Problem 4.3.8

ee25btech11023-Venkata Sai

September 5, 2025

Problem

The vector equation of the line

$$\frac{x-5}{3}=\frac{y+4}{7}=\frac{z-6}{2} \tag{1.1}$$

is

Finding **a** and **b**

A point on the line is given as

$$\begin{pmatrix} x-5 \\ y+4 \\ z-6 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{2.1}$$

$$\mathbf{a} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 5 \\ -4 \\ 6 \end{pmatrix} \tag{2.2}$$

The direction vectors of given line are

$$\mathbf{b} = \begin{pmatrix} 3 \\ 7 \\ 2 \end{pmatrix} \tag{2.3}$$

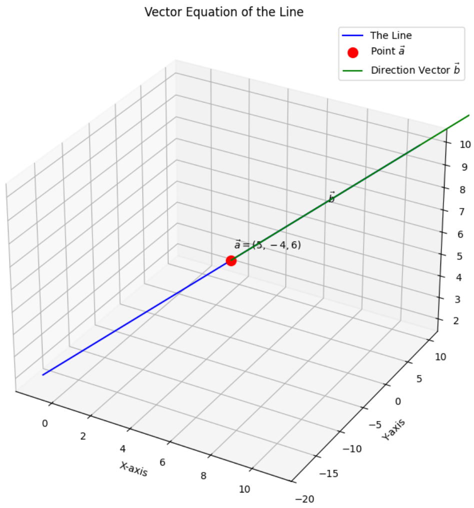
Conclusion

The vector equation of a line is given as

$$\mathbf{r} = \mathbf{a} + t\mathbf{b} \tag{2.4}$$

$$\mathbf{r} = \begin{pmatrix} 5 \\ -4 \\ 6 \end{pmatrix} + t \begin{pmatrix} 3 \\ 7 \\ 2 \end{pmatrix} \tag{2.5}$$

Plot



Vector Equation of the Line
The Line
Point $\vec{a}$
Direction Vector $\vec{b}$
$\vec{a} = (5, -4, 6)$
$\vec{b}$
X-axis
Y-axis
Z-axis
0
2
4
6
8
10
−20
−15
−10
−5
0
5
10
2
3
4
5
6
7
8
9
10

C Code

```c
void get_line_vectors(double* out_data) {

    double point_a[3] = {5.0, -4.0, 6.0};
    double dir_b[3] = {3.0, 7.0, 2.0};

    out_data[0] = point_a[0];
    out_data[1] = point_a[1];
    out_data[2] = point_a[2];

    out_data[3] = dir_b[0];
    out_data[4] = dir_b[1];
    out_data[5] = dir_b[2];
}
```

Calling C Function

```python
import ctypes
import numpy as np

def get_vectors_from_c():
    lib = ctypes.CDLL('./vector.so')

    # The C function expects a pointer to a C double array of
        size 6
    double_array_6 = ctypes.c_double * 6
    lib.get_line_vectors.argtypes = [ctypes.POINTER(ctypes.
        c_double)]

    # Create the C-style array to receive the output data
    out_data_c = double_array_6()

    # Call the C function, which will fill the array
    lib.get_line_vectors(out_data_c)
```

Calling C Function

```
# Convert the C array back into a NumPy array
    all_data = np.array(out_data_c)

    # Split the data into the point vector and direction vector
    point_a = all_data[:3]
    dir_b = all_data[3:]

    return point_a, dir_b
```

Python Code for Plotting

```python
#Code by GVV Sharma
#September 12, 2023
#Revised July 21, 2024
#released under GNU GPL

import sys
import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits.mplot3d import Axes3D
sys.path.insert(0, '/workspaces/urban-potato/matgeo/codes/
    CoordGeo/')
from call import get_vectors_from_c
hat_symbol = '\u0302'
from line.funcs import *
from triangle.funcs import *
from conics.funcs import circ_gen

point_a, dir_b = get_vectors_from_c()
```

Python Code for Plotting

```python
from call import get_vectors_from_c

point_a, dir_b = get_vectors_from_c()

lambda_vals = np.array([-2, 2])
line_points = point_a + lambda_vals[:, np.newaxis] * dir_b

# --- Plotting ---
fig = plt.figure(figsize=(9, 9))
ax = fig.add_subplot(111, projection='3d')

# Plot the line segment itself
ax.plot(line_points[:, 0], line_points[:, 1], line_points[:, 2],
    color='blue', label='The Line')

# Plot the point 'a' on the line
ax.scatter(point_a[0], point_a[1], point_a[2], color='red', s
    =100, label='Point $\\vec{a}$')
```

Python Code for Plotting

```
  # Plot the direction vector 'd' starting from point 'a'
ax.quiver(point_a[0], point_a[1], point_a[2],
          dir_b[0], dir_b[1], dir_b[2],
          color='green', label='Direction Vector $\\vec{b}$',
              length=5, arrow_length_ratio=0.3)

# Add text labels for the point and vectors
ax.text(point_a[0], point_a[1], point_a[2] + 0.5, f' $\\vec{{a}}
    = ({point_a[0]:.0f}, {point_a[1]:.0f}, {point_a[2]:.0f})$')
ax.text(point_a[0] + dir_b[0], point_a[1] + dir_b[1], point_a[2]
    + dir_b[2], ' $\\vec{b}$')
```

Python Code for Plotting

```python
# --- Formatting ---
ax.set_title('Vector Equation of the Line')
ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.set_zlabel('Z-axis')

ax.grid(True)
ax.legend()
plt.show()
plt.savefig('fig.png')
```