4.2.16

EE25BTECH11018 - Darisy Sreetej

October 1 , 2025

Question

Find the direction and normal vector for the line

$$F = \frac{9}{5}C + 32 \tag{1}$$

Theoretical Solution

The line can be written as:

$$y = \frac{9}{5}x + 32 \tag{2}$$

$$5y - 9x = 160 \tag{3}$$

This equation can be expressed in terms of matrices
Let

$$\mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix} \tag{4}$$

$$\mathbf{n}^{\mathbf{T}} = \begin{pmatrix} -9 & 5 \end{pmatrix} \tag{5}$$

$$c = 160 \tag{6}$$

The line equation can be written as:

$$\mathbf{n}^{\mathbf{T}}\mathbf{x} = c \tag{7}$$

Where **n** is the normal vector of the given line

Direction Vector

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

$$\mathbf{m} = \begin{pmatrix} 5 \\ 9 \end{pmatrix} \tag{8}$$

This is true because if the director vector is represented as

$$\mathbf{m} = \begin{pmatrix} 1 \\ m \end{pmatrix} \tag{9}$$

then the normal vector can be represented as

$$\mathbf{n} = \begin{pmatrix} -m \\ 1 \end{pmatrix} \tag{10}$$

Verification

This can be verified by the following equation:

$$\mathbf{n}^{\mathbf{T}}\mathbf{m} = 0 \tag{11}$$

$$\begin{pmatrix} -9 & 5 \end{pmatrix} \begin{pmatrix} 5 \\ 9 \end{pmatrix} = 0 \tag{12}$$

Final Answer

1. Normal vector: $\mathbf{n} = \begin{pmatrix} -9 \\ 5 \end{pmatrix}$
2. Direction vector: $\mathbf{m} = \begin{pmatrix} 5 \\ 9 \end{pmatrix}$

C Code

```c
#include <stdio.h>

// Calculate dot product of two 2D vectors
int dot_product(int a[2], int b[2]) {
    return a[0]*b[0] + a[1]*b[1];
}

// Check if vectors are orthogonal (dot product = 0)
int is_orthogonal(int a[2], int b[2]) {
    return dot_product(a, b) == 0;
}

// Given the x-coordinate, calculate the corresponding y on the
    line
double line_equation(double x) {
    return (9.0*x)/5.0 + 32.0;
}
```

Python + C Code

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

lib = ctypes.CDLL("./line.so")

lib.dot_product.argtypes = [ctypes.POINTER(ctypes.c_int), ctypes.
    POINTER(ctypes.c_int)]
lib.dot_product.restype = ctypes.c_int

lib.is_orthogonal.argtypes = [ctypes.POINTER(ctypes.c_int),
    ctypes.POINTER(ctypes.c_int)]
lib.is_orthogonal.restype = ctypes.c_int

lib.line_equation.argtypes = [ctypes.c_double]
lib.line_equation.restype = ctypes.c_double

normal_vector = (ctypes.c_int * 2)(-9, 5)
direction_vector = (ctypes.c_int * 2)(5, 9)
```

Python + C code

```python
vector_origin = np.array([0, 32]) # Example: a point on the line

dp = lib.dot_product(normal_vector, direction_vector)
print(f"Dot product of n and m: {dp}")
if lib.is_orthogonal(normal_vector, direction_vector):
    print("The vectors are orthogonal (as expected).")
else:
    print("The vectors are NOT orthogonal.")

# Use the full x-range for your plot limits
x_min, x_max = -20, 30
x_vals = np.array([x_min, x_max])
y_vals = [lib.line_equation(float(x)) for x in x_vals]

plt.style.use('seaborn-v0_8-whitegrid')
plt.figure(figsize=(8, 8))

plt.plot(x_vals, y_vals, label='Line: 5y - 9x = 160', color='blue
    ', zorder=1)
```

Python + C code

```
plt.quiver(vector_origin[0], vector_origin[1],
           direction_vector[0], direction_vector[1],
           angles='xy', scale_units='xy', scale=1,
           color='green', label='Direction Vector', zorder=2)

plt.quiver(vector_origin[0], vector_origin[1],
           normal_vector[0], normal_vector[1],
           angles='xy', scale_units='xy', scale=1,
           color='red', label='Normal Vector', zorder=2)

plt.plot(vector_origin[0], vector_origin[1], 'o', color='purple',
     markersize=8,
         label='Vector Origin (0, 32)')

plt.title('Line with Direction and Normal Vectors')
plt.xlabel('x-axis')
plt.ylabel('y-axis')
plt.axis('equal')
```

Python + C code

```
plt.legend()
plt.grid(True)
plt.xlim(x_min, x_max)
plt.ylim(0, 60)


plt.show()
```

Python code

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

normal_vector = np.array([-9, 5])
direction_vector = np.array([5, 9])

print(f"Normal Vector (n): {normal_vector}")
print(f"Direction Vector (m): {direction_vector}")

dot_product = np.dot(normal_vector, direction_vector)
print(f"Dot product of n and m: {dot_product}")
if np.isclose(dot_product, 0):
    print("The vectors are orthogonal (as expected).")
else:
    print("The vectors are NOT orthogonal (something is wrong).")

def line_equation(x):
    return (9 * x) / 5 + 32
```

Python code

```
# Use x values covering the plotting range
x_vals = np.linspace(-20, 30, 100)
y_vals = line_equation(x_vals)

vector_origin = np.array([0, 32])

plt.style.use('seaborn-v0_8-whitegrid')
plt.figure(figsize=(8, 8))

plt.plot(x_vals, y_vals, label='Line: 5y - 9x = 160', color='blue
    ', zorder=1)

plt.quiver(vector_origin[0], vector_origin[1],
           direction_vector[0], direction_vector[1],
           angles='xy', scale_units='xy', scale=1,
           color='green', label='Direction Vector', zorder=2)
```

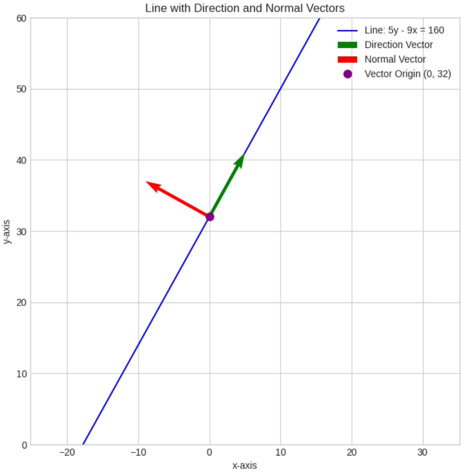
Python code

```
plt.quiver(vector_origin[0], vector_origin[1],
           normal_vector[0], normal_vector[1],
           angles='xy', scale_units='xy', scale=1,
           color='red', label='Normal Vector', zorder=2)

plt.plot(vector_origin[0], vector_origin[1], 'o', color='purple',
     markersize=8, label='Vector Origin (0, 32)')

plt.title('Line with Direction and Normal Vectors')
plt.xlabel('x-axis')
plt.ylabel('y-axis')
plt.axis('equal')
plt.legend()
plt.grid(True)
plt.xlim(-20, 30)
plt.ylim(0, 60)
plt.show()
```

Plot



Line with Direction and Normal Vectors
Line: 5y - 9x = 160
Direction Vector
Normal Vector
Vector Origin (0, 32)
y-axis
x-axis
60
50
40
30
20
10
0
−20
−10
0
10
20
30