Matgeo 1-1.11-16

AI24BTECH11001 - Abhijeet Kumar

November 6, 2024

1 Question

2 Solution
- Terms Used
- Solution
- Plot
- C Code
- Python Code

Question

The cartesian equation of a line AB is $\frac{2x-1}{12} = \frac{y+2}{2} = \frac{z-3}{3}$.Find the direction cosines of a line parallel to line AB.

Terms Used

Table: Terms used

Term	**Description**
m	Direction vector of line

Solution

The direction vector of the given line is:

$$\mathbf{m} = \begin{pmatrix} 6 \\ 2 \\ 3 \end{pmatrix}$$

The unit vector of a line having direction vector m is given by :

$$= \frac{m}{||m||} \tag{3.1}$$

The direction cosines are elements of above vector

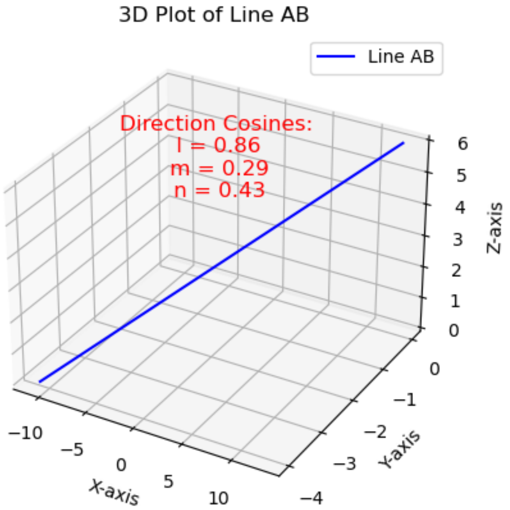
Solution

From 3.1 the unit vector along AB is:

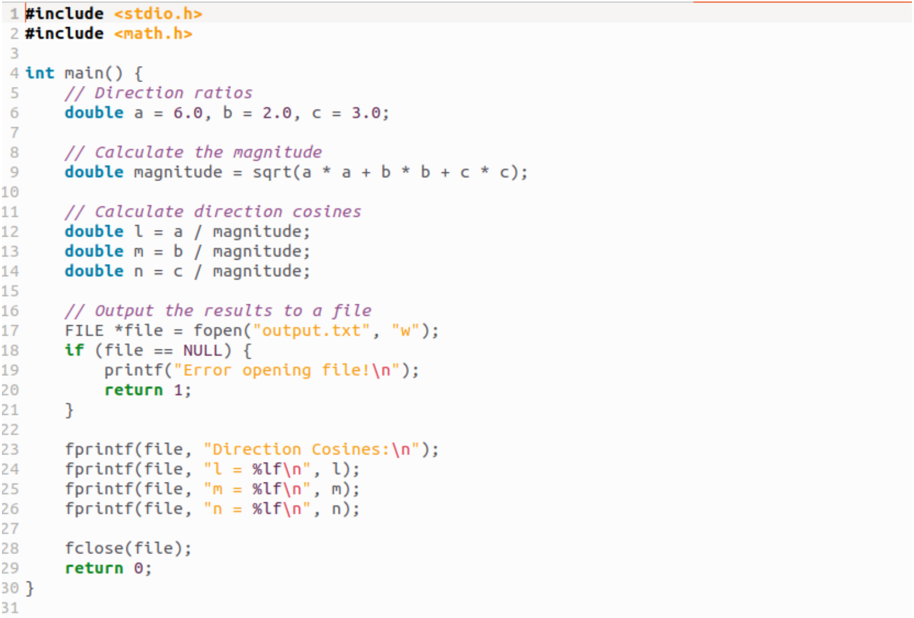
$$= \frac{1}{\sqrt{49}} \begin{pmatrix} 6 \\ 2 \\ 3 \end{pmatrix} \tag{3.2}$$

$\therefore$ The direction cosines of line parallel to line AB are the elements of above vector.

Plot

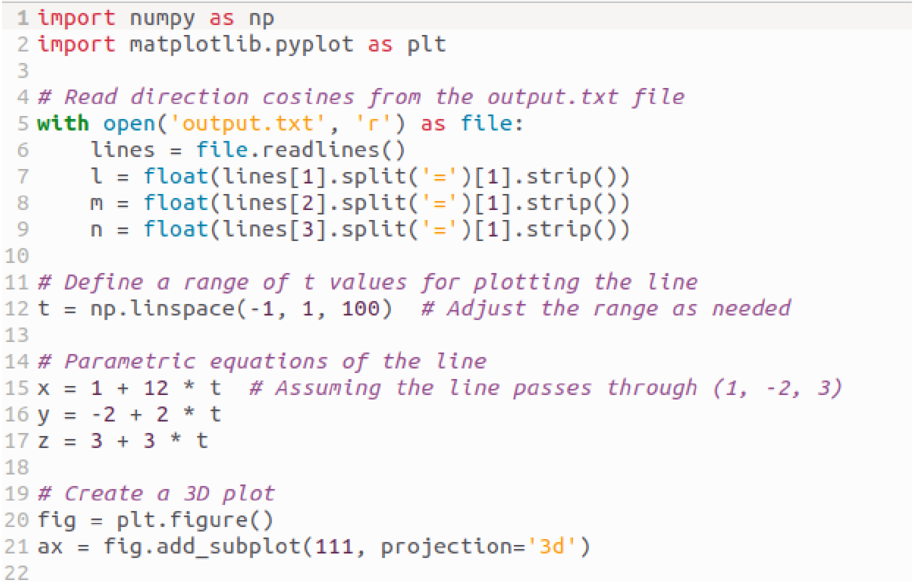

3D Plot of Line AB
Line AB
Direction Cosines:
l = 0.86
m = 0.29
n = 0.43
X-axis
−10
−5
0
5
10
Y-axis
0
−1
−2
−3
−4
Z-axis
0
1
2
3
4
5
6

C Code



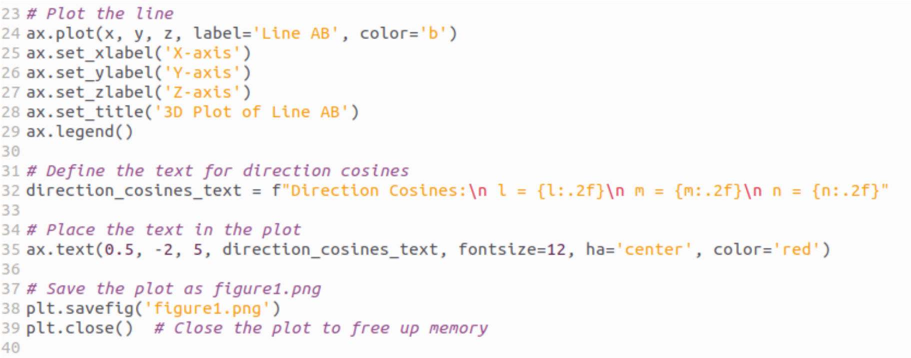
```c
#include <stdio.h>
#include <math.h>

int main() {
    // Direction ratios
    double a = 6.0, b = 2.0, c = 3.0;

    // Calculate the magnitude
    double magnitude = sqrt(a * a + b * b + c * c);

    // Calculate direction cosines
    double l = a / magnitude;
    double m = b / magnitude;
    double n = c / magnitude;

    // Output the results to a file
    FILE *file = fopen("output.txt", "w");
    if (file == NULL) {
        printf("Error opening file!\n");
        return 1;
    }

    fprintf(file, "Direction Cosines:\n");
    fprintf(file, "l = %lf\n", l);
    fprintf(file, "m = %lf\n", m);
    fprintf(file, "n = %lf\n", n);

    fclose(file);
    return 0;
}

```

Python Code



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

# Read direction cosines from the output.txt file
with open('output.txt', 'r') as file:
    lines = file.readlines()
    l = float(lines[1].split('=')[1].strip())
    m = float(lines[2].split('=')[1].strip())
    n = float(lines[3].split('=')[1].strip())

# Define a range of t values for plotting the line
t = np.linspace(-1, 1, 100)  # Adjust the range as needed

# Parametric equations of the line
x = 1 + 12 * t  # Assuming the line passes through (1, -2, 3)
y = -2 + 2 * t
z = 3 + 3 * t

# Create a 3D plot
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

```

Python Code



```python
# Plot the line
ax.plot(x, y, z, label='Line AB', color='b')
ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.set_zlabel('Z-axis')
ax.set_title('3D Plot of Line AB')
ax.legend()

# Define the text for direction cosines
direction_cosines_text = f"Direction Cosines:\n l = {l:.2f}\n m = {m:.2f}\n n = {n:.2f}"

# Place the text in the plot
ax.text(0.5, -2, 5, direction_cosines_text, fontsize=12, ha='center', color='red')

# Save the plot as figure1.png
plt.savefig('figure1.png')
plt.close()  # Close the plot to free up memory

```