4.2.12

Anshu kumar ram - EE25BTECH11009

September 6, 2025

Question

Find the direction and normal vector for the line

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

Theoretical Solution

The line can be written as

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

Let

$$\mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix}, \quad \mathbf{n}^{\mathbf{T}} = \begin{pmatrix} 2 & 1 \end{pmatrix}, \quad c = 3 \tag{3}$$

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.

$$\mathbf{m} = \begin{pmatrix} 1 \\ -2 \end{pmatrix} \tag{5}$$

This is true because if the direction 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:

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

$$\begin{pmatrix} 2 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ -2 \end{pmatrix} = 0 \tag{9}$$

Final Answer

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

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

Python + C Code

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

# Load the shared library
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 and direction vectors
```

Python + C code

```python
# Dot product check
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.")

# Evaluate line y = 3 - 2x
x_vals = np.linspace(-5, 7, 100)
y_vals = [lib.line_equation(float(x)) for x in x_vals]

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

#  Updated label
plt.plot(x_vals, y_vals, label='Line: 2x + y = 3', color='blue',
```

```
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)
```

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

# Normal and direction vectors for 2x + y = 3
normal_vector = np.array([2, 1])
direction_vector = np.array([1, -2])

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).")

# Line equation: y = 3 - 2x
def line_equation(x):
```

Python code

```
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: 2x + y = 3', 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,
```

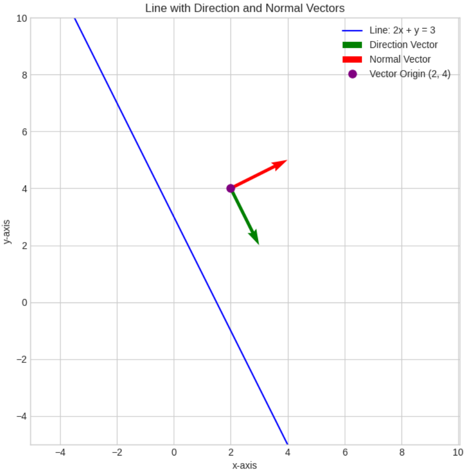
Python code

```
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_2.png")
plt.show()
```

Plot



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