4.2.16

Bhargav - EE25BTECH11013

September 6, 2025

Question

Find the direction and normal vector for the line

$$2 + 3y = 7x \tag{1}$$

Theoretical Solution

The line can be written as

$$7x - 3y = 2 \tag{2}$$

Let

$$\mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix}, \quad \mathbf{n}^{\mathbf{T}} = \begin{pmatrix} 7 & -3 \end{pmatrix}, \quad c = 2 \tag{3}$$

Thus, the line equation is

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

where **n** is the normal vector.

Direction Vector

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

$$\mathbf{m} = \begin{pmatrix} 3 \\ 7 \end{pmatrix} \tag{5}$$

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:

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

$$\begin{pmatrix} 7 & -3 \end{pmatrix} \begin{pmatrix} 3 \\ 7 \end{pmatrix} = 0 \tag{9}$$

Final Answer

1. Normal vector: $\mathbf{n} = \begin{pmatrix} 7 \\ -3 \end{pmatrix}$
2. Direction vector: $\mathbf{m} = \begin{pmatrix} 3 \\ 7 \end{pmatrix}$

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 (7.0*x - 2.0)/3.0;
}
```

```
import ctypes
import numpy as np
import matplotlib.pyplot as plt
lib = ctypes.CDLL(./libcode.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)(7, -3)
direction_vector = (ctypes.c_int * 2)(3, 7)
vector_origin = np.array([2, 4])
dp = lib.dot_product(normal_vector, direction_vector)
print(fDot product of n and m: {dp})
```

Python + C code

```
if lib.is_orthogonal(normal_vector, direction_vector):
    print(The vectors are orthogonal (as expected).)
else:
    print(The vectors are NOT orthogonal.)
x_vals = np.linspace(-5, 7, 100)
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: 7x - 3y = 2', 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)
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)
```

Python + C code

```
plt.plot(vector_origin[0], vector_origin[1], 'o', color='purple',
     markersize=8,
         label='Vector Origin (2, 4)')
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(-5, 10)
plt.ylim(-5, 10)
plt.show()
```

Python code

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

normal_vector = np.array([7, -3])

direction_vector = np.array([3, 7])

print(fNormal Vector (n): {normal_vector})
print(fDirection Vector (m): {direction_vector})

dot_product = np.dot(normal_vector, direction_vector)
print(fDot 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).)
```

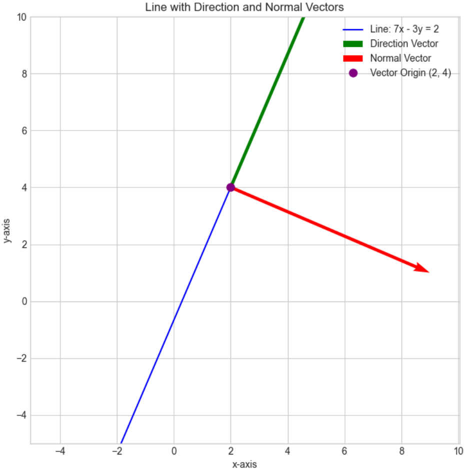
Python code

```
def line_equation(x):
    return (7 * x - 2) / 3

x_vals = np.linspace(-5, 7, 100)
y_vals = line_equation(x_vals)

vector_origin = np.array([2, 4])

plt.style.use('seaborn-v0_8-whitegrid')
plt.figure(figsize=(8, 8))
plt.plot(x_vals, y_vals, label='Line: 7x - 3y = 2', 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 (2, 4)')
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(-5, 10)
plt.ylim(-5, 10)
plt.savefig(/Users/bhargavkrish/Desktop/BackupMatrix/
    ee25btech11013/matgeo/4.2.16/figs/Figure_1.png)
plt.show()
```

Plot



Line with Direction and Normal Vectors
Line: 7x - 3y = 2
Direction Vector
Normal Vector
Vector Origin (2, 4)
x-axis
y-axis
−4
−2
0
2
4
6
8
10