Directional and Normal Vectors of a Line

Bhukya Prajwal Naik
AI24BTECH11005

November 13, 2024

Outline

1. Problem

2. Solution
    - Setting Up the Equation
    - Directional and Normal Vectors

3. Codes
    - C code for calculating vectors
    - Plotting the figure using Python

Problem Statement

Question: Find the directional and normal vectors of the line given by:

$$x + y = 4 \tag{1}$$

Setting Up the Equation

The equation of the line can be rearranged as follows:

$$x + y = 4 \tag{2}$$
$$y = 4 - x \tag{3}$$

We can express the line in vector form:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ 4 \end{pmatrix} + x \begin{pmatrix} 1 \\ -1 \end{pmatrix} \tag{4}$$

Here, we identify:

- A point on the line: $\begin{pmatrix} 0 \\ 4 \end{pmatrix}$
- The direction vector: $\begin{pmatrix} 1 \\ -1 \end{pmatrix}$

Directional and Normal Vectors

From the vector form, we find:

$$\text{Direction vector, } m = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \tag{5}$$

$$\text{Normal vector, } n = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \tag{6}$$

Element	**Mathematical Representation**
Given Line	$X = h + km$
Direction vector	$m = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$
Normal vector	$n = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

Table: Results

Visual Representation

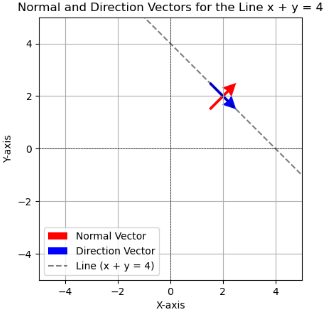

Figure: Directional and Normal Vectors of the Line

The code at `https://github.com/Prajwal-code77/EE1030/blob/main/presentations/codes/plot_vectors.py` verifies equations (5) and (6).

C Code

```c
// vector_calculator.c
#include <stdio.h>

int main() {
    // Normal vector for the line x + y = 4
    int normal_vector[] = {1, 1}; // Coefficients of x and y
    // Direction vector, perpendicular to the normal vector
    int direction_vector[] = {1, -1}; // Any vector perpendicular to the normal

    // Output the vectors
    printf("Normal Vector: [%d, %d]\n", normal_vector[0], normal_vector[1]);
    printf("Direction Vector: [%d, %d]\n", direction_vector[0], direction_vector[1]);

    // Calculate and print line points
    printf("Line Points: ");
    for (int x = -8; x <= 8; x++) {
        int y = 4 - x; // y = 4 - x
        printf("[%d, %.2f]", x, (float)y); // Format to two decimal places
        if (x < 8) {
            printf(", "); // Add comma except for the last point
        }
    }
    printf("\n");

    return 0;
}
```

Plotting the Figure using Python

```
import matplotlib.pyplot as plt

# Normal and direction vectors
normal_vector = [1, 1]
direction_vector = [1, -1]
line_points = [(x, 4 - x) for x in range(-8, 9)]

x_line, y_line = zip(*line_points)
shifted_origin = [2, 2]

# Create the plot
plt.quiver(*shifted_origin, *normal_vector, color=’r’, angles=’xy’, scale_units=’xy’,
     scale=1,
           width=0.01, headwidth=5, label=’Normal Vector’, pivot=’middle’)
plt.quiver(*shifted_origin, *direction_vector, color=’b’, angles=’xy’, scale_units=’xy’,
     scale=1,
           width=0.01, headwidth=5, label=’Direction Vector’, pivot=’middle’)

plt.plot(x_line, y_line, ’k--’, label=’Line (x + y = 4)’, alpha=0.5)  # Dashed line

plt.xlim(-5, 5)
plt.ylim(-5, 5)
plt.axhline(0, color=’black’, linewidth=0.5, ls=’--’)
plt.axvline(0, color=’black’, linewidth=0.5, ls=’--’)
plt.grid()
plt.title(’Normal and Direction Vectors for the Line x + y = 4’)
plt.xlabel(’X-axis’)
plt.ylabel(’Y-axis’)
plt.legend()
plt.gca().set_aspect(’equal’, adjustable=’box’)
plt.show()
```