Matgeo Presentation - Problem 2.7.33

ee25btech11021 - Dhanush sagar

September 18, 2025

Problem Statement

Find the equation of the plane containing the two parallel lines $\frac{x-1}{2} = \frac{y+1}{-1} = \frac{z}{3}$ and $\frac{x}{4} = \frac{y-2}{-2} = \frac{z+1}{6}$. Also, determine whether the plane thus obtained contains the line $\frac{x-2}{3} = \frac{y-1}{1} = \frac{z-2}{5}$.

solution

A plane can be written in matrix form as $\mathbf{n}^T(\mathbf{r}-\mathbf{P})=0$, where $\mathbf{n}$ is the normal vector, $\mathbf{r}$ is a general point on the plane, and $\mathbf{P}$ is a point on the plane:

$$\mathbf{n}^T(\mathbf{r}-\mathbf{P})=0 \tag{0.1}$$

The given lines are in symmetric form:

$$\text{L1} = \frac{x-1}{2} = \frac{y+1}{-1} = \frac{z}{3} \tag{0.2}$$

$$\text{L2} = \frac{x}{4} = \frac{y-2}{-2} = \frac{z+1}{6} \tag{0.3}$$

$$\text{L3} = \frac{x-2}{3} = \frac{y-1}{1} = \frac{z-2}{5} \tag{0.4}$$

Extract points and directions from L1 and L2. The vector joining points from the two lines is:

$$\mathbf{P}_1 = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}, \quad \mathbf{d}_1 = \begin{pmatrix} 2 \\ -1 \\ 3 \end{pmatrix}, \quad \mathbf{P}_2 = \begin{pmatrix} 0 \\ 2 \\ -1 \end{pmatrix}, \quad \mathbf{v} = \mathbf{P}_2 - \mathbf{P}_1 = \begin{pmatrix} -1 \\ 3 \\ -1 \end{pmatrix} \tag{0.5}$$

Problem Statement

The plane's normal vector is orthogonal to both $\mathbf{d}_1$ and $\mathbf{v}$, so it lies in the nullspace of the constraint matrix:

$$\mathbf{A} = \begin{pmatrix} 2 & -1 & 3 \\ -1 & 3 & -1 \end{pmatrix} \tag{0.6}$$

Row-reduction to find the nullspace:

$$\mathbf{A} \xrightarrow{R_1 \to \frac{1}{2} R_1} \begin{pmatrix} 1 & -\frac{1}{2} & \frac{3}{2} \\ -1 & 3 & -1 \end{pmatrix} \tag{0.7}$$

$$\mathbf{A} \xrightarrow{R_2 \to R_2 + R_1} \begin{pmatrix} 1 & -\frac{1}{2} & \frac{3}{2} \\ 0 & \frac{5}{2} & \frac{1}{2} \end{pmatrix} \tag{0.8}$$

$$\mathbf{A} \xrightarrow{R_2 \to \frac{2}{5} R_2} \begin{pmatrix} 1 & -\frac{1}{2} & \frac{3}{2} \\ 0 & 1 & \frac{1}{5} \end{pmatrix} \tag{0.9}$$

$$\mathbf{A} \xrightarrow{R_1 \to R_1 + \frac{1}{2} R_2} \begin{pmatrix} 1 & 0 & \frac{8}{5} \\ 0 & 1 & \frac{1}{5} \end{pmatrix} \tag{0.10}$$

Problem Statement

Now, we need to form a vector $\mathbf{n}_0$ whose product with $\mathbf{A}$ gives a null vector.

Express leading variables in terms of the free variable n_3 to get a vector in the nullspace, which is the plane's normal vector:

$$n_1 + \frac{8}{5}n_3 = 0 \quad \Rightarrow \quad n_1 = -\frac{8}{5}n_3 \tag{0.11}$$

$$n_2 + \frac{1}{5}n_3 = 0 \quad \Rightarrow \quad n_2 = -\frac{1}{5}n_3 \tag{0.12}$$

Let $n_3 = 1$ for simplicity:

$$\mathbf{n}_0 = \begin{pmatrix} -\frac{8}{5} \\ -\frac{1}{5} \\ 1 \end{pmatrix} \tag{0.13}$$

Clearing denominators and adjusting the sign gives the normal vector:

$$\mathbf{n} = \begin{pmatrix} 8 \\ 1 \\ -5 \end{pmatrix} \tag{0.14}$$

Problem Statement

Let $\mathbf{r}$ be a general point on the plane:

$$\mathbf{r} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} \tag{0.15}$$

The plane equation using point $\mathbf{P}_1$ and normal vector $\mathbf{n}$:

$$\mathbf{n}^T(\mathbf{r} - \mathbf{P}_1) = 0 \tag{0.16}$$

$$\mathbf{n}^T\mathbf{P}_1 = \begin{pmatrix} 8 & 1 & -5 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix} = 7 \tag{0.17}$$

$$\boxed{\begin{pmatrix} 8 & 1 & -5 \end{pmatrix} \mathbf{r} = 7} \tag{0.18}$$

Problem Statement

Check if the third line L3 lies in the plane by verifying the point and direction:

$$\mathbf{P}_3 = \begin{pmatrix} 2 \\ 1 \\ 2 \end{pmatrix}, \quad \mathbf{d}_3 = \begin{pmatrix} 3 \\ 1 \\ 5 \end{pmatrix} \tag{0.19}$$

$$\mathbf{n}^T \mathbf{P}_3 = \begin{pmatrix} 8 & 1 & -5 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \\ 2 \end{pmatrix} = 7 \tag{0.20}$$

$$\mathbf{n}^T \mathbf{d}_3 = \begin{pmatrix} 8 & 1 & -5 \end{pmatrix} \begin{pmatrix} 3 \\ 1 \\ 5 \end{pmatrix} = 0 \tag{0.21}$$

Therefore, the plane containing the first two lines has the matrix form:

$$\begin{pmatrix} 8 & 1 & -5 \end{pmatrix} \mathbf{r} = 7$$

and it also contains the third line.

C Source Code:line data.c

```
#include <stdio.h>
typedef struct {
    double x, y, z;
} Vec3;
// Function to return points and direction vectors for lines
void get_lines(Vec3* P1, Vec3* d1, Vec3* P2, Vec3* d2, Vec3* P
    // Line 1
    P1->x = 1; P1->y = -1; P1->z = 0;
    d1->x = 2; d1->y = -1; d1->z = 3;
    // Line 2
    P2->x = 0; P2->y = 2; P2->z = -1;
    d2->x = 4; d2->y = -2; d2->z = 6;
    // Line 3
    P3->x = 2; P3->y = 1; P3->z = 2;
    d3->x = 3; d3->y = 1; d3->z = 5;
}
```

Python Script:plane solver.py

```
import ctypes
import numpy as np
# Load shared library
lib = ctypes.CDLL("./libline_data.so")
# Define Vec3 struct
class Vec3(ctypes.Structure):
    _fields_ = [("x", ctypes.c_double),
                ("y", ctypes.c_double),
                ("z", ctypes.c_double)]
# Create instances
P1 = Vec3(); d1 = Vec3()
P2 = Vec3(); d2 = Vec3()
P3 = Vec3(); d3 = Vec3()
# Populate points and directions
lib.get_lines(ctypes.byref(P1), ctypes.byref(d1), ctypes.byref
```

Python Script:plane solver.py

```
# Convert to numpy arrays
P1 = np.array([P1.x, P1.y, P1.z])
d1 = np.array([d1.x, d1.y, d1.z])
P2 = np.array([P2.x, P2.y, P2.z])
d2 = np.array([d2.x, d2.y, d2.z])
P3 = np.array([P3.x, P3.y, P3.z])
d3 = np.array([d3.x, d3.y, d3.z])
# Vector between points on first two lines
v = P2 - P1
# Constraint matrix
A = np.array([d1, v])
# Null space to find plane normal
U, S, Vt = np.linalg.svd(A)
n = Vt.T[:, -1]
# Scale normal for simplicity
n = n / n[-1]
```

Python Script:plane solver.py

```
# Plane equation: n . r = n . P1
d_plane = np.dot(n, P1)
print("Plane normal:", n)
print("Plane equation: {}*x + {}*y + {}*z = {}".format(n[0], n
# Check if third line lies on plane
dot_point = np.dot(n, P3) - d_plane
dot_dir = np.dot(n, d3)
if abs(dot_point) < 1e-6 and abs(dot_dir) < 1e-6:
    print("Line 3 lies on the plane")
else:
    print("Line 3 does NOT lie on the plane")
```

Python Script: plot lines plane.py

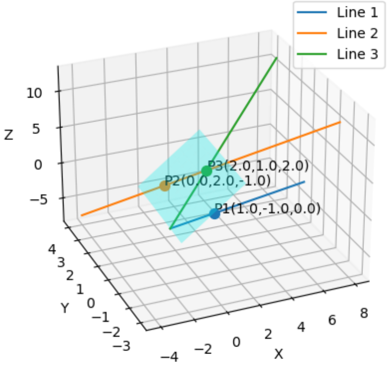
```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import numpy as np
from plane_solver import P1, d1, P2, d2, P3, d3, n, d_plane  #
points = [P1, P2, P3]
directions = [d1, d2, d3]
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
# Plot lines
t = np.linspace(-1, 2, 50)
for i, (P, d) in enumerate(zip(points, directions)):
    X = P[0] + d[0]*t
    Y = P[1] + d[1]*t
    Z = P[2] + d[2]*t
    ax.plot(X, Y, Z, label=f'Line {i+1}')
    ax.scatter(P[0], P[1], P[2], s=50)
    ax.text(P[0], P[1], P[2], f'P{i+1}({P[0]},{P[1]},{P[2]})')
```

Python Script: plot lines plane.py

```
# Plot plane
xx, yy = np.meshgrid(np.linspace(-1,3,10), np.linspace(-1,3,1(
zz = (d_plane - n[0]*xx - n[1]*yy)/n[2]
ax.plot_surface(xx, yy, zz, alpha=0.3, color='cyan')

ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.legend()
plt.savefig("lines_and_plane.png", dpi=300, bbox_inches='tight
plt.show()
```

Result Plot



Line 1
Line 2
Line 3
P3(2.0,1.0,2.0)
P2(0.0,2.0,-1.0)
P1(1.0,-1.0,0.0)
Z
10
5
0
−5
Y
4
3
2
1
0
−1
−2
−3
X
−4
−2
0
2
4
6
8