Matgeo Presentation - Problem 5.8.19

ee25btech11021 - Dhanush sagar

October 3, 2025

Problem Statement

Let a, b, c be real numbers. Consider the following system of equations in x, y, z:

$$\begin{aligned} \frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} &= 1, \\ \frac{x^2}{a^2} - \frac{y^2}{b^2} + \frac{z^2}{c^2} &= 1, \\ -\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} &= 1. \end{aligned}$$

The system has:

1) no solution
2) unique solution
3) infinitely many solutions
4) finitely many solutions

solution

Let

$$A = \frac{x^2}{a^2}, \tag{0.1}$$

$$B = \frac{y^2}{b^2}, \tag{0.2}$$

$$C = \frac{z^2}{c^2}. \tag{0.3}$$

Then the system becomes

$$A + B - C = 1, \tag{0.4}$$

$$A - B + C = 1, \tag{0.5}$$

$$-A + B + C = 1. \tag{0.6}$$

solution

The augmented matrix is

$$
\begin{aligned}
&\begin{pmatrix} 1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 \\ -1 & 1 & 1 & 1 \end{pmatrix} \xrightarrow{R_2 \to R_2 - R_1} \begin{pmatrix} 1 & 1 & -1 & 1 \\ 0 & -2 & 2 & 0 \\ -1 & 1 & 1 & 1 \end{pmatrix} \xrightarrow{R_3 \to R_3 + R_1} \begin{pmatrix} 1 & 1 \\ 0 & -2 \\ 0 & 2 \end{pmatrix} \\
&\xrightarrow{R_3 \to R_3 + R_2} \begin{pmatrix} 1 & 1 & -1 & 1 \\ 0 & -2 & 2 & 0 \\ 0 & 0 & 2 & 2 \end{pmatrix} \xrightarrow{R_2 \to -\frac{1}{2} R_2} \begin{pmatrix} 1 & 1 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 2 \end{pmatrix} \\
&\xrightarrow{R_3 \to \frac{1}{2} R_3} \begin{pmatrix} 1 & 1 & -1 & 1 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \xrightarrow{R_2 \to R_2 + R_3} \begin{pmatrix} 1 & 1 & -1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \\
&\xrightarrow{R_1 \to R_1 + R_3} \begin{pmatrix} 1 & 1 & 0 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \xrightarrow{R_1 \to R_1 - R_2} \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}.
\end{aligned}
\tag{0.7}
$$

solution

From the final matrix we read

$$A = 1, \qquad B = 1, \qquad C = 1. \tag{0.8}$$

Therefore,

$$\frac{x^2}{a^2} = 1, \qquad \frac{y^2}{b^2} = 1, \qquad \frac{z^2}{c^2} = 1, \tag{0.9}$$

which gives

$$x = \pm a, \qquad y = \pm b, \qquad z = \pm c. \tag{0.10}$$

Hence there are $2^3 = 8$ distinct solutions for (x, y, z), so the correct choice is

$$\boxed{\text{(d) finitely many solutions}} \tag{0.11}$$

C Source Code:fraction matrix.c

```
#include <stdio.h>

void gen_system_points(double *A, double *B) {
    // Coefficient matrix (3x3)
    double tempA[9] = {
         1,  1, -1,   // Eqn 1
         1, -1,  1,   // Eqn 2
        -1,  1,  1    // Eqn 3
    };
    // RHS vector
    double tempB[3] = {1, 1, 1};

    for (int i = 0; i < 9; i++) A[i] = tempA[i];
    for (int i = 0; i < 3; i++) B[i] = tempB[i];
}
```

Python Script:fraction matrix.py

```
import ctypes
import numpy as np
import itertools
# --- Load C shared library ---
lib = ctypes.CDLL("./gen_system_points.so")
lib.gen_system_points.argtypes = [ctypes.POINTER(ctypes.c_doub
# Storage
A_storage = (ctypes.c_double * 9)()
B_storage = (ctypes.c_double * 3)()
lib.gen_system_points(A_storage, B_storage)
# Convert to numpy
A = np.array(A_storage).reshape(3, 3)
B = np.array(B_storage)
print("Coefficient matrix A:\n", A)
print("RHS vector B:\n", B)
# Solve AX = B
X, Y, Z = np.linalg.solve(A, B)
```

Python Script:fraction matrix.py

```
print("\nSolution (X,Y,Z) = ({}, {}, {})".format(X, Y, Z))
# Symbolic answer
symbolic_points = []
for sx, sy, sz in itertools.product([1, -1], repeat=3):
    symbolic_points.append((f"{'+' if sx>0 else '-'}a",
                            f"{'+' if sy>0 else '-'}b",
                            f"{'+' if sz>0 else '-'}c"))
print("\nSymbolic solutions (in terms of a,b,c):")
for p in symbolic_points:
    print(p)
a, b, c = 2.0, 3.0, 1.0   # change these values
numeric_points = [(sx*a, sy*b, sz*c) for sx, sy, sz in itertoo
print("\nNumeric solutions (for a={}, b={}, c={}):".format(a,
for p in numeric_points:
    print(p)
np.savetxt("points_abc.txt", numeric_points)
print("\nSaved numeric points to points_abc.txt")
```

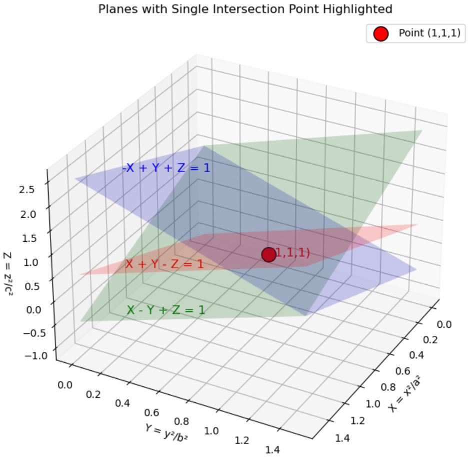
Python Script: plot matrix.py

```
import numpy as np
import matplotlib.pyplot as plt
# Parameters
a = b = c = 1.0  # main intersection point
# Meshgrid for planes
xx = np.linspace(0, 1.5, 30)
yy = np.linspace(0, 1.5, 30)
XX, YY = np.meshgrid(xx, yy)
# Plane equations
Z1 = XX + YY - 1       # X + Y - Z = 1
Z2 = 1 - XX + YY       # X - Y + Z = 1
Z3 = 1 + XX - YY       # -X + Y + Z = 1
# Plot
fig = plt.figure(figsize=(10,8))
ax = fig.add_subplot(111, projection='3d')
# Plot planes with low opacity
ax.plot_surface(XX, YY, Z1, color="red", alpha=0.2)
```

Python Script: plot matrix.py

```
ax.plot_surface(XX, YY, Z2, color="green", alpha=0.2)
ax.plot_surface(XX, YY, Z3, color="blue", alpha=0.2)
# Add plane equations as text
ax.text(1.3, 0.2, Z1[0,-1], "X + Y - Z = 1", color='red', font
ax.text(1.3, 0.2, Z2[0,-1], "X - Y + Z = 1", color='green', fo
ax.text(1.3, 0.2, Z3[0,-1], "-X + Y + Z = 1", color='blue', fo
# Plot the single intersection point (1,1,1)
ax.scatter(1,1,1, color='red', s=200, edgecolor='black', label
# Annotate the intersection point
ax.text(1.02, 1.02, 1.02, "(1,1,1)", fontsize=12, color='red')
# Axes labels
ax.set_xlabel("X = x²/a²")
ax.set_ylabel("Y = y²/b²")
ax.set_zlabel("Z = z²/c²")
ax.set_title("Planes with Single Intersection Point Highlighte
ax.legend()
plt.show()
```

Result Plot



Planes with Single Intersection Point Highlighted
Point (1,1,1)
-X + Y + Z = 1
X + Y - Z = 1
X - Y + Z = 1
(1,1,1)
Z = z²/c²
Y = y²/b²
X = x²/a²