1.11.16

AI25BTECH11014 - Suhas

September 9, 2025

Question

The Cartesian equation of a line AB is

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

Find the direction cosines of a line parallel to line AB.

Solution

Extract the direction vector from the denominators:

$$\mathbf{D} = \begin{pmatrix} 2 \\ 2 \\ 3 \end{pmatrix} \tag{1}$$

Compute the magnitude of **D**:

$$\|\mathbf{D}\| = \sqrt{2^2 + 2^2 + 3^2} = \sqrt{17} \tag{2}$$

Normalize the direction vector:

$$\mathbf{L} = \frac{1}{\sqrt{17}} \cdot \mathbf{D} = \frac{1}{\sqrt{17}} \cdot \begin{pmatrix} 2 \\ 2 \\ 3 \end{pmatrix} \tag{3}$$

Matrix Form

Let the point vector be:

$$\mathbf{P} = \begin{pmatrix} 1 \\ -2 \\ 3 \end{pmatrix} \tag{4}$$

Then the line can be expressed in matrix form as:

$$\mathbf{R} = \mathbf{P} + \lambda \cdot \mathbf{D} \tag{5}$$

Where $\lambda \in \mathbb{R}$ is a scalar parameter.

Conclusion

The direction cosines of a line parallel to AB are:

$$\mathbf{L} = \frac{1}{\sqrt{17}} \cdot \begin{pmatrix} 2 \\ 2 \\ 3 \end{pmatrix}$$

The line can be expressed in matrix form as:

$$\mathbf{R} = \begin{pmatrix} 1 \\ -2 \\ 3 \end{pmatrix} + \lambda \cdot \begin{pmatrix} 2 \\ 2 \\ 3 \end{pmatrix}$$

C Code: Solve Direction Cosines

```c
#include <stdio.h>
#include <math.h>

int main() {
    double D[3] = {2.0, 1.0, 3.0};
    double mag = sqrt(D[0]*D[0] +
                      D[1]*D[1] +
                      D[2]*D[2]);
    printf("Magnitude: %f\n", mag);
}
```

C Code: Solve Direction Cosines (2/2)

```c
    printf("Direction Cosines:\n");
    for (int i = 0; i < 3; i++) {
        printf("%.5f\n", D[i]/mag);
    }
    return 0;
}
```

C Code: Formula Library (1/2)

```c
#include <math.h>

void direction_cosines(double* D,
                     double* L) {
    double mag = sqrt(D[0]*D[0] +
                    D[1]*D[1] +
                    D[2]*D[2]);
```

C Code: Formula Library (2/2)

```c
    for (int i = 0; i < 3; i++) {
        L[i] = D[i] / mag;
    }
}
```

Python Code: Using .so (1/2)

```python
import ctypes
import numpy as np
from sympy import Matrix, sqrt

lib = ctypes.CDLL('./libdircos.so')
dir_cos = lib.direction_cosines
dir_cos.argtypes = [
    ctypes.POINTER(ctypes.c_double),
    ctypes.POINTER(ctypes.c_double)
]
```

Python Code: Using .so (2/2)

```python
D = np.array([2.0, 2.0, 3.0])
L = np.zeros(3)

dir_cos(D.ctypes.data_as(ctypes.POINTER(
        ctypes.c_double)),
        L.ctypes.data_as(ctypes.POINTER(
        ctypes.c_double)))

print("Direction Cosines:", L)
```

Plot

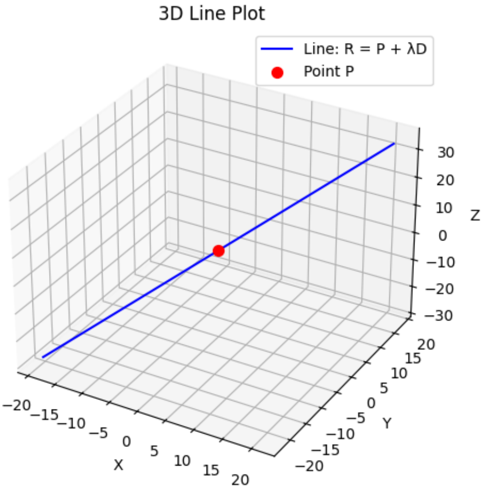

Figure: Line AB