4.3.9

AI25BTECH11003 - Bhavesh Gaikwad

September 3,2025

Question

The vector equation of the line $\dfrac{x-5}{3} = \dfrac{y+4}{7} = \dfrac{z-6}{2}$ is?

Theoretical Solution

Given:

$$\frac{x-5}{3}=\frac{y+4}{7}=\frac{z-6}{2} \tag{1}$$

Let $\mathbf{A}$ be the parallel vector of the given line.
Let $\mathbf{B}$ be the position vector of a point on the given line.

From Equation 1,

$$\mathbf{A}=\begin{pmatrix}3\\7\\2\end{pmatrix} \tag{2}$$

Putting x=8 in Equation 1 to get an arbitrary point on the line,

$$\frac{8-5}{3}=\frac{y+4}{7}=\frac{z-6}{2} \quad \Rightarrow x=8,\ y=3,\ z=8. \tag{3}$$

$$\therefore\ \mathbf{B}=\begin{pmatrix}8\\3\\8\end{pmatrix} \tag{4}$$

Theoretical Solution

From Equations 1 and 4,
The Vector Equation of the given line is:

$$\mathbf{L} = \mathbf{B} + k\mathbf{A}, \text{ Where } k \text{ is a real parameter OR } k \,\epsilon\, \mathbb{R} \tag{5}$$

$$\boxed{\mathbf{L} = \begin{pmatrix} 8 \\ 3 \\ 8 \end{pmatrix} + k \begin{pmatrix} 3 \\ 7 \\ 2 \end{pmatrix}} \tag{6}$$

Python Code

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

# Direction vector A and point B
A = np.array([3, 7, 2])
B = np.array([8, 3, 8])

# Parameter k for the line
k = np.linspace(-10, 10, 400)

# Compute line points: L = B + kA
line_points = B.reshape(3,1) + np.outer(A, k)

# Create a perpendicular offset so vector A doesn't overlap the line
```

Python Code

```
      # Find a vector perpendicular to A by crossing with an arbitrary vector
arbitrary = np.array([1, 0, 0])
if np.allclose(np.cross(A, arbitrary), 0):
    arbitrary = np.array([0, 1, 0])
offset_dir = np.cross(A, arbitrary)
offset_dir = offset_dir / np.linalg.norm(offset_dir)
offset = offset_dir * 2 # magnitude of offset for visibility

# Base point for drawing A offset from the line
base_point = B + offset
arrow_end = base_point + A

# Plot setup
fig = plt.figure(figsize=(8, 6))
ax = fig.add_subplot(111, projection='3d')
```

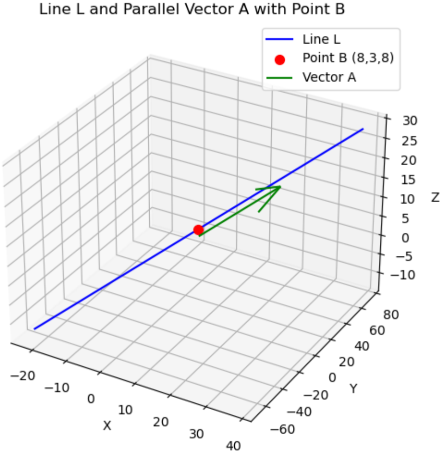
Python Code

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

# Mark point B on the line
ax.scatter(B[0], B[1], B[2], color='red', s=50, label='Point B (8,3,8)')

# Draw vector A at the offset base_point
ax.quiver(base_point[0], base_point[1], base_point[2],
         A[0], A[1], A[2], color='green', length=5, normalize=False, label='Vector A')
```

Python Code

```python
# Labels and legend
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_title('Line L and Parallel Vector A with Point B')
ax.legend()

# Save the figure
plt.savefig('fig1.png')
plt.close()
```

Line


Line L and Parallel Vector A with Point B
Line L
Point B (8,3,8)
Vector A
30
25
20
15
10
5
0
−5
−10
Z
80
60
40
20
0
−20
−40
−60
Y
−20
−10
0
10
20
30
40
X