4.3.32

Nipun Dasari - EE25BTECH11042

September 18, 2025

Question

Find the coordinates of the point where the line through the points $P(4, 3, 2)$ and $Q(5, 1, 6)$ crosses the XZ plane. Also find the angle which this line makes with the XZ plane. Solve using matrices and vectors only.

Theoretical Solution

First, we represent the points P and Q as position vectors.

$$\mathbf{p} = \begin{pmatrix} 4 \\ 3 \\ 2 \end{pmatrix} \text{and} \mathbf{q} = \begin{pmatrix} 5 \\ 1 \\ 6 \end{pmatrix} \tag{1}$$

The direction vector, **d**

$$\mathbf{d} = \mathbf{q} - \mathbf{p} = \begin{pmatrix} 5 \\ 1 \\ 6 \end{pmatrix} - \begin{pmatrix} 4 \\ 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 1 \\ -2 \\ 4 \end{pmatrix} \tag{2}$$

Line can be written as $\mathbf{r} = \mathbf{p} + \lambda \mathbf{d}$.

$$\mathbf{r} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 4 \\ 3 \\ 2 \end{pmatrix} + \lambda \begin{pmatrix} 1 \\ -2 \\ 4 \end{pmatrix} = \begin{pmatrix} 4 + \lambda \\ 3 - 2\lambda \\ 2 + 4\lambda \end{pmatrix} \tag{3}$$

The line crosses the XZ plane where the y-coordinate is 0.

$$3 - 2\lambda = 0 \implies 2\lambda = 3 \implies \lambda = \frac{3}{2} \tag{4}$$

Theoretical Solution

The line crosses the XZ plane where the y-coordinate is 0.

$$3 - 2\lambda = 0 \implies 2\lambda = 3 \implies \lambda = \frac{3}{2} \tag{5}$$

By (**??**)

$$\mathbf{r}_{\text{intersection}} = \begin{pmatrix} 4 + \frac{3}{2} \\ 3 - 2(\frac{3}{2}) \\ 2 + 4(\frac{3}{2}) \end{pmatrix} = \begin{pmatrix} \frac{11}{2} \\ 0 \\ 8 \end{pmatrix} \tag{6}$$

The intersection point is $(\frac{11}{2}, 0, 8)$.
The angle θ between a line with direction vector **d** and a plane with normal vector **n** is given by:

$$\cos\left(\pi/2 - \theta\right) = \frac{|\mathbf{d}^T\mathbf{n}|}{\|\mathbf{d}\|\,\|\mathbf{n}\|} \tag{7}$$

The direction vector of the line is $\mathbf{d} = \begin{pmatrix} 1 \\ -2 \\ 4 \end{pmatrix}$. The normal vector to the

$$\mathbf{d}^T\mathbf{n} = \begin{pmatrix} 1 \\ -2 \\ 4 \end{pmatrix}^T \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = (1)(0) + (-2)(1) + (4)(0) = -2 \tag{8}$$

$$\|\mathbf{d}\| = \sqrt{1^2 + (-2)^2 + 4^2} = \sqrt{1 + 4 + 16} = \sqrt{21} \tag{9}$$

$$\|\mathbf{n}\| = \sqrt{0^2 + 1^2 + 0^2} = 1 \tag{10}$$

By (**??**),(**??**),(**??**),(**??**)

$$\sin\theta = \frac{-2}{\sqrt{211}} = \frac{-2}{\sqrt{21}} \tag{11}$$

Therefore, the angle the line makes with the XZ plane is:

$$\theta = \arcsin\left(\frac{-2}{\sqrt{21}}\right) \tag{12}$$

C Code- Triangle Area function

```c
#include <stdio.h>
void calculate_plot_data(const double* p,
    const double* q,
double* intersection_point,
double* line_segment_x,
double* line_segment_y,
double* line_segment_z,
int num_line_points) {
        // Direction vector d = q - p
        double d[3];
        d[0] = q[0] - p[0]; // dx
        d[1] = q[1] - p[1]; // dy
        d[2] = q[2] - p[2]; // dz
```

C Code- Triangle Area function

```
// Find lambda for intersection with the XZ plane
    (where y=0)
// y(lambda) = p[1] + lambda * d[1] = 0 => lambda
    = -p[1] / d[1]
double lambda_intersect = -p[1] / d[1];

// Calculate and store the intersection point
intersection_point[0] = p[0] + lambda_intersect *
    d[0];
intersection_point[1] = 0.0; // By definition of
    the XZ plane
intersection_point[2] = p[2] + lambda_intersect *
    d[2];
```

C Code- Triangle Area function

```c
// Define a range for the parameter lambda to plot a nice segment of the line
double lambda_start = -1.0;
double lambda_end = 2.5;
double lambda_step = (lambda_end - lambda_start) / (num_line_points - 1);

// Generate points for the line segment
for (int i = 0; i < num_line_points; ++i) {
    double lambda = lambda_start + i * lambda_step;
    line_segment_x[i] = p[0] + lambda * d[0];
    line_segment_y[i] = p[1] + lambda * d[1];
```

Python Code using shared output

```
import ctypes
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import os
# --- Step 1: Load the compiled shared
    library ---
lib = ctypes.CDLL(./4.11.19.so)
# --- Step 2: Define the function signature
     (argument and return types) ---
lib.calculate_plot_data.argtypes = [
np.ctypeslib.ndpointer(dtype=np.float64,
    ndim=1, flags='C_CONTIGUOUS'), # p
np.ctypeslib.ndpointer(dtype=np.float64,
    ndim=1, flags='C_CONTIGUOUS'), # q
np.ctypeslib.ndpointer(dtype=np.float64,
    ndim=1, flags='C_CONTIGUOUS'), #
    intersection_point (output)
np.ctypeslib.ndpointer(dtype=np.float64,
```

Python Code using shared output

```python
# --- Step 3: Prepare data and call the C function ---
p_point = np.array([4, 3, 2], dtype=np.float64)
q_point = np.array([5, 1, 6], dtype=np.float64)

# Allocate memory for the output arrays that the C function will modify
num_line_points = 100
intersection_point = np.zeros(3, dtype=np.float64)
line_x = np.zeros(num_line_points, dtype=np.float64)
line_y = np.zeros(num_line_points, dtype=np.float64)
line_z = np.zeros(num_line_points, dtype=np.float64)

# Execute the function from our .so library
lib.calculate_plot_data(
p_point, q_point, intersection_point,
```

Python Code using shared output

```
# --- Step 4: Create the 3D plot with Matplotlib
    ---
fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')

# Plot the XZ plane surface
plane_x_range = np.arange(2, 9, 1)
plane_z_range = np.arange(0, 13, 1)
plane_xx, plane_zz = np.meshgrid(plane_x_range,
    plane_z_range)
plane_yy = np.zeros_like(plane_xx)
ax.plot_surface(plane_xx, plane_yy, plane_zz,
    alpha=0.2, color='c', rstride=10, cstride=10)

# Plot the line calculated by the C function
ax.plot(line_x, line_y, line_z, color='m', label='
    Line through P and Q')
```

Python Code using shared output

```
# Plot the points
ax.scatter(*p_point, color='blue', s=100,
    label=f'P {tuple(p_point)}')
ax.scatter(*q_point, color='green', s=100,
    label=f'Q {tuple(q_point)}')
ax.scatter(*intersection_point, color='red'
    , s=150, zorder=10, marker='*', label=f
    'Intersection')

# Formatting the plot
ax.set_xlabel('X-axis'), ax.set_ylabel('Y-
    axis'), ax.set_zlabel('Z-axis')
ax.set_title('Line Intersection with XZ
    Plane')
ax.legend()
ax.view_init(elev=20, azim=-60)
plt.show()
```

Plot by python using shared output from c