4.2.9

EE25BTECH11006 - ADUDOTLA SRIVIDYA

september 12,2025

Question

Find the direction and normal vector for the line

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

Theoretical Solution

The line can be written as

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

Let

$$\mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix}, \quad \mathbf{n}^{\mathbf{T}} = \begin{pmatrix} 1 & 1 \end{pmatrix}, \quad c = 4 \tag{3}$$

Thus, the line equation is

$$\mathbf{n}^{\mathbf{T}}\mathbf{x} = c \tag{4}$$

where $\mathbf{n}$ is the normal vector.

Direction Vector

The direction vector of the line can be found by observing the normal vector.

$$\mathbf{m} = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \tag{5}$$

This is true because if the director vector is represented as

$$\mathbf{m} = \begin{pmatrix} 1 \\ m \end{pmatrix} \tag{6}$$

then the normal vector can be represented as

$$\mathbf{n} = \begin{pmatrix} -m \\ 1 \end{pmatrix} \tag{7}$$

Verification

This can be verified by the following equation:

$$\mathbf{n^T m} = 0 \tag{8}$$

$$\begin{pmatrix} 1 & 1 \end{pmatrix} \begin{pmatrix} -1 \\ 1 \end{pmatrix} = 0 \tag{9}$$

Final Answer

- Normal vector: $\mathbf{n} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$
- Direction vector: $\mathbf{m} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$

C code

```c
#include <stdio.h>
int dot_product(int a[2], int b[2]) {
return a[0]*b[0] + a[1]*b[1];
}
int is_orthogonal(int a[2], int b[2]){
return dot_product(a,b) == 0;
}
double line_equation(double x) {
return (4.0 - 1.0*x)/1.0;
}
```

cpython code

```
import sys
import ctypes
import numpy as np
import matplotlib.pyplot as plt
# Load C library
c_lib = ctypes.CDLL('./formula.so')
c_lib.line_vectors.argtypes = [
    ctypes.POINTER(ctypes.c_float),
    ctypes.POINTER(ctypes.c_float),
    ctypes.c_float, ctypes.c_float
]
c_lib.line_vectors.restype = None
```

cpython code

```
# For line: x + y = 4
a, b, c = 1.0, 1.0, 4.0
normal = np.zeros(2, dtype=np.float32)
direction = np.zeros(2, dtype=np.float32)

c_lib.line_vectors(
    normal.ctypes.data_as(ctypes.POINTER(ctypes.c_float)),
    direction.ctypes.data_as(ctypes.POINTER(ctypes.c_float)),
    ctypes.c_float(a), ctypes.c_float(b)
)

print("Normal Vector:", normal)
print("Direction Vector:", direction)
```

cpython code

```
# Plot line
x = np.linspace(-1, 6, 100)
y = (c - a*x)/b
plt.plot(x, y, label="x+y=4")
# Plot normal at (2,2)
P = np.array([2, 2])
plt.quiver(P[0], P[1], normal[0], normal[1], angles='xy',
    scale_units='xy', scale=1, color='r', label='Normal')
plt.quiver(P[0], P[1], direction[0], direction[1], angles='xy',
    scale_units='xy', scale=1, color='g', label='Direction')
plt.xlabel('$x$')
plt.ylabel('$y$')
plt.legend()
plt.grid(True)
plt.axis('equal')
plt.savefig("figs/Plot_P.png")
plt.show()
```

pure python code

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

# Define the line equation x + y = 4
x_vals = np.linspace(-2, 6, 400)
y_vals = 4 - x_vals

# Normal and direction vectors
n = np.array([1, 1])
d = np.array([1, -1])
```

pure python code

```
# Point on the line
point_on_line = np.array([2, 2])

# Plot the line
plt.plot(x_vals, y_vals, label='Line: $x + y = 4$', color='blue')

# Normal vector
plt.quiver(point_on_line[0], point_on_line[1], n[0], n[1],
           angles='xy', scale_units='xy', scale=1, color='red',
               label='Normal Vector')

# Direction vector
plt.quiver(point_on_line[0], point_on_line[1], d[0], d[1],
           angles='xy', scale_units='xy', scale=1, color='orange',
                label='Direction Vector')
```

pure python code

```
# Annotate
plt.scatter(point_on_line[0], point_on_line[1], color='black')
plt.annotate('Point on line (2, 2)', (2, 2), textcoords="offset
    points",
             xytext=(10,-10), ha='center')
plt.annotate('Normal [1, 1]', (3, 3), textcoords="offset points",
             xytext=(10,10), color='red')
plt.annotate('Direction [1, -1]', (3, 1), textcoords="offset
    points",
             xytext=(10,-20), color='orange')
```

pure python code

```
# Axes
ax = plt.gca()
ax.spines['left'].set_position('zero')
ax.spines['bottom'].set_position('zero')
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')

plt.xlabel('$x$')
plt.ylabel('$y$')
plt.legend(loc='best')
plt.grid(True)
plt.axis('equal')

plt.savefig('fig6.png')
plt.show()
```

Plot

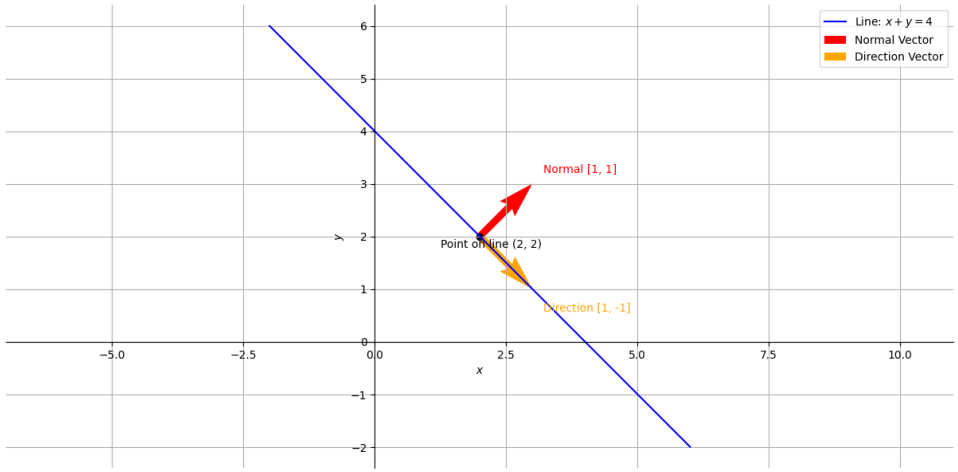

Figure: Line $x + y = 4$ with normal and direction vectors