4.11.20

EE25BTECH11043-Nishid Khandagre

September 29, 2025

Question

Find the coordinates of the point where the line through the points $\mathbf{A}\begin{pmatrix}3\\4\\1\end{pmatrix}$ and $\mathbf{B}\begin{pmatrix}5\\1\\6\end{pmatrix}$ crosses the XZ plane. Also find the angle which this line makes with the XZ plane.

Theoretical Solution

Direction vector

$$\mathbf{d} = \mathbf{B} - \mathbf{A} \tag{1}$$

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

The normal vector $\mathbf{n}$ to the XZ-plane is:

$$\mathbf{n} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \tag{3}$$

Theoretical Solution

General point $\mathbf{P}$ on the line:

$$\mathbf{P} = \mathbf{A} + t\mathbf{d} \tag{4}$$

For the line to intersect the XZ-plane, the point $\mathbf{P}$ must lie on the plane. Therefore

$$\mathbf{n}^\top \mathbf{P} = 0 \tag{5}$$
$$\mathbf{n}^\top (\mathbf{A} + t\mathbf{d}) = 0 \tag{6}$$
$$\mathbf{n}^\top \mathbf{A} + t(\mathbf{n}^\top \mathbf{d}) = 0 \tag{7}$$
$$t = -\frac{\mathbf{n}^\top \mathbf{A}}{\mathbf{n}^\top \mathbf{d}} \tag{8}$$

Theoretical Solution

$$\mathbf{n}^\top \mathbf{A} = \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 3 \\ 4 \\ 1 \end{pmatrix} = 4 \tag{9}$$

$$\mathbf{n}^\top \mathbf{d} = \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 2 \\ -3 \\ 5 \end{pmatrix} = -3 \tag{10}$$

$$t = -\frac{4}{-3} = \frac{4}{3} \tag{11}$$

Theoretical Solution

Intersection Point

$$\mathbf{P} = \mathbf{A} + \frac{4}{3}\mathbf{d} \tag{12}$$

$$= \begin{pmatrix} 3 \\ 4 \\ 1 \end{pmatrix} + \frac{4}{3}\begin{pmatrix} 2 \\ -3 \\ 5 \end{pmatrix} \tag{13}$$

$$= \begin{pmatrix} \frac{17}{3} \\ 0 \\ \frac{23}{3} \end{pmatrix} \tag{14}$$

The angle θ between a line with direction vector $\mathbf{d}$ and a plane with normal vector $\mathbf{n}$ is given by:

$$\sin\theta = \frac{\left|\mathbf{n}^\top \mathbf{d}\right|}{\left\|\mathbf{n}\right\| \left\|\mathbf{d}\right\|} \tag{15}$$

$$\left\|\mathbf{n}\right\| = \sqrt{\mathbf{n}^\top \mathbf{n}} \tag{16}$$

$$\left\|\mathbf{n}\right\| = \sqrt{0^2 + 1^2 + 0^2} \tag{17}$$

$$\left\|\mathbf{n}\right\| = 1 \tag{18}$$

$$\left\|\mathbf{d}\right\| = \sqrt{\mathbf{d}^\top \mathbf{d}} \tag{19}$$

$$\left\|\mathbf{d}\right\| = \sqrt{2^2 + (-3)^2 + 5^2} \tag{20}$$

$$\left\|\mathbf{d}\right\| = \sqrt{38} \tag{21}$$

Theoretical Solution

$$\sin\theta = \frac{|-3|}{1 \cdot \sqrt{38}} = \frac{3}{\sqrt{38}} \tag{22}$$

C Code

```
#include <math.h>

// Function to calculate the intersection point and angle
void findIntersectionAndAngle(
double x1, double y1, double z1,
double x2, double y2, double z2,
double *ix, double *iy, double *iz,
double *angle_degrees) {

// Direction vector of the line (L = B - A)
double Lx = x2 - x1;
double Ly = y2 - y1;
double Lz = z2 - z1;
```

C Code

```c
// Parametric equation: P(t) = A + t * L
// P(t) = (x1 + t*Lx, y1 + t*Ly, z1 + t*Lz)
// For the XZ plane, y-coordinate must be 0:
// y1 + t*Ly = 0 => t = -y1 / Ly
double t = -y1 / Ly;
// Calculate the intersection point coordinates
*ix = x1 + t * Lx;
*iy = y1 + t * Ly; // Should be 0
*iz = z1 + t * Lz;
// Calculate angle with XZ plane (normal N = (0,1,0))
// sin(theta) = |L . N| / (||L|| * ||N||)
// L . N = Ly, ||N|| = 1
double dot_product = Ly;
double magnitude_L = sqrt(Lx * Lx + Ly * Ly + Lz * Lz);
double sin_theta = fabs(dot_product) / magnitude_L;
double angle_radians = asin(sin_theta);
*angle_degrees = angle_radians * 180.0 / M_PI;
```

Python Code through shared output

```
import ctypes
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
# Load the shared library
lib_geometry = ctypes.CDLL(./code8.so)
# Define the argument types for the C function
lib_geometry.findIntersectionAndAngle.argtypes = [
    ctypes.c_double, ctypes.c_double, ctypes.c_double,
    ctypes.c_double, ctypes.c_double, ctypes.c_double,
    ctypes.POINTER(ctypes.c_double), # ix
    ctypes.POINTER(ctypes.c_double), # iy
    ctypes.POINTER(ctypes.c_double), # iz
    ctypes.POINTER(ctypes.c_double) # angle_degrees
]
lib_geometry.findIntersectionAndAngle.restype = None
```

Python Code through shared output

```
# Given points
A = np.array([3.0, 4.0, 1.0])
B = np.array([5.0, 1.0, 6.0])
# Create ctypes doubles to hold the results
ix_result = ctypes.c_double()
iy_result = ctypes.c_double()
iz_result = ctypes.c_double()
angle_result = ctypes.c_double()
# Call the C function
lib_geometry.findIntersectionAndAngle(
    A[0], A[1], A[2], B[0], B[1], B[2],
    ctypes.byref(ix_result),
    ctypes.byref(iy_result),
    ctypes.byref(iz_result),
    ctypes.byref(angle_result)
)
```

Python Code through shared output

```
# Retrieve and print the results
intersection_point = np.array([ix_result.value, iy_result.value,
    iz_result.value])
angle_with_xz_plane = angle_result.value
print(fIntersection C: ({intersection_point[0]:.2f}, {
    intersection_point[1]:.2f}, {intersection_point[2]:.2f}))
print(fAngle with XZ plane: {angle_with_xz_plane:.2f} degrees)
# Plotting
fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')
# Plot points A, B, and C
ax.scatter(A[0], A[1], A[2], c='r', s=100, label='A(3,4,1)')
ax.scatter(B[0], B[1], B[2], c='b', s=100, label='B(5,1,6)')
```

Python Code through shared output

```
ax.scatter(intersection_point[0], intersection_point[1],
    intersection_point[2],
          c='g', s=100, zorder=5, label=f'C (Intersection)')
# Plot the full line
L = B - A
t_vals = np.linspace(-1, 2, 100)
line_full_x = A[0] + t_vals * L[0]
line_full_y = A[1] + t_vals * L[1]
line_full_z = A[2] + t_vals * L[2]
ax.plot(line_full_x, line_full_y, line_full_z, 'purple',
    linestyle='--', label='Line')
```

Python Code through shared output

```
# Plot the XZ plane (y=0)
x_plane = np.linspace(2, 8, 10)
z_plane = np.linspace(0, 8, 10)
X_plane, Z_plane = np.meshgrid(x_plane, z_plane)
Y_plane = np.zeros_like(X_plane)
ax.plot_surface(X_plane, Y_plane, Z_plane, alpha=0.2, color='gray
    ')
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()
plt.savefig(fig1.png)
plt.show()
```

Python Code: Direct

```python
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
# Given points
A = np.array([3, 4, 1])
B = np.array([5, 1, 6])
# Direction vector of the line
L = B - A
# For a point on the XZ plane, y=0.
# P(t) = A + t*L => y = 4 + t*(-3) = 0 => t = 4/3
t_intersect = 4/3
# Calculate the intersection point
C = A + t_intersect * L
print(fIntersection at C: ({C[0]:.2f}, {C[1]:.2f}, {C[2]:.2f}))
```

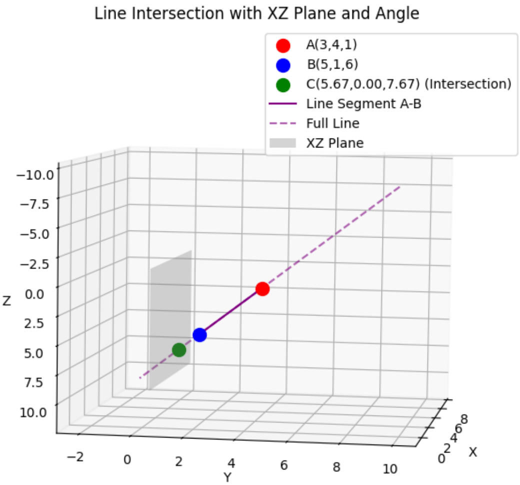
Python Code: Direct

```
# Find the angle with the XZ plane
# Normal vector to the XZ plane is N = (0, 1, 0)
normal_xz_plane = np.array([0, 1, 0])
# sin(theta) = |L . N| / (||L|| * ||N||)
dot_product = np.dot(L, normal_xz_plane)
magnitude_L = np.linalg.norm(L)
magnitude_N = np.linalg.norm(normal_xz_plane)
sin_theta = abs(dot_product) / (magnitude_L * magnitude_N)
angle_radians = np.arcsin(sin_theta)
angle_degrees = np.degrees(angle_radians)
print(fAngle with XZ plane: {angle_degrees:.2f} degrees)
# Plotting
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
```

Python Code: Direct

```
# Plot points and line
ax.scatter(A[0], A[1], A[2], c='r', s=100, label='A')
ax.scatter(B[0], B[1], B[2], c='b', s=100, label='B')
ax.scatter(C[0], C[1], C[2], c='g', s=100, label='C (Intersection
    )')
line_pts = np.array([A, B, C])
ax.plot(line_pts[:,0], line_pts[:,1], line_pts[:,2], 'purple')
# Plot XZ plane
x_plane = np.linspace(2, 8, 10)
z_plane = np.linspace(0, 8, 10)
X_plane, Z_plane = np.meshgrid(x_plane, z_plane)
Y_plane = np.zeros_like(X_plane)
ax.plot_surface(X_plane, Y_plane, Z_plane, alpha=0.3, color='gray
    ')
ax.set_xlabel('X'); ax.set_ylabel('Y'); ax.set_zlabel('Z')
ax.legend()
plt.savefig(fig2.png)
plt.show()
```

Plot by Python using shared output from C



Plot by Python only