1.10.9

Hema Havil - EE25BTECH11050

October 6, 2025

Question

Find the equation of the line that passes through the point with position vector $2\hat{i} - \hat{j} + 4\hat{k}$ and is in direction $\hat{i} + 2\hat{j} - \hat{k}$.

Theoretical Solution

Given,
the point on the line,

$$\mathbf{r_0} = \begin{pmatrix} 2 \\ -1 \\ 4 \end{pmatrix} \tag{1}$$

the direction vector of the line,

$$\mathbf{d} = \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix} \tag{2}$$

Let the position vector of any point on the line be $\mathbf{r_t}$ then,

$$\mathbf{r_t} = \mathbf{r_0} + t\mathbf{d} \tag{3}$$

Theoretical Solution

$$\mathbf{r_t} = \begin{pmatrix} 2+t \\ -1+2t \\ 4+-t \end{pmatrix} \tag{4}$$

where t is the parameter,
Therefore the equation of the line is

$$\mathbf{r_t} = \begin{pmatrix} 2+t \\ -1+2t \\ 4+-t \end{pmatrix} \tag{5}$$

C Code- Computing the unit vector

```c
#include <stdio.h>

void line_point(double lambda, double result[3]) {
    // Point vector a
    double a[3] = {2.0, -1.0, 4.0};
    // Direction vector d
    double d[3] = {1.0, 2.0, -1.0};

    for (int i = 0; i < 3; i++) {
        result[i] = a[i] + lambda * d[i];
    }
}
```

Python Code using shared output

```python
import numpy as np
import matplotlib.pyplot as plt
from ctypes import CDLL, c_double, POINTER

# Load the shared library
lib = CDLL('./4.3.40.so')

# Define function signature for line_point
lib.line_point.argtypes = [c_double, POINTER(c_double)]
lib.line_point.restype = None

def get_point(lambda_val):
    # Prepare array for results
    result = (c_double * 3)()
    lib.line_point(lambda_val, result)
    return np.array([result[0], result[1], result[2]])
    # Generate points on the line for lambda in [-10, 10]
```

Python Code using shared output

```python
lambdas = np.linspace(-10, 10, 400)
points = np.array([get_point(l) for l in lambdas])

# Plotting the line in 3D
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot(points[:,0], points[:,1], points[:,2], label='Line: r = a + d', color='blue')

# Plot the point a
ax.scatter(2, -1, 4, color='red', label='Point a (2, -1, 4)')

ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.legend()
plt.title('3D Line plot')
plt.show()
```

Plot by python using shared output from c

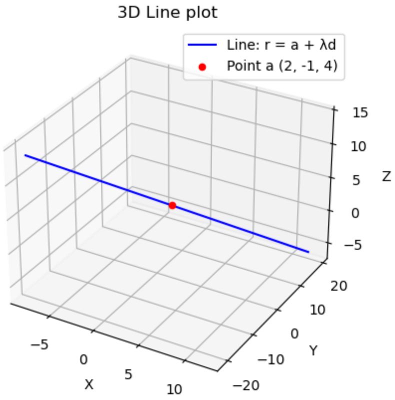

Figure: Plot of the 3D line

Python code for the plot

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

# Given vectors
a = np.array([2, -1, 4]) # point vector
d = np.array([1, 2, -1]) # direction vector

# Parameter lambda values
lambdas = np.linspace(-10, 10, 400)

# Calculate points on the line for each lambda
points = np.array([a + l * d for l in lambdas])
```

Python code for the plot

```python
# Plotting the line in 3D
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

ax.plot(points[:, 0], points[:, 1], points[:, 2], label='Line: r = a + d', color='blue')

# Plot the given point 'a'
ax.scatter(a[0], a[1], a[2], color='red', label='Point a (2, -1, 4)')

ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.legend()
plt.title('3D Line plot')
plt.show()
```

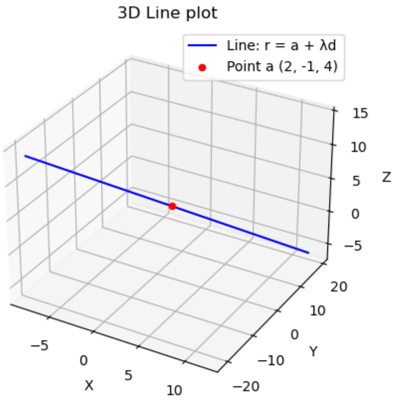

Figure: Plot for the 3D line