4.4.12-Beamer

Varun-ai25btech11016

September 1, 2025

Question

Find the equation of the plane passing through the points $(2, 5, -3)$, $(-2, -3, 5)$ and $(5, 3, -3)$. Also find the point of intersection of this plane with the line passing through points $(3, 1, 5)$ and $(-1, -3, -1)$.

Theoretical Solution

Let the vectors be

$$\mathbf{A} = \begin{pmatrix} 2 \\ 5 \\ -3 \end{pmatrix}, \qquad \mathbf{B} = \begin{pmatrix} -2 \\ -3 \\ 5 \end{pmatrix}, \qquad \mathbf{C} = \begin{pmatrix} 5 \\ 3 \\ -3 \end{pmatrix}. \tag{1}$$

The vectors lying on the plane are

$$\mathbf{AB} = \mathbf{B} - \mathbf{A} = \begin{pmatrix} -2 \\ -3 \\ 5 \end{pmatrix} - \begin{pmatrix} 2 \\ 5 \\ -3 \end{pmatrix} = \begin{pmatrix} -4 \\ -8 \\ 8 \end{pmatrix}, \tag{2}$$

$$\mathbf{AC} = \mathbf{C} - \mathbf{A} = \begin{pmatrix} 5 \\ 3 \\ -3 \end{pmatrix} - \begin{pmatrix} 2 \\ 5 \\ -3 \end{pmatrix} = \begin{pmatrix} 3 \\ -2 \\ 0 \end{pmatrix}. \tag{3}$$

Theoretical Solution

The normal vector to the plane is given by the cross product

$$\mathbf{n} = \mathbf{AB} \times \mathbf{AC} \tag{4}$$
$$= \begin{pmatrix} -4 \\ -8 \\ 8 \end{pmatrix} \times \begin{pmatrix} 3 \\ -2 \\ 0 \end{pmatrix}.$$

Expanding the determinant,

$$\mathbf{n} = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ -4 & -8 & 8 \\ 3 & -2 & 0 \end{vmatrix} \tag{5}$$
$$\mathbf{n} = \begin{pmatrix} 16 \\ 24 \\ 32 \end{pmatrix}.$$

Theoretical Solution

Equation of the plane passing through $\mathbf{A}$ is

$$\mathbf{n}^\top(\mathbf{x}-\mathbf{A}) = 0, \tag{6}$$

$$\begin{pmatrix} 16 & 24 & 32 \end{pmatrix}\left(\begin{pmatrix} x \\ y \\ z \end{pmatrix} - \begin{pmatrix} 2 \\ 5 \\ -3 \end{pmatrix}\right) = 0.$$

Hence, the equation of the plane is

$$2x + 3y + 4z = 7. \tag{7}$$

Now, the line passes through

$$\mathbf{P} = \begin{pmatrix} 3 \\ 1 \\ 5 \end{pmatrix}, \quad \mathbf{Q} = \begin{pmatrix} -1 \\ -3 \\ -1 \end{pmatrix}. \tag{8}$$

Theoretical Solution

The direction vector is

$$\mathbf{d} = \mathbf{Q} - \mathbf{P} = \begin{pmatrix} -4 \\ -4 \\ -6 \end{pmatrix}. \tag{9}$$

Thus, the parametric equation of the line is

$$\mathbf{r} = \mathbf{P} + \lambda \mathbf{d}, \tag{10}$$

$$\mathbf{r} = \begin{pmatrix} 3 - 4\lambda \\ 1 - 4\lambda \\ 5 - 6\lambda \end{pmatrix}. \tag{11}$$

Substitute into the plane equation:

$$2x + 3y + 4z = 7 \tag{12}$$

$$2(3 - 4\lambda) + 3(1 - 4\lambda) + 4(5 - 6\lambda) = 7, \tag{13}$$

$$\lambda = \frac{1}{2}. \tag{14}$$

Theoretical Solution

Thus, the point of intersection is

$$\mathbf{r} = \begin{pmatrix} 3 - 4\left(\frac{1}{2}\right) \\ 1 - 4\left(\frac{1}{2}\right) \\ 5 - 6\left(\frac{1}{2}\right) \end{pmatrix} \tag{15}$$

$$= \begin{pmatrix} 1 \\ -1 \\ 2 \end{pmatrix}. \tag{16}$$

Conclusion

Final Answer:

The plane equation is $2x + 3y + 4z = 7$,

the point of intersection is $\begin{pmatrix} 1 \\ -1 \\ 2 \end{pmatrix}$.

C code

```c
#include <stdio.h>

typedef struct {
    double x, y, z;
} Vec3;

// Cross product
Vec3 cross(Vec3 a, Vec3 b) {
    Vec3 res;
    res.x = a.y*b.z - a.z*b.y;
    res.y = a.z*b.x - a.x*b.z;
    res.z = a.x*b.y - a.y*b.x;
    return res;
}
```

C code

```c

// Dot product
double dot(Vec3 a, Vec3 b) {
    return a.x*b.x + a.y*b.y + a.z*b.z;
}

// Subtraction
Vec3 sub(Vec3 a, Vec3 b) {
    Vec3 res = {a.x-b.x, a.y-b.y, a.z-b.z};
    return res;
}

// Addition
Vec3 add(Vec3 a, Vec3 b) {
    Vec3 res = {a.x+b.x, a.y+b.y, a.z+b.z};
    return res;
}
```

C code

```c
// Scalar multiply
Vec3 mul(Vec3 a, double s) {
    Vec3 res = {a.x*s, a.y*s, a.z*s};
    return res;
}

/*
   Function: find_intersection
   Input: points A, B, C (plane), P, Q (line)
   Output: intersection point (returned as Vec3)
*/
Vec3 find_intersection(Vec3 A, Vec3 B, Vec3 C, Vec3 P, Vec3 Q) {
    // Plane normal
    Vec3 AB = sub(B, A);
    Vec3 AC = sub(C, A);
    Vec3 n = cross(AB, AC);
```

C code

```
    // Plane constant
    double d = -dot(n, A);

    // Line direction
    Vec3 d_line = sub(Q, P);

    // Solve n.(P + t*d_line) + d = 0
    double t = -(dot(n, P) + d) / dot(n, d_line);

    // Intersection point
    Vec3 inter = add(P, mul(d_line, t));
    return inter;
}
```

C plus Python code

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

# Load the C shared library (compile first: gcc -shared -o
    geometry.so -fPIC geometry.c)
lib = ctypes.CDLL(./geometry.so)

# Define Vec3 struct (same as in C)
class Vec3(ctypes.Structure):
    _fields_ = [(x, ctypes.c_double),
                (y, ctypes.c_double),
                (z, ctypes.c_double)]
```

C plus Python code

```
# Configure function
lib.find_intersection.argtypes = [Vec3, Vec3, Vec3, Vec3, Vec3]
lib.find_intersection.restype = Vec3

# Define points
A = Vec3(2, 5, -3)
B = Vec3(-2, -3, 5)
C = Vec3(5, 3, -3)
P = Vec3(3, 1, 5)
Q = Vec3(-1, -3, -1)

# Call C function to get intersection
inter = lib.find_intersection(A, B, C, P, Q)
```

C plus Python code

```
# Convert to numpy arrays for plotting
A_np = np.array([A.x, A.y, A.z])
B_np = np.array([B.x, B.y, B.z])
C_np = np.array([C.x, C.y, C.z])
P_np = np.array([P.x, P.y, P.z])
Q_np = np.array([Q.x, Q.y, Q.z])
inter_np = np.array([inter.x, inter.y, inter.z])

# --- Compute plane for plotting ---
AB = B_np - A_np
AC = C_np - A_np
n = np.cross(AB, AC) # normal
d = -np.dot(n, A_np) # plane constant

xx, yy = np.meshgrid(range(-5, 8), range(-5, 8))
zz = (-n[0]*xx - n[1]*yy - d) / n[2]
```

C plus Python code

```
# --- Compute line for plotting ---
d_line = Q_np - P_np
t = np.linspace(-5, 5, 100)
line_points = P_np[:, None] + d_line[:, None] * t

# --- Plot ---
fig = plt.figure(figsize=(10, 7))
ax = fig.add_subplot(111, projection='3d')

# Plane
ax.plot_surface(xx, yy, zz, alpha=0.5, color='cyan')

# Line
ax.plot(line_points[0], line_points[1], line_points[2], color='
    red', label=Line)
```

C plus Python code

```
# Intersection
ax.scatter(*inter_np, color='black', s=60, label=Intersection)

# Points
ax.scatter(*A_np, color='blue', s=50, label='A')
ax.scatter(*B_np, color='green', s=50, label='B')
ax.scatter(*C_np, color='purple', s=50, label='C')
ax.scatter(*P_np, color='orange', s=50, label='P')
ax.scatter(*Q_np, color='brown', s=50, label='Q')

# Labels
ax.set_xlabel(X)
ax.set_ylabel(Y)
ax.set_zlabel(Z)
ax.legend()

plt.savefig(/sdcard/4.4.12.png)
plt.show()
```

Python code

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

# Given points
A = np.array([2, 5, -3])
B = np.array([-2, -3, 5])
C = np.array([5, 3, -3])
P = np.array([3, 1, 5])
Q = np.array([-1, -3, -1])

# Step 1: Normal to the plane (AB x AC)
AB = B - A
AC = C - A
n = np.cross(AB, AC)
```

Python code

```

# Plane equation: n . (x - A) = 0
d = -np.dot(n, A) # plane constant
print(Plane equation: {}x + {}y + {}z + {} = 0.format(n[0], n[1], n[2], d))

# Step 2: Line parametric equation
d_line = Q - P # direction vector
t = np.linspace(-5, 5, 100)
line_points = P[:, None] + d_line[:, None] * t # shape (3, len(t))
```

Python code

```
# Step 3: Find intersection of line with plane
# Solve n.(P + t*d_line) + d = 0
t_inter = -(np.dot(n, P) + d) / np.dot(n, d_line)
intersection = P + t_inter * d_line
print(Intersection point:, intersection)

# Step 4: Plot
fig = plt.figure(figsize=(10, 7))
ax = fig.add_subplot(111, projection='3d')
```

Python code

```
# Plot plane
xx, yy = np.meshgrid(range(-5, 8), range(-5, 8))
zz = (-n[0]*xx - n[1]*yy - d) / n[2]
ax.plot_surface(xx, yy, zz, alpha=0.5, color='cyan')

# Plot line
ax.plot(line_points[0], line_points[1], line_points[2], color='
    red', label=Line)

# Plot intersection
ax.scatter(*intersection, color='black', s=60, label=Intersection
    )
```

Python code

```
# Plot given points
ax.scatter(*A, color='blue', s=50, label='A')
ax.scatter(*B, color='green', s=50, label='B')
ax.scatter(*C, color='purple', s=50, label='C')
ax.scatter(*P, color='orange', s=50, label='P')
ax.scatter(*Q, color='brown', s=50, label='Q')
```

Python code

```
# Labels
ax.set_xlabel(X)
ax.set_ylabel(Y)
ax.set_zlabel(Z)
ax.legend()
ax.set_title(Plane, Line, and Intersection)
plt.savefig(\sdcard\4.4.12.png)
plt.show()
```

Plot

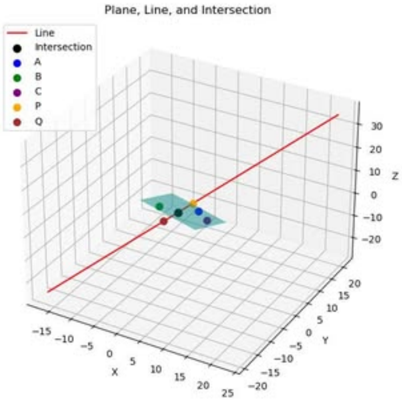

Figure: