4.10.15

Bhargav - EE25BTECH11013

September 12, 2025

Question

Show that the lines

$$\frac{x-1}{2} = \frac{y-2}{3} = \frac{z-3}{4} \tag{1}$$

and

$$\frac{x-4}{5} = \frac{y-1}{2} = z \tag{2}$$

intersect. Also, find their point of intersection.

Vector Equations

The vector equations of the given lines are

$$\mathbf{r}_1 = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} + \lambda \begin{pmatrix} 2 \\ 3 \\ 4 \end{pmatrix}, \tag{3}$$

$$\mathbf{r}_2 = \begin{pmatrix} 4 \\ 1 \\ 0 \end{pmatrix} + \mu \begin{pmatrix} 5 \\ 2 \\ 1 \end{pmatrix}. \tag{4}$$

At the point of intersection,

$$\mathbf{r}_1 = \mathbf{r}_2. \tag{5}$$

Matrix Equation

Thus,

$$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} + \lambda \begin{pmatrix} 2 \\ 3 \\ 4 \end{pmatrix} = \begin{pmatrix} 4 \\ 1 \\ 0 \end{pmatrix} + \mu \begin{pmatrix} 5 \\ 2 \\ 1 \end{pmatrix}. \tag{6}$$

This can be written as

$$\begin{pmatrix} 2 & -5 \\ 3 & -2 \\ 4 & -1 \end{pmatrix} \begin{pmatrix} \lambda \\ \mu \end{pmatrix} = \begin{pmatrix} 3 \\ -1 \\ -3 \end{pmatrix}. \tag{7}$$

Row Reduction

The corresponding augmented matrix is

$$\left(\begin{array}{cc|c} 2 & -5 & 3 \\ 3 & -2 & -1 \\ 4 & -1 & -3 \end{array}\right) \xleftrightarrow[R_2 \leftarrow R_2 - \frac{3}{2}R_1]{R_3 \leftarrow R_3 - 2R_1} \left(\begin{array}{cc|c} 2 & -5 & 3 \\ 0 & 11/2 & -11/2 \\ 0 & 9 & -9 \end{array}\right) \tag{8}$$

$$\xleftrightarrow[R_3 \leftarrow 11R_3 - 9R_2]{R_2 \leftarrow 2R_2} \left(\begin{array}{cc|c} 2 & -5 & 3 \\ 0 & 11 & -11 \\ 0 & 0 & 0 \end{array}\right) \tag{9}$$

Solution

From this,

$$\begin{pmatrix} \lambda \\ \mu \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \end{pmatrix}. \tag{10}$$

Substituting into $\mathbf{r}_1$:

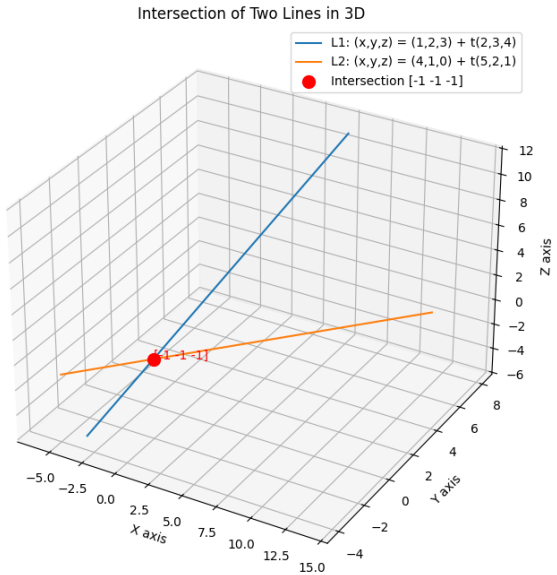
$$\mathbf{r}_1 = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} + (-1) \begin{pmatrix} 2 \\ 3 \\ 4 \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix}. \tag{11}$$

Intersection Point

Therefore, the lines intersect at the point

$$\begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix} \tag{12}$$

Figure


Intersection of Two Lines in 3D
L1: (x,y,z) = (1,2,3) + t(2,3,4)
L2: (x,y,z) = (4,1,0) + t(5,2,1)
Intersection [-1 -1 -1]
[-1 -1 -1]
X axis
Y axis
Z axis

C Code

```c
#include <stdio.h>
int line_intersection(double p1[3], double d1[3],
                      double p2[3], double d2[3],
                      double intersection[3]) {
    double A[2][2] = {
        { d1[0], -d2[0] },
        { d1[1], -d2[1] }
    };
    double b[2] = { p2[0] - p1[0], p2[1] - p1[1] };

    double det = A[0][0]*A[1][1] - A[0][1]*A[1][0];
    if(det == 0) {
        return -1; // Parallel or skew
    }
    double lam = (b[0]*A[1][1] - b[1]*A[0][1]) / det;
    for(int i = 0; i < 3; i++) {
        intersection[i] = p1[i] + lam*d1[i];
    }
    return 0;
```

Python + C Code

```
import numpy as np
import matplotlib.pyplot as plt
import ctypes
lib = ctypes.CDLL(./line_intersection.so)
lib.line_intersection.argtypes = [
    np.ctypeslib.ndpointer(ctypes.c_double, ndim=1, flags=
        C_CONTIGUOUS),
    np.ctypeslib.ndpointer(ctypes.c_double, ndim=1, flags=
        C_CONTIGUOUS),
    np.ctypeslib.ndpointer(ctypes.c_double, ndim=1, flags=
        C_CONTIGUOUS),
    np.ctypeslib.ndpointer(ctypes.c_double, ndim=1, flags=
        C_CONTIGUOUS),
    np.ctypeslib.ndpointer(ctypes.c_double, ndim=1, flags=
        C_CONTIGUOUS),
]
lib.line_intersection.restype = ctypes.c_int
p1, d1 = np.array([1.0, 2.0, 3.0]), np.array([2.0, 3.0, 4.0])
p2, d2 = np.array([4.0, 1.0, 0.0]), np.array([5.0, 2.0, 1.0])
```

Python + C Code

```
intersection = np.zeros(3, dtype=np.float64)


status = lib.line_intersection(p1, d1, p2, d2, intersection)
if status != 0:
    print(The lines are parallel or skew; no intersection.)
else:
    print(fThe lines intersect at: {intersection.astype(int)})


    eq1 = fL1: (x,y,z) = ({int(p1[0])},{int(p1[1])},{int(p1[2])})
         + t({int(d1[0])},{int(d1[1])},{int(d1[2])})
    eq2 = fL2: (x,y,z) = ({int(p2[0])},{int(p2[1])},{int(p2[2])})
         + t({int(d2[0])},{int(d2[1])},{int(d2[2])})
    fig = plt.figure(figsize=(10, 8))
    ax = fig.add_subplot(111, projection=3d)
    t = np.linspace(-2, 2, 50)
```

Python + C Code

```
line1_points = p1 + t[:, None] * d1
line2_points = p2 + t[:, None] * d2
ax.plot(line1_points[:, 0], line1_points[:, 1], line1_points
    [:, 2], label=eq1)
ax.plot(line2_points[:, 0], line2_points[:, 1], line2_points
    [:, 2], label=eq2)
ax.scatter(intersection[0], intersection[1], intersection[2],
          color=red, s=100, zorder=5, label=fIntersection {
              intersection.astype(int)})
ax.text(intersection[0], intersection[1], intersection[2],
        f{intersection.astype(int)}, color=red, fontsize=10)
ax.set_xlabel(X axis)
ax.set_ylabel(Y axis)
ax.set_zlabel(Z axis)
ax.set_title(Intersection of Two Lines in 3D)
ax.legend()
plt.savefig('/Users/bhargavkrish/Desktop/BackupMatrix/
    ee25btech11013/matgeo/4.10.15/figs/Figure_1.png')
plt.show()
```

Python Code

```
import numpy as np
import matplotlib.pyplot as plt

p1, d1 = np.array([1, 2, 3]), np.array([2, 3, 4])
p2, d2 = np.array([4, 1, 0]), np.array([5, 2, 1])

A = np.array([d1[:2], -d2[:2]]).T
b = p2[:2] - p1[:2]

try:
    lambda_val, mu_val = np.linalg.solve(A, b)
    intersection_point = p1 + lambda_val * d1
    print(fThe lines intersect at the point: {intersection_point}
        )

    # Line equations for legend
    eq1 = fL1: (x,y,z) = ({p1[0]},{p1[1]},{p1[2]}) + t({d1[0]},{
        d1[1]},{d1[2]})
    eq2 = fL2: (x,y,z) = ({p2[0]},{p2[1]},{p2[2]}) + t({d2[0]},{
```

Python Code

```
fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')
t = np.linspace(-2, 2, 50)
line1_points = p1 + t[:, np.newaxis] * d1
line2_points = p2 + t[:, np.newaxis] * d2
ax.plot(line1_points[:, 0], line1_points[:, 1], line1_points
    [:, 2], label=eq1)
ax.plot(line2_points[:, 0], line2_points[:, 1], line2_points
    [:, 2], label=eq2)
ax.scatter(intersection_point[0], intersection_point[1],
    intersection_point[2],
          color='red', s=100, zorder=5, label=f'Intersection
              {intersection_point}')
ax.text(intersection_point[0], intersection_point[1],
    intersection_point[2],
        f'{intersection_point}', color='red', fontsize=10)
ax.set_xlabel('X axis')
ax.set_ylabel('Y axis')
ax.set_zlabel('Z axis')
```

Python Code

```
    ax.set_title('Intersection of Two Lines in 3D')
    ax.legend()
    plt.savefig('/Users/bhargavkrish/Desktop/BackupMatrix/
        ee25btech11013/matgeo/4.10.15/figs/Figure_1.png')
    plt.show()

except np.linalg.LinAlgError:
    print(The lines are parallel or skew; they do not intersect.)
```