Matgeo Presentation - Problem 2.7.33

ee25btech11021 - Dhanush sagar

September 7, 2025

Problem Statement

A variable plane at a distance of one unit from the origin cuts the coordinate axes at A, B and C.
If the centroid $D(x, y, z)$ of triangle ABC satisfies the relation

$$\frac{1}{x^2} + \frac{1}{y^2} + \frac{1}{z^2} = k,$$

then the value of k is :

1) 3
2) 1
3) $\frac{1}{3}$
4) 9

solution

Let the plane meet the coordinate axes at

$$\mathbf{A} = \begin{pmatrix} a \\ 0 \\ 0 \end{pmatrix}, \mathbf{B} = \begin{pmatrix} 0 \\ b \\ 0 \end{pmatrix}, \mathbf{C} = \begin{pmatrix} 0 \\ 0 \\ c \end{pmatrix}.$$

Define

$$\mathbf{M} = \mathrm{diag}(a, b, c), \quad \mathbf{e} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

The normal vector of the plane is given by

$$\mathbf{n} = \mathbf{M}^{-1}\mathbf{e} = \begin{pmatrix} \frac{1}{a} \\ \frac{1}{b} \\ \frac{1}{c} \end{pmatrix} \tag{0.1}$$

The squared norm of the normal vector is

$$\|\mathbf{n}\|^2 = \mathbf{e}^T\mathbf{M}^{-2}\mathbf{e} = \frac{1}{a^2} + \frac{1}{b^2} + \frac{1}{c^2} \tag{0.2}$$

Problem Statement

The perpendicular distance of the plane from the origin is

$$d = \frac{|1|}{\|\mathbf{n}\|} = \frac{1}{\sqrt{\mathbf{e}^T \mathbf{M}^{-2} \mathbf{e}}} \tag{0.3}$$

Thus, the relation between a, b, c and d is

$$\mathbf{e}^T \mathbf{M}^{-2} \mathbf{e} = \frac{1}{d^2} \tag{0.4}$$

The centroid of the triangle ABC is

$$D = \frac{1}{3} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \frac{1}{3} \mathbf{M} \mathbf{e} \tag{0.5}$$

Hence, the coordinates of the centroid are

$$x = \tfrac{a}{3}, \quad y = \tfrac{b}{3}, \quad z = \tfrac{c}{3} \tag{0.6}$$

Now, the required expression is

$$\frac{1}{x^2} + \frac{1}{y^2} + \frac{1}{z^2} = \frac{1}{(a/3)^2} + \frac{1}{(b/3)^2} + \frac{1}{(c/3)^2} \tag{0.7}$$

Problem Statement

Simplifying, we get

$$\frac{1}{x^2} + \frac{1}{y^2} + \frac{1}{z^2} = 9\left(\frac{1}{a^2} + \frac{1}{b^2} + \frac{1}{c^2}\right) \tag{0.8}$$

In matrix form, this becomes

$$\frac{1}{x^2} + \frac{1}{y^2} + \frac{1}{z^2} = 9\,\mathbf{e}^T\mathbf{M}^{-2}\mathbf{e} \tag{0.9}$$

Using the relation obtained earlier,

$$\frac{1}{x^2} + \frac{1}{y^2} + \frac{1}{z^2} = \frac{9}{d^2} \tag{0.10}$$

For $d = 1$, we finally obtain

$$\frac{1}{x^2} + \frac{1}{y^2} + \frac{1}{z^2} = 9 \tag{0.11}$$

$$\boxed{k = 9}$$

C Source Code:plane points.c

```
#include <stdio.h>

typedef struct {
    double x, y, z;
} Point;
// Generate points array for A, B, C
void generate_plane_points(double a, double b, double c, doubl
    points_array[0] = a; points_array[1] = 0; points_array[2]
    points_array[3] = 0; points_array[4] = b; points_array[5]
    points_array[6] = 0; points_array[7] = 0; points_array[8]
}
// Compute k
double compute_plane_k(double a, double b, double c) {
    double x = a/3.0, y = b/3.0, z = c/3.0;
    return 1.0/(x*x) + 1.0/(y*y) + 1.0/(z*z);
}
```

Python Script: solve plane.py

```
import ctypes
import sympy as sp
import numpy as np
# --- 1. Load C library ---
lib = ctypes.CDLL("./plane_points.so")
lib.generate_plane_points.argtypes = [ctypes.c_double, ctypes.
lib.generate_plane_points.restype = None
# --- 2. Symbolic variables ---
a, b, c = sp.symbols('a b c', positive=True)
# --- 3. Compute centroid symbolically ---
A = sp.Matrix([a, 0, 0])
B = sp.Matrix([0, b, 0])
C = sp.Matrix([0, 0, c])
D = (A + B + C)/3
# --- 4. Compute k symbolically ---
k = 1/D[0]**2 + 1/D[1]**2 + 1/D[2]**2
# --- 5. Apply plane condition: 1/a^2 + 1/b^2 + 1/c^2 = 1 ---
```

Python Script: solve plane.py

```
plane_condition = 1/a**2 + 1/b**2 + 1/c**2
k_final = k.subs(plane_condition, 1)
print("Centroid D =", D)
print("k in terms of a,b,c =", k)
print("Using plane condition 1/a^2 + 1/b^2 + 1/c^2 = 1")
print("Final k =", k_final)
# --- 6. Optional: Call C program to generate points numerical
# Here we must provide numeric values to C, just as placeholde
points_array = (ctypes.c_double * 9)()
lib.generate_plane_points(1.0, 1.0, 1.0, points_array)
A_num = np.array(points_array[0:3])
B_num = np.array(points_array[3:6])
C_num = np.array(points_array[6:9])
print("\nPoints from C code (numeric placeholders):")
print("A =", A_num)
print("B =", B_num)
print("C =", C_num)
```

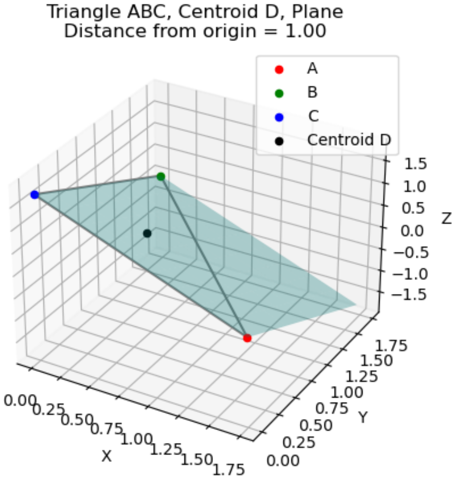
Python Script: plot plane.py

```
import matplotlib.pyplot as plt
import numpy as np
import ctypes
# Load C library
lib = ctypes.CDLL("./plane_points.so")
lib.generate_plane_points.argtypes = [ctypes.c_double, ctypes.
lib.generate_plane_points.restype = None
# --- Choose numeric a, b, c satisfying 1/a^2 + 1/b^2 + 1/c^2
a = b = c = np.sqrt(3)  # This ensures plane distance = 1
# Generate points
points_array = (ctypes.c_double * 9)()
lib.generate_plane_points(a, b, c, points_array)
A = np.array(points_array[0:3])
B = np.array(points_array[3:6])
C = np.array(points_array[6:9])
D = (A + B + C)/3  # centroid
# Plane distance from origin
```

Python Script: plot plane.py

```
distance = 1 / np.sqrt(1/a**2 + 1/b**2 + 1/c**2)
# Plot triangle, centroid, and plane
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
for P, color, label in zip([A, B, C, D], ['red','green','blue
    ax.scatter(*P, color=color, label=label)
# Triangle edges
for P1, P2 in [(A, B), (B, C), (C, A)]:
    ax.plot([P1[0], P2[0]], [P1[1], P2[1]], [P1[2], P2[2]], 'g
# Plane surface
xx, yy = np.meshgrid(np.linspace(0, a, 10), np.linspace(0, b,
zz = c*(1 - xx/a - yy/b)
ax.plot_surface(xx, yy, zz, alpha=0.3, color='cyan')
ax.set_xlabel('X'); ax.set_ylabel('Y'); ax.set_zlabel('Z')
ax.set_title(f'Triangle ABC, Centroid D, Plane\nDistance from
ax.legend()
plt.savefig('triangle_centroid_plane_distance.png')  plt.show(
```

Result Plot


Triangle ABC, Centroid D, Plane
Distance from origin = 1.00
A
B
C
Centroid D
X
Y
Z
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5