MatGeo Assignment - Problem 4.3.9

EE25BTECH11024

IIT Hyderabad

October 5, 2025

Problem Statement

The vector equation of the line

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

is _____.

Formula Used

The general vector equation of a line in 3D is

$$\mathbf{x} = \mathbf{h} + \kappa \mathbf{m}, \tag{2}$$

Solution:

From (1) we get the following equations.

$$x = 5 + 3\kappa, \tag{3}$$
$$y = -4 + 7\kappa, \tag{4}$$
$$z = 6 + 2\kappa. \tag{5}$$

comparing (3), (4), (5), and (2) we get,

$$\mathbf{h} = \begin{pmatrix} 5 \\ -4 \\ 6 \end{pmatrix}, \quad \mathbf{m} = \begin{pmatrix} 3 \\ 7 \\ 2 \end{pmatrix}. \tag{6}$$

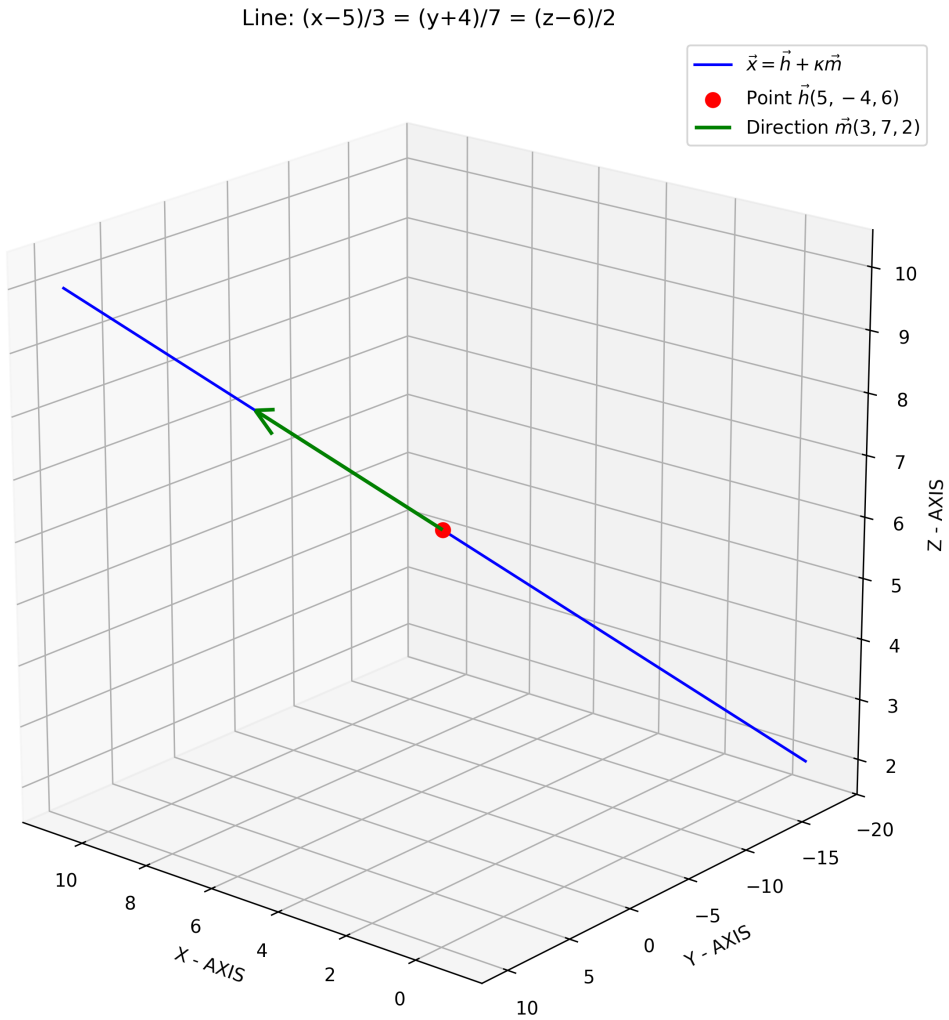
Final Answer

Therefore, the vector equation of the line is:

$$\mathbf{x} = \begin{pmatrix} 5 \\ -4 \\ 6 \end{pmatrix} + \kappa \begin{pmatrix} 3 \\ 7 \\ 2 \end{pmatrix} \tag{7}$$

See Figure 1.

Figure


Line: (x−5)/3 = (y+4)/7 = (z−6)/2
$\vec{x} = \vec{h} + \kappa\vec{m}$
Point $\vec{h}(5, -4, 6)$
Direction $\vec{m}(3, 7, 2)$
X - AXIS
Y - AXIS
Z - AXIS

Python Code: plot.py (Native)

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

h = np.array([5, -4, 6])
m = np.array([3, 7, 2])

k_values = np.linspace(-2, 2, 100)
line_points = np.array([h + k * m for k in k_values])

fig = plt.figure(figsize=(10,10))
ax = fig.add_subplot(111, projection='3d')

ax.plot(line_points[:,0], line_points[:,1], line_points[:,2], color='
    blue', label=r"$\vec{x} = \vec{h} + \kappa \vec{m}$")

ax.scatter(h[0], h[1], h[2], color='red', s=60, label=r"Point $\vec{h
    }(5, -4, 6)$")
```

Python Code (Native Implementation – plot.py)

```
ax.quiver(h[0], h[1], h[2], m[0], m[1], m[2],
          color='green', arrow_length_ratio=0.1, linewidth=2, label=r"
              Direction $\vec{m}(3,7,2)$")

ax.set_xlabel("X - AXIS")
ax.set_ylabel("Y - AXIS")
ax.set_zlabel("Z - AXIS")
ax.set_title("Line: (x5)/3 = (y+4)/7 = (z6)/2")
ax.legend()
ax.set_box_aspect([1,1,1])
ax.view_init(elev=20, azim=130)
plt.savefig("line_vector_equation.png", dpi=300)
plt.show()
```

C Code (Shared Library – findlinepoints.c)

```c
#include <stdio.h>

void find_line_points(double *h, double *m, double k1, double k2, double
     *P1, double *P2)
{
    for (int i = 0; i < 3; i++) {
        P1[i] = h[i] + k1 * m[i];
        P2[i] = h[i] + k2 * m[i];
    }
}
```

Python Code: call.py (C + Python)

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

lib = ctypes.CDLL("./find_line_points.so")

lib.find_line_points.argtypes = [
    ctypes.POINTER(ctypes.c_double),
    ctypes.POINTER(ctypes.c_double),
    ctypes.c_double,
    ctypes.c_double,
    ctypes.POINTER(ctypes.c_double),
    ctypes.POINTER(ctypes.c_double)
]
lib.find_line_points.restype = None

h = np.array([5.0, -4.0, 6.0], dtype=np.float64)
m = np.array([3.0, 7.0, 2.0], dtype=np.float64)
```

Python Code (C Integrated – call.py)

```
P1 = np.zeros(3, dtype=np.float64)
P2 = np.zeros(3, dtype=np.float64)

k1, k2 = -2.0, 2.0
lib.find_line_points(h.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
                     m.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
                     k1, k2,
                     P1.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
                     P2.ctypes.data_as(ctypes.POINTER(ctypes.c_double)))

line_points = np.linspace(P1, P2, 100)

fig = plt.figure(figsize=(10,10))
ax = fig.add_subplot(111, projection='3d')

ax.plot(line_points[:,0], line_points[:,1], line_points[:,2], color='
    blue', label=r"$\vec{x} = \vec{h} + \kappa \vec{m}$")

ax.scatter(h[0], h[1], h[2], color='red', s=60, label=r"Point $\vec{h
    }(5, -4, 6)$")
```

Python Code (C Integrated – call.py)

```
ax.quiver(h[0], h[1], h[2], m[0], m[1], m[2],
          color='green', arrow_length_ratio=0.1, linewidth=2, label=r"
              Direction $\vec{m}(3,7,2)$")

ax.set_xlabel("X - AXIS")
ax.set_ylabel("Y - AXIS")
ax.set_zlabel("Z - AXIS")
ax.set_title("Line: (x5)/3 = (y+4)/7 = (z6)/2")
ax.legend()
ax.set_box_aspect([1,1,1]) # Equal aspect ratio
ax.view_init(elev=20, azim=130)
plt.savefig("line_vector_equation_from_dll.png", dpi=300)
plt.show()
```