u[2], v[2], w[2]}};
// Determinant
double det = determinant(M);
double V = fabs(det);
double result = (80.0/3.0) * V;
```

```
    printf("Chosen vectors:\n");
    printf("u = (%.2lf, %.2lf, %.2lf)\n", u[0], u[1], u[2]);
    printf("v = (%.2lf, %.2lf, %.2lf)\n", v[0], v[1], v[2]);
    printf("w = (%.2lf, %.2lf, %.2lf)\n", w[0], w[1], w[2]);
    printf("Determinant = %.2lf\n", det);
    printf("Volume V = %.2lf\n", V);
    printf("Final (80/3)*V = %.2lf\n", result);

    return 0;
}
```

Python : call_c.py

```
import ctypes
import os

# Load shared library
lib = ctypes.CDLL(os.path.abspath("./libvolume.so"))

# Set return types
lib.get_determinant.restype = ctypes.c_double
lib.get_volume.restype = ctypes.c_double
lib.compute_volume.restype = ctypes.c_double
lib.get_u.restype = ctypes.c_double
lib.get_v.restype = ctypes.c_double
lib.get_w.restype = ctypes.c_double

# Fetch vectors u, v, w
u = [lib.get_u(i) for i in range(3)]
v = [lib.get_v(i) for i in range(3)]
```

```
w = [lib.get_w(i) for i in range(3)]

# Fetch determinant, volume, and final result
det = lib.get_determinant()
V = lib.get_volume()
result = lib.compute_volume()

# Print everything
print("Vector u =", u)
print("Vector v =", v)
print("Vector w =", w)
print("Determinant =", det)
print("Volume V =", V)
print("Final (80/3)*V =", result)
```

Python Code for Plotting

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import art3d

# --- Create the Figure ---
fig = plt.figure(figsize=(16, 14))
ax = fig.add_subplot(111, projection='3d')

# --- Generate Grid and Define Boundaries ---
lim = 10
x_range = np.linspace(-lim, lim, 50)
y_range = np.linspace(-lim, lim, 50)
X, Y = np.meshgrid(x_range, y_range)

# --- AESTHETIC AND COLOR DEFINITIONS ---
# Define surface colors and darker corresponding outline colors
```

```
colors = {
    'P+': {'surface': '#E57373', 'outline': '#C62828'}, # Light Red /
        Dark Red
    'P−': {'surface': '#64B5F6', 'outline': '#1565C0'}, # Light Blue /
        Dark Blue
    '': {'surface': '#81C784', 'outline': '#2E7D32'}, # Light Green /
        Dark Green
    'P': {'surface': '#BDBDBD', 'outline': '#424242'} # Light Grey /
        Dark Grey}
# −−− PLOTTING PLANES WITH SURFACES AND OUTLINES −−−
# Function to plot a plane with its surface and a dark outline
def plot_complete_plane(Z, color_pair):
    # Plot the highly transparent surface
    ax.plot_surface(X, Y, Z, alpha=0.15, color=color_pair['surface'],
        rstride=5, cstride=5, edgecolor='none')
    # Plot a crisp, dark wireframe outline
    ax.plot_wireframe(X, Y, Z, color=color_pair['outline'], linewidth=1.2,
        rstride=100, cstride=100)
```

```
# 1. Plane : x + y + z = 7
Z_Pi = 7 - X - Y
plot_complete_plane(Z_Pi, colors[''])
# 2. Plane P+: 3x+2y+3z=18
Z_P_plus = (18 - 3*X - 2*Y) / 3
plot_complete_plane(Z_P_plus, colors['P+'])
# 3. Plane P-: 3x+2y+3z=14
Z_P_minus = (14 - 3*X - 2*Y) / 3
plot_complete_plane(Z_P_minus, colors['P-'])
# 4. Original Plane P: 3x+2y+3z=16
Z_P = (16 - 3*X - 2*Y) / 3
plot_complete_plane(Z_P, colors['P'])

# --- EMPHASIZE AND LABEL LINES/POINTS ---
FOREGROUND_ZORDER = 10
t_start, t_end = -15, 15
t_lines = np.linspace(t_start, t_end, 100)
d = np.array([1, 0, -1]) # Direction vector
```

```
# Line L1
a = np.array([0, 3, 4])
L1 = a + t_lines[:, np.newaxis] * d
ax.plot(L1[:, 0], L1[:, 1], L1[:, 2], color='#BF360C', lw=5, zorder=
    FOREGROUND_ZORDER) # Deep Orange
# Line L2
b = np.array([0, 7, 0])
L2 = b + t_lines[:, np.newaxis] * d
ax.plot(L2[:, 0], L2[:, 1], L2[:, 2], color='#0D47A1', lw=5, zorder=
    FOREGROUND_ZORDER) # Deep Blue
# Points u, v, w
point_size = 250
u, s = b, 4 * np.sqrt(2)
t_proj = np.dot(u - a, d) / np.dot(d, d)
M = a + t_proj * d
dist_M_v = s/2
v = M + (dist_M_v / np.linalg.norm(d)) * d
w = M - (dist_M_v / np.linalg.norm(d)) * d
```

```
ax.scatter([u[0],v[0],w[0]], [u[1],v[1],w[1]], [u[2],v[2],w[2]],
           color=['cyan','magenta','yellow'], s=point_size, ec='black', lw
               =1.5, zorder=FOREGROUND_ZORDER + 1)

# --- ADDING ALL LABELS (PLANES, LINES, AND POINTS) ---
# Plane Labels
plane_label_props = {'ha':'center', 'va':'center', 'fontsize':10, 'bbox':dict(
    facecolor='white', alpha=0.7, ec='none', pad=0.2)}
ax.text(-8,-8, 7-(-8)-(-8), " : x+y+z=7 ", color=colors[''], ['outline'],
     **plane_label_props)
ax.text(8, 8, (18-3*8-2*8)/3, " P+: 3x+2y+3z=18 ", color=colors['P
    +']['outline'], **plane_label_props)
ax.text(8,-8, (14-3*8-2*(-8))/3, " P-: 3x+2y+3z=14 ", color=colors
    ['P-']['outline'], **plane_label_props)
ax.text(-8, 8, (16-3*(-8)-2*8)/3, " P: 3x+2y+3z=16 ", color=colors['
    P']['outline'], **plane_label_props)
# Line Endpoint Labels
line_label_props = {'ha':'center', 'va':'center', 'fontsize':14, 'fontweight':'
    bold'}
```

```
l1_start_pos = a + t_start * d
l2_start_pos = b + t_start * d
ax.text(l1_start_pos[0], l1_start_pos[1], l1_start_pos[2] + 1.5, "L1", color
    ='#BF360C', **line_label_props)
ax.text(l2_start_pos[0] + 2, l2_start_pos[1], l2_start_pos[2] , "L2", color
    ='#0D47A1', **line_label_props)
# Point Labels (u, v, w)
point_label_props = {'ha':'center', 'va':'bottom', 'fontsize':14, 'fontweight
    ':'bold'}
ax.text(u[0], u[1], u[2] + 0.5, 'u', color='black', **point_label_props)
ax.text(v[0], v[1], v[2] + 0.5, 'v', color='black', **point_label_props)
ax.text(w[0], w[1], w[2] - 1.5, 'w', color='black', **point_label_props) #
    slight offset for w

# --- FINAL PLOT SETUP ---
ax.view_init(elev=28, azim=-55)
```

```
ax.set_xlim(-15, 15); ax.set_ylim(-15, 15); ax.set_zlim(-15, 15)
ax.set_xlabel('X-axis', fontsize=12); ax.set_ylabel('Y-axis', fontsize=12);
      ax.set_zlabel('Z-axis', fontsize=12)
ax.set_title('Final Geometric Construction with Full Labeling', fontsize
     =20, pad=20)

# Clean background
ax.xaxis.pane.fill=False; ax.yaxis.pane.fill=False; ax.zaxis.pane.fill=False
ax.grid(True, linestyle=':', alpha=0.5)

plt.tight_layout()
plt.savefig("../figs/fig5_.png")
plt.show()
```